

< HTML 5 >

saturngod

HTML 5

ကျွန်တော် အရင်က HTML စာအုပ်ရေးပြီးတဲ့ နောက်ပိုင်းမှာ HTML 5 ထွက်လာပါတယ်။ HTML စာအုပ်ကို ပြန်ပြီးတော့ update လုပ်ပြီး HTML 5 ထုတ်မယ်လို့ စဉ်းစားမိတယ်။ HTML စာအုပ်ကို ပြန်ဖတ်ကြည့်တော့ လိုက်တော့ လိုအပ်နေတာတွေ တော်တော်များများ တွေ့တယ်။ ဒါနဲ့ HTML 5 ကို အစကနေ ပြန်ရေးမယ် ဖြစ်တယ်။ ဒီစာအုပ်ဟာ Wiley က ထုတ်တဲ့ Mike Wooldridge ရေးသားထားတဲ့ Teach Yourself VISUALLY HTML5 စာအုပ် က ခေါင်းစဉ်တွေ ကို အခြေခံထားပါတယ်။ ကျောရိုးကို သာ အခြေခံထားပြီး အထဲမှာ ရေးသားထားတဲ့ စာတွေ တော်တော်များများ ကတော့ ကျွန်တော် ကိုယ်ပိုင် ရေးသား ထားတာပါ။ အချို့ စာ အနည်းငယ် သာ ဘာသာပြန်ပြီး ထည့်သွင်းထားပါတယ်။ အဲဒီ စာအုပ်ကို အခြေခံရတဲ့ အဓိက က တော့ CSS အပြင် Javascript အခြေခံပါ ပါဝင် တဲ့အတွက် HTML 5 ကို လေ့လာတဲ့ သူတွေ အနေနဲ့ CSS , Javascript အခြေခံတွေကို ပါ သိသွားပါလိမ့်မယ်။ ဒီစာအုပ်မှာ HTML 5 နဲ့ JavaScript ကို အဓိက တွဲသုံးရတဲ့ web storage , Drag/Drop , Geolocation , Web Workers စတာတွေတွေ့ မပါဝင် သေးပါဘူး။ ဒီစာအုပ်ဟာ စတင်လေ့လာသူတွေအတွက် ရေးသားထားတဲ့ စာအုပ်ဖြစ်အတွက် ကြောင့် HTML ကို စတင်လေ့လာသူတွေ အတွက် အသုံးဝင်မယ့် စာအုပ် တစ်အုပ် ဖြစ်မယ်လို့ မျှော်လင့်ပါတယ်။

HTML ဆိုတာ ဘာလဲ

HTML ဆိုတာကတော့ website တွေတည်ဆောက်ရာမှာ အခြေခံ အကျဆုံးနဲ့ web developer ဖြစ် ချင်သူတွေ ပထမဆုံး မဖြစ်မနေ သိထားသင့်တဲ့ language တစ်ခုပါ။ ယနေ့အခါမှာ Internet ဟာ အလွန်ကျယ်ပြန့်လာပြီး လူတိုင်း နီးပါး အသုံးပြုနိုင်ကြပါတယ်။ ကျွန်တော်တို့ဟာလည်း Internet ပေါ်မှာ ကိုယ်ပိုင် စာမျက်နှာတွေ အလွယ်တကူ တည်ဆောက်နိုင်ပါတယ်။ Internet အကြောင်းကိုတော့ အကျယ်ရှင်း မနေတော့ပါဘူး။ အလွယ်ဆုံးပြောရရင်တော့ အင်တာနက်ဆို တာက ကမ္ဘာဘာတစ်ခုလုံး ချိတ်ဆက်ထားတဲ့ ကြီးမားတဲ့ ကွန်ယက်ကြီးပါ။ အဲဒီ ကွန်ယက်ကြီး မှာ သတင်းအချက်အလက်တွေ ဖော်ပြဖို့ ဖလှယ်ဖို့အတွက် Web Page တွေ ရေးသားရပါတယ်။ အဲလို မျိုး web page တွေ ရေးဆွဲဖို့ရာအတွက် HTML က အဓိက ကျလှပါတယ်။ ရေးထားတဲ့ web page တွေကို web browser တွေကနေ တဆင့် ကြည့်ရှုနိုင်ပါတယ်။ web browser တွေ ကတော့ Internet Explorer , Firefox , Chrome , Opera , Safari စတာတွေ ဖြစ်ပါတယ်။ HTML ဟာ tag တွေနဲ့ တည်ဆောက်ထားတဲ့ အတွက်ကြောင့် HTML အကြောင်း မရေးခင်မှာ HTML သမိုင်း အနည်းအကျဉ်း tag တွေ အကြောင်းကို လေ့လာကြည့်ရအောင်ဗျာ။

HTML သမိုင်း

1989 ခုနှစ်က Tim Berners Lee နဲ့ Rober Calliau ဟာ Research Document တွေကို ပိုမိုမြန်ဆန် ကောင်းမွန်မယ့် နည်းလမ်းကောင်းတွေကို ရှာဖွေခဲ့ပါတယ်။ ၂ နှစ်လောက်ကြာတော့ Berners Lee ဟာ browser တွေက ကြည့်ရှုဖတ်သားနိုင်ပြီး tag များနဲ့ ဖွဲ့စည်းတည်ဆောက်ထားတဲ့ နည်း ပညာတစ်ခုကို ပြုလုပ်နိုင်ခဲ့ပါတယ်။ ထို့နောက် အဲဒီ tag တွေထဲမှာ Hyper-Link နဲ့ document တစ် ခုနဲ့ တစ်ခု ချိတ်ဆက်နိုင်တဲ့ tag တွေကိုပါ ထည့်သွင်းနိုင်ခဲ့ပါတယ်။ Berners Lee ရဲ့ နည်းပညာ ကတော့ Hyper Text Markup Language ဆိုတဲ့ HTML ပါပဲ ။

Tag ဆိုတာ ဘာလဲ

Tag ဆိုတာကတော့ ပုံမှန် စာသားတွေနဲ့ HTML code တွေကို ခွဲခြား လိုရအောင် လုပ်ထားပေး တယ်လို့ အလွယ်ဆိုလို့ရပါတယ်။ ဥပမာ။ **This is bold** ကို web browser မှာ ကြည့်ရင် စာလုံးက bold ဖြစ်နေပါမယ်။ **က အဖွင့်ပြီးတော့** ကတော့ အပိတ်ပါ။ ပုံမှန် အားဖြင့် tag တွေက အဖွင့်နဲ့ အပိတ်ပါပါတယ်။ အပိတ်တွေကတော့ / လေးနဲ့ ဆုံးပါတယ်။

HTML ရေးဖို့ အတွက် Editor

HTML အတွက် Editor တွေ IDE တွေအများကြီး ရှိပါတယ်။ ကျွန်တော် လက်ရှိ သုံးနေတာ ကတော့ Mac OSX မှာ Text Mate ကို အသုံးပြုနေပါတယ်။ စတင်လေ့လာတဲ့ သူတွေ အနေနဲ့ dreamweaver လိုမျိုး IDE ထက် notepad , notepad++ , sublime text လိုမျိုး editor တွေကို အသုံးပြုနိုင်ပါတယ်။

Introduction to Creating and Editing Web Pages

Text Editor

Windows Users

Windows Key+R ကို နှိပ် လိုက်ပါ။ Run Dialog Box တက်လာပါမည်။ ထို့နောက် notepad လို့ရိုက် လိုက်ပါ။ ပြီးလျှင် OK နှိပ်ပါ။ သို့မဟုတ် Notepad++ သွင်းထားပါ။ Notepad ++ ကို ဖွင့်ပြီး HTML စတင် အသုံးပြုနိုင်ပါသည်။

Linux Users

Ubuntu အသုံးပြုသူများအနေနဲ့ HTML အတွက် gedit ကို အသုံးပြုနိုင်ပါသည်။ Applications -> Accessories -> Text Editor ကို ဖွင့်ပါ။ သို့မဟုတ် Alt+F2 နှိပ်ပြီး gedit လို့ ရိုက်လိုက်ပါ။

Mac Users

Mac ကို အသုံးပြုသူများအနေနဲ့ VS Code, Sublime Text, Zed, CotEditor စတဲ့ text editor တစ်ခုခုကို အသုံးပြုနိုင်ပါသည်။

All Platforms

Platform အားလုံးအတွက် Visual Studio Code (VS Code) သို့မဟုတ် Sublime Text ကို အသုံးပြု နိုင်ပါသည်။ ဒီ စာအုပ်ထဲမှာ ကျွန်တော် VS Code နဲ့ ရေးသားပြသသွားပါမယ်။

HTML File

အောက်ကလို code ကို နှစ်သက်ရာ editor တွင် ရေးသားလိုက်ပါ။ ကျွန်တော်ကတော့ VS Code တွင် ရေးသားပြသသွားပါမယ်။

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>This is title</title>
</head>
<body>
This is body
```

```
</body>
</html>
```

```
untitled
1 <html>
2 <head>
3 <title>This is title</title>
4 </head>
5 <body>
6 This is body
7 </body>
8 </html>
```

Figure: 1-2

HTML ရေးသားရာမှာ `<html>` နဲ့ စပြီး `</html>` နဲ့ ပြန်ပိတ်ရပါတယ်။ အဲဒီကြားမှာ html code တွေရေးရပါတယ်။ `<head>` နဲ့ `</head>` ကြားတွင် Title ထည့်ထားတာကို တွေ့မှာပါ။ Title သည် Browser ၏ ထိပ် bar တွင် တွေ့မြင်ရန်ဖြစ်သည်။ `<body>` နဲ့ `</body>` ကြားမှာ မိမိ နှစ်သက်ရာ စာထည့်သွင်းရေးနိုင်ပါတယ်။

ကျွန်တော် တို့ Save သွင်းတဲ့ အခါမှာ .html နဲ့ သိမ်းပေးဖို့ လိုပါတယ်။ ကျွန်တော် အခု hello.html နာမည် နဲ့ သိမ်းလိုက်ပါတယ်။ Notepad နဲ့ ရေးသူများ အနေနဲ့ file save လုပ်တဲ့ အခါမှာ Save As Type ကို All File လုပ်ခဲ့ဖို့လိုပါတယ်။ ဥပမာ ကို အောက်က ပုံမှာ ကြည့်နိုင်ပါတယ်။

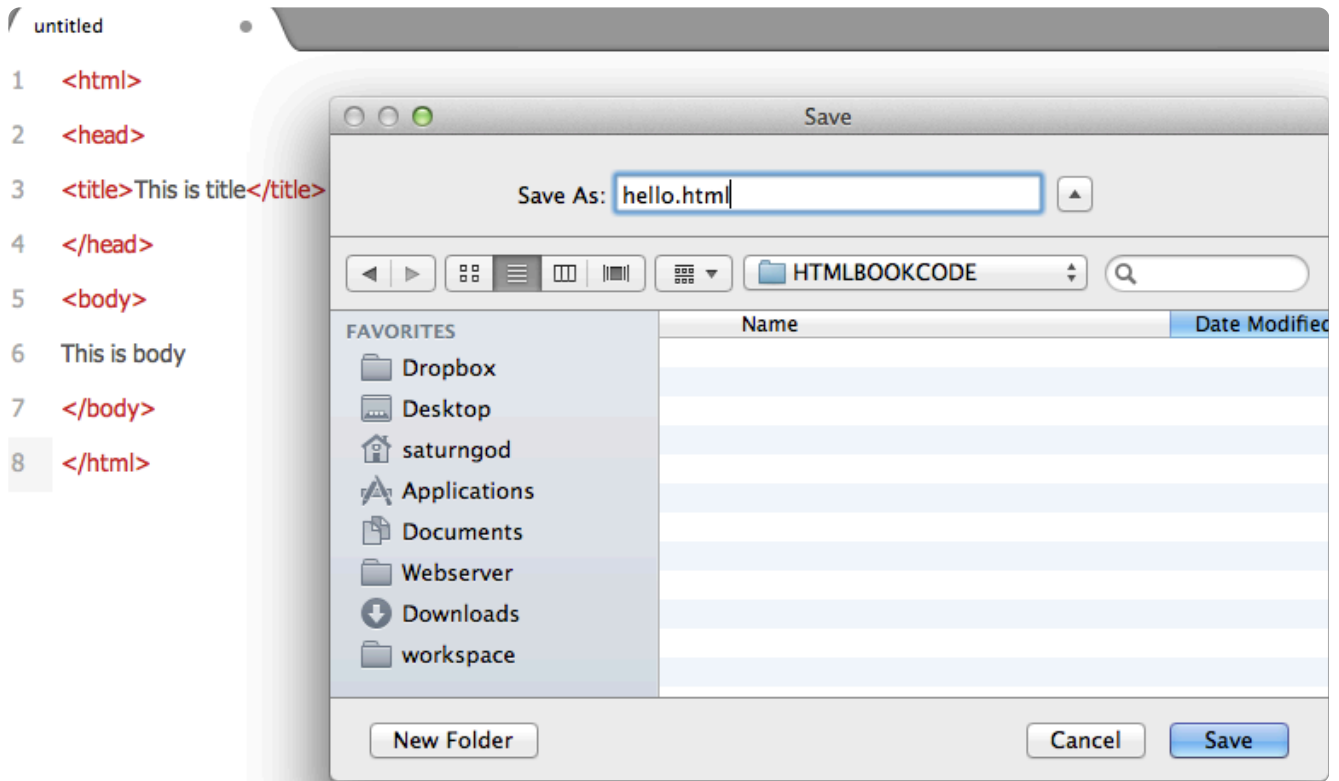


Figure 1-2. Save HTML

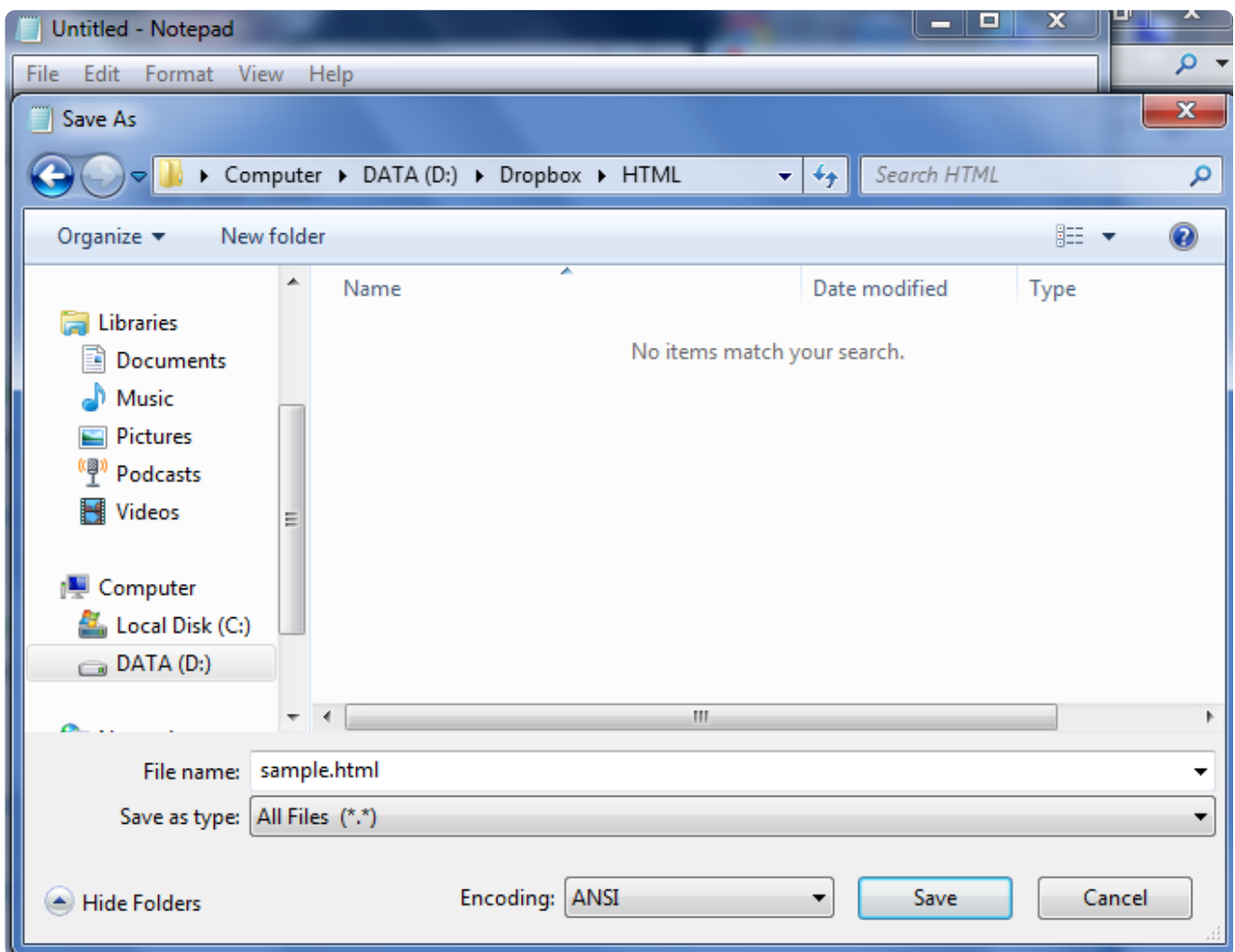


Figure 1-3. Notepad ကို အသုံးပြုပြီး Save သိမ်းခြင်း

ပြီးလျှင် သိမ်းထားတဲ့ File ကို double click နှိပ်ပြီး run လိုက်ပါ။ Web Browser တစ်ခု အနေနဲ့ တက်လာပါမယ်။ ကျွန်တော်ကတော့ Chrome Web Browser ကို သုံးထားတဲ့ အတွက် Chrome နဲ့ တက်လာပါတယ်။ အဲဒီ အခါမှာတော့ အောက်ကလို ပုံ မြင်ရပါမယ်။

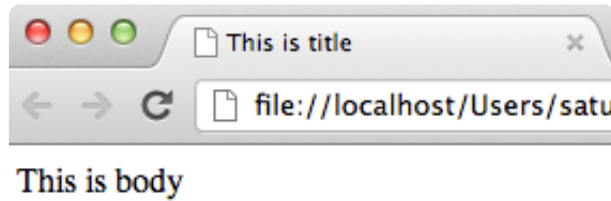


Figure 1-4.

Tab မှာတော့ <title> နဲ့ </title> ကြားမှာ ရေးထားခဲ့တဲ့ စာလေးကို တွေ့နိုင်ပါတယ်။ Browser ထဲမှာတော့ <body> နဲ့ </body> ကြားမှာ ရေးထားခဲ့တဲ့ စာတွေ ပေါ်မှာပါ။

HTML Structure

HTML ရေးတော့မယ်ဆိုရင် အောက်ဖော်ပြပါ code လိုမျိုး structure ရှိပါတယ်။

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
  </head>
  <body>
  </body>
</html>
```

ကျွန်တော်တို့ <head> လိုမျိုးကိုတော့ head tag လို့ခေါ်ပါတယ်။ <body> ဆိုရင်တော့ body tag ပေါ့။ နောက်ပြီး <!DOCTYPE html> ဆိုတာကတော့ doctype declaration ပါ။ ဘာ document type ဖြစ်သလဲဆိုတာ ကြေငြာထားတာပါ။ ဖြစ်နိုင်တဲ့ document တွေကတော့ SGML , HTML , DTD စတာတွေ ရှိပါတယ်။ အခုနောက်ပိုင်း modern HTML (HTML5 လို့လည်း ခေါ်ကြပါတယ်) မှာ တော့ ကျွန်တော်တို့ အနေနဲ့

```
<!DOCTYPE html>
```

ဆိုပြီး သုံးလိုက်ရုံပါပဲ။ HTML 4 မှာ ကျွန်တော်တို့တွေဟာ DTD အတွက် အောက်ကလိုမျိုး code တွေ သုံးခဲ့ပါတယ်။ အခုအခါမှာတော့ ဒီ section ကို historical reference အနေနဲ့သာ သိထားရင် လုံလောက်ပါတယ်။

HTML 4.01 DTDs

HTML 4.01 Strict DTD ဟာ အသုံးပြုခွင့်မပေးတော့တဲ့ tag တွေကို အသုံးမပြုတော့ပါဘူး။ ဥပမာ `font` လိုမျိုး tag တွေပေါ့။ Framesets ကိုလည်း အသုံးပြုခွင့်မပေးပါဘူး။

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

HTML 4.01 Transitional လိုမျိုး doctype ကတော့ အသုံးပြုခွင့်မပေးတော့တဲ့ tag တွေကိုလည်း အသုံးပြုနိုင်ပါတယ်။ ဥပမာ `font` tag ။ သို့ပေမယ့် Frameset ကိုလည်း အသုံးပြုခွင့်မပေးပါဘူး။

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

HTML 4.01 Transitional ကို support လုပ်ပြီး frameset ပါ အသုံးပြုခွင့်ပေးချင်ရင်တော့ အောက်လိုမျိုး Doctype ကြေငြာဖို့ လိုပါတယ်။

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

XHTML 1.0 Strict ကိုတော့ အောက်ကလို ရေးသားနိုင်ပါတယ်။ သူ့ကို ကြေငြာထားရင်တော့ အသုံးမပြုတဲ့ Tag တွေကို အသုံးပြုခွင့်မပေးတော့ပါဘူး။ နောက်ပြီးတော့ Frameset လည်း အသုံးပြုခွင့်မရှိပါဘူး။ နောက်ပြီးတော့ XML format ကို လိုက်နာဖို့ လိုပါတယ်။

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

XHTML 1.0 Transitional ကတော့ အသုံးပြုခွင့်မပေးထားတဲ့ Font Tag လိုမျိုး အသုံးပြုခွင့်ရပါတယ်။ Frameset တော့ အသုံးပြုခွင့်မရှိပါဘူး။ နောက်ပြီး XML format ကို လိုက်နာဖို့ လိုပါတယ်။

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

XHTML 1.0 Frameset ကတော့ အသုံးပြုခွင့်မပေးထားတဲ့ Font Tag လိုမျိုး အသုံးပြုခွင့်ရပါတယ်။ Frameset လည်း အသုံးပြုနိုင်ပါတယ်။ နောက်ပြီး XML format ကို လိုက်နာဖို့ လိုပါတယ်။

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

XHTML 1.1 ကတော့ XHTML 1.0 Strict နဲ့ အတူတူပါပဲ။ သို့မှားကတော့ module တွေ ထပ်ထည့် ခွင့် ပြုထားပါတယ်။

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

ယခု စာအုပ်မှာတော့ ကျွန်တော်တို့တွေဟာ modern HTML doctype ကိုပဲ အသုံးပြုပြီး ရေးသွား မှာ ဖြစ်တဲ့ အတွက်တော့ page တွေမှာ

```
<!DOCTYPE html>
```

ကိုပဲ ကြေငြာသွားမှာပါ။ WHATWG HTML Living Standard ဟာ ဆက်လက် update ဖြစ်နေတဲ့ standard ဖြစ်ပေမယ့် doctype ကတော့ ဒီပုံစံတိုတိုလေးပဲ ဖြစ်ပါတယ်။

View Source

ကျွန်တော်တို့တွေ အနေနဲ့ အခြား သူတွေရေးထားတဲ့ HTML code ကို browser ကနေ တဆင့် ဖတ်ရှုနိုင်ပါတယ်။

Edge / Modern Browsers

Edge တွင် source ကြည့်ချင်ရင်တော့ right click နှိပ်ပြီး View Page Source ကို နှိပ်လိုက်ပါ။ သို့မဟုတ် **Ctrl + U** ကို နှိပ်နိုင်ပါသည်။

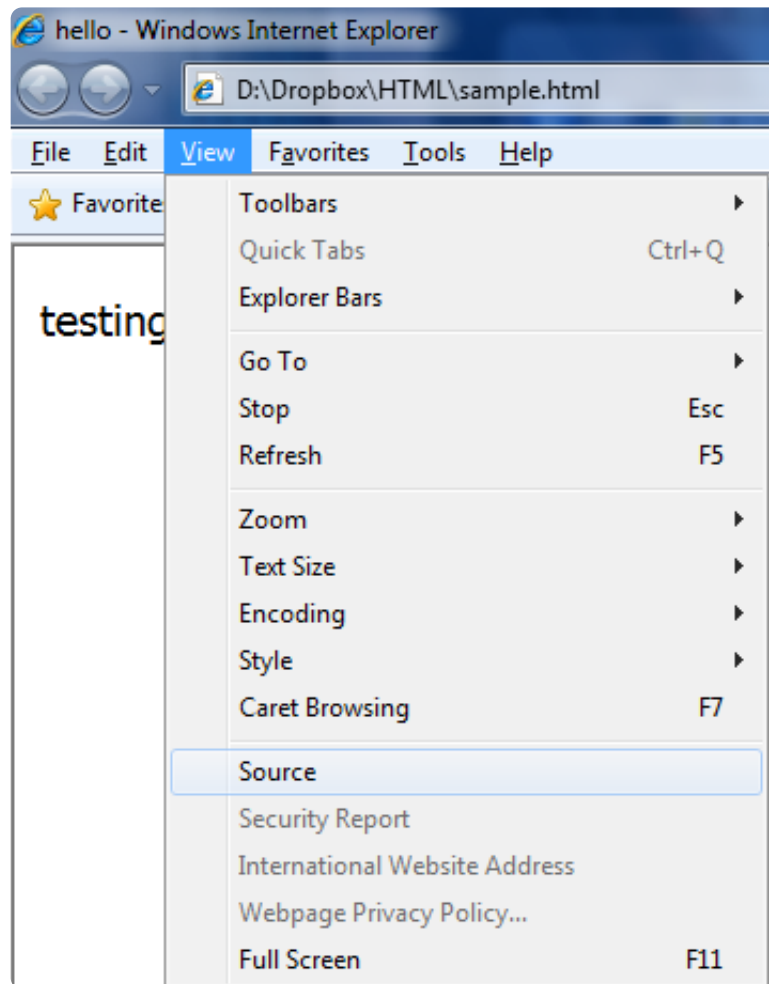


Figure 1-5. View Source

ထိုအခါ အောက်ကလို မျိုး မိမိရေးထားတဲ့ Code ကို ပြန်မြင်ရပါလိမ့်မယ်။

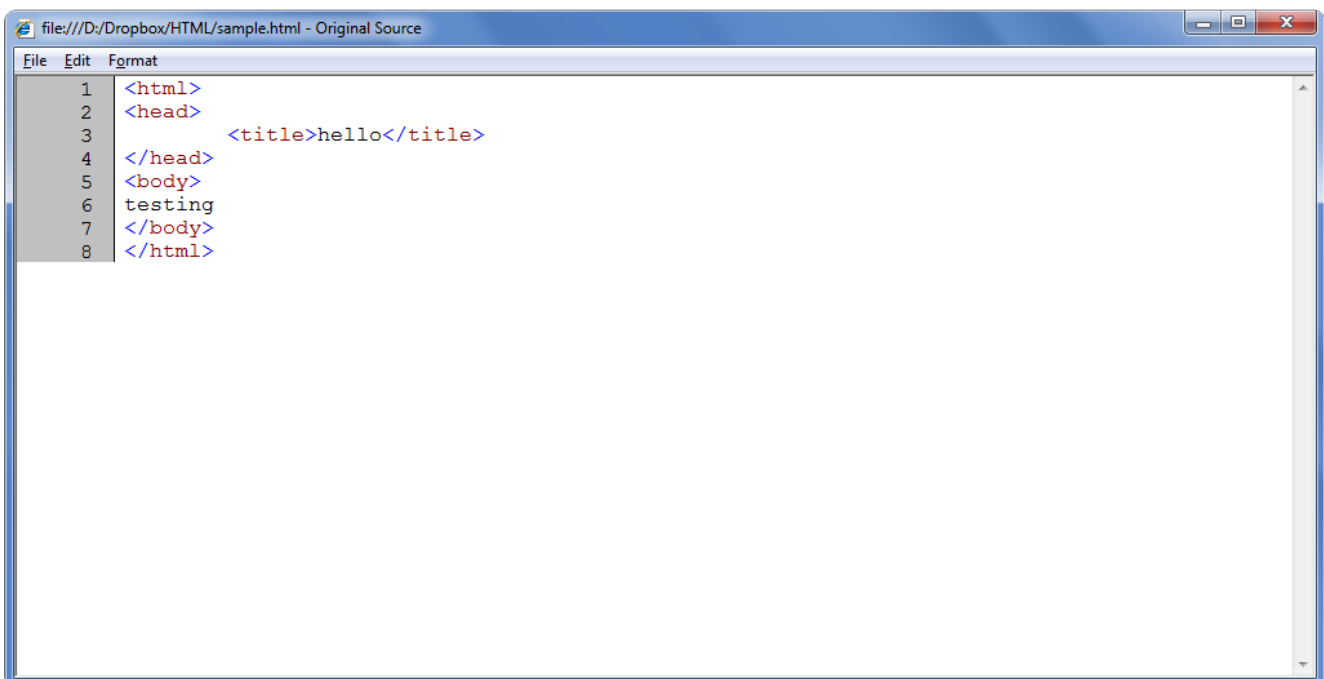


Figure 1-6. Browser မှ မြင်ရသော Source Code

Chrome

Chrome မှာဆိုရင်တော့ Menu > Tools > View Source ကို နှိပ်လိုက်ပါ။

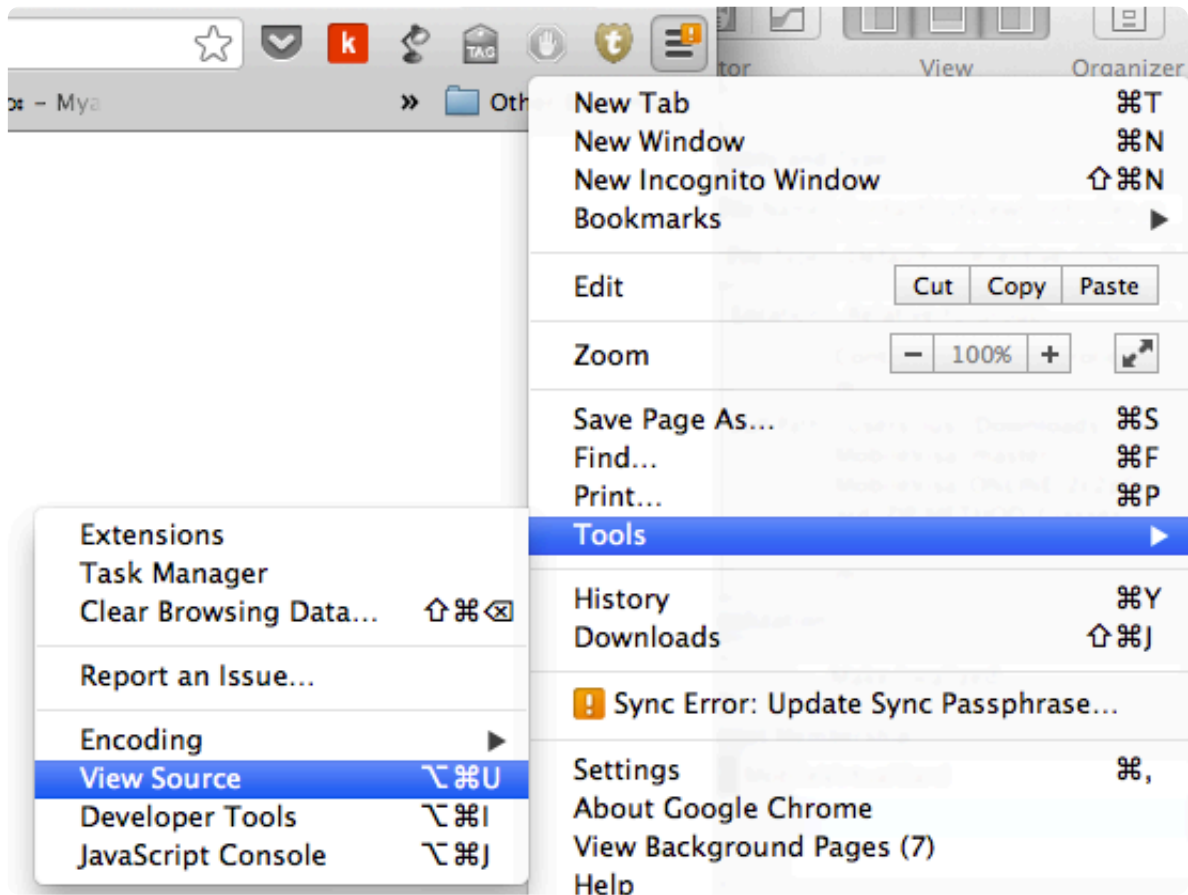


Figure 1-7. Chrome View Source

အောက်ဖော်ပြပါ ပုံအတိုင်း မြင်ရပါလိမ့်မယ်။

```

1 <!DOCTYPE html>
2 <html>
3     <head>
4     </head>
5     <body>
6         Hello World!
7     </body>
8 </html>
9

```

Figure 1-8. Source Code at Chrome

Firefox

Firefox မှာ ဆိုရင်တော့ Tool > Web Developers > View Source ကို နှိပ်လိုက်ပါ။

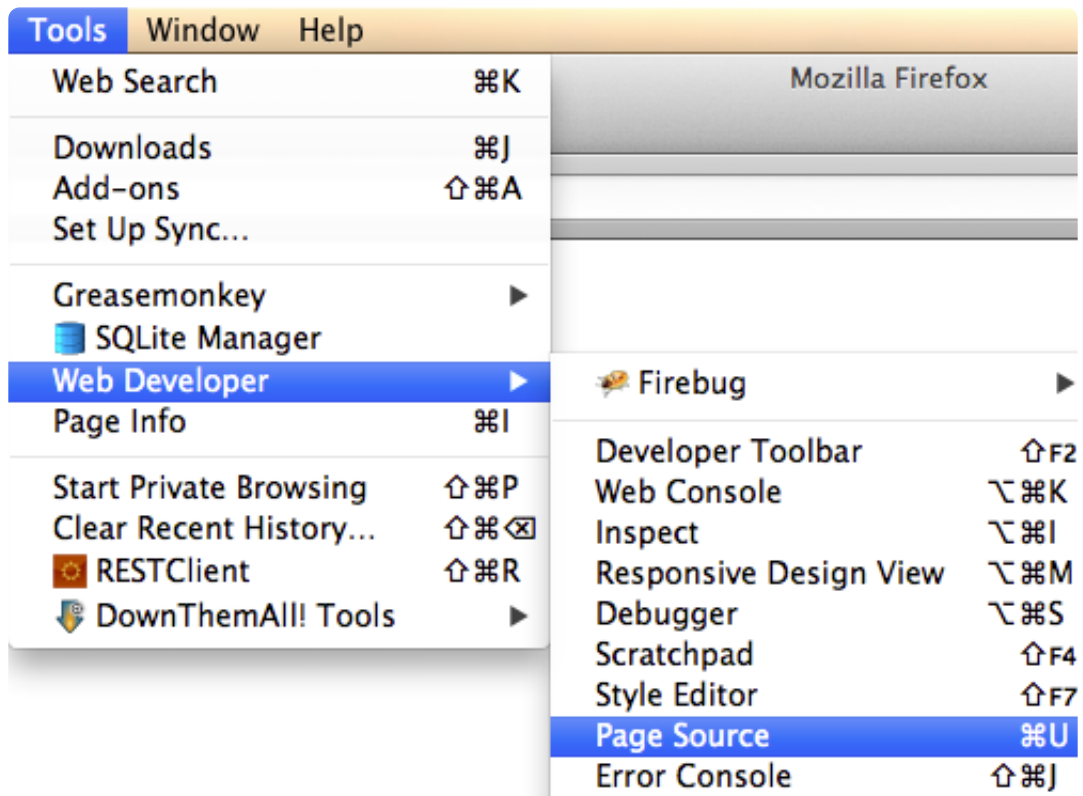


Figure 1-9. View Source Menu

အောက်ဖော်ပြပါ ပုံအတိုင်း မြင်ရပါလိမ့်မယ်။

 A screenshot of a source code window titled 'Source of: file:///Use...'. The window displays the following HTML code:


```

1 <!DOCTYPE html>
2 <html>
3   <head>
4   </head>
5   <body>
6     Hello World!
7   </body>
8 </html>
9
  
```

Figure 1-10. Source Code

Adding and Modifying Text

ကျွန်တော်တို့ Chapter 1 မှာတုန်းကတော့ HTML အကြောင်း အနည်းငယ်မျှ သိပြီးပါပြီ။ အခု ကျွန်တော်တို့တွေ HTML မှာ စာတွေ ထည့်မယ်။ Bold , Italic စတာတွေကို ထည့်မယ်။ Paragraph တွေ ဖန်တီးတာတွေကို အခု chapter မှာ တွေ့နိုင်ပါလိမ့်မယ်။

Meta

HTML တစ်ခု ကို ဖန်တီးတဲ့ အတွက် ကျွန်တော်တို့တွေ Header ပိုင်းမှာ meta ကို လည်း အသုံးပြု ကြပါတယ်။ Meta ဆိုတာကတော့ လက်ရှိ webpage ရဲ့ information တွေ ထည့်သွင်းဖော်ပြထား တာပါ။

```
<!DOCTYPE html>
<html>
<head>
  <title>This is title</title>
  <meta name="description" content="HTML5 book">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">
  <meta name="author" content="saturngod">
  <meta charset="UTF-8">
</head>
<body>
  Hello World!
</body>
</html>
```

အထက်ပါ code ကို ကြည့်ရင် လက်ရှိ webpage ရဲ့ author က saturngod လို့ရေးထားတာတွေမှာ ပါ။ **description** ကတော့ search engine တွေက ကိုယ့် website ကို ဖော်ပြတဲ့အခါမှာ မြင်ရနိုင် တဲ့ text ပါ။ **keywords** meta ကိုတော့ အခုခေတ် search engine အများစုက ranking အတွက် မ သုံးတော့ပါဘူး။ ဒါကြောင့် SEO အတွက် အရေးကြီးတာက စာသားအကြောင်းအရာကောင်း ခြင်း၊ heading structure မှန်ခြင်း၊ page title နဲ့ description ကိုရှင်းလင်းအောင်ရေးခြင်းတွေ ဖြစ်ပါ တယ်။

Create a New Paragraph

Paragraph တစ်ခု ကို တည်ဆောက်ဖို့ ရန်အတွက် ကျွန်တော်တို့ `<p>` tag ကို အသုံးပြုပါမယ်။ အောက်က ဥပမာ လေး ကို တချက်ကြည့်လိုက်ပါ။

```
<!DOCTYPE html>
```

```

<html>
  <head>
    <title>This is title</title>
    <meta name="description" content="HTML5 book">
    <meta name="keywords" content="HTML,CSS,XML,JavaScript">
    <meta name="author" content="saturngod">
    <meta charset="UTF-8">
  </head>
  <body>
    <p>HyperText Markup Language (HTML) is the main markup language for
displaying web pages and other information that can be displayed in a web browser.</p>
    <p>The purpose of a web browser is to read HTML documents and compose
them into visible or audible web pages. The browser does not display the HTML tags, but
uses the tags to interpret the content of the page.</p>
  </body>
</html>

```

ကျွန်တော်တို့ တွေ စာတွေ ရေးသားရာမှာ အပိုဒ်တွေကို ခွဲပြီး ရေးသားလိုတဲ့ အခါမှာ `<p>` ကို အသုံးပြုကြပါတယ်။
အထက်ပါ code ကို browser မှာ ဖွင့်ကြည့်လိုက်ရင် အောက်ကလို မြင်ရပါလိမ့်မယ်။

HyperText Markup Language (HTML) is the main markup language for displaying web pages and other information that can be displayed in a web browser.

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

Figure 2-1. Paragraph

Add a Line Break

Paragraph မှာ စာတွေကို တစ်ကြောင်း တည်း မရေးလိုပဲ လိုင်းတွေ ခြားခြင် ရင်တော့ Line break tag `
` ကို အသုံးပြုနိုင်ပါတယ်။

```

<!DOCTYPE html>
<html>
  <head>
    <title>This is title</title>
    <meta name="description" content="HTML5 book">
    <meta name="keywords" content="HTML,CSS,XML,JavaScript">
    <meta name="author" content="saturngod">
    <meta charset="UTF-8">
  </head>
  <body>
    <p>HyperText Markup Language (HTML) is the main markup language for
displaying web pages <br/> and other information that can be displayed in a web browser.
</p>
  </body>

```

```
</html>
```

ဆိုတဲ့ code မှာ `
` ထည့်ထားတာ တွေ့မှာပါ။

HyperText Markup Language (HTML) is the main markup language for displaying web pages and other information that can be displayed in a web browser.

Figure 2-2. Line Breaking

ပုံမှာ ပြထားတဲ့ အတိုင်း web pages နဲ့ and မှာ စာ တစ်ကြောင်း ဆင်းထားတာကို တွေ့ရမှာပါ။

Add a Line Break Opportunity

ကျွန်တော် တို့ အနေနဲ့ နောက်ထပ် line break အမျိုးအစား နောက်တစ်ခု `<wbr/>` ဆိုတာ ရှိပါ သေးတယ်။ သူက `
` နဲ့ မတူတာကတော့ line break လိုအပ်တဲ့ အချိန်မှသာ line break လုပ် သွားပေးတာပါ။ မလိုတဲ့ အချိန်မှာတော့ စာလုံးကို ဆက်ပြီး ဖော်ပြပေးပါတယ်။

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is title</title>
    <meta name="description" content="HTML5 book">
    <meta name="keywords" content="HTML,CSS,XML,JavaScript">
    <meta name="author" content="saturngod">
    <meta charset="UTF-8">
  </head>
  <body>
    <p>HyperText Markup Language (HTML) is the main markup language for
    displaying web pages and other information that can be displayed in a web browser.
    Yayyyyy<wbr/>!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!</p>
  </body>
</html>
```

code ထဲမှာ `<wbr/>` ကို Yayyyyy နဲ့ !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! ကြားမှာ ထည့်ထားပေးပါတယ်။

HyperText Markup Language (HTML) is the main markup language for displaying web pages and other information that can be displayed in a web browser. Yayyyyy!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Figure 2-3 : စာကြောင်းဆန့်၍ Line Break မလုပ်ခြင်း

screen မှာ ဖော်ပြလို့ ရရင်တော့ Line break မလုပ်ပဲ screen သေးသွားပြီး မဆန့်တော့ရင်တော့ အလိုလို line break လုပ်ပေးပါတယ်။

HyperText Markup Language (HTML) is the main markup language for displaying web pages and other information that can be displayed in a web browser. Yayyyyyy
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Figure 2-4 : စာကြောင်းမဆန့်၍ Line Break လုပ်ခြင်း

Add a Horizontal Rule

HTML မှာ မျဉ်းကို အလျားလိုက်ထားပြီးတော့ တစ်ခု နဲ့ တစ်ခုကို ပိုင်းဖြတ်လိုရင်တော့ ကျွန်တော်တို့ `<hr/>` ကို အသုံးပြုနိုင်ပါတယ်။

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is title</title>
    <meta name="description" content="HTML5 book">
    <meta name="keywords" content="HTML,CSS,XML,JavaScript">
    <meta name="author" content="saturngod">
    <meta charset="UTF-8">
  </head>
  <body>
    <p>HyperText Markup Language (HTML) is the main markup language for
displaying web pages and other information that can be displayed in a web browser.</p>
    <hr/>
    <p>The purpose of a web browser is to read HTML documents and compose
them into visible or audible web pages. The browser does not display the HTML tags, but
uses the tags to interpret the content of the page.</p>
  </body>
</html>
```

ဆိုတဲ့ Code မှာ `<hr/>` လို့ ရေးထားတာကို တွေ့နိုင်ပါတယ်။ အဲဒီ code လေးကို browser မှာ ဆိုရင် အောက်ကလို မြင်ရပါမယ်။

HyperText Markup Language (HTML) is the main markup language for displaying web pages and other information that can be displayed in a web browser.

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

Figure 2-5. Line Break

အထက်ပါ ပုံထဲမှာ စာ ပိုဒ် ၂ ခု ကြားမှာ မျဉ်းကြောင်းလေး ကို တွေ့ရပါလိမ့်မည်။


```

<head>
  <title>This is title</title>
  <meta name="description" content="HTML5 book">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">
  <meta name="author" content="saturngod">
  <meta charset="UTF-8">
</head>
<body>
  <p>HyperText Markup Language <strong>(HTML)</strong> is the main markup
  language for displaying web pages and other information that can be displayed in a web
  browser.</p>

</body>

</html>

```

အထက်ပါ code မှာ (HTML) ကို ကျွန်တော်တို့ **bold** လုပ်ထားတာ တွေ့နိုင်ပါတယ်။
အဲဒီ code လေးကို ကြည့်လိုက်ရင် အောက်ကလို မြင်ရပါလိမ့်မယ်။

**HyperText Markup Language (HTML) is the main markup
language for displaying web pages and other information
that can be displayed in a web browser.**

Figure 2-7. (HTML) ကို bold ဖြစ်နေသည်။

Make Italic Text

စာလုံးတွေကို emphasize လုပ်ချင်တဲ့ အခါမှာတော့ `` နဲ့ `` ကြားမှာ ထည့်ရေးရပါတယ်။ အဓိပ္ပါယ်မပြောင်းဘဲ visual အနေနဲ့သာ italic ဖြစ်ချင်ရင် CSS ကို အသုံးပြုတာ ပိုကောင်းပါတယ်။ `<i>` tag ကလည်း modern HTML မှာ valid ဖြစ်နေသေးပေမယ့် emphasis အတွက်တော့ `` ကို သုံးပါ။

```

<!DOCTYPE html>
<html>
  <head>
    <title>This is title</title>
    <meta name="description" content="HTML5 book">
    <meta name="keywords" content="HTML,CSS,XML,JavaScript">
    <meta name="author" content="saturngod">
    <meta charset="UTF-8">
  </head>
  <body>
    <p>HyperText Markup Language <em>(HTML)</em> is the main markup language
    for displaying web pages and other information that can be displayed in a web browser.
  </p>

  </body>

</html>

```

အထက်ပါ code ကို browser မှာ ကြည့်လျှင် အောက်ပါ အတိုင်း မြင်ရပါလိမ့်မယ်။

HyperText Markup Language (*HTML*) is the main markup language for displaying web pages and other information that can be displayed in a web browser.

Figure 2-8: (HTML) ကို italic ဖြစ်နေသည်။

Preformatted Text

line breaks တွေ space တွေ အားလုံးကို ရိုက်ထည့်လိုက်သလို အလုပ်လုပ်စေချင်ရင်တော့ preformatted text ကို အသုံးပြုနိုင်ပါတယ်။ `<pre>` နဲ့ `</pre>` ကြားထဲမှာ ရေးထားလိုက်ရင် ရပါတယ်။ ပုံမှန် အားဖြင့် web browser က line break တွေ space တွေ အားလုံးကို ပြသတဲ့ အခါမှာ အသုံးမပြုပါဘူး။ ကျွန်တော် တို့ အပေါ်မှာ ရေးခဲ့တဲ့ ဥပမာလို space တွေ အတွက် သီးသန့် special code ကို အသုံးပြုခဲ့ရပါတယ်။ HTML code မှာ လိုင်း ခြားထားပေမယ့် `
` အသုံးမပြုထားရင်တော့ browser မှာ လိုင်း မခြားပါဘူး။ အောက်က code လေးကို တချက်လောက် လေ့လာကြည့်ရအောင်။

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is title</title>
    <meta name="description" content="HTML5 book">
    <meta name="keywords" content="HTML,CSS,XML,JavaScript">
    <meta name="author" content="saturngod">
    <meta charset="UTF-8">
  </head>
  <body>
    <pre>
Name:      Saturngod
Book:      HTML5
Year:      2013
    </pre>
    <pre>
Name:      Saturngod
Book:      HTML5
Year:      2013
    </pre>
  </body>
</html>
```

အဲဒီ code လေးကိုတော့ အောက်ကလို မြင်ရပါလိမ့်မယ်။

```
Name: Saturngod
Book: HTML5
Year: 2013
```

```
Name: Saturngod
Book: HTML5
Year: 2013
```

Figure 2-9.

ကျွန်တော် Tab တွေနဲ့ ခြားထားတဲ့အတွက်ကြောင့် browser မှာ ပေါ်တဲ့ အခါလည်း ကျွန်တော် ရေးထားတဲ့ အတိုင်းပေါ်ပါတယ်။ အထက်ပါ code မှာ `<pre>` နဲ့ `</pre>` ကြားက text တွေမှာ စ ခြင်း space မခြားထားတာကို တွေ့နိုင်ပါတယ်။ ဒုတိယ code မှာတော့ space တွေ ထည့်ထားပါတယ်။ ဒါကြောင့် ပြတဲ့ အခါမှာလည်း space တွေ ခြားပြီး ထွက်လာပါတယ်။

Add Heading

Page တစ်ခု ရေးသားတဲ့ အခါမှာ ကျွန်တော်တို့တွေ ခေါင်းစဉ် တွေ ထည့်ရတာတွေ ရှိပါတယ်။ ဥပမာ အခု စာအုပ်မှာဆိုရင် ကျွန်တော်တို့တွေ `<h1>Chapter 2: Adding and Modifying Text</h1>` ဆိုပြီး Chapter တွေရဲ့ heading ကို အသုံးပြုထားပါတယ်။ Heading level 6 ခု အထိ ရှိပါတယ်။ h1,h2,h3 ကနေ h6 ထိပေါ့။

```
<!DOCTYPE html>
<html>
<head>
  <title>This is title</title>
  <meta name="description" content="HTML5 book">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">
  <meta name="author" content="saturngod">
  <meta charset="UTF-8">
</head>
<body>
<h1>Chapter 1</h1>
<h2>sub header 1.1</h2>
<h3>header h3</h3>
<h4>header h4</h4>
<h5>header h5</h5>
<h6>header h6</h6>
</body>
</html>
```

ဆိုတဲ့ code လေးကို run လိုက်ရင် အောက်လို မျိုး မြင်ရပါလိမ့်မယ်။

Chapter 1

sub header 1.1

header h3

header h4

header h5

header h6

Figure 2-10 : Header Level

H1 က အကြီးဆုံးဖြစ်နေပြီး တဖြည်းဖြည်း သေးသွားတာကို တွေ့နိုင်ပါတယ်။ h1 ကိုတော့ main header အဖြစ် အသုံးပြုပြီးတော့ level အလိုက် header တွေ ကို ခွဲထားတာပါ။

Add Block Quote

ကျွန်တော်တို့တွေ စာတွေနဲ့ အခြား quote တွေကို သီးခြားခွဲပြ ချင်တဲ့ အခါမှာတော့ block quotes ကို အသုံးပြုပါတယ်။ `<blockquote>` ကို အသုံးပြုတဲ့ အခါမှာ စာတွေက ဘယ်ဘက်ကနေ အနည်းငယ်ခွာ ပြီး ပြပေးပါလိမ့်မယ်။

```

<!doctype html>
<html>
<head>
  <title>Quote</title>
</head>
<body>
<h1>Quotes</h1>
<blockquote>Live as if you were to die tomorrow. Learn as if you were to live forever.
</blockquote>
Mahatma Gandhi

<blockquote>The more that you read, the more things you will know. The more that you
learn, the more places you'll go.</blockquote>
Dr. Seuss, I Can Read With My Eyes Shut!

<blockquote>The love of learning, the sequestered nooks, And all the sweet serenity of
books</blockquote>
Henry Wadsworth Longfellow

</body>
</html>

```

ဆိုတဲ့ code လေးကို browser မှာ ကြည့်လိုက်ရင် အောက်ကလို မြင်ရပါလိမ့်မယ်။

Quotes

Live as if you were to die tomorrow . Learn as if you were to live forever.

Mahatma Gandhi

The more that you read, the more things you will know . The more that you learn, the more places you'll go.

Dr. Seuss, I Can Read With My Eyes Shut!

The love of learning, the sequestered nooks, And all the sweet serenity of books

Henry Wadsworth Longfellow

Figure 2-11

`<blockquote>` သုံးထားတဲ့ စာတွေဟာ ဘယ်ဘက်ကနေ ကွာနေပြီးတော့ သီးသန့် ဖြစ်နေတာကို တွေ့နိုင်ပါတယ်။

Add Small Text

ကျွန်တော်တို့တွေ စာကြောင်း အသေးလေးတွေကို အသုံးပြုချင်တဲ့ အခါမှာတော့ `<small>` ကို အသုံးပြုရပါတယ်။ အထူးသဖြင့် ကျွန်တော်တို့တွေ copyright လိုမျိုး စာသားတွေ မှာ အသုံးများပါတယ်။

```

<!DOCTYPE html>
<html>
<head>
  <title>Small</title>
  <meta name="description" content="HTML5 book">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">
  <meta name="author" content="saturngod">
  <meta charset="UTF-8">
</head>
<body>
<h1>HTML</h1>
<p>HyperText Markup Language (HTML) is the main markup language for creating web pages and other information that can be displayed in a web browser.</p>
<p>HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like &lt;html&gt;), within the web page content. HTML tags most commonly come in pairs like &lt;h1&gt; and &lt;/h1&gt;, although some tags, known as empty elements, are unpaired, for example <img>. The first tag in a pair is the start tag, the second tag is the end tag (they are also called opening tags and closing tags). In between these tags web designers can add text, tags, comments and other types of text-based content.
</p>
<small>Copyright 2012 Wikipedia</small>

```

```
</body>
</html>
```

အထက်ပါ code ကို browser မှာ ကြည့်လိုက်ရင် အောက်ကလို မြင်ရပါမယ်။ small tag နဲ့ ရေးထားတဲ့ စာလေးတွေက သေးနေတာကို တွေ့နိုင်ပါတယ်။

HTML

HyperText Markup Language (HTML) is the main markup language for creating web pages and other information that can be displayed in a web browser.

HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>), within the web page content. HTML tags most commonly come in pairs like <h1> and </h1>, although some tags, known as empty elements, are unpaired, for example . The first tag in a pair is the start tag, the second tag is the end tag (they are also called opening tags and closing tags). In between these tags web designers can add text, tags, comments and other types of text-based content.

Copyright 2012 Wikipedia

Figure 2-12

Comment

ကျွန်တော်တို့တွေ HTML မှာ မှတ်ချက်တွေကို ထည့်သွင်းလို့ရပါတယ်။ comment ဆိုတာကတော့ web browser မှာ ဖော်မပြထားပဲ code ကိုပဲ ရည်ညွှန်းထားတာကို ဆိုလိုတာပါ။ အောက်က code ကို ကြည့်လိုက်ပါ။

```
<!DOCTYPE html>
<html>
<head>
  <title>Small</title>
  <meta name="description" content="HTML5 book">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">
  <meta name="author" content="saturngod">
  <meta charset="UTF-8">
</head>
<body>
<h1>HTML</h1>
<p>HyperText Markup Language (HTML) is the main markup language for creating web pages
and other information that can be displayed in a web browser.</p>

<!-- Another Paragraph Start -->

<p>HTML is written in the form of HTML elements consisting of tags enclosed in angle
```

brackets (like `<html>`), within the web page content. HTML tags most commonly come in pairs like `<h1>` and `</h1>`, although some tags, known as empty elements, are unpaired, for example ``. The first tag in a pair is the start tag, the second tag is the end tag (they are also called opening tags and closing tags). In between these tags web designers can add text, tags, comments and other types of text-based content.

