



Ex;braiN

education

Ver 1.1

A Web Developer – I

Copyright @Ex;braiN Education

Basic Web Development Course

Content

1. Introduction to Web Development
2. IDE Installation (VS code)
3. Started with HTML5
4. CSS (Cascading Style Sheets)
5. Learning to Code with JavaScript
6. JS Playground

1. Introduction to Web Development

3 Type of Development

- Frontend Development

User တွေမြင်ရတဲ့ project ရဲ face ဖြစ်ပါတယ်။

- Backend Development

User တွေမြင်ရပဲနောက်ကွွယ်ကနေ အလုပ်လုပ်နေတဲ့ အရာတွေ
ဖြစ်ပါတယ်။ Server, database, etc...,

- Full-stack Development

Frontend and backend နှစ်ခုစလုံး ပေါင်းထားတာကို ခေါ်ပါတယ်။

2. IDE Installation (VS code)

Html Css JavaScript အစရိတ္တော် code တွေကို ရေးနိုင်ပြီး webpageမှာပေါ်စေနိုင်၏ IDE ကိုသုံးရပါတယ်။ IDE တွေ အများကြီးရှိကြပါတယ်။ VS code ကို စုစုလင်လင်ရှိတဲ့အတွက် vs code IDEကို သုံးပါတယ်။

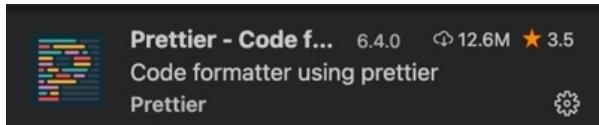
အောက်ပါ လင့်ကနေတဆင့် Download လုပ်ပါ။

<https://code.visualstudio.com/#alt-downloads>

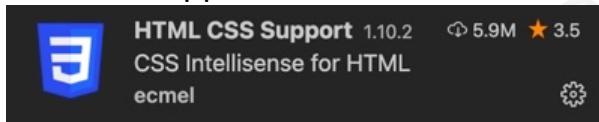
Installed လုပ်ပြီးတဲ့အခါ

Extension မှာ အောက်ပါ extension တွေကို install လုပ်ပေးပါ။

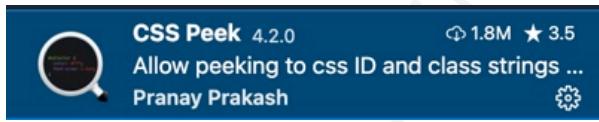
1. Prettier - Code Formatter



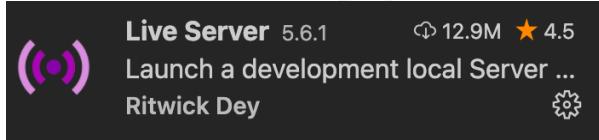
2. Html Css Support



3. Css Peek



4. Live Server

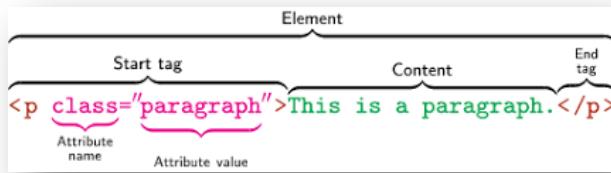


3. Started With HTML5 (HyperText Markup Language)

- 1) Element of HTML
- 2) Tree Structure of HTML
- 3) Block-Level & inline Element
- 4) Heading
- 5) Body Background
- 6) Paragraph & Line Break
- 7) Division
- 8) Lists
- 9) Table
- 10) Text Style
- 11) Link (Local & Global)
- 12) Image
- 13) Form
- 14) Special Characters

1) Elements of HTML

▪Element



Element ဆိုတာ HTMLရဲ့ Statement တစ်ခုအစ အဆုံးကို ခေါပါတယ်။

Open tag ကနေ close tag အထိ ကို Element တစ်လို့သတ်မှတ်ပါတယ်။

▪Tag

Tag ကတော့ Element အစမှ ကိုယ်ပုံဖော်ချင်တဲ့ design ကို Computer

ကနားလည်တဲ့ Machine Language အနေနဲ့ ပြောင်းရေးရပါတယ်။

Image ပုံပေါ်ချင်တယ်ဆို tag

Paragraph နဲ့ရေးချင်တယ်ဆို <p> tag နဲ့ရေးပါတယ်။

များသောအားဖြင့် tag တစ်မှု အဖွင့် tag နဲ့ အပိတ် tag ရေးပေးရပါတယ်။

※Open tag နဲ့ close tag ရောထားတဲ့
 ဆိုတာ မျိုးလည်းရှိပါတယ်။

▪Attribute

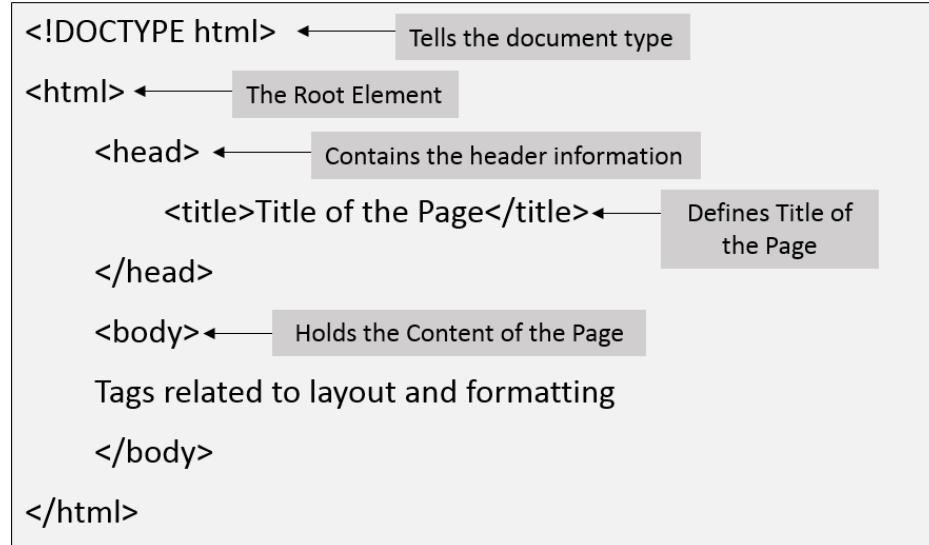
Attribute ကိုတော့ open tag ထဲမှာပဲရေးပါတယ်

Tag တစ်ကို customize လုပ်တဲ့ အခါ အသုံးပြုပါတယ်။

Eg. img tag မှာ ကိုယ်ထည့်ချင်တဲ့ ပုံရဲ့တည်နေရာကို ပြတဲ့အခါ

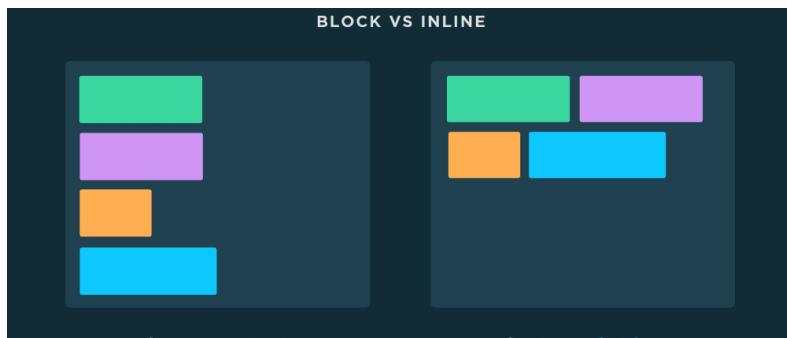
 ဆိုပြီး ရေးပါတယ်။

2) Tree Structure of HTML



- <!DOCTYPE HTML>
HTML ဟာ browserမှာ run မှုဖစ်တာကြောင့် browserကို ဒီfile ဟာ HTML file ဖြစ်ကြောင်းကြညာဖို့လိုပါတယ်။
- <html>
စစချင်း Root element ကိုကြညာပေးရပါတယ်
html tag ထဲမှာ ဆိုရင်တော့ head section and body section tag ရှိပါတယ်။
- <head>
head tag ထဲမှာပဲရေးနိုင်တဲ့ tagတွေကတော့
title (browser ရဲ့ tab bar မှာ ပေးချင်တဲ့ နာမည်)
style(css ကို HTML file ထဲထည့်ရေးခြင်း) ,
script(JavaScript ကို HTML file ထဲထည့်ရေးခြင်း),
link (Css or JavaScript fileများနှင့် ချိတ်ဆက်ခြင်း) , etc..
တွေရေးဖို့အတွက်ဖြစ်ပါတယ်။
- <body>
Body ထဲမှာတော့ ကိုယ့်စိတ်ကြိုက်tag တွေ နဲ့ layout formatting ချေရေးနိုင်ပါတယ်။

3) Block-Level & inline Element



- Block-Level

Block Level က width တခုလုံးအပြည့်ယူတဲ့အတွက် အမြဲnew line နဲ့စပါတယ်။

- Inline

Inline ကတော့ block လိုမဟုတ်ပဲ content တွေကြားထဲညံ့သုံးပါတယ်။

※block level က အံဆွဲတခုနဲ့ဆင်တူပါတယ်။

inline ကတော့ အံဆွဲထဲကပစ္စည်းတခုခြင်းစီနဲ့သဏ္ဌာန်တူပါတယ်။

4) Heading

Heading tag မှာ 1 က 6 အထိရှိပါတယ်

<h1> ...</h1>

<h2>...</h2>

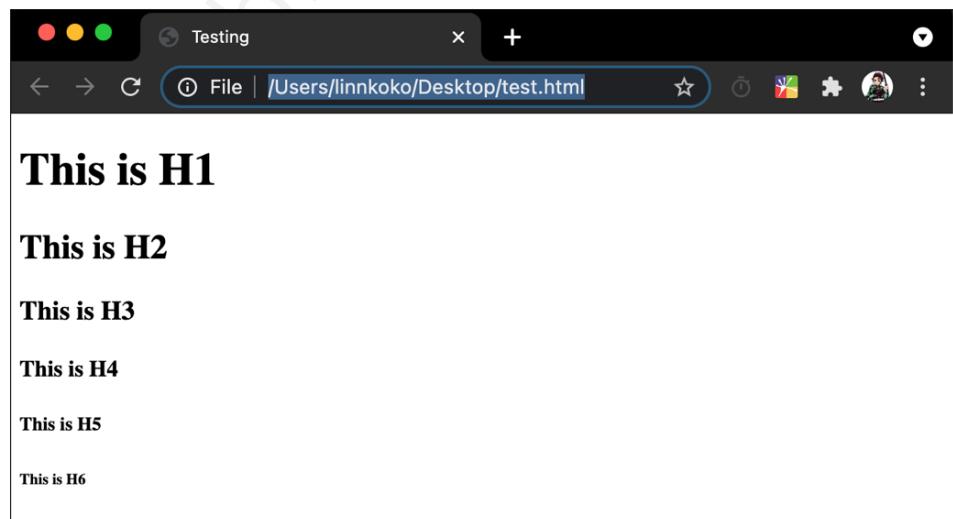
.....

<h6>...</h6>

<h1> ကအကြီးဆုံးဖြစ်ပြီးတော့ <h6> က အသေးဆုံးဖြစ်ပါတယ်။

Demo

```
<!DOCTYPE html>
<html>
  <head>
    <title>Testing</title>
  </head>
  <body>
    <h1>This is H1</h1>
    <h2>This is H2</h2>
    <h3>This is H3</h3>
    <h4>This is H4</h4>
    <h5>This is H5</h5>
    <h6>This is H6</h6>
  </body>
</html>
```



5) Body Background

Body တခုလုံးကို color ဒါမှမဟုတ် image ထည့်ချင်တဲ့အခါမှာသုံးပါတယ်။

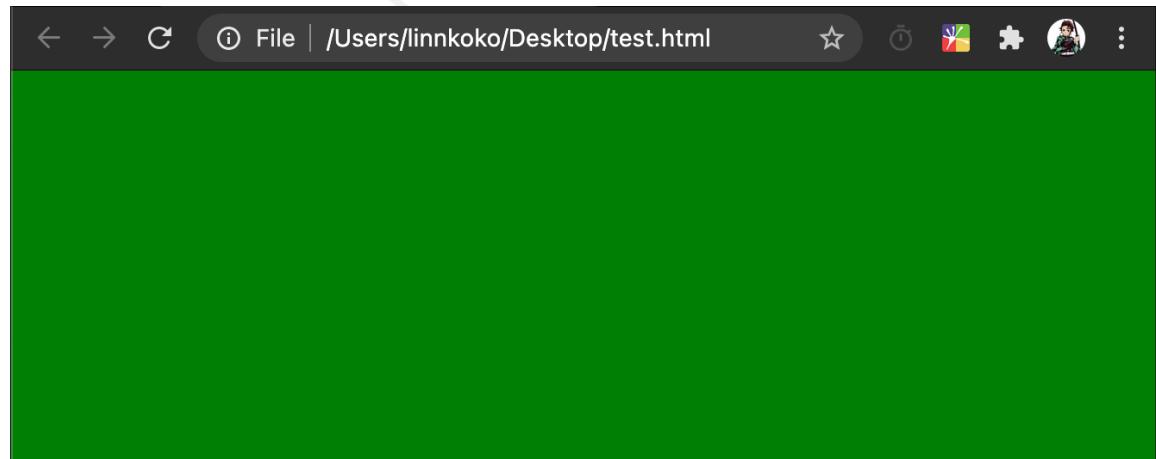
Color အတွက်ဆိုရင် <body bgcolor = "green"> ရေးလိုက်ရင် web page