```

<!-- Small Tag for coyright -->

<small>Copyright 2012 Wikipedia</small>
</body>
</html>

```

အထက်ပါ code ကို ကြည့်လိုက်ရင် comment ၂ ခု ထည့်ထားတာကိုတွေ့နိုင်ပါတယ်။ comment က `<!--` နဲ့ စပြီးတော့ `-->` နဲ့ ဆုံးပါတယ်။ အဲဒီကြားထဲမှာ ရေးရပါတယ်။ အဲဒီ code ကို run ကြည့်ရင် အပေါ်က small text example အတိုင်းပဲ မြင်ရမှာပါ။ ဒါကြောင့် ကျွန်တော် example ပုံ မပြတော့ပါဘူး။ ကိုယ်တိုင် စမ်းကြည့်လိုက်ပါ။

Create a Numbered List

Number list ဆိုတာကတော့ 1,2,3,4 စတဲ့ list တွေပေါ့။ ကျွန်တော်တို့တွေ number list ကို ဖန်တီးချင်ရင်တော့ `` နဲ့ `` ကို အသုံးပြုရပါတယ်။ `` ကတော့ မိမိ ထည့်ချင်တဲ့ list ပေါ့။ အောက်က code လေးကို ကြည့်လိုက်ရင် ရှင်းသွားမှာပါ။

```

<!DOCTYPE html>
<html>
<head>
  <title>Small</title>
  <meta name="description" content="HTML5 book">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">
  <meta name="author" content="saturngod">
  <meta charset="UTF-8">
</head>
<body>
<h1>Order List</h1>
<ol type="A">
  <li>List 1 </li>
  <li>List 2 </li>
  <li>List 3 </li>
  <li>List 4 </li>
  <li>List 5 </li>
</ol>

<ol type="a">
  <li>List 1 </li>
  <li>List 2 </li>
  <li>List 3 </li>
  <li>List 4 </li>
  <li>List 5 </li>
</ol>

<ol type="I">
  <li>List 1 </li>
  <li>List 2 </li>

```

```

    <li>List 3 </li>
    <li>List 4 </li>
    <li>List 5 </li>
</ol>

<ol type="i">
  <li>List 1 </li>
  <li>List 2 </li>
  <li>List 3 </li>
  <li>List 4 </li>
  <li>List 5 </li>
</ol>

<ol type="1">
  <li>List 1 </li>
  <li>List 2 </li>
  <li>List 3 </li>
  <li>List 4 </li>
  <li>List 5 </li>
</ol>

<ol type="i" start="4">
  <li>List 4 </li>
  <li>List 5 </li>
  <li>List 6 </li>
  <li>List 7 </li>
  <li>List 8 </li>
</ol>
</body>
</html>

```

type ဆိုတာကတော့ အမျိုးအစားပါ။ type ကို မထည့်ထားခဲ့ရင် default အနေနဲ့ 1 ဖြစ်နေမှာပါ။ စချင်တဲ့ နေရာကို start နဲ့ အသုံးပြုပါတယ်။ `<ol type="i" start="4">` လို့ရေးထားတဲ့အတွက် 4 ကနေ စနေတာ တွေ့နိုင်ပါတယ်။

Order List

- A. List 1
- B. List 2
- C. List 3
- D. List 4
- E. List 5

- a. List 1
- b. List 2
- c. List 3
- d. List 4
- e. List 5

- I. List 1
- II. List 2
- III. List 3
- IV. List 4
- V. List 5

- i. List 1
- ii. List 2
- iii. List 3
- iv. List 4
- v. List 5

- 1. List 1
- 2. List 2
- 3. List 3
- 4. List 4
- 5. List 5

- iv. List 4
- v. List 5
- vi. List 6
- vii. List 7
- viii. List 8

Figure 2-13

Create a Bulleted List

```

<!DOCTYPE html>
<html>
<head>
  <title>Small</title>
  <meta name="description" content="HTML5 book">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">

```

```
<meta name="author" content="saturngod">
<meta charset="UTF-8">
</head>
<body>
<h1>Bulleted List</h1>
<ul style="list-style-type: square">
  <li>List 1 </li>
  <li>List 2 </li>
  <li>List 3 </li>
  <li>List 4 </li>
  <li>List 5 </li>
</ul>

<ul style="list-style-type: disc">
  <li>List 1 </li>
  <li>List 2 </li>
  <li>List 3 </li>
  <li>List 4 </li>
  <li>List 5 </li>
</ul>

<ul style="list-style-type: circle">
  <li>List 1 </li>
  <li>List 2 </li>
  <li>List 3 </li>
  <li>List 4 </li>
  <li>List 5 </li>
</ul>
</body>
</html>
```

Bulleted List

- List 1
- List 2
- List 3
- List 4
- List 5

- List 1
- List 2
- List 3
- List 4
- List 5

- List 1
- List 2
- List 3
- List 4
- List 5

Figure 2-14

အထက်က code မှာ `` ကိုပဲ အသုံးပြုထားတာ တွေ့မှာပါ။ HTML5 မှာ `type` attribute ကို `` အတွက် အသုံးမပြုတော့ပါဘူး။ Bullet ပုံစံ ပြောင်းချင်ရင် CSS ရဲ့ `list-style-type` ကို အသုံးပြုရပါမယ်။

```
<ul style="list-style-type:square">
  <li>List 1</li>
</ul>
```

Create Nested List

Order List ပဲ ဖြစ်ဖြစ် Bulleted List ပဲ ဖြစ်ဖြစ် ကျွန်တော်တို့တွေ အဆင့်ဆင့် ရေးလို့ရပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <title>Small</title>
  <meta name="description" content="HTML5 book">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">
  <meta name="author" content="saturngod">
  <meta charset="UTF-8">
</head>
<body>
<h1>Order List</h1>
<ol>
  <li>List 1</li>
  <li>List 2
    <ol>
      <li>List 1 </li>
      <li>List 2 </li>
      <li>List 3 </li>
      <li>List 4 </li>
      <li>List 5 </li>
    </ol>
  </li>
  <li>List 3 </li>
  <li>List 4 </li>
  <li>List 5 </li>
</ol>

<ul>
  <li>List 1</li>
  <li>List 2
    <ul>
      <li>List 1 </li>
      <li>List 2 </li>
      <li>List 3 </li>
      <li>List 4 </li>
      <li>List 5 </li>
    </ul>
  </li>
  <li>List 3 </li>
  <li>List 4 </li>
```

```

    <li>List 5 </li>
</ul>

</body>
</html>

```

မှာဆိုရင် ကျွန်တော်တို့တွေ `` ထဲမှာ ထပ်ပြီးတော့ `` နဲ့ ရေးထားတာကို တွေ့နိုင်ပါတယ်။

Nested List

1. List 1
 2. List 2
 1. List 1
 2. List 2
 3. List 3
 4. List 4
 5. List 5
 3. List 3
 4. List 4
 5. List 5
- List 1
 - List 2
 - List 1
 - List 2
 - List 3
 - List 4
 - List 5
 - List 3
 - List 4
 - List 5

Figure 2-15

Create Definition List

အဓိပ္ပာယ် ဖော်ပြတဲ့ စကားလုံးတွေ ဥပမာ။။ အဘိဓာန် လိုမျိုး စာလုံး နဲက အဓိပ္ပာယ် ဖော်ပြတဲ့ ကိစ္စတွေမှာ Definition List ကို အသုံးများပါတယ်။

```

<!DOCTYPE html>
<html>
<head>
  <title>Small</title>
  <meta name="description" content="HTML5 book">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">
  <meta name="author" content="saturngod">
  <meta charset="UTF-8">
</head>
<body>
<h1>What's in a Bag ?</h1>

```

```

<dl>
  <dt>Backpack</dt>
  <dd>A bag for carrying hiking supplies and lunch</dd>
  <dt>Fanny pack</dt>
  <dd>A bag for carrying a passport wallet, and camera</dd>
  <dt>Suitcase</dt>
  <dd>A bag for carrying clothes and toilets</dd>
  <dt>Briefcase</dt>
  <dd>A bag for carrying business documents, files and pens</dd>
</dl>
</body>
</html>

```

ဆိုတဲ့ code လေးကို အောက်ကလို ပေါ်ပါမယ်။

What's in a Bag ?

Backpack
A bag for carrying hiking supplies and lunch

Fanny pack
A bag for carrying a passport wallet, and camera

Suitcase
A bag for carrying clothes and toilets

Briefcase
A bag for carrying business documents, files and pens

Figure 2-16

`<dl>` နဲ့ စပြီးတော့ `</dl>` နဲ့ ဆုံးပါတယ်။ `<dt>` ကတော့ definition title ပေါ့။ `<dd>` ကတော့ definition အနေနဲ့ သုံးပါတယ်။

Abbreviation

အတိုချုံး စကားလုံးတွေကို အသုံးပြုချင်တဲ့ အခါမှာတော့ `<abbr>` tag ကို အသုံးပြုပါတယ်။

```

<!DOCTYPE html>
<html>
<head>
  <title>Small</title>
  <meta name="description" content="HTML5 book">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">
  <meta name="author" content="saturngod">
  <meta charset="UTF-8">
</head>
<body>
<h1>Medical Alert</h1>
<p>Ascending to high elevations can cause altitude sickness, including <abbr title="high altitude pulmonary edema">HAPE</abbr>.</p>
</body>
</html>

```

အပေါ်က code မှာ `title=` ဆိုတာကို တွေ့မှာပါ။ `title` ဆိုတာကတော့ mouse pointer စာပေါ်တင် လိုက်ရင် တွေ့ရမယ့် tooltip စာပါ။ အဲဒီ code လေးကို အောက်ကလို မြင်ရမှာပါ။

Medical Alert

Ascending to high elevations can cause altitude sickness, including HAPE.

high altitude pulmonary edema

Figure 2-17

Insert Date and Time

အချိန်နဲ့ နေ့စွဲတွေကို ဖော်ပြချင်တဲ့ အခါမှာတော့ `<time>` ကို ကျွန်တော် တို့တွေ အသုံးပြုပါ တယ်။ အောက်က code လေးကို တချက်ကြည့်လိုက်ပါ။

```
<!DOCTYPE html>
<html>
<head>
  <title>Small</title>
  <meta name="description" content="HTML5 book">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">
  <meta name="author" content="saturngod">
  <meta charset="UTF-8">
</head>
<body>
<h1>Snorkel Cruise Information</h1>
<p>The boat will leave the harbor at <time>8:30 a.m. on Sunday morning</time>.</p>
</body>
</html>
```

browser မှာတော့ ပုံမှန် စာပဲ ဖော်ပြနေမှာပါ။ `<time>` ကို သုံးထားတဲ့အတွက် ထူးထူးခြားခြား ပြောင်းလဲ သွားတဲ့ ပုံစံ မရှိပါဘူး။ အဲဒီ code လေးကို အခုလို ပြောင်းလဲပါမယ်။

```
<!DOCTYPE html>
<html>
<head>
  <title>Small</title>
  <meta name="description" content="HTML5 book">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">
  <meta name="author" content="saturngod">
  <meta charset="UTF-8">
</head>
<body>
<h1>Snorkel Cruise Information</h1>
<p>The boat will leave the harbor at <time datetime="2011-08-21T08:30:00">8:30 a.m. on
Sunday morning</time>.</p>
</body>
```

```
</html>
```

အထက်ပါ code မှာ `datetime="2011-08-21T08:30:00"` ဆိုတာကတော့ `YYYY-MM-DD` format နဲ့ ရေးထားပြီး T ရဲ့ အနောက်မှာတော့ `HH:MM:SS` ကို ထည့်သွင်းထားတာပါ။ browser မှာတော့ ဘာမှ မထူးပါဘူး။ ဒါပေမယ့် ကျွန်တော်တို့ javascript နဲ့ တွဲသုံးလို့ရပါတယ်။

`datetime` attribute ကို အမြဲသုံးဖို့လိုလားဆိုတော့ မလိုပါဘူး။ မထည့်သွင်းလည်း ရပါတယ်။

Insert Special Character

Special character တွေကို အသုံးပြုချင်တဲ့ အခါ သီးသန့် code တွေ နဲ့ ကျွန်တော်တို့တွေ အသုံးပြုရပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <title>Small</title>
  <meta name="description" content="HTML5 book">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">
  <meta name="author" content="saturngod">
  <meta charset="UTF-8">
</head>
<body>
<h1>M and B Adventure Travel &reg;</h1>
<p>For more than 10 years, M and B adventure travel has helped customers fulfill their dreams and conquer their fears.
We specialize in one-of-a-kind vaction package to the most destination in the world. Let us help you with your hotel bookings, helicopter tour , and shark-cage rentals, either online or in person.</p>
</body>
</html>
```

ဆိုတဲ့ code မှာ `®` ကို ® အနေနဲ့ မြင်ရမှာပါ။

M and B Adventure Travel ®

For more than 10 years, M and B adventure travel has helped customers fulfill their dreams and conquer their fears. We specialize in one-of-a-kind vaction package to the most destination in the world. Let us help you with your hotel bookings, helicopter tour , and shark-cage rentals, either online or in person.

Figure 2-18

Special characters တွေ ကို အောက်က ဇယားမှာ ကြည့်နိုင်ပါတယ်။

Description	Special Character	Code

Copyright	©	©
Registered trademark	®	®
Trademark	™	™
paragraph mark	¶	¶
nonbreaking space		
quotation mark	“	"
left angle quote	«	«
right angle quote	»	»
ampersand	&	&
inverted exclamation	¡	¡
inverted question mark	¿	¿
broken vertical bar		¦
section	§	§
not	¬	¬
acute accent	´	´
cedilla	¸	¸
bullet	•	•
capital N, tilde	Ñ	&NTilde;
small n, tilde	ñ	ñ
capital A, tilde	Ã	Ã
small a, tilde	ã	ã
capital A, grave accent	À	À
small a, grave accent	à	à
capital O, slash	Ø	Ø
small o, slash	ø	ø
em dash	—	—
en dash	–	–
micro	μ	µ
macron	—	¯

superscript one	1	¹
superscript two	2	²
superscript three	3	³
one-half fraction	1/2	½
one-fourth fraction	1/4	¼
three-fourths fraction	3/4	¾
degree	°	°
multiply	×	×
division	÷	÷
plus-or-minus	±	±
less-than	<	<
greater-than	>	>
dagger	†	†
double dagger	‡	‡
cent	¢	¢
pound sterling	£	£
euro	€	€
yen	¥	¥
general currency	¤	¤

Adding CSS Styles

စာလုံးတွေရဲ့ size တွေ အရောင်တွေ နောက်ပြီးတော့ space တွေ အတွက် HTML code အပြင် CSS ကိုပါထည့်သွင်းပြီး အသုံးပြုကြတယ်။ CSS ရဲ့ အရည်ကောက်ကတော့ Cascading Style Sheets လို့ ခေါ်ပါတယ်။ အခု Chapter မှာ CSS အကြောင်းကို လေ့လာလို့ ရမှာပါ။ CSS ဟာ HTML5 မှာ အရေးကြီးပါတယ်။ အရင် HTML version က ပြုလုပ်လို့ရတဲ့ အရာတွေကို အခု HTML5 မှာတော့ CSS နဲ့ မဖြစ်မနေ အသုံးပြုမှ ဖြစ်မှာပါ။

Style Sheets

HTML5 မှာ ကျွန်တော်တို့တွေ အနေနဲ့ heading တွေ paragraphs , list , image စတာတွေ အပြင် အခြား video , music စသည့် multimedia အပိုင်းတွေ လည်း ပါဝင်ပါတယ်။ ကျွန်တော်တို့ရဲ့ content ကို ဘယ်လို မြင်ချင်တယ် ဘယ်လိုမျိုးပုံစံနဲ့ ဖော်ပြချင်လဲဆိုတာကို Cascading Style Sheets သို့မဟုတ် CSS ကို အသုံးပြုရပါတယ်။ CSS က color , font families, backgrounds , margin ,padding စတာတွေ အခြား အများကြီးကို ဖန်တီး ထိန်းချုပ်ပေးပါတယ်။ CSS ကို HTML အတွင်းမှာ ဒါမှမဟုတ် သီးသန့် file အနေနဲ့ အသုံးပြုလို့ရပါတယ်။ HTML လိုမျိုး CSS ဟာ လည်း text file တစ်ခု ဖြစ်ပါတယ်။ external css ကို ကျွန်တော်တို့တွေ .css ဆိုပြီး သိမ်းပါတယ်။

Create an Internal Style Sheet

HTML မှာ internal style sheet နဲ့ external style sheet ဆိုပြီး ၂ မျိုး အသုံးပြုနိုင်ပါတယ်။ internal style sheet ကို `<style>` နဲ့ `</style>` ကြားမှာ ရေးသားရပါတယ်။ ကျွန်တော် တို့တွေ အောက်က CSS example လေးကို ကြည့်ရအောင်။

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>CSS</title>
    <style>
      h1 {
        color: red;
      }
      .myclass {
        color: orange;
      }
      #myid {
        color: blue;
      }
    </style>
  </head>
```

```

<body>

<h1>This is header</h1>
<p class="myclass">Color green</p>
<p id="myid">Color blue</p>

</body>
</html>

```

အဲဒီ code လေးကို ကျွန်တော်တို့တွေ အောက်ကလို မြင်ရပါလိမ့်မယ်။

This is header

Color green

Color blue

Figure 3-1

CSS မှာ tag name , class နဲ့ id ဆိုပြီး ၃ ခု ကို အသုံးပြုပါတယ်။ tag name ကတော့ HTML tag name တွေပေါ့။ p , h1,h2 စတာတွေပေါ့။ အပေါ်က ဥပမာမှာ ကျွန်တော်တို့ h1 ဆိုတဲ့ tag name ကို အသုံးပြုထားပါတယ်။ class ကတော့ dot (.) လေးနဲ့ တွဲပြီး အသုံးပြုပါတယ်။ class ကို ကြေငြာချင်ရင် class="myclass" ဆိုပြီး ရေးပါတယ်။ ကျွန်တော်က myclass ဆိုပြီး နာမည်ပေးထားခဲ့ပါတယ်။ ဒါကြောင့် .myclass ဆိုပြီး သုံးထားပါတယ်။ id ကတော့ # နဲ့ အသုံးပြုပါတယ်။ id ကို HTML tag မှာ id="myid" ဆိုပြီး အသုံးပြုတယ်။ အခု ဥပမာမှာ myid လို့ နာမည်ပေးထားတယ်။ ဒါကြောင့် #myid ဆိုပြီး အသုံးပြုထားပါတယ်။

class နဲ့ id ကွာခြားချက်ကတော့ id က တစ်ခု တည်း အနေနဲ့ အသုံးပြုပြီး class ကတော့ တစ်ခု မက အသုံးပြုပါတယ်။ id="myid" လို့ ပေးထားတဲ့ အတွက် ထပ်ပြီးတော့ id နာမည်ပေးတဲ့ အခါ myid ဆိုပြီး ထပ် မပေးတော့ပါဘူး။ class ကတော့ ထပ်ပြီး ပေးခွင့်ရှိပါတယ်။ class="myclass" လို့ ပေးထားတဲ့ အတွက် အခြား နေရာမှာလည်း ဒီ class ကို ပဲ အသုံးပြုချင်ရင် class="myclass" ဆိုပြီး ထပ်ပေးလို့ရပါတယ်။

css ကို စရေးရင် { နဲ့ စပြီးတော့ } နဲ့ ဆုံးပါတယ်။ အဲဒီ ကြားထဲမှာ code ရေးပါတယ်။ color:orange; ဆိုတာကတော့ စာလုံး အရောင် လိမ္မော်ရောင် သုံးမယ်လို့ ဆိုထားတာပါ။ code တစ်ကြောင်း ဆုံးရင် ; ထည့်ရပါတယ်။

ဒီတစ်ခေါက်တော့ font ပြောင်းကြည့်ရအောင်

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>CSS</title>
  <style>
    h1 {

```

```

        color:red;
        font-family: Arial;
    }
    .myclass {
        color:orange;
        font-style: italic;
    }
    #myid {
        color: blue;
        font-size: 12px;
    }
</style>
</head>
<body>

<h1>This is header</h1>
<p class="myclass">Color green</p>
<p id="myid">Color blue</p>

</body>
</html>

```

အဲဒီ code ကို browser မှာ ကြည့်ရင် အောက်ကလို မြင်ရပါမယ်။

This is header

Color green

Color blue

Figure 3-2

အဲဒီ code မှာ font-family: Arial; ဆိုတာကတော့ font ကို Arial သုံးမယ်လို့ ပြောထားပါတယ်။ font-style: italic; ကတော့ font style ကို italic (စာလုံးစောင်း) ပုံစံ သုံးမယ်လို့ ပြောထားပါတယ်။ font-size: 12px; ကတော့ font ရဲ့ size က 13px ကို အသုံးပြုမယ်လို့ ဆိုထားတာပါ။ နောက် chapter တစ်ခုမှာ ဒီထက်ပိုပြီးတော့ CSS ကို ရှင်းပြပေးသွားမှာပါ။

Create an External Style Sheet

External Style sheet ဆိုတာကတော့ CSS တွေကို file တစ်ခုတည်းမှာ ထည့်သွင်းထားပြီးတော့ HTML file ကနေ တဆင့်ပြန်ခေါ်သုံးတာပါ။ ဒါကြောင့် website တစ်ခုထဲမှာ CSS တွေ ထပ်ခါ ထပ်ခါရေးနေစရာမလိုတော့ပါဘူး။ အခု ကျွန်တော်တို့ CSS file တစ်ခု တည်ဆောက်ရအောင်။ အရင်ဆုံး index.html ကို folder တစ်ခုထဲမှာ ဆောက်လိုက်ပါ။ ပြီးရင် index.html file သိမ်းထားတဲ့ နေရာမှာပဲ style.css ဆိုပြီး file တစ်ခု ဖန်တီးပါ။ .css ကို ဖန်တီးခြင်းဟာ .html ဖန်တီးတာနဲ့ အတူတူပါပဲ။ file extension ကို .css ဟု ပြောင်းပေးရုံသာ ရှိပါတယ်။ တကယ်လို့ မေ့သွားခဲ့ရင်တော့ Chapter 1 မှာ ပြန်လည် ဖတ်ရှုနိုင်ပါတယ်။ ပြီးရင် .css ကို ဖွင့်လိုက်ပါ။ အဲဒီ အထဲမှာ အောက်ကလို ရေးလိုက်ပါ။

```

/* style.css */
h1 {
    color:red;
    font-family: Arial;
}
.myclass {
    color:orange;
    font-style: italic;
}
#myid {
    color: blue;
    font-size: 12px;
}

```

`/* style.css */` ဆိုတာကတော့ comment ပါ။ comment ရေးတဲ့ အခါမှာတော့ `/*` နဲ့ စပြီးတော့ `*/` နဲ့ ဆုံးပါတယ်။ comment ပေးထားတဲ့ အခါမှာ `/*` နဲ့ မဆုံးမခြင်း လိုင်းကြောင်း တစ်ကြောင်း ထက် ပို အသုံးပြုလို့ရပါတယ်။

ကျွန်တော်တို့ index.html ကို အောက်ကလို ရေးရအောင်။

```

<!DOCTYPE html>
<html>
<head>
    <title>CSS</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>

<h1>This is header</h1>
<p class="myclass">Color green</p>
<p id="myid">Color blue</p>

</body>
</html>

```

အဲဒီ code လေးကို save လုပ်ပြီး index.html ကို browser မှာ ကြည့်လိုက်ပါ။ Figure 3-2 က အတိုင်း မြင်ရပါလိမ့်မယ်။

အပေါ်က code မှာ `<link rel="stylesheet" type="text/css" href="style.css">` နဲ့ CSS ကို ချိတ်ဆက်ထားပါတယ်။ `<link rel="stylesheet" type="text/css"` ဆိုတာကတော့ CSS ကို အသုံးပြုမယ်လို့ ဆိုထားပြီး `href=` ကတော့ ဘယ် file ကို အသုံးပြုမယ်လို့ ပြောထားပါတယ်။

ကျွန်တော်တို့ နောက်ထပ် file တစ်ခု ဖန်တီးရအောင်။ index.html နေရာမှာပဲ page2.html ဆိုပြီး ဖန်တီးလိုက်ပါ။ အဲဒီ file ထဲမှာ အောက်ကလို ကျွန်တော်တို့တွေ ရေးထားလိုက်ပါမယ်။

```

<!DOCTYPE html>
<html>
<head>
    <title>Page 2</title>
    <link rel="stylesheet" href="style.css">

```

```

</head>
<body>

<h1>Page 2</h1>
<p class="myclass">My Class</p>
<p id="myid">My id</p>

</body>
</html>

```

page2.html ကို browser မှာ ကြည့်ရင် အောက်ကလို မြင်ရပါလိမ့်မယ်။

Page 2

My Class

My id

Figure 3-3

အရင် index.html မှာ ရေးထားသလို myclass , myid စတာတွေကို ပြန်ပြီး အသုံးပြုထားပါတယ်။ အဲဒီ class နဲ့ id နဲ့ ဆိုင်တဲ့ CSS တွေကို style.css မှာ ရေးထားခဲ့ပါတယ်။ အဲဒီ css ကိုပဲ page2.html က အသုံးပြုထားတယ်။ ဒါကြောင့် index.html အတိုင်း style က အတူတူ ဖြစ်နေတာကို တွေ့နိုင်ပါတယ်။

Apply a Style Locally

တစ်ခါတစ်ရံ ကျွန်တော်တို့တွေဟာ style ကို class , id , tag တွေနဲ့ သီးသန့် ရေးမယ့် အစား တစ်ကြောင်းထဲမှာ ရေးချင်တဲ့ အခါတွေ ရှိတတ်ပါတယ်။ အဲဒီ အခါ အောက်ကလို မျိုး အသုံးပြုနိုင်ပါတယ်။

```

<!DOCTYPE html>
<html>
<head>
  <title>Page 2</title>
</head>
<body>

<p style="color:red">Red</p>
<p style="color:green">Green</p>
<p style="color:blue">Blue</p>

</body>
</html>

```

အဲဒီ code လေးကို browser မှာ ဆိုရင် အောက်ကလို မြင်ရပါလိမ့်မယ်။

Red

Green

Blue

Figure 3-4

လိုင်းတစ်ကြောင်းတည်းမှာ CSS ကို အသုံးပြုချင်တဲ့ အခါ `style=` ဆိုတဲ့ attribute ထဲမှာ ထည့် ရေးရပါတယ်။

Apply a Style to a Tag

HTML tag တွေကိုလည်း CSS သုံးပြီး လွယ်လင့် တကူပြောင်းလဲလို့ရပါတယ်။ ဥပမာ။ Paragraph က စာလုံးတွေအားလုံးကို Arial font face ထားမယ် စာလုံး ကို 15px ထားမယ် ဆိုပါ တော့။ `<p>` က စာတွေ အားလုံးပြောင်းလဲမှာဖြစ်တဲ့အတွက်ကြောင့် အောက် က ပုံစံလို့ ရေးလို့ရ ပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS</title>
  <style>
    p {
      font-family: Arial;
    }
  </style>
</head>
<body>
<p>Modern HTML is the standard markup language for structuring and presenting content on
the Web. People still commonly say HTML5, but the current standard is maintained as the
HTML Living Standard.</p>
<p>HTML works together with CSS and JavaScript. HTML describes the meaning and structure
of content, CSS controls presentation, and JavaScript adds behavior. Keeping those
responsibilities separate makes pages easier to maintain and more accessible across
browsers and devices.</p>
</body>
</html>
```

p tag တွေ အားလုံးဟာ font က Arial ဖြစ်ပြီးတော့ font-size က 15px ဖြစ်သွားပါမယ်။ အဲဒီ code ကိုပဲ ကျွန်တော်တို့တွေ h1 ထည့်ထားတဲ့ Example လေးကြည့်လိုက်ပါ။

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS</title>
  <style>
    h1 {
      font-size:200%;
    }
  </style>
</head>
<body>
<h1>Hello World</h1>
</body>
</html>
```

```

        color:#DDDDDD;
    }
    p {
        font-family: Arial;
    }
</style>
</head>
<body>
<h1>HTML5</h1>
<p>Modern HTML is the standard markup language for structuring and presenting content on
the Web. People still commonly say HTML5, but the current standard is maintained as the
HTML Living Standard.</p>
<p>HTML works together with CSS and JavaScript. HTML describes the meaning and structure
of content, CSS controls presentation, and JavaScript adds behavior. Keeping those
responsibilities separate makes pages easier to maintain and more accessible across
browsers and devices.</p>
</body>
</html>

```

CSS ထဲမှာ h1 tag ကို မီးခိုးရောင် သုံးထားတာ တွေ့နိုင်ပါတယ်။ အဲဒီ code ကို browser မှာ ကြည့် လိုက် ရင် အောက်ကလို မြင်ရပါမယ်။

HTML5

HTML5 is a markup language for structuring and presenting content for the World Wide Web and a core technology of the Internet. It is the fifth revision of the HTML standard (created in 1990 and standardized as HTML 4 as of 1997) and, as of December 2012, is a W3C Candidate Recommendation. Its core aims have been to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understood by computers and devices (web browsers, parsers, etc.). HTML5 is intended to subsume not only HTML 4, but also XHTML 1 and DOM Level 2 HTML.

Following its immediate predecessors HTML 4.01 and XHTML 1.1, HTML5 is a response to the observation that the HTML and XHTML in common use on the World Wide Web are a mixture of features introduced by various specifications, along with those introduced by software products such as web browsers, those established by common practice, and the many syntax

Figure 3-5

Apply a Style Using a Class

ကျွန်တော်တို့တွေ အနေနဲ့ tag နဲ့ class ကို တွဲပြီးတွေ့လည်း အသုံးပြုနိုင်ပါတယ်။ ဥပမာ။

```

<!DOCTYPE html>
<html>
<head>
  <title>CSS</title>
  <style>
    p {
      color: blue;
    }

    .sample {
      color:green;
    }
    p.sample {
      color:red;
    }
  </style>
</head>
<body>
<p>Only tag</p>
<div class="sample">Div sample class</div>
<p class="sample">Sample class with p tag</p>
</body>
</html>

```

အထက်ပါ code ကို ကြည့်ရင် အောက်ကလို မြင်ရပါမယ်။

Only tag

Div sample class

Sample class with p tag

Figure 3-6

sample class ကို ၂ နေရာမှာ သုံးထားပါတယ်။ div tag နဲ့ p tag ပါ။ p.sample လို့ ရေးထားတဲ့ သဘောက p tag ဖြစ်ရမယ်။ class ကို sample ဖြစ်ရမယ်လို့ ဆိုလိုတာပါ။

Apply a Style Using an ID

တကယ်လို့ tag နဲ့ id ကို တွဲရေးမယ် ဆိုလည်း ရပါတယ်။ အောက်က ဥပမာ ကို ကြည့်ကြည့်ပါ။

```

<!DOCTYPE html>
<html>
<head>
  <title>CSS</title>
  <style>
    p {
      color: blue;
    }

    #sample {
      color:green;
    }
  </style>
</head>
<body>
<p>Only tag</p>
<div id="sample">Div sample id</div>
</body>
</html>

```

```

}
p.sample {
  color:red;
}

p#sample {
  color:gray;
}
</style>
</head>
<body>
<p>Only tag</p>
<div class="sample">Div sample class</div>
<p class="sample">Sample class with p tag</p>
<p id="sample">Sample ID with p tag</p>
</body>
</html>

```

အဲဒီ code ကို အောက်ကလို မြင်ရပါလိမ့်မယ်။

Only tag

Div sample class

Sample class with p tag

Sample ID with p tag

Figure 3-7

`p#sample` ဆိုပြီး p Tag ဖြစ်ပြီး `sample ID` ဖြစ်တာကို စာလုံး ခဲရောင်ပြောင်းထားတာကို တွေ့နိုင်ပါတယ်။

Link to Media-specific Style Sheets

ကျွန်တော်တို့တွေ CSS ကို အသုံးပြုတဲ့အခါမှာ browser ပေါ်မှာ ပေါ်တဲ့ CSS နဲ့ print ထုတ်တဲ့ အခါမှာပေါ်တဲ့ CSS သီးသန့် ခွဲထုတ်ထားလို့ရပါတယ်။ အောက်က ဥပမာ လေးကို အရင် ကြည့်ရအောင်။

```

<!DOCTYPE html>
<html>
<head>
  <title>Media</title>
  <link rel="stylesheet" type="text/css" href="blue.css" media="screen">
  <link rel="stylesheet" type="text/css" href="purple.css" media="print">
</head>
<body>
<h1>Media-specific Style Sheets</h1>

<p class="special">Sample Text. Blue for screen and purple for print.
</body>

```

```
</html>
```

အဲဒီမှာ `media="screen"` နဲ့ `media="print"` ဆိုပြီး ရေးထားတာကို တွေ့နိုင်ပါတယ်။ screen ဆိုတာကတော့ browser ကနေ ကြည့်တဲ့ အခါမှာ ပေါ်ဖို့ သီးသန့် အတွက်ပါ။ print ကတော့ print ထုတ်တဲ့ အခါမှာ အသုံးပြုဖို့ css ပါ။ default value ကတော့ all ပါ။

blue.css မှာ ဒီလိုရေးထားပါတယ်။

```
.special {
    color: blue;
}
```

purple.css မှာတော့ ဒီလိုရေးထားပါတယ်။

```
.special {
    color: purple;
}
```

အောက်က ပုံလေးကို ကြည့်ကြည့်ပါ။ Print ထုတ်တဲ့ အခါနဲ့ browser ပေါ်မှာ မြင်ရတဲ့ ပုံ မတူတာကို တွေ့နိုင်ပါတယ်။

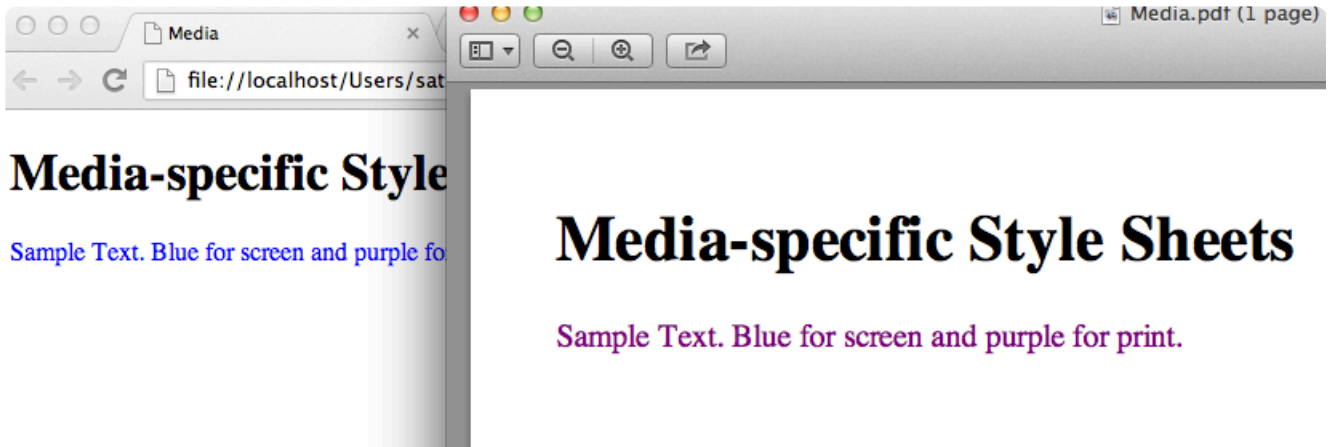


Figure 3-8

Link to Style Sheets for Mobile Phones

အခုနောက်ပိုင်းမှာ mobile phone တွေ လူသုံးများလာတဲ့ အညီ css ကိုလည်း mobile တွေပေါ်မှာ ကောင်းစွာ ပေါ်အောင် ပြုပြင်ရေးဆွဲရပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
    <title>Media</title>
```

```

    <link rel="stylesheet" href="ipad-style.css" media="only screen and (max-width:
1024px)">
    <link rel="stylesheet" href="iphone-style.css" media="only screen and (max-width:
480px)">
</head>
<body>
<h1>Media-specific Style Sheets</h1>

<p class="special">Sample Text. Blue for screen and purple for print.
</body>
</html>

```

အထက်ပါ code မှာ `media="only screen and (max-width: 1024px)"` နဲ့ `media="only screen and (max-width: 480px)"` ဆိုတာကို တွေ့နိုင်ပါတယ်။

`max-device-width` ဆိုတာကတော့ `device width` ရဲ့ အမြင့်ဆုံး အရွယ်အစားဆိုရင် ဒီ style ကို အသုံးပြုဆိုပြီး ရေးသားထားတာပါ။ `ipad` ရဲ့ အကျယ်ဆုံး `device width` က 1024 ပါ။ `iPhone` ရဲ့ ဒေါင်လိုက် အရွယ်မှာ အကျယ်ဆုံးကတော့ 480px ပါ။ `max-width` မှာ ထည့်ထားတဲ့ `width` ရောက်မှသာ အဲဒီ `css` က အလုပ်လုပ်မှာ ဖြစ်ပါတယ်။ တခြားအချိန်တွေမှာတော့ အလုပ်မလုပ် ပေးပါဘူး။

တကယ်လို့ ကျွန်တော်တို့တွေ အနေနဲ့ `css file` မှာ `media queries` ကို အသုံးပြုချင်ရင် အောက်က လို အသုံးပြုနိုင်ပါတယ်။

```

/* default CSS */
.special {
    color: blue;
}

@media screen {
    .special {
        color:blue
    }
}

@media print {
    .special {
        color:purple;
    }
}

/* for iphone */
@media screen and (max-width: 480px) {
    .special {
        color :red;
    }
}

/* for iphone */
@media screen and (max-width: 1024px) {
    .special {
        color :green;
    }
}

```

အထက်ပါ code လေးက လွယ်လွယ်ကူကူနဲ့ ရှင်းရှင်းလင်းလင်း နားလည်နိုင်ပါတယ်။ အထက်ပါ code မှာ screen , print နဲ့ iphone , iPad တို့အတွက် css တွေ အကုန် ပေါင်းရေးထားပေးပါတယ်။ media queries ရေးတဲ့ အခါမှာ { နဲ့ စပြီးတော့ } နဲ့ ဆုံးပါတယ်။ အဲဒီ ကြားထဲမှာ သက်ဆိုင်တဲ့ CSS ကို ရေးပေးဖို့လိုပါတယ်။

Styles for Nested Tags

CSS ကို အသုံးပြုရမှာ ကျွန်တော်တို့တွေ အနေနဲ့ class တွေ tag တွေကို အဆင့်ဆင့် ပြီး အသုံးပြု လို့ရပါတယ်။ အောက်က code လေးကို တချက်လေ့လာကြည့်ရအောင်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Nested CSS</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
<section class='first'>
  <p>This is first paragraph.</p>
  <div id="example">
    <p>This is second paragraph.</p>
  </div>
</section>

<section class='second'>
  <p>This is first paragraph.</p>
  <div id="example">
    <p>This is second paragraph.</p>
  </div>
</section>
</body>
</html>
```

အထက်ပါ code မှာ ကျွန်တော် section ၂ ခုကို အသုံးပြုထားပါတယ်။ ပထမ section မှာ paragraph ၂ ခု ပါသလို ဒုတိယ section မှာလည်း paragraph ၂ ခု ပါပါတယ်။ သူ့မှာ သုံးထားတဲ့ style.css ကို ကြည့်ကြည့်ရအောင်။

```
p {
  color:red;
}
section.first p{
  color:blue;
}

section.second > p {
  color: purple;
}
```

တကယ်လို့ ကျွန်တော်တို့တွေ paragraph အားလုံးရဲ့ စာကို ပြောင်းချင်ရင် p လို့ ခေါ်လိုက်ရင်ရပါ ပြီ။ သို့ပေမယ့် ပထမ section ရဲ့ paragraph ကို သာ အရောင်ပြောင်းချင်တဲ့အတွက် section.first p ဆိုပြီး သုံးထားပါတယ်။ section tag ဖြစ်ရမယ် အဲဒီ tag ရဲ့ class က first ဖြစ်ရမယ်။ အဲဒီ first ရဲ့ အောက်က p တွေကို ဒီ css သုံးမယ်ဆိုပြီး ရေးထားတာပါ။

section.second > p ကတော့ နည်းနည်းလေး ကွဲပြားပါယတ်။ section tag ဖြစ်ရမယ်။ class နာမည်က second ဖြစ်ရမယ်။ သူ့ရဲ့ အောက်က p ကို သာ ပြောင်းမယ်။ ကျွန်တော်တို့ .second ရဲ့ အောက်မှာ div တစ်ခု ရှိတာကို တွေ့နိုင်ပါတယ်။ အဲဒီ အောက်က p က .second အောက်မှာ ရှိတဲ့ p class မှာ မဟုတ်တော့တဲ့အတွက် သူက default color အနီ ရောင်သာ ပြနေပါလိမ့်မယ်။

အဲဒီ code လေးကို run လိုက်ရင် အောက်ကလိုမျိုးလေး မြင်ရပါလိမ့်မယ်။

This is first paragraph.
This is second paragraph.
This is first paragraph.
This is second paragraph.

Figure 3-9

အခု ဆိုရင် chapter 3 ပြီးဆုံးပါပြီ။ Chapter 3 မှာ CSS နဲ့ ပတ်သက်ပြီး အခြေခံ အဆင့်ထိ သိ လောက်ပြီလို့ ထင်ပါတယ်။ Chapter 3 မှာ tag , class , ID စတာတွေကို CSS နဲ့ တွဲသုံးပြီး သုံး နည်းကို ဖော်ပြထားပါတယ်။ နောက်ပြီး chapter မှာတော့ text styling နဲ့ ပတ်သက်ပြီးတော့ ဆက်လက် လေ့လာနိုင်ပါတယ်။

Styling Text

ပြီးခဲ့တဲ့ chapter 3 မှာ CSS နဲ့ ပတ်သက်ပြီးတော့ သိသင့်သလောက် သိပြီးလောက်ပါပြီ။ ဒီ chapter မှာတော့ Text တွေကို အရောင်ပြောင်းခြင်း နောက်ပြီးတော့ စာလုံး အရွယ်အစား ပြောင်းတာတွေကို ပြောပြပေးသွားပါမယ်။

Changing Font Size

ကျွန်တော်တို့တွေ စာလုံးပုံစံကို အောက်က ပုံမျိုးလေး ဖြစ်အောင် လုပ်ကြည့်ရအောင်

Figure 4-1

အထက်ပါ ဥပမာမှာ စာလုံးတွေက ပုံမှန် ထက်ကြီးထားတာကို တွေ့နိုင်ပါလိမ့်မယ်။ အဲဒီအတွက် ကျွန်တော်တို့တွေ CSS မှာ font-size ဆိုတာကို အသုံးပြုရပါတယ်။

HTML code ကို ဒီလိုရေးထားပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Styling Text</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>One Piece</h1>
  <p>One Piece is a Japanese shounen manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 70th volume released as of March 2013. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.</p>
</body>
</html>
```

အဲဒီမှာ external style sheet ဖြစ်တဲ့ style.css ကို ခေါ်ထားတာကို တွေ့နိုင်ပါတယ်။ အဲဒီ style.css ထဲမှာတော့

```
h1 {
  font-size: 40px;
}
```

h1 ရဲ့ font size ကို ကျွန်တော် 40px ပေးထားပါတယ်။ ဒါကြောင့် ပုံမှန် h1 ထက် ကြီးနေရတာပါ။
font size ကို **points (pt)**, **pixels (px)**, **millimeters (mm)**, **centimeters (cm)**, **inches (in)**, **picas (pc)**, **x-height (ex)**, သို့မဟုတ် em space (em) စတာတွေကို အသုံးပြုနိုင်ပါတယ်
ဒါ့အပြင် (**xx-small**, **x-small**, **small**, **medium**, **large**, **x-large**, or **xx-large**) တွေကိုလည်း အသုံးပြုနိုင်ပါတယ်။

Indent Text

အခု ထပ်ပြီးတော့ ကျွန်တော်တို့တွေ text indent ကို ပြင်ရအောင်။ paragraph မှာ စာ အစ index ကို ပြင်ပါမယ်။

style.css ကို ဒီလို ထပ်ပြင်လိုက်ပါမယ်။

```
h1 {
    font-size: 40px;
}

p {
    text-indent: 20px;
}
```

အဲဒီ အခါ text က အောက်ကလို ဖြစ်သွားပါမယ်။

One Piece

One Piece is a Japanese shounen manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 70th volume released as of March 2013. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

Figure 4-2

Change the Color of Text

Text Color ပြောင်းဖို့အတွက် color ဆိုတာကို အသုံးပြုရပါတယ်။

```
h1 {
    font-size: 40px;
}
```

ဆိုတာ ကို ဒီလို ပြောင်းလိုက်ပါမယ်။

```
h1 {
    font-size: 40px;
    color: green;
}
```

အဲဒီ code မှာ color ဆိုတာ ကို တွေ့ပါလိမ့်မယ်။ အဲဒီ code ကို run လိုက်ရင် အောက်ကလို မြင်ရမယ်။

One Piece

One Piece is a Japanese shounen manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 70th volume released as of March 2013. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

Figure 4-3

အဲဒီမှာ ခေါင်းစဉ် အရောင်က အစိမ်းရောင် ဖြစ်သွားတာကို တွေ့လိမ့်မယ်။ ကျွန်တော်တို့ h1 ရဲ့ color ဟာ green ပါဆိုပြီး ရေးခဲ့တာကြောင့်ပါ။ color နာမည်တွေ အပြင် ကျွန်တော်တို့ အသုံးများတဲ့ Hexadecimal Code ကို ပါအသုံးပြုလို့ရပါတယ်။

```
h2 {
    font-size: 25px;
    color: #336699;
}
```

အဲဒီ code လေးကို လက်ရှိ CSS မှာ ထပ်ဖြည့်လိုက်မယ်။ HTML မှာလည်း h2 ကို ကျွန်တော်တို့ တွေ ထပ်ဖြည့်လိုက်ပါမယ်။ အဲဒီ အခါ HTML code က အောက်ကလို ဖြစ်သွားလိမ့်မယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Styling Text</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>One Piece</h1>
  <h2>by Eiichiro Oda</h2>
  <p>One Piece is a Japanese shounen manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 70th volume released as of March 2013. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.</p>
</body>
</html>
```

အဲဒီ code လေးက browser မှာ အောက်ကလို ပေါ်ပါလိမ့်မယ်။

One Piece

by Eiichiro Oda

One Piece is a Japanese shounen manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 70th volume released as of March 2013. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

Figure 4-4

Color ကို Hex value (#RRGGBB) နဲ့ color name ၂ မျိုး လုံး ထည့်လို့ရပါတယ်။ အောက်မှာ ဥပမာလေးတွေ နဲ့ ပြထားပါတယ်။

Color Name	HEX	Color
AliceBlue	#F0F8FF	

AntiqueWhite	#FAEBD7	
Aqua	#00FFFF	
Aquamarine	#7FFFD4	
Azure	#F0FFFF	
Beige	#F5F5DC	
Bisque	#FFE4C4	
Black	#000000	
BlanchedAlmond	#FFEBCD	
Blue	#0000FF	
BlueViolet	#8A2BE2	
Brown	#A52A2A	
BurlyWood	#DEB887	
CadetBlue	#5F9EA0	
Chartreuse	#7FFF00	
Chocolate	#D2691E	
Coral	#FF7F50	
CornflowerBlue	#6495ED	
Cornsilk	#FFF8DC	
Crimson	#DC143C	
Cyan	#00FFFF	
DarkBlue	#00008B	
DarkCyan	#008B8B	
DarkGoldenRod	#B8860B	
DarkGray	#A9A9A9	
DarkGreen	#006400	
DarkKhaki	#BDB76B	
DarkMagenta	#8B008B	
DarkOliveGreen	#556B2F	
DarkOrange	#FF8C00	

DarkOrchid	#9932CC	
DarkRed	#8B0000	
DarkSalmon	#E9967A	
DarkSeaGreen	#8FBC8F	
DarkSlateBlue	#483D8B	
DarkSlateGray	#2F4F4F	
DarkTurquoise	#00CED1	
DarkViolet	#9400D3	
DeepPink	#FF1493	
DeepSkyBlue	#00BFFF	
DimGray	#696969	
DodgerBlue	#1E90FF	
FireBrick	#B22222	
FloralWhite	#FFFAF0	
ForestGreen	#228B22	
Fuchsia	#FF00FF	
Gainsboro	#DCDCDC	
GhostWhite	#F8F8FF	
Gold	#FFD700	
GoldenRod	#DAA520	
Gray	#808080	
Green	#008000	
GreenYellow	#ADFF2F	
HoneyDew	#F0FFF0	
HotPink	#FF69B4	
IndianRed	#CD5C5C	
Indigo	#4B0082	
Ivory	#FFFFFF	
Khaki	#F0E68C	

Lavender	#E6E6FA	
LavenderBlush	#FFF0F5	
LawnGreen	#7CFC00	
LemonChiffon	#FFFACD	
LightBlue	#ADD8E6	
LightCoral	#F08080	
LightCyan	#E0FFFF	
LightGoldenRodYellow	#FAFAD2	
LightGray	#D3D3D3	
LightGreen	#90EE90	
LightPink	#FFB6C1	
LightSalmon	#FFA07A	
LightSeaGreen	#20B2AA	
LightSkyBlue	#87CEFA	
LightSlateGray	#778899	
LightSteelBlue	#B0C4DE	
LightYellow	#FFFFE0	
Lime	#00FF00	
LimeGreen	#32CD32	
Linen	#FAF0E6	
Magenta	#FF00FF	
Maroon	#800000	
MediumAquaMarine	#66CDAA	
MediumBlue	#0000CD	
MediumOrchid	#BA55D3	
MediumPurple	#9370DB	
MediumSeaGreen	#3CB371	
MediumSlateBlue	#7B68EE	
MediumSpringGreen	#00FA9A	

MediumTurquoise	#48D1CC	
MediumVioletRed	#C71585	
MidnightBlue	#191970	
MintCream	#F5FFFA	
MistyRose	#FFE4E1	
Moccasin	#FFE4B5	
NavajoWhite	#FFDEAD	
Navy	#000080	
OldLace	#FDF5E6	
Olive	#808000	
OliveDrab	#6B8E23	
Orange	#FFA500	
OrangeRed	#FF4500	
Orchid	#DA70D6	
PaleGoldenRod	#EEE8AA	
PaleGreen	#98FB98	
PaleTurquoise	#AFEEEE	
PaleVioletRed	#DB7093	
PapayaWhip	#FFEFD5	
PeachPuff	#FFDAB9	
Peru	#CD853F	
Pink	#FFC0CB	
Plum	#DDA0DD	
PowderBlue	#B0E0E6	
Purple	#800080	
Red	#FF0000	
RosyBrown	#BC8F8F	
RoyalBlue	#4169E1	
SaddleBrown	#8B4513	

Salmon	#FA8072	
SandyBrown	#F4A460	
SeaGreen	#2E8B57	
SeaShell	#FFF5EE	
Sienna	#A0522D	
Silver	#C0C0C0	
SkyBlue	#87CEEB	
SlateBlue	#6A5ACD	
SlateGray	#708090	
Snow	#FFFAFA	
SpringGreen	#00FF7F	
SteelBlue	#4682B4	
Tan	#D2B48C	
Teal	#008080	
Thistle	#D8BFD8	
Tomato	#FF6347	
Turquoise	#40E0D0	
Violet	#EE82EE	
Wheat	#F5DEB3	
White	#FFFFFF	
WhiteSmoke	#F5F5F5	
Yellow	#FFFF00	
YellowGreen	#9ACD32	

Change the Font

Font ကို ပြောင်းဖို့ အတွက် font-family: ဆိုတဲ့ property ကို အသုံးပြုရပါမယ်။ အောက်က code လေးကို တချက်ကြည့်လိုက်ပါ။

```
body {
  font-family: "Zawgyi-One", arial;
```

```

}
h1 {
  font-size: 40px;
  color: green;
}

h2 {
  font-size: 25px;
  color: #336699;
}

p {
  text-indent: 20px;
}

```

body ရဲ့ font ကို Zawgyi-One ပြောင်းထားတာကို တွေ့ရပါလိမ့်မယ်။ "Zawgyi-One",arial လို့ ရေးထားတဲ့ သဘောကျ Zawgyi-One font စက်ထဲမှာ မရှိရင် arial font ကို အသုံးပြုမယ်လို့ ဆို ထားပါတယ်။ comma (,) ကို ခံပြီးတော့ တစ်ခုထက် မက font-family တွေ ထည့်ထားလို့ရပါ တယ်။

အဲဒီ code လေးကိုပဲ ကျွန်တော်တို့ ထပ်ပြီး ပြောင်းရအောင်

```

body {
  font-family: "Zawgyi-One",arial;
}
h1 {
  font-size: 40px;
  color: green;
  font-family: "Times New Roman",serif;
}

h2 {
  font-size: 25px;
  color: #336699;
  font-family: "Verdana",arial;
}

p {
  text-indent: 20px;
}

```

အထက်ပါ code မှာ h1 ကို Times New Roman သုံးပြီးတော့ h2 ကိုတော့ Verdana သုံးထားတာ တွေ့နိုင်ပါတယ်။ body မှာ Zawgyi-One လို့ ပြောခဲ့တဲ့ အတွက်ကြောင့် တခြား code တွေ အားလုံး ဟာ Zawgyi-One font နဲ့ ဖော်ပြနေပါလိမ့်မယ်။ အဲဒီ code လေးကို run လိုက်ရင် အောက်က ပုံလို တွေ့ပါလိမ့်မယ်။

One Piece

by Eiichiro Oda

One Piece is a Japanese shounen manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 70th volume released as of March 2013. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

Figure 4-5

Change Text Alignment

Text alignment ကို ပြောင်းဖို့အတွက် `text-align` ဆိုတာကို အသုံးပြုရပါတယ်။ `text-align` မှာ `left` , `right` , `center` နဲ့ `justify` ဆိုတဲ့ value တွေကို ထည့်သွင်းနိုင်ပါတယ်။

ကျွန်တော်တို့ လက်ရှိ code က `h1` နဲ့ `h2` ကို `center` ပို့ဖို့ အတွက် code ကို အောက်ကလို ပြောင်းလိုက်ပါတယ်။

```
h1 {
  font-size: 40px;
  color: green;
  font-family: "Times New Roman", serif;
  text-align: center;
}

h2 {
  font-size: 25px;
  color: #336699;
  font-family: "Verdana", arial;
  text-align: center;
}
```

အထက်ပါ code မှာ `text-align: center;` ဆိုတာ လက်ရှိ text alignment ကို `center` ပြောင်းလိုက်တာပါ။ အဲဒီ code ကို အောက်ကလို မြင်ရပါလိမ့်မယ်။

One Piece

by Eiichiro Oda

One Piece is a Japanese shounen manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 70th volume released as of March 2013. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

Figure 4-6

တကယ်လို့ right alignment ကို အသုံးပြုချင်ရင်တော့ `text-align: right;` ဆိုပြီး ခေါ်အသုံးပြုနိုင်ပါတယ်။

Change the Text Case

Text Case ပြောင်းတယ်ဆိုတာကတော့ လက်ရှိ text တွေကို css ကနေ အလိုအလျောက် capitalize, uppercase, lowercase စသည်ဖြင့် ပြောင်းလဲတာပါ။ text-transform property ကို အသုံးပြုနိုင်ပြီး value တွေကတော့ capitalize, uppercase, lowercase , none တို့ကို အသုံးပြုနိုင်ပါတယ်။ လက်ရှိ code က h2 ကို အောက်ကလို ပြောင်းရေးပါမယ်။

```
h2 {
  font-size: 25px;
  color: #336699;
  font-family: "Verdana",arial;
  text-align: center;
  text-transform: uppercase;
}
```

အထက်ပါ code မှာ text-transform: uppercase; ဆိုတာကတော့ လက်ရှိ text ကို စာလုံး အကြီးနဲ့ ဖော်ပြမယ်လို့ ဆိုထားတာပါ။ အဲဒီ code လေးက အောက်ကလို မြင်ရပါလိမ့်မယ်။

One Piece

by Eiichiro Oda

One Piece is a Japanese shounen manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 70th volume released as of March 2013. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

Figure 4-7

Control Line Spacing

စာတစ်ကြောင်းနဲ့ တစ်ကြောင်း အကွာအဝေးကို ညှိချင်တဲ့ အခါမှာတော့ line-height ကို အသုံးပြုရပါတယ်။ လက်ရှိ HTML က p tag နဲ့ ရေးထားတဲ့ စာတွေကို line spacing ပြောင်းကြည့်ရအောင်။

```
p {
  text-indent: 20px;
  line-height: 40px;
}
```

အထက်ပါ code မှာ line-height ကို 40px ပြောင်းလိုက်ပါတယ်။ ဒါကြောင့် စာကြောင်းတွေက ကျသွားတာကို တွေ့ရပါလိမ့်မယ်။

One Piece

BY EIICHIRO ODA

One Piece is a Japanese shounen manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 70th volume released as of March 2013. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

Figure 4-8

Control Letter Spacing

Letter Spacing ကတော့ letter တစ်ခု နဲ့ တစ်ခုကြား အကွာအဝေးကို ပြောင်းတာပါ။ h1 ရဲ့ letter spacing ကို ပြောင်းကြည့်ရအောင်။

```
h1 {
  font-size: 40px;
  color: green;
  font-family: "Times New Roman", serif;
  text-align: center;
  letter-spacing: 10px;
}
```

letter-spacing ကို 10px ပေးထားတာ တွေ့ရပါလိမ့်မယ်။ အဲဒီ code လေးကို run ကြည့်လိုက်ရင် အောက်ကလို မြင်ရပါလိမ့်မယ်။

One Piece

BY EIICHIRO ODA

One Piece is a Japanese shounen
manga series written and illustrated by
Eiichiro Oda. It has been serialized in
Weekly Shōnen Jump since August 4,
1997; the individual chapters are being

Figure 4-9

Add Background Colors

ကျွန်တော်တို့ အခု page ကို background color ထည့်ပြီးတော့ စာ အချို့ကို highlight လုပ်တဲ့ ပုံစံ လေး ဖန်တီးကြည့်ရအောင်။ လက်ရှိ HTML code ကို အောက်ကလို ပြောင်းလိုက်ပါတယ်။

```
<html>
<head>
  <title>Styling Text</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <h1>One Piece</h1>
  <h2>by Eiichiro Oda</h2>
  <p>One Piece is a Japanese <span class='highlight'>shounen manga series</span>
written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump
since August 4, 1997; the individual chapters are being published in tankōbon volumes by
Shueisha, with the first released on December 24, 1997, and the 70th volume released as
of March 2013. One Piece follows the adventures of <span class='highlight'>Monkey D.
Luffy</span>, a young boy whose body gains the properties of rubber after unintentionally
eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy
explores the ocean in search of the world's ultimate treasure known as One Piece in order
to become the next Pirate King.</p>
</body>
</html>
```

အထက်ပါ code မှာ span နဲ့ highlight class ကို ထည့်ထားတာကို တွေ့နိုင်ပါတယ်။ အခု CSS ထပ် ပြောင်းကြည့်ရအောင်။

```
body {
  font-family: "Zawgyi-One",arial;
  background-color: #ECF0F1;
}
.highlight {
  background-color: #F1C40F;
}
```

လက်ရှိ CSS မှာ body ကို background color ထည့်ပြီးတော့ highlight class တစ်ခု ဖန်တီးပါတယ်။ ပြီးတော့ background color ထည့်ထားတာကို တွေ့နိုင်ပါတယ်။ အဲဒါကို လက်တွေ့ စမ်းကြည့်ရင် အောက်ကလို မြင်ရပါမယ်။

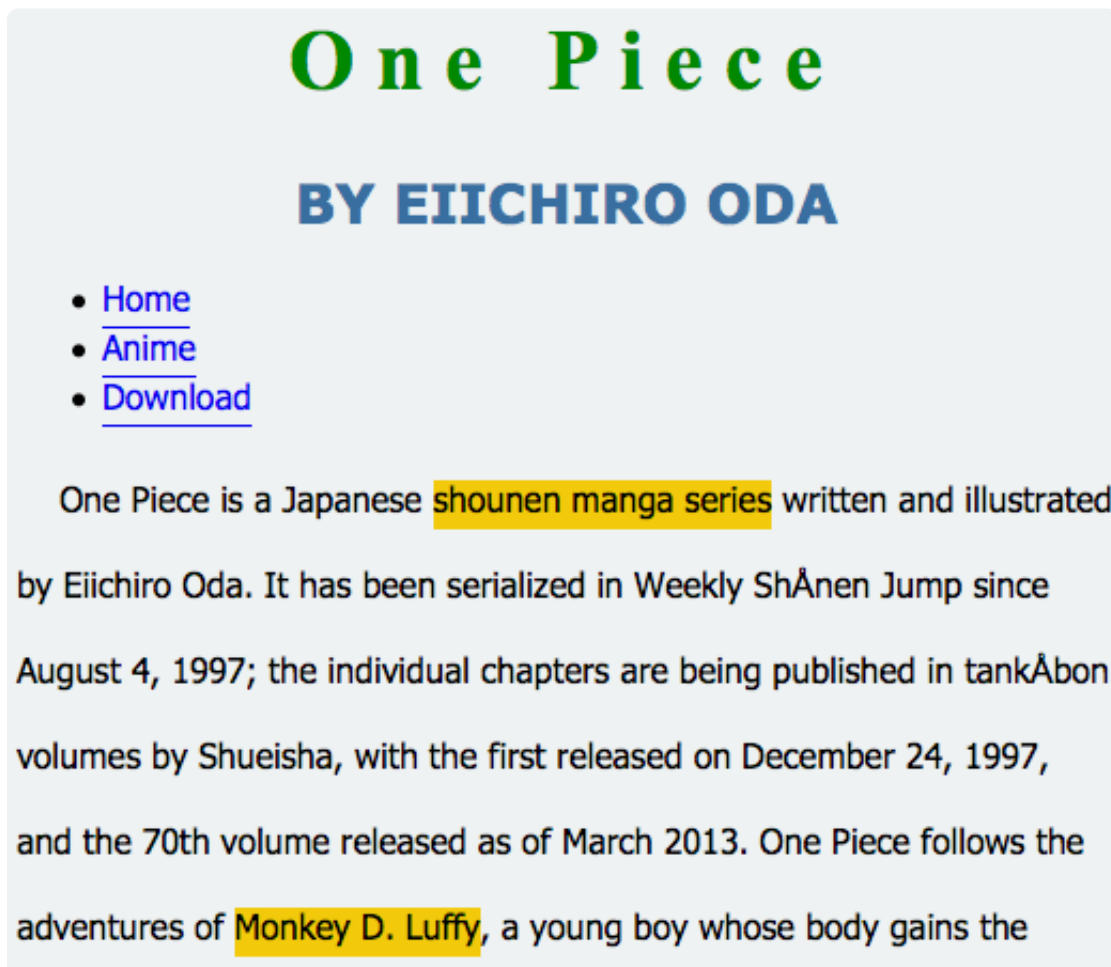


Figure 4-10

Style a Bulleted List

List ကို CSS အသုံးပြုပြီးတော့ ကျွန်တော်တို့တွေ menu လေး တစ်ခု ဖန်တီးကြည့်ရအောင်။ လက်ရှိ html code မှာ list ထပ်ဖြည့်ရအောင်။

```

<html>
<head>
  <title>Styling Text</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <h1>One Piece</h1>
  <h2>by Eiichiro Oda</h2>
  <ul class="menu">
    <li><a href="/home.html">Home</a></li>
    <li><a href="/anime.html">Anime</a></li>
    <li><a href="/download.html">Download</a></li>
  </ul>
  <p>One Piece is a Japanese <span class='highlight'>shounen manga series</span>
written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump
since August 4, 1997; the individual chapters are being published in tankōbon volumes by
Shueisha, with the first released on December 24, 1997, and the 70th volume released as
of March 2013. One Piece follows the adventures of <span class='highlight'>Monkey D.
Luffy</span>, a young boy whose body gains the properties of rubber after unintentionally
eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy
explores the ocean in search of the world's ultimate treasure known as One Piece in order
to become the next Pirate King.</p>
</body>
</html>

```

အဲဒီ code မှာ ကျွန်တော်တို့ ul ကို အသုံးပြုပြီးတော့ list တစ်ခု ဖန်တီးထားပါတယ်။ class name ကိုတော့ menu ပေးထားတယ်။ အဲဒီ code လေးကို browser မှာ ကြည့်လိုက်ရင် အောက်ကလို မြင်ရပါမယ်။

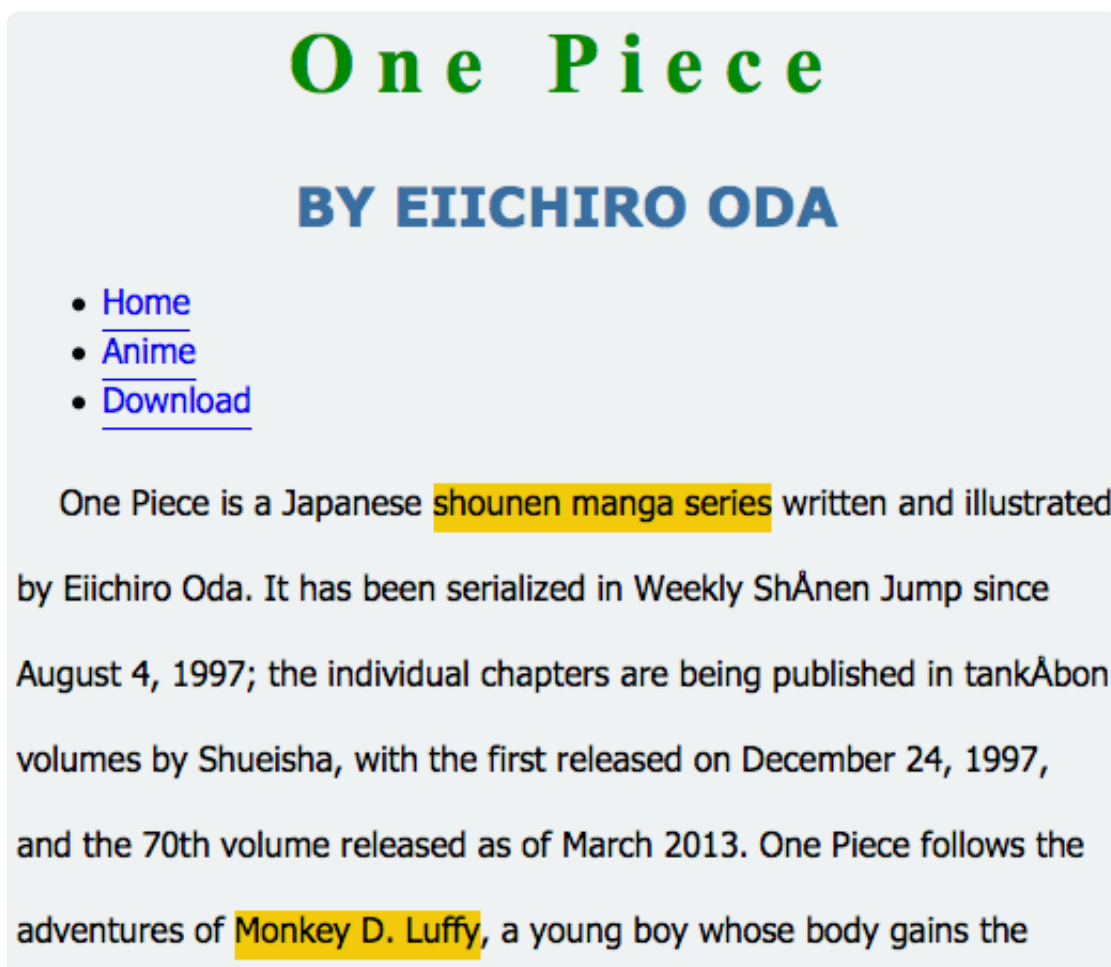


Figure 4-10

အခု menu css ကို အောက်လို ထည့်လိုက်ရင်အောင်

```
.menu {
  list-style: none;
  padding: 0px;
  text-align: center;
}

.menu li {
  display: inline;
  padding: 5px;
}

.menu li a{
  text-decoration: none;
  color:#E74C3C;
  border-bottom: 1px dotted #E74C3C;
}
```

ul တစ်ခုလုံးကို list-style ကို none ထားပြီးတော့ ကိုလည်း အသုံးမပြုပါဘူး။ text-align ကို center ထားလိုက်ပါတယ်။ .menu အောက်မှာ li ရှိပါတယ်။ သူ့အောက်မှာ ရှိတဲ့ li တွေကို ခေါ်ချင်တဲ့ အတွက်ကြောင့် .menu li ဆိုပြီး အသုံးပြုထားပါတယ်။ list တွေအားလုံးကို တစ်လိုင်းထဲမှာ ပြစေချင်တဲ့ အတွက်ကြောင့် display: inline; ကို အသုံးပြုထားပါတယ်။ li တွေ တစ်ခုနဲ့ တစ်ခု မကပ်ထားစေချင်တဲ့ အတွက်ကြောင့် padding: 5px; ဆိုပြီး padding ကို သုံးပြီး ခွာထားပါတယ်။ ပြီးတဲ့ အခါမှာ anchor tag (a) ကို ပြင်ဖို့အတွက် .menu li a ကို အသုံးပြုပါတယ်။ anchor မှာ လိုင်းတားထားတာကို ဖျက်လိုက်တယ်။ နောက်ပြီးတော့ အရောင်ပြောင်းတယ်။ border-bottom ထပ်ဖြည့်ထားပါတယ်။ anchor tag အကြောင်း အသေးစိတ်ကို နောက်ပိုင်းမှာ ရှင်းပြပါမယ်။ အထက်ပါ code ကို browser မှာ ကြည့်လိုက်ရင် အောက်ကလို မြင်ရပါမယ်။



Figure 4-11

အခု chapter 4 မှာ CSS နဲ့ ပတ်သက်ပြီး အတော်အသင့် သိလောက်ပါပြီ။ Text တွေကို အရောင်တွေ စာလုံးတွေ ပြောင်းတတ်လောက်ပါပြီ။ နောက် chapter တစ်ခုမှာ image ဘယ်လို ထည့်ရမလဲ ဆိုတာကို ဆက်လက်ပြီး ရှင်းပြသွားပါမယ်။

Adding Images

သင့်ရဲ့ website မှာ image , ဓာတ်ပုံတွေ တွေ ထည့်ဖို့ရန်အတွက် `` tag ကို အသုံးပြုပြီး ထည့်သွင်းနိုင်ပါတယ်။ image ကို padding , alignment တွေအတွက် CSS style sheets နဲ့ တွဲပြီး အသုံးပြုနိုင်ပါတယ်။

Web Page Images

Website မှာ image ကို နည်းလမ်း အမျိုးမျိုး နဲ့ အသုံးပြုနိုင်ပါတယ်။ အခုအချိန်မှာ အသုံးပြုကြတဲ့ format ကတော့ JPG, PNG, GIF အပြင် WebP, AVIF, SVG စတာတွေပါ ပါလာပါတယ်။ Image တွေကို အသုံးပြုရမှာ file size ကို သတိထားရပါတယ်။ အရွယ်အစားကြီးမားတဲ့ image တွေဟာ web page ကို ဖော်ပြရန် ပုံတွေကို download ချရမှာ နှေးကွေးပါတယ်။

JPG

JPEG, ရဲ့ အရှည်ကတော့ Joint Photographic Experts Group ဖြစ်ပြီး 24 bit color (millions colors) ကို support လုပ်ပါတယ်။ JPEG ကို ကာလာတွေ အများကြီးပါတဲ့ ဓာတ်ပုံတွေ ရုပ်ပုံတွေ မှာ အသုံးပြုကြပါတယ်။ JPEG ဟာ background transparent ပုံတွေ ကို အဖြူရောင် နောက်ခံ အဖြစ်မြင်တွေ့ရပါလိမ့်မယ်။ JPEG ရဲ့ file extension ကတော့ .jpg ဖြစ်ပါတယ်။

GIF

GIF ရဲ့ နာမည်အပြည့်အစုံကတော့ Graphics Interchange Format ပါ။ အရောင်ပေါင်း ၂၅၆ ခုသာ support လုပ်ပါတယ်။ GIF ကို အများအားဖြင့် ရိုးရှင်းလှတဲ့ image တွေမှာ အသုံးပြုကြပါတယ်။ ဒါ့အပြင် logo နဲ့ လိုင်းတွေ တစ်ရောင်တည်းရှိတဲ့ ပုံတွေမှာ အသုံးပြုကြပါတယ်။ တကယ်လို့ image ဟာ အရောင်နည်းနည်းပဲ ပါဝင်ပြီး အသေးစိတ် မလိုအပ်ဘူးဆိုရင် GIF format ကို ရွေးချယ်သင့်ပါတယ်။ GIF ဟာ background transparent ပုံတွေကို သိမ်းနိုင်ပါတယ်။ GIF ဟာ images တွေ အများကြီး သိမ်းထားနိုင်ပြီးတော့ လှုပ်ရှားမှုပုံတွေကို ပြသနိုင်ပါတယ်။ GIF image ကို .gif file extension နဲ့ သိမ်းပါတယ်။

PNG

PNG (Portable Network Graphics) format ဟာ အရောင် မြောက်များစွာကို support လုပ်ပါတယ်။ ဒါ့အပြင် file အရွယ်အစားကိုပါ ချို့ပေးထားပါတယ်။ PNG ကို ပုံ အမျိုးမျိုးအတွက် ရွေးချယ် နိုင်ပါတယ်။ JPEG လိုပဲ PNG ဟာ 24 bit color ကို support လုပ်ပါတယ်။ ဒါ့အပြင် အရောင်အနည်းငယ်သာ ပါဝင်တဲ့ ပုံတွေ ကိုလည်း GIF ကဲ့သို့ save လုပ်နိုင်ပါတယ်။ GIF လိုပဲ background transparent ပုံတွေကို သိမ်းဆည်းနိုင်ပါတယ်။ PNG ဟာ GIF, JPEG တို့လို browser အားလုံးနီးပါးမှာ support လုပ်ပါတယ်။ PNG ကို .png file extension နဲ့သိမ်းပါတယ်။

WebP and AVIF

WebP နဲ့ AVIF တို့ဟာ modern browser တွေမှာ အသုံးများလာတဲ့ image format တွေပါ။ ပုံ အရည်အသွေးကောင်းကောင်းကို file size သေးသေးနဲ့ ထိန်းနိုင်တာကြောင့် website loading speed အတွက် အကျိုးရှိပါတယ်။ Browser support မသေချာတဲ့ audience အတွက်တော့ `<picture>` tag နဲ့ fallback image ထည့်ထားနိုင်ပါတယ်။

Downloading Considerations

Browser ဟာ web page မှာ ပုံတွေ ပြသဖို့ရန်အတွက် ပုံတွေကို ဦးစွာ download ဆွဲချရပါတယ်။ တကယ်လို့ image size ဟာ ကြီးနေရင်တော့ ချက်ခြင်း မပြသနိုင်ပဲ အချိန် အနည်းငယ်ကြာမြင့် ပါလိမ့်မယ်။ Internet connection ရဲ့ speed ပေါ်မှာ မူတည်ပြီး download ချတဲ့ အချိန်ကြာမြင့် တတ်ပါတယ်။ ပုံအကြီးကြီးတွေကို မိမိ computer ထဲမှာ ကြည့်လျှင် မြန်ဆန်ပေမယ့် internet ပေါ်ရောက်သွားပြီး web server ပေါ်ကနေ download ချတဲ့ အခါ ကြာမြင့်ပါလိမ့်မယ်။ ဒါကြောင့် website တည်ဆောက်ရာမှာ ပုံတွေကို တတ်နိုင်သလောက် file အရွယ်အစား သေးအောင်ဖန်တီး ထားသင့်ပါတယ်။ ပုံအကြီးကြီးတွေ အများကြီးပါခဲ့ရင်တော့ download လုပ်နေတာကြောင့် သင့် website ဟာ browser ပေါ်မှာ အပြည့်အစုံ ဖော်ပြဖို့အတွက် အချိန်အကြာကြီး စောင့်ဆိုင်းနေရပါ လိမ့်မယ်။

Optimize Images

Image editing programs (Photoshop, Pixelmator, GIMP) စတာတွေမှာ image quality ကို ချိန်ညှိ တာပါပါတယ်။ image quality ပေါ်မှာ မူတည်ပြီးတော့ image size တွေ ပြောင်းလဲနိုင်ပါတယ်။ တကယ်လို့ ပုံအကြီးကြီးတွေ သုံးရမယ်။ image size က 100 KB ထက်ကျော်နေရင်တော့ connection နှေးတဲ့ နေရာတွေမှာ ပုံတွေကို download ချရမှာ နှေးနေပါမယ်။ GIF သို့မဟုတ် PNG ကို အသုံးပြုရရင် colors အရေအတွက်ကို လျော့ချခြင်းဖြင့် ဖိုင်အရွယ်အစားကို သေးလိုက်နိုင် ပါတယ်။ Photoshop လိုမျိုး editor တွေမှာ save for web လိုမျိုး feature တွေပါပါတယ်။ ဒါမှ မဟုတ် Windows မှာပါတဲ့ Paint, macOS မှာပါတဲ့ Preview တို့ကလည်း image size ပြောင်းတဲ့ နေရာမှာ အသုံးပြုနိုင်ပါတယ်။

Alternative Text

အသုံးပြုသူတွေဟာ စာဖတ်ရတာ မြန်အောင် ပုံတွေကို မမြင်ရတဲ့ download ချတဲ့ setting ကို browser မှာ ထားလို့ရပါတယ်။ အဲဒီအခါမှာ ပုံတွေ ကို အသုံးပြုသူမမြင်ရပေမယ့် ပုံနဲ့ သက်ဆိုင် တဲ့ စာကို ပုံအစား ဖော်ပြလို့ရပါတယ်။ အဲဒီ အတွက် alternative text ကို အသုံးပြုပါတယ်။ Alternative text ဆိုတာဟာ user ကို ဒါ ဘာပုံလဲဆိုတာကို နားလည်စေရန် ရည်ရွယ်ပါတယ်။ Alternative ထည့်သွင်းပုံကို ဒီ အခန်းမှာ ဖော်ပြပေးမှာပါ။

Insert an Image

အခု ကျွန်တော်တို့တွေ ပုံလေးကို ထည့်ရအောင်။ ပြီးခဲ့တဲ့ code မှာ onepiece_cover.jpg ဆိုတာ လေးကို ထည့်ပါမယ်။ အောက်မှာ code လေးကို ကြည့်လိုက်ပါ။

```
<html>
<head>
  <title>Styling Text</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <h1>One Piece</h1>
  <h2>by Eiichiro Oda</h2>
  <ul class="menu">
    <li><a href="/home.html">Home</a></li>
    <li><a href="/anime.html">Anime</a></li>
    <li><a href="/download.html">Download</a></li>
  </ul>
  
  <p>One Piece is a Japanese <span class='highlight'>shounen manga series</span>
written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump
since August 4, 1997; the individual chapters are being published in tankōbon volumes by
Shueisha, with the first released on December 24, 1997, and the 70th volume released as
of March 2013. One Piece follows the adventures of <span class='highlight'>Monkey D.
Luffy</span>, a young boy whose body gains the properties of rubber after unintentionally
eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy
explores the ocean in search of the world's ultimate treasure known as One Piece in order
to become the next Pirate King.</p>
</body>
</html>
```

အထက်ပါ code မှာ `` ဆိုတာလေးကို တွေ့ပါလိမ့်မယ်။ အဲဒီ code လေးက image ထည့်သွင်းမယ်လို့ ဆိုထားတာပါ။ image path ကို src မှာ ရေးရပါတယ်။ လက်ရှိ `index.html` နဲ့ ပုံနဲ့ သိမ်းထားတဲ့ နေရာ အတူတူမှာ ရှိတဲ့ အတွက်ကြောင့် src မှာ ပုံရဲ့ file name တစ်ခါတည်းရေးလို့ရပါတယ်။

လက်ရှိ file တွေက အောက်ဖော်ပြပါ ပုံအတိုင်း ရှိပါတယ်။

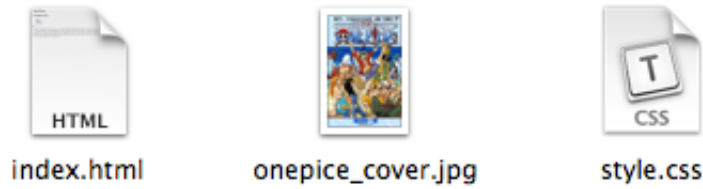


Figure 5-1. File Structure

အထက်ပါ code ကို browser မှာကြည့်ရင် အောက်ကလို မြင်ရပါလိမ့်မယ်။

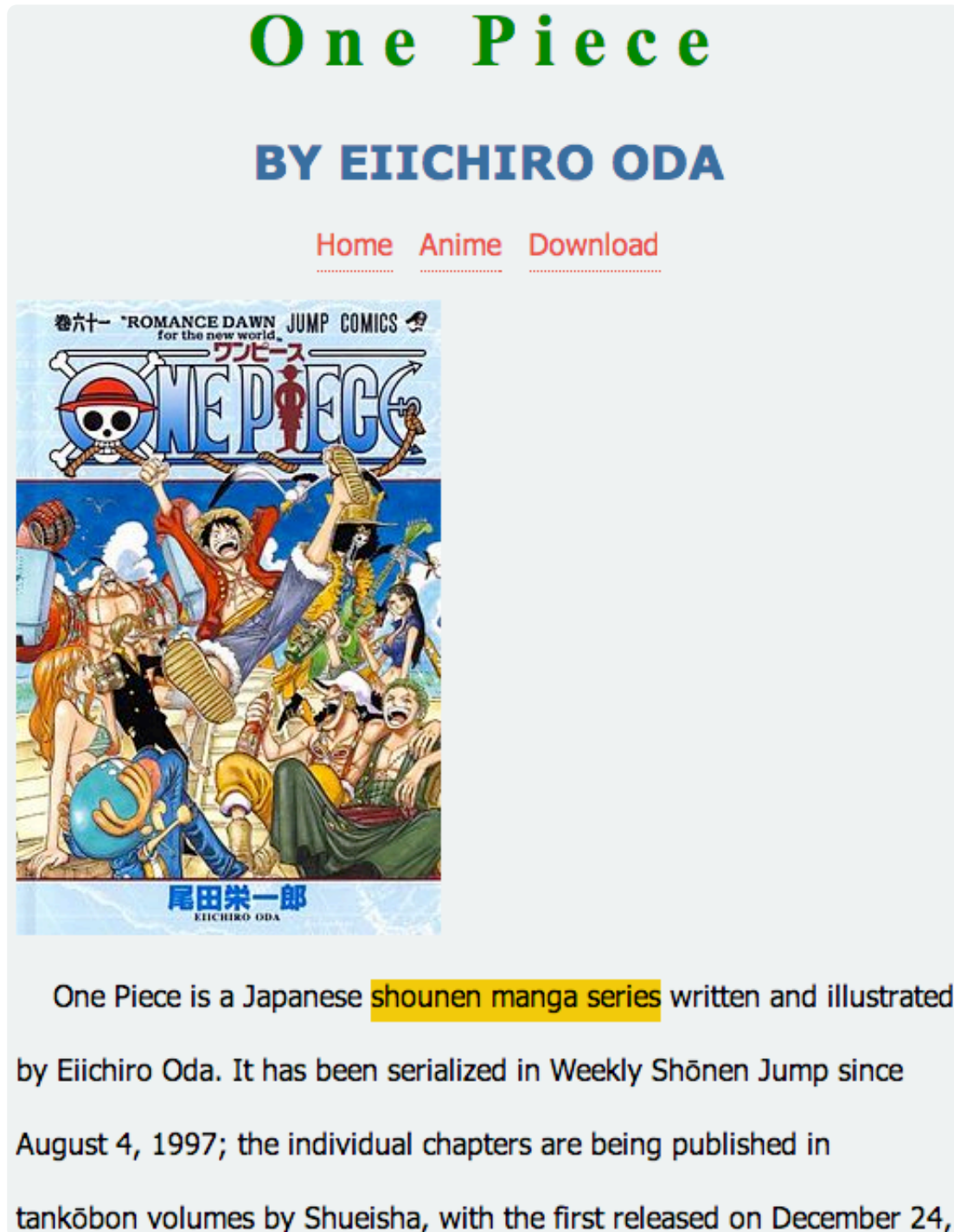


Figure 5-2

လက်ရှိ code ကိုပဲ အောက်မှာ နောက်ထပ် ပုံ တစ်ပုံ ထည့်ပါမယ်။

```
<html>
<head>
  <title>Styling Text</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <h1>One Piece</h1>
  <h2>by Eiichiro Oda</h2>
  <ul class="menu">
    <li><a href="./home.html">Home</a></li>
    <li><a href="./anime.html">Anime</a></li>
    <li><a href="./download.html">Download</a></li>
  </ul>
  
  <p>One Piece is a Japanese <span class='highlight'>shounen manga series</span>
written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump
since August 4, 1997; the individual chapters are being published in tankōbon volumes by
Shueisha, with the first released on December 24, 1997, and the 70th volume released as
of March 2013. One Piece follows the adventures of <span class='highlight'>Monkey D.
Luffy</span>, a young boy whose body gains the properties of rubber after unintentionally
eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy
explores the ocean in search of the world's ultimate treasure known as One Piece in order
to become the next Pirate King.</p>
  
</body>
</html>
```

အထက်ပါ code မှာ `` ဆိုပြီး ထည့်ထားတာကို တွေ့ရပါမယ်။
images ဆိုတဲ့ folder အောက်က luffy.png ကို သုံးမယ်လို့ ဆိုထားတာပါ။

လက်ရှိ file တွေက အောက်ဖော်ပြပါ ပုံအတိုင်း ရှိပါတယ်။

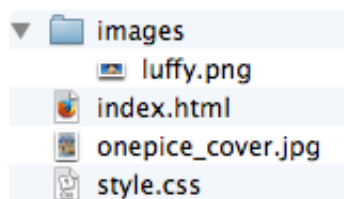


Figure 5-3. File Structure

အထက်ပါ code ကို browser မှာကြည့်ရင် အောက်ကလို မြင်ရပါလိမ့်မယ်။

Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.



Figure 5-4

Image Size

ပုံတွေ ရဲ့ အရွယ်အစားပြောင်းချင်ရင်တော့ width နဲ့ height attribute ကို အသုံးပြုနိုင်ပါတယ်။

```

```

အထက်ပါ code မှာ width ကို 150px ပေးထားပါတယ်။ height က width ရဲ့ size ပေါ်မှာ မူတည်ပြီး ratio အတိုင်း အလိုလို ပြောင်းပေးပါတယ်။ တကယ်လို့ height ကိုလည်း ပုံသေ value ပေးချင်တယ်ဆိုရင်တော့ height attribute မှာ value ထည့်ပေးဖို့လိုပါတယ်။

```

```

အထက်ပါ code မှာ ဆိုရင် image ရဲ့ width ကော height ကော က 160px ဖြစ်သွားပါလိမ့်မယ်။ pixel အပြင် value ကို percentage နဲ့လည်း ပေးလို့ရပါတယ်။

```

```

လက်ရှိ ပုံ အရွယ်အစားရဲ့ 80% ထိသာ ပြတော့မှာပါ။ တကယ်လို့ ပုံ အရွယ်အစားက 100px ဖြစ်ရင် ပြတဲ့ အခါ ပုံအရွယ်အစားကို 80px အနေဖြင့်သာ ပြပါလိမ့်မယ်။ အခု ကျွန်တော်တို့ရဲ့ code ကို အောက်ကလို ပြင်လိုက်ပါမယ်။

```
<html>
<head>
  <title>Styling Text</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <h1>One Piece</h1>
  <h2>by Eiichiro Oda</h2>
  <ul class="menu">
    <li><a href="/home.html">Home</a></li>
    <li><a href="/anime.html">Anime</a></li>
    <li><a href="/download.html">Download</a></li>
  </ul>
  
  <p>One Piece is a Japanese <span class='highlight'>shounen manga series</span>
written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump
since August 4, 1997; the individual chapters are being published in tankōbon volumes by
Shueisha, with the first released on December 24, 1997, and the 70th volume released as
of March 2013. One Piece follows the adventures of <span class='highlight'>Monkey D.
Luffy</span>, a young boy whose body gains the properties of rubber after unintentionally
eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy
explores the ocean in search of the world's ultimate treasure known as One Piece in order
to become the next Pirate King.</p>
  
</body>
</html>
```

onepiece_cover.jpg ကို width 150px နဲ့ ပြပါလို့ရေးထားပါတယ်။ ဒါကြောင့် browser မှာ ကြည့်လိုက်ရင် ပုံ size သေးသွားတာကို တွေ့ပါလိမ့်မယ်။



Figure 5-5

Alternative Text

Alternative text ကို ပုံကို download ချဖို့ ပိတ်ထားတဲ့ အခါ ၊ ဒါမှမဟုတ် ပုံရှာမတွေ့တဲ့ အခါမှာ ဖော်ပြဖို့အတွက် အသုံးပြုပါတယ်။

```

<html>
<head>
  <title>Styling Text</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <h1>One Piece</h1>
  <h2>by Eiichiro Oda</h2>
  <ul class="menu">
    <li><a href="/home.html">Home</a></li>
    <li><a href="/anime.html">Anime</a></li>
    <li><a href="/download.html">Download</a></li>
  </ul>
  
  <p>One Piece is a Japanese <span class='highlight'>shounen manga series</span>
written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump
since August 4, 1997; the individual chapters are being published in tankōbon volumes by

```

Shueisha, with the first released on December 24, 1997, and the 70th volume released as of March 2013. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.</p>

```

</body>
</html>
```

အထက်ပါ code မှာ cover image ကို ပုံနှာမည် အမှားထည့်ထားပါတယ်။ luffy.png မှာလည်း alt="luffy" ဆိုပြီး ထည့်ထားတာ ရှိပါတယ်။ ပုံ ရဲ့ လမ်းကြောင်း အမှားထည့်ထားတာ ကြောင့် browser မှာ ပုံ မမြင်ရပါဘူး။ ထည့်ထားတဲ့ alternative text ကို သာမြင်ရပါလိမ့်မယ်။



Figure 5-6

Image Label

အခု image ကို မှန်တဲ့ URL ထည့်ပြီးတော့ title attribute ကို အောက်ကလိုထည့်ပါမယ်။

```

```

`title="One Piece Cover"` ဆိုတာကတော့ ဒီ image ရဲ့ ခေါင်းစဉ်ပါ။ mouse pointer ထောက်လိုက်ရင် tooltip အနေနဲ့ ပေါ်လာမယ့် စာပါ။

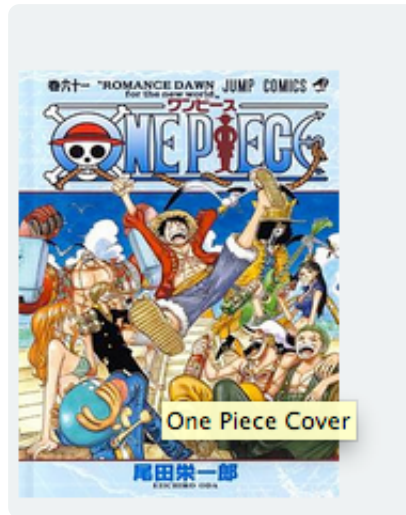


Figure 5-7

ပုံတွေမှာ mouse ထောက်လိုက်ပြီးတော့ ဘာ ပုံလည်းဆိုတာကို ညွှန်းတဲ့ အနေနဲ့ အသုံးပြုပါတယ်။ title ဟာ search engine လိုမျိုးတွေမှာ ပုံကို ရှာတွေ့ဖို့လည်း အထောက်အကူပြုပါတယ်။ ဒါကြောင့် ပုံတွေကို ထည့်သွင်းရာမှာ alternative text နဲ့ title နှစ်ခုလုံးကို အသုံးပြုသင့်ပါတယ်။

Add an Image as a Figure

`<figure>` ကို အဓိက စာတွေနဲ့ အတူ ပုံတွေကို ခေါင်းစီးနဲ့ တွဲပြီး ပြဖို့အတွက် အသုံးပြုနိုင်ပါတယ်။ ဥပမာ ။။ ကျောင်းစာတွေထဲမှာ ပုံကို ရည်ညွှန်းပြီးတော့ ပုံတွေ ဇယားတွေ စတာတွေ ပါလာတတ်ပါတယ်။ အဲဒီအခါမှာ ကျွန်တော်တို့တွေ အနေနဲ့ `<figcaption>` ကို `<figure>` အထဲမှာ ထည့်ပြီး ညွှန်းထားတဲ့ ခေါင်းစဉ် အနေနဲ့ ဖော်ပြလိုရပါတယ်။

index.html မှာ figure ထပ်ဖြည့်လိုက်ပါတယ်။

```
<html>
<head>
  <title>Styling Text</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <h1>One Piece</h1>
  <h2>by Eiichiro Oda</h2>
  <ul class="menu">
```

```

<li><a href="./home.html">Home</a></li>
<li><a href="./anime.html">Anime</a></li>
<li><a href="./download.html">Download</a></li>
</ul>

<p>One Piece is a Japanese <span class='highlight'>shounen manga series</span>
written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump
since August 4, 1997; the individual chapters are being published in tankōbon volumes by
Shueisha, with the first released on December 24, 1997, and the 70th volume released as
of March 2013. One Piece follows the adventures of <span class='highlight'>Monkey D.
Luffy</span>, a young boy whose body gains the properties of rubber after unintentionally
eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy
explores the ocean in search of the world's ultimate treasure known as One Piece in order
to become the next Pirate King.</p>
<figure>

<figcaption>Luffy</figcaption>
</figure>
</body>
</html>

```

ပြီးတော့ style.css မှာ figure နဲ့ figurecaption ကို ထပ်ဖြည့်ပါတယ်။

```

body {
    font-family: "Zawgyi-One",arial;
    background-color: #ECF0F1;
}
.highlight {
    background-color: #F1C40F;
}
h1 {
    font-size: 40px;
    color: green;
    font-family: "Times New Roman",serif;
    text-align: center;
    letter-spacing: 10px;
}
h2 {
    font-size: 25px;
    color: #336699;
    font-family: "Verdana",arial;
    text-align: center;
    text-transform: uppercase;
}
p {
    text-indent: 20px;
    line-height: 40px;
}
.menu {
    list-style: none;
    padding: 0px;
    text-align: center;
}
.menu li {

```

```

    display: inline;
    padding: 5px;
}

.menu li a{
    text-decoration: none;
    color:#E74C3C;
    border-bottom: 1px dotted #E74C3C;
}

figure {
    padding:10px;
    padding-bottom: 0px;
    width: 230px;
    background-color:#16A085;
}

figcaption {
    padding-top:8px;
    font-weight: bold;
    text-align: center;
    color:#FFF;
}

```

အဲဒီ code ကို browser မှာဆိုရင် အောက်ပုံလိုမြင်ရပါမယ်။

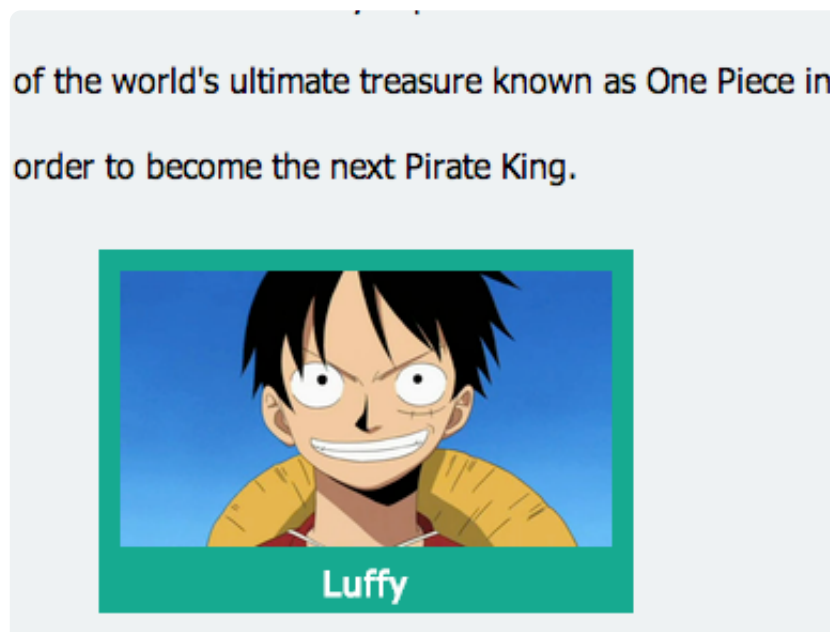


Figure 5-8

figure ရဲ့ width ကို 230px ပေးထားပါတယ်။ background ကို အစိမ်းရင့်ရောင် သုံးထားပါတယ်။ figure ရဲ့ caption ကိုတော့ အဖြူရောင် သုံးပြီး bold လုပ်ထားပါတယ်။ figure နဲ့ figcaption ဟာ css နဲ့ တွဲပြီး သုံးမှသာ အဆင်ပြေပါတယ်။

Align an Image Horizontally

CSS မှာ float ဆိုတဲ့ property က image ကို စာရဲ့ horizontal positioning ချိန်ညှိရာမှာ အသုံးပြုနိုင်ပါတယ်။ float အတွက် left ဆိုရင် စာတွေဟာ ပုံရဲ့ ညာဘက်မှာ ပေါ်မှာ ဖြစ်ပြီး float right ဆိုရင် တော့ စာတွေဟာ ဘယ်ဘက် ရောက်သွားမယ်။ အောက်မှာ style.css ကို ပြင်ထားတာလေး ကြည့်လိုက်ပါ။

style.css မှာ cover class ကို ထပ်ဖြည့်လိုက်တယ်။

```
.cover {
  float: right;
  padding-left: 10px;
}
```

index.html က cover ပုံမှာ css class ဖြည့်လိုက်ပါတယ်။

```

```

အဲဒီ အခါမှာ အောက်ကလို မြင်ရပါတယ်။

One Piece

BY EIICHIRO ODA

[Home](#) [Anime](#) [Download](#)

One Piece is a Japanese **shounen** **manga series** written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 70th volume released as of March 2013. One Piece follows the adventures of **Monkey D. Luffy**, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores



Figure 5-9

ပုံဟာ ညာဘက် အခြမ်းကပ်သွားပြီး စာသားနဲ့ padding 10px ခွာထားပါတယ်။

Align an Image Vertically

ပုံနှင့်စာ တွဲပြီးတော့ ပြချိန်မှာ စာ ကို ပုံ နဲ့ align ချိန်မယ်ဆိုရင်တော့ vertical-align ကို အသုံးပြု နိုင်ပါတယ်။ middle | top | bottom ဆိုပြီး ရှိပါတယ်။

css မှာ အောက်က code လေး ထပ်ဖြည့်လိုက်ပါ။

```
.middle {
  vertical-align: middle;
}
```

html မှာတော့ luffy image ကို ပုံမှန် ရိုးရိုးပဲထားပါမယ်။ middle class ကို အသုံးပြုပါမယ်။

```
luffy
```

စာက ပုံရဲ့ အလယ်ကို ရောက်နေတာ တွေ့ပါလိမ့်မယ်။ တကယ်လို့ ထိပ်ပိုင်းမှာ ရှိချင်ရင် top ကို အသုံးပြုပြီးတော့ အောက်ပိုင်းရောက်စေချင် bottom ကို အသုံးပြုရပါတယ်။



Figure 5-10

Center an Image

အခု image ကို စာ ရဲ့ center မှာ ပေါ်အောင် လုပ်ကြည့်ပါမယ်။

css မှာ text-center ဆိုတဲ့ class ထပ်ဖြည့်ပါမယ်။

```
.text-center {
  text-align: center;
}
```

index.html ကိုတော့ အောက်ကလို ပြောင်းလိုက်မယ်။

```
<html>
<head>
  <title>Styling Text</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <h1>One Piece</h1>
  <h2>by Eiichiro Oda</h2>
  <ul class="menu">
    <li><a href="./home.html">Home</a></li>
    <li><a href="./anime.html">Anime</a></li>
    <li><a href="./download.html">Download</a></li>
  </ul>
  
  <p>One Piece is a Japanese <span class='highlight'>shounen manga series</span>
  written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump
  since August 4, 1997; the individual chapters are being published in tankōbon volumes by
```

Shueisha, with the first released on December 24, 1997, and the 70th volume released as of March 2013. </p>

```
<div class='text-center'></div>
```

```
<p>One Piece follows the adventures of <span class='highlight'>Monkey D.
Luffy</span>, a young boy whose body gains the properties of rubber after unintentionally
eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy
explores the ocean in search of the world's ultimate treasure known as One Piece in order
to become the next Pirate King.</p>
```

```
</body>
```

```
</html>
```

အဲဒီ code ထဲမှာ

```
<div class='text-center'></div>
```

ဆိုတာက div ကို အသုံးပြုပြီးတော့ image ကို text-align:center နဲ့ အလယ်ကို ပို့လိုက်တာပါ။
image ကိုသာမက text တွေကိုလည်း text-align:center ကို သုံးပြီး စာကို အလယ်မှာ ပေါ်အောင်
ဖန်တီးနိုင်ပါတယ်။ အဲဒီ code လေးကို အောက်ကလို မြင်ရပါမယ်။

One Piece

BY EIICHIRO ODA

[Home](#) [Anime](#) [Download](#)

One Piece is a Japanese **shounen manga** **series** written and illustrated by Eiichiro Oda.

It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by



Shueisha, with the first released on December 24, 1997, and the 70th volume released as of March 2013.



One Piece follows the adventures of **Monkey D. Luffy**, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

Figure 5-11

Stop Text Wrap

အောက်က ပုံကို ကြည့်လိုက်ပါ။ ပထမ အပိုဒ်ပြီးတော့ ဒုတိယ အပိုဒ်က ပုံနဲ့ ကပ်နေပါတယ်။ ပုံ အောက်မှာသာ ပေါ်ချင်တယ်ဆိုရင် ကျွန်တော်တို့တွေ clear ဆိုတာကို အသုံးပြုရပါတယ်။ clear မှာ left | right | both ဆိုပြီး ရှိပါတယ်။ ၂ ခြမ်း လုံး ရှင်းမယ်ဆိုရင်တော့ both ကို အသုံးပြုပါတယ်။

One Piece is a Japanese **shounen manga series** written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 70th volume released as of March 2013.

One Piece follows the adventures of **Monkey D. Luffy**, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

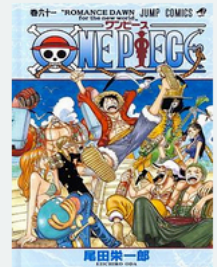


Figure 5-12

style.css မှာ clear ဆိုတဲ့ class ထပ်ဖြည့်ပါတယ်။

```
.clear {
    clear: both;
}
```

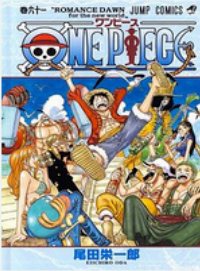
index.html မှာတော့ clear ကို အသုံးပြုထားတဲ့ div ကို ထည့်လိုက်ပါတယ်။

```
<html>
<head>
  <title>Styling Text</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <h1>One Piece</h1>
  <h2>by Eiichiro Oda</h2>
  <ul class="menu">
    <li><a href="/home.html">Home</a></li>
    <li><a href="/anime.html">Anime</a></li>
    <li><a href="/download.html">Download</a></li>
  </ul>
  
  <p>One Piece is a Japanese <span class='highlight'>shounen manga series</span>
  written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump
  since August 4, 1997; the individual chapters are being published in tankōbon volumes by
  Shueisha, with the first released on December 24, 1997, and the 70th volume released as
  of March 2013. </p>
  <div class="clear"></div>
  <p>One Piece follows the adventures of <span class='highlight'>Monkey D.
  Luffy</span>, a young boy whose body gains the properties of rubber after unintentionally
  eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy
  explores the ocean in search of the world's ultimate treasure known as One Piece in order
```

```
to become the next Pirate King.</p>
</body>
</html>
```

အဲဒီ အခါ စာက အောက်ကလို ပုံစံ ဖြစ်သွားပါမယ်။

One Piece is a Japanese **shounen manga series** written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 70th volume released as of March 2013.



One Piece follows the adventures of **Monkey D. Luffy**, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

Figure 5-13

Add Space around an Image

ပုံနဲ့ စာနဲ့ မကပ်ပဲ ခွာနေအောင် padding ကို အသုံးပြုနိုင်ပါတယ်။ cover image မှာ padding သုံးခဲ့တာ မှတ်မိပါလိမ့်မယ်။ padding မှာ padding-left | padding-right | padding-top | padding-bottom ဆိုပြီးရှိပါတယ်။

padding တကြောင်းထဲ လည်း ရေးလို့ရပါတယ်။

```
padding:25px 50px 75px 100px
```

မှာ top က 25px , right က 50px , bottom က 75px နှင့် left က 100px ပါ။

```
padding:25px 50px 75px;
```

မှာ top 25px, right နှင့် left က 50px ။ bottom ကတော့ 75px ပါ။

```
padding:25px 50px;
```

မှာ top နဲ့ bottom က 25px, right နှင့် left က 50px ပါ။

```
padding:25px;
```

မှာ padding အားလုံးက 25px ပါ

Padding ကို image တွေသာမက text အတွက် div ,span စတာတွေမှာလည်း အသုံးပြုနိုင်ပါတယ်။

Add a Background Image

အခု body ရဲ့ နောက်ခံ ထည့်ဖို့အတွက် tree_bark.png ဆိုတဲ့ image ကို အသုံးပြုပါမယ်။



Figure. tree_bark.png

tree_bark.png ကို images folder အောက်မှာ ထားပါမယ်။ ပြီးရင် style.css က body မှာ background-image ထည့်ပါမယ်။

```
body {
  font-family: "Zawgyi-One", arial;
  background-color: #ECF0F1;
  background-image: url("../images/tree_bark.png");
}
```

အဲဒီ အခါ အောက်ကလို မြင်ရပါမယ်။

One Piece

BY EIICHIRO ODA

[Home](#) [Anime](#) [Download](#)

One Piece is a Japanese **shounen manga series** written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 70th volume released as of March 2013.

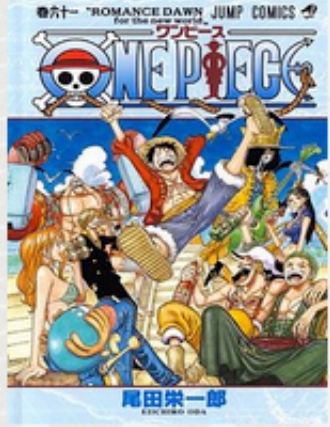


Figure 5-14

Pattern ဖြစ်တဲ့အတွက် x position ကော y position ပါ fill လုပ်သွားတာပါ။ တကယ်လို့ x position တစ်ခုပဲ fill လုပ်စေချင်ရင်

```
background-repeat: repeat-x;
```

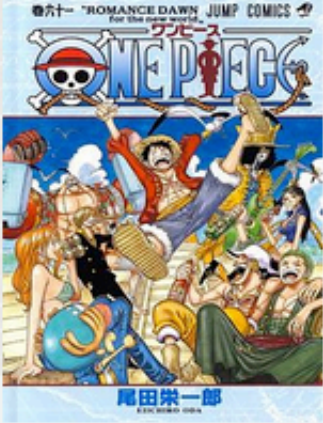
ဆိုပြီး သုံးနိုင်ပါတယ်။ အဲဒီ အခါ အလျားလိုက် ပဲ fill လုပ်သွားပါမယ်။

One Piece

BY EIICHIRO ODA

[Home](#) [Anime](#) [Download](#)

One Piece is a Japanese **shounen manga series** written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 70th volume released as of March 2013.



One Piece follows the adventures of **Monkey D. Luffy**, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

Figure 5-15

ဒေါင်လိုက်ပဲ fill လုပ်ချင်ရင်တော့

`background-repeat: repeat-y;`

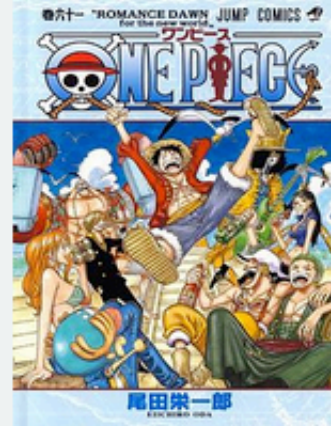
ကို အသုံးပြုနိုင်ပါတယ်။

One Piece

BY EIICHIRO ODA

[Home](#) [Anime](#) [Download](#)

One Piece is a Japanese **shounen manga series** written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 70th volume released as of March 2013.



One Piece follows the adventures of **Monkey D. Luffy**, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

Figure 5-16

default value က repeat ပါ။ တကယ်လို့ ပုံကို repeat မလုပ်ချင်ရင်

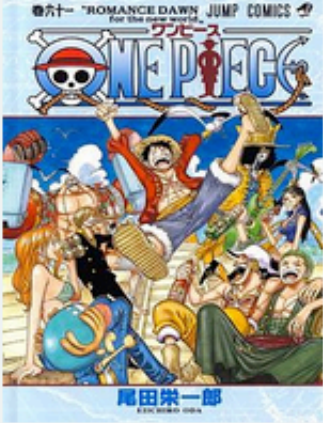
`background-repeat: no-repeat;`

One Piece

BY EIICHIRO ODA

[Home](#) [Anime](#) [Download](#)

One Piece is a Japanese **shounen manga series** written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 70th volume released as of March 2013.



One Piece follows the adventures of **Monkey D. Luffy**, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

Figure 5-17

Add a Meter Image

ကျွန်တော်တို့ အနေနဲ့ `<meter>` tag ကို လက်ရှိ range value ကို ဖော်ပြတဲ့ နေရာမှာ အသုံးပြုနိုင်ပါတယ်။

```
<meter>54 points</meter>
```

ဆိုရင် လက်ရှိ value မပါရှိတဲ့ အတွက် အောက်ကလို သာ ဖော်ပြပါလိမ့်မယ်။



Figure 5-18

```
<meter value="38" min="0" max="100">38 points</meter>
```

value က 38 ဖြစ်ပြီး အနိမ့်ဆုံးက 0 ။ အမြင့်ဆုံးက ၁၀၀ ဖြစ်တယ်လို့ ဆိုထားပါတယ်။



Figure 5-19

အဲဒီ code ကို low , high , optimum attribute တွေ ထည့်သွင်းပါမယ်။ အနိမ့် ၊ အမြင့် နှင့် အမြင့်ဆုံးပေါ့။ အနီ ၊ အဝါ နှင့် အစိမ်းရောင်တွေ နဲ့ ဖော်ပြပါလိမ့်မယ်။



Figure 5-20

Loading အနေနဲ့ ပြချင်တာ ။ progress အနေနဲ့ အလုပ်လုပ်စေချင်ရင်တော့ meter အစား progress tag ကို သုံးသင့်ပါတယ်။