တခုလုံးက အစိမ်းရောင်ဖြစ်သွားပါမယ်။

ပုံအတွက်ဆိုရင် <body background = "wallpaper.png">ဆိုပြီး
ကိုယ်လိုချင်တဲ့ပုံကို ထည့်လိုပါတယ်။

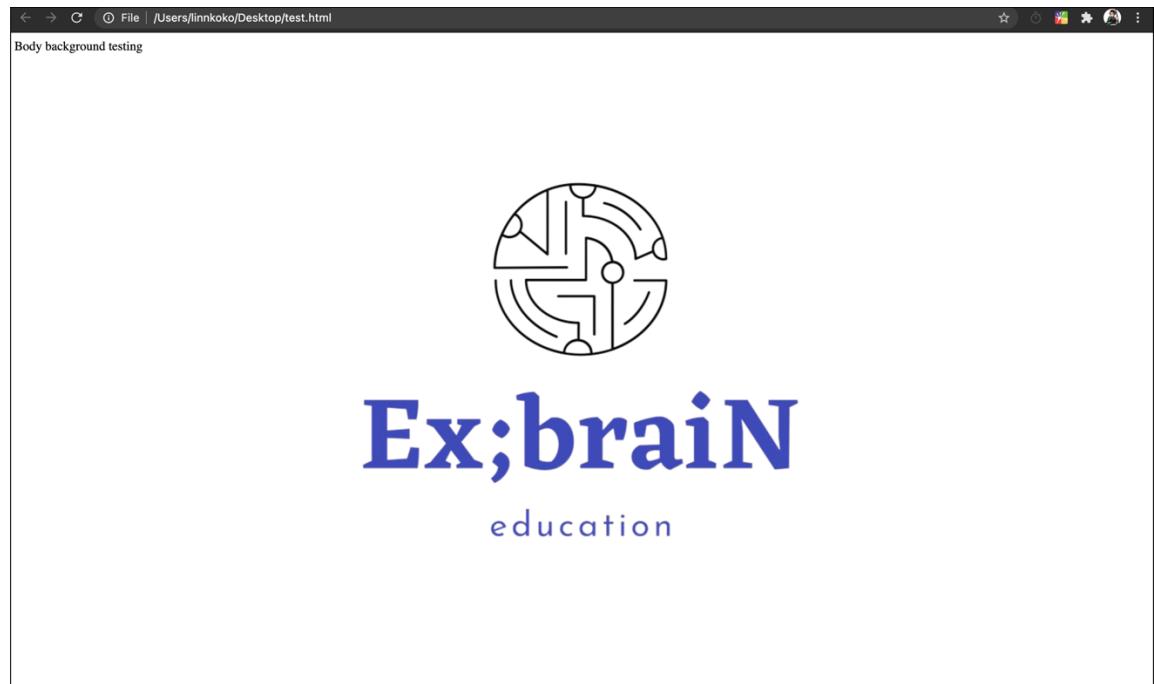
Demo 1

```
<!DOCTYPE html>
<html>
  <head>
    <title>Testing</title>
  </head>
  <body bgcolor="green"></body>
</html>
```



Demo 2

```
<!DOCTYPE html>
<html>
  <head>
    <title>Testing</title>
  </head>
  <body background="logo.png">
    Body background testing
  </body>
</html>
```



6) Paragraph & Line Break

▪Paragraph

- <p>

Paragraph တခုတည်ဆောက်ဖို့အတွက် <p> tag ကိုအသုံးပြုပါတယ်။

Example

```
<p> Paragraph Content </p>
```

<p> tag မှာပါဝင်တဲ့ Attributes ကတော့ align ဖြစ်ပါတယ်။

Alignဆိုတာ Alignment ကိုအတိုကောက်ခေါ်ဆိုတာဖြစ်ပြီးတော့

value မှာတော့

Left : စာသားကို ဘယ်ဘက်ကပ်ပြီးပေါ်စေချင်တဲ့အခါ

Center: စာသားကို အလယ်တည့်တည့်ပေါ်စေချင်တဲ့အခါ

Justify: စာသားကိုပြေပြစ်အောင် ပေါ်စေချင်တဲ့အခါ

Right: စာသားကိုညာဘက်ကပ်ပြီး ပေါ်စေချင်တဲ့အခါ

Example:

```
<p align="right"> Paragraph Content </p>
```

Demo

```
<!DOCTYPE html>
<html>
  <head>
    <title>Testing</title>
  </head>
  <body>
    <p align="right">
      HYPER-TEXT MARKUP LANGUAGE Is the standardised markup language
    </p>
  </body>
</html>
```

▪Line Breaks

- <hr>

<hr> tag ကတေသာ horizontal ruleဖြစ်ပြီးတေသာ paragraph တခုကို
မျဉ်းကြောင်းနှင့်ပိုင်းတဲ့အခါဖြစ်ဖြစ် web page ပေါ်မှာရေပြင်ညီ မျဉ်းကြောင်း
ပေါ်ချင်တဲ့အခါမှာသုံးပါတယ်။ hr မှာclose tag မရှိပါဘူး။

သူ့ရဲ့အသုံးများတဲ့ Attributes တွေကတေသာ

Size: hr ရဲ့အမြင် (pixel or percentage)

Width: hr ရဲ့အလျား (pixel or percentage)

Color: hr ရဲ့အရောင် , **Align:** hr ရဲ့Alignment

Demo

```
<!DOCTYPE html>
<html>
  <head>
    <title>Testing</title>
  </head>
  <body>
    <p align="right">
      HYPER-TEXT MARKUP LANGUAGE Is the standardised markup language
    </p>
    <hr size=""4px"" width=""12px"" align=""center"" color=""blue"" />
  </body>
</html>
```

-

New line ဆင်းဖို့အတွက်သုံးတာဖြစ်ပါတယ်။
Open tag နဲ့ close tag ရောရေးတဲ့ tag တခုဖြစ်ပါတယ်။

7) Division

- <div>
အသုံးအများဆုံးလို့ပြောရမယ် tag တခုပါ။<div> tag ဟာ ရှိုးရှင်းဆုံး
အကန်နှစ်တော့ view တွေပိုင်းတာတွေ အတွက်အသုံးပြုပါတယ်။
-
Inline elements တခုဖြစ်ပါတယ်။<text> တွေကြားမှာနေရာ အနည်းငယ်
ယူပြီးစာသားဖြစ်ဖြစ် style ချိန်တဲ့အခါ ဖြစ်ဖြစ်အသုံးပြုပါတယ်။

8) Lists

- Unordered List

list တွေကို shape တွေနဲ့ပြချင်တဲ့အခါအသုံးပြပါတယ်။

 <dl> တွေအားလုံးအတွက် tag ကို သုံးရပါတယ်။

Style ကို အသုံးပြပြီး List တွေshape မျိုးစံနဲ့ပြနိုင်ပါတယ်။

Example:

```
<ul style="list-style-type:circle">
    <li>Coffee</li>
    <li>Tea</li>
    <li>Milk</li>
</ul>
```

```
<ul style="list-style-type:disc">
    <li>Coffee</li>
    <li>Tea</li>
    <li>Milk</li>
</ul>
```

```
<ul style="list-style-type:square">
    <li>Coffee</li>
    <li>Tea</li>
    <li>Milk</li>
</ul>
```

Demo

```
<!DOCTYPE html>
<html>
    <head>
        <title>Testing</title>
    </head>
    <body>
        <ul style="list-style-type: circle">
            <li>Coffee</li>
            <li>Tea</li>
            <li>Milk</li>
        </ul>

        <ul style="list-style-type: disc">
            <li>Coffee</li>
            <li>Tea</li>
        </ul>
    </body>
</html>
```

```

<li>Milk</li>
</ul>

<ul style="list-style-type: square">
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ul>
</body>
</html>

```

- Coffee
- Tea
- Milk

- Coffee
- Tea
- Milk

- Coffee
- Tea
- Milk

- Ordered List
 နံပါတ်စဉ်အလိုက် list တွေကိုပြချင်တဲ့အခါအသုံးပြုပါတယ်။
 ul လိုမျိုး number style တွေလည်းရှိပါတယ်။

Example:

```

<ol style="list-style-type:upper-roman">
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ol>

```

```

<ol style="list-style-type:lower-alpha">
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ol>

```

Demo

```
<!DOCTYPE html>
<html>
  <head>
    <title>Testing</title>
  </head>
  <body>
    <ol style="list-style-type: upper-roman">
      <li>Coffee</li>
      <li>Tea</li>
      <li>Milk</li>
    </ol>

    <ol style="list-style-type: lower-alpha">
      <li>Coffee</li>
      <li>Tea</li>
      <li>Milk</li>
    </ol>
  </body>
</html>
```

I. Coffee
II. Tea
III. Milk

a. Coffee
b. Tea
c. Milk

- <dl> Description List
Description listတွေအတွက်အသုံးပြုပါတယ်။
<dl>
 <dt>Coffee</dt>
 <dd>Black hot drink</dd>
 <dt>Milk</dt>
 <dd>White cold drink</dd>
</dl>
- <dt> description ရဲ့title အတွက်ပါ။
- <dd> description အတွက်tag ဖြစ်ပါတယ်။

Demo

```
<!DOCTYPE html>
<html>
  <head>
    <title>Testing</title>
  </head>
  <body>
    <dl>
      <dt>Coffee</dt>
      <dd>Black hot drink</dd>
      <dt>Milk</dt>
      <dd>White cold drink</dd>
    </dl>
  </body>
</html>
```

Coffee

 Black hot drink

Milk

 White cold drink

9) Table

Table တည်ဆောင်ဖို့ သို့မဟုတ် division တွေပိုင်းဖို့အတွက်အသုံးပြုပါတယ်။
သူမှာဆိုရင်

- <table> tag ကို စစချင်းအသုံးပြုရပါတယ်။
- <tr> row တခုစဖွင့်တဲ့ အခါအသုံးပြုပါတယ်။
- <td> row တခုမှာ ရှိတဲ့ column တခုစီအတွက်သုံးပါတယ်။
- <th> အခြေခံအားဖြင့် td နဲ့အတူတူဖြစ်ပြီးတော့ ကွဲပြားတဲ့အချက်ကတော့ header column အဖြစ်သုံးပါတယ်။ th tag အတွင်းမှာရေးလိုက်တဲ့ စာသားတွေက Bold Text အနေနဲ့ပေါ်လာမှာဖြစ်ပါတယ်။

※column တခုကို ကို cell တခုလို့ခေါ်ပါတယ်။

table tag မှာအသုံးများတဲ့ Attributes တွေကတော့

border: table တခုလုံးရဲ့ border အထူကိုသတ်မှတ်တဲ့ အခါသုံးပါတယ်။

Example:

<table border="3">

width: table တခုလုံးရဲ့ width ကိုသတ်မှတ်ပေးပါတယ်။ pixel or percentage နဲ့သတ်မှတ်ပါတယ်။

height: table တခုလုံးရဲ့ height ကိုသတ်မှတ်ပေးပါတယ်။ pixel or percentage နဲ့သတ်မှတ်ပါတယ်။

align: table ကို webpage ရဲ့ left, center, right လိုသလို alignment ချိန်ဖို့အတွက် အသုံးပြုပါတယ်။

bgcolor: table ရဲ့ background color ခြောင်းဖို့အတွက်အသုံးပြုပါတယ်။

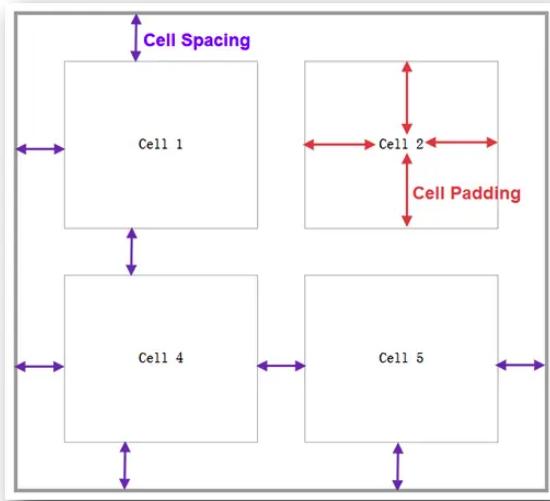
Example:

<table bgcolor="red">

ဆိုရင် table တခုလုံးရဲ့ background ဟာ အနီရောင်ဖြစ်သွားပါလိမ့်မယ်။

cellspacing: table ထဲက cell နှစ်ခုကြားအကွာအဝေးကို သတ်မှတ်ပါတယ်။

cellpadding: cell border နဲ့ cellထဲကစာသားရဲ့အကွာအဝေးကို သတ်မှတ်ပါတယ်။



Demo 1 (Default)

```
<!DOCTYPE html>
<html>
<head>
<title>Testing</title>
</head>
<body>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Age</th>
</tr>
<tr>
<td>Jill</td>
<td>Smith</td>
<td>50</td>
</tr>
<tr>
<td>Eve</td>
<td>Jackson</td>
<td>94</td>
</tr>
</table>
</body>
</html>
```