```
$0 <progress value="850" max="1000">$850</progress> $1000
```

အဲဒီ code က chrome မှာ ဆိုရင် အောက်ကလို မြင်ရပါမယ်။

One Piece is a Japanese **shounen manga series** written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 70th volume released as of March 2013.

One Piece follows the adventures of **Monkey D. Luffy**, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

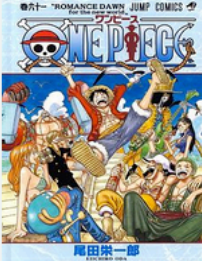


Figure 5-21

meter ကို လက်တွေ့ အသုံးချထားတဲ့ code ကို လေ့လာကြည့်ရ အောင်

```
<html>
<head>
<title>Meter</title>
```

```

    <style>
    @import url(http://fonts.googleapis.com/css?family=Roboto);

body {

    font-family: "Roboto", "Lucida Grande", "Lucida", sans-serif;
    margin: 5em auto;
    background: radial-gradient(#f9f9f9 20%, #ddd 80%);
    width: 550px;
}

h1 {

    font-size: 1.5em;
    margin: 0 auto 2.5em;
    font-weight: 700;
}

pre {

    font-family: Consolas, Monaco, monospace;
    font-size: 85%;
    color: #666;
    font-weight: 400;
}

meter {

    display: block;
    margin: 0 auto;

    width: 550px;
    height: 25px;

    -webkit-appearance: none;
    -moz-appearance: none;
    appearance: none;

    border: 1px solid #ccc;
    border-radius: 3px;

    /* Firefox */
    background: none; /* Required to get rid of the default background prop. */
    background-color: whiteSmoke;
    box-shadow: 0 5px 5px -5px #333 inset;
}

meter::-webkit-meter-bar {

    background: none; /* Important to get rid of default background. */
    background-color: whiteSmoke;

    box-shadow: 0 5px 5px -5px #333 inset;
}

meter::-webkit-meter-optimum-value {

    transition: width .5s;
    box-shadow: 0 5px 5px -5px #999 inset;

    background-image: linear-gradient( 90deg,
        #8bcf69 5%,
        #e6d450 5%,
        #e6d450 15%,

```

```

        #f28f68 15%,
        #f28f68 55%,
        #cf82bf 55%,
        #cf82bf 95%,
        #719fd1 95%,
        #719fd1 100%);

    background-size: 100% 100%;
}

meter::-webkit-meter-optimum-value:hover {

    background-image: linear-gradient( 90deg,
        #8bcf69 20%,
        #e6d450 20%,
        #e6d450 40%,
        #f28f68 40%,
        #f28f68 60%,
        #cf82bf 60%,
        #cf82bf 80%,
        #719fd1 80%,
        #719fd1 100%);

    transition: width .5s;
    width: 100% !important; /* !important keyword used to override the inline style in
ebkit browsers. */
}

meter::-moz-meter-bar {

    box-shadow: 0 5px 5px -5px #999 inset;

    background-image: linear-gradient( 90deg,
        #8bcf69 5%,
        #e6d450 5%,
        #e6d450 15%,
        #f28f68 15%,
        #f28f68 55%,
        #cf82bf 55%,
        #cf82bf 95%,
        #719fd1 95%,
        #719fd1 100%);

    background-size: 100% 100%;
}

p {

    margin: 0 auto ;
    width: 550px;
    line-height: 2.5;
}

.free-space {

    float: right;
}

.swatch {

    margin: 0; padding:0;
    margin: 5em auto;
    list-style-type: none;
    width: 550px;

```

```

    display: block;
    position: relative;
}

.swatch::before {

    content: '';
    width: 100%;
    height: 10px;
    outline: 0px solid #000;
    position: absolute;
    top: -20px;
    left: 0;
    border-radius: 2px;

    background-image: linear-gradient( 90deg,
                                        #8bcf69 20%,
                                        #e6d450 20%,
                                        #e6d450 40%,
                                        #f28f68 40%,
                                        #f28f68 60%,
                                        #cf82bf 60%,
                                        #cf82bf 80%,
                                        #719fd1 80%,
                                        #719fd1 100%);

}

.swatch__elem {

    outline: 0px solid #c00;
    float: left;
    width: 110px;
    padding-left: 5px;

    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    box-sizing: border-box;
}

.used-space {

    display: block;
    line-height: 2;
    font-size: 85%;
    color: #666;
}

.meter-gauge {

    border: 1px solid #ccc;
    border-radius: 3px;

    background-color: whiteSmoke;
    box-shadow: 0 5px 5px -5px #333 inset;

    width: 550px;
    height: 25px;

    display: block;
}

.meter-gauge > span {

```

```

height: inherit;
box-shadow: 0 5px 5px -5px #999 inset;

background-color: blue;
background-image: linear-gradient( 90deg,
                                   #8bcf69 5%,
                                   #e6d450 5%,
                                   #e6d450 15%,
                                   #f28f68 15%,
                                   #f28f68 55%,
                                   #cf82bf 55%,
                                   #cf82bf 95%,
                                   #719fd1 95%,
                                   #719fd1 100%);

background-size: 100% 100%;

display: block;
text-indent: -9999px;
}
</style>
</head>
<body>
  <h1>OSX-style Disk Usage <xmp>..using html5 <meter></xmp></h1>

<p>
  Macintosh HD <span class="free-space">64.54 GB free out of 120.47 GB</span></p>

<meter value="55.93" min="0" max="120.47" title="GB">
  <div class="meter-gauge">
    <span style="width: 46.42%;">Disk Usage - 55.93GB out of 120GB</span>
  </div>
</meter>

<ul class="swatch">
  <li class="swatch__elem">Audio <span class="used-space">670.5 MB</span></li>
  <li class="swatch__elem">Movies <span class="used-space">10.1 GB</span></li>
  <li class="swatch__elem">Photos <span class="used-space">25.19 GB</span></li>
  <li class="swatch__elem">Apps <span class="used-space">19.31 GB</span></li>
  <li class="swatch__elem">Other <span class="used-space">660.5 MB</span></li>
</ul>

</body>
</html>

```

အထက်ပါ code က browser မှာ ဆိုရင် အောက်ကလို မြင်ရပါမယ်။

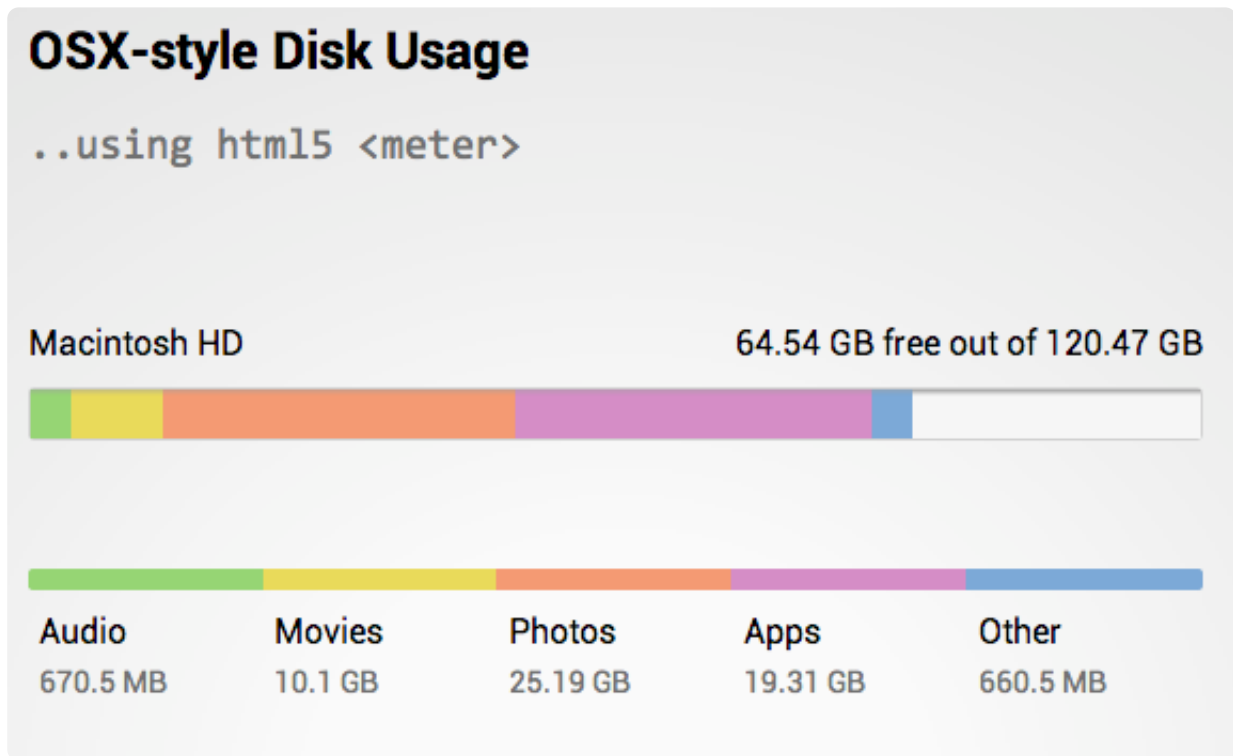


Figure 5-22

background image ကို linear-gradient အသုံးပြုပြီး meter နဲ့ ဖော်ပြထားတဲ့ example ကောင်း တစ်ခုပါ။

အခု ဆိုရင်တော့ image နဲ့ ပတ်သက်ပြီး သိသင့်သိထိုက်သလောက် သိသွားပါပြီ။ နောက်တစ်ခန်းမှာ ကျွန်တော်တို့တွေ links တွေ ချိတ်ဆက်တဲ့ အကြောင်းကို လေ့လာကြည့်ရအောင်။

Adding Links

Web page တွေ တစ်ခုနဲ့ တစ်ခု ချိတ်ဆက်ဖို့ ဘယ်လို ရေးရမလဲ ဆိုတာကို ဒီ အခန်း မှာ ဖော်ပြပေးပါမယ်။ ဒီအခန်းမှာတော့ ကိုယ့်ရဲ့ HTML file အချင်းချင်း ချိတ်ဆက်ရုံသာမက အခြား website နဲ့ link ချိတ်တာ၊ email link ချိတ်တာ၊ နောက်ပြီး link ချိတ်တဲ့ စာရဲ့ ပုံသဏ္ဍန် ပြင်တာတွေကို ဖော်ပြပေးပါမယ်။ အရင်ဆုံး ကျွန်တော်တို့တွေ အနေနဲ့ URL အကြောင်းကို နားလည်သိရှိဖို့လိုပါတယ်။ Link ကိုတော့ `<a>` ဆိုတဲ့ tag ကို သုံးပြီးတော့ href attribute နဲ့ ပေါင်းသုံးသွားမှာပါ။

Understanding Links

Hyper links ဒါမှမဟုတ် link ဆိုပြီး အတိုကောက် ခေါ်ကြပါတယ်။ လက်ရှိ html page ကနေ အခြား page တစ်ခုကို ရောက်သွားစေချင်ရင် ဒါမှမဟုတ် အခြား website ကို ရောက်စေချင်ရင်တော့ `<a>` ကို အသုံးပြုရပါတယ်။

Link ချိတ်ထားတာတွေက စာတွေ ဖြစ်နိုင်သလို ပုံတွေ လည်း ဖြစ်နိုင်ပါတယ်။ Link လုပ်ထားတဲ့ စာက ပုံမှန် အားဖြင့် အောက်ဘက်လှိုင်းတားထားပြီး ဖော်ပြတတ်ပါတယ်။ ဥပမာ ။ Navigation menu မှာ Home , Gallery , Profile စတာတွေ ရှိပါတယ်။ Gallery ကို နှိပ်လိုက်ရင် ပုံတွေထည့်ထားတဲ့ gallery.html ကို ရောက်သွားပြီး profile ကိုနှိပ်လိုက်ရင် profile.html ကို ရောက်သွားမယ်။ အဲဒီလိုမျိုး စာကို click နှိပ်လိုက်ပြီး page တွေပြောင်းသွားဖို့အတွက် စာ ကို anchor tag (`<a>`) နဲ့ တွဲသုံးရပါတယ်။

ဒီပုံလေးမှာ ဆိုရင် gallery လေးကို နှိပ်လိုက်ရင် gallery page ရောက်သွားတာလေးကို ပြထားပါတယ်။



Figure 6-1

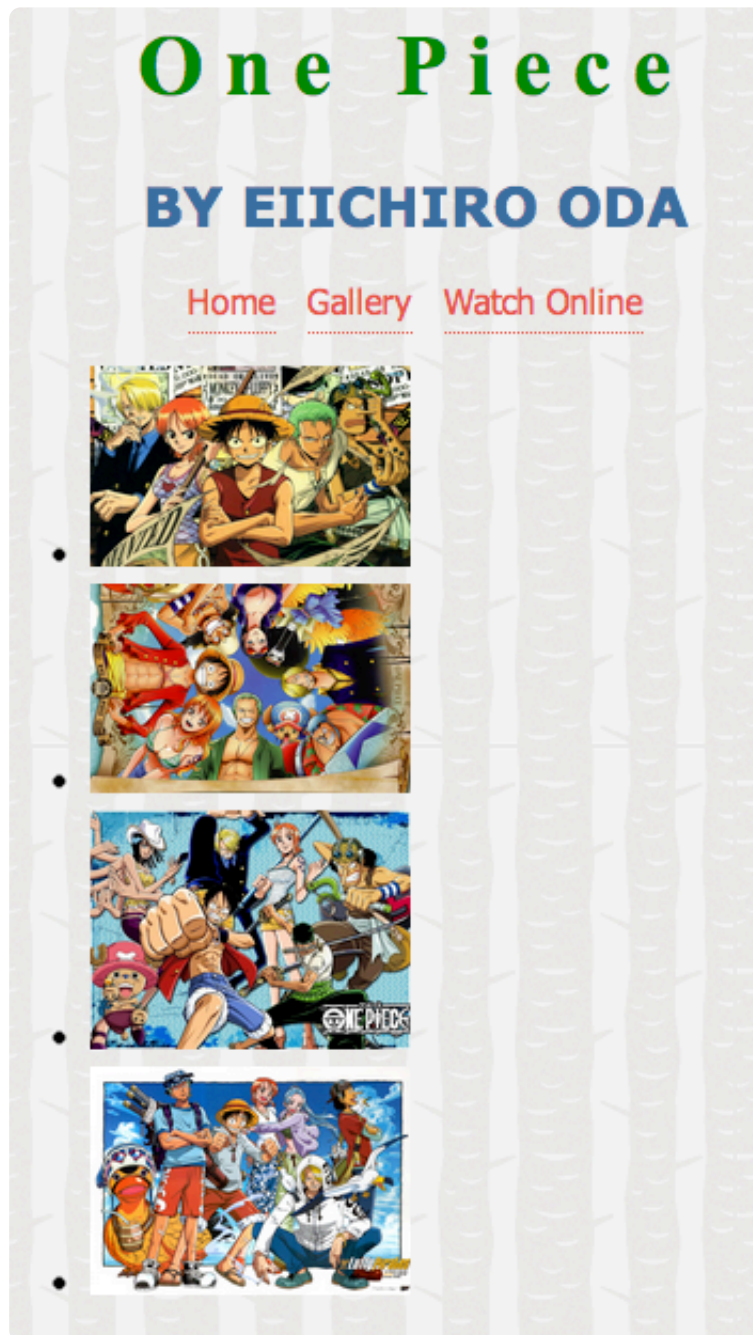


Figure 6-2

Understanding URLs

HTTP Prefix

URL ဆိုတာကတော့ Uniform Resource Locator ကို ဆိုတာပါ။ URL ဟာ domain နဲ့ web server ပေါ်မှာ ရှိတဲ့ directory path တွေကို ရည်ညွှန်းပါတယ်။ webpage ကိုလည်း ညွှန်းဆိုထားတဲ့ လိပ်စာ ဆိုလည်း မမှားပါဘူး။ ပုံမှန် website URL တွေမှာ အခုခေတ်မှာ HTTPS (<https://>) ကို သုံးသင့်ပါတယ်။ HTTPS က connection ကို encrypt လုပ်ပေးတာကြောင့် login, payment transaction, web email စတဲ့ လုံခြုံရေး အရေးကြီးတဲ့ နေရာတွေမှာ မဖြစ်မနေလိုအပ်ပါတယ်။ <http://> ကိုတော့ local testing သို့မဟုတ် legacy site တွေမှာ တွေ့ရနိုင်ပါတယ်။ FTP(File Transfer Protocol) အတွက်ကတော့ <ftp://> ဆိုပြီး အသုံးပြုပါတယ်။ website မှာ email link ကို ချိတ်မယ်ဆိုရင်တော့ <mailto:> ကို အသုံးပြုနိုင်ပါတယ်။

Domain Name

HTTP ရဲ့ နောက်မှာ domain name ပါပါတယ်။ <http://www.example.com> မှာ example.com က domain name ပါ။ www ကတော့ world wide web ရဲ့ အတိုကောက်ပါ။ domain name ရှေ့မှာ အသုံးပြုကြပါတယ်။ domain name ကို domain name server က web server IP ကို ညွှန်းထားပြီး တော့ server မှာက domain name အတွက် apache , nginx စတဲ့ webserver ကို အသုံးပြုပြီးတော့ ချိတ်ဆက်ထားပါတယ်။ ဥပမာ ။ 128.199.247.176 က server IP ဆိုပါစို့။ domain name server မှာ saturngod.net ဆိုရင် IP 128.199.247.176 ကို သွားဆိုပြီး ထည့်ထားတယ်။ ဒါကြောင့် browser မှာ saturngod.net လို့ ရိုက်လိုက်ရင် အဆင့်ဆင့်သွားပြီးတော့ နောက်ဆုံး saturngod.net ကို register လုပ်ထားတဲ့ name server ကို ရောက်သွားတယ်။ သူကနေပြီးတော့ IP address 128.199.247.176 ကို အကြောင်းပြန်လိုက်တယ်။ အဲဒီ အခါ browser က IP 128.199.247.176 ကို သွားခေါ်တယ်။ web server က domain က saturngod.net နဲ့ လာတာ ဆိုရင် ဒီ directory ထဲက index file ကို တွေ့ ပြဆိုပြီး ထုတ်ပြန်လိုက်တယ်။

အောက်မှာ အလုပ်လုပ်ပုံလေးကို [youtube video](#) မှာ သွားကြည့်နိုင်ပါတယ်။

Directory Path and File Name

Domain name ရဲ့ နောက်မှာတော့ Directory Path တွေပါလာပါတယ်။ <http://www.example.com/images/profile.png> ဆိုရင် images ဆိုတဲ့ folder အောက်က profile.png ကို ဆိုလိုတာပါ။ directory path က directory တစ်ခု ထက် မက ဖြစ်နိုင်ပါတယ်။ <http://www.example.com/images/logo/mylog.png> ဆိုရင် images folder အောက်က logo အောက် က mylog.png file ကို ရည်ညွှန်းတာပါ။

Link to Another Page

ပထမဆုံး ကျွန်တော်တို့တွေ page ၁ ခု နဲ့ တစ်ခု ချိတ်ဆက်ကြည့်တာပေါ့။ folder တစ်ခုထဲမှာ index.html နဲ့ gallery.html ဆိုပြီး file ၂ ခု ဖန်တီးလိုက်မယ်။

index.html မှာ

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Link</title>
</head>
<body>
  <h1>Home Page</h1>
  <a href="gallery.html">Gallery</a>
</body>
</html>

```

ဆိုပြီး ရှိတယ်။

gallery.html မှာ

```

<html>
<head>
  <title>Link</title>
</head>
<body>
  <h1>Gallery</h1>
  <a href="index.html">Home</a>
</body>
</html>

```

ဆိုပြီး ရှိတယ်။

ကျွန်တော်တို့ index.html ကို ဖွင့်ကြည့်ရအောင်။ အဲဒီမှာ Gallery ဆိုပြီး အပြာရောင် မျဉ်းတား ထားတဲ့ စာကို တွေ့ရပါလိမ့်မယ်။ Click လုပ်ကြည့်တဲ့ အခါမှာ gallery page ကို ရောက်သွားပါ မယ်။ Browser ရဲ့ address bar မှာလည်း link ပြောင်းသွားတာကို တွေ့ရပါမယ်။

Home Page

[Gallery](#)

Figure 6-3

Gallery

[Home](#)

Figure 6-4

Home ကို ပြန်နှိပ်ပြီးတဲ့ အခါမှာတော့ Gallery ဆိုတဲ့ စာက ခရမ်းရောင်ဖြစ်နေတာကို တွေ့ရပါ မယ်။ ဘာကြောင့်လည်းဆိုတော့ Gallery link ကို နှိပ်ထား ပြီးသားဖြစ်လို့ပါ။

ကျွန်တော်တို့ index.html ကို gallery ဆိုတဲ့ folder ထဲမှာ ဆောက်မယ်။ ပြီးတော့ index.html မှာ link ကို ပြင်လိုက်တယ်။

```
<html>
<head>
  <title>Link</title>
</head>
<body>
  <h1>Home Page</h1>
  <a href="./gallery/index.html">Gallery</a>
</body>
</html>
```

gallery folder အောက်က index.html မှာ

```
<html>
<head>
  <title>Link</title>
</head>
<body>
  <h1>Gallery</h1>
  <a href=" ../index.html">Home</a>
</body>
</html>
```

အဲဒီ code မှာ ../ ဆိုတာကို တွေ့နိုင်ပါတယ်။ ../ ဆိုတာက up ကိုပြောတာပါ။ လက်ရှိ folder ရဲ့ အပေါ်တဆင့် လို့ ဆိုလိုတာပါ။

Path

Link တွေ image တွေကို ထည့်မယ်ဆိုရင် HTML မှာ ဘယ်လိုမျိုး path တွေကို ရေးရသလဲ ဆိုတာကို သိဖို့လိုပါတယ်။

Path	Description
./	လက်ရှိ file ၏ folder ခုနဲ့ level အတူတူ ။ ဥပမာ။ ./index.html , ./sample.jpg
../	လက်ရှိ file ၏ folder ရဲ့ တစ်ဆင့် အထက် တွင် ရှိသော လမ်းကြောင်း။
../../	လက်ရှိ file ၏ folder ရဲ့ နှစ်ဆင့် အထက် တွင် ရှိသော လမ်းကြောင်း။ ဥပမာ။ index.html , sample/sub/index.html တွင် sample/sub/index.html ကနေ index.html ကို ခေါ်လိုပါက ../../index.html ဟု ခေါ်ရသည်။
/	parent သို့မဟုတ် domain အစ။ ဥပမာ။ www.example.com/gallery/sample.html တွင် link ၏ src ကို / ဟု ရေးထားပါက www.example.com သို့ ပြန်ခေါ်ထားခြင်းဖြစ်သည်။

Open a New Tab with a Link

Link တစ်ခုကို click နှိပ်ပြီး ဖွင့်လိုက်တဲ့ အခါမှာ လက်ရှိ page ကနေ တခြား page တစ်ခုကို ပြောင်းသွားပါတယ်။ လက်ရှိ page ကို မပျောက်ချင်ဘူး။ နောက်ထပ် page တစ်ခုကို tab အသစ် တစ်ခုမှာ ဖွင့်လိုတယ်ဆိုရင်တော့ a tag မှာ `target="_blank"` ကို အသုံးပြုနိုင်ပါတယ်။ External link တွေမှာတော့ လုံခြုံရေးအတွက် `rel="noopener noreferrer"` ကိုပါ ထည့်သင့်ပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Link</title>
</head>
<body>
  <h1>Home Page</h1>
  <a target="_blank" rel="noopener noreferrer" href="/gallery/index.html">Gallery</a>
</body>
</html>
```

မှာ ဆိုရင် Gallery ကို click နှိပ်လိုက်တဲ့ အခါမှာ browser မှာ tab အသစ်တစ်ခု နဲ့ ဖွင့်ပေးတာကို တွေ့ပါလိမ့်မယ်။

Make All Links Open New Tabs

a tag တစ်ခု ခြင်းဆီမှာ လိုက်မထည့် ချင်ဘူး။ a tag အားလုံးကို page တစ်ခု အနေနဲ့ ဖွင့်ချင်တယ် ဆိုရင်တော့ base tag ကို အသုံးပြုရပါမယ်။

```
<base target="_blank">
```

ကို head tag ထဲမှာ ထည့်ထားပေးဖို့လိုပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Link</title>
  <base target="_blank">
</head>
<body>
  <h1>Home Page</h1>
  <a href="/gallery/index.html">Gallery</a>
</body>
</html>
```

တကယ်လို့ new tab ဖွင့်စေချင်တယ်။ ဒါပေမယ့် အချို့ link တွေကိုတော့ ဖွင့်ထားတဲ့ tab အသစ် ထဲမှာ ပဲ ပေါ်စေချင်ရင်တော့ `target="new_tab"` ဆိုပြီး နာမည်ပေးထားလို့ ရပါတယ်။ အသစ်ဖွင့် လိုက်တဲ့ tag name က new_tab ဖြစ်ပြီးတော့ target က new_tab လေး ပေးထားတဲ့ နေရာတွေမှာပဲ အသစ်ပွင့်ပါလိမ့်မယ်။

Link to an Area on the Same Page

လက်ရှိ page ထဲမှာ ရှိတဲ့ နေရာ တစ်ခုခု ကို ပဲ link လုပ်ချင်တယ်ဆိုရင်တော့ ကျွန်တော်တို့တွေ a tag နဲ့ id တွဲ သုံးရပါလိမ့်မယ်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Link Same Page</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>One Piece</h1>
  <h2>by Eiichiro Oda</h2>
  <ul class="menu">
    <li><a href="/home.html">Home</a></li>
    <li><a href="/anime.html">Anime</a></li>
    <li><a href="/download.html">Download</a></li>
  </ul>
  
  <p>One Piece is a Japanese <span class='highlight'>shounen manga series</span>
  written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump
  since August 4, 1997; the individual chapters are being published in tankōbon volumes by
  Shueisha, with the first released on December 24, 1997, and the 70th volume released as
  of March 2013. </p>

  <div class="clear"></div>

  <p>One Piece follows the adventures of <a href="#luffy">Monkey D. Luffy</a>, a young
  boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit,
  and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in
  search of the world's ultimate treasure known as One Piece in order to become the next
  Pirate King.</p>

  <h3 id="luffy">Monkey D. Luffy</h3>
  <p>Monkey D. Luffy,[1] also known as "Straw Hat Luffy" and commonly as "Straw Hat",[4] is
  a pirate and the main protagonist of the anime and manga, One Piece. He is the son of the
  Revolutionary Army's commander, Monkey D. Dragon,[12] the grandson of the famed Marine,
  Monkey D. Garp,[13] the foster son of a mountain bandit, Curly Dadan, and the adopted
  brother of the late "Fire Fist" Portgas D. Ace and the revolutionary Sabo. His life long
  dream is to become the Pirate King by finding the legendary treasure left behind by the
  previous Pirate King, Gol D. Roger.[14] He believes that being Pirate King means one has
  the most freedom in the world. He has eaten the Gomu Gomu no Mi. As the founder and
  captain of the Straw Hat Pirates, he is the first member who makes up the crew, as well
  as one of its top three fighters. </p>

</body>
</html>

```

အထက် ပါ code မှာ `<h3 id="luffy">Monkey D. Luffy</h3>` ဆိုပြီး h3 ကို id နာမည်ပေးထားတာ တွေ့ရပါလိမ့်မယ်။

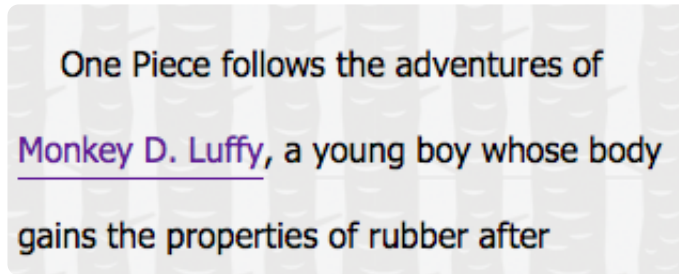


Figure 6-5.

`Monkey D. Luffy` ဆိုပြီး link ကို id luffy ကို ပေးထားတာ တွေ့ပါမယ်။
 Monkey D. Luffy ကို နှိပ်လိုက်တဲ့ အခါမှာ web page က `<h3 id="luffy">Monkey D. Luffy</h3>`
 နေရာကို ရောက်သွားတာ တွေ့ရပါလိမ့်မယ်။

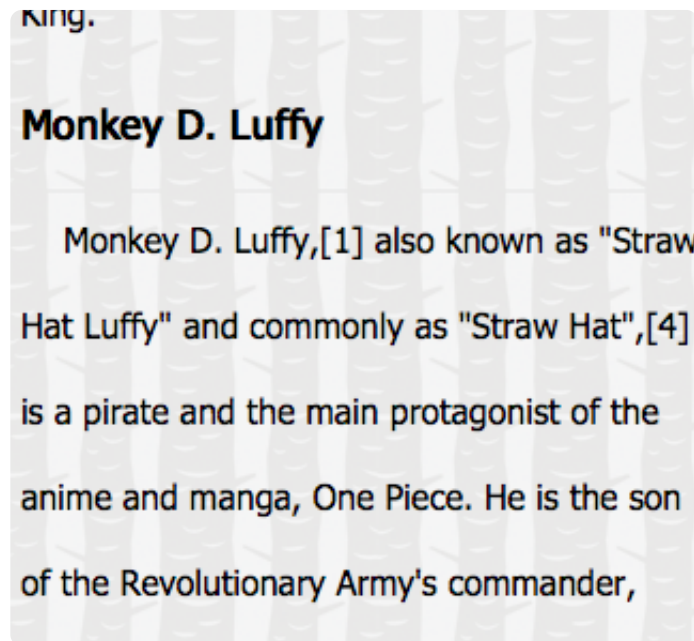


Figure 6-6.

တကယ်လို့ အခြား page တစ်ခုက id ကို ချိတ်ချင်ရင်တော့ page နာမည်နဲ့ တွဲရေးလို့ ရပါတယ်။

```
<h3 id="profile.html#luffy">Monkey D. Luffy</h3>
```

အထက်ပါ code မှာ ဆိုရင် link ကို နှိပ်လိုက်ရင် profile.html က id luffy ဆီ ရောက်သွားပါလိမ့်မယ်။

Link to Another File Type

HTML မှာ link တွေကို ချိတ်တဲ့ အခါမှာ .html file ကို သာမက image file တွေ ဖြစ်တဲ့ jpg , png ဒါ့အပြင် pdf စတာတွေကိုလည်း link ပေးလို့ရပါတယ်။ ဥပမာ ကျွန်တော်တို့တွေ file download အတွက် zip file တစ်ခု ကို download ချခွင့်ပေးချင်တဲ့ အခါမှာ အဲဒီ zip file ရဲ့ လမ်းကြောင်းကို a tag မှာ href နဲ့ ချိတ်ပေးလိုက်ရင်ရပါပြီ။ အောက်က code လေးကို ကြည့်လိုက်ပါ။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Link</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>One Piece</h1>
  <h2>by Eiichiro Oda</h2>
  <ul class="menu">
    <li><a href="/home.html">Home</a></li>
    <li><a href="/anime.html">Anime</a></li>
    <li><a href="/download.html">Download</a></li>
  </ul>
  
  <p>One Piece is a Japanese <span class='highlight'>shounen manga series</span>
written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump
since August 4, 1997; the individual chapters are being published in tankōbon volumes by
Shueisha, with the first released on December 24, 1997, and the 70th volume released as
of March 2013. </p>

<div class="clear"></div>

  <p>One Piece follows the adventures of <a href="#luffy">Monkey D. Luffy</a>, a young
boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit,
and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in
search of the world's ultimate treasure known as One Piece in order to become the next
Pirate King.</p>

  <p><a href="/download/trailer.zip">Download</a> Trailer</a></p>

  <h3 id="luffy">Monkey D. Luffy</h3>
  <p>Monkey D. Luffy, [1] also known as "Straw Hat Luffy" and commonly as "Straw Hat", [4] is
a pirate and the main protagonist of the anime and manga, One Piece. He is the son of the
Revolutionary Army's commander, Monkey D. Dragon, [12] the grandson of the famed Marine,
Monkey D. Garp, [13] the foster son of a mountain bandit, Curly Dadan, and the adopted
brother of the late "Fire Fist" Portgas D. Ace and the revolutionary Sabo. His life long
dream is to become the Pirate King by finding the legendary treasure left behind by the
previous Pirate King, Gol D. Roger. [14] He believes that being Pirate King means one has
the most freedom in the world. He has eaten the Gomu Gomu no Mi. As the founder and
captain of the Straw Hat Pirates, he is the first member who makes up the crew, as well
as one of its top three fighters. </p>

</body>
</html>

```

အထက်ပါ code မှာ

```
<p><a href="./download/trailer.zip">Download</a> Trailer</a></p>
```

ဆိုတာ တွေ့ပါလိမ့်မယ်။ Download ဆိုတဲ့ link လေးကို နှိပ်လိုက်ရင် trailer.zip file လေး download ကျလာပါလိမ့်မယ်။

zip file အစား ကျွန်တော်တို့ တွေ pdf , mp3 စတဲ့ file link တွေကို ပေးထားနိုင်ပါတယ်။

Link to an E-Mail Address

ကျွန်တော်တို့တွေ အနေနဲ့ link ကို webpage , file တွေသာမက email နဲ့လည်း link ချိတ်နိုင်ပါတယ်။ email ပေးထားပြီး email ကို နှိပ်လိုက်တဲ့ အခါမှာ email ရေးဖို့ အတွက် outlook ဒါမှ မဟုတ် လက်ရှိ computer မှာ သုံးနေတဲ့ email client တက်လာပြီး စာပို့လို့ ရအောင် ဖန်တီးလို့ရပါတယ်။

```
<html>
<head>
  <title>Email</title>
</head>
<body>
  If you have a question , send email to <a
href="mailto:me@saturngod.net">me@saturngod.net</a>.
</body>
</html>
```

အထက်ပါ code ကို browser မှာဆိုရင်အောက်ကိုလို ပေါ်ပါမယ်။

If you have a question , send email to me@saturngod.net.

Figure 6-7.

email link ကို နှိပ်လိုက်ရင် သင့် email client နဲ့ တက်လာပါလိမ့်မယ်။

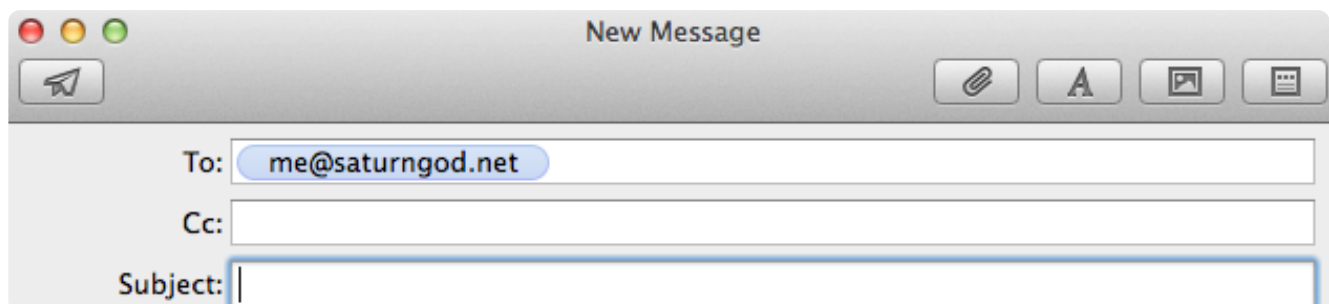


Figure 6-8.

Email link ကို ထည့်တဲ့ အခါမှာ မသက်ဆိုင်တဲ့ email တွေ spam တွေ ရောက်လာနိုင်ပါတယ်။ ဒါကိုတော့ သတိထားဖို့ လိုအပ်ပါတယ်။ ဒါကြောင့် ကိုယ့် ကိုယ်ပိုင် email link ကို မထည့်သင့်ပါဘူး။

Change Link Colors

Link ဟာ ပုံမှန် အပြာရောင် အနေနဲ့ ဖော်ပြထားပါတယ်။ link ကို သွားပြီးသွားရင် နီညိုရောင် ပြောင်းသွားပါတယ်။ ကျွန်တော်တို့ တွေ CSS ကို အသုံးပြုပြီးတော့ စိတ်ကြိုက် အရောင်ကို ပုံစံတို့ကို ပြောင်းလဲ နိုင်ပါတယ်။

```
<a href="/home.html">Home</a>
<a href="/anime.html">Anime</a>
<a href="/download.html">Download</a>
```

[Home](#) [Anime](#) [Download](#)

Figure 6-9.

ပုံမှန် link အရ ဆိုရင်တော့ အပြာရောင် ဖြစ်ပါတယ်။ အရောင်ပြောင်းဖို့အတွက် css မှာ

```
a{
  padding: 10px;
}
a:link{
  text-decoration:none;
  color: red;
}
a:visited {
  color:green;
}
a:active {
  color: purple;
}
```

[Home](#) [Anime](#) [Download](#)

Figure 6-10.

အဲဒါဆိုရင် link အရောင်ပြောင်းသွားတာကို တွေ့ရပါလိမ့်မယ်။

a:link ဆိုတာကတော့ normal link အတွက်ပါ။

a:visited ဆိုတာကတော့ visited link အတွက်ပါ။

a:active ဆိုတာကတော့ selected link အတွက်ပါ။

Change Link Hover Effects

ကျွန်တော်တို့တွေ အနေနဲ့ link အပေါ်မှာ mouse တင်လိုက်တာနဲ့ အရောင်ပြောင်းစေချင်ရင်တော့ `a:hover` ကို အသုံးပြုရပါတယ်။ ပြီးခဲ့တဲ့ code မှာ hover ထပ်ဖြည့်ပါမယ်။

```
a{
  padding: 10px;
}
a:link{
  text-decoration:none;
  color: red;
}
a:visited {
  color:green;
}
a:active {
  color: purple;
}
a:hover {
  text-decoration:underline;
}
```

အဲဒါဆိုရင် link ပေါ်မှာ mouse တင်လိုက်တာနဲ့ underline ဖြစ်နေတာကို တွေ့ရပါလိမ့်မယ်။

[Home](#) [Anime](#) [Download](#)

Figure 6-11.

Define Link Relationships

`<a>` tag link တွေ ကြားက ဆက်နွယ်မှုတွေကို `rel` attribute သုံးပြီးတော့ ဖော်ပြနိုင်ပါတယ်။ ဒီ link ဟာ author link လား prev , next စတဲ့ link လား။ help လား စသည်ဖြင့် သတ်မှတ်နိုင်ပါတယ်။ Search engine အနေနဲ့ outgoing link ကို analyze လုပ်တဲ့ အခါမှာ အမျိုးအစား ခွဲပြီး သတ်မှတ်ဖို့ အတွက် အသုံးဝင်ပါတယ်။

```
<html>
<head>
  <title>Link</title>
</head>
<body>
<p>This page authored by <a rel="author" href="http://www.saturngod.net">Saturngod</a>
</p>
<p>Test out the <a rel="external nofollow" href="http://beta.newdesign.com">beta version
of the site</a>
</body>
</html>
```

အဲဒီ code မှာ `Saturngod` ဆိုတဲ့ နေရာမှာ rel ကို author လို့ ပေးထားတာ တွေ့ပါလိမ့်မယ်။ link က author နဲ့ သက်ဆိုင်တယ်လို့ ဆိုထားပါတယ်။ နောက်ပြီးတော့ multiple relation အတွက် space ထားပြီး အသုံးပြုနိုင်ပါတယ်။ link relationship type တွေကိုတော့ အောက်က table မှာ ကြည့်နိုင်ပါတယ်။

Name	Description
alternate	ယခု စာမျက်နှာ ရဲ့ အခြား ပုံစံ တစ်ခု
author	ယခု စာမျက်နှာရဲ့ စာရေးသူ
external	site ရဲ့ အပြင်ဘက် link
first , prev , next , last	စာမျက်နှာများ အတွက် link
help	အကူအညီ စာမျက်နှာကို ချိတ်ဆက်ထားခြင်း
licence	copyright လိုင်စင်နှင့် သက်ဆိုင်သော
nofollow	search engine အနေနဲ့ လက်ရှိ link ဟာ လက်ရှိ document နှင့် သက်ဆိုင် မှု မရှိခြင်း။ ဒါကြောင့် page ranking အတွက် အသုံးမပြုရန်
noreferrer	browser အနေနဲ့ referrer information ကို click လုပ်လျှင် မပို့ရန်
search	search page ကို သွားမည့် link
tag	tag သို့မဟုတ် keyword ဖော်ပြထားသည့် link

Working with Tables

Data တွေကို rows တွေ columns တွေနဲ့ ဖော်ပြချင်သလား ? အခု အခန်းမှာတော့ HTML table ကို အသုံးပြုနည်း ဖော်ပြပေးပါမယ်။ Table ကို ဖန်တီးတဲ့ အခါမှာ `<table>` tag ကို အသုံးပြုရပါတယ်။ `<tr>` tag ကတော့ row ကို ဖန်တီးပါတယ်။ `<tr>` tag ကတော့ row ထဲက data cell တွေ ဖန်တီးဖို့ အတွက်ပါ။ Table မှာ color , background images စတာတွေ ထည့်သွင်းလို့ရပါတယ်။ Table ကို အသုံးပြုပြီးတော့ website ရဲ့ layout ကို လည်း ဖန်တီးအသုံးပြုနိုင်ပါတယ်။

Understanding Table Structure

HTML tables တွေဟာ data တွေ အများကြီးကို rows နဲ့ columns နဲ့ ပြတဲ့ အခါမှာ အသုံးပြုပါတယ်။ Table ထဲမှာ number တွေ သာမက စာတွေ ၊ ပုံတွေ ၊ multimedia တွေ ပါ ပါဝင်နိုင်ပါတယ်။ ဥပမာ ။။။ သင့် အနေနဲ့ table ကို company ရဲ့ အရောင်း ရလဒ် ကို ဖော်ပြဖို့အတွက် grid ထဲမှာ ပုံတွေ နောက်ပြီးတော့ data တွေ CSS ကို အသုံးပြုပြီးတော့ border , alignment , size စတာ တွေ နဲ့ ဖန်တီးနိုင်ပါတယ်။

Table Column 1	Table Column 2
Cell	Cell

Add a Table

Table ကို ထည့်ဖို့ အတွက် `<table>` tag ကို အရင် ထည့်ရပါတယ်။ ပြီးတဲ့ အခါမှာတော့ ကျွန်တော်တို့တွေ `<tr>` tag နဲ့ row ကို ဖန်တီးပါတယ်။ row ထဲမှာ cell အရေအတွက် အတွက် ကတော့ `<td>` ကို ဖန်တီးရပါတယ်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Table</title>
</head>
<body>
  <table>
    <!-- Row 1 -->
    <tr>
      <td>Cell 1</td>
      <td>Cell 2</td>
    </tr>
    <!-- Row 2 -->
    <tr>

```

```

        <td>Cell 1</td>
        <td>Cell 2</td>
    </tr>
</table>
</body>
</html>

```

အထက်ပါ code ကို browser မှာ ဆိုရင် အောက်ကလို မြင်ရပါလိမ့်မယ်။

Cell 1	Cell 2
Cell 1	Cell 2

Figure 7-1. Sample Table

ကျွန်တော်တို့ နောက်ထပ် table တစ်ခု ဖန်တီးကြည့်ရအောင် ။ လက်ရှိ table နေရာမှာ အောက်က table အစားထိုးလိုက်ပါမယ်။

```

<table>
  <tr>
    <th>Company</th>
    <th>Product</th>
  </tr>
  <tr>
    <td>Apple</td>
    <td>Tablet , Phone , Computer</td>
  </tr>
  <tr>
    <td>Google</td>
    <td>Search Engine , Chrome Browser , Android</td>
  </tr>
  <tr>
    <td>Microsoft</td>
    <td>Operating system, Software</td>
  </tr>
</table>

```

အဲဒါလေးကို browser မှာ ဆိုရင် ဒီလိုလေး မြင်ရပါမယ်။

Company	Product
Apple	Tablet , Phone , Computer
Google	Search Engine , Chrome Browser , Android
Microsoft	Operating system, Software

Figure 7-2. Table

`<th>` ဆိုတာကတော့ header အတွက် အသုံးပြုတာပါ။ Bold ဖြစ်ပြီးတော့ text ကို center ပို့ထားတာ ကိုတွေ့နိုင်ပါတယ်။

Add Table Borders

ပုံမှန် အားဖြင့် table မှာ ဘောင်တွေ မပါဝင်ပါဘူး။ table ကို ဘောင်တွေ ထည့်ဖို့ အတွက် `border` ကို အသုံးပြုရပါမယ်။ ဘောင်ထူထူလိုချင်ရင်တော့ `border-width` ရဲ့ တန်ဖိုးကို ပြောင်းပေးဖို့ လိုပါတယ်။ အရင်ဆုံး table ကို border ထည့်ကြည့်ရအောင်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Table</title>
  <style>
    table { border : 1px solid red;}
  </style>
</head>
<body>
<table>
  <tr>
    <th>Company</th>
    <th>Product</th>
  </tr>
  <tr>
    <td>Apple</td>
    <td>Tablet , Phone , Computer</td>
  </tr>
  <tr>
    <td>Google</td>
    <td>Search Engine , Chrome Browser , Android</td>
  </tr>
  <tr>
    <td>Microsoft</td>
    <td>Operating system, Software</td>
  </tr>
</table>
</body>
</html>
```

table ကို border အနီရောင် ထည့်ထားတဲ့ အတွက် table တစ်ခု တည်း ဘောင် အနီရောင်နဲ့ မြင်ရပါမယ်။

Company	Product
Apple	Tablet , Phone , Computer
Google	Search Engine , Chrome Browser , Android
Microsoft	Operating system, Software

Figure 7-3.

ထပ်ပြီးတော့ th နဲ့ td မှာ border ထည့်ကြည့်ရအောင်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Table</title>
  <style>
    table { border : 3px solid red;}
    th { border: 2px solid green;}
    td { border: 2px solid blue;}
  </style>
</head>
<body>
<table>
  <tr>
    <th>Company</th>
    <th>Product</th>
  </tr>
  <tr>
    <td>Apple</td>
    <td>Tablet , Phone , Computer</td>
  </tr>
  <tr>
    <td>Google</td>
    <td>Search Engine , Chrome Browser , Android</td>
  </tr>
  <tr>
    <td>Microsoft</td>
    <td>Operating system, Software</td>
  </tr>
</table>
</body>
</html>

```

Company	Product
Apple	Tablet , Phone , Computer
Google	Search Engine , Chrome Browser , Android
Microsoft	Operating system, Software

Figure 7-4

Adjust Cell Padding and Border Spacing

Cell တစ်ခု နဲ့ တစ်ခုကြားမှာ space လေးတွေ ကွာနေတာလေးတွေကို တွေ့မှာပါ။ အဲဒါက border-spacing ပါ။ cell က စာနဲ့ ဘောင် ကပ်နေတာကတော့ padding ကြောင့်ပါ။

ကျွန်တော်တို့တွေ td မှာ padding ထည့်ကြည့်ရအောင်။

```

<!DOCTYPE html>

```

```

<html>
<head>
  <meta charset="UTF-8">
  <title>Table</title>
  <style>
    table { border : 3px solid red;}
    th { border: 2px solid green;}
    td { border: 2px solid blue; padding:30px;}
  </style>
</head>
<body>
<table>
  <tr>
    <th>Company</th>
    <th>Product</th>
  </tr>
  <tr>
    <td>Apple</td>
    <td>Tablet , Phone , Computer</td>
  </tr>
  <tr>
    <td>Google</td>
    <td>Search Engine , Chrome Browser , Android</td>
  </tr>
  <tr>
    <td>Microsoft</td>
    <td>Operating system, Software</td>
  </tr>
</table>
</body>
</html>

```

td မှာ padding 30px ထည့်လိုက်တာကြောင့် cell အထဲက စာနဲ့ border 30px ကွာသွားတာကို တွေ့ရမှာပါ။

Company	Product
Apple	Tablet , Phone , Computer
Google	Search Engine , Chrome Browser , Android
Microsoft	Operating system, Software

Figure 7-5.

အခု table ရဲ့ border-spacing ကို ပြောင်းကြည့်ရအောင်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Table</title>
  <style>
    table { border : 3px solid red; border-spacing:20px;}
    th { border: 2px solid green;}
    td { border: 2px solid blue; padding:30px;}
  </style>
</head>
<body>
<table>
  <tr>
    <th>Company</th>
    <th>Product</th>
  </tr>
  <tr>
    <td>Apple</td>
    <td>Tablet , Phone , Computer</td>
  </tr>
  <tr>
    <td>Google</td>
    <td>Search Engine , Chrome Browser , Android</td>
  </tr>
  <tr>
    <td>Microsoft</td>
    <td>Operating system, Software</td>
  </tr>
</table>
</body>
</html>
```

table မှာ `border-spacing:20px;` ထည့်လိုက်ပါတယ်။ အဲဒါကြောင့် cell ရဲ့ spacing တွေ ကွာသွားတာကို တွေ့နိုင်ပါတယ်။

Company	Product
Apple	Tablet , Phone , Computer
Google	Search Engine , Chrome Browser , Android
Microsoft	Operating system, Software

Figure 7-6.

Adjust Cell Width and Height

Table မှာ cell ရဲ့ width နဲ့ height ကို လိုအပ်သလို ချိန်ညှိနိုင်ပါတယ်။ အဲဒီ အတွက် CSS က width နှင့် height ကို အသုံးပြုဖို့လိုပါတယ်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Table</title>
  <style>
    table { border : 3px solid red;}
    th { border: 2px solid green;height:100px;}
    td { border: 2px solid blue; width:100px;}
  </style>
</head>
<body>
<table>
  <tr>
    <th>Company</th>
    <th>Product</th>
  </tr>
  <tr>
    <td>Apple</td>
    <td>Tablet , Phone , Computer</td>
  </tr>
  <tr>

```

```

        <td>Google</td>
        <td>Search Engine , Chrome Browser , Android</td>
    </tr>
    <tr>
        <td>Microsoft</td>
        <td>Operating system, Software</td>
    </tr>
</table>
</body>
</html>

```

အဲဒီ code ဟာ browser မှာ ဆိုရင် အောက်က ပုံလို မြင်ရပါမယ်။

Company	Product
Apple	Tablet , Phone , Computer
Google	Search Engine , Chrome Browser , Android
Microsoft	Operating system, Software

Figure 7-7.

Cell ရဲ့ width ဟာ 100px ဖြစ်နေတာကို တွေ့ပါလိမ့်မယ်။ နောက်ပြီးတော့ th ရဲ့ height ဟာ 100px ဖြစ်တာကြောင့် ကြီးနေတာကို တွေ့ရပါလိမ့်မယ်။

အကယ်၍ ကျွန်တော်တို့တွေက cell တွေ အကုန် မပြင်ချင်ပဲ အချို့ကိုသာ ပြင်လိုရင်တော့ class သို့မဟုတ် id နဲ့ တွဲ သုံးဖို့လိုပါတယ်။

```
td.custom { height:80px }
```

html က td မှာ

```
<td class="custom">My Data</td>
```

ဆိုပြီး အသုံးပြုနိုင်ပါတယ်။

Add a Caption to a Table

Table မှာ caption ထည့်ဖို့ အတွက် `<caption>` tag ကို `<table>` tag အတွင်းထဲမှာ ထည့်ပြီးတော့ အသုံးပြုနိုင်ပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Table</title>
  <style>
    table { border : 3px solid red;}
    th { border: 2px solid green;height:100px;}
    td { border: 2px solid blue; width:100px;}
  </style>
</head>
<body>
<table>
  <caption>Technology Companies</caption>
  <tr>
    <th>Company</th>
    <th>Product</th>
  </tr>
  <tr>
    <td>Apple</td>
    <td>Tablet , Phone , Computer</td>
  </tr>
  <tr>
    <td>Google</td>
    <td>Search Engine , Chrome Browser , Android</td>
  </tr>
  <tr>
    <td>Microsoft</td>
    <td>Operating system, Software</td>
  </tr>
</table>
</body>
</html>
```

Browser မှာ ဆိုရင် အောက်ပါ ပုံအတိုင်း မြင်ရပါလိမ့်မယ်။

Technology Companies

Company	Product
Apple	Tablet , Phone , Computer
Google	Search Engine , Chrome Browser , Android
Microsoft	Operating system, Software

Figure 7-8.

Add a Background Color to Cells

ကျွန်တော်တို့ cell ရဲ့ background color ကို ထည့်မယ်ဆိုရင်တော့ အရင်တုန်းက သုံးခဲ့တဲ့ `background-color` ကိုပဲ ပြန်ပြီး အသုံးပြုရပါမယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Table</title>
  <style>
    table {border : 3px solid red;}
    th {border: 2px solid green;height:100px;}
    td {border: 2px solid blue; width:100px;}
    td.highlight {background-color :#FFBCC;}
  </style>
</head>
<body>
<table>
  <caption>Technology Companies</caption>
  <tr>
    <th>Company</th>
    <th>Product</th>
  </tr>
  <tr>
    <td class='highlight'>Apple</td>
    <td>Tablet , Phone , Computer</td>
  </tr>
  <tr>
    <td>Google</td>
    <td>Search Engine , Chrome Browser , Android</td>
  </tr>
  <tr>
    <td>Microsoft</td>
    <td>Operating system, Software</td>
  </tr>
</table>
```

```

        <td>Microsoft</td>
        <td>Operating system, Software</td>
    </tr>
</table>
</body>
</html>

```

အဲဒီ code မှာ Apple ရဲ့ td ကို highlight class သုံးထားပြီးတော့ css မှာ

```
td.highlight {background-color :#FFBCC;}
```

လို့ ရေးထားတာ တွေ့နိုင်ပါတယ်။

Technology Companies

Company	Product
Apple	Tablet , Phone , Computer
Google	Search Engine , Chrome Browser , Android
Microsoft	Operating system, Software

Figure 7-9.

Add a Background Color to a Table

Table တစ်ခု လုံးကို background color ထည့်လိုရင်တော့ CSS မှာ table အတွက် `background-color` ကို အသုံးပြုရပါမယ်။

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Table</title>
    <style>
        table.pretty {border : 1px solid red;background-color :#FFBCC; border-
spacing:0px;}
        th {border: 1px solid gray;}
    </style>

```

```
  |
```

အဲဒီ code မှာ table ကို class pretty ပေးထားပြီးတော့

```
table.pretty {border : 1px solid red;background-color :#FFFBC; border-spacing:0px;}
```

လို့ ရေးထားပြီး background color ထည့်ထားတာ တွေ့နိုင်ပါတယ်။ css ကို class မသုံးပဲ table သုံး လိုက်ရင် table တွေ အားလုံးမှာ သွားပြီး အကျိုးသက်ရောက်ပါလိမ့်မယ်။ အဲဒီ code လေးက browser မှာ ဆိုရင်အောက်ကလို ပေါ်ပါလိမ့်မယ်။

Company	Product
Apple	Tablet , Phone , Computer
Google	Search Engine , Chrome Browser , Android
Microsoft	Operating system, Software

Figure 7-10.

Adjust the Table Size

သင့်အနေနဲ့ table အရွယ်အစားကို စိတ်ကြိုက်ပြင်လို့ရပါတယ်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Table</title>
  <style>
    table.pretty {border : 1px solid red;background-color :#FFBCC; border-
spacing:0px;width:100%;}
    th {border: 1px solid gray;}
    th.small {width:200px;}
    td {border: 1px solid gray;}
  </style>
</head>
<body>
<table class="pretty">
  <caption>Technology Companies</caption>
  <tr>
    <th class='small'>Company</th>
    <th>Product</th>
  </tr>
  <tr>
    <td>Apple</td>
    <td>Tablet , Phone , Computer</td>
  </tr>
  <tr>
    <td>Google</td>
    <td>Search Engine , Chrome Browser , Android</td>
  </tr>
  <tr>
    <td>Microsoft</td>
    <td>Operating system, Software</td>
  </tr>
</table>
</body>
</html>

```

ကျွန်တော်တို့ table ရဲ့ width ကို 100% ပေးထားလိုက်ပါပြီ။ ဒါကြောင့် table ရဲ့ အကျယ် ဟာ browser အပြည့် မြင်ရပါမယ်။

Technology Companies	
Company	Product
Apple	Tablet , Phone , Computer
Google	Search Engine , Chrome Browser , Android
Microsoft	Operating system, Software

Figure 7-11

အခု လက်ရှိ table မှာပဲ ကျွန်တော်တို့ ထပ်ပြီးတော့ height ကို 500px ထည့်ကြည့်ရအောင်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Table</title>
  <style>

```

```

table.pretty {
    border : 1px solid red;
    background-color :#FFFBC;
    border-spacing:0px;width:100%;
    height:500px;
}
th {border: 1px solid gray;}
th.small {width:200px;}
td {border: 1px solid gray;}
</style>
</head>
<body>
<table class="pretty">
  <caption>Technology Companies</caption>
  <tr>
    <th class='small'>Company</th>
    <th>Product</th>
  </tr>
  <tr>
    <td>Apple</td>
    <td>Tablet , Phone , Computer</td>
  </tr>
  <tr>
    <td>Google</td>
    <td>Search Engine , Chrome Browser , Android</td>
  </tr>
  <tr>
    <td>Microsoft</td>
    <td>Operating system, Software</td>
  </tr>
</table>
</body>
</html>

```

အဲဒီ အခါ table ရဲ့ အရွယ်အစားကြီးသွားပြီး cell height တွေပါ ကျယ်သွားတာကို တွေ့နိုင်ပါတယ်။

Technology Companies	
Company	Product
Apple	Tablet , Phone , Computer
Google	Search Engine , Chrome Browser , Android
Microsoft	Operating system, Software

Figure 7-12

Change Table Alignment

ကျွန်တော်တို့တွေ table ကို စာတွေ နဲ့ တွဲပြီး ဖော်ပြတဲ့ အခါမှာတော့ Table ကို align ပြန်လုပ်ဖို့ လိုပါတယ်။

ပထမဆုံး table ကို ညာဘက် ကပ်ကြည့်ရအောင်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Table</title>
  <style>
    table.pretty {
      border : 1px solid red;
      background-color :#FFFBCC;
      border-spacing:0px;width:300px;
      float:right;
    }
    th {border: 1px solid gray; text-align:right;}
  </style>
</head>
<table border="1" class="pretty">
  <thead>
    <tr>
      <th colspan="2">Technology Companies</th>
    </tr>
    <tr>
      <th>Company</th>
      <th>Product</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Apple</td>
      <td>Tablet , Phone , Computer</td>
    </tr>
    <tr>
      <td>Google</td>
      <td>Search Engine , Chrome Browser , Android</td>
    </tr>
    <tr>
      <td>Microsoft</td>
      <td>Operating system, Software</td>
    </tr>
  </tbody>
</table>
</html>
```

```

    th.small {width:200px;}
    td {border: 1px solid gray;}
    td.align-bottom {vertical-align:bottom;}
  </style>
</head>
<body>
<table class="pretty">
  <caption>Technology Companies</caption>
  <tr>
    <th class='small'>Company</th>
    <th>Product</th>
  </tr>
  <tr>
    <td>Apple</td>
    <td>Tablet , Phone , Computer</td>
  </tr>
  <tr>
    <td>Google</td>
    <td>Search Engine , Chrome Browser , Android</td>
  </tr>
  <tr>
    <td class="align-bottom">Microsoft</td>
    <td>Operating system, Software</td>
  </tr>
</table>
  <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.</p>
</body>
</html>

```

အဲဒီ code မှာ table ကို ညာဘက် ကပ်ထားတာ တွေ့ရပါလိမ့်မယ်။

```

table.pretty {
  float:right;
}

```

ဆိုတဲ့ သဘောက table တစ်ခုလုံးကို ညာ ဘက် ရွှေ့လိုက်တဲ့ သဘောပါ။

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Company	Product
Apple	Tablet , Phone , Computer
Google	Search Engine , Chrome Browser , Android
Microsoft	Operating system, Software

Figure 7-14.

တကယ်လို့ website ရဲ့ အလယ်မှာ ထားစေချင်ရင်တော့

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Table</title>
  <style>
    table.pretty {
      border : 1px solid red;
      background-color :#FFFBCC;
      border-spacing:0px;width:300px;
      margin-left:auto;
      margin-right:auto;
    }
    th {border: 1px solid gray; text-align:right;}
    th.small {width:200px;}
    td {border: 1px solid gray;}
    td.align-bottom {vertical-align:bottom;}
  </style>
</head>
<body>
<table class="pretty">
  <caption>Technology Companies</caption>
  <tr>
    <th class='small'>Company</th>
    <th>Product</th>
  </tr>
  <tr>
    <td>Apple</td>
    <td>Tablet , Phone , Computer</td>
  </tr>
  <tr>
    <td>Google</td>
    <td>Search Engine , Chrome Browser , Android</td>
  </tr>
  <tr>
    <td class="align-bottom">Microsoft</td>
  </tr>
</table>

```

```

        <td>Operating system, Software</td>
    </tr>
</table>
    <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.</p>
</body>
</html>

```

အဲဒီ code မှာ `float:right` မသုံးတော့ပဲ `margin` ပြောင်းလိုက်ပါတယ်။

```

margin-left:auto;
margin-right:auto;

```

`margin-left` ကော `margin-right` ကော `auto` ပေးလိုက်တာကြောင့် `table` က အလယ်ကို ရွှေ့သွားတာ တွေ့နိုင်ပါတယ်။

Technology Companies	
Company	Product
Apple	Tablet , Phone , Computer
Google	Search Engine , Chrome Browser , Android
Microsoft	Operating system, Software

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Figure 7-15.

Extend Cells Across Columns and Rows

တစ်ခါတစ်လေ ကျွန်တော်တို့တွေ row ၂ ခု ဒါမှ မဟုတ် columns ၂ ခု စတာတွေကို ပြောင်းဖို့ လိုတာတွေ ရှိပါတယ်။ အဲဒီ အခါ colspan နဲ့ rowspan attribute ကို <td> tags နဲ့ တွဲ သုံးဖို့ လိုပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Table</title>
  <style>
    table.pretty {
      border : 1px solid red;
      background-color :#FFFBCC;
      border-spacing:0px;width:300px;
      margin-left:auto;
      margin-right:auto;
    }
    th {border: 1px solid gray;}
    th.small {width:200px;}
    td {border: 1px solid gray;}
    td.align-bottom {vertical-align:bottom;}
  </style>
</head>
<body>
<table class="pretty">
  <caption>Technology Companies</caption>
  <tr>
    <th colspan="2">Sample</th>
  </tr>
  <tr>
    <th class='small'>Company</th>
    <th>Product</th>
  </tr>
  <tr>
    <td>Apple</td>
    <td>Tablet , Phone , Computer</td>
  </tr>
  <tr>
    <td>Google</td>
    <td>Search Engine , Chrome Browser , Android</td>
  </tr>
  <tr>
    <td class="align-bottom">Microsoft</td>
    <td>Operating system, Software</td>
  </tr>
</table>
</body>
</html>
```

အဲဒီ code မှာ

```
<tr>
  <th colspan="2">Sample</th>
</tr>
```

ဆိုတာ columns ၂ ခု ကို ပြောင်းမယ်လို့ ဆိုထားတာပါ။ ကျွန်တော်တို့ ဆီမှာ columns ၂ ခု ရှိပါတယ်။ Company နဲ့ Product ပါ။ columns ၂ ခုကို ပြောင်းချင်တာကြောင့် colspan မှာ 2 ထည့်ထားတာပါ။ တကယ်လို့ columns ၃ ခု သာ ရှိရင်တော့ colspan = "3" ကို အသုံးပြုနိုင်ပါတယ်။

Technology Companies	
Sample	
Company	Product
Apple	Tablet , Phone , Computer
Google	Search Engine , Chrome Browser , Android
Microsoft	Operating system, Software

Figure 7-16.

```

<html>
<head>
  <title>Table</title>
  <style>
    table.pretty {
      border : 1px solid red;
      background-color :#FFFBCC;
      border-spacing:0px;width:300px;
      margin-left:auto;
      margin-right:auto;
    }
    th {border: 1px solid gray;}
    th.small {width:200px;}
    td {border: 1px solid gray;}
    td.align-bottom {vertical-align:bottom;}
  </style>
</head>
<body>
<table class="pretty">
  <caption>Technology Companies</caption>
  <tr>
    <th colspan="2">Sample</th>
  </tr>
  <tr>
    <th class='small'>Company</th>
    <th>Product</th>
  </tr>

```

```

<tr>
  <td rowspan="3">Apple</td>
  <td>Tablet , Phone , Computer</td>
</tr>
<tr>
  <td>Browser</td>
</tr>
<tr>
  <td>Operating system, Software</td>
</tr>
<tr>
  <td>Google</td>
  <td>Search Engine , Chrome Browser , Android</td>
</tr>
<tr>
  <td class="align-bottom">Microsoft</td>
  <td>Operating system, Software</td>
</tr>
</table>
</body>
</html>

```

အဲဒီ code မှာ ဆိုရင်တော့ Apple မှာ rowspan ကို ၃ လို့ သုံးထားပါတယ်။ row span သုံးထားတဲ့ အတွက် သူ့အောက်က Browser နဲ့ Operating system, Software တို့မှာ <td> တစ်ခု ပဲ ရှိတာကို တွေ့ပါလိမ့်မယ်။

Technology Companies	
Sample	
Company	Product
Apple	Tablet , Phone , Computer
	Browser
	Operating system, Software
Google	Search Engine , Chrome Browser , Android
Microsoft	Operating system, Software

Figure 7-17.

Historical: Table-Based Page Layout

ကျွန်တော်တို့ အခု website တစ်ခုလုံးကို table ကို အသုံးပြုပြီး ရေးဆွဲကြည့်ရအောင်။ ဒီနည်းလမ်းဟာ HTML အဟောင်းတွေမှာ အသုံးများခဲ့တဲ့ historical technique ဖြစ်ပါတယ်။ Modern HTML/CSS မှာတော့ page layout အတွက် table မသုံးသင့်တော့ပါဘူး။ Tables ကို tabular data အတွက်ပဲ သုံးပြီး page layout အတွက် CSS Flexbox, CSS Grid, semantic elements (`<header>` , `<nav>` , `<main>` , `<aside>` , `<footer>`) တို့ကို သုံးတာ ပိုမှန်ပါတယ်။

အရင်ဆုံး ကျွန်တော်တို့ ရဲ့ page layout ကို အကြမ်းဆွဲပါမယ်။

```
<html>
<head>
  <title>One Piece</title>
</head>
<body>
<table>
  <tr>
    <!-- Banner -->
    <td colspan=2>
      <!-- image here -->
    </td>
  </tr>
  <tr>
    <td>
      <!-- menu -->
    </td>
    <td>
      <!-- text -->
    </td>
  </tr>
</table>
</body>
</html>
```

အပေါ်မှာ banner ပါပြီးတော့ ဘယ်ဘက်မှာ menu ပါပါမယ်။ ညာဘက်မှာတော့ စာတွေ ရေးပါမယ်။

banner က အပြည့်ဖြစ်မှာ ဖြစ်တဲ့ အတွက်ကြောင့် column ၂ ခု ကို ပေါင်းထားပါတယ်။ အခု စာတွေ ပုံတွေ ထည့်ကြည့်ရအောင်။

```
<html>
<head>
  <title>One Piece</title>
  <meta charset="utf-8">
</head>
<body>
<table>
  <tr>
    <!-- Banner -->
    <td colspan=2>
      
    </td>
  </tr>
</table>
```

```

</tr>
<tr>
  <td>
    <ul>
      <li><a href="index.html">Home</a></li>
      <li><a href="gallery.html">Gallery</a></li>
      <li><a href="videos.html">Videos</a></li>
    </ul>
  </td>
  <td>
    <p>One Piece is a Japanese manga series written and illustrated
    by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the
    individual chapters are being published in tankōbon volumes by Shueisha, with the first
    released on December 24, 1997, and the 73rd volume released as of March 2014. One Piece
    follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of
    rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named
    the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate
    treasure known as One Piece in order to become the next Pirate King.</p>

    <p>The chapters have been adapted into an original video
    animation (OVA) produced by Production I.G in 1998, and an anime series produced by Toei
    Animation, which began broadcasting in Japan in 1999. Since then, the still ongoing
    series has aired over 600 episodes. Additionally, Toei has developed eleven animated
    feature films, two OVA's, and five television specials. Several companies have developed
    various types of merchandising such as a trading card game, and a large number of video
    games.</p>

    <p>The manga series was licensed for an English language release
    in North America by Viz Media, in the United Kingdom by Gollancz Manga, and in Australia
    and New Zealand by Madman Entertainment. The anime series has been licensed by Funimation
    Entertainment for an English-language release in North America, although the series was
    originally licensed and distributed by 4Kids Entertainment.</p>

    <p>One Piece has received wide critical acclaim, primarily for
    its art, characterization, humor and story. Several volumes of the manga have broken
    publishing records, including highest initial print run of any book in Japan and the
    first book to sell over three million copies in Oricon history. As of 2013, the series
    had over 345 million volumes in circulation worldwide, making it the best-selling manga
    series in history.</p>
  </td>
</tr>
</table>
</body>
</html>

```

အဲဒီ အခါ browser မှာ အောက်ကလို မျိုး မြင်ရပါမယ်။



One Piece (ワンピース Wan Pisu?) is a Japanese manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 73rd volume released as of March 2014. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

The chapters have been adapted into an original video animation (OVA) produced by Production I.G in 1998, and an anime series produced by Toei Animation, which began broadcasting in Japan in 1999. Since then, the still ongoing series has aired over 600

- [Home](#) episodes. Additionally, Toei has developed eleven animated feature films, two OVA's, and five television specials. Several companies
- [Videos](#) have developed various types of merchandising such as a trading card game, and a large number of video games.

The manga series was licensed for an English language release in North America by Viz Media, in the United Kingdom by Gollancz Manga, and in Australia and New Zealand by Madman Entertainment. The anime series has been licensed by Funimation Entertainment for an English-language release in North America, although the series was originally licensed and distributed by 4Kids Entertainment.

One Piece has received wide critical acclaim, primarily for its art, characterization, humor and story. Several volumes of the manga have broken publishing records, including highest initial print run of any book in Japan and the first book to sell over three million copies in Oricon history. As of 2013, the series had over 345 million volumes in circulation worldwide, making it the best-selling manga series in history.

အခု ထပ်ပြီးတော့ ကျွန်တော်တို့ တွေ css ထည့်ရအောင်။

```
<html>
<head>
  <title>One Piece</title>
  <meta charset="utf-8">
  <style>
    body {
      padding:0px;
      margin:0px;
      background-color:#ecf0f1;
    }
    table.container {
      width:776px;
      margin-left:auto;
      margin-right:auto;
      background-color:#FFFFFF;
    }
    td.banner {
      text-align:center;
    }
    td.menu {
      width: 100px;
      vertical-align:top;
      border-right: 1px solid #2980b9;
    }
    td.menu ul {
      list-style:none;
      padding:0px;
    }
    td.menu ul li {
```

```

        padding-bottom:5px;
    }

    td.menu ul li a {
        text-decoration: none;
    }

    td.menu ul li a:link , td.menu ul li a:visited {
        color:#2980b9;
    }

    td.menu ul li a:hover {
        color:#3498db;
        text-decoration:underline;
    }

    td.content {
        padding-left : 8px;
    }
</style>
</head>
<body>
<table class="container">
  <tr>
    <!-- Banner -->
    <td colspan=2 class='banner'>
      
    </td>
  </tr>
  <tr>
    <td class="menu">
      <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="gallery.html">Gallery</a></li>
        <li><a href="videos.html">Videos</a></li>
      </ul>
    </td>
    <td class="content">
      <p>One Piece is a Japanese manga series written and illustrated
by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the
individual chapters are being published in tankōbon volumes by Shueisha, with the first
released on December 24, 1997, and the 73rd volume released as of March 2014. One Piece
follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of
rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named
the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate
treasure known as One Piece in order to become the next Pirate King.</p>

      <p>The chapters have been adapted into an original video
animation (OVA) produced by Production I.G in 1998, and an anime series produced by Toei
Animation, which began broadcasting in Japan in 1999. Since then, the still ongoing
series has aired over 600 episodes. Additionally, Toei has developed eleven animated
feature films, two OVA's, and five television specials. Several companies have developed
various types of merchandising such as a trading card game, and a large number of video
games.</p>

      <p>The manga series was licensed for an English language release
in North America by Viz Media, in the United Kingdom by Gollancz Manga, and in Australia
and New Zealand by Madman Entertainment. The anime series has been licensed by Funimation
Entertainment for an English-language release in North America, although the series was
originally licensed and distributed by 4Kids Entertainment.</p>

      <p>One Piece has received wide critical acclaim, primarily for
its art, characterization, humor and story. Several volumes of the manga have broken

```

publishing records, including highest initial print run of any book in Japan and the first book to sell over three million copies in Oricon history. As of 2013, the series had over 345 million volumes in circulation worldwide, making it the best-selling manga series in history.

```

                </td>
            </tr>
        </table>
    </body>
</html>

```

အခု ဆိုရင် ကျွန်တော်တို့ တွေ CSS တွေ ထပ်ဖြည့်ထားပါတယ်။ ဒါကြောင့် website က အောက် ကလို မြင်ရပါလိမ့်မယ်။ သို့ပေမယ့် production website အသစ်ရေးမယ်ဆိုရင် ဒီ table layout နည်းကို မသုံးဘဲ CSS Grid/Flexbox နဲ့ semantic HTML ကို သုံးပါ။



[Home](#)
[Gallery](#)
[Videos](#)

One Piece (ワンピース Wan Pisu?) is a Japanese manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 73rd volume released as of March 2014. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

The chapters have been adapted into an original video animation (OVA) produced by Production I.G in 1998, and an anime series produced by Toei Animation, which began broadcasting in Japan in 1999. Since then, the still ongoing series has aired over 600 episodes. Additionally, Toei has developed eleven animated feature films, two OVA's, and five television specials. Several companies have developed various types of merchandising such as a trading card game, and a large number of video games.

The manga series was licensed for an English language release in North America by Viz Media, in the United Kingdom by Gollancz Manga, and in Australia and New Zealand by Madman Entertainment. The anime series has been licensed by Funimation Entertainment for an English-language release in North America, although the series was originally licensed and distributed by 4Kids Entertainment.

One Piece has received wide critical acclaim, primarily for its art, characterization, humor and story. Several volumes of the manga have broken publishing records, including highest initial print run of any book in Japan and the first book to sell over three million copies in Oricon history. As of 2013, the series had over 345 million volumes in circulation worldwide, making it the best-selling manga series in history.

Creating Forms

Website မှာ contact us , comments စတာတွေ ပေးတာတွေ လုပ်ချင်သလား ။ အခု အခန်းမှာ တော့ form တွေ ဘယ်လို တည်ဆောက်ရမလဲ ၊ ဘယ်လို အသုံးပြုရမလဲ ဆိုတာ ကို ပြောပြပေး သွားမှာပါ။

Understanding Forms

ကျွန်တော်တို့တွေဟာ form ကို information တွေ စုဆောင်းဖို့ အတွက် အသုံးပြုပါတယ်။ ဥပမာ။။ ကျွန်တော်တို့ရဲ့ website ကို လာရောက်သူတွေက ကျွန်တော် တို့ကို ဆက်သွယ်ချင်ရင် ဒါမှမဟုတ် အကြံဉာဏ်တွေ ပေးစေချင်ရင် form မှာ data တွေ ဖြည့်ပြီးတော့ submit လေးကို နှိပ် ပြီး ကျွန်တော် တို့ ဆီ ပေးပို့ နိုင်ပါတယ်။ Data တွေဟာ ကျွန်တော်တို့ front end ပိုင်း HTML ကနေ backend ပိုင်းကို ရောက်သွားပါတယ်။ backend ပိုင်းက data တွေကို collect လုပ်ပြီး database ထဲမှာ ထည့်သိမ်းတာ ဒါမှမဟုတ် email ပို့တာ စသည်ဖြစ် လိုအပ်တာတွေ လုပ်ပါ တယ်။

အခု modern HTML စာအုပ်မှာတော့ backend ပိုင်း အလုပ်လုပ်တာ မပါပါဘူး။ အဓိက HTML နဲ့ Form ဘယ်လို တည်ဆောက်ရမလဲ ဆိုတာကိုပဲ ရေးသားသွား မှာပါ။

Backend

Backend ပိုင်းကို [PHP](#)၊ [ASP.NET](#) စတဲ့ server-side language တစ်ခုခု နဲ့ ရေးသားရပါတယ်။ အဲဒီ အပိုင်းဟာ Programming language နဲ့ သက်ဆိုင်ပါတယ်။ HTML, CSS, JavaScript ကို သဘောပေါက်ပြီး server side ပိုင်းတွေ ရေးမယ်ဆိုရင်တော့ Server Side language တစ်ခုခု ကို ရွေးချယ်ပြီး လေ့လာဖို့ လိုပါတယ်။ Data တွေကို သိမ်းဖို့ အတွက် [MySQL](#) , [Microsoft SQL Server](#) စတဲ့ database server တွေ မှာ သိမ်းဆည်းထားပါတယ်။ ဒီအပိုင်းတွေဟာ server side language တွေကို လေ့လာတဲ့ အခါမှာ သိလာပါလိမ့်မယ်။

ဒီ စာအုပ်မှာတော့ အခြေခံဖြစ်တဲ့ HTML ကို သာ အဓိကထားပြီးရေးသားမှာ ဖြစ်တဲ့ အတွက် ကြောင့် Form ဟာ UI Design ပုံစံသာ ဖြစ်နေပါလိမ့်မယ်။ လက်တွေ့ အလုပ်ဖြစ်ဖို့ အတွက် အထက်က ပြောခဲ့သလို Server Side နဲ့ ပေါင်းပြီး ရေးသား ဖို့လိုပါတယ်။ Server Side နဲ့ HTML form ပေါင်းမှသာ comment system , Feedback Form , File upload စတာတွေဟာ အသက်ဝင် လာပါလိမ့်မယ်။

Types of Form Elements

Forms ဖန်တီးတော့မယ်ဆိုရင် form ထဲမှာ ဘယ် လို data တွေ ထည့်လို့ရလဲ ဆိုတာ သိဖို့လိုပါတယ်။ Form ထဲမှာ text box တွေပဲ ဖြစ်နိုင်သလို radio buttons , checkbox စတာတွေပါတာလည်း ဖြစ်နိုင်ပါတယ်။ Form တစ်ခုမှာ ပုံမှန် အားဖြင့် submit button ပါပါတယ်။ submit button ကတော့ form က data တွေကို web server ဆီ ပို့ပေးတဲ့ အပိုင်း ဖြစ်ပါတယ်။

Text Boxes

Textbox ကတော့ user တွေ data ရိုက်ထည့် ဖို့ အတွက် အသုံးပြုပါတယ်။ textbox data တွေဟာ limited character တွေ ဖြစ်နိုင်သလို phone number , postal code စတာတွေ လည်း ဖြစ်နိုင်ပါတယ်။ Modern HTML မှာ text content တွေကို email address သို့မဟုတ် URL အဖြစ် သတ်မှတ်လို့ရပါတယ်။

Radio Buttons

Radio button တွေ အပိုင်းလေးတွေ ကို form တွေမှာ မြင်ဖူးပါလိမ့်မယ်။ ရွေးချယ်စရာတွေ အများကြီးထဲမှာ တစ်ခုကို ရွေးချယ်စေလိုရင် radio button ကို အသုံးပြုပါတယ်။

Check Boxes

Check boxes က ရွေးချယ်စရာတွေ တစ်ခု ထက် မက ပေးထားတဲ့ အထဲမှာ တစ်ခု သို့မဟုတ် တစ်ခု ထက်ပို ရွေးချယ်ဖို့ အတွက် အသုံးပြုပါတယ်။

Menu

Menu ကတော့ တစ်ခုခု ကို ရွေးချယ်ဖို့ အတွက်ပဲ အသုံးပြုတာပါ။ Drop down menu နဲ့ ဖြစ်ပြီး နှိပ်လိုက်တဲ့ အခါမှသာ list ကျလာပြီး တစ်ခုကို ရွေးချယ်နိုင်ပါတယ်။

File Upload

File တွေ image တွေ upload လုပ်ဖို့ အတွက် File type ကို အသုံးပြုရပါတယ်။ File ကို အသုံးပြုထားရင် browser မှာ file ရွေးဖို့ ပေါ်လာပါတယ်။

Range Slider

Modern HTML မှာ range slider ပါလာပါတယ်။ Range Slider ကတော့ ပေးထားတဲ့ ကြားထဲက value ကို ရွေးချယ်ဖို့ အတွက် အသုံးပြုပါတယ်။

Date and Time Fields

Modern HTML မှာ နေ့စွဲတွေ အချိန်တွေကို အသုံးပြုဖို့အတွက် date/time input type တွေ ပါလာပါတယ်။ Browser အများစုမှာ date picker သို့မဟုတ် time picker UI ကို ပြပေးနိုင်ပြီး browser ပေါ်မူတည်ပြီး UI ပုံစံ ကွာနိုင်ပါတယ်။

Submit Button

Submit Button က နှိပ်လိုက်တဲ့ အခါမှာ form က data တွေကို စုဆောင်းပြီးတော့ web server ဆီ ပို့ပေးပါတယ်။

Reset Button

Form မှာ data တွေ ထည့်ထားပြီးတော့ အကုန်လုံးကို ပြန်ရှင်းလိုရင်တော့ reset Button လေးကို နှိပ်လိုက်ရင်ရပါတယ်။ ပုံမှန် အားဖြင့် Submit Button ဘေးမှာ ထား၍ပီး အသုံးပြုတတ်ကြပါတယ်။

Text Box

Radio Buttons

- Option 1
 Option 2
 Option 3

Check Box

- Data 1
 Data 2
 Data 3

Drop Down

Data 1 ▾

File Upload No file chosen

Range Slider

Date

Figure 8-1

Create a Form

Form တစ်ခု ဖန်တီးဖို့ အတွက် `<form>` tag နဲ့ အသုံးပြုပါတယ်။ `action` attribute ကတော့ data တွေကို ပို့ပေးဖို့ server address ဖြစ်ပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Simple Form</title>
</head>
<body>
  <form method="post" action="./post.php">
  </form>
</html>
```

အထက်ပါ code မှာ form ရဲ့ method က post သုံးမယ်လို့ ဆိုထားပါတယ်။ action ကတော့ post.php ကို form က data တွေ ပို့မယ်လို့ ဆိုထားတာပါ။

Form မှာ method ၂ မျိုးရှိပါတယ်။ post method နဲ့ get method ပါ။ POST က data တွေကို submit လုပ်တဲ့ အခါမှာ အသုံးပြုပြီးတော့ GET method ကတော့ data တွေကို ရှာဖွေ တဲ့ အခါမှာ အသုံးပြုပါတယ်။

GET method နဲ့ပို့လိုက်ရင်

```
server.php?name=value1&name2=valu2
```

ဆိုပြီး query string နဲ့ data တွေကို ပို့ပါတယ်။

POST method ဆိုရင်တော့ HTTP message body ကတော့ data တွေကို ပို့ပါတယ်။

```
POST server.php
Host : example.com
name=value1&name2=valu2
```

Send Form Data to an E-Mail Address

တကယ်လို့ form data ကို backend server ကို မပို့ချင်ပဲ အသုံးပြုသူကို သူ့ရဲ့ email အနေနဲ့ ပို့စေချင်ရင်တော့ action မှာ ပြောင်းပေးဖို့လိုပါတယ်။ data တွေကတော့ form မှာ ရှိတဲ့ data name နဲ့ value တွဲပြီး ဖော်ပြပါလိမ့်မယ်။ ဥပမာ။။။ `month=June` ။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Simple Form</title>
```

```

</head>
<body>
  <form method="post" enctype="text/plain" action="mailto:saturngod@example.com">
    <input type="text" name="message"/>
    <input type="submit">
  </form>
</html>

```

အဲဒီမှာ action ကို `mailto:saturngod@example.com` ဆိုပြီး ရေးထားတာတွေ့နိုင်ပါတယ်။ email ကို ပို့စေချင်ရင်တော့ `mailto:` ကို အရှေ့မှာ ထားပြီး နောက်မှာ ကိုယ့်ရဲ့ email ကို ထည့်ရပါတယ်။ `enctype="text/plain"` ကိုလည်း ထည့်ဖို့လိုပါတယ်။

ဒီနည်းလမ်းက user ရဲ့ device ထဲမှာ email app ပြင်ဆင်ထားမှ အလုပ်လုပ်နိုင်ပါတယ်။ Contact form ကို production website မှာ သုံးမယ်ဆိုရင် backend server သို့မဟုတ် form handling service ကို submit လုပ်တာ ပိုယုံကြည်စိတ်ချရပါတယ်။

အဲဒီ code လေးက browser မှာ ဆိုရင် ဒီလိုပေါ်လာပါမယ်။



Figure 8-2.

စာထည့်ပြီးတော့ submit ကို နှိပ်လိုက်ရင် email form လေး တက်လာပါမယ်။

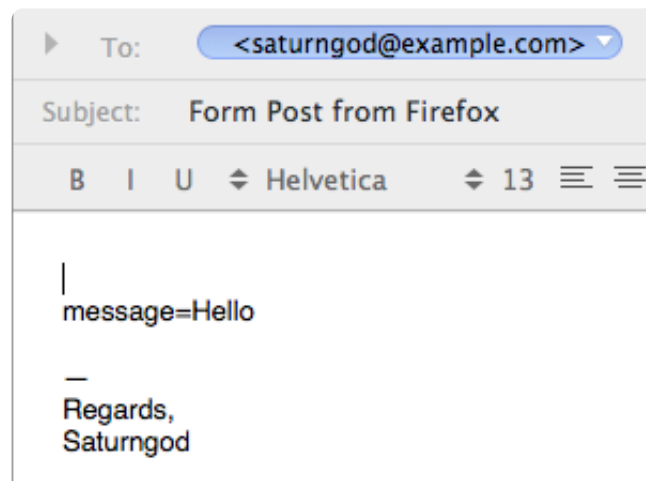


Figure 8-3.

အဆင်သင့် email ပို့လိုက်ရုံပါပဲ။

Add a Text Box

Text box ဟာ အသုံးပြုသူကို စာတစ်ကြောင်းထဲရှိတဲ့ data တွေ ထည့်သွင်းစေချင်တဲ့ အခါမှာ အသုံးပြုပါတယ်။ ကျွန်တော်တို့ အနေနဲ့ text box ရဲ့ size နောက်ပြီး အများဆုံး စာလုံး အရေအတွက် တွေကို ထိန်းချုပ်လို့ရပါတယ်။ ပုံမှန် အားဖြင့် textbox က စာလုံးရေ ၂၀ သာရှိတဲ့ အရွယ်သာ ရှိပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Simple Form</title>
</head>
<body>
  <form method="post" action="./post.php">
    Enter your name :
    <input type="text" name="name" size=40 maxlength=50/>
  </form>
</html>
```

ဒီ code လေးမှာ input type ရဲ့ နာမည်ကို name လို့ ပေးထားပြီးတော့ size ကတော့ text box ရဲ့ အကျယ်ပါ။ 40 ဆိုတာကတော့ စာလုံးအရေအတွက် ၄၀ စာ ကျယ်တဲ့ text box လို့ ဆိုလိုတာပါ။ maxlength ကတော့ စုစုပေါင်း လက်ခံမယ့် စာလုံးအရေအတွက်ပါ။ အခု code မှာတော့ အများဆုံး အလုံး ၅၀ ပဲ ရိုက်ထည့်ခွင့်ပေးထားပါတယ်။

Enter your name :

Figure 8-4.

တကယ်လို့ text မထည့်ချင်ပဲ password ထည့်ချင်ရင်တော့ type နေရာမှာ password ထည့်ပေးဖို့ လိုပါတယ်။ text နဲ့ password ကွာခြားချက်ကတော့ password ဟာ ရိုက်ထည့်ထားတဲ့ စာလုံးကို မမြင်ရပါဘူး။

```
<input type="password" name="password"/>
```

.....

Figure 8-5.

Add a Large Text Area

အကယ်၍ စာတစ်ကြောင်းတည်းမက data တွေကို ထည့်သွင်းစေလိုတယ်ဆိုရင်တော့ text box အစား textarea ကို အသုံးပြုဖို့လိုပါလိမ့်မယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Simple Form</title>
</head>
<body>
  <form method="post" action="./post.php">
    Enter your comment :
    <br/>
    <textarea name="comment" rows="5" cols="60" wrap="hard">Add your comment
here</textarea>
  </form>
</html>
```

Enter your comment :

Add your comment here

Figure 8-6. Text Area

အဲဒီ code မှာ rows ကတော့ textarea ထဲမှာ စာ ဘယ်နှစ်ကြောင်းပြမလဲ နောက်ပြီး cols ကတော့ textarea အကျယ်ကို စာလုံး အရေအတွက် ဘယ်လောက်ဖော်ပြမလဲကို ဆိုလိုပါတယ်။ wrap ကတော့ ၃ မျိုးရှိပါတယ်။

soft wrap ကတော့ text area ထဲမှာ စာကြောင်း ဖြတ်ပေးပေမယ့် data ကို server ဆီကို ပို့တဲ့ အခါမှာ မဖြတ်ပါဘူး။

hard wrap ကတော့ textarea ကော form data ၂ ခုလုံးမှာ ဖြတ်ပါတယ်။

off ကတော့ စာကြောင်း မဖြတ်ပေးတော့ပါဘူး။ line break ကို သုံးစွဲသူ ကိုယ်တိုင် enter ခေါက် မှသာ line break လုပ်ပေးပါလိမ့်မယ်။

Enter your comment :

Add your comment here Add your comment here Add your comment here Add
 your comment here Add your comment here Add your comment here Add your
 comment here Add your comment here

Figure 8-7. Hard Wrap

Enter your comment :

Figure 8-8. Off Wrap

တကယ်လို့ အသုံးပြုသူကို စာရိုက်ထည့်ခွင့် မပေးချင်ရင်တော့ `readonly` attribute ကို အသုံးပြု နိုင်ပါတယ်။

```
<textarea cols="20" rows="5" wrap="hard" readonly="yes">
```

ပုံမှန် textarea ပါပဲ။ သို့ပေမယ့် စာရိုက်ထည့်လို့ မရပါဘူး။

Add Check Boxes

Data တွေကို တစ်ခု ထက်မက ရွေးချယ်ဖို့အတွက် ကျွန်တော်တို့တွေ checkbox ကို အသုံးပြုပါ တယ်။ checkbox ကို အသုံးပြုဖို့ input ရဲ့ type ကို checkbox လို့ ပြောင်းလိုက်ရင်ရပါပြီ။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Simple Form</title>
</head>
<body>
  <form method="post" action="./post.php">
    Service:

    <input type="checkbox" name="flight" value="yes"> Flight
    <input type="checkbox" name="hotel" value="yes"> Hotel
    <input type="checkbox" name="flight" value="yes"> Car
  </form>
</html>
```

checkbox မှာ name နဲ့ value attribute ကို ထည့်ပေးရပါတယ်။ Form submit လုပ်တဲ့ အခါ checkbox ရဲ့ data ဟာ value ထဲမှာ ထည့်ထားတဲ့ data ကို submit လုပ်ပေးပါလိမ့်မယ်။ Checkbox အတွက်စာကို သီးသန့် ရေးပေးရပါတယ်။

အထက်ပါ code ကို run ရင် အောက်ကလို မြင်ရပါမယ်။

Service: Flight Hotel Car

Figure 8-9

တကယ်လို့ page တက်လာကတည်းက checkbox ကို check လုပ်ထားစေချင်ရင်တော့ checked attribute လေး ထည့်ပေးဖို့လိုပါတယ်။

```
<input type="checkbox" name="flight" value="yes" checked>
```

Returning customer : Yes No

Figure 8-10

Add Radio Buttons

Group ထဲမှာ တစ်ခုကိုပဲ ရွေးချယ်စေချင်ရင်တော့ radio button ကို အသုံးပြုနိုင်ပါတယ်။ ဥပမာ။ Yes , No အဖြေလိုမျိုးပေါ့။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Simple Form</title>
</head>
<body>
  <form method="post" action="./post.php">
    Returning customer :

    <input type="radio" name="returning" value="yes">Yes
    <input type="radio" name="returning" value="no">No

  </form>
</html>
```

radio အတွက် group ဖြစ်အောင် name ကို အတူတူဖြစ်အောင် ပေးထားဖို့လိုပါတယ်။ select လုပ်လိုက်တဲ့ value ကို backend server ဆီကို ပို့ပေးပါလိမ့်မယ်။

Returning customer : Yes No

Figure 8-10

တကယ်လို့ page တက်လာကတည်းက radio button တစ်ခုကို check လုပ်ထားစေချင်ရင်တော့ checked attribute လေး ထည့်ပေးဖို့လိုပါတယ်။

```
<input type="radio" name="returning" value="yes" checked>
```

Add a Menu List

Value တစ်ခုကို ရွေးချယ်စေချင်ရင် menu list ကို ကျွန်တော်တို့တွေ အသုံးပြုပါတယ်။ Option နဲ့ မတူတာကတော့ option ဟာ ရွေးချယ်ရမယ့် value တွေကို အကုန်ကို ဖော်ပြထားပြီးတော့ menu list ကတော့ မဖော်ပြထားပါဘူး။ Drop down arrow လေးကို နှိပ်လိုက်မှသာ ဖော်ပြပါလိမ့်မယ်။ `<select>` tag ကို အသုံးပြုပြီးတော့ ရွေးချယ်စေချင်တဲ့ data တွေကိုတော့ `<option>` tag နဲ့ ထည့်သွင်းပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Simple Form</title>
</head>
<body>
  <form method="post" action="./post.php">
    Continent:
    <select name="continent">
      <option value="africa">Africa</option>
      <option value="antarctica">Antarctica</option>
      <option value="asia">Asia</option>
      <option value="australia">Australia</option>
      <option value="eruoep">Eruope</option>
      <option value="northamerica">North America</option>
      <option value="southamerica">South America</option>
    </select>

  </form>
</html>
```

ဆိုတဲ့ code လေးမှာ select tag ကို အသုံးပြုထားပါတယ်။ option ရဲ့ value ဟာ data ကို server ဆီ ပို့တဲ့ အခါမှာ ရောက်သွားမယ့် တန်ဖိုးပါ။ ဥပမာ။။ South America ကို ရွေးချယ်ထားရင် server ဆီ data ရောက်သွားတဲ့ အခါမှာ `continent=southamerica` နဲ့ ရောက်သွားပါလိမ့်မယ်။

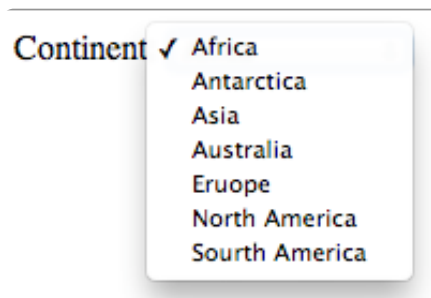


Figure 8-11

တကယ်လို့ တစ်ခု ထက်မက ရွေးချယ်ဖို့ ဖော်ပြစေလိုရင်တော့ `size` attribute ကို အသုံးပြုနိုင်ပါတယ်။

```
<select name="continent" size=5>
```

`size=5` ဆိုတာကတော့ ၅ ခု စာဖော်ပြမယ်လို့ ဆိုလိုတာပါ။ ကျန်တဲ့ စာတွေကိုတော့ scroll ဆွဲဖို့ လိုပါတယ်။

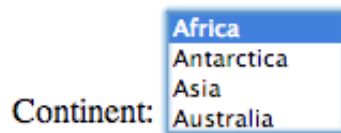


Figure 8-12

တကယ်လို့ value တစ်ခုထက် မက ကို select လုပ်ခွင့်ပေးချင်ရင်တော့ `multiple` attribute ကို သုံးဖို့လိုပါတယ်။ select ကိုတော့ windows မှာ ဆိုရင်တော့ Ctrl ကို ဖိကာ click လုပ်ဖို့ လိုအပ်ပါတယ်။ mac မှာဆိုရင်တော့ command ကို ဖိပြီး click လုပ်ဖို့ လိုအပ်ပါတယ်။

```
<select name="continent" multiple>
```



Figure 8-13

Add a Date and Time Input

ကျွန်တော်တို့တွေ date တွေကို ရွေးချယ်ခွင့်ပေးစေချင်ရင် modern HTML မှာ date input type ကို အသုံးပြုနိုင်ပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Simple Form</title>
</head>
<body>
  <form method="post" action="./post.php">
    Date : <input type="date" step="1">
    <br/>
    Month : <input type="month" name="lastMonth" min="2000-01" max="2014-06"
    step="1">
```

```

<br/>
Week : <input type="week" name="lastweek" step="2">
<br/>
Time : <input type="time" name="time" step="1">
      </form>
</html>

```

Date နဲ့ Time အတွက် input type တွေဟာ

- date
- month
- week
- time

စတဲ့ input type တွေကို အသုံးပြုနိုင်ပါတယ်။ `max` နဲ့ `min` ကတော့ အနည်းဆုံး date နဲ့ အများဆုံး ထည့်လို့ ရတဲ့ input ဖြစ်ပါတယ်။ `up` , `down` arrow နှိပ်ပြီး value ကို ပြောင်းနိုင်ပါတယ်။ အခု code မှာ အတိုးအလျှော့ကို 1 ပဲ ထည့်ထားပါတယ်။ တကယ်လို့ ၁ မဟုတ်စေချင်ရင်တော့ `step` attribute ကို ပြင်နိုင်ပါတယ်။

Date နဲ့ time ကို field တစ်ခုတည်းမှာ တွဲထည့်ချင်ရင်တော့ `datetime-local` ကို အသုံးပြုနိုင်ပါတယ်။ Time zone ပါဝင်တဲ့ old `datetime` input type ကိုတော့ modern HTML မှာ မသုံးတော့ပါဘူး။

```
<input type="datetime-local" name="meetingTime">
```

Date :

Month :

Week :

Time :

Figure 8-14

တကယ်လို့ နံပါတ်တွေပဲ လက်ခံစေချင်ရင်တော့ number value ကို အသုံးပြုနိုင်ပါတယ်။

```
<input type="number" name="age" value="21" min="1" max="120" step="0.5">
```

submit လုပ်တဲ့ အခါမှာ data တွေထည့်လိုက်စေချင်တယ်။ သို့ပေမယ့် user က မပြစေချင်ဘူး ။ ဆိုရင်တော့ hidden type ကို အသုံးပြုနိုင်ပါတယ်။

```
<input type="hidden" name="special" value="123">
```

Add an E-Mail Field

ကျွန်တော်တို့တွေ input data ထည့်သွင်းတဲ့ အခါမှာ email ကိုသာ လက်ခံစေလို ရင် email type ကို အသုံးပြုလို့ရပါတယ်။ ပုံမှန် text field နဲ့ အတူတူပါပဲ။ type မှာ email ပြောင်းဖို့သာ လိုပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Simple Form</title>
</head>
<body>
  <form method="post" action="./post.php">
    Email :
    <input type="email" name="email" size="20" maxlength=255>
    <input type="submit">
  </form>
</html>
```

အခု code မှာ email type ကို

```
<input type="email" name="email" size="20" maxlength=255>
```

ဆိုပြီး ထည့်သွင်းထားပါတယ်။ text box က attribute လိုပါပဲ။ character အရေအတွက် အလုံး ၂၀ ဖော်ပြမယ်။ အများဆုံး စာလုံး အရေအတွက် ၂၅၅ လုံးပဲ လက်ခံမယ်။ iPhone , Android စတဲ့ mobile phone က အသုံးပြုရင်တော့ keyboard ကို email keyboard နဲ့ ပြပေးပါလိမ့်မယ်။

အဲဒီ code က browser မှာ ဆိုရင် အောက်ကလို ပေါ်နေပါလိမ့်မယ်။

Email :

Figure 8-15

တကယ်လို့ email format အမှန် မဟုတ်ပဲ submit လုပ်လိုက်ရင် error ပြနေပါလိမ့်မယ်။

Email :

! Please include an '@' in the email address. 'test' is missing an '@'.

Figure 8-16. Email validation

တကယ်လို့ telephone ကို လက်ခံချင်ရင်တော့ tel type ကို အသုံးပြုရပါလိမ့်မယ်။

```
<input type="tel" name="telephone">
```

tel က modern HTML မှာ ပါတဲ့ input type ပါ။ သို့ပေမယ့် browser မှာတော့ validation မရှိပါဘူး။ iPhone , Android စတဲ့ mobile phone က အသုံးပြုရင်တော့ keyboard ကို telephone keyboard နဲ့ ပြပေးပါလိမ့်မယ်။

Add a URL Field

ကျွန်တော်တို့တွေ form မှာ URL ကို လက်ခံလိုရင်တော့ url type ကို အသုံးပြုရပါတယ်။ url type ဟာ modern HTML မှာ ပါတဲ့ feature ဖြစ်ပြီးတော့ form ကို submit လုပ်တဲ့ အခါမှာလည်း validation လုပ်ပေးပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Simple Form</title>
</head>
<body>
  <form method="post" action="./post.php">
    Website :
    <input type="url" name="website" size="20" maxlength=255>

    <input type="submit">
  </form>

</html>
```

URL အတွက်က email လိုပဲ။ type မှာ url ပြောင်းထားလိုက်ပါတယ်။

Website :

Figure 8-17

Form submit လုပ်တဲ့ အခါမှာ URL မဟုတ်တဲ့ အခါမှာ error ပြပေးပါလိမ့်မယ်။

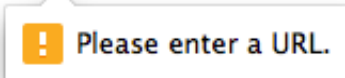
Website :


Figure 8-18

Add a Range Slider

Value တွေကို တစ်ခု နဲ့ တစ်ခုကြားက ရွေးချယ်ပေးစေလိုရင်တော့ ကျွန်တော်တို့တွေကို Range ကို အသုံးပြုနိုင်ပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Simple Form</title>
</head>
<body>
  <form method="post" action="./post.php">
    Budget :
    <input type="range" name="budget" min="100" max="5000" value="1000" step="100">
  </form>
</html>
```

အထက်ပါ code မှာ

```
<input type="range" name="budget" min="100" max="5000" value="1000" step="100">
```

ကို အသုံးပြုထားပြီးတော့ range လေး ထည့်ထားပါတယ်။ min ကတော့ အနည်းဆုံး တန်ဖိုးနဲ့ max ကတော့ အများဆုံး တန်ဖိုးပေါ့။ အခု ဥပမာ မှာ ၁၀၀ ကနေ ၅၀၀၀ အတွင်း ရွေးခွင့်ပေး ထားတယ်။ `step="100"` ဆိုတဲ့ သဘောကတော့ value ပြောင်းလဲရင် 100 အတိုးအလျော့လုပ် မယ်။ Browser မှာ ဆိုရင် အောက်ကလိုမျိုး မြင်ရပါမယ်။

Budget : 

Figure 8-19

သို့ပေမယ့် range value ပြောင်းသွားရင် ကျွန်တော်တို့တွေ မသိပါဘူး။ ဒါကြောင့် ကျွန်တော်တို့ တွေ JavaScript ကို သုံးပြီးတော့ value ပြောင်းသွားတာကို ဖော်ပြပေးပါမယ်။ ဒီအခန်းမှာတော့ JavaScript အကြောင်း အသေးစိတ်မပြောတော့ပါဘူး။ JavaScript အခန်းမှာလည်း event တွေ အကြောင်းပါဝင်လာပါလိမ့်မယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Simple Form</title>
</head>
<body>
  <form method="post" action="./post.php">
    Budget :
    <input type="range" name="budget" min="100" max="5000" value="1000" step="100"
```

```

<span id="display">1000</span>
  </form>
</html>

```

အဲဒီ code မှာ ကျွန်တော်တို့ကတွေ့ range ရဲ့ **onchange** event မှာ script လေးရေးထားတာကို တွေ့ရပါလိမ့်မယ်။

```
onchange="document.getElementById('display').innerHTML=this.value"
```

value change သွားတိုင်း id ကို display လို့ပေးထားတဲ့ element က html ကို သွားပြင်ပေးပါလိမ့်မယ်။ ဒါကြောင့် range slider value ပြင်လိုက်တိုင်း လိုက်ပြီး update လုပ်ပေးနေပါလိမ့်မယ်။

Budget :  2300

Figure 8-20

Add a File Upload

File တွေကို တင်မယ် ၊ profile ပုံကိုတင်မယ် စတာတွေအတွက် form မှာ file upload ကို အသုံးပြုရပါတယ်။ File upload တင်မယ်ဆိုရင်တော့ method ဟာ post ကို သုံးရပါမယ်။ နောက်ပြီးတော့ enctype ကို multipart/form-data သုံးရပါမယ်။

Type	Description
application/x-www-form-urlencoded	Default တန်ဖိုး ဖြစ်ပါတယ်။ စာလုံးတွေ အားလုံးကို encode လုပ်ပြီးတော့ backend server ဆီပို့ပါမယ်။ ဥပမာ space ဆိုရင် + symbols ပြောင်းပြီး ပို့ပါတယ်။
multipart/form-data	File upload လုပ်တဲ့ အခါမှာ အသုံးပြုရန် အတွက်ပါ။
text/plain	Space ကို "+" ပြောင်းပေးပါတယ်။ သို့ပေမယ့် special characters တွေကိုတော့ encoded မလုပ်ပါဘူး။

HTML file upload code လေးကြည့်ရအောင်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Simple Form</title>
</head>
<body>
  <form method="post" action="./post.php" enctype="multipart/form-data">
    Photo :
    <input type="file" name="photo">

```

```

    </form>
</html>

```

အဲဒီ code လေးကို browser မှာ run လိုက်ရင် file upload control လေးနဲ့ မြင်ရပါလိမ့်မယ်။ Choose File ကို နှိပ်ပြီးတော့ file ကို ရွေးချယ်နိုင်ပါတယ်။

Photo : No file chosen

Figure 8-21. File upload

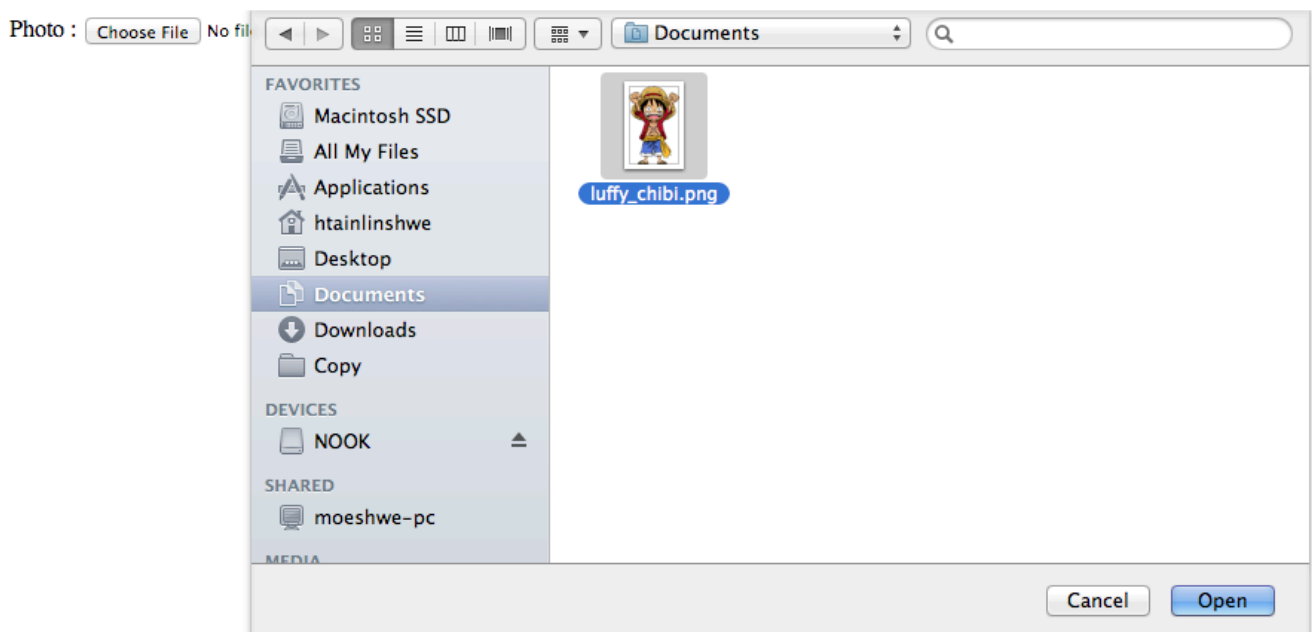


Figure 8-22. file ရွေးချယ်ရန်

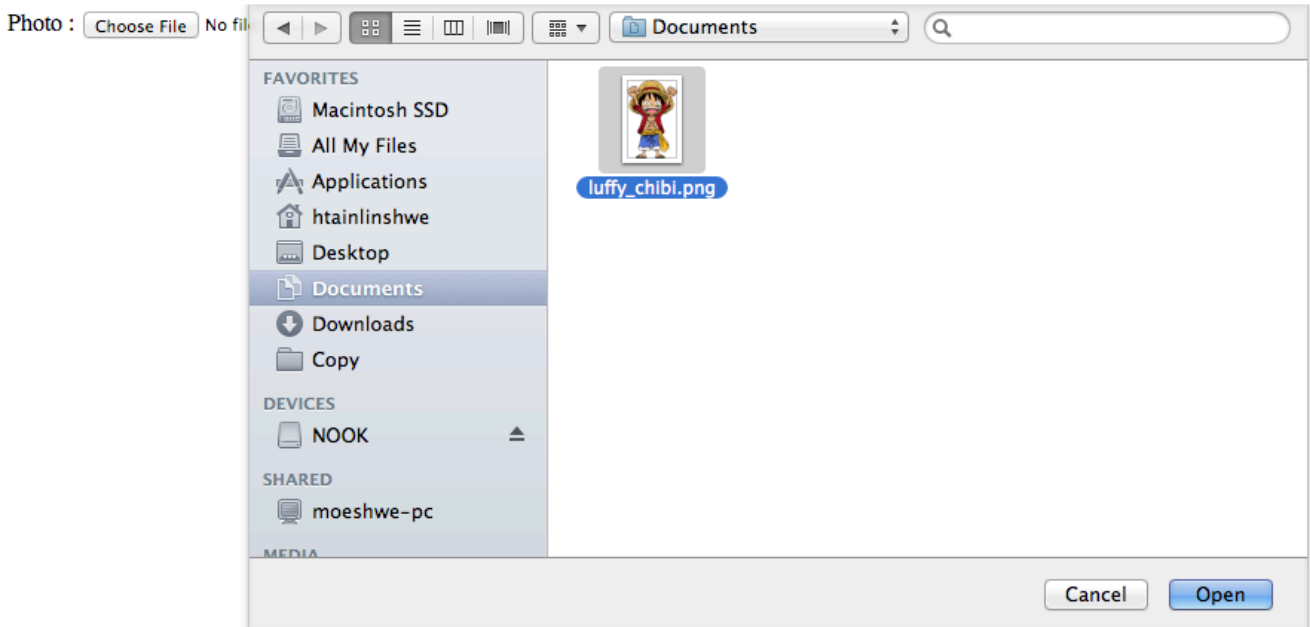


Figure 8-22

တကယ်လို့ file အားလုံးကို လက်မခံလိုဘူးဆိုရင်တော့ MIME (Multipurpose Internet Mail Extensions) types ကို accept attribute ထဲမှာထည့်လို့ရပါတယ်။

```
<input type="file" name="photo" accept="image/gif, image/jpeg, image/png">
```