Firstname Lastname Age

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94

Demo 2 (with border)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Testing</title>
  </head>
  <body>
    <table border="2">
      <tr>
        <th>Firstname</th>
        <th>Lastname</th>
        <th>Age</th>
      </tr>
      <tr>
        <td>Jill</td>
        <td>Smith</td>
        <td>50</td>
      </tr>
      <tr>
        <td>Eve</td>
        <td>Jackson</td>
        <td>94</td>
      </tr>
    </table>
  </body>
</html>
```

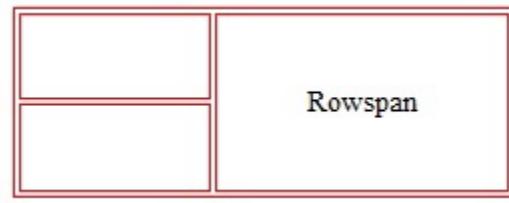
Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94

Demo 3

```
<!DOCTYPE html>
<html>
<head>
<title>Testing</title>
</head>
<body>
<table border="2" cellpadding="10" cellspacing="5">
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Age</th>
</tr>
<tr>
<td>Jill</td>
<td>Smith</td>
<td>50</td>
</tr>
<tr>
<td>Eve</td>
<td>Jackson</td>
<td>94</td>
</tr>
</table>
</body>
</html>
```

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94

▪ Rowspan



ပုံမှန် Row တခုကိုအခြား row တွင့် Merge လုပ်ချင်တဲ့ အခါသုံးပါတယ်။

rowspan ကို စကြညာလိုက်တဲ့ cell ကစလို သူအောက်ကcell အားလုံးကို

ပေးလိုက်တဲ့ တန်ဖိုးအတိုင်းMergeလုပ်ပါတယ်။

ဥပမာ tableမှာ row3ခုရှိပြီးပထမဆုံးrowမှာ rowspan = "3" လိုက်ပေးလိုက်မယ်ဆိုရင်

သူအပါအဝင် အောက်ကrow တွေ အကုန်တစုတည်း Mergeဖြစ်နေပါလိမ့်မယ်။

Demo 1

```
<!DOCTYPE html>
<html>
<head>
<title>Testing</title>
</head>
<body>
<table border="2" cellpadding="10" cellspacing="5">
<tr>
<th>Month</th>
<th>Savings</th>
<th rowspan="3">Empty Savings for holiday!</th>
</tr>
<tr>
<td>January</td>
<td>$100</td>
</tr>
<tr>
<td>February</td>
<td>$80</td>
</tr>
</table>
</body>
</html>
```

Month	Savings	
January	\$100	Empty Savings for holiday!
February	\$80	

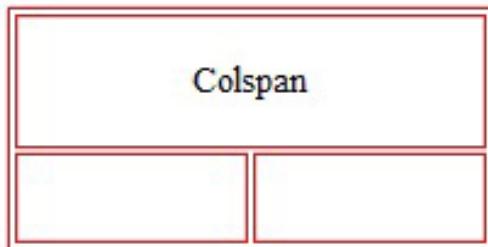
ပထမဆုံး row မှာ rowspan = "2" လိုက်ပေးလိုက်မယ်ဆိုရင်တော့အောက်ဆုံး cell ကို Merge လုပ်မှာမဟုတ်ပါဘူး

Demo 2

```
<!DOCTYPE html>
<html>
  <head>
    <title>Testing</title>
  </head>
  <body>
    <table border="2" cellpadding="10" cellspacing="5">
      <tr>
        <th>Month</th>
        <th>Savings</th>
        <th rowspan="2">Empty Savings for holiday!</th>
      </tr>
      <tr>
        <td>January</td>
        <td>$100</td>
      </tr>
      <tr>
        <td>February</td>
        <td>$80</td>
        <td>$30</td>
      </tr>
    </table>
  </body>
</html>
```

Month	Savings	
January	\$100	
February	\$80	\$30
Empty Savings for holiday!		

- Colspan



Colspan ကတေသူ columnတွေကို Mergeလုပ်တာပါ။ Rowspan နဲ့တူတူပါပဲ။
※ Rowspan ရော့ Colspan ရော့ကို <td> or <th> tag ထဲမှာပဲရေးရပါတယ်။

Demo

```
<!DOCTYPE html>
<html>
<head>
    <title>Testing</title>
</head>
<body>
    <table border="2" cellpadding="10" cellspacing="5">
        <tr>
            <th>Month</th>
            <th>Savings</th>
        </tr>
        <tr>
            <td>January</td>
            <td>$100</td>
        </tr>
        <tr>
            <td>February</td>
            <td>$80</td>
        </tr>
        <tr>
            <td colspan="2">Sum: $180</td>
        </tr>
    </table>
</body>
</html>
```

```

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

```

Month	Savings
January	\$100
February	\$80
Sum: \$180	

10) Text Style

-
စာလုံးတွေကို Bold tag ဖြစ်ချင်တဲ့အခါသုံးပါတယ်။
- or <i>
စာလုံးအစောင်း italic text များရေးတဲ့အခါသုံးပါတယ်။
- <u>
စာလုံးကို underline တားတဲ့အခါ အသုံးပြုပါတယ်။
- <sup>
 x^3 လို့မျိုး 3 လို့ Power တင်ရေးတဲ့ စာလုံးတွေမှာ super tag
ကိုသုံးပြီးရေးရပါတယ်။
- <sub>
 H_2O စာလုံးမှာဆိုရင် 2 ကိုရေးတဲ့အခါမျိုးမှာ sub tag ကို သုံးပါတယ်။

Demo

```
<!DOCTYPE html>
<html>
<head>
<title>Testing</title>
</head>
<body>
<b>Bold Text</b> <br/>
<i>italic text </i><br/>
 $2^{<sup>2</sup>} \text{ is } 4$  <br/>
H<sub>2</sub>O = water <br/>
</dl>
</body>
</html>
```

Bold Text

italic text

2^2 is 4

H₂O = water

11) Link (Local & Global)

- <a>
anchor tag လိုခေါပါတယ်။ a tag က Same Document တွေမှာ ရှိတဲ့ tag တွေကို ချိတ်ပေးတဲ့အခါ ဒါမှာမဟုတ် တခြား document တွေ site တွေကို ချိတ်ပေးတဲ့အခါ အသုံးပြုပါတယ်။
အသုံးများတဲ့ Attribute ကတော့
name : same document မှာရောက်စေချင်တဲ့ target point
href: ချိတ်ချင်တဲ့ target name or other document လမ်းကြောင်း or web address

■ Link with same document

အရင်ဆုံး target point ကို သတ်မှတ်ပေးရပါတယ်။ target point မှာ name သတ်မှတ်ပေးရပါတယ်။
ဒုတိယအနေနဲ့ ခေါမယ် <a> tag မှာ သတ်မှတ်ထားတဲ့ #target name ကို href ကိုသုံးပြီး လုမ်းချေဖို့ပါတယ်။

Demo

```
<!DOCTYPE html>
<html>
  <head>
    <title>Testing</title>
  </head>
  <body>
    <a href="#section2">Go to Section 2</a>
    <p>
      In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since. "Whenever you feel like criticizing any one," he told me, "just remember that all the people in this world haven't had the advantages that you've had." He didn't say any more, but we've always been unusually communicative in a reserved way, and I understood that he meant a great deal more than that. In consequence, I'm inclined to reserve all judgments, a habit that has opened up many curious natures to me and also made me the victim of not a few veteran bores. The abnormal mind is quick to detect and attach itself to this quality when it appears in a normal person, and so it came about that in college I was unjustly accused of being a politician, because I was privy to the secret griefs of wild, unknown men. Most of the confidences were unsought-frequently I have feigned sleep, preoccupation, or a hostile
```

levity when I realized by some unmistakable sign that an intimate revelation was quivering on the horizon; for the intimate revelations of young men, or at least the terms in which they express them, are usually plagiaristic and marred by obvious suppressions. Reserving judgments is a matter of infinite hope. I am still a little afraid of missing something if I forget that, as my father snobbishly suggested, and I snobbishly repeat, a sense of the fundamental decencies is parcelled out unequally at birth. And, after boasting this way of my tolerance, I come to the admission that it has a limit. Conduct may be founded on the hard rock or the wet marshes, but after a certain point I don't care what it's founded on. When I came back from the East last autumn I felt that I wanted the world to be in uniform and at a sort of moral attention forever; I wanted no more riotous excursions with privileged glimpses into the human heart. Only Gatsby, the man who gives his name to this book, was exempt from my reaction-Gatsby, who represented everything for which I have an unaffected scorn.

</p>

Section 2

<p>This is Section 2</p>

</body>

</html>

[Go to Section 2](#)

In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since. "Whenever you feel like criticizing any one," he told me, "just remember that all the people in this world haven't had the advantages that you've had." He didn't say any more, but we've always been unusually communicative in a reserved way, and I understood that he meant a great deal more than that. In consequence, I'm inclined to reserve all judgments, a habit that has opened up many curious natures to me and also made me the victim of not a few veteran bores. The abnormal mind is quick to detect and attach itself to this quality when it appears in a normal person, and so it came about that in college I was unjustly accused of being a politician, because I was privy to the secret griefs of wild, unknown men. Most of the confidences were unsought-frequently I have feigned sleep, preoccupation, or a hostile levity when I realized by some unmistakable sign that an intimate revelation was quivering on the horizon; for the intimate revelations of young men, or at least the terms in which they express them, are usually plagiaristic and marred by obvious suppressions. Reserving judgments is a matter of infinite hope. I am still a little afraid of missing something if I forget that, as my father snobbishly suggested, and I snobbishly repeat, a sense of the fundamental decencies is parcelled out unequally at birth. And, after boasting this way of my tolerance, I come to the admission that it has a limit. Conduct may be founded on the hard rock or the wet marshes, but after a certain point I don't care what it's founded on. When I came back from the East last autumn I felt that I wanted the world to be in uniform and at a sort of moral attention forever; I wanted no more riotous excursions with privileged glimpses into the human heart. Only Gatsby, the man who gives his name to this book, was exempt from my reaction-Gatsby, who represented everything for which I have an unaffected scorn. If personality is an unbroken series of successful gestures, then there was something gorgeous about him, some heightened sensitivity to the promises of life, as if he were related to one of those intricate machines that register earthquakes ten thousand miles away. This responsiveness had nothing to do with that flabby impressionability which is dignified under the name of the "creative temperament"-it was an extraordinary gift for hope, a romantic readiness such as I have never found in any other person and which it is not likely I shall ever find again. No-Gatsby turned out all right at the end; it is what preyed on Gatsby, what foul dust floated in the wake of his dreams that temporarily closed out my interest in the abortive sorrows and short-winded elations of men. In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since.

I shall ever find again. No-Gatsby turned out all right at the end; it is what preyed on Gatsby, what foul dust floated in the wake of his dreams that temporarily closed out my interest in the abortive sorrows and short-winded elations of men. In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since. "Whenever you feel like criticizing any one," he told me, "just remember that all the people in this world haven't had the advantages that you've had." He didn't say any more, but we've always been unusually communicative in a reserved way, and I understood that he meant a great deal more than that. In consequence, I'm inclined to reserve all judgments, a habit that has opened up many curious natures to me and also made me the victim of not a few veteran bores. The abnormal mind is quick to detect and attach itself to this quality when it appears in a normal person, and so it came about that in college I was unjustly accused of being a politician, because I was privy to the secret griefs of wild, unknown men. Most of the confidences were unsought-frequently I have feigned sleep, preoccupation, or a hostile levity when I realized by some unmistakable sign that an intimate revelation was quivering on the horizon; for the intimate revelations of young men, or at least the terms in which they express them, are usually plagiaristic and marred by obvious suppressions. Reserving judgments is a matter of infinite hope. I am still a little afraid of missing something if I forget that, as my father snobbishly suggested, and I snobbishly repeat, a sense of the fundamental decencies is parcelled out unequally at birth. And, after boasting this way of my tolerance, I come to the admission that it has a limit. Conduct may be founded on the hard rock or the wet marshes, but after a certain point I don't care what it's founded on. When I came back from the East last autumn I felt that I wanted the world to be in uniform and at a sort of moral attention forever; I wanted no more riotous excursions with privileged glimpses into the human heart. Only Gatsby, the man who gives his name to this book, was exempt from my reaction-Gatsby, who represented everything for which I have an unaffected scorn. If personality is an unbroken series of successful gestures, then there was something gorgeous about him, some heightened sensitivity to the promises of life, as if he were related to one of those intricate machines that register earthquakes ten thousand miles away. This responsiveness had nothing to do with that flabby impressionability which is dignified under the name of the "creative temperament"-it was an extraordinary gift for hope, a romantic readiness such as I have never found in any other person and which it is not likely I shall ever find again. No-Gatsby turned out all right at the end; it is what preyed on Gatsby, what foul dust floated in the wake of his dreams that temporarily closed out my interest in the abortive sorrows and short-winded elations of men.

Section 2

This is Section 2

■ Link with other document

တွေး file တွေကို Link ချိတ်တဲ့ အခါမှာတော့ href မှာ click
 နှင့်လိုက်တဲ့အခါသွားစေချင်တဲ့ Document directory
 ကိုရေးပေးရပါတယ်။

Demo

```
<!DOCTYPE html>
<html>
  <head>
    <title>Testing</title>
  </head>
  <body>
    <a href="test2.html" target="_blank">Go to Test2.</a>
  </body>
</html>
```

[Go to Test2.](#)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Test2</title>
  </head>
  <body>
    Welcome to test2 page.
  </body>
</html>
```

Welcome to test2 page.

■ Link to website

Webpage တွေကို link ချိတ်တဲ့အခါ href မှာ webaddress

ကိုရေးပေးရပါတယ်။

တွဲသံဃာရတဲ့ Attribute ကတော့ target ပါ။

သူရဲ့value တွေမှာဆိုရင်

_blank = webpage ပေါ်တဲ့ အခါ New tab အနေနဲ့ပေါ်လာမှာဖြစ်ပါတယ်။

_self = လက်ရှုpageမှာပဲ webpage ကပေါ်လာမှာဖြစ်ပါတယ်။

_parent = frame တွေသုံးထားတဲ့ webpage တွေမှာ ဆိုရင် <a> tag
သုံးထားတဲ့ frame ရဲအပြင်ဘက်က frame မှာ webpage
ကပေါ်လာမှာဖြစ်ပါတယ်။

_top = _parent နဲ့ဆင်တူပြီးတော့ သူတော့ ထိပ်ဆုံး main frame မှာ
ပေါ်မှာဖြစ်ပါတယ်။

framename = webpage ကို ကိုယ်ပေါ်စေချင်တဲ့ frame ရဲ့
nameကိုထည့်ပြီးသုံးနိုင်ပါတယ်။

Demo

```
<!DOCTYPE html>
<html>
  <head>
    <title>Testing</title>
  </head>
  <body>
    <a href="https://www.apple.com" target="_blank">Visit Apple!</a>
  </body>
</html>
```



Visit Apple!

12) Image

-
Webpage ထဲ ပုံထည့်ချင်တဲ့ အခါ tag ကိုသုံးရပါတယ်။
သူ့ရဲ့Attributeတွေကတော့
src: ပေါ်စေချင်တဲ့ ပုံရဲ့တည်နေရာ directory
alt: ပုံရဲ့name လို့
ပြောလို့ရပါတယ်။ပုံကိုနာမည်ပေးလိုက်တာဖြစ်ပါတယ်။အကြောင်းတခုခုကြောင့်
ပုံerror တတ်ပြီးမဖြင့်ရတဲ့အခါ altမှာပေးလိုက်တဲ့ပုံရဲ့name ကပေါ်လာပါတယ်။
width: ပုံရဲအကျယ် ,**height:**ပုံရဲအမြင့်

Example:

```

```

13) Form

from ဟာ htmlရဲ့
အသက်လို့ပြောလို့ရပါတယ်။ဘာလို့အရေးကြီးတာလည်းဆိုတော့ အပေါ်မှာ
လေ့လာခဲ့တဲ့ tag တွေအားလုံးဟာ user ကို information ပေးတာဖြစ်ပြီးတော့
form tag ကတော့ user ဆီက information ယူတဲ့အခါသုံးရတဲ့ tag
ဖြစ်လိုပါ။form tag ကို သုံးပြီး server side language တွေဆီ data ထွေ
transferလုပ်ပြီး process တခုလုံးလည်ပတ်စေနိုင်ပါတယ်။ form tag ကို
တနည်းအားဖြင့် data entry လို့လည်းသတ်မှတ်လို့ရပါတယ်။

- <form>
Form tag ထဲမှာရေးရတဲ့ tagတွေ အများကြီးရှိပါတယ်။ form tag ကို server side
တွေဆီ data transfer လုပ်ဖို့လိုပြီဆိုတဲ့အခါမှာ စပြီး open tag စရေးပါတယ်။
Form ရဲ့attributes ကတော့
action : server side ရဲ့directory

Form tag ၏ open tag နဲ့ close tag ထဲမှာ ရေးရတဲ့ form element တွေကတော့

- **<input>**

Input tag ကတော့ user ဆိုကနေ typing data တွေကိုလက်ခံဖို့အတွက်
အသုံးပြုပါတယ်။ သူ့၏ attributes တွေကတော့

value: ပေါ်စေချင်တဲ့ စာသား

type: ကိုယ့်ရဲ့ input ကို လိုချင်တဲ့ ပုံစံ Design နဲ့ setting ချေပေးနိုင်ပါတယ်။

type မှာထည့်နိုင်တဲ့ value တွေကတော့

*button
checkbox
color
date
datetime-local
email
file
hidden
image
month
number
password
radio
range
reset
search
submit
tel
text (default)
time
url
week*

အသုံးများတဲ့ type တွေကတော့

button:

Click နိုင်တဲ့ အခါ လုပ်ဆောင်မယ် action တွေအတွက်သုံးပါတယ်။

Example:

```
<input type="button" value="Click Me"/>
```

checkbox:

one or more data တွေကိုရွှေးချယ်ခိုင်းတဲ့ အခါသုံးပါတယ်။

※ သတိထားရမှာကတော့ checkbox ကို နာမည်ပေး

(name) ရမှာပဲဖြစ်ပါတယ်။ပေးလိုက်တဲ့ name

တွေကတူညီနေရပါမယ်။

စစချင်း ရွှေးချယ်ထားချင်ဆိုရင် checked ကို သုံးပေးရပါတယ်။

Example:

```
<input type="checkbox" name="vehicle1" value="Bike">
<input type="checkbox" name="vehicle2" value="Car">
<input type="checkbox" name="vehicle3" value="Boat" checked>
```

file:

server side ကို ပုံတွေ file data တွေပို့တဲ့ အခါသုံးပါတယ်။

hidden:

သူကတော့ webpage မှာ ပေါ်လာမှာ မဟုတ်ပါဘူး။ data တွေကို user မမြင်စေပဲ hidden function နဲ့ server side ဆိုပို့ပေးမှာဖြစ်ပါတယ်။

Example:

```
<input type="hidden" value="secretdata">
```

number:

user ကို ကိုန်းဂဏန်းပဲရိုက်စေချင်တဲ့ အခါ သုံးပါတယ်။ ဥပမာ (amount တို့ quantity တို့) number အတွက်ပဲထည့်နိုင်တဲ့ Attribute ကတော့ Min (အနည်းဆုံး value) Max (အများဆုံး value)

Example:

```
<input type="number" min="1" max="9">
```

ဆိုရင် 1 ကနေ 9 ထိ ပဲထည့်နိုင်ပါတယ်။

password:

password ဆိုတဲ့အတိုင်း user ရှိက်လိုက်တဲ့ data တိုင်းကို မမြင်ရအောင် •
အနေနဲ့ပြပေးမှာဖြစ်ပါတယ်။

Example:

```
<input type="password" value="secretdata">
```

reset:

button အနေနဲ့ပေါ် ပြီးတော့ form tag အစကနေ အဆုံးကြားထဲ
ရေးထားတဲ့ tag တွေရဲ့ user ရှိက်ထည့်ထားတဲ့ data တွေဟာ default state
ဖြစ်သွားပါလိမ့်မယ်။

Example:

```
<input type="reset" value="Default">
```

submit:

user ရဲ့ data တွေကို server side ဆို ပို့မယ်လို့ command
ပေးတဲ့ပုံစံနဲ့ပါတယ်။ Button ပုံစံနဲ့ပြပြီးတော့ နှိပ်လိုက်တဲ့အခါ form ရဲ့
action မှာပေးထားတဲ့ လမ်းကြောင်းဆီ data တွေပို့ပေးမှာဖြစ်ပါတယ်။

Example:

```
<input type="submit" value="Sent to Server">
```

radio:

radio ကတော့ only one data ကို
ရွှေးချယ်ချင်တဲ့အခါသုံးပါတယ်။သတိထားရမှာကတော့ name
တွေတူညီနေရမှာဖြစ်ပါတယ်။

Example:

```
<input type="radio" name="gender" value="male">  
<input type="radio" name="gender" value="female">  
<input type="radio" name="gender" value="other">
```

text:

Default ဖြစ်ပြီးတော့ ကြိုက်တဲ့ character ထည့်လို့ရပါတယ်။

Example:

```
<input type="text">
```

value: type က button ဆိုရင် button ပေါ်ကစာသားဖြစ်ပြီး text,number,password တို့လို typing value တွေအတွက်ဆိုတော့ box ထဲကစာသားဖြစ်ပါမယ်။

accept: file type only အတွက်ပဲသုံးနိုင်ပါတယ်။ လက်ခံနိုင်မယ့် file type တွေကိုပဲ သတ်မှတ်ပေးထားနိုင်ပါတယ်။ထည့်နိုင်တဲ့ value တွေကတော့ *audio/*, audio file* တွေအားလုံး
video/, video file* တွေအားလုံး
image/, image file* တွေအားလုံး
File_extension ကိုယ်သတ်မှတ်ချင်တဲ့ *file extension*
(example: .jpg,.png,.mp4,etc..,)

Example:

```
<input type="file" accept="image/*">
```

autofocus: webpage စစချင်းပေါ်လာတဲ့အခါ သို့မဟုတ် reload လုပ်တဲ့ အခါ mouse cursor ကိုသတ်မှတ်ပေးချင်တဲ့အခါ သုံးပါတယ်။

Example:

```
<input type="text" autofocus>
```

checked: အပေါ်မှာရှင်းပြခဲ့သလိုပဲ checkbox နဲ့ radio type တွေမှာ စစချင်းရွှေးချယ်ပေးထားချင်တဲ့အခါမှာသုံးပါတယ်။

disabled: User ကိုပေးမရိုက်စေချင်တဲ့ အခါ သုံးပါတယ်။ Cursor ချမှတ်ပါဘူး။disabledသုံးလိုက်တဲ့အခါ data ဟာ submit လုပ်မယ့် server Side ဆီမပို့နိုင်ပါဘူး။

Example:

```
<input type="text" value="Don't Input this text box" disabled>
```

min: type မှုန်များသုံးတဲ့အခါ အနည်းဆုံးထည့်နိုင်မည့်တန်ဖိုး

max: type မှုန်များသုံးတဲ့အခါ အများဆုံးထည့်နိုင်မည့်တန်ဖိုး

maxlength:စာလုံးရေကို ကန့်သတ်ချင်တဲ့အခါသုံးပါတယ်။

Defaultအနေနဲ့ကတော့ 524288ဖြစ်ပါတယ်။

multiple: input type file နဲ့ email တွေအတွက်အသုံးပြုပါတယ်။file အတွက်ဆို တခုထက်ပိုတဲ့ file တွေ upload လုပ်ချင်တဲ့အခါemail အတွက်ဆို comma နဲ့ တခုထက်ပိုတဲ့ email တွေကို ထည့်နိုင်ပါတယ်။

Example:

```
<input type="file" multiple>
```

name: input tagကိုနာမည်ပေးတဲ့ attribute ဖြစ်ပါတယ်။css js တို့လေ့လာတဲ့အခါ ပြန်ရှင်းပြသွားပါမယ်။

placeholder: User တွေကို hint ပေးတဲ့ attribute တခုဖြစ်ပါတယ်။ ဥပမာ ဒီ input ဟာ ဘယ်လိုdata တွေကိုလက်ခံတဲ့ဆိုတာကို userဆီ ပြောပြချင်တဲ့ အခါမျိုးမှာသုံးပါတယ်။Mouse cursor ချလိုက်တဲ့အခါ စာသားကပျက်သွားမှာ ဖြစ်ပြီး ဘာမှ data မှုမရှိပဲ cursor ဖယ်လိုက်တဲ့အခါ ပြန်ပေါ်လာမှာဖြစ်ပါတယ်။

readonly: Value ကိုကြိုးထည့်ထားပြီးတော့ Userကို ရိုက်ခွင့်မပေးချင်တဲ့အခါသုံးပါတယ်။Disabledနဲ့ ကွာတဲ့အချက်တွေကတော့ server side ဆီ data ပို့နိုင်ပြီး Mouse Cursor လည်း focus လုပ်နိုင်ပါတယ်။

required: User ကို မရှိက်မဖြစ် ရိုက်စေချင်တဲ့အခါ အသုံးပြုပါတယ်။required attitude ဟာ input type submit နဲ့တဲ့သုံးပေးရပါတယ်။ User က data မထည့်ပဲ

submit နှင့်လိုက်တဲ့အခါ error or alert ဖြစ်ပြီး server side or next transaction ကို မရောက်ပဲ လက်ရှိ page မှာပဲ ရှိနေပါလိမ့်မယ်။

Example:

```
<input type="text" required>  
<input type="submit" value="OK">
```

size: input tag ရဲ့ width ဖြစ်ပြီးတော့ User ကို ပြစ်ချင်တဲ့ scroll မလုပ်ပဲပြစ်ချင်တဲ့ အရည်အတွက် ဖြစ်ပါတယ်။ type အနေနဲ့ကတော့ text, search, tel, url, email, and password တွေမှာပဲသုံးနိုင်ပါတယ်။

- **<label>**

input tag အတွက် label ထိုးပေးတဲ့အခါသုံးပါတယ်။

Attribute ကတော့

for: for ကိုသုံးလိုက်မယ်ဆိုရင် input tag မှာ id ကိုမဖြစ်မနေ ထည့်ပေးရပါတယ်။ idမှာပေးထားတဲ့ နာမည်ကို forမှာ ထည့်ပေးလိုက်ချင်းဖြင့် label ကို clickနိုင်တဲ့ အခါ input tag ထဲက cursor ကျလာမှာဖြစ်ပါတယ်။

- **<select>**

drop down list လိုလည်းခေါ်ပါတယ်။ List တွေထဲက

တခုကိုရွေးချယ်ချင်တဲ့အခါ သုံးပါတယ်။

Data တခုချင်းဆီ အတွက် option tag ကို သုံးပေးရပါတယ်။

Attribute တွေ အနေနဲ့ကတော့

autofocus,
disabled,
multiple,
required,
name,
size တို့ရှိပါတယ်။

Example:

```

<select id="cars">
<option value="volvo">Volvo</option>
<option value="saab">Saab</option>
<option value="opel">Opel</option>
<option value="audi">Audi</option>
</select>