အထက်ပါ code မှာဆိုရင်တော့ gif, jpg , png စတဲ့ file တွေကိုပဲ လက်ခံပါလိမ့်မယ်။

Add a Submit Button

Data တွေကို server ဆီကို ပို့တော့မယ်ဆိုရင် submit button ကို နှိပ်ပြီး ပို့နိုင်ပါတယ်။ submit button ဟာ form submit event ကို သွားခေါ်ပေးပြီး form ထဲမှာရှိရင် data တွေကို name=value နဲ့ backend ဆီကို ပို့ပေးပါတယ်။ ပုံမှန် အားဖြင့် submit button ကို form ရဲ့ အောက်ဆုံး မှာ ထားတတ်ပါတယ်။ input type နေရာမှာ submit ကို အစားထိုး အသုံးပြုရုံပါပဲ။ value မထည့်ထားရင်တော့ default စာလုံး ဖြစ်တဲ့ Submit ဆိုပြီး ဖော်ပြပေးနေမှာပါ။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Simple Form</title>
</head>
<body>
  <form method="post" action="./post.php" enctype="multipart/form-data">
    Name : <input type="text" name="name">
    <br/>
    Birth Date : <input type="date" step="1">
    <br/>
    Budget :
```

```

<input type="range" name="budget" min="100" max="5000" value="1000" step="100"
<span id="display">1000</span>
  <br/>
  <input type="submit" value="Send">
    </form>
</html>

```

အထက်ပါ code ဟာ browser မှာ ဆိုရင် အောက်က ပုံလို မြင်ရပါလိမ့်မယ်။ Submit button ဟာ Send ဆိုပြီး ဖော်ပြနေပါလိမ့်မယ်။

The screenshot shows a web form with the following elements:

- Name :** A text input field.
- Birth Date :** A date input field with a placeholder "dd/mm/yyyy".
- Budget :** A range slider with a circular handle and a value of "1000" displayed to its right.
- Send** : A button with the text "Send".

Figure 8-24

Add a Reset Button

Form မှာ data တွေ ထည့်သွင်းထားပြီးရင် ပြန်ပြီးတော့ clear လုပ်စေချင်ရင် reset button ကို ထည့်သွင်းနိုင်ပါတယ်။ reset button ကို နှိပ်လိုက်တာနဲ့ form မှာ ဖြည့်ထားတဲ့ data တွေ အားလုံးကို ရှင်းလိုက်မှာ ဖြစ်ပါတယ်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Simple Form</title>
</head>
<body>
  <form method="post" action="./post.php" enctype="multipart/form-data">
    Name : <input type="text" name="name">
    <br/>
    Birth Date : <input type="date" step="1">
    <br/>
    Budget :
    <input type="range" name="budget" min="100" max="5000" value="1000" step="100"
    <span id="display">1000</span>
    <br/>
    <input type="submit" value="Send">
    <input type="reset" value="Reset">
    </form>
  </body>
</html>

```

ပြီးခဲ့တဲ့ submit code အောက်မှာ reset button ထပ်ဖြည့်ထားပါတယ်။

Name :

Birth Date :

Budget : 1000

Figure 8-25

Require a Field

Form ကို submit လုပ်တဲ့ အခါမှာ အချို့ data တွေကို မဖြစ်မနေ ထည့်သွင်းစေချင်ပါတယ်။ အဲဒီ အခါမှာ require attribute ကို ထည့်သွင်းနိုင်ပါတယ်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Simple Form</title>
</head>
<body>
  <form method="post" action="./post.php" enctype="multipart/form-data">
    Name : <input type="text" name="name" required>
    <br/>
    Birth Date : <input type="date" step="1" required>
    <br/>
    Budget :
    <input type="range" name="budget" min="100" max="5000" value="1000" step="100">
    <span id="display">1000</span>
    <br/>
    <input type="submit" value="Send">
    <input type="reset" value="Reset">
  </form>
</body>
</html>

```

အဲဒီ code လေးဟာ browser မှာ ဆိုရင် အောက်ဖော်ပြပါ ပုံအတိုင်း မြင်ရပါလိမ့်မယ်။

Name :

Birth Date :

Budget : 1000

Figure 8-26

name နဲ့ birth date ကို required ထည့်သွင်းထားပါတယ်။

```

Name : <input type="text" name="name" required>
<br/>

```

Birth Date : `<input type="date" step="1" required>`

data တွေ မထည့်ပဲ submit လုပ်လိုက်တဲ့ အခါမှာ browser က error ပြပါလိမ့်မယ်။

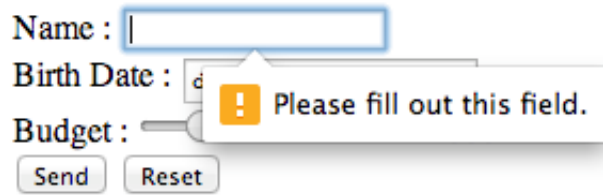


Figure 8-27

Add a Placeholder

Text box မှာ data မထည့်သွင်းထားရင် စာ တစ်ခုခု ဖော်ပြထားစေချင်တယ်ဆိုရင် placeholder attribute ကို အသုံးပြုနိုင်ပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Simple Form</title>
</head>
<body>
  <form method="post" action="./post.php">
    Website :
    <input type="url" name="website" size="25" maxlength=255
    placeholder="http://www.yoursite.com">
    <input type="submit">
  </form>
</html>
```

code ထဲမှာ placeholder ကို http://www.yoursite.com ဆိုပြီး ထည့်ထားပါတယ်။ စာရိုက်လိုက်တဲ့ အခါမှာ အဲဒီ placeholder မှာ ထည့်ထားတဲ့ စာသားပျောက်သွားတာကို တွေ့ရပါလိမ့်မယ်။

Website :

Figure 8-28

Validate Input with a Pattern

ကျွန်တော်တို့တွေ form မှာ data input ကို require သာ မဟုတ်ပဲ ထည့်တဲ့ data ကို ပြန်စစ်ချင် တယ်ဆိုရင် pattern attribute ကို အသုံးပြုနိုင်ပါတယ်။ ဥပမာ။။ 0-9 ပဲ ရိုက်ထည့်ခွင့်ပေးမယ်။ ဒါ မှမဟုတ် A-z နဲ့ 0-9 ပဲ လက်ခံမယ်။ special character တွေ လက်မခံလိုဘူး။ ဒါတွေ အတွက် pattern ကို အသုံးပြုနိုင်ပါတယ်။ pattern ကို အသုံးပြုဖို့ regular expression ကိုတော့ နားလည် ထားဖို့လိုပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Simple Form</title>
</head>
<body>
  <form method="post" action="./post.php">
    Promotion code (only number and letters) :
    <input type="text" name="promocode" size="15" maxlength="15" pattern="[A-z0-9]*">
    <input type="submit">
  </form>
</html>
```

pattern ထဲမှာ [] က range ကိုဆိုလိုပါတယ်။ * ကတော့ zero or more ပါ။ pattern="[A-z0-9]*" ဆိုတဲ့ သဘောကတော့ A ကနေ z အထိ 0 ကနေ 9 အထိ လက်ခံတယ်။ ထည့်တဲ့ value က မရှိတာလည်း ဖြစ်နိုင်သလို တစ်ခု သို့မဟုတ် တစ်ခု ထက်မကလည်း ဖြစ်နိုင်တယ်။

Promotion code (only number and letters) :

Figure 8-29

ဒါကြောင့် special character ထည့်ပြီးတော့ submit လုပ်လိုက်တဲ့ အခါမှာ error ပြပါလိမ့်မယ်။

Promotion code (only number and letters) :

! Please match the requested format.

Figure 8-30

နောက်ထပ် ဥပမာ တစ်ခု ကြည့်ရအောင်။

```
<!DOCTYPE html>
<html>
<head>
  <title>Simple Form</title>
</head>
<body>
  <form method="post" action="./post.php">
    Phone Number (example: 95-91111111) :
```

```

<input type="text" name="phone" size="12" maxlength="12" pattern="[0-9]{2}-[0-9]{9}">
<input type="submit">
</form>
</html>

```

အဲဒီ code မှာ phone number ကို ရှေ့ဆုံးမှာ နံပါတ် ၂ ခု လာမယ်။ ပြီးရင် - လာမယ်။ နောက်ဆုံးမှာ နံပါတ် ၉ ခု ထည့်ရမယ်။ စာလုံး အရေအတွက် {1} ကိုအသုံးပြုပါတယ်။ [0-9]{2}-[0-9]{9} ဆိုတဲ့ သဘောကတော့ 0 ကနေ ၉ အတွင်း ဖြစ်ရမယ်။ {2} ဆိုတဲ့ အတွက်ကြောင့် စာလုံး ၂ လုံး ဖြစ်ရမယ်။ ပြီးတော့ - လာတယ်။ ပြီးတဲ့ အခါမှာ 0 ကနေ 9 အထိ စာလုံး ၉ လုံး ဖြစ်တဲ့ အတွက်ကြောင့် [0-9]{9} ကို အသုံးပြုထားပါတယ်။

Using Semantic HTML Elements

Modern HTML မှာ website ကို တည်ဆောက်ရာမှာ **Semantic tags** တွေ ထည့်သွင်း တည်ဆောက်ပါတယ်။ ဥပမာ။ navigation အတွက် ဆိုရင် `<nav>` tag ကို အသုံးပြုပါတယ်။ header အပိုင်းအတွက်ဆိုရင်တော့ `<header>` tag ကို အသုံးပြုပြီး တည်ဆောက်ပါတယ်။ အဲဒီ tag တွေကို CSS နဲ့ တွဲပြီးတော့ အရောင်တွေ စာလုံးပုံစံတွေ စတာတွေကို ပြောင်းလဲပါတယ်။

Define a Section

`<section>` tag ဟာ ပုံမှန် အားဖြင့် web page မှာ အုပ်စုတစ်ခု အနေနဲ့ ဖွဲ့ဖို့ အတွက် အသုံးပြုပါတယ်။ `<section>` tag ထဲမှာ ပုံမှန်အားဖြင့် content တွေ ပါဝင်တတ်ပါတယ်။ Single page website တွေမှာ ဆိုရင် section tag နဲ့ အပိုင်းတွေ ခွဲပြီး သုံးတတ်ပါတယ်။ Modern HTML မှာ section tag ဟာ div tag ထက် ပိုပြီး အသေးစိတ်အကြောင်းအရာတွေ ပါဝင်တယ်လို့ ဆိုနိုင်တယ်။ ဥပမာ `<article>` tag ဆိုရင် blog post တွေ သတင်း အကြောင်းအရာတွေ ရေးသားထားတယ်လို့ ဆိုနိုင်ပါတယ်။ `<nav>` ဆိုရင်တော့ navigation link တွေ ဖြစ်တယ်ဆိုတာကို သိနိုင်ပါတယ်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Section</title>
</head>
<body>
  <h1>One Piece</h1>
  <section id="intro">
    <p>One Piece is a Japanese manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 73rd volume released as of March 2014. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.</p>
  </section>
</body>
</html>

```

အဲဒီ code ဟာ browser မှာ ဆိုရင် ပုံမှန် div နဲ့ အသုံးပြုထားသလို ဘာမှ ထူးထူးခြားခြား မပြောင်းလဲပါဘူး။

ကျွန်တော်တို့ section အတွက် margin လေးထည့်ကြည့်ရအောင်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Section</title>
  <style>
    section#intro {
      margin-left:20px;
    }
  </style>
</head>
<body>
  <h1>One Piece</h1>
  <section id="intro">
    <p>One Piece is a Japanese manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 73rd volume released as of March 2014. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.</p>
  </section>
</body>
</html>

```

အဲဒါဆိုရင်တော့ ဘယ်ဘက်ကနေ 20px ကွာသွားတာကို တွေ့ရပါလိမ့်မယ်။

One Piece

One Piece is a Japanese manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 73rd volume released as of March 2014. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

Figure 9-1

ဘကြောင့် semantic tags ဟာ အရေးပါလဲ ?

HTML အဟောင်းတွေမှာ structure နဲ့ ပတ်သက်ပြီးတော့ သတ်သတ်မှတ်မှတ် နည်းပါးခဲ့ပါတယ်။ ဥပမာ။ page header ကိုလည်း <div> အသုံးပြုပြီးတော့ class သို့မဟုတ် id name ကို header လို့ပေးပြီး အသုံးပြုကြတယ်။ Modern HTML မှာတော့ header, navigation, footer နဲ့ အခြား element တွေဟာ tag အနေနဲ့ သတ်သတ်မှတ်မှတ်ရှိလာပါတယ်။ အဲဒါကြောင့် ပိုပြီး နားလည်မှု လွယ်ကူပြီးတော့ web designer တွေ အနေနဲ့လည်း code ရေးရတာ ပိုပြီး ရှင်းလင်း သွားပါတယ်။

Define an Article

`<article>` tag ကို သတင်းစာ ဆောင်းပါးတွေ ၊ blog post တွေ နောက်ပြီးတော့ မှတ်ချက် တွေ နေရာမှာ အသုံးပြုကြပါတယ်။ `<article>` tag ထဲမှာ ပုံမှန် အားဖြင့် ခေါင်းစဉ် ၊ စာရေးသူ ၊ ရက်စွဲ စတာတွေ ပါဝင်တတ်ပါတယ်။ CSS ကို အသုံးပြုပြီးတော့ style ကို ပြင်နိုင်ပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Section</title>
  <style>

  article.feature {
    margin-left:20px;
    font-size:16px;
  }

  article.feature h1 {
    font-size:30px;
  }
  article.feature h2 {
    font-size:20px;
  }
</style>
</head>
<body>
  <article class="feature">
    <h1>One Piece</h1>
    <p>One Piece is a Japanese manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 73rd volume released as of March 2014. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.</p>
  </article>
</body>
</html>
```

ဒီ code က ပုံမှန် div အစား ကျွန်တော်တို့တွေ article ကို ပြောင်းပြီး အသုံးပြုထားပါတယ်။ CSS ကိုတော့ ကျွန်တော်တို့ ရှေ့မှာ ရေးထားခဲ့သလို font-size ပြောင်းထားတာရယ် နောက်ပြီးတော့ article ကို margin-left အတွက် 20px သုံးထားပါတယ်။

Define a Header

`<header>` tag ကိုတော့ web page ရဲ့ title ဒါမှမဟုတ် မိတ်ဆက် စတဲ့ အချက်အလက်တွေကို ဖော်ပြတဲ့ နေရာမှာ အသုံးပြုကြပါတယ်။ `<header>` tag အတွင်းမှာ `<hgroup>` tag သို့မဟုတ် header image အတွက် လည်း အသုံးပြုပါတယ်။

```

<!DOCTYPE html>
<html>
<head>
  <title>Header</title>
</head>
<body>
  <header>
    
  </header>
  <section>
    <p>One Piece is a Japanese manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 73rd volume released as of March 2014. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.</p>
    <p>The chapters have been adapted into an original video animation (OVA) produced by Production I.G in 1998, and an anime series produced by Toei Animation, which began broadcasting in Japan in 1999. Since then, the still ongoing series has aired over 600 episodes. Additionally, Toei has developed eleven animated feature films, two OVA's, and five television specials. Several companies have developed various types of merchandising such as a trading card game, and a large number of video games.</p>
  </section>
</body>
</html>

```

Hgroup

h tag တွေ တစ်ခုထက် မက အသုံးပြုထားရင် `<hgroup>` ကို အသုံးပြုနိုင်ပါတယ်။ ဥပမာ။

```
<hgroup><h1>My new blog</h1><h2>Enjoy</h2></hgroup>
```

hgroup ကို `<header>` tag အတွင်းမှာ အသုံးပြုနိုင်ပါတယ်။

Define Navigation

Navigation တွေအတွက်တော့ `<nav>` ကို အသုံးပြုနိုင်ပါတယ်။ nav tag ကတော့ browser မှာ ထူးထူးခြားခြား မပြောင်းလဲသွားပါဘူး။ CSS ကို အသုံးပြုပြီး ပုံသဏ္ဍန်ကို ပြောင်းလဲဖို့ လိုပါတယ်။

```

<!DOCTYPE html>
<html>
<head>
  <title>Header</title>
</head>
<body>
  <header>
    
  </header>
  <nav>

```

```

<ul>
  <li><a href="index.html">Home</a></li>
  <li><a href="gallery.html">Gallery</a></li>
  <li><a href="video.html">Video</a></li>
</ul>
</nav>
<section>
  <p>One Piece is a Japanese manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 73rd volume released as of March 2014. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.</p>
  <p>The chapters have been adapted into an original video animation (OVA) produced by Production I.G in 1998, and an anime series produced by Toei Animation, which began broadcasting in Japan in 1999. Since then, the still ongoing series has aired over 600 episodes. Additionally, Toei has developed eleven animated feature films, two OVA's, and five television specials. Several companies have developed various types of merchandising such as a trading card game, and a large number of video games.</p>
</section>
</body>
</html>

```

CSS ကို အသုံးမပြုထားတဲ့အတွက် nav က ဘာမှ ထူးထူးခြားခြား မရှိသေးပါဘူး။



- [Home](#)
- [Gallery](#)
- [Video](#)

One Piece (ワンピース Wan Pīsu?) is a Japanese manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 73rd volume released as of March 2014. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

The chapters have been adapted into an original video animation (OVA) produced by Production I.G in 1998, and an anime series produced by Toei Animation, which began broadcasting in Japan in 1999. Since then, the still ongoing series has aired over 600 episodes. Additionally, Toei has developed eleven animated feature films, two OVA's, and five television specials. Several companies have developed various types of merchandising such as a trading card game, and a large number of video games.

Figure 9-2.

အခု code မှာ CSS တွေ ထပ်ဖြည့်ပါမယ်။

```

<!DOCTYPE html>
<html>
<head>

```

```

<title>Header</title>
<style>
#container {
  margin:0px auto;
  width:760px;
}
header.banner {
  width:100%;
}
header.banner img {
  width:100%;
}
nav.sidebar {
  float:left;
  width:95px;
}
nav.sidebar ul,nav.sidebar li{
  margin:0px;
  padding:0px;
  list-style:none;
}
nav.sidebar li {
  padding-bottom:5px;
}
nav.sidebar ul li a {
  text-decoration : none;
  color: #2980b9;
}
nav.sidebar ul li a:hover {
  text-decoration: underline;
}
section.content {
  float:left;
  width:665px;
}
</style>

```

```
</head>
```

```
<body>
```

```
<div id="container">
```

```
<header>
```

```

```

```
</header>
```

```
<nav class="sidebar">
```

```
<ul>
```

```
<li><a href="index.html">Home</a></li>
```

```
<li><a href="gallery.html">Gallery</a></li>
```

```
<li><a href="video.html">Video</a></li>
```

```
</ul>
```

```
</nav>
```

```
<section class="content">
```

```
<p>One Piece is a Japanese manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 73rd volume released as of March 2014. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.</p>
```

```
<p>The chapters have been adapted into an original video animation (OVA) produced by Production I.G in 1998, and an anime series produced by Toei Animation, which began broadcasting in Japan in 1999. Since then, the still ongoing series has aired over 600 episodes. Additionally, Toei has developed eleven animated feature films, two OVA's, and five television specials. Several companies have developed various types of merchandising such as a trading card game, and a large number of video games.</p>
```

```

</section>
</div>
</body>
</html>

```

CSS ထည့်သွင်းပြီးတဲ့ အချိန်မှာတော့ code ဟာ browser မှာ အောက်ကပုံလို ပေါ်ပါလိမ့်မယ်။ ကျွန်တော်တို့တွေ table နဲ့ ဖန်တီးခဲ့သလို css ကို အသုံးပြုပြီး ပြန်လည်ရေးသားလိုက်တာပါ။



[Home](#)
[Gallery](#)
[Video](#)

One Piece (ワンピース Wan Pisu?) is a Japanese manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 73rd volume released as of March 2014. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

The chapters have been adapted into an original video animation (OVA) produced by Production I.G in 1998, and an anime series produced by Toei Animation, which began broadcasting in Japan in 1999. Since then, the still ongoing series has aired over 600 episodes. Additionally, Toei has developed eleven animated feature films, two OVA's, and five television specials. Several companies have developed various types of merchandising such as a trading card game, and a large number of video games.

Figure 9-3

Define an Aside

အဓိက content နဲ့ သက်ဆိုင်တာတွေကို `<aside>` tag ကို အသုံးပြုပြီး ရေးသားပါတယ်။ `<aside>` tag ကို quotes , glossary terms စသည့် အနေနဲ့ အသုံးပြုကြပါတယ်။ Modern HTML specification အရ `<aside>` tag ဟာ main content နဲ့ ဆက်နွယ်မှုရှိရပါမယ်။

ပြီးခဲ့တဲ့ HTML မှာ ကျွန်တော်တို့တွေ aside ကို ထပ်ဖြည့်ပါမယ်။

```

<aside class="right-side">
  <h2>Straw Hat Pirates</h2>
  <a href="/crew/luffy.html"></a>
  <a href="/crew/zoro.html"></a>
  <a href="/crew/nami.html"></a>
  <a href="/crew/usopp.html"></a>
  <a href="/crew/sanji.html"></a>
  <a href="/crew/chopper.html"></a>
  <a href="/crew/robin.html"></a>
  <a href="/crew/franky.html"></a>

```

```
<a href="./crew/brook.html"></a>
</aside>
```

CSS မထည့်ရသေးတဲ့ အတွက် အောက်ကပုံလို မြင်နေရပါမယ်။



[Home](#)
[Gallery](#)
[Video](#)

One Piece (ワンピース Wan Pīsu?) is a Japanese manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 73rd volume released as of March 2014. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

The chapters have been adapted into an original video animation (OVA) produced by Production I.G in 1998, and an anime series produced by Toei Animation, which began broadcasting in Japan in 1999. Since then, the still ongoing series has aired over 600 episodes. Additionally, Toei has developed eleven animated feature films, two OVA's, and five television specials. Several companies have developed various types of merchandising such as a trading card game, and a large number of video games.

Straw Hat Pirates



Figure 9-4.

CSS ကို အောက်က code ထပ်ဖြည့်လိုက်ပါတယ်။

```
section.content {
  float:left;
  width:570px;
}

aside.right-side {
  float:right;
  width:95px;
}
```

```
aside.right-side h2 {
  font-size:15px;
}
```

section ရဲ့ width ကိုတော့ ၅၇၀ ပြောင်းထားလိုက်ပါတယ်။ code အပြည့်အစုံကတော့

```
<!DOCTYPE html>
<html>
<head>
  <title>Nav</title>
  <style>
    #container {
      margin:0px auto;
      width:760px;
    }
    header.banner {
      width:100%;
    }
    header.banner img {
      width:100%;
    }
    nav.sidebar {
      float:left;
      width:95px;
    }
    nav.sidebar ul,nav.sidebar li{
      margin:0px;
      padding:0px;
      list-style:none;
    }
    nav.sidebar li {
      padding-bottom:5px;
    }
    nav.sidebar ul li a {
      text-decoration : none;
      color: #2980b9;
    }
    nav.sidebar ul li a:hover {
      text-decoration: underline;
    }
    section.content {
      float:left;
      width:570px;
    }

    aside.right-side {
      float:right;
      width:95px;
    }

    aside.right-side h2 {
      font-size:15px;
    }
  </style>
</head>
<body>
  <div id="container">
    <header>
      
    </header>
```

```

<nav class="sidebar">
  <ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="gallery.html">Gallery</a></li>
    <li><a href="video.html">Video</a></li>
  </ul>
</nav>
<section class="content">
  <h1>One Piece</h1>
  <p>One Piece is a Japanese manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 73rd volume released as of March 2014. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.</p>
  <p>The chapters have been adapted into an original video animation (OVA) produced by Production I.G in 1998, and an anime series produced by Toei Animation, which began broadcasting in Japan in 1999. Since then, the still ongoing series has aired over 600 episodes. Additionally, Toei has developed eleven animated feature films, two OVA's, and five television specials. Several companies have developed various types of merchandising such as a trading card game, and a large number of video games.</p>
</section>

<aside class="right-side">
  <h2>Straw Hat Pirates</h2>
  <a href="./crew/luffy.html"></a>
  <a href="./crew/zoro.html"></a>
  <a href="./crew/nami.html"></a>
  <a href="./crew/usopp.html"></a>
  <a href="./crew/sanji.html"></a>
  <a href="./crew/chopper.html"></a>
  <a href="./crew/robin.html"></a>
  <a href="./crew/franky.html"></a>
  <a href="./crew/brook.html"></a>
</aside>
</div>
</body>
</html>

```

browser မှာဆိုရင် အောက်ကလို မြင်ရပါလိမ့်မယ်။



[Home](#)
[Gallery](#)
[Video](#)

One Piece

One Piece (ワンピース Wan Pisu?) is a Japanese manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 73rd volume released as of March 2014. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

The chapters have been adapted into an original video animation (OVA) produced by Production I.G in 1998, and an anime series produced by Toei Animation, which began broadcasting in Japan in 1999. Since then, the still ongoing series has aired over 600 episodes. Additionally, Toei has developed eleven animated feature films, two OVA's, and five television specials. Several companies have developed various types of merchandising such as a trading card game, and a large number of video games.

**Straw Hat
Pirates**



Figure 9-5.

Define a Footer

Footer ကိုတော့ page ရဲ့ အောက်ဆုံး `</body>` ရဲ့ အပေါ်မှာ အသုံးပြုပါတယ်။ HTML specification အရ `<footer>` tag အတွင်းမှာ `<header>` tag ထည့်မသွင်းရပါဘူး။ footer ကို section tag ရဲ့ အောက်ဆုံးမှာလည်း အသုံးပြုနိုင်ပါတယ်။ page တစ်ခုမှာ footer တွေ အများကြီးလည်း ဖြစ်နိုင်ပါတယ်။

အခု footer ကို `</body>` မတိုင်ခင်မှာ ထပ်ဖြည့်လိုက်ပါမယ်။

```
<footer class="bottom">
  <small>&copy; Copyright 2011
  <a href="terms.html">Terms and conditions</a>
</small>
</footer>
```

အဲဒီ အခါ browser မှာ ဒီလို မြင်ရပါလိမ့်မယ်။

The chap
Productio
broadcast
episodes.
and five t
merchand

© Copyright 2011 [Terms and conditions](#)

Figure 9-6.

ကျွန်တော်တို့ css ထပ်ဖြည့်ပါမယ်။

```
footer {
  background:#3498db;
  float:right;
  width:100%;
  color :#FFF;
  text-align:center;
}
```

ကျွန်တော်တို့ရဲ့ code အပြည့်အစုံကို ကြည့်ရအောင်။

```
<!DOCTYPE html>
<html>
<head>
  <title>Nav</title>
  <style>
    #container {
      margin:0px auto;
      width:760px;
    }
    header.banner {
      width:100%;
    }
    header.banner img {
      width:100%;
    }
    nav.sidebar {
      float:left;
      width:95px;
    }
    nav.sidebar ul,nav.sidebar li{
      margin:0px;
      padding:0px;
      list-style:none;
    }
    nav.sidebar li {
      padding-bottom:5px;
    }
    nav.sidebar ul li a {
      text-decoration : none;
      color: #2980b9;
    }
  </style>
</head>
<body>
  <div id="container">
    <div id="header">
      <div id="header-banner">
        <img alt="Banner image" data-bbox="145 585 295 615"/>
      </div>
      <div id="nav">
        <ul>
          <li><a href="#">Home</a></li>
          <li><a href="#">About</a></li>
          <li><a href="#">Contact</a></li>
        </ul>
      </div>
    </div>
  </div>
</body>
</html>
```

```

nav.sidebar ul li a:hover {
  text-decoration: underline;
}
section.content {
  float:left;
  width:570px;
}

aside.right-side {
  float:right;
  width:95px;
}

aside.right-side h2 {
  font-size:15px;
}

footer {
  background:#3498db;
  float:right;
  width:100%;
  color :#FFF;
  text-align:center;
}

</style>
</head>
<body>
<div id="container">
  <header>
    
  </header>
  <nav class="sidebar">
    <ul>
      <li><a href="index.html">Home</a></li>
      <li><a href="gallery.html">Gallery</a></li>
      <li><a href="video.html">Video</a></li>
    </ul>
  </nav>
  <section class="content">
    <h1>One Piece</h1>
    <p>One Piece is a Japanese manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 73rd volume released as of March 2014. One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.</p>
    <p>The chapters have been adapted into an original video animation (OVA) produced by Production I.G in 1998, and an anime series produced by Toei Animation, which began broadcasting in Japan in 1999. Since then, the still ongoing series has aired over 600 episodes. Additionally, Toei has developed eleven animated feature films, two OVA's, and five television specials. Several companies have developed various types of merchandising such as a trading card game, and a large number of video games.</p>
  </section>

  <aside class="right-side">
    <h2>Straw Hat Pirates</h2>
    <a href="./crew/luffy.html"></a>
    <a href="./crew/zoro.html"></a>
    <a href="./crew/nami.html"></a>
    <a href="./crew/usopp.html"></a>
    <a href="./crew/sanji.html"></a>

```

```

    <a href="./crew/chopper.html"></a>
    <a href="./crew/robin.html"></a>
    <a href="./crew/franky.html"></a>
    <a href="./crew/brook.html"></a>
</aside>

</div>

<footer class="bottom">
  <small>&copy; Copyright 2011
  <a href="terms.html">Terms and conditions</a>
</small>
</footer>
</body>
</html>

```

browser မှာ ဆိုရင်တော့ အောက်ကလို မြင်ရပါလိမ့်မယ်။

© Copyright 2011 [Terms and conditions](#)

Figure 9-7.

Working with JavaScript

website မှာ action တွေ ထည့်ချင်တယ် interactive ဖြစ်ချင်တယ်ဆိုရင်တော့ JavaScript ကို အသုံးပြုရပါမယ်။ ဒီအခန်းမှာ JavaScript code ကို ဘယ်လို ထည့်သွင်းရမယ်ဆိုတာကို ဖော်ပြပေးပါမယ်။ ဒီစာအုပ် HTML အဓိကထားပြီး ရေးသားထားတဲ့ စာအုပ်ဖြစ်တာကြောင့် JavaScript နဲ့ ပတ်သက်ပြီးတော့ အသေးစိတ်တော့ ပါဝင်မှာ မဟုတ်ပါဘူး။

Understanding JavaScript

JavaScript ကို web page မှာ dynamic effect တွေ ဖြစ်အောင် ဖန်တီးဖို့ အသုံးပြုကြပါတယ်။ HTML page ဟာ JavaScript နဲ့ ပေါင်းလိုက်တဲ့အခါမှာ ပုံမှန် သမားရိုးကျ HTML ထက် ပိုပြီး ထူးခြားတဲ့ ပုံစံကို ဖန်တီးနိုင်ပါတယ်။ JavaScript ကို အသုံးပြုပြီး div တွေကို show/hide လုပ်တာတွေ နောက်ပြီးတော့ message box ပြတာတွေ image ပြောင်းတာတွေ စသည်ဖြင့် အများကြီး ဖန်တီးနိုင်ပါတယ်။

JavaScript ဟာ script language အမျိုးအစားဖြစ်ပြီးတော့ အသုံးပြုသူ click လုပ်လိုက်တဲ့အခါ ဒါမှမဟုတ် cursor ကို ပုံပေါ်မှာ တင်လိုက်တဲ့ အခါ စသည့် event တွေ ပါဝင်လာပါတယ်။ Script ဟာ ကျွန်တော်တို့ရဲ့ website ထဲက ဖြစ်နိုင်သလို အခြား တနေရာက script တွေကိုလည်း ယူသုံးနိုင်ပါတယ်။

JavaScript ဟာ popular script language ဖြစ်တာကြောင့် browser တွေ အကုန်လုံးမှာ support လုပ်ပါတယ်။ Web browser ဟာ client-side မှာ ရေးသားတဲ့အတွက် JavaScript ကို client-side scripting လို့ ခေါ်ကြပါတယ်။ အခုနောက်ပိုင်း [Node.js](#) ကို အသုံးပြုပြီးတော့ server side အတွက်ပါ ရေးသားနိုင်ပါပြီ။

JavaScript ဟာ Java programming language နဲ့ မတူပါဘူး။ Java ဟာ JavaScript နှင့် ယှဉ်ရင်တော့ အသုံးပြုတဲ့ နေရာလည်း မတူပါဘူး။

Adding Scripts

JavaScript ကို html code နဲ့ တွဲပြီး ရေးနိုင်သလို အပြင်က .js file ကိုလည်း ယူသုံးနိုင်ပါတယ်။ html code အထဲမှာ ရေးလိုရင်တော့ `<script>` နဲ့ `</script>` ကြားမှာ အသုံးပြုနိုင်ပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>JavaScript</title>
```

```

<script type="text/javascript" src="external.js"></script>
<script type="text/javascript">
  function showalert()
  {
    alert("Hello World!");
  }
</script>
</head>
</html>

```

ဒီ code မှာ ဆိုရင် external.js file ကို

```

<script type="text/javascript" src="external.js"></script>

```

နဲ့ အသုံးပြုထားပါတယ်။

Internal အတွက်ကိုတော့

```

<script type="text/javascript">
  function showalert()
  {
    alert("Hello World!");
  }
</script>

```

function ဆိုတာကတော့ အလုပ်လုပ်မယ့် အပိုင်းလေး တစ်ခုပါ။ function ရဲ့ အနောက်မှာ function name ပါပါတယ်။ function name ကို showalert ဆိုပြီး ပေးထားပါတယ်။ function name အနောက်မှာ argument parameter ပါပါတယ်။ ဘာ data မှာ ထည့်တာကို allow မလုပ်ထားတာကြောင့် showalert() ဆိုပြီးပဲ ရေးသားထားတာပါ။ alert ဆိုတာကတော့ message box ပြမယ့် function ပါ။ JavaScript မှာ ပါပြီးသား function ပေါ့။ double quote ကြားမှာ ရေးချင်တဲ့ စာကို ထည့်ပြီး ရေးသားရပါတယ်။

Document Object Model

JavaScript နဲ့ HTML ကို တွဲသုံးတဲ့အခါမှာတော့ DOM (document object model) က မဖြစ်မနေ အရေးပါပါတယ်။ DOM ဆိုတာကတော့ HTML ကို အဆင့်ဆင့် ခွဲသွားတဲ့ tree ပုံစံပါ။

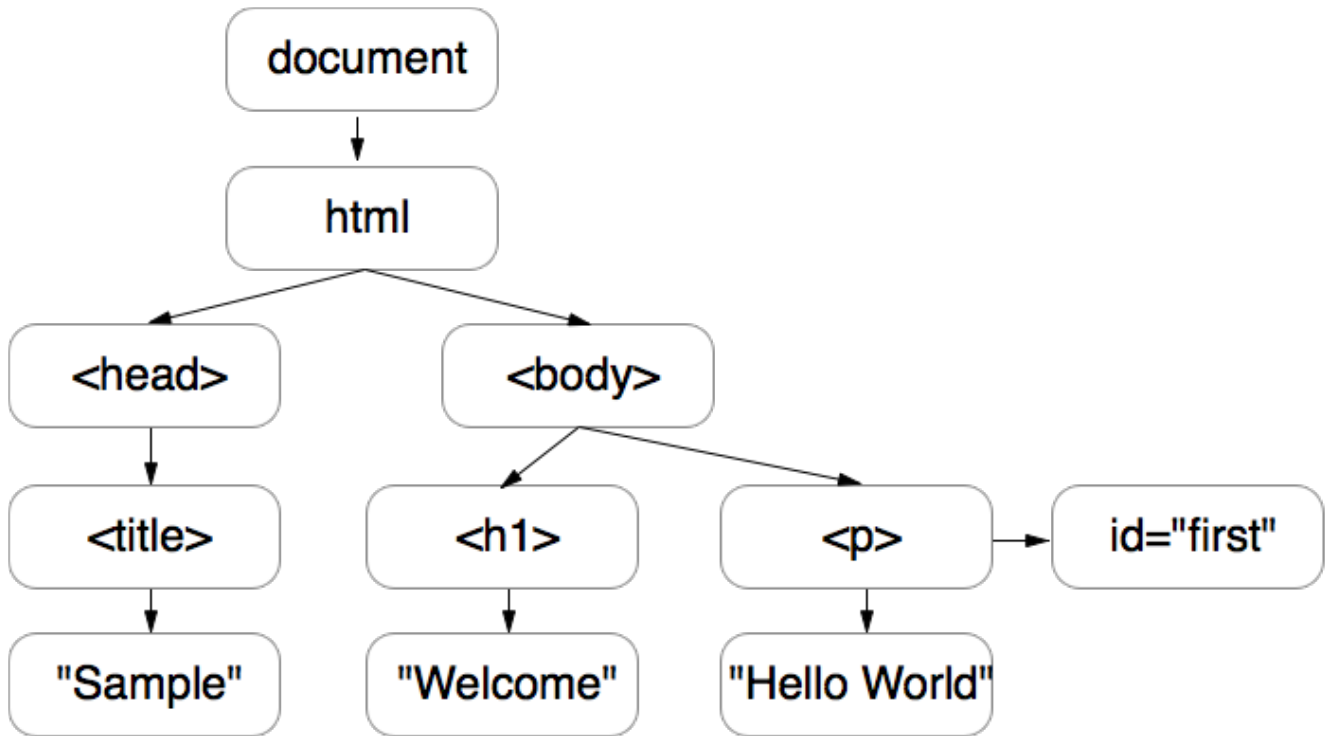


Figure 10-1

Document root ဟာ `<html>` က စပြီးတော့ သူ့အောက်မှာ `<head>` နဲ့ `<body>` ရှိပါတယ်။ `<head>` အောက်မှာတော့ `<title>` ရှိတယ်။ `<title>` ထဲမှာတော့ My Page ဆိုတဲ့ စာ ရှိပါတယ်။ JavaScript ဟာ DOM ကို အသုံးပြုပြီးတော့ အဆင့်ဆင့် ခေါ်ပြီး လိုအပ်တဲ့ value ကို ရအောင် access လုပ်နိုင်ပါတယ်။ ဒါမှမဟုတ် id attribute ကို ခေါ်ပြီးတော့ DOM object ကို ရနိုင်ပါတယ်။

JavaScript Syntax

JavaScript ကို အသုံးပြုပြီးရေးရင် ကျွန်တော်တို့တွေ အနေနဲ့ တစ်ကြောင်းပြီး တစ်ကြောင်း ဘာလုပ်မလဲဆိုတာကို ခိုင်းရပါတယ်။ document ကနေ စာကြောင်းကို ယူမယ်။ ပြီးရင် စာ ကို နောက်မှာ ထပ် ပေါင်းမယ်။ ပြီးရင် document ထဲကို ပြန်ထည့်မယ်။ အဲလိုမျိုး တစ်ကြောင်းဆီ ဘာလုပ်မလဲဆိုတာကို ရေးပေးရပါမယ်။

ဥပမာ။။။

```

var k = 1;
k = k + 5;

```

အဲဒီမှာဆိုရင် k ထဲကို ၁ ထည့်ထားပါတယ်။ ပြီးလျှင် k ထဲကို k ထဲက တန်ဖိုးနဲ့ ၅ နဲ့ ပေါင်းပါဆို ပြီး ရေးသားထားပါတယ်။

သချာက ရေးသားပုံနဲ့ မတူတာကတော့ ညာဘက်ကနေ ဘယ်ဘက် variable ထဲကို တန်ဖိုး ထည့်ပါတယ်။ JavaScript မှာ အဆုံး ကို semicolons (;) နဲ့ ဆုံးရပါတယ်။ programming ကို မလေ့လာဖူးသေးရင်တော့ JavaScript ရေးရမှာ နည်းနည်းခက်ခဲပေမယ့် နားလည် သွားရင် ရေးသားရတာ လွယ်ကူပါလိမ့်မယ်။

နောက်ထပ် ဥပမာ ကြည့်ရအောင်။

```
var x = "Hello";
x = x + " World!";
alert(x);
```

ဒီမှာဆိုရင်တော့ x ကို Hello ဆိုတဲ့ စာထည့်လိုက်ပါတယ်။ ပြီးမှ World! ဆိုတဲ့ စာကြောင်း ကပ်လိုက်ပါတယ်။ alert ဆိုတာကတော့ JavaScript မှာ messagebox ဖော်ပြဖို့ အတွက် အသုံးပြုပါတယ်။

ဒီလောက်ဆိုရင်တော့ အနည်းငယ်လေးတော့ တီးမခေါက်မိရှိလောက်ပါပြီ။

Understanding Script Events and Handlers

JavaScript ကို ထည့်တော့မယ်ဆိုရင် JavaScript ဟာ ဘယ်လို အလုပ်လုပ်တယ်ဆိုတာကို သိဖို့လိုပါတယ်။ JavaScript ဟာ ပုံမှန်အားဖြင့် webpage တက်လာတာနဲ့ အလုပ်လုပ်ပါတယ်။ တချို့တွေကတော့ action တစ်ခုခု ရှိမှသာ အလုပ်လုပ်တာတွေရှိပါတယ်။ web developer အနေနဲ့ script တွေကို ဘယ်လို အလုပ်လုပ်စေချင်လဲဆိုတာကို ရေးသားထားပေးဖို့ လိုပါတယ်။

Script Events

သို့ပေမယ့် ကျွန်တော်တို့တွေဟာ button click လုပ်လိုက်တဲ့ အခါ ဒါမှမဟုတ် ပုံပေါ်ကို mouse ရောက်သွားတဲ့အခါမှသာ အလုပ်လုပ်စေချင်တဲ့အခါတွေမှာ သူနဲ့ သက်ဆိုင်ရာ event တွေမှာ ရေးသားထားပေးရပါတယ်။ JavaScript events action တွေဟာ web page ကို အသုံးပြုနေတဲ့ အချိန်မှာ အလုပ်လုပ်နေပါတယ်။ ဥပမာ။။ button ကို နှိပ်လိုက်ရင် alert ပြတာမျိုးပေါ့။ mouse event အတွက်ကတော့ mouse ရွှေ့လိုက်တဲ့အခါမှာ position ဘယ်နေရာမှာ ရှိလဲဆိုတာကို သိနိုင်ဖို့အတွက်ပါ။ နောက်ပြီး keyboard event တွေအတွက် key တစ်ခုခု နှိပ်လိုက်တိုင်း ဘယ် စာလုံးကို နှိပ်တယ်ဆိုတာကို သိနိုင်ပါတယ်။

Event Handlers

```

```

Event handlers ကတော့ event တစ်ခု ထဲရောက်လာပြီဆိုရင် ဘယ် function ကို သွားခေါ်မလဲဆိုတာကို ရေးသားထား ပေးရပါတယ်။ အထက်ပါ code မှာ ဆိုရင် dog.png ပုံလေးကို နှိပ်လိုက်တဲ့ အခါမှာ onclick event ကို ရောက်လာပါ တယ်။ အဲဒီအခါ handler က bark() function ဖြစ်နေတဲ့အတွက် bark() ကို လှမ်းခေါ်လိုက်ပါတယ်။

နောက်ထပ် ဥပမာလေး ကြည့်ရအောင်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Sample</title>
  <script>
    function showAlert()
    {
      alert('Clicked');
    }
  </script>
</head>
<body>
  
</html>
```

ဒီ code လေးမှာဆိုရင် world.png ပုံလေးကို နှိပ်လိုက်တာနဲ့ alert box လေးတက်လာတာကို တွေ့ရပါလိမ့်မယ်။

Event Handler	Action
onload	browser က page ကို load ဖြစ်တဲ့အချိန် တက်လာတဲ့ event
onunload	Browser က page ကနေ ထွက်သွားတဲ့ အခါမှာ ခေါ်တဲ့ event
onmouseover	element ပေါ်မှာ mouse တင်လိုက်တဲ့ အခါ ရောက်လာတဲ့ event
onmousedown	mouse ခလုတ် နှိပ်လိုက်တဲ့ အချိန် ရောက်လာတဲ့ event
onmouseup	mouse ခလုတ် နှိပ်ပြီး လွှတ်လိုက်တဲ့ အချိန် ရောက်လာတဲ့ event
onmousemove	mouse ရွေ့လိုက်တဲ့အချိန်အတွက်
onmouseout	element ကနေ mouse ထွက်သွားတဲ့ အချိန်
onclick	click နှိပ်လိုက်တဲ့ အခါမှာ ရောက်လာတဲ့ event
ondblclick	double click နှိပ်လိုက်တဲ့ အခါမှာ ရောက်လာတဲ့ event
onkeypress	keyboard ကနေ key ကို ဖိပြီး လွှတ်လိုက်တဲ့ အချိန်မှာ ရောက်လာတဲ့ event
onkeydown	key ကို ဖိထားချိန်အတွက်
onkeyup	key ကို လွှတ်လိုက် ချိန်အတွက်
onsubmit	submit button ကို နှိပ်လိုက်တဲ့ အချိန်အတွက်

Add JavaScript to a Web Page

JavaScript ကို `<script>` နဲ့ `</script>` ကြားမှာ code တွေကို ရေးသားရပါတယ်။ အရင် code တွေမှာ `type="text/javascript"` ကို တွေ့ရတတ်ပေမယ့် modern HTML မှာ JavaScript က default ဖြစ်တာကြောင့် `type` attribute ကို မထည့်လည်း ရပါတယ်။ ပုံမှန် အားဖြင့် `script` တွေကို `<head>` tag ကြားမှာရေးပေမယ့် ကျွန်တော်တို့တွေ `body` မှာ လည်း အသုံးပြုနိုင်ပါတယ်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>JavaScript</title>
</head>
<body>
  <script type="text/javascript">
    document.write("Screen width : " + screen.width);
    document.write("<br>");
    document.write("Screen height : " + screen.height);
  </script>
</body>
</html>

```

အဲဒီ code ဟာ `screen` ရဲ့ `width` နဲ့ `height` ကို `webpage` မှာ ရိုက်ထုတ်ပြထားတာပါ။

Screen width : 1920
Screen height : 1080

Figure 10-2

ViewSource မှာ ဆိုရင်တော့ ကျွန်တော်တို့ ရေးထားသည့် code အတိုင်းပဲ မြင်ရပါလိမ့်မယ်။ Chrome သို့မဟုတ် firefox ကို အသုံးပြုရင် Browser ပေါ်မှာ right click နှိပ်ပြီး Inspect Element ကို နှိပ်လိုက်ပါ။

Screen width : 1920
Screen height : 1080

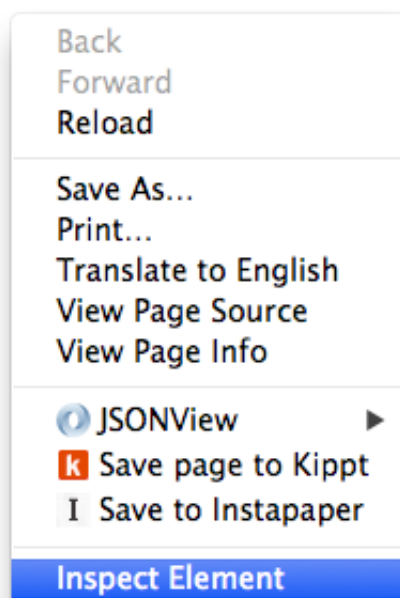


Figure 10-4

Elements ထဲမှာ `document.write` နဲ့ရေးထားတာ မဟုတ်ပဲ HTML code တွေကို တွေ့ရပါလိမ့်မယ်။

```

<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <script type="text/javascript">...</script>
      "Screen width : 1920"
      <br>
      "Screen height : 1080"
    "
  </body>
</html>

```

Figure 10-4

Link to a JavaScript File

JavaScript ကို ကျွန်တော်တို့ တွေ သီးသန့် ခွဲရေးပြီးတော့ HTML ထဲမှာ ပြန်ပေါင်းထည့်လို့လည်း ရပါတယ်။ အရင်ဆုံး ကျွန်တော်တို့တွေ browser.js file ကို ဖန်တီးပါမယ်။

browser.js မှာ

```

function browserDetails()
{
  document.write("Browser code name : " + navigator.appCodeName + "<br>");
  document.write("Browser name : " + navigator.appName + "<br>");
  document.write("Browser version : " + navigator.appVersion + "<br>");
  document.write("Browser platform : " + navigator.platform + "<br>");
  document.write("Browser details : " + navigator.userAgent + "<br>");
}

```

ကျွန်တော်တို့ browser.js file နေရာနဲ့ အတူတူ index.html တစ်ခု ဖန်တီးပါမယ်။

index.html ထဲမှာ

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>JavaScript</title>
  <script type="text/javascript" src="browser.js"></script>
</head>
<body>
  <script type="text/javascript">
    browserDetails();
  </script>
</body>
</html>

```

ကျွန်တော်တို့ အနေနဲ့ HTML ကနေ ပြင်ပ JavaScript ကို ခေါ်ချင်ရင် src နဲ့မှာ ပတ်လမ်းကြောင်းထည့်ပေးရပါတယ်။

src နဲ့ JavaScript ကို ခေါ်ပြီးတဲ့အခါမှာတော့ JavaScript file ထဲက function တွေကို ကျွန်တော်တို့ တိုက်ရိုက် ခေါ်လို့ရပါပြီ။ body မှာ ကျွန်တော် JavaScript file ထဲမှာ ရေးထားတဲ့ browserDetails() function ကို ခေါ် လိုက်ပါတယ်။

```

Browser code name : Mozilla
Browser name : Netscape
Browser version : 5.0 (Macintosh; Intel Mac OS X 10_9_3)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/35.0.1916.114 Safari/537.36
Browser platform : MacIntel
Browser details : Mozilla/5.0 (Macintosh; Intel Mac OS X
10_9_3) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/35.0.1916.114 Safari/537.36

```

Figure 10-5

navigator ဟာ browser နဲ့ ပတ်သက်တဲ့ အချက်အလက်တွေ ကို ရယူနိုင်ပါတယ်။ browserDetails function ဟာ browser information တွေကို document.write နဲ့ web page မှာ ရိုက်ထုတ်ထားပါတယ်။

Insert the Current Date and Time

ကျွန်တော်တို့တွေ JavaScript ကို အသုံးပြုပြီးတော့ web page မှာ အချိန်တွေ ထည့်လို့ရပါတယ်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>JavaScript</title>
</head>
<body>
  <h2>Current Date and Time</h2>
  <time>
    <script type="text/javascript">
      document.write(Date());
    </script>
  </time>
</body>
</html>

```

ကျွန်တော်တို့ Date() ဆိုတဲ့ function ကို ခေါ်ထားပြီးတော့ ရလာတဲ့ ရလဒ်ကို document.write နဲ့ body မှာ ရိုက်ပြထားပါတယ်။ Page ကို refresh လုပ်လိုက်တုန်း second ပါ လိုက်ပြောင်းသွားတာကို တွေ့ရပါလိမ့်မယ်။ Date() ဟာ အခု လက်ရှိ computer မှာ သတ်မှတ်ထားတဲ့ အချိန်ကို ဖော်ပြပေးပါတယ်။

Current Date and Time

Fri May 30 2014 00:30:09 GMT+0800 (SGT)

Figure 10-6

Display an Alert Message Box

JavaScript ကို အသုံးပြုပြီးတော့ ကျွန်တော်တို့တွေ web page တွေမှာ alert message box ဖော်ပြလိုရပါတယ်။ alert message ဟာ အသုံးပြုသူကို အရေးကြီးတဲ့ အချက်အလက်တွေ ဖော်ပြတဲ့ အနေနဲ့ အသုံးပြုကြပါတယ်။ ဥပမာ။။။ form ကို submit လုပ်လိုက်တဲ့ အခါမှာ data တွေ ထည့်ထားတာမှားနေရင် မှားနေကြောင်းကို alert message box နဲ့ ဖော်ပြပါတယ်။ အခု ကျွန်တော်တို့တွေ page တက်တက်ခြင်းမှာ alert message box တစ်ခု တက်လာအောင် ဖန်တီးကြည့်ရအောင်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>JavaScript</title>
</head>
<body to our new site!')">
  <h2>Alert!</h2>
</body>
</html>
```

ဒီ code လေးဟာ body ရဲ့ onload event မှာ ရေးထားပါတယ်။ ဒါကြောင့် page load တက်လာတာနဲ့ alert function ကို သွားခေါ်ပါမယ်။ alert message မှာ Welcome to our new site! လို့ရေးသားထားတဲ့အတွက် Welcome to our new site! ဆိုတဲ့ စာလေး တက်လာတာကို တွေ့ရပါမယ်။

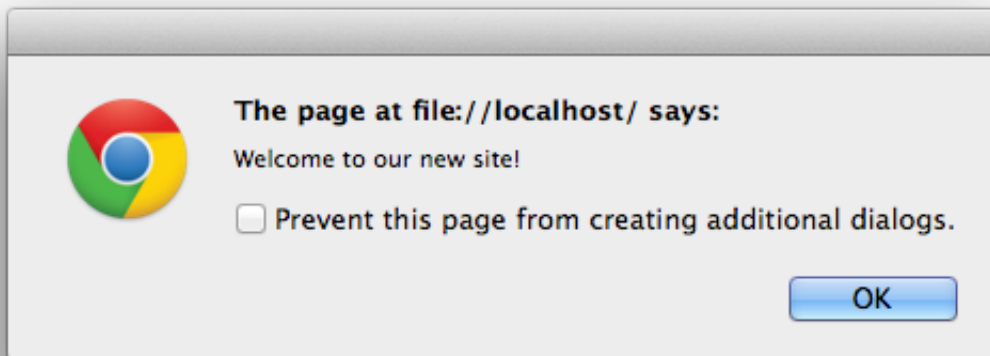


Figure 10-7

Display a Pop-Up Window

JavaScript ကို အသုံးပြုပြီးတော့ pop-up window ကို web page မှာ ဖော်ပြနိုင်ပါတယ်။ Pop-up windows ဟာ နောက်ဆုံး သတင်းတွေကို ဖော်ပြရမှာ အသုံးပြုနိုင်ပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>JavaScript</title>
</head>
<body>
  <a href="#" this first</a>
```

```
</body>
</html>
```

အဲဒီ code မှာ link ကိုနှိပ်လိုက်တာနဲ့ pop up windows လေး တက်လာပါလိမ့်မယ်။ href attribute ကို # လို့ ရေးထားတာကတော့ link ကို ဘာနဲ့ မှ မချိတ်ထားတာပါ။ onclick လုပ်လိုက်တဲ့ အခါမှာတော့ popup windows ခေါ်ပါတယ်။

```
window.open('readme.html','Readme','width=350,height=180');
```

readme.html ကတော့ ကိုယ် ဖွင့်ချင်တဲ့ file ပါ။ ဒုတိယ parameter ကတော့ windows name ပါ။ ပြီးတော့ တတိယ parameter မှာ width နဲ့ height ကို ပေးထားပါတယ်။ တက်လာမယ့် popup windows ရဲ့ width နဲ့ height ပါ။

[Read this first](#)



Figure 10-8

တကယ်လို့ popup windows ကို screen ပေါ်မှာ နေရာ အတိအကျနဲ့ ပေါ်ချင်ရင်တော့ top နဲ့ left ကို အသုံးပြုနိုင်ပါတယ်။

```
window.open('readme.html','Readme','width=350,height=180,left=150,top=200');
```

အထက်ပါ code အရဆိုရင်တော့ ဘယ်ဘက်ကနေ အကွာ ၁၅၀ pixel နဲ့ အပေါ်ကနေ အကွာ ၂၀၀ pixel မှာ ပေါ်ပါလိမ့်မယ်။

Create an Image Rollover Effect

JavaScript မှာ mouse hover event ပါပါတယ်။ ဒါကို အသုံးပြုပြီးတော့ image ပေါ်ကို ပုံတင်လိုက်တဲ့ အခါ နောက်ပုံ တစ်ခု ပြောင်းသွားအောင် ဖန်တီးနိုင်ပါတယ်။

Rollover image ကို ဖန်တီးဖို့ အတွက် mouse event handlers ၂ခု ကို အသုံးပြုပါတယ်။ image အရွယ်အစား တူသည့် image ၂ ခု လိုပါမယ်။

```
<!DOCTYPE html>
<html>
```

```

<head>
  <meta charset="UTF-8">
  <title>JavaScript</title>
</head>
<body>

</html>

```

အဲဒီ code မှာ mouse ကို ပုံပေါ်တင်လိုက်တာနဲ့ src ကို house.jpg ပြောင်းလိုက်ပါတယ်။ mouse က ပြန်ထွက်သွားရင် မူရင်းပုံ house_bw.jpg ကို ပြန်ပြောင်းထားပါတယ်။



Figure 10-9. house_bw.jpg



Figure 10-10. house.jpg

Show a Hidden Element

ကျွန်တော်တို့တွေ web site တွေမှာ link တစ်ခုကို ဒါမှမဟုတ် button ကို click နှိပ်လိုက်မှ သက်ဆိုင်ရာ စာတွေ ကို အောက်မှာ ပေါ်လာတာမျိုးတွေကို ဖန်တီးလို့ရပါတယ်။ ပထမဆုံး သက်ဆိုင်ရာ element ကို css က display property မှာ none လုပ်ထားပါမယ်။ ပြီးမှ click နှိပ်လိုက်မှ ပြန်ပြပေးပါမယ်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>JavaScript</title>

  <script>
    function showmore()
    {
      document.getElementById('more').style.display='block';

```

```

    document.getElementById('more_link').style.display='none';
}

function hide()
{
    document.getElementById('more').style.display='none';
    document.getElementById('more_link').style.display='block';
}
</script>
</head>
<body>
<p>
    One Piece is a Japanese manga series written and illustrated by Eiichiro Oda. It has
    been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are
    being published in tankōbon volumes by Shueisha, with the first released on December 24,
    1997, and the 73rd volume released as of March 2014.
</p>

<a href="#" id="more_link">Show more</a>
<div style="display:none" id="more">
    <p>One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains
    the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew
    of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the
    world's ultimate treasure known as One Piece in order to become the next Pirate King.</p>

    <a href="#"
</div>
</body>
</html>

```

အဲဒီ code လေးမှာ show more ဆိုတာကို နှိပ်လိုက်ရင် ဒုတိယ စာပိုဒ် ပေါ်လာပါလိမ့်မယ်။ အောက်ဆုံးမှာ Hide ဆိုတာ လေးကို နှိပ်လိုက်ရင်တော့ ဒုတိယ စာပိုဒ် ပြန်ပျောက်သွားတာကို တွေ့ရပါလိမ့်မယ်။ document.getElementById နဲ့ paragram ရှိတဲ့ <div> ရဲ့ css ကို ပြင်လိုက်ပါတယ်။ ပြီးတော့ showmore ဖြစ်နေရင် div layer ကို ဖော်ပြပြီးတော့ id='more_link' ကို hidden ပြန်လုပ်ထားတာကို တွေ့ရပါလိမ့်မယ်။

One Piece is a Japanese manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 73rd volume released as of March 2014.