```

- **<textarea>**

text area ကတော့ input type ကို text နဲ့သုံးတာကို category
 တခုထက်ခွဲလိုက်တာဖြစ်ပါတယ်။ Information
 များများရေးလိုလိုတဲ့အခါမှာသုံးပါတယ်။ ကွဲပြားတာကတော့ သူမှာ
 လိုင်းအကြောင်းရေးများများ ရေးနိုင်သလို width ကို ချုံချုံလိုပါတယ်။

- **<button>**

Pure button ဖြစ်ပါတယ်။ input မှာရှိတဲ့ reset, submit တို့နဲ့ကွဲပြားစေတဲ့
 တချက်ကတော့
 button tag ဟာ open tag close tag [ကြားထဲမှာ image နဲ့ text
 တွေကိုရေးသားနိုင်ပါတယ်။

Example:

```

<button onclick="alert('Learn HTML')>
    
</button>

```

- **<fieldset>**

fieldset ကတော့ form တွေကို box အနေနဲ့ပိုင်းပြပေးတာဖြစ်ပါတယ်။

- **<legend>**

field set ခဲ့ Element box အတွက် Caption ဖြစ်ပါတယ်။

- **<datalist>**

input tag အတွက် [ကြိုတင်သတ်မှတ်ထားတဲ့ list အနေနဲ့သုံးပါတယ်။
 autocomplete လည်း support လုပ်ပေးနိုင်ပါတယ်။

Example:

```
<input list="browsers" name="browser" id="browser">
<datalist id="browsers">
  <option value="Edge">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

- **<option>**

select tag နဲ့ data list တို့အတွက် data တရာ့ခြင်းစီထည့်ဖို့အသုံးပြုပါတယ်။

- **<optgroup>**

Select tag မှာ Category ခွဲချွင်တဲ့ အခါမှာအသုံးပြုပါတယ်။

Categoryအတွက် name ကိုသတ်မှတ်ဖို့ label ကိုအသုံးပြုပါတယ်။

Example:

```
<select name="cars" id="cars">
  <optgroup label="Swedish Cars">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
  </optgroup>
  <optgroup label="German Cars">
    <option value="mercedes">Mercedes</option>
    <option value="audi">Audi</option>
  </optgroup>
</select>
```

14) Special Characters

အသုံးဝင်တဲ့ Special Character ကိုဖော်ပြပေးထားပါတယ်။

အသုံးပြုပို့တွေကတော့

Copyright © Ex;braiN Education

Char	Number	Description
∇	∇	NABLA
∈	∈	ELEMENT OF
∉	∉	NOT AN ELEMENT OF
∋	∋	CONTAINS AS MEMBER
Σ	∑	N-ARY SUMMATION
©	©	COPYRIGHT SIGN
®	®	REGISTERED SIGN
€	€	EURO SIGN
™	™	TRADEMARK
←	←	LEFTWARDS ARROW
↑	↑	UPWARDS ARROW
→	→	RIGHTWARDS ARROW
↓	↓	DOWNWARDS ARROW
♠	♠	BLACK SPADE SUIT
♣	♣	BLACK CLUB SUIT
♥	♥	BLACK HEART SUIT
♦	♦	BLACK DIAMOND SUIT

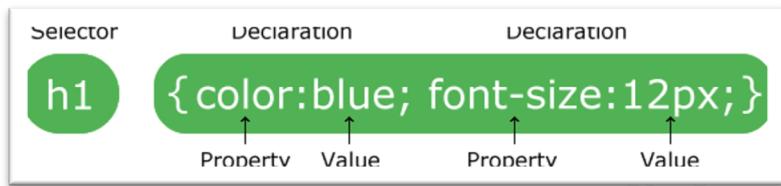
4. Css(Cascading Style Sheets)

- 1) Why Css?
- 2) Css Syntax
- 3) Accessing Css Selector
- 4) 3 Ways of Css inserting
- 5) Color
- 6) Background
- 7) Border
- 8) Margin & Padding
- 9) Text
- 10) Font
- 11) Display & Visibility
- 12) Max-Width
- 13) Position
- 14) Overflow
- 15) Float & Clear
- 16) !important

1) Why Css?

CSSသုံးရခြင်း၏ရည်ရွယ်ချက်ကတော့ webpage မှာ ကိုယ်လိုချင်တဲ့ design တွေ color တွေ animation တွေဟာ HTML tag တွေနဲ့တည်ဆောင်လို့မရနိုင်ပါဘူး။html ဟာအိမ်တလုံးဆောက်ဖို့အတွက် foundation ချလိုက်တာဖြစ်ပြီးတော့ cssကတော့ အနုစိတ် design နဲ့အချေသပ်ပေးတဲ့ပြစ်ပါတယ်။ဒါကြောင့်မို့ CSSကို သုံးပေးရပါတယ်။

2) Css Syntax



Selector : ကိုယ့် customize လုပ်ချင်တဲ့ target

Property: Css Property example:background-color,margin,padding etc...,

Value: Property၏ value

3) Accessing Css Selector

ကိုယ်လိုချင်တဲ့ target တွေကို accessing လုပ်နိုင်ဖို့အတွက် သုံးနိုင်တဲ့

selectorတွေကတော့

■ id

Example: #targetname

■ class

Example: .target name

■ element

Example: h1

■ element.class

Example: p.target name

■ * (universal)

■ element,element

Example: p,div,input

4) 3 Ways of Css inserting

Css ကို ပုံစံ၏ခုနှင့်ရေးနှင့်ပါတယ်။

Internal Css:

internal ကတေသာ html <head> tag ထဲ <style> tag ကိုသုံးပြီး HTML file ထဲမှာပဲ
Cssကို ရေးတာဖြစ်ပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
<style>
  body {
    background-color: linen;
  }

  h1 {
    color: maroon;
    margin-left: 40px;
  }
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

External Css:

external ဆိုတာကတေသာ html <head> tag ရဲ့link tag မှာ Css file ကိုချက်ပြီး
ချက်ထားတဲ့ Css File မှာပဲ ရေးတာဖြစ်ပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css" />
</head>
<body>
<p>Text Something</p>
</body>
</html>
```

Inline Css:

html tag အတွင်းမှာပဲ style attribute ကိုသုံးပြီး Css ကိုရေးတာဖြစ်ပါတယ်။

```
<!DOCTYPE html>
<html>
<body>
    <h1 style="color: blue; text-align: center">This is a heading</h1>
    <p style="color: red">This is a paragraph.</p>
</body>
</html>
```

5) Color

Text Color

စာသားတွေရဲ့အရောင်တွေကို ပြောင်းချင်တဲ့အခါသုံးပါတယ်။

```
<p style="color:Tomato;">So Sweet</h1>
<p style="color:green;">Tree is green</p>
<p style="color:yellow;">Mango /p>
```

Background Color

နောက်ခံအရောင်တွေပြောင်းတဲ့အခါသုံးပါတယ်။

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
<p style="background-color:red;"> Rose is red</p>
```

Border Color

ဘောင်တွေရဲ့အရောင်ကိုသတ်မှတ်ချင်တဲ့အခါသုံးနိုင်ပါတယ်။

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>
```

Color Value

Color value မှာဆိုရင်တော့ red,green,yellow စသည်ဖြင့်ရေးနိုင်သလို

Rgb() color,Hex color တွေနဲ့လည်းသတ်မှတ်ထားနိုင်ပါတယ်။

```
<h1 style="background-color:rgb(255, 99, 71);">...</h1>
<h1 style="background-color:#ff6347;">...</h1>
```

Color Code value တွေရဲ့ Detail တွေကို ဒီlink မှာလေ့လာလို့ရနိုင်ပါတယ်။

<https://imagecolorpicker.com/color-code/>

6) Background

အသုံးများတဲ့ Background ရဲ့ Category မှာတော့

background-color :

အပေါ်မှာလေ့လာခဲ့တဲ့ အတိုင်းနောက်ခံအရောင်တွေကိုပြောင်းတဲ့ အခါမှာသုံးပါတယ်။

သူနဲ့တွဲသုံးလို့ရတဲ့ Attribute တွေကတော့ opacity ဖြစ်ပါတယ်။

Opacity မှာတော့ 1 ကနေ 0.0 ထိ ရှိပါတယ်။ 1 ကတော့ original ဖြစ်ပြီးတော့ သူအောက်ရောက်သွားရင်တော့ အကြည်ဘက်ရောက်သွားမှာ ဖြစ်ပါတယ်။

ဥပမာ

opacity:0.6,
opacity:0.4

```
p {  
    background-color: "#2596be";  
}
```



background-image : Background ကို အရောင်တင်မဟုတ်ပဲ imageပါထည့်နိုင်ပါတယ်။

သူရဲ့အသုံးပြုပုံတော့

```
p {  
    background-image: url("paper.gif");  
}
```

background-repeat : repeat ကတေသူပုံတွေကိုပါးတဲ့သဘောပါပဲ။
repeat မှာတေသူ repeat-x , repeat-y ဆိုပြီး ရှိပါတယ်။
repeat-x ကတေသူ ပုံတွေကို horizontally ပါးတာဖြစ်ပြီးတေသူ repeat-y ကတေသူ
vertically ပါးတာဖြစ်ပါတယ်။

```
body {  
    background-image: url("gradient_bg.png");  
    background-repeat: repeat-x;  
}
```

နောက်တခုကတေသူ no-repeat ပါ။
သူကတေသူ ပုံကိုတပ်ပုံတည်းထားစေချင်တဲ့အခါသုံးနိုင်ပါတယ်။

background-position : သူကတေသူပုံကို နေရာချုတဲ့အခါသုံးပါတယ်။

```
body {  
    background-image: url("img_tree.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
}
```

အပေါ်ကလ္လာခဲ့တဲ့ Background ကို တောက်ဌာန်းတည်းနဲ့
ရေးချင်ရင်လည်းရပါတယ်။

ဥပမာ

```
body {  
    background: #ffffff url("img_tree.png") no-repeat right top;  
}
```

7) Border

ဘောင်ခတ်ချင်တဲ့အခါ border ကိုသုံးနိုင်ပါတယ်။

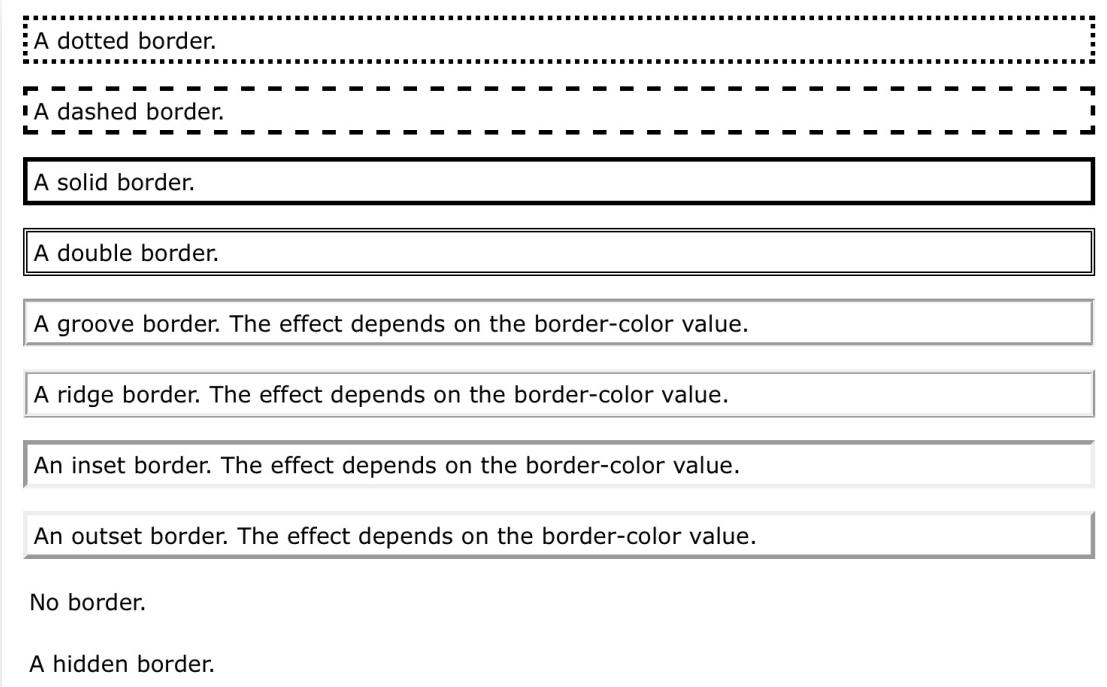
Border မှာဆိုရင်တော့

border-style:

ဘောင်တွေရဲ design တွေကိုပုံစံအမျိုးမျိုးနဲ့ပေးနိုင်ပါတယ်။

solid,dotted,dashed,double,groove,ridge,inset,outset,none,hidden
မျိုးစုအောင်ပေးနိုင်ပါတယ်။

```
{border-style: dotted;}  
{border-style: dashed;}  
{border-style: solid;}  
{border-style: double;}  
{border-style: groove;}  
{border-style: ridge;}  
{border-style: inset;}  
{border-style: outset;}  
{border-style: none;}  
{border-style: hidden;}
```



လေးဖက်လေးလုပ်စံကို တမျိုးစီရေးချင်တဲ့အခါရေးနိုင်တဲ့နည်းမျိုးရှိပါတယ်။

တခုကတော့

{border-style: dotted dashed solid double;}

အပေါကစပြီး နာရီလက်တံ့လည်သလိုသွားပါတယ်။

up -> right -> bottom -> left

နောက်တခုကတော့ Side တခုချင်းစီအလိုက်သတ်မှတ်ပေးတာဖြစ်ပါတယ်။

```
{  
border-top-style: dotted;  
border-right-style: solid;  
border-bottom-style: dotted;  
border-left-style: solid;  
}
```

border-width:

ဘောင်ရဲအထူကိုသတ်မှတ်ပေးတာပဲဖြစ်ပါတယ်။px,in အစရှိတဲ့calculation

တွေနဲ့သတ်မှတ်နိုင်ပါတယ်။

{border-width: 20px;}

border-style လိုမျိုး ဂျို့မျိုးလည်းရေးနိုင်ပါတယ်။

```
{  
border-width: 2px 5px 10px 1px;  
}  
  
{  
border-top-width: 2px;  
border-right-width: 5px;  
border-bottom-width: 10px;  
border-left-width: 1px;  
}
```

border-color : color မှာလေ့လာခဲ့ပြီးပါ၍။ border-colorကို border-style လို့မျိုးတကြောင်းတည်းရေးနိုင်သလို side တခုချင်းစီလည်းရေးနိုင်ပါတယ်။

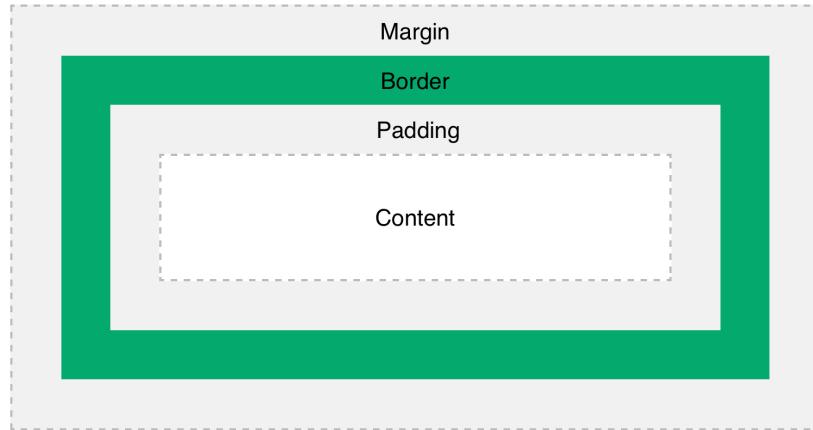
```
{  
    border-color: red green blue yellow;  
}  
  
{  
    border-top-color: red;  
    border-right-color: green;  
    border-bottom-color: blue;  
    border-left-color: yellow;  
}
```

border-radius : ဘောင်ရဲထောင့်ချိုးတွေကို လေးထောင့်ပုံစံမဟုတ်ပဲအပိုင်းပုံစံအနေနဲ့ရေးချင်တဲ့အခါသုံးပါတယ်။
အပေါ်ကလေ့လာခဲာတွေကို shorten နည်းနဲ့ရေးချင်တဲ့အခါ

```
{  
    border: 2px solid red;  
    border-radius: 5px;  
}
```

သူရဲ့order အစဉ်လိုက်ကတော့ width,style,colorပဲဖြစ်ပါတယ်။

8) Margin & Padding



Margin vs Padding

အပေါက်ပုံကိုလေ့လာကြည့်မယ်ဆိုရင် Margin ဆိုတာ element
တွေကြားအကွာအဝေးဖြစ်ပြီးတော့ padding ဆိုတာကတော့ element
အတွင်းမှာရှိတဲ့ဟာတွေခဲ့အကွာအဝေးဖြစ်ပါတယ်။
Margin နဲ့ Padding ဟာ တူညီတဲ့ရေးသားပုံတွေဖြစ်ပါတယ်။

```
margin-top : 20px;  
margin-right : 10px;  
margin-bottom : 15px;  
margin-left : 30px;
```

Or

```
margin : 20px 10px 15px 30px;
```

```
padding-top : 20px;  
padding-right : 10px;  
padding-bottom : 15px;  
padding-left : 30px;
```

Or

```
padding : 20px 10px 15px 30px;
```

9) Text

text-align : text-alignကတ္တု စာသားတွေရဲ့ horizontal alignment

ကိုချိန်တဲ့အခါသုံးပါတယ်။သူမှာကတ္တု center, left, right နဲ့ justify ဆိုပြီးရှိပါတယ်။

```
h1 {  
    text-align: center;  
}
```

```
h2 {  
    text-align: left;  
}
```

```
h3 {  
    text-align: right;  
}
```

direction : ပုံနှိပ်စာသားတွေကို ပြောင်းပြန် directionနဲ့ရေးချင်တဲ့အခါမျိုးမှာ သုံးနိုင်ပါတယ်။ Irthelessဆိုတာကတ္တု right to left ပါ။

```
p {  
    direction: rtl;  
}
```

text-decoration : စာသားတွေကို over line တားတာ । underline တားတာ၊ strike ဖျက်တာတွေအတွက် အသုံးပြနိုင်ပါတယ်။ Default ကတ္တု noneပါ။

```
h1 {  
    text-decoration: overline;  
}
```

```
h2 {  
    text-decoration: line-through;  
}
```

```
h3 {  
    text-decoration: underline;  
}
```

text-indent : ပထမဆုံးစာကြောင်းကို ညိုချင်တဲ့အခါမှာသုံးပါတယ်။

```
p {  
    text-indent: 50px;  
}
```

letter-spacing : စာသားတရုချင်းစီရဲအကွာအဝေးကို ချိန်ချင်တဲ့အခါသုံးပါတယ်။

```
h1 {  
    letter-spacing: 3px;  
}
```

line-height : စာကြောင်းတွေရဲ အကွာအဝေးကို ချိန်ချင်တဲ့အခါသုံးပါတယ်။

```
p {  
    line-height: 0.8;  
}
```

```
p {  
    line-height: 1.8;  
}
```

word-spacing : စာစုတစုချင်းစီရဲအကွာအဝေးဖြစ်ပါတယ်။

```
h1 {  
    word-spacing: 10px;  
}
```

```
h2 {  
    word-spacing: -5px;  
}
```

white-space : စာသားတွေဟာ webpageရဲ့ size ကို ကျော်လွန်သွားတဲ့အခါ နောက်တကြောင်းဆင်းပါတယ်။ အဲလိုမဖြစ်ချင်ပဲ တဆက်တည်း ဖြစ်စေချင်တဲ့အခါသုံးပါတယ်။

```
p {  
    white-space: nowrap;  
}
```

10) Font

font-family: သူကတော့ ရေးလိုက်တဲ့ Text တွေရဲ့ font
တွေကိုသတ်မှတ်တဲ့အခါမျိုးမှာသုံးပါတယ်။ Family ဖြစ်တဲ့အတွက် တခုထက်
ပိုရေးလို့ရပါတယ်။ commaနဲ့ ရေးရတာဖြစ်ပြီးတော့ ပထမဆုံး font ကတော့
ကိုယ်အလိုချင်ဆုံးfont, ဒုတိယကစပြီး general fontကိုရေးပေးရပါတယ်။
Browserက ကိုက်ညီတဲ့ font တွေကို အစဉ်လိုက်ရှာပြီး
တူညီရင်ပြပေးမှာဖြစ်ပါတယ်။

```
.fonttarget {  
    font-family: "Times New Roman", Times, serif;  
}
```

font-style : style မှာဆိုရင်တော့ ၃မျိုးရှိပါတယ်။normal,italic,oblique
ဖြစ်ပါတယ်။

```
p.normal {  
    font-style: normal;  
}
```

```
p.italic {  
    font-style: italic;  
}
```

```
p.oblique {  
    font-style: oblique;  
}
```

font-weight : font weight ကတော့ font ရဲ့ အထူကိုဆိုလိုတာပါ။
font-weightကိုတော့ normal,lighter,boldဆိုပြီးရေးနိုင်သလို 100 ကနေ 900ထိ
customize ရေးသားနိုင်ပါတယ်။ 100 ကတော့ အထူနည်းပြီးတော့ bold or bolder
ကတော့ အထူဆုံးဖြစ်ပါတယ်။

font-size : စာလုံးရွှေ့အရွယ်အစားကို ချိန်ညီတာဖြစ်ပါတယ်။ px, em,% အစရိတ္ထဲ calculation တွေနဲ့ချိန်ညီပါတယ်။ Responsive အတွက်ဆိုရင်တော့ vw ကိုသုံးပါတယ်။

```
body {  
    font-size: 100%;  
}  
  
h1 {  
    font-size: 2.5em;  
}  
  
h2 {  
    font-size: 1.875em;  
}  
  
p {  
    font-size: 0.875em;  
}
```

11) Display & Visibility

display: ပထမဦးဆုံး Display ကတေသ့ element တွေကို
ဘယ်လိုပုံစံနဲ့ပြုမယ်ဆိုတာ သတ်မှတ်ပေးပါတယ်။

Html မှာလေ့လာခဲ့တဲ့ inline elements နဲ့ block elements တွေကို
ပြောင်းလဲသုံးချင်တဲ့အခါ အသုံးပြုပါတယ်။ ဥပမာ ဟာ block element
ပါ။ အဲဒါကို inline အဖြစ်နဲ့သုံးချင်တဲ့အခါ

```
li{  
    display: inline;  
}
```

 tag ဟာ inline ဖြစ်ပြီးသူကို block အနေနဲ့သုံးချင်တဲ့အခါ

```
span {  
    display: block;  
}
```

နောက်တခုကတေသ့အသုံးများတဲ့ display: none ပါ။ သူက html မှာရေးထားပေမယ့် run
လိုက်တဲ့အခါ မပေါ်စေချင်တဲ့အခါသုံးပါတယ်။ JavaScript နဲ့တွဲသုံးပြီး display တွေ
show/hide လုပ်ချင်တဲ့အခါ သုံးပါတယ်။

```
h1.hidden {  
    display: none;  
}
```

visibility:

Visibility ကတေသ့ element တွေကို မြင်စေချင်/မမြင်စေချင်တဲ့အခါသုံးပါတယ်။

Visible ကတေသ့ မြင်စေချင်တဲ့အခါ original default ပါ။

hidden ကတေသ့ မမြင်စေချင်တဲ့အခါသုံးပါတယ်။

```
h2.a {  
    visibility: visible;  
}
```

```
h2.b {  
    visibility: hidden;  
}
```

display:none Vs visibility : hidden

Display:none ကတေသ့ element ရှိနေတဲ့ နေရာပါ ပျောက်သွားတာဖြစ်ပြီးတေ့ Visibility: hidden ကတေသ့ မြင်ပဲမြင်ရတာပါ နေရာတေ့ ယူထားခဲ့ပါ။

12) Max-width

Responsive အတွက်သုံးပါတယ်။ width ကိုသုံးမယ်ရင် webpage ခဲ့size ထွေပြောင်းသွားတဲ့အခါ သို့မဟုတ် smartphone တွေကနေကြည့်တဲ့အခါ သတ်မှတ်ထားတဲ့ size အောက်ရောက်သွားရင် webpage မှာ scroll bar တွေက်ပြီးကြည့်ရတာ အမြင်မလှတေ့ပါဘူး။အဲလိုမဖြစ်ရအောင် max-width ကိုသုံးပါတယ်။

အောက်ပါ code ကိုrun ကြည့်ရင်သိနိုင်ပါတယ်။

```
<!DOCTYPE html>
<html>
<head>
<style>
div.ex1 {
    width: 500px;
    margin: auto;
    border: 3px solid #73AD21;
}
div.ex2 {
    max-width: 500px;
    margin: auto;
    border: 3px solid #73AD21;
}
</style>
</head>
<body>
<h2>CSS Max-width</h2>
<div class="ex1">This div element has width: 500px;</div>
<br>
<div class="ex2">This div element has max-width: 500px;</div>
</body>
</html>
```

13) Position

Element တွေအတွက် position ချတဲ့အခါသုံးပါတယ်။ element တွေကို fixed ဖြစ်ချင်တဲ့ အခါ သို့မဟုတ် တွေးelement တွေနဲ့ ဆက်ဆက်ပြီး position ချချင်တဲ့အခါ သုံးနှင့်ပါတယ်။

※ position static ကလွှဲလို့ ကျန်တဲ့ position တွေမှာ top , right ,left ,bottom ဆိုတဲ့ align ချိန်တဲ့ property တွေရှိပါတယ်။

position: static

Default value ဖြစ်ပါတယ်။ top ,right, left ,bottom စတဲ့ alignment တွေသုံးလို့မရှိနိုင်ပါဘူး။

position: relative

Relative ကတော့ top,right,left, bottom တွေသုံးနှင့်ပါတယ်။
ဥပမာ relative လုပ်မယ် element မှာ parent element ရှိမယ်၊ relative လုပ်မယ့် element ကို top: 5 px ပေးထားလိုက်မယ်ဆိုရင် parent element နဲ့ top 5px အကွာအဝေးထိ ဆင်းသွားမှာဖြစ်ပါတယ်။ relative ဆိုတဲ့ အတိုင်း တခုနဲ့တခု ဆက်ဆက်နေပါတယ်။

position: absolute

Relative ဆင်တူပါတယ်။သို့ပေမယ့် သူရဲ့ parent ဟာ static ဖြစ်တဲ့ အခါ Top,right,left,bottom တန်ဖိုးတွေပေးတဲ့အခါ webpage ရဲ့စမှတ်ကနေပဲ သတ်မှတ်ပါတယ်။အကယ်၍ သူရဲ့ parent ဟာ relative တန်ဖိုးပေးထားခဲ့ရင်တော့ parent ရဲ့စမှတ်ကနေ အကွာအဝေးစမှာဖြစ်ပါတယ်။

position: fixed

Absolute နဲ့တူပါတယ်လို့ပြောနိုင်ပါတယ်။ ကဲပြားစေတဲ့ အချက်ကတော့ fixed ဟာ Webpage မှာ scrolling လုပ်တဲ့ အခါ နေရာ မရွှေ့ပဲ သတ်မှတ်ထားတဲ့နေရာမှာပဲ ပေါ်နေမှာဖြစ်ပါတယ်။absolute ကတော့ scrolling လုပ်တဲ့အခါ ကွယ်သွားမှာပဲဖြစ်ပါတယ်။