[Show more](#)

Figure 10-11. Show more link

One Piece is a Japanese manga series written and illustrated by Eiichiro Oda. It has been serialized in Weekly Shōnen Jump since August 4, 1997; the individual chapters are being published in tankōbon volumes by Shueisha, with the first released on December 24, 1997, and the 73rd volume released as of March 2014.

One Piece follows the adventures of Monkey D. Luffy, a young boy whose body gains the properties of rubber after unintentionally eating a Devil Fruit, and his diverse crew of pirates, named the Straw Hat Pirates. Luffy explores the ocean in search of the world's ultimate treasure known as One Piece in order to become the next Pirate King.

[Hide](#)

Figure 10-12

Change Page Content

ကျွန်တော်တို့တွေ JavaScript နဲ့ DOM object ကို ရယူပြီးတော့ အထဲမှာ ရှိတဲ့ content တွေကိုလည်း ပြောင်းလဲ နိုင်ပါတယ်။ ကျွန်တော်တို့တွေ id နဲ့ DOM ကို ရယူပါမယ်။ ပြီးတော့ innerHTML method ကို သုံးပြီး အထဲက HTML code ကို ပြောင်းလဲနိုင်ပါတယ်။ အောက်မှာ ဥပမာ code လေးကြည့်ရအောင်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>JavaScript</title>
  <script>
    function updateText()
    {
      document.getElementById('content').innerHTML =
document.getElementById('update_text').value;
    }
  </script>
</head>
<body>

  <section>
    <input type="text" value="" id="update_text">
    <input type="button" value="Update"
  </section>

  <section id="content">
    Hello World!
  </section>
</body>
</html>
```

ဒီ code လေးမှာ text box ထဲကို စာရိုက်ထည့်ပြီး update လုပ်လိုက်တဲ့ အခါ Hello World! ဆိုတဲ့စာ အစား text box က စာ ပြောင်းသွားတာကို တွေ့ရပါလိမ့်မယ်။

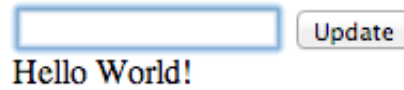


Figure 10-13

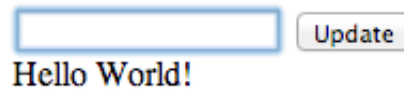


Figure 10-14

Text box ထဲက data ကို လိုချင်ရင် value property နဲ့ ဖတ်ရပါတယ်။ div , section စတဲ့ element ထဲက HTML data တွေကို ပြောင်းချင်ရင်တော့ innerHTML property ကို update လုပ်ပေးရပါတယ်။

Display a Calculation

Modern HTML မှာ <output> ဆိုတဲ့ tag ပါပါတယ်။ <output> tag ကို အသုံးပြုပြီးတော့ calculation တွေရဲ့ ရလဒ်ကို ဖော်ပြရမှာ အသုံးပြုနိုင်ပါတယ်။ <output> tag ရဲ့ ပုံစံကို ပြင်ချင်ရင်တော့ CSS ကို အသုံးပြုဖို့လိုပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>JavaScript</title>
</head>
<body>
  <h2>Ticket Price</h2>
  <form name="ticket" adult.valueAsNumber * 10 + child.valueAsNumber * 8">
    Adult : <input type="number" step="1" value="0" name="adult"><br/>
    Children : <input type="number" step="1" value="0" name="child"><br/>
    Total : $<output name="total">0</output>
  </form>
</body>
</html>
```

ဒီ code မှာ ဆိုရင် textbox မှာ စာရိုက်တာနဲ့ oninput event ကို ရောက်သွားပြီးတော့ output မှာ ရလဒ်ကို ထုတ်ပြပေးမှာ ဖြစ်ပါတယ်။ ကျွန်တော်တို့တွေဟာ form မှာ JavaScript ရေးထားတာကြောင့် name နဲ့ တိုက်ရိုက်ခေါ်လိုရပါတယ်။

ကျွန်တော်တို့ function ခွဲပြီး ရေးရင်တော့ input နဲ့ output tag ကို ခေါ်ဖို့အတွက် form နာမည်နဲ့ ခေါ်ဖို့လိုပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>JavaScript</title>
  <script>
    function calculate()
    {
      ticket.total.value= ticket.adult.valueAsNumber * 10 + ticket.child.valueAsNumber * 8;
    }
  </script>
</head>
<body>
```

```

<h2>Ticket Price</h2>
<form name="ticket">
  Adult : <input type="number" step="1" value="0" name="adult"><br/>
  Children : <input type="number" step="1" value="0" name="child"><br/>
  Total : $<output name="total">0</output>
</form>
</body>
</html>

```

အထက်ပါ code မှာ calculate() ဆိုတဲ့ function ကို ခေါ်ထားပြီးတော့

```
ticket.total.value= ticket.adult.valueAsNumber * 10 + ticket.child.valueAsNumber * 8;
```

လို့ ရေးထားပါတယ်။ ticket ဆိုတာကတော့ form name ဖြစ်ပါတယ်။

အခုလောက်ဆိုရင်တော့ JavaScript နဲ့ ပတ်သက်ပြီး ဘာတွေ လုပ်လို့ရလဲ ဘယ်လိုရေးလို့ရသလဲ ဆိုတာကို သိလောက်ပါပြီ။
နောက်အခန်းမှာတော့ canvases ကို စပါမယ်။

Drawing with the Canvas Element

ပုံတွေ လိုင်းတွေကို web page ပေါ်မှာ ဆွဲချင်သလား ? Modern HTML မှာ canvas ပါဝင်ပါတယ်။ canvas က image editors အနေနဲ့ အသုံးပြုလို့ရသလို ပုံတွေကိုလည်း JavaScript အသုံးပြုပြီး ဆွဲနိုင်ပါတယ်။

Understanding Canvases

Modern HTML မှာ `<canvas>` tag ဟာ web page ပေါ်မှာ ပုံဆွဲဖို့အတွက် နေရာတစ်ခု ဖန်တီးပေးပါတယ်။ အဲဒီ အပေါ်မှာ shapes, lines တွေ အပြင် စာတွေ ပုံတွေ ထည့်သွင်းလို့ရပါတယ်။

Canvas ကို JavaScript နဲ့ တွဲပြီး အသုံးပြုရပါတယ်။ နောက်ပြီးတော့ coordinates အကြောင်းကိုလည်း အနည်းငယ် နားလည်ထားမှသာ လိုချင်တဲ့ နေရာမှာ ပုံပေါ်အောင် ဆွဲလို့ရပါလိမ့်မယ်။

`<canvas>` tag ကို အသုံးပြုတဲ့အခါမှာ `width` နဲ့ `height` attributes ကို လည်း ကြေငြာခဲ့ဖို့လိုပါတယ်။ နောက်ပြီး `id` attribute ကို ထည့်သွင်းဖို့လိုပါတယ်။ ဒါမှ JavaScript ကနေ canvas ပေါ်မှာ ပုံဆွဲလို့ ရပါလိမ့်မယ်။

JavaScript လိုအပ်တာကြောင့် ပြီးခဲ့တဲ့ အခန်းက ပြောထားခဲ့သလို `<script>` tag အသုံးပြုဖို့လိုပါတယ်။ ဒါမှမဟုတ် external .js file မှာ ဖြစ်ဖြစ် code ရေးပြီးတော့ canvas ပေါ်မှာ ပုံဆွဲလို့ရပါတယ်။ JavaScript ကနေ canvas ကို ပြီးခဲ့တဲ့ chapter မှာ အသုံးပြုခဲ့သလို `getElementById` နဲ့ ပဲ ခေါ်သုံးရပါလိမ့်မယ်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Canvas</title>
  <style>
    canvas#sample {border: 1px solid gray;}
  </style>
</head>
<body>
  <canvas id="sample" width="600" height="400"></canvas>
  <script>
    var cnv = document.getElementById('sample');
    var ctx = cnv.getContext('2d');
    ctx.strokeStyle = 'green';
    ctx.lineWidth = '15';
    ctx.strokeRect(160,80,360,240);
  </script>
</body>
</html>

```

canvas အတွက် အထက်ပါ code လေးကို လေ့လာကြည့်ရအောင်။ ကျွန်တော်တို့တွေ HTML မှာ canvas တစ်ခု ဆောက်ထားပါတယ်။ width ကတော့ 600px နဲ့ height ကတော့ 400px ပါ။ CSS နဲ့ canvas ကို ခဲရောင် ဘောင်ခတ်ထားပါတယ်။ JavaScript မှာတော့ `cnv` ထဲကို `document.getElementById('sample');` ကို အသုံးပြုပြီး canvas object ကို ယူလိုက်ပါတယ်။ canvas object ဖြစ်တဲ့ `cnv` ကနေ `cnv.getContext('2d');` နဲ့ context ကို ရလာပါတယ်။ ကျွန်တော်တို့ context ရလာပြီးတဲ့အခါမှာတော့ ဘာ ဆွဲမလဲ ။ ဘယ်လို ဆွဲမလဲ စတာတွေကို ထည့်သွင်းရပါတယ်။ အခု code မှာတော့ အစိမ်းရောင် စတုဂံကို x position 160 နဲ့ y position 80 မှာ ဆွဲမယ်လို့ ဆိုထားပါတယ်။ width ကတော့ 360 နဲ့ height ကတော့ 240 ပါ။ အဲဒီ code ဟာ browser မှာ ဆိုရင် အောက်ကလို အစိမ်းရောင် စတုဂံကို မြင်ရပါလိမ့်မယ်။

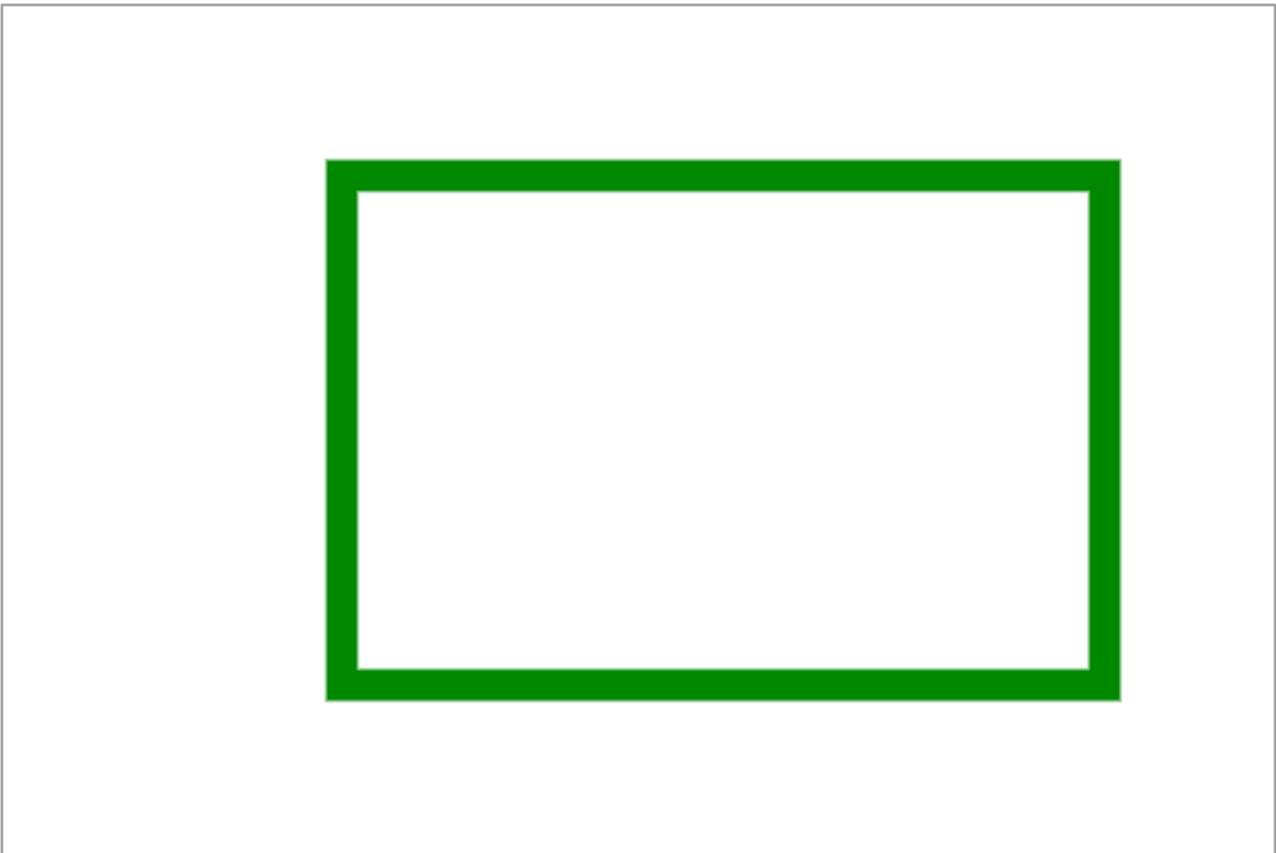


Figure 11-1

Set Up a Canvas

ကျွန်တော်တို့ canvas ဖန်တီးဖို့အတွက် HTML မှာ `<canvas>` tag ကို ထည့်ပါမယ်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Canvas</title>
</head>
<body>

```

```

<canvas></canvas>
</body>
</html>

```

ဒီ code ကို browser မှာ ဖွင့်ရင်တော့ ဘာမှ မြင်ရမှာ မဟုတ်ပါဘူး။ ကျွန်တော်တို့ canvas ကို မြင်ရအောင် canvas ကို css သုံးပြီး border ထည့်ပါမယ်။

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Canvas</title>
<style>
canvas#sample {border: 1px solid gray;}
</style>
</head>
<body>
<canvas id="sample" width="600" height="400"></canvas>
</body>
</html>

```

ကျွန်တော်တို့ border ထည့်ထားတာကြောင့် အဖြူရောင် page မဟုတ်တော့ပဲ canvas ရှိတဲ့ နေရာမှာ ဘောင်ခတ်ထားတာကို တွေ့နိုင်ပါတယ်။

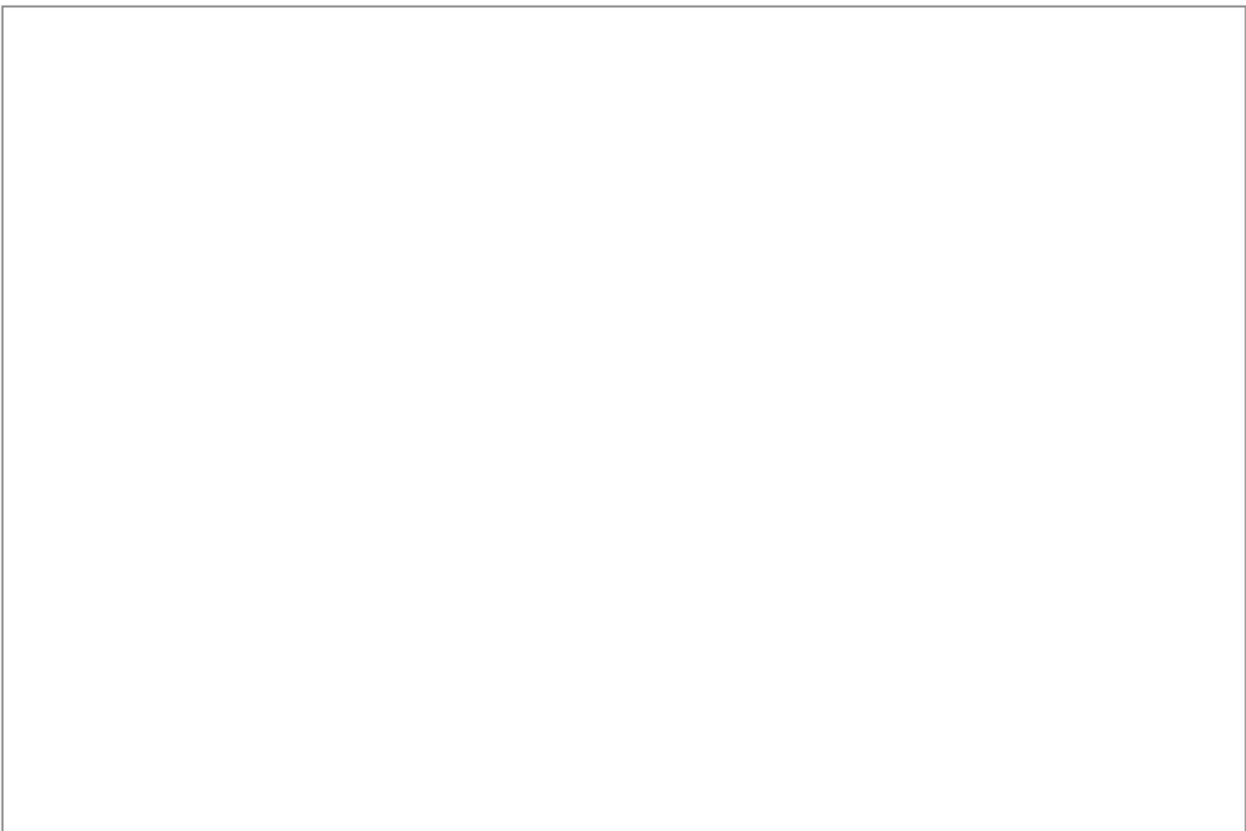


Figure 11-2

ကျွန်တော်တို့ canvas ဖန်တီးပြီးသွားရင်တော့ canvas ပေါ်မှာ ပုံတွေ စပြီး ဆွဲလို့ရပါပြီ။

Draw Rectangles

ကျွန်တော်တို့တွေ အခု စတုဂံ ပုံလေး တစ်ခုဆွဲကြည့်ရအောင်။ စတုဂံ ပုံဆွဲတဲ့ အခါမှာ ကျွန်တော်တို့ တွေ အနေနဲ့ context တစ်ခု ဖန်တီးဖို့ လိုပါတယ်။ ပြီးတဲ့အခါ ပုံဆွဲဖို့အတွက် x နဲ့ y ကို သိဖို့ လိုပါတယ်။ x ဆိုတာကတော့ ဘယ်ဘက်ကနေ အကွာ အဝေးပါ။ y ကတော့ ထိပ် ကနေ အကွာ အဝေးပါ။ x,y က 0,0 ဆိုရင်တော့ ဘယ်ဘက် အပေါ်ထိပ်ဆုံးကို ပေါ်တာပါ။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Canvas</title>
  <style>
    canvas#sample {border: 1px solid gray;}
  </style>
</head>
<body>
  <canvas id="sample" width="600" height="400"></canvas>

  <script>
    var cnv = document.getElementById('sample');
    var ctx = cnv.getContext('2d');
    ctx.fillStyle = 'orange';
    ctx.fillRect(10,10,200,100);
  </script>
</body>
</html>
```

ဒီ code မှာဆိုရင်တော့ x က 10 နဲ့ y 10 ကို အသုံးပြုထားပါတယ်။ width ကတော့ ၂၀၀ ဖြစ်ပြီး တော့ height ကတော့ ၁၀၀ ကို အသုံးပြုထားတယ်။

```
ctx.fillStyle = 'orange';
ctx.fillRect(10,10,200,100);
```

fill style ကတော့ လိမ္မော်ရောင်ကို အသုံးပြုထားပြီးတော့ fillRect နဲ့ လိမ္မော်ရောင် ခြယ်ထားတဲ့ စတုဂံတစ်ခုကို ဖန်တီးထားပါတယ်။



Figure 11-3

Stock ဆွဲချင်ရင်တော့ `strokeStyle` , `lineWidth` လိုပါတယ်။ ပြီးရင် `strokeRect` ဆွဲရပါတယ်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Canvas</title>
  <style>
    canvas#sample {border: 1px solid gray;}
  </style>
</head>
<body>
  <canvas id="sample" width="600" height="400"></canvas>
  <script>
    var cnv = document.getElementById('sample');
    var ctx = cnv.getContext('2d');
    ctx.fillStyle = 'orange';
    ctx.fillRect(10,10,200,100);
    ctx.strokeStyle = 'blue';
    ctx.lineWidth = '10';
    ctx.strokeRect(360,120,200,240);
  </script>
</body>
</html>

```

အခု code မှာ ဆိုရင် ကျွန်တော်တို့တွေ stroke တစ်ခုနဲ့ စတုဂံ ဆွဲထားပါတယ်။

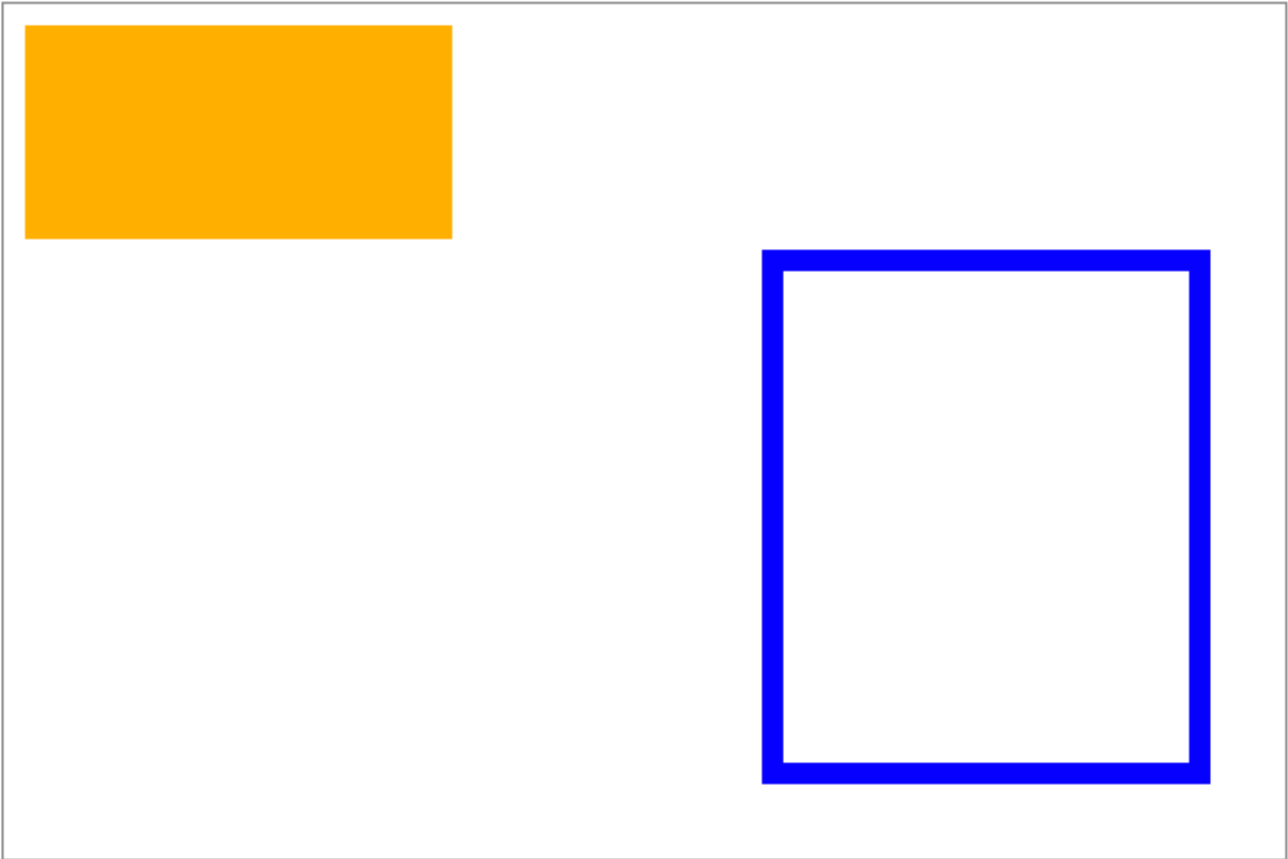


Figure 11-4

တကယ်လို့ စတုဂံမှာ အပေါက်လုပ်စေချင်ရင်တော့ `clearRect` ကို အသုံးပြုလို့ရပါတယ်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Canvas</title>
  <style>
    canvas#sample {border: 1px solid gray;}
  </style>
</head>
<body>
  <canvas id="sample" width="600" height="400"></canvas>
  <script>
    var cnv = document.getElementById('sample');
    var ctx = cnv.getContext('2d');
    ctx.fillStyle = 'orange';
    ctx.fillRect(10,10,200,100);
    ctx.strokeStyle = 'blue';
    ctx.lineWidth = '10';
    ctx.strokeRect(360,120,200,240);
    ctx.clearRect (30,20,150,50);
  </script>
</body>
</html>

```

အခု code မှာ ဆိုရင် x 30 နဲ့ y 20 မှာ width 150 , height 50 ရှိတဲ့ အပေါက် ဖြစ်နေတာကို တွေ့ရပါမယ်။

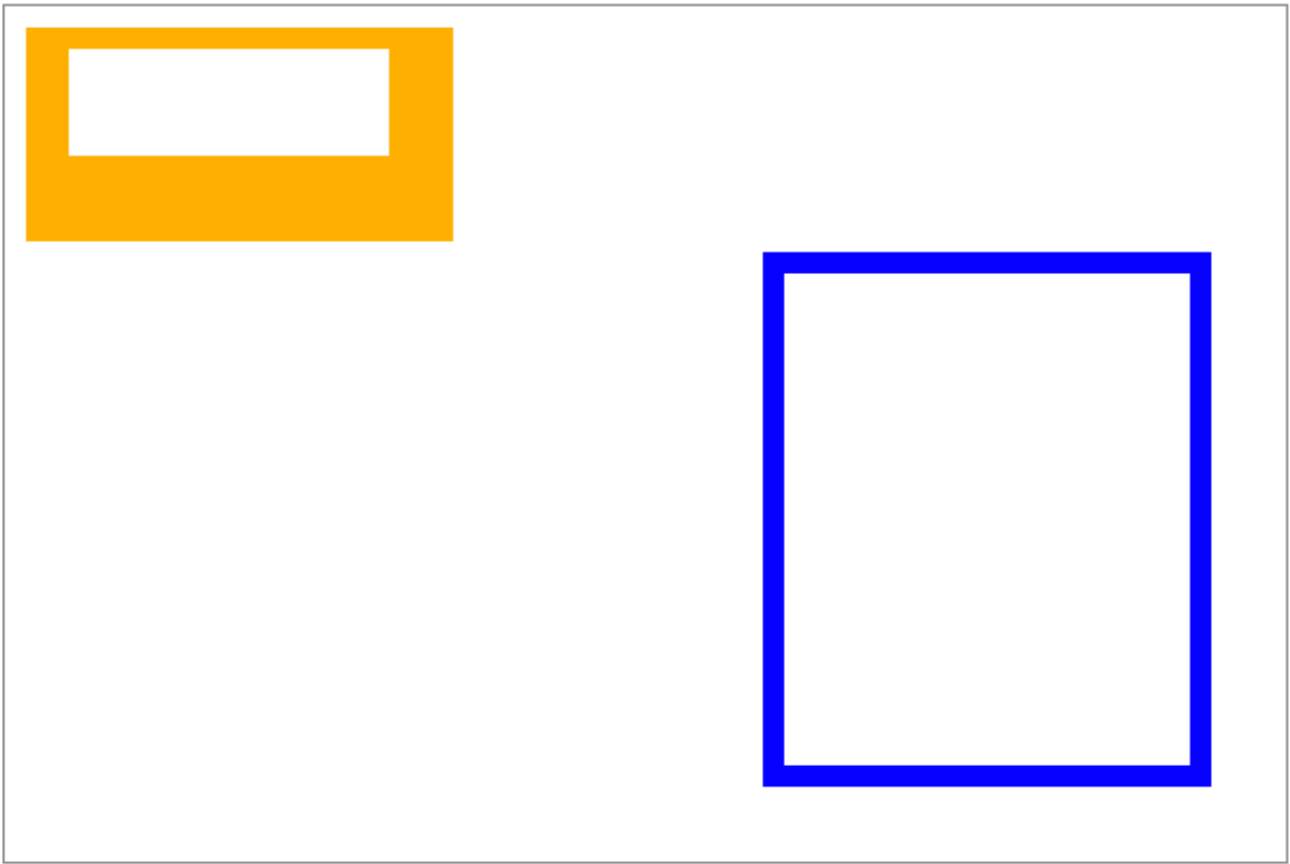


Figure 11-5

Draw Circles

ကျွန်တော်တို့ canvas မှာ စတုဂံသာမက စက်ဝိုင်းလည်း ဆွဲလို့ရပါတယ်။ စက်ဝိုင်း ရဲ့ coordinate ဟာ အလယ်ကနေ ဆွဲပါတယ်။ radius ကို အသုံးပြုပြီးတော့ စက်ဝိုင်းအရွယ်အစား ဖန်တီးရပါတယ်။ စက်ဝိုင်းဆွဲရာမှာ radians ကို ပဲ ဆွဲပေးပါတယ်။ ဒါကြောင့် စက်ဝိုင်းအပြည့်အတွက် ဆိုရင် 2 PI လိုပါတယ်။ စက်ဝိုင်း တစ်ဝက်သာဆိုရင်တော့ PI ကို အသုံးပြုရပါမယ်။

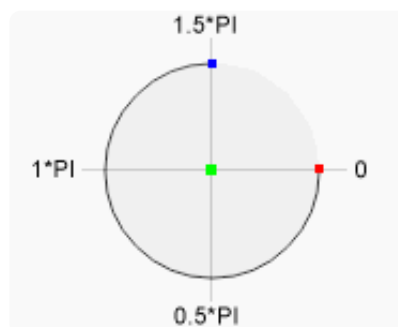


Figure 11-6

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Canvas</title>
    <style>
      canvas#myCanvas {border: 1px solid gray;}
    </style>
  </head>
  <body>
    <canvas id="myCanvas" width="578" height="200"></canvas>
    <script>
      var canvas = document.getElementById('myCanvas');
      var context = canvas.getContext('2d');
      var centerX = canvas.width / 2;
      var centerY = canvas.height / 2;
      var radius = 70;
      context.beginPath();
      context.arc(centerX, centerY, radius, 0, 2 * Math.PI, false);
      context.fillStyle = 'green';
      context.fill();
      context.lineWidth = 5;
      context.strokeStyle = '#FF0000';
      context.stroke();
    </script>
  </body>
</html>

```

ဒီ code မှာ အစိမ်းရောင် ခြယ်ထားပြီး ဘောင်ကို အနီရောင်ခြယ်ထားတဲ့ စက်ဝိုင်း တစ်ခုကို browser မှာ တွေ့ရပါလိမ့်မယ်။

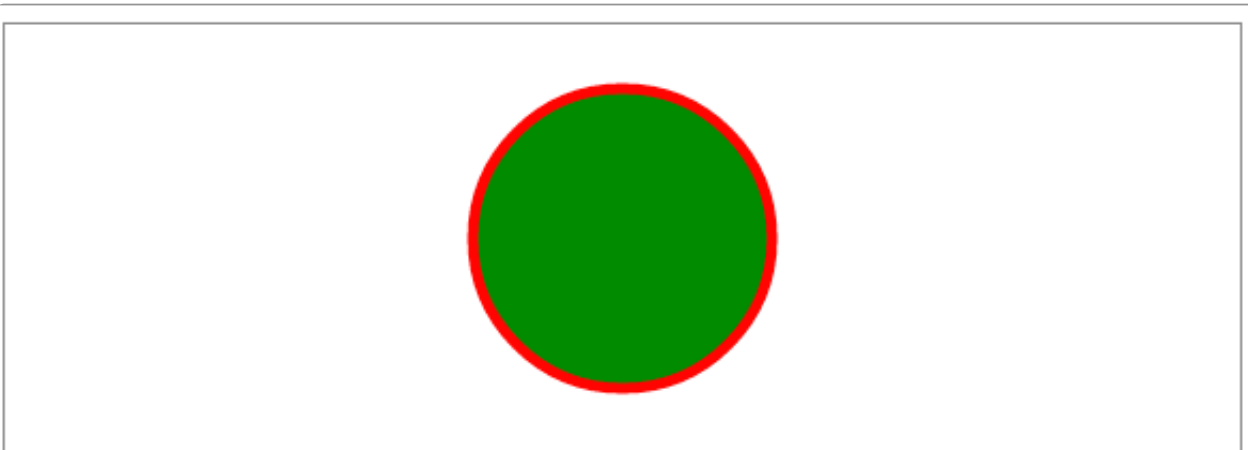


Figure 11-7

```
context.arc(centerX, centerY, radius, 0, 2 * Math.PI, false);
```

ဆိုတာကတော့

```
context.arc(x,y,radius,start_Angle,end_Angle,counterclockwise);
```

ပုံမှာ angle သည် က စထားပြီးတော့ စက်ဝိုင်းအပြည့်ဆွဲထားပါတယ်။ စက်ဝိုင်းကို နာရီလက်တံ အတိုင်း ဆွဲထားပါတယ်။ တကယ်လို့ နာရီလက်တံ ပြောင်းပြန်ဆွဲလိုရင်တော့ နောက်ဆုံး parameter false အစား true ပေးဖို့လိုပါတယ်။

တကယ်လို့ code ကို 2π အစား π ပဲ သုံးရင် စက်ဝိုင်းက တစ်ဝက်ပဲ ဖြစ်နေပါလိမ့်မယ်။

```
context.arc(centerX, centerY, radius, 0, Math.PI, false);
```

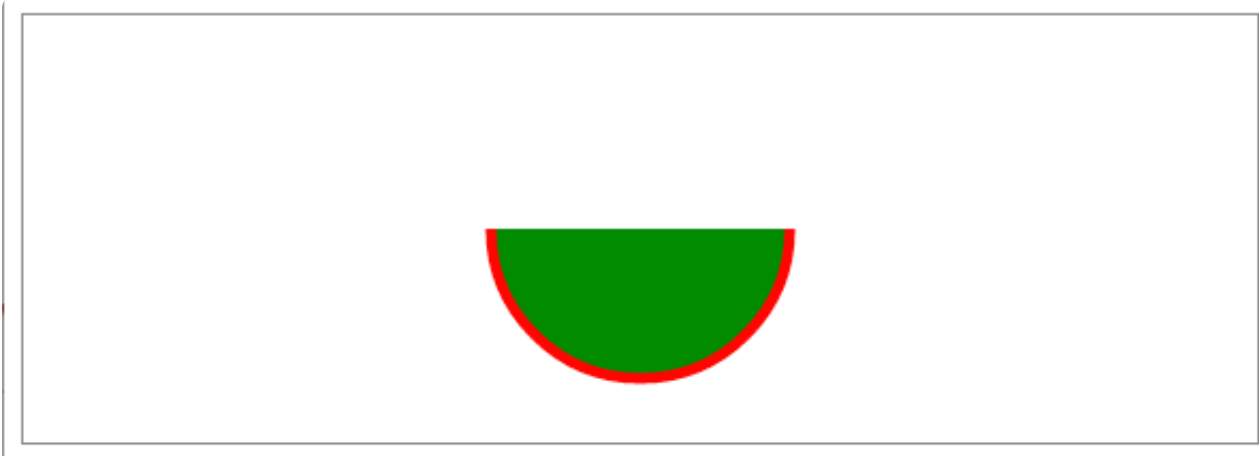


Figure 11-8

Draw Lines

မျဉ်းကြောင်းတွေ ဆွဲရင်တော့ စမှတ် အတွက် `moveTo` ကို အသုံးပြုပြီး ဆုံးမှတ်အတွက် `lineTo` ကို အသုံးပြုပါတယ်။ `x` နဲ့ `y` အမှတ်တွေကို parameter အဖြစ်ထည့်ပေးရပါမယ်။ ပြီးရင်တော့ `stroke()` နဲ့ မျဉ်းဆွဲလို့ရပါပြီ။

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Canvas</title>
    <style>
      canvas#myCanvas {border: 1px solid gray;}
    </style>
  </head>
  <body>
    <canvas id="myCanvas" width="578" height="500"></canvas>
    <script>
      var canvas = document.getElementById('myCanvas');
      var context = canvas.getContext('2d');
      context.lineWidth = 5;
      context.strokeStyle = 'orange';
      context.moveTo(50,50);
      context.lineTo(50,360);
      context.stroke();
    </script>
```

```

</body>
</html>

```

ဒီ code မှာဆိုရင် မျဉ်းကြောင်းကို x ၅၀ y ၅၀ ကနေ ပြီးတော့ x ၅၀ y ၃၆၀ အထိ ဆွဲထားပါတယ်။



Figure 11-9

ကျွန်တော်တို့ နောက်ထပ် မျဉ်း တစ်ကြောင်း ထပ်ဖြည့်ဆွဲကြည့်ရအောင်။

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Canvas</title>
    <style>
      canvas#myCanvas {border: 1px solid gray;}
    </style>
  </head>
  <body>
    <canvas id="myCanvas" width="578" height="500"></canvas>
    <script>
      var canvas = document.getElementById('myCanvas');

```

```

var context = canvas.getContext('2d');
context.lineWidth = 5;
context.strokeStyle = 'orange';
context.moveTo(50,50);
context.lineTo(50,360);
context.stroke();

context.moveTo(50,50);
context.lineTo(300,360);
context.stroke();

</script>
</body>
</html>

```

စမှတ် ကို ၅၀ , ၅၀ ပြန်ထားလိုက်တယ်။ ပြီးတော့ ၃၀၀ , ၃၆၀ ကို ဆွဲလိုက်တယ်။ ဒါကြောင့် ပုံ
မှာ ပြထားတဲ့ ပုံအတိုင်း မြင်ရပါလိမ့်မယ်။

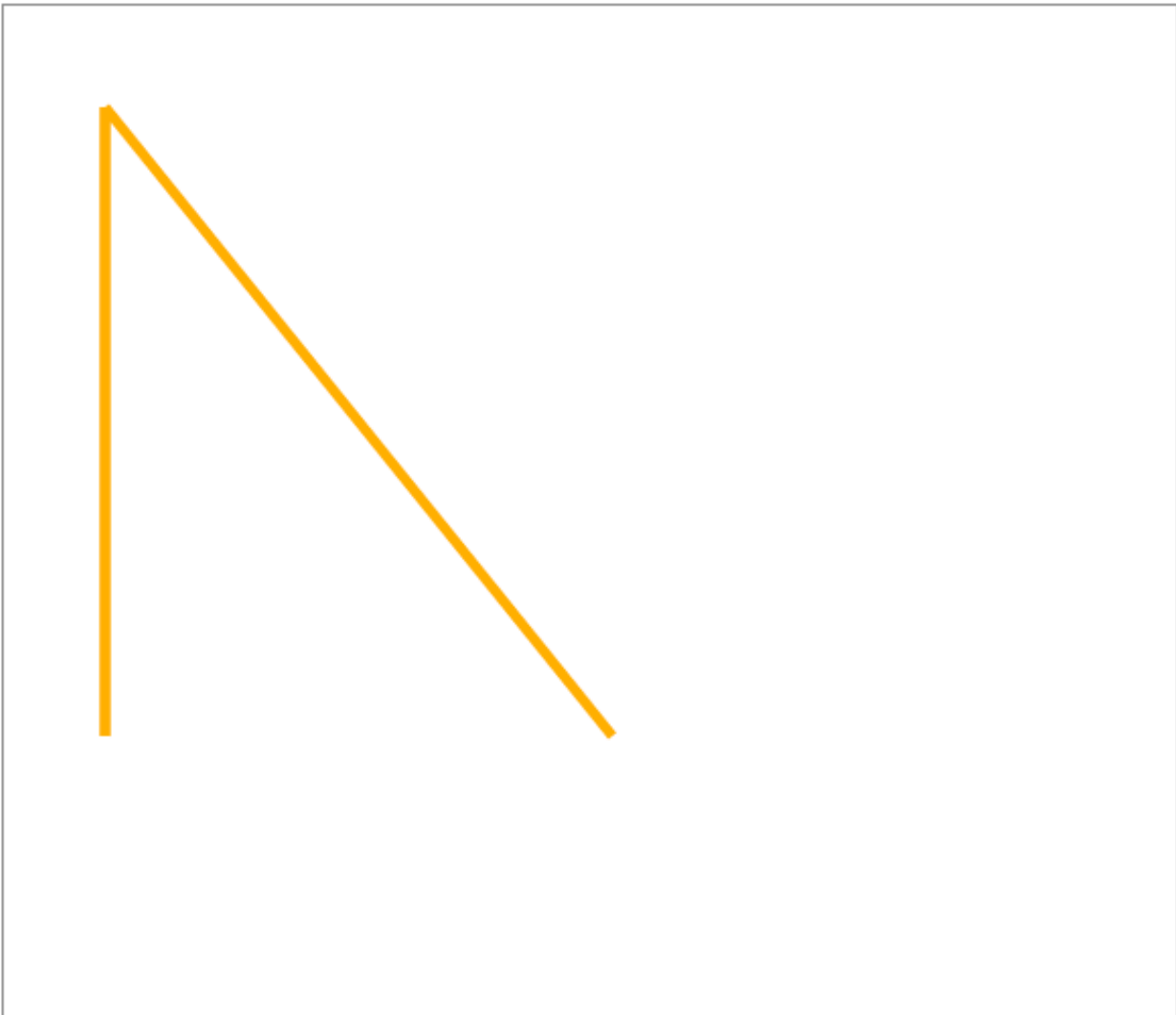


Figure 11-10

တစ်ကြောင်း ဆွဲလိုက် ပြန်ရွှေ့လိုက် မလိုက်ပဲနဲ့ပဲ တောက်လျှောက် မျဉ်းကြောင်း ဆွဲသွားလို့ရပါတယ်။

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Canvas</title>
    <style>
      canvas#myCanvas {border: 1px solid gray;}
    </style>
  </head>
  <body>
    <canvas id="myCanvas" width="578" height="500"></canvas>
    <script>
      var canvas = document.getElementById('myCanvas');
      var context = canvas.getContext('2d');
      context.lineWidth = 5;
      context.strokeStyle = 'orange';
      context.moveTo(50,50);
      context.lineTo(50,360);
      context.lineTo(300,360);
      context.lineTo(50,50);
      context.stroke();
    </script>
  </body>
</html>

```

ဒီ code မှာ ဆိုရင် ပထမ ၅၀ , ၅၀ က စတယ်။ ပြီးတော့ ၅၀ , ၃၆၀ ကို ရောက်တယ်။ အဲဒီ point ကနေ ၃၀၀ , ၃၆၀ ကို လိုင်း ထပ်ဆွဲတယ်။ ပြီးတော့ ၃၀၀ , ၃၆၀ အမှတ်ကနေ ၅၀ , ၅၀ ဆီ လိုင်း ထပ်ဆွဲတယ်။ ဒါကြောင့် အောက်ကလို ပုံ ပေါ်လာပါတယ်။

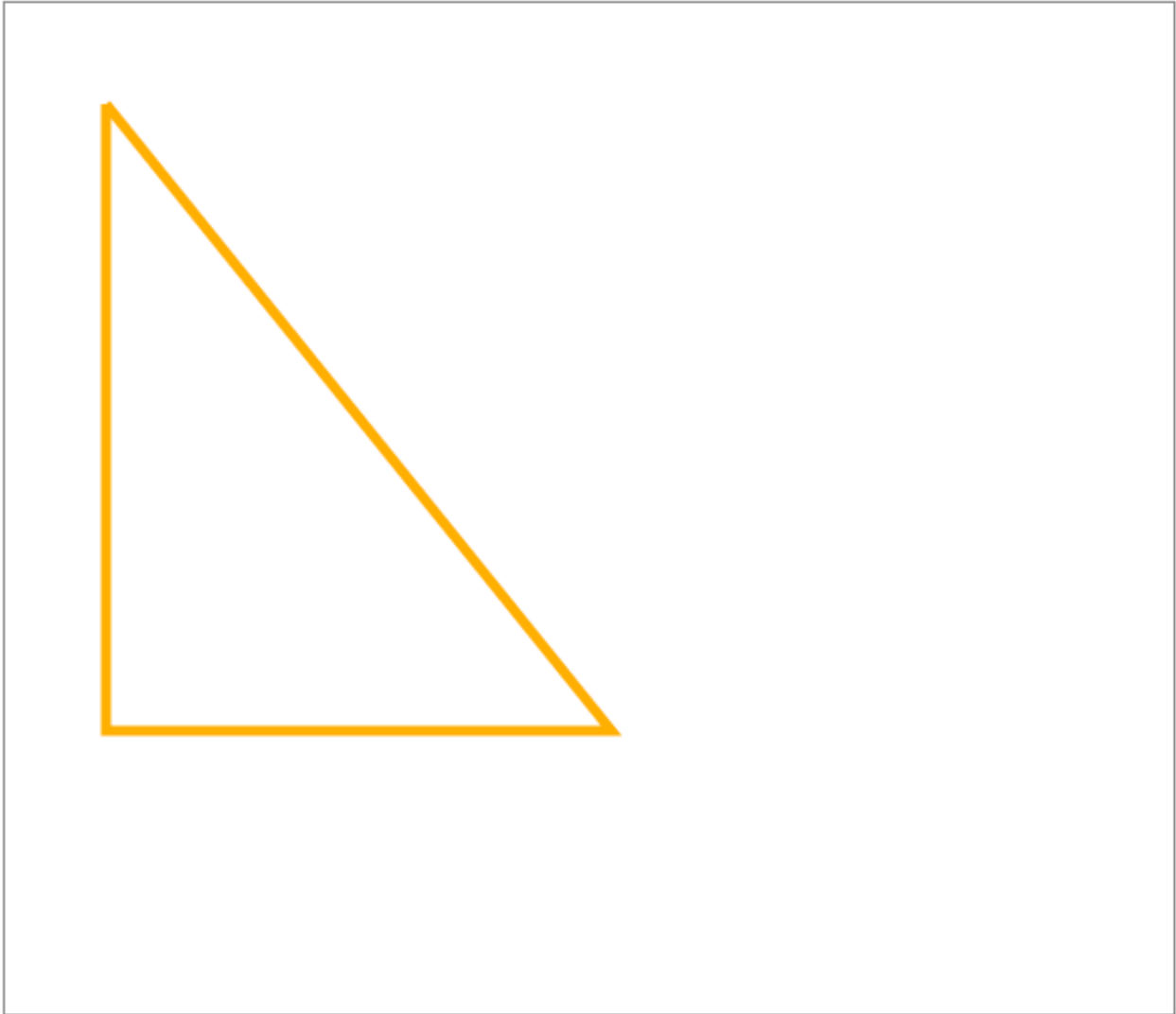


Figure 11-11

တကယ်လို့ ဖြတ်ထားတဲ့ ပုံကို ၂ ဆကြီးစေချင်ရင်တော့ နောက်ဆုံး width နဲ့ height မှာ သွားပြင် လို့ရပါတယ်။

ဥပမာ။

```
ctx.drawImage(pic,18,100,80,35,0,0,80 * 2,35 * 2);
```

Add Text

Canvas ဟာ ပုံဆွဲရုံသာမက စာတွေ ပါထည့်လို့ရပါတယ်။ စာထည့်ဖို့အတွက် font family , size စ တာတွေလိုပါတယ်။ ပြီးတဲ့ အခါမှာ fillText နဲ့ စာ ကို canvas ထဲကို ထည့်ရပါတယ်။

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
```

```

<title>Canvas</title>
<style>
canvas#myCanvas {border: 1px solid gray;}
</style>
</head>
<body>
<canvas id="myCanvas" width="300" height="300"></canvas>
<script>
var canvas = document.getElementById('myCanvas');
var ctx = canvas.getContext('2d');
//filled text
ctx.fillStyle = "red";
ctx.font = '40px "times new roman"'
ctx.fillText('Hello World',50,150);
</script>
</body>
</html>

```

code ကတော့ ရှင်းရှင်းလေးပါ။ စာလုံး အနီရောင် သုံးတယ်။ font size က 40px ဖြစ်ပြီးတော့ font family ကတော့ times new roman ပါ။ ပြီးတော့ Hello World ဆိုတဲ့ စာလုံးကို x ၅၀ ၊ y ၁၅၀ တို့ မှာ ထည့်လိုက်ပါတယ်။



Figure 11-12

ထပ်ပြီးတော့ color မဖြစ်ည့်ပဲ stroke နဲ့ စာထည့်ကြည့်ရအောင်။

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Canvas</title>
<style>
canvas#myCanvas {border: 1px solid gray;}
</style>

```

```

</head>
<body>
  <canvas id="myCanvas" width="300" height="300"></canvas>
  <script>
    var canvas = document.getElementById('myCanvas');
    var ctx = canvas.getContext('2d');
    //filled text
    ctx.fillStyle = "red";
    ctx.font = '40px "times new roman"'
    ctx.fillText('Hello World',50,150);

    //outline text
    ctx.strokeStyle = 'blue';
    ctx.font = '30px "arial"';
    ctx.strokeText('Stroke',50,200);
  </script>
</body>
</html>

```

အဲဒီ အခါ browser မှာ အောက်ကလို ပေါ်ပါလိမ့်မယ်။

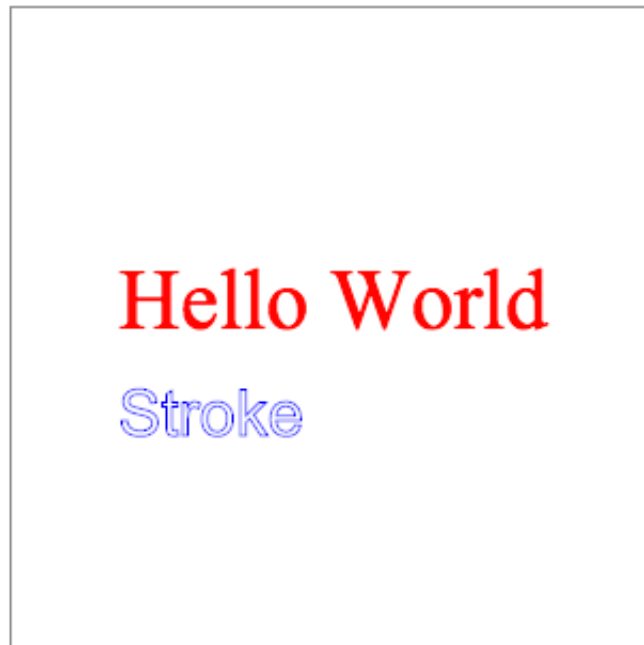


Figure 11-13

တကယ်လို့ လက်ရှိ text မှာ stroke ထည့်ချင်ရင် x နဲ့ y position ကို အတူတူပေးလိုက်ရင် ရပါ တယ်။

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Canvas</title>
    <style>
      canvas#myCanvas {border: 1px solid gray;}
    </style>
  </head>
  <body>