Position တွေအတွက် ဒီvideo ကို reference ကြည့်နိုင်ပါတယ်။

<https://youtu.be/jx5jmI0UIXU>

14) Overflow

Border တွေ သို့မဟုတ် frame တွေထဲ text တွေထည့်ရေးတဲ့အခါ
မဆန့်တွေဖြစ်တဲ့အခါ ဖြေရှင်းတဲ့နည်းလမ်းတွေထဲက တခုဖြစ်ပါတယ်။
Overflowမှာတော့ ငါရှိပါတယ်။

overflow: visible

Default ဖြစ်ပါတယ်။စာသားတွေဟာ ဘောင်တွေကျော်ထွက်နေတဲ့အခါ
ကျော်ထွက်တဲ့အတိုင်းထားစေချင်တဲ့အခါမျိုးမှာသုံးပါတယ်။

overflow: hidden

စာသားတွေကျော်ထွက်တဲ့အခါ ကျော်တဲ့စာသားတွေကို မပြုစေချင်တဲ့အခါ
သုံးပါတယ်။

overflow: scroll

စာသားတွေကျော်သွားရင် scroll ဖြစ်ပြီးပြချင်ရင် သုံးပါတယ်။

overflow: auto

Scroll နဲ့တူတူပါပဲ။ကွဲပြားစေတဲ့ အချက်ကတော့ auto က စာသားတွေက fit
ဖြစ်နေရင် scroll bar မထည့်ပါဘူး။

15) Float & Clear

Float ကတေသ့ အသုံးများတဲ့ အထပါပါတယ်။ elements တွေကို left right ခိုင်ညိုတဲ့ အခါမှာသုံးပါတယ်။

float:right

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...



float:left



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...

float: none



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...

16) !important

!important ကတော့ တူညီတဲ့ အရေးအသာ:တွေရှိနေတဲ့အခါ !important ပါတဲ့
ဘာက priority ပိုမြင့်ပါတယ်။ ဥပမာ background: red လိုပေးထားတာ တခုရယ်၊
background:green ; !important ရယ်ရှိတဲ့အခါ green ပဲပေါ်ပါတယ်။

```
.myclass {  
    background-color: red;  
}  
  
p {  
    background-color: green !important;
```

1. Learning to Code with Javascript

- 1) Why JS?
- 2) How to write JS?
- 3) The ways of writing
- 4) Comments
- 5) Syntax & Data Types
- 6) Variables Rules
- 7) Operators, Assignment & Comparisons
- 8) If, Switch, Loops Condition
- 9) Methods
- 10) Function
- 11) DOM Events

1) Why JS?

Html နှင့် Css ကိုလွှလာပြီးသွားတဲ့အခါ web page design
တွေကိုတော်တော်ရေးနိုင်နေပါပြီ။သို့ပေမယ့် Animation တွေ(Mouse Events ...etc) စတာတွေရေးနိုင်ပြီ။ Html Css နဲ့တင်မလုံလောက်တွေ့ပါဘူး။
JavaScript ဟာ Dom elements တွေကို ပြောင်းတာရွှေ့တာပေါ်တာ ဖျောက်တာ၊
design တွေ color တွေကို dynamically ပြောင်းချင်တဲ့အခါ စတဲ့ အချက်တွေကို
လုပ်နိုင်ပါတယ်။အဲဒါတွေအပြင် client side ရဲcoding ကို JavaScript နဲ့ရေးနိုင်ပါတယ်။

2) How to write JS? (Accessing to JavaScript)

Jsကို <script> tag နဲ့ရေးပါတယ်။<body> tag ရဲ့အောက်ဆုံးနားမှာရေးခြင်းက display speedကိုမြန်စေပါတယ်။JavaScript ကို Internal(html file နှင့်တူတူရေးသားခြင်း) နှင့် External (standalone file ခဲ့ရေးသားခြင်း) တို့ဖြစ်ပါတယ်။

■ Internal

```
<!DOCTYPE html>
<html>
  <body>
    <h1>How to Write Javascript</h1>
    <p id="demo">Apple</p>
    <button type="button" onclick="myFunction()">Change</button>

    <script>
      function myFunction() {
        document.getElementById("demo").innerHTML = "Orange";
      }
    </script>
  </body>
</html>
```

■ External

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

3) The ways of writing

- DOM

```
<!DOCTYPE html>
<html>
  <body>
    <p id="note"></p>
    <script>
      document.getElementById("note").innerHTML = "Hello World";
    </script>
  </body>
</html>
```

Finding HTML Elements

1) `document.getElementById()`

Html မှာ id ပေးထားတယ်ဆိုရင်တော့ `document.getElementById()`
နဲ့ target ကို select လုပ်တာပါ။

2) `document.getElementsByTagName()`

Html tag name ကို target ထားပြီး `select` လုပ်တာဖြစ်ပါတယ်။
အဲလို `select` လုပ်ခြင်းအားဖြင့် target tag name
တွေအားလုံးပေါ်မှာသက်ရောက်ပါတယ်။

3) `document.getElementsByClassName()`

Html မှာ class နဲ့ ကြညာပြီး အဲဒီ target ကို `select`
လုပ်ချင်တဲ့ အခါမျိုးမှာသုံးပါတယ်။ တူညီတဲ့ class name တဲ့ ထက်မက html
မှာခေါ်သုံးထားတယ်ဆိုရင်တော့ အားလုံးပေါ်မှာသက်ရောက်ပါတယ်။

Changing HTML Elements

1) `innerHTML`

`innerHTML` ဆိုတာကတော့ html tag ထဲကစာတွေ သို့မဟုတ် ထပ်ဖြည့်ချင်တဲ့
elements တွေကို text အနေနဲ့ ရေးနိုင်ပါတယ်။

2) **element.attribute = new value**

HTML element ရဲ့ attribute မှာရေးထားတဲ့ value ကိုပြောင်းချင်တဲ့အခါသုံးပါတယ်။

Example:

Html

```
<img id =“image” src=“a.png”/>
```

JavaScript

```
<script>
document.getElementById(“image”).src = “b.png”
</script>
```

3) **element.style.property = new style**

HTML element မှာ သုံးထားတဲ့ Css style တွေကိုပြောင်းချင်တဲ့အခါသုံးပါတယ်။

Example:

Html

```
<p id =“p1”> Something </p>
```

JavaScript

```
<script>
    document.getElementById(“p1”).style.color = “red”;
    document.getElementById(“p1”).style.fontSize = “20px”;
</script>
```

4) **element.setAttribute(attribute,value)**

Target HTML element ကို attribute အသစ်ထပ်ထည့်ချင်တဲ့အခါသုံးပါတယ်။

Example:

Html

```
<p id="p1" class="p1">Happy Day</p>
```

JavaScript

```
<script>
document.getElementById("p1").setAttribute("class", "p1change")
</script>
```

- **document.write()**

```
<!DOCTYPE html>
<html>
<body>
<script>
  document.write("The Sum of 10+2 is = ");
  document.write(10 + 2);
</script>
</body>
</html>
```

Document.write ကတေသာ အခါနမှုပြုပေးချင်တဲ့အခါ
သုံးပါတယ်။

- **window.alert()**

```
<!DOCTYPE html>
<html>
<body>
<script>
  window.alert("This is alert!");
</script>
</body>
</html>
```

Webpage မှာ alert box အနေဖွဲ့ဖြတ်ပါတယ်။ User ကို warning သိမဟုတ်
information တွေပေးချင်တဲ့အခါ သုံးနိုင်ပါတယ်။ Developer တွေလည်း
Debugging လိုက်တဲ့ အခါလည်း output ထုတ်တဲ့အခါလည်းသုံးကြပါတယ်။

- **console.log()**

```
<!DOCTYPE html>
<html>
<body>
<script>
  console.log("This is console!");
</script>
</body>
</html>
```

Developer တွေ debugging လိုက်တဲ့အခါ သုံးကြပါတယ်။ result ကို webpage
မှာမပြုပဲ browser ရဲ့ console tag မှာကြည့်နိုင်ပါတယ်။

4) Comments

Developer တွေဟာ သူရေးထားတဲ့ code ကို အများသူနားလည်အောင်
လက်ဆင့်ကမ်းဖို့နဲ့ပြန်ကြည့်တဲ့အခါ ကိုယ်တိုင်နားလည်နိုင်အောင် သို့မဟုတ်
code တွေရဲ့ step by step ရှင်းပြတဲ့အခါ မဖြစ်မနေ comment ရေးရပါတယ်။
နောက်ပြီး code တွေကို အလုပ်မလုပ်အောင် comment ပိတ်ပြီး testing
လုပ်တဲ့အခါလည်းသုံးပါတယ်။

- Single Line

```
// This is comment
```

```
တောက်တည်းရေးချင်တဲ့အခါ // ကိုသုံးပါတယ်။
```

- Multi-line

```
Block လိုက် comment ရေးတာ သို့မဟုတ် coding
```

```
တွေပိတ်တဲ့အခါသုံးပါတယ်။
```

```
/*
```

```
This is block comment
```

```
document.getElementById("myH").innerHTML = "My First Page";
```

```
*/
```

```
document.getElementById("myP").innerHTML = "My first paragraph.;"
```

5) Syntax & Data Types

▪ Syntax

```
var x; // Declare Variables  
x = 5; // Assign Values  
z = x + y; // Compute Values
```

`var x;`

သူကတော့ computerရဲ့ memory မှာနေရာတခုယူဖို့အတွက် computerကို
ကြညာပေးလိုက်တာပါ။သူထဲကိုကိန်းဂဏန်းတွေအပြင်စာသားတွေထည့်လိုပါ
တယ်။

`x = 5;`

အရင်ဆုံး x ကို computerမှာနေရာယူမယ်လို့ကြညာလိုက်ပါတယ်။
ပြီးတဲ့နောက် xရဲ့တန်ဖိုးကို 5 ထည့်လိုက်တာပါ။

`z = x + y;`

Calculation ဖြစ်ပါတယ်။ zထဲကို xနှင့် yပေါင်းခြင်းကို
ထည့်လိုက်တာပါ။အပေါင်း (+) နေရာမှာ၊ အနှစ် (-)၊ အမြောက် (*)၊
အစား (/) စတာတွေလည်း သုံးနိုင်ပါတယ်။

▪ Datatypes

- **string**

စာသားတွေအတွက်သုံးပါတယ်။ သူကို double quote အစနဲ့ double
qoute အဆုံးရေးရပါတယ်။

`var x = "test for string";`

ဆိုရင် x ရဲ့တန်ဖိုးမှာ test for string ဆိုတဲ့ စာသားကိုထည့်လိုက်တာ
ဖြစ်ပါတယ်။

- **number**

ကိန်းဂဏန်းတွေအသာမကိန်းတွေထည့်ချင်တဲ့အခါသုံးပါတယ်။

`var x = 100;`

`var y = 12.5;`

- **Boolean**

စာသားတွေကိန်းဂဏန်းတွေ မဟုတ်ပဲ Boolean

တန်ဖိုးနဲ့လည်းပေးနိုင်ပါတယ်။ Boolean မှာ value နှစ်ခုရှိပါတယ်။ True , False ။

- **undefined**

အပေါ်က data type တွေသတ်မှတ်ထားခြင်းမရှိတာကို undefined
လိုအပ်ပါတယ်။

- **null**

Null ကတော့ တန်ဖိုးမရှိတဲ့ type တခုဖြစ်ပါတယ်။ variable တခုကို Null ပေးလိုက်တဲ့အခါ သူရဲ့တန်ဖိုးဘာမှုမရှိတော့ပါဘူး

6) Variables Rules

Syntax:

```
var <variable-name>;  
var <variable-name> = <value>;
```

Var ကိုကြညာတဲ့အခါသတ်မှတ်ထားတဲ့rule တွေရှိပါတယ်။

1. Space မပါရပါဘူး။
2. အစစာလုံးကို letter,underscore(_),dolla sign(\$) နှဲပဲရေးပါတယ်။
3. Variable name ကို letter,number,underscore,dolla sign တွေနဲ့ရေးနိုင်ပါတယ်။
4. JavaScript မှာတခြားအဓိပ္ပာယ်ရှိပြီးသားnameတွေမပါရပါဘူး။

Example: if,for,while,etc...,

တို့ဖြစ်ပါတယ်။Syntax error တွေမဖြစ်အောင်လိုက်နာရမယ်rule တွေဖြစ်ပါတယ်။

International standard တွေမှာတော့

1. Variable name ကို အရမ်းတို့တဲ့short name တွေမပေးရပါ။
Example: var a, var b.
2. စာစုတစုထက်ပိုရေးရပါတယ်။ကြားထဲမှာ underscore ခံရေးရပါတယ်။
3. အဓိပ္ပာယ်ရှိတဲ့nameကိုပဲပေးရပါတယ်။

7) Operators, Assignment & Comparisons

▪ Operator

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
**	Exponentiation (ES2016)
/	Division
%	Modulus (Division Remainder)
++	Increment
--	Decrement