```

```

<canvas id="myCanvas" width="300" height="300"></canvas>
<script>
  var canvas = document.getElementById('myCanvas');
  var ctx = canvas.getContext('2d');
  //filled text
  ctx.fillStyle = "red";

  ctx.font = '40px "times new roman"'
  ctx.fillText('Hello World',50,150);

  //outline text
  ctx.strokeStyle = 'black';
  ctx.lineWidth = 2;
  ctx.strokeText('Hello World',50,150);
</script>
</body>
</html>

```

အခု code မှာဆိုရင် stroke width ကို 2 px ပေးထားပါတယ်။



Figure 11-14

တကယ်လို့ bold , italic စတာတွေ ထည့်ချင်ရင်တော့ font မှာ ထည့်လို့ရပါတယ်။

```
ctx.font = 'italic bold 44px "Times New Roman"');
```

textAlign ထည့်မယ်ဆိုရင်တော့

```
ctx.textAlign='center';
```

ဆိုပြီး ထည့်လို့ရပါတယ်။ textAlign တွေကတော့ left , center , right တို့ပဲ ဖြစ်ပါတယ်။

Add an Image

Canvas မှာ ကျွန်တော်တို့ ဓာတ်ပုံတွေပါ ထည့်လို့ရပါတယ်။ ဓာတ်ပုံနဲ့ canvas မှာ ဆွဲထားတဲ့ ပုံ တွေ ပြောင်းပြီး အသုံးပြုလို့ရပါတယ်။ image ကို အသုံးပြုရာမှာ အခြား canvas drawing နဲ့ မတူ တာကတော့ အရင်ဆုံး image ကို load လုပ်ရပါတယ်။ load ပြီးမှသာ canvas ပေါ်မှာ image ကို ဆွဲ လိုက်ပါတယ်။

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Canvas</title>
    <style>
      canvas#myCanvas {border: 1px solid gray;}
    </style>
  </head>
  <body>
    <canvas id="myCanvas" width="300" height="300"></canvas>
    <script>
      var canvas = document.getElementById('myCanvas');
      var ctx = canvas.getContext('2d');
      var pic = new Image();
      pic.src = 'house.jpg';
      pic.onload = function () {
        ctx.drawImage(pic,10,10);
      }
    </script>
  </body>
</html>
```

ဒီ code မှာ ဆိုရင် house.jpg ကို image object နဲ့ ခေါ်လိုက်ပါတယ်။ load ပြီးသွားတဲ့ အခါမှသာ `ctx.drawImage` နဲ့ x 10 | y 10 နေရာမှာ ဆွဲလိုက်ပါတယ်။



Figure 11-15

တကယ်လို့ ပုံ အရွယ်အစားကို ပြင်ချင်ရင်တော့ drawImage မှာ width နဲ့ height parameter ထပ် ဖြည့်လို့ရပါတယ်။

```
ctx.drawImage(pic,10,10,pic.width*0.5,pic.height*0.5);
```

Image object ရဲ့ အရွယ်အစားလိုချင်ရင်တော့ width နဲ့ height property ကနေ ရနိုင်ပါတယ်။ အခု ဥပမာဟာ ပုံမှန် image အရွယ်အစားရဲ့ တဝက်ကို ချုံ့လိုက်တာပါ။

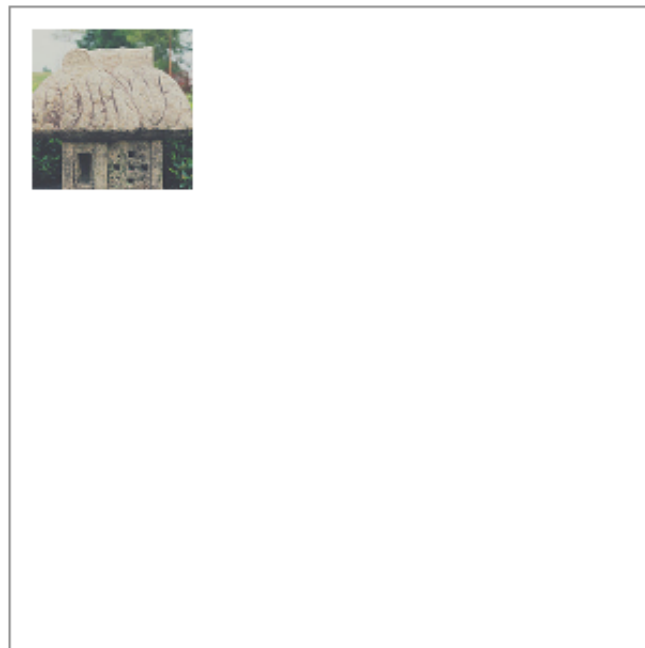


Figure 11-16

Slice an Image

ပုံဖြတ်တာကတော့ ပုံထည့်တာနဲ့ သိပ်မကွာပါဘူး။ သို့ပေမယ့် parameter တွေ ကွာသွားပါတယ်။

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Canvas</title>
    <style>
      canvas#myCanvas {border: 1px solid gray;}
    </style>
  </head>
  <body>
    <canvas id="myCanvas" width="300" height="300"></canvas>
    <script>
      var canvas = document.getElementById('myCanvas');
      var ctx = canvas.getContext('2d');
      var pic = new Image();
```

```

    pic.src = 'house.jpg';
    pic.onload = function () {
        ctx.drawImage(pic,18,100,80,35,0,0,80,35);
    }
</script>
</body>
</html>

```

ဆိုတဲ့ code လေးဟာ အောက်ကလို ပေါ်ပါလိမ့်မယ်။



Figure 11-17

drawImage ရဲ့ parameter ကတော့

```

ctx.drawImage(imageObject, startCropPointX, startCropPointY, cropWidth, cropHeight, xPos, yPos,
width, height);

```

Parameter	Description
imageObject	JavaScript image object ပါ
startCropPointX	စပြီးဖြတ်ချင်တဲ့ အမှတ် x point ပါ
startCropPointY	စပြီးဖြတ်ချင်တဲ့ အမှတ် y point ပါ
cropWidth	ဖြတ်မှတ် x,y point ကနေ ဖြတ်မယ့် အကျယ်ပါ
cropHeight	ဖြတ်မှတ် x,y point ကနေ ဖြတ်မယ့် အမြင့်ပါ
xPos	ပုံဖြတ်ပြီးတဲ့ အခါ canvas ပေါ်မှာ ပေါ်မယ့် x အမှတ်ပါ
yPos	ပုံဖြတ်ပြီးတဲ့ အခါ canvas ပေါ်မှာ ပေါ်မယ့် y အမှတ်ပါ
width	ပုံရဲ့ အကျယ်ပါ
height	ပုံရဲ့ အမြင့်ပါ

Add a Gradient

Gradient ဆိုတာကတော့ အရောင် တစ်ခုကနေ နောက် အရောင်တစ်ခုကို သန်းပြီး ပြောင်းသွားတဲ့ အရောင်ပါ။ gradient ကို ပုံတွေ ဆွဲသည့် အခါမှာ ထည့်သုံးကြတာကို တွေ့မိမှာပါ။

```

<!DOCTYPE html>
<html>
  <head>

```

```

<meta charset="UTF-8">
<title>Canvas</title>
<style>
canvas#myCanvas {border: 1px solid gray;}
</style>
</head>
<body>
<canvas id="myCanvas" width="600" height="400"></canvas>
<script>
var canvas = document.getElementById('myCanvas');
var ctx = canvas.getContext('2d');
var grd = ctx.createLinearGradient(0,0,0,400);

grd.addColorStop(0,"yellow");
grd.addColorStop(0.5,"orange");
grd.addColorStop(1,"red");

ctx.fillStyle = grd;
ctx.fillRect(0,0,600,400);
</script>
</body>
</html>

```

gradient အတွက် code က နည်းနည်းများပါတယ်။ ပထမဆုံး `createLinearGradient` နဲ့ စမှတ် နဲ့ ဆုံးမှတ်ကို သတ်မှတ်ပါတယ်။ ကျွန်တော်တို့ဟာ 600,400 ရှိတဲ့ စတုဂံကို ဆွဲမှာ ဖြစ်ပြီးတော့ အဲဒီ စတုဂံကို gradient အရောင် ခြယ်ပါမှာ။ gradient ကို စမှတ် 0,0 ကနေ 0, 400 အထိ ဆွဲလိုက်ပါတယ်။ ဒါကြောင့် အရောင်ခြယ်တဲ့ အခါမှာ ဒေါင်လိုက် gradient ဖြစ်နေပါမယ်။

ပြီးတဲ့ အခါမှာတော့ အဝါရောင်ကို စမှတ် ထားလိုက်တယ်။ ပြီးတော့ လိမ္မော်ရောင်ကို တဝက်မှာ ထားပြီးတော့ အဆုံးမှာ အနီရောင်ထားလိုက်ပါတယ်။ ပြီးတဲ့ စတုဂံဆွဲလိုက်ပါတယ်။ အဲဒီ အခါမှာတော့ အောက်ကလို ပုံရလာပါတယ်။



Figure 11-18

တကယ်လို့ စက်ဝိုင်းပုံစံ လိုချင်ရင်တော့ `createRadialGradient` ကို အသုံးပြုရပါတယ်။

```
var grd = ctx.createRadialGradient(300,200,10,300,200,360);
```

လို့ ပြင်လိုက်ရင် အောက်က ပုံမြင်ရပါမယ်။



Figure 11-19

စက်ဝိုင်းပုံစံ အတွက် `createRadialGradient` parameter ကတော့

```
ctx.createRadialGradient(x0,y0,r0,x1,y1,r1);
```

Parameter	Description
x0	gradient ဆွဲဖို့ အတွက် x point စမှတ်
y0	gradient ဆွဲဖို့ အတွက် y point စမှတ်
r0	စက်ဝိုင်းရဲ့ အချင်း
x1	x-coordinate ဆုံးမှတ်
y1	y-coordinate ဆုံးမှတ်
r1	စက်ဝိုင်းရဲ့ ဆုံးမှတ် အခြင်း

Rotate Canvas Content

Canvas မှာ ထည့်ထားတဲ့ item တွေကို ကျွန်တော်တို့တွေ လှည့်လို့ရပါတယ်။ Canvas ထဲက Shapes တွေ images တွေ text တွေ အပြင် အခြား elements တွေကို လှည့်လို့ရပါတယ်။ `rotate` ဆိုတဲ့ function အတွက် radian ကို ပေးရပါတယ်။ အပြည့်လှည့်ချင်ရင်တော့ 2PI value ပေးရပါမယ်။ ဥပမာလေး ကြည့်ရအောင်။

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Canvas</title>
    <style>
      canvas#myCanvas {border: 1px solid gray;}
    </style>
  </head>
  <body>
    <canvas id="myCanvas" width="300" height="300"></canvas>
    <script>
      var canvas = document.getElementById('myCanvas');
      var ctx = canvas.getContext('2d');
      var pic = new Image();
      pic.src = 'house.jpg';
      pic.onload = function () {
        ctx.rotate(10 * Math.PI/180);
        ctx.drawImage(pic,50,50);
      }
    </script>
  </body>
</html>
```

ဒီ code က ကျွန်တော်တို့ ပုံထည့်တဲ့ code နဲ့ အတူတူပါပဲ။ ကွာခြားချက်ကတော့

```
ctx.rotate(10 * Math.PI/180);
```

10 နေရာမှာ ကျွန်တော်တို့ လှည့်ချင်တဲ့ degree ထည့်လို့ရပါတယ်။ အနှုတ်တန်ဖိုးလည်း ထည့်လို့ရပါတယ်။ ကျွန်တော်တို့တွေ အနေနဲ့ ပုံတွေသာမက စာတွေ ကော ၊ စတုဂံပုံ စတာတွေကို ပါ rotate လုပ်လို့ရပါတယ်။



Figure 11-20

Animate Canvas Content

ကျွန်တော်တို့တွေ canvas မှာ ပုံဆွဲ ရှိသာမက animation တွေ ပါထည့်လို့ရပါတယ်။ animation လုပ်တဲ့ အခါမှာ canvas ကို `beginPath()` နဲ့ `clear` လုပ်ပြီး ပုံတွေ ထပ်ဆွဲတာပါ။

ကျွန်တော်တို့တွေ ဘောလုံးလေးကို အလျားလိုက် ပေးစေချင်တယ်။ ဘောင်နဲ့ ရိုက်ရင် ပြန်လှည့် လာမယ့် ပုံစံမျိုးလုပ်ချင်တယ်။

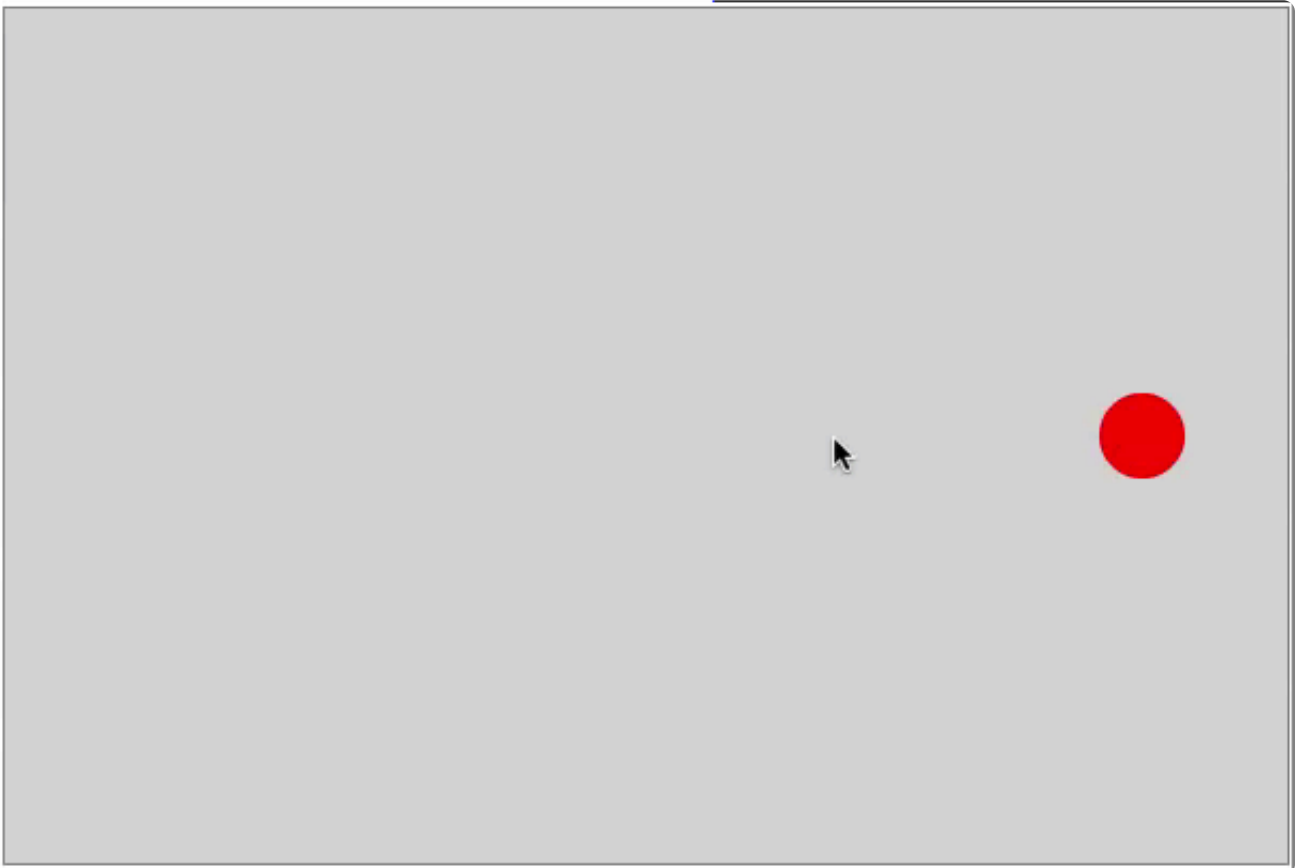


Figure 11-21

အဲဒီ အတွက် ကျွန်တော် တို့တွေ JavaScript ရေးရပါမယ်။

code ကတော့ နည်းနည်းလေး ရှည်ပါတယ်။ တချက်ကြည့်လိုက်ပါ။

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Canvas</title>
    <style>
      canvas#myCanvas {border: 1px solid gray;}
    </style>
  </head>
  <body>
    <canvas id="myCanvas" width="600" height="400"></canvas>
    <script>
      var canvas = document.getElementById('myCanvas');
      var ctx = canvas.getContext('2d');
      var x = 200;
      var y = 200;
      var change = 4;
      var w = 600;
      var h = 400;

      function animate() {
        ctx.fillStyle = 'lightgray';
        ctx.fillRect(0,0,w,h);
        ctx.fillStyle='red';
        ctx.beginPath();
        ctx.arc(x,y,20,0,Math.PI*2,true);
      }
    </script>
  </body>
</html>

```

```

    ctx.fill();

    if(x >= w || x <= 0) {
        change = - change;
    }
    x = x + change;
}

setInterval(animate,10);
</script>
</body>
</html>

```

ကျွန်တော်တို့တွေဟာ ဘောလုံးရဲ့ စမယ့် နေရာကို x,y variable ပေးထားပါတယ်။

```

var x = 200;
var y = 200;

```

ပြီးတော့ ဘောလုံး ရဲ့ x position ကို တစ်ခါရွေ့လိုက်တိုင်း 4 pixel တိုးသွားစေချင်တော့

```

var change = 4;

```

ဆိုပြီးပေးထားတယ်။

canvas ရဲ့ အကျယ် နဲ့ အမြင့်အတွက်

```

var w = 600;
var h = 400;

```

10 milliseconds တိုင်းမှာ canvas ရဲ့ animation ကို ပြောင်းစေချင်တဲ့ အတွက်ကြောင့်

```

setInterval(animate,10);

```

ဆိုပြီး ရေးထားပါတယ်။ 10 milliseconds တိုင်းမှာ animate ဆိုတဲ့ function ကို ခေါ်ပါမယ်။

```

ctx.fillStyle = 'lightgray';
ctx.fillRect(0,0,w,h);

```

ကတော့ canvas ရဲ့ background ကို ခဲရောင် ထားလိုက်တာပါ။

```

ctx.beginPath();

```

ကတော့ လက်ရှိ canvas မှာ အောက်မှာ ရှိတာတွေကို animate လုပ်တိုင်း clear လုပ်ပြီး ထပ်ထပ် ဆွဲပါလိမ့်မယ်။

```
ctx.arc(x,y,20,0,Math.PI*2,true);
ctx.fill();
```

ကတော့ စက်ဝိုင်း တစ်ခု ဆွဲလိုက်ပါတယ်။

```
if(x >= w || x <= 0) {
  change = - change;
}
```

စက်ဝိုင်းရဲ့ x ဟာ ဘောင်ရောက်သွားရင် ဒါမှမဟုတ် 0 ထက် နည်းသွားရင် position ပြောင်း လိုက်ပါတယ်။

```
x = x + change;
```

ကတော့ နောက်ထပ် animation အတွက် x position ပြောင်းဖို့ပါ။

ဒီ chapter မှာ ကျွန်တော့် အနေနဲ့ canvas တစ်ခုလုံးကို အသေးစိတ် မရေးနိုင်ပေမယ့် အတတ်နိုင် ဆုံး အကုန်လုံးကို ခြုံမိအောင် ရေးသားထားပါတယ်။ ဒါကြောင့် စလေ့လာမယ့် သူတွေ အတွက် အဆင်ပြေမယ်လို့ မျှော်လင့်ပါတယ်။

Adding Video and Audio

Video နဲ့ audio တွေကို ကိုယ့်ရဲ့ website ထဲမှာ ထည့်သွင်းနိုင်ပါတယ်။ video ဟာ ကိုယ့်ရဲ့ product ကို ကြော်ငြာ သဘောမျိုး ဒါမှမဟုတ် ရုပ်ရှင် စတာတွေ အတွက် ထည့်သွင်းကြပါတယ်။

Video

Video တွေဟာ internet connection ကောင်းဖို့ လိုပါတယ်။ video ကို ထည့်သွင်းရာမှာ file size ကြီးတဲ့ အခါမှာ video မြင်ရတာ နှေးသွားနိုင်သလို browser တွေ တစ်ခု နဲ့ တစ်ခု ဟာ support လုပ်တဲ့ file format တွေ မတူပါဘူး။

Modern HTML မှာ video ထည့်သွင်းဖို့အတွက် `<video>` tag ပါဝင်ပါတယ်။ `src` attribute မှာ `` tag လိုမျိုး video ရဲ့ ပတ်လမ်းကြောင်းကို ထည့်သွင်းနိုင်ပါတယ်။

Audio

Audio file တွေဟာ ကိုယ့်ရဲ့ site ထဲမှာ သီချင်းတွေ နောက်ပြီးတော့ အသံသွင်းထားတာတွေကို ထည့်ချင်တဲ့ အခါမှာ အသုံးပြုနိုင်ပါတယ်။ နောက်ပြီးတော့ podcast လိုမျိုး စကားပြော အခန်းဆက်တွေလိုမျိုး website တွေမှာလည်း အသုံးပြုနိုင်ပါတယ်။

Modern HTML မှာ audio ထည့်သွင်းဖို့အတွက် `<audio>` tag ပါဝင်ပါတယ်။ `src` attribute မှာ `` tag လိုမျိုး audio file ရဲ့ ပတ်လမ်းကြောင်းကို ထည့်သွင်းနိုင်ပါတယ်။

Files

Video file တွေဟာ audience, device, connection speed ပေါ်မူတည်ပြီး 480p, 720p, 1080p, 4K စတာတွေကို ရွေးချယ်အသုံးပြုနိုင်ပါတယ်။ Website မှာတော့ file size, loading speed, mobile data usage တွေကို သတိထားပြီး resolution အမျိုးမျိုး ထားပေးတာ ပိုကောင်းပါတယ်။

Audio, Video file တွေကို ပြောင်းဖို့အတွက် converter တော်တော်များများ ရှိပါတယ်။

[HandBrake](#) converter ဟာ video တွေကို ပြောင်းလဲဖို့ အတွက် ကောင်းမွန်တဲ့ App ဖြစ်ပြီးတော့ free လည်း ဖြစ်ပါတယ်။ WebM, MP4 စတဲ့ format တွေကို ပြောင်းလဲနိုင်ပါတယ်။

Different Browsers, Different Formats

Video, Audio တွေဟာ browser ပေါ်မှာ မှုတည်ပြီး support လုပ်တာ ကွာနိုင်ပါတယ်။ အခုခေတ် မှုတော့ MP4/H.264 video နဲ့ MP3/AAC audio ကို browser အများစုမှာ support ကောင်းကောင်း ရှိပါတယ်။ WebM ကိုလည်း modern browsers တွေမှာ အသုံးများလာပါတယ်။ Ogg/Theora ကို compatibility အတွက် အဓိက format အဖြစ် မရွေးတော့ဘဲ fallback အနေနဲ့သာ ထည့်သင့်ပါတယ်။

Format	Common Use
MP4 (H.264/AAC)	အများဆုံး compatible ဖြစ်တဲ့ video format
WebM (VP9/AV1/Opus)	Modern browser တွေမှာ သုံးနိုင်တဲ့ efficient video format
MP3	အများဆုံး compatible ဖြစ်တဲ့ audio format
Ogg/Opus	Open format audio အတွက် အသုံးပြုနိုင်

Insert a Video File

ကျွန်တော်တို့ webpage မှာ video ကို `<video>` tag အသုံးပြုပြီး ထည့်သွင်းနိုင်ပါတယ်။ `src` attribute မှာ video သိမ်းထားတဲ့ နေရာလေးကို ထည့်ပေးလိုက်ရင် ရပါပြီ။ Browser မှာ play , pause စတဲ့ control နဲ့ video progress စတာတွေ ပါလာပါလိမ့်မယ်။ UI က browser ပေါ်မှာ မှုတည်ပြီး ပြောင်းလဲ မှု ရှိပါတယ်။ နောက်တနည်းကတော့ youtube လိုမျိုး video host မှာ တင်ပြီးတော့ မိမိ ရဲ့ website ထဲမှာ embed လုပ်ပြီး ထည့်တဲ့ နည်းပါပဲ။

အရင်ဆုံး ကျွန်တော်တို့တွေ video tag နဲ့ ထည့်ကြည့်ရအောင်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Video</title>
</head>
<body>
  <h1>Big Buck Bunny</h1>
  <video src="movie.mp4" controls>
    Your browser does not support the video tag.
  </video>
</body>
</html>

```

video မှာ movie.mp4 file လေးကို ထည့်ထားပါတယ်။ Modern browser အများစုမှာ ကြည့်လို့ရ ပါလိမ့်မယ်။ တကယ်လို့ video tag ကို support မလုပ်တဲ့ browser ဖြစ်ခဲ့ရင် **Your browser does not support the video tag.** ဆိုပြီး ဖော်ပြပါလိမ့်မယ်။

Big Buck Bunny



Figure 12-1

video tag မှာ controls ဆိုတာကတော့ play, pause စတဲ့ control တွေ ပါမယ်လို့ ဆိုလိုပါတယ်။

တကယ်လို့ browser တက်လာခြင်းမှာ video ကို play လုပ်စေချင်ရင်တော့ `autoplay` ဆိုတာကို ထည့်ထားနိုင်ပါတယ်။ သို့ပေမယ့် အသံပါသော media ကို user action မရှိဘဲ autoplay လုပ်တာ ကို browser အများစုက block လုပ်နိုင်ပါတယ်။ Autoplay လိုအပ်ရင် muted video အနေနဲ့ သုံး ပါ။

```
<video src="movie.mp4" controls autoplay muted>
```

Insert an Audio File

Audio file ကို background music ပုံစံ အနေနဲ့ ထည့်သွင်းလို့ရသလို controls တွေကို ဖော်ပြထား ပြီး user ကို play, pause လုပ်ခွင့်ပေးထားလို့လည်း ရပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Video</title>
</head>
<body>
  <h1>Music</h1>
  <audio src="music.mp3" controls>
    Your browser does not support the audio tag.
  </audio>
</body>
</html>
```

ဒီ code လေးမှာ controls ဆိုတာကို ထည့်ထားတဲ့ အတွက်ကြောင့် play,pause button တွေကို မြင် နေရပါလိမ့်မယ်။

background music အနေနဲ့ဆိုရင်

```
<audio src="music.mp3" controls>
  Your browser does not support the audio tag.
</audio>
```

ဆိုပြီး အသုံးပြုနိုင်ပါတယ်။ အခုခေတ် browser တွေမှာ အသံကို user action မရှိဘဲ autoplay လုပ်တာကို block လုပ်နိုင်တဲ့အတွက် audio အတွက် controls ထည့်ပြီး user ကို play လုပ်ခွင့် ပေးတာ ပိုကောင်းပါတယ်။

Music



Figure 12-2

Resize a Video

Video ရဲ့ အရွယ်အစားကို မူလ အရွယ်အစားမဟုတ်ပဲ ကျွန်တော်တို့တွေ width နှင့် height attribute ကို အသုံးပြုပြီး ပြောင်းနိုင်ပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Video</title>
</head>
<body>
  <h1>Big Buck Bunny</h1>
  <video src="movie.mp4" controls width="400">
    Your browser does not support the video tag.
  </video>
</body>
</html>
```

ပုံမှန် video html code လေးပါပဲ။ ကျွန်တော်တို့ width ကို 400px ပေးထားတာကြောင့် browser မှာ 400px အရွယ်အစားဖြစ်နေပါလိမ့်မယ်။

Big Buck Bunny



Figure 12-3

height ကတော့ အလိုအလျောက် width ပေါ်မှာမူတည်ပြီး အရွယ်အစားကို ပြောင်းပေးပါလိမ့်မယ်။ တကယ်လို့ စိတ်ကြိုက် အရွယ်အစား ဆိုရင်တော့ height attribute ကို ထည့်နိုင်ပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Video</title>
</head>
<body>
  <h1>Big Buck Bunny</h1>
  <video src="movie.mp4" controls width="400" height="400" style="background:black">
    Your browser does not support the video tag.
  </video>
</body>
</html>
```

ဒီ code မှာဆိုရင် video control မြင်ရအောင် CSS နဲ့ background ကို အမည်းရောင်ထည့်ထားပါတယ်။ width နဲ့ height ကို 400 ပေးထားပါတယ်။

Big Buck Bunny

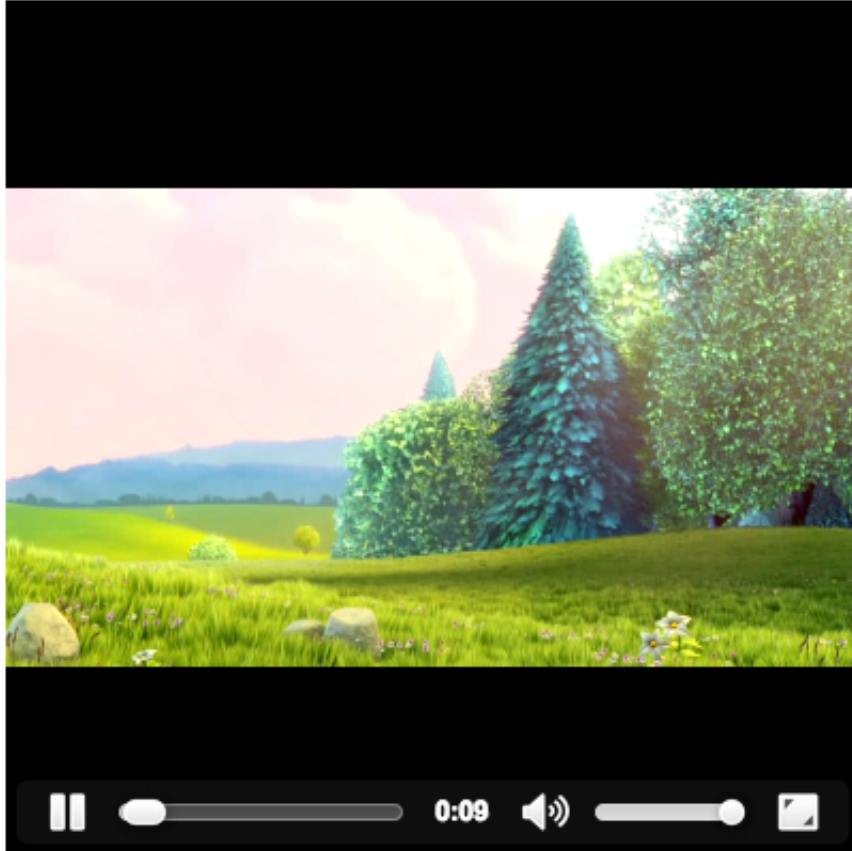


Figure 12-4

Preload Multimedia

Web page မှာ multimedia file တွေကို preload ခေါ်လို့ရပါတယ်။ `preload` ဆိုတာကတော့ browser ကို video/audio အတွက် ဘယ်လောက်အထိ ကြိုတင် load လုပ်စေချင်လဲ ဆိုတဲ့ hint ပေးတာပါ။ Browser က network condition နဲ့ user setting ပေါ်မူတည်ပြီး အဲဒီ hint ကို မလိုက်နာ နိုင်ပါဘူး။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Video</title>
</head>
<body>
  <h1>Big Buck Bunny</h1>
  <video src="movie.mp4" controls preload="metadata">
    Your browser does not support the video tag.
  </video>
</body>
</html>
```

`preload="metadata"` လေးထည့်ထားတာကို တွေ့ရပါလိမ့်မယ်။ ဒါကြောင့် browser က file တစ်ခုလုံးမဟုတ်ဘဲ duration, dimensions စတဲ့ metadata ကိုပဲ ကြိုတင် load လုပ်နိုင်ပါတယ်။ `preload="auto"` က data အသုံးများနိုင်တာကြောင့် video အများကြီးပါတဲ့ page တွေမှာ သတိထားသုံးသင့်ပါတယ်။

Loop Multimedia

`loop` attribute ဟာ `<video>` နဲ့ `<audio>` tag တွေကို ပြီးသွားတဲ့ အခါ အစကနေ ပြန်စချင်တဲ့ အခါမှာ အသုံးပြုပါတယ်။ ဥပမာ audio ဆိုရင် background music ဖွင့်ထားပြီးတော့ သီချင်းဆိုးတာနဲ့ အစက ပြန်စချင်ရင် အသုံးပြုနိုင်ပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Video</title>
</head>
<body>
  <h1>Music</h1>
  <audio src="music.mp3" controls loop>
    Your browser does not support the audio tag.
  </audio>
</body>
</html>
```

ဒီ code ဟာ ဆိုရင် user က play လုပ်ပြီးနောက် သီချင်း ဆိုးသွားတာနဲ့ အစကနေ ပြန်စပါလိမ့်မယ်။

Offer Multiple Sources

ကျွန်တော်တို့တွေ အနေနဲ့ audio ၊ video တွေရဲ့ source ကို တစ်ခုထက် မက ထည့်လို့ရပါတယ်။ `<video>` မှာဆိုရင် .mp4, .webm စတာတွေကို တစ်ခါတည်း ထည့်ထားလို့ရပါတယ်။ Browser က ပထမ source နဲ့ ဖွင့်လို့ မရရင် နောက်ထပ် source တစ်ခုကို အလိုအလျောက် စမ်းဖွင့်ပေးပါလိမ့်မယ်။ Compatibility ကောင်းဖို့ MP4 ကို fallback အနေနဲ့ ထည့်ထားတာက အများဆုံး အသုံးဝင်ပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Video</title>
</head>
<body>
  <h1>Big Buck Bunny</h1>
```

```

<video controls>
  <source src="movie.webm" type="video/webm">
  <source src="movie.mp4" type="video/mp4">
  Your browser does not support the video tag.
</video>
</body>
</html>

```

ဒီ example မှာဆိုရင်တော့ ပုံမှန် video နဲ့ အတူတူပါပဲ။ ကွာတာကတော့ source အသုံးပြုထားတာပါ။

<source> tag ရဲ့ src attribute မှာ ထည့်မယ့် video ရဲ့ file လမ်းကြောင်းကို ထည့်ရပါတယ်။ type ကတော့ video ရဲ့ MIME ပါ။ ဒီ code မှာ multiple source ကို အသုံးပြုထားပြီး webm နဲ့ mp4 ထည့်ထားတဲ့အတွက် browser က support လုပ်နိုင်တဲ့ format ကို ရွေးဖွင့်ပါလိမ့်မယ်။

Support Older Browsers

အရင်ခေတ်မှာ QuickTime, Flash, <object> , <embed> စတဲ့ plugin fallback နည်းတွေကို multimedia အတွက် အသုံးပြုခဲ့ကြပါတယ်။ အခုခေတ် browser တွေမှာတော့ plugin-based fallback ကို မသုံးသင့်တော့ပါဘူး။ Browser support အတွက် format မျိုးစုံကို <source> နဲ့ ထည့်ပြီး fallback text ကို <video> သို့မဟုတ် <audio> ထဲမှာ ရေးထားပါ။

```

<video controls width="640" poster="poster.jpg">
  <source src="movie.webm" type="video/webm">
  <source src="movie.mp4" type="video/mp4">
  Your browser does not support the video tag.
</video>

```

တကယ်လို့ browser အဟောင်းတစ်ခုက <video> ကို support မလုပ်ဘူးဆိုရင် fallback text ကို ပြပါလိမ့်မယ်။ Production website တွေမှာတော့ analytics ကြည့်ပြီး support လုပ်ရမယ့် browser တွေကို ဆုံးဖြတ်ပါ။ အလွန်ဟောင်းတဲ့ browser တွေအတွက် plugin fallback ထည့်တာထက် downloadable link သို့မဟုတ် external video hosting ကို အသုံးပြုတာက ပိုလုံခြုံပြီး ပြုပြင်ထိန်းသိမ်းရလွယ်ပါတယ်။

Embed a YouTube Video

video တွေကို ကျွန်တော်တို့ရဲ့ website မှာ မထားချင်ဘူး။ youtube ပေါ်မှာ တင်ချင်တယ်။ ပြီးမှ website ထဲမှာ ထည့်သုံးချင်တယ် ဆိုရင် လွယ်လင့် တကူ အသုံးပြုနိုင်ပါတယ်။ ကိုယ့်ထည့်ချင်တဲ့ youtube video ဆီ သွားလိုက်ပါ။ Video အောက်မှာ share ဆိုတာ ပါပါတယ်။ share အောက်မှာ Embed ဆိုတာ ရှိပါတယ်။

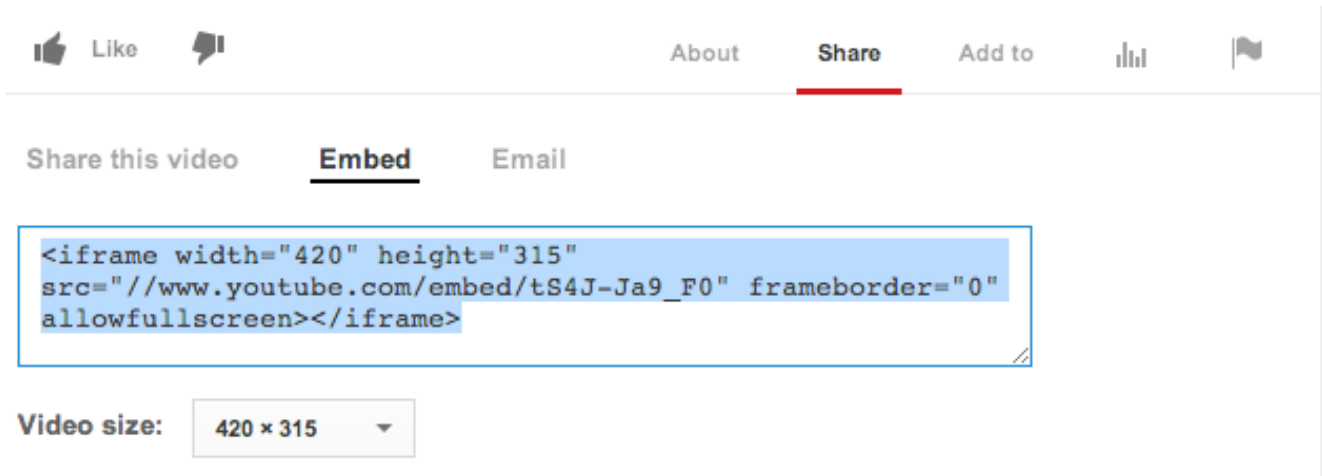


Figure 12-6

အဲဒီ က code လေးကို copy ကူးပြီးတော့ ကျွန်တော်တို့ website ထဲမှာ ထည့်လိုက်ရုံပါပဲ။ အဲဒါဆို ရင် code လေးက ဒီလို မျိုး ဖြစ်ပါလိမ့်မယ်။

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Video</title>

</head>
<body>
  <h1>Youtube</h1>

  <iframe
    width="420"
    height="315"
    src="https://www.youtube.com/embed/tS4J-Ja9_F0"
    title="YouTube video player"
    style="border:0"
    loading="lazy"
    allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-
in-picture; web-share"
    allowfullscreen>
  </iframe>

</body>
</html>
```

ဒီ example မှာ <https://> ကို အသုံးပြုထားပါတယ်။ `frameborder` attribute က obsolete ဖြစ်နေ ပြီဖြစ်တဲ့အတွက် border ဖျောက်ချင်ရင် CSS (`style="border:0"` သို့မဟုတ် stylesheet) ကို အသုံးပြုပါ။ `loading="lazy"` ကတော့ iframe ကို viewport နားရောက်မှ load လုပ်စေနိုင်တဲ့ အတွက် page speed အတွက် အကူအညီဖြစ်နိုင်ပါတယ်။

Youtube

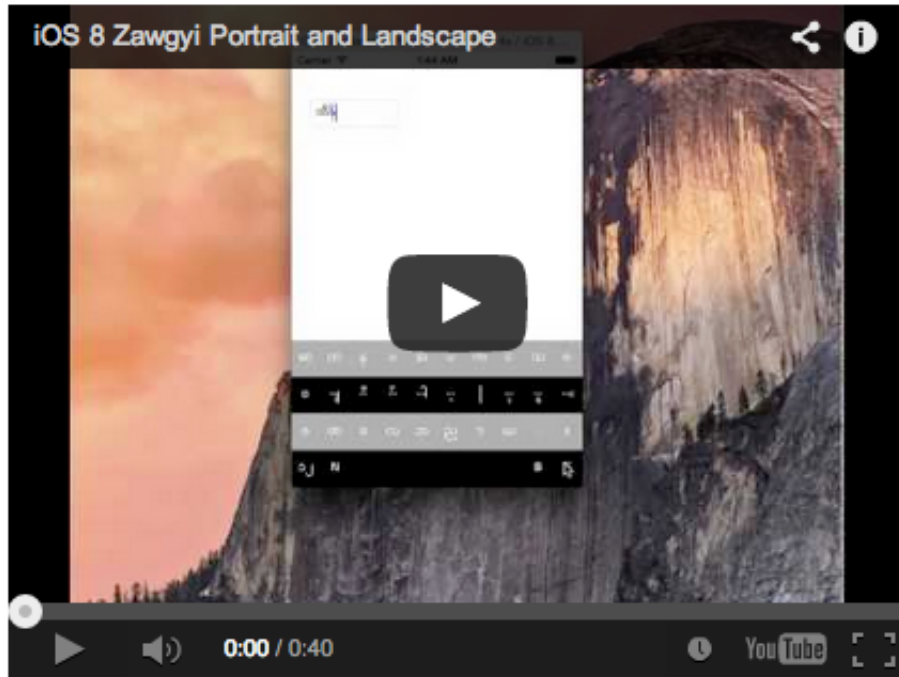


Figure 12-7

Modern HTML Features

HTML5 ဆိုတဲ့အမည်ကို လူအများသုံးနေကြသေးပေမယ့် အခုခေတ် HTML ကတော့ HTML Living Standard အနေနဲ့ ဆက်လက် update ဖြစ်နေပါတယ်။ Web page တွေကို layout ချရာမှာ အရင်က float တွေကို အသုံးပြုခဲ့ကြပေမယ့် အခုနောက်ပိုင်းမှာတော့ Flexbox နဲ့ Grid ကို အသုံးပြုပြီး ပိုမို လွယ်ကူ လျင်မြန်စွာ တည်ဆောက်လာကြပါတယ်။ ဒါ့အပြင် browser မှာ data သိမ်းဆည်းဖို့အတွက် Web Storage, user ရဲ့ တည်နေရာကို ရယူဖို့ Geolocation API, responsive images, dialog စတဲ့ element/API တွေလည်း အသုံးများလာပါတယ်။

Flexbox Layout

Flexbox (Flexible Box Layout) ဟာ one-dimensional layout system ဖြစ်ပါတယ်။ Row (သို့မဟုတ်) Column တစ်ခုတည်းအတွက် layout ချချင်တဲ့အခါမှာ အလွန် အသုံးဝင်ပါတယ်။ Element တွေကို အလယ်မှာ ထားချင်တာ၊ အညီအမျှ နေရာ ခွဲပေးချင်တာတွေအတွက် အရမ်း လွယ်ကူပါတယ်။

`display: flex;` လို့ ကြေညာလိုက်တာနဲ့ အဲဒီ container ထဲက child elements တွေဟာ flex item တွေ ဖြစ်သွားပါမယ်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Flexbox Example</title>
  <style>
    .container {
      display: flex;
      justify-content: center; /* အလျားလိုက် အလယ်ပိုမယ် */
      align-items: center; /* ဒေါင်လိုက် အလယ်ပိုမယ် */
      height: 200px;
      border: 1px solid gray;
    }
    .box {
      background-color: orange;
      padding: 20px;
      margin: 10px;
      color: white;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="box">1</div>
    <div class="box">2</div>
    <div class="box">3</div>
  </div>

```

```

</div>
</body>
</html>

```

အထက်ပါ code မှာ `justify-content: center` ကြောင့် box လေးတွေဟာ အလျားလိုက် အလယ် ကို ရောက်နေမှာ ဖြစ်ပြီး `align-items: center` ကြောင့် ဒေါင်လိုက် အလယ်ကို ရောက်နေမှာ ဖြစ်ပါတယ်။

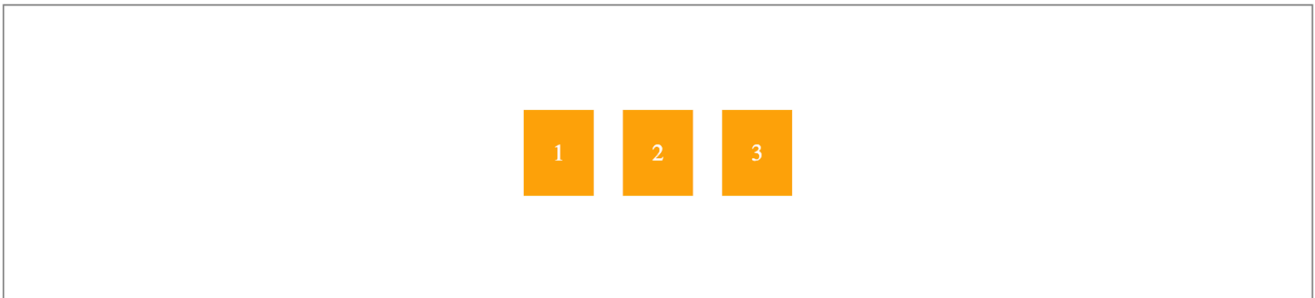


Figure 13-1

CSS Grid Layout

Flexbox က one-dimensional ဖြစ်ပြီး CSS Grid ကတော့ two-dimensional layout system ဖြစ်ပါတယ်။ Row တွေရော Column တွေရော ပါဝင်တဲ့ ရှုပ်ထွေးတဲ့ layout တွေကို တည်ဆောက်ဖို့ အတွက် အကောင်းဆုံးပါပဲ။

`display: grid;` နဲ့ layout စဆောက်နိုင်ပြီး `grid-template-columns` နဲ့ column ဘယ်နှစ်ခု ထားမလဲဆိုတာ သတ်မှတ်နိုင်ပါတယ်။

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Grid Example</title>
<style>
.grid-container {
display: grid;
grid-template-columns: auto auto auto; /* Column 3 ခု */
gap: 10px; /* Box တွေ ကြားက အကွာအဝေး */
padding: 10px;
}
.grid-item {
background-color: #2196F3;
border: 1px solid rgba(0, 0, 0, 0.8);
padding: 20px;
font-size: 30px;
text-align: center;
color: white;
}
</style>
</head>

```

```

<body>
  <div class="grid-container">
    <div class="grid-item">1</div>
    <div class="grid-item">2</div>
    <div class="grid-item">3</div>
    <div class="grid-item">4</div>
    <div class="grid-item">5</div>
    <div class="grid-item">6</div>
  </div>
</body>
</html>

```

`grid-template-columns: auto auto auto;` လို့ ပေးထားတဲ့အတွက် browser က အလိုအလျောက် ညီတူညီမျှ column ၃ ခု ခွဲပေးသွားမှာ ဖြစ်ပါတယ်။



Figure 13-2

Web Storage API

အရင်တုန်းက browser မှာ data သိမ်းဖို့ cookies တွေကို သုံးခဲ့ကြပါတယ်။ HTML5 မှာတော့ `localStorage` နဲ့ `sessionStorage` ဆိုပြီး ပိုမို ကောင်းမွန်တဲ့ နည်းလမ်းတွေ ပါဝင်လာပါတယ်။

- `localStorage` - Browser ပိတ်လိုက်လည်း data မပျောက်ပါဘူး။
- `sessionStorage` - Browser tab ပိတ်လိုက်တာနဲ့ data ပျောက်သွားပါမယ်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Web Storage</title>
</head>
<body>
  <input type="text" id="username" placeholder="Enter Name">
  <button
  <button
  <p id="display"></p>

  <script>
    function saveName() {
      var name = document.getElementById('username').value;
      localStorage.setItem('myname', name); // Data သိမ်းမယ်
      alert("Saved!");
    }
  </script>

```

```

function showName() {
  var storedName = localStorage.getItem('myname'); // Data ပြန်ယူမယ်
  if (storedName) {
    document.getElementById('display').innerText = "Hello " + storedName;
  } else {
    alert("No name saved!");
  }
}
</script>
</body>
</html>

```

ဒီ example မှာ Save နှိပ်လိုက်ရင် `localStorage.setItem` နဲ့ နာမည်ကို သိမ်းလိုက်ပါမယ်။
Browser ကို ပိတ်ပြီး ပြန်ဖွင့်ရင်တောင် Show ကို နှိပ်လိုက်ရင် သိမ်းထားတဲ့ နာမည်ကို ပြန်ပြပေး
နိုင်တာ တွေ့ရပါလိမ့်မယ်။

Hello Saturngod

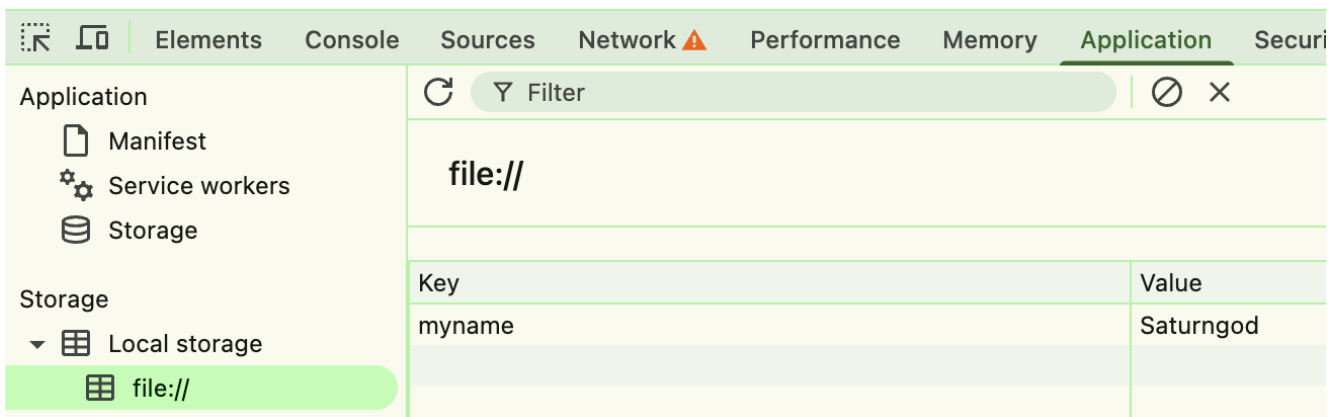


Figure 13-3

Geolocation API

User ရဲ့ တည်နေရာ (Latitude, Longitude) ကို ရယူချင်ရင် Geolocation API ကို သုံးနိုင်ပါတယ်။
ဒီ feature ကို သုံးမယ်ဆိုရင် user ဆီက permission အရင် တောင်းပါလိမ့်မယ်။

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Geolocation</title>
</head>
<body>
  <button Location</button>

```

```

<p id="locationDisplay"></p>

<script>
  var x = document.getElementById("locationDisplay");

  function getLocation() {
    if (navigator.geolocation) {
      navigator.geolocation.getCurrentPosition(showPosition);
    } else {
      x.innerHTML = "Geolocation is not supported by this browser.";
    }
  }

  function showPosition(position) {
    x.innerHTML = "Latitude: " + position.coords.latitude +
      "<br>Longitude: " + position.coords.longitude;
  }
</script>
</body>
</html>

```

`navigator.geolocation.getCurrentPosition` ကို ခေါ်လိုက်ရင် browser က "Allow location access?" ဆိုပြီး မေးပါလိမ့်မယ်။ Allow ပေးလိုက်မှသာ Latitude နဲ့ Longitude ကို ပြပေးမှာ ဖြစ်ပါတယ်။

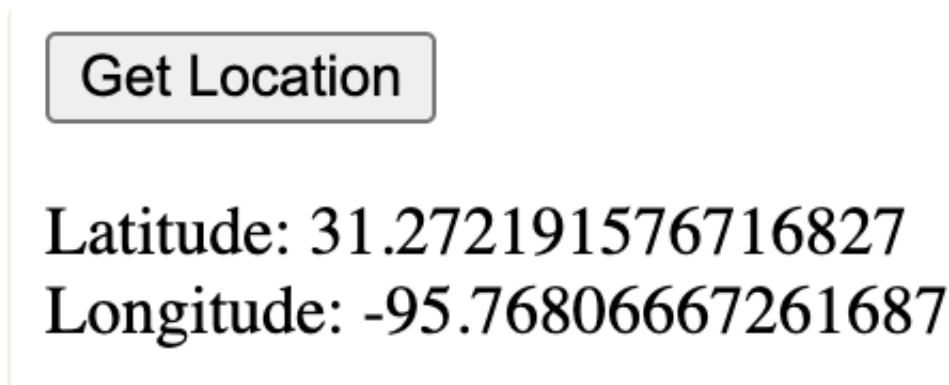


Figure 13-4

Responsive Images

Mobile, tablet, desktop စတဲ့ screen size မတူတဲ့ device တွေမှာ ပုံတစ်ပုံတည်းကို အမြဲတမ်း တင်ပေးရင် file size မလိုအပ်ဘဲ ကြီးသွားနိုင်ပါတယ်။ Modern HTML မှာ `srcset` , `sizes` , `<picture>` တို့ကို အသုံးပြုပြီး browser ကို သင့်တော်တဲ့ image ကို ရွေးခိုင်းနိုင်ပါတယ်။

```



```

Format fallback လိုချင်ရင် `<picture>` ကို အသုံးပြုနိုင်ပါတယ်။

```
<picture>
  <source srcset="photo.avif" type="image/avif">
  <source srcset="photo.webp" type="image/webp">
  
</picture>
```

Dialog Element

Modal dialog တွေ ဖန်တီးရာမှာ `<dialog>` element ကို အသုံးပြုနိုင်ပါတယ်။ JavaScript နဲ့ `showModal()` ခေါ်လိုက်ရင် browser က modal behavior ကို built-in အနေနဲ့ ထိန်းပေးပါတယ်။

```
<button id="openDialog">Open</button>

<dialog id="welcomeDialog">
  <p>Welcome to modern HTML.</p>
  <button id="closeDialog">Close</button>
</dialog>

<script>
  const dialog = document.getElementById("welcomeDialog");

  document.getElementById("openDialog").addEventListener("click", function () {
    dialog.showModal();
  });

  document.getElementById("closeDialog").addEventListener("click", function () {
    dialog.close();
  });
</script>
```

Details and Summary

အသုံးပြုသူ click နှိပ်မှ ဖွင့်ပြချင်တဲ့ information တွေအတွက် `<details>` နဲ့ `<summary>` ကို အသုံးပြုနိုင်ပါတယ်။ JavaScript မလိုဘဲ browser က open/close behavior ကို ထိန်းပေးပါတယ်။

```
<details>
  <summary>More information</summary>
  <p>This content is visible after the user opens the details element.</p>
</details>
```

ဒီ Chapter 13 မှာတော့ modern HTML ရဲ့ မရှိမဖြစ် features တွေကို မိတ်ဆက်ပေးခဲ့ပါတယ်။ Layout တွေအတွက် Flexbox နဲ့ Grid ကို ကျွမ်းကျင်အောင် လေ့လာထားရင် responsive website တွေ တည်ဆောက်ရာမှာ အရမ်း အဆင်ပြေပါလိမ့်မယ်။ Images အတွက် `srcset / picture` , interaction အတွက် `dialog / details` , data storage အတွက် Web Storage တို့ကိုပါ သိထားရင် modern website တွေကို ပိုပြီး စနစ်တကျ တည်ဆောက်နိုင်ပါတယ်။