Operator တော့ကိန်းဂဏ်နံတန်ဖိုးတွေကို calculate လုပ်ပေးတာဖြစ်ပါတယ်။ + - * / တွေအပြင်

1. Modulus(အကွင်းကိုယူ)

2. ++ အဆတိုး

Example:

```
var operator_test = 1;  
operator_test ++;  
ဆိုရင် operator_test ဟာ 2 ဖြစ်လာပါမယ်။
```

3. -- အဆလျှော့

Example:

```
var operator_test = 1;  
operator_test --;  
ဆိုရင် operator_test ဟာ 0 ဖြစ်လာပါမယ်။
```

```

var x = 5, y = 10, z = 15;

x + y; //returns 15
y - x; //returns 5
x * y; //returns 50
y / x; //returns 2
x % 2; //returns 1
x++; //returns 6
x--; //returns 4

```

※ Plus(+) :

+ plus sign ကို ကိန်းကဏ္ဍားတွေပေါင်းတာအပြင် စာသား(string)တွေဆက်တဲ့ အခါမှုလည်းသုံးပါတယ်။

Example :

```

var a = 5, b = "Hello ", c = "World!", d = 10;
a + b; // "5Hello "
b + c; // "Hello World!"
a + d; // 15

```

■ Assignment

Operator	Example	Same As
=	x = y	
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y

Assignment (တန်ဖိုးထည့်ခြင်း)မှာ သချုပ်သီအိုရီအရ ညီမှုခြင်းရဲ့ဘယ်ဘက် မှာဆိုရင်တော့ လက်ခံမယ့်variable ရယ်၊ ညာဘက်မှာဆိုရင်တော့ ထည့်လိုက်မယ် တန်ဖိုးရယ်ဖြစ်ပါတယ်။

$x = x + y$
 ဆိုရင် x ကဲတန်ဖိုးနဲ့ y တန်ဖိုးနဲ့ပေါင်းပြီး x ထဲပြန်ထည့်တာဖြစ်ပါတယ်။
 အဲလိုမျိုးကို နောက်တနည်းနဲ့ရေးချင်ရင်တော့
 $x+=y$ ဆိုပြီးရေးနိုင်ပါတယ်။ + - * / % တွေနဲ့လည်းရေးနိုင်ပါတယ်။

▪ Comparisons

Operator	Description
<code>==</code>	equal to
<code>====</code>	equal value and equal type
<code>!=</code>	not equal
<code>!==</code>	not equal value or not equal type
<code>></code>	greater than
<code><</code>	less than
<code>>=</code>	greater than or equal to
<code><=</code>	less than or equal to
<code>?</code>	ternary operator

Comparison တွေကို conditions တွေစစ်တဲ့ အခါမှာ အသုံးပြုပါတယ်။

If တို့ while loop တို့ for loop တို့မှာ အသုံးပြုပါတယ်။

If, for loop, while loop တို့ကို နောက်သင်ခန်းစာမှာ လေ့လာနိုင်ပါတယ်။

8) if,switch,for,while,do while Condition

- **if**

Syntax:

```
if (condition) {  
    // block of code to be executed if the condition is true  
}
```

If የ condition

ຕွေမှန်မမှန်စစ်တဲ့အခါဖြစ်လာနိုင်တဲ့အမျိုးအစားကသိပ်မများတဲ့အခါifကိုအသုံးများပါတယ်။

```
var a = 10;  
  
if (a > 5) {  
    window.alert("Greater than 5");  
}
```

if else

```
var a = 10;  
  
if (a > 5) {  
    window.alert("Greater than 5");  
} else {  
    window.alert("Less than 5");  
}
```

if elseif ... else

```
var a = 10;  
  
if (a > 5) {  
    window.alert("Greater than 5");  
} else if(a == 5) {  
    window.alert("This is 5");  
} else {  
    window.alert("Less than 5");  
}
```

■ switch

Syntax:

```
switch(expression) {  
    case x:  
        // code block  
        break;  
    case y:  
        // code block  
        break;  
    default:  
        // code block  
}
```

Switch ကတေသာ multiple case တွေစစ်ချင်တဲ့အခါသုံးနိုင်ပြီးတော့ ဖြစ်လာနိုင်တဲ့ case တွေဟာ အများကြီးရှိနိုင်တယ်ဆိုရင်တော့ switchကိုသုံးပါတယ်။ Break; ကတေသာရှုဆက်မသွားတော့ပဲ switch ထဲထွက်မယ်ဆိုတဲ့ keyword ဖြစ်ပါတယ်။

```
var today = new Date().getDay();  
  
switch (today) {  
    case 0:  
        day = "Sunday";  
        break;  
    case 1:  
        day = "Monday";  
        break;  
    case 2:  
        day = "Tuesday";  
        break;  
    case 3:  
        day = "Wednesday";  
        break;  
    case 4:  
        day = "Thursday";  
        break;  
    case 5:  
        day = "Friday";  
        break;  
    case 6:  
        day = "Saturday";  
}  
window.alert(day);
```

- **for**

```
for (statement 1; statement 2; statement 3) {  
    // code block to be executed  
}
```

အကြမ်ရေတာကြမ်ထက်မကရှိနေတဲ့ process တွေကိုရေးချင်တဲ့ အခါ for loop ကိုသုံးနိုင်ပါတယ်။

Statement 1မှာတော့ variable ကိုကြညာပေးရပါတယ်။

Statement 2မှာတော့ for loop ကိုဘယ်conditionထိ looping လုပ်မှာဆိုတာရေးရပါတယ်။

Statement 3မှာတော့ Statement 1ရဲတန်ဖိုးကို တိုးပေးရပါတယ်။

```
<!DOCTYPE html>  
<html>  
    <body onload="myFunction()">  
        <p id="demo"></p>  
  
        <script>  
            function myFunction() {  
                var result = 0;  
                for (var i = 0; i < 3; i++) {  
                    result += i;  
                    console.log(i);  
                }  
                document.getElementById("demo").innerHTML = result;  
            }  
        </script>  
    </body>  
</html>
```

- **while**

```
while (condition) {  
    // code block to be executed  
}
```

While ကတေသ့ပေးထားတဲ့ condition မှန်နေသလို |looping

ဖြစ်နေမှာပဲဖြစ်ပါတယ်။အရေးကြီးတာကတေသ့ while loop ကိုရပ်တန်ဖို့ Condition ပေးရပါတယ်။

```
<!DOCTYPE html>  
<html>  
  <body>  
    <h2>Odd Number</h2>  
    <p id="demo"></p>  
    <script>  
      var text = "";  
      var i = 0;  
  
      while (i < 11) {  
        text += "<br>The number is " + i;  
        i = i + 2;  
      }  
  
      document.getElementById("demo").innerHTML = text;  
    </script>  
  </body>  
</html>
```

EX

- **do while**

```
do {  
    // code block to be executed  
}  
while (condition);
```

```
<!DOCTYPE html>  
<html>  
  <body>  
    <h2>Odd Number</h2>  
    <p id="demo"></p>  
    <script>  
      var text = "";  
      var i = 0;  
  
      do {  
        text += "<br>The number is " + i;  
        i = i + 2;  
      } while (i < 11);  
  
      document.getElementById("demo").innerHTML = text;  
    </script>  
  </body>  
</html>
```

While နဲ့ဆင်တူပါတယ်။အမိကကွာခြားချက်ကတော့ do whileဟာ
ဘာပဲဖြစ်ဖြစ်တို့မှတ်လုပ်ပါတယ်။ပြီးမှ condition ကိုစစ်ပြီးဆက်လုပ်မလုပ်
ဆုံးဖြတ်ပါတယ်။

9) Methods

- String Methods

- 1) charAt()
- 2) charCodeAt()
- 3) concat()
- 4) endsWith()
- 5) fromCharCode()
- 6) includes()
- 7) indexOf()
- 8) lastIndexOf()
- 9) localeCompare()
- 10) match()
- 11) repeat()
- 12) replace()
- 13) search()
- 14) slice()
- 15) split()

- 16) startsWith()
- 17) substr()
- 18) substring()
- 19) toLocaleLowerCase()
- 20) toLocaleUpperCase()
- 21) toLowerCase()
- 22) toString()
- 23) toUpperCase()
- 24) trim()
- 25) valueOf()

- Number Methods

- 1) toExponential()
- 2) toFixed()
- 3) toPrecision()
- 4) toString()
- 5) valueOf()

- Math(Mathematical) Methods

- 1) Math.PI
- 2) Math.round()
- 3) Math.cell()
- 4) Math.floor()
- 5) Math.trunc()
- 6) Math.sign()
- 7) Math.pow()
- 8) Math.sqrt()
- 9) Math.abs()
- 10) Math.sin()
- 11) Math.cos()
- 12) Math.min()
- 13) Math.random()

10) Function

Syntax:

```
function name(parameter1, parameter2, parameter3) {  
    // code to be executed  
    return  
}
```

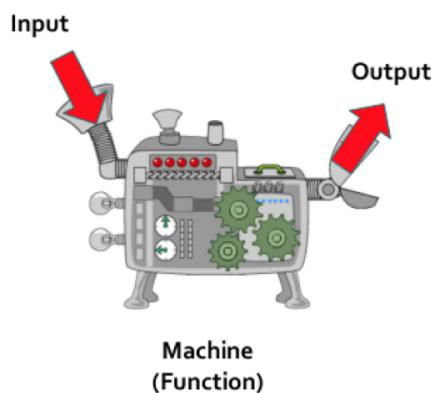
Function ဆိတ်ကတော့ စက်တခုနဲ့တူပါတယ်။ input င်လာတဲ့ တန်ဖိုးတွေကို

အခြေခံပြီး calculate လုပ်ပြီး output ထဲတိပေးတာပဲဖြစ်ပါတယ်။

Function ကိုသုံးရတဲ့ ရည်ရွယ်ချက်ကတော့ System processes တွေကို သပ်သပ်ဆီ

ခွဲချင်တဲ့ အခါ။ maintenance လွယ်ကူအောင်ရေး ချင်တဲ့ အခါ။

HTML ဘက်က event တွေကိုရေးတဲ့ အခါသုံးပါတယ်။



အရေးကြီးတာကတော့ င်လာတဲ့ input ကိုပဲ အခြေခံပြီး process လုပ်ရတာပါ။

Global variable မဟုတ်တဲ့ တွေး function တွေထဲက variable တွေသုံးမရပါဘူး။

return key ကတော့ optional ဖြစ်ပြီး တွေ့ ခေါ်လိုက်တဲ့ function ကို variable တခုထဲထည့်ချင်တယ်ဆိုရင်တော့ *return*

ကိုသုံးပေးရပါတယ်။ အဲလိုမှု မဟုတ်ဘူးဆိုရင်တော့ သုံးစရာမလိုပါဘူး။ *return* မပါတဲ့

function ကို တန်ည်းအားဖြင့် void လိုခေါ်ပါတယ်။

```
var x = myFunction(4, 3);
```

```
function myFunction(a, b) {  
    return a * b;  
}
```

11) DOM Events

- Mouse Events

onclick :

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function fun() {
        document.getElementById("demo").innerHTML = "Welcome From Ex;braiN!";
      }
    </script>
  </head>
  <body>
    <p id="demo">Click the following button</p>
    <button onclick="fun()">Click me</button>
  </body>
</html>
```

Click နှင့်လိုက်တဲ့အခါအလုပ်လုပ်မည့်event ဖြစ်ပါတယ်။

onmouseover :

```
<!DOCTYPE html>
<html>
  <body>
    <div
      onmouseover="mOver(this)"
      style="background-color: red; width: 150px; height: 150px; padding: 50px"
    >
      Mouse Over Me
    </div>

    <script>
      function mOver(obj) {
        obj.innerHTML = "Welcome";
        obj.style.backgroundColor = "green";
      }
    </script>
  </body>
</html>
```

Mouse တင်လိုက်တဲ့အခါဖြစ်လာတဲ့event ဖြစ်ပါတယ်။

onmouseout :

```
<!DOCTYPE html>
<html>
  <body>
    <div
      onmouseover="mOver(this)"
      onmouseout="mOut(this)"
      style="background-color: red; width: 150px; height: 150px; padding: 100px"
    >
      Mouse Over Me
    </div>

    <script>
      function mOver(obj) {
        obj.innerHTML = "Welcome";
        obj.style.backgroundColor = "green";
      }
      function mOut(obj) {
        obj.innerHTML = "Thank You";
        obj.style.backgroundColor = "blue";
        obj.style.color = "white";
      }
    </script>
  </body>
</html>
```

Mouse ခွဲလိုက်တဲ့အခါဖြစ်လာမည့်eventsပဲဖြစ်ပါတယ်။

onmousedown :

```
<!DOCTYPE html>
<html>
  <body>
    <div
      onmousedown="mdown(this)"
      style="background-color: red; width: 150px; height: 150px; padding: 100px"
    >
      Mouse Over Me
    </div>

    <script>
      function mdown(obj) {
        obj.innerHTML = "Welcome";
        obj.style.backgroundColor = "green";
      }
    </script>
  </body>
</html>
```

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

Mouse ကိုနိပ်ထားစဉ်ဖြစ်လာတဲ့event ဖြစ်ပါတယ်။

onmouseup :

```
<!DOCTYPE html>
<html>
<body>
<div
  onmousedown="mdown(this)"
  onmouseup="mup(this)"
  style="background-color: red; width: 150px; height: 150px; padding: 100px"
>
  Mouse Over Me
</div>

<script>
function mdown(obj) {
  obj.innerHTML = "Welcome";
  obj.style.backgroundColor = "green";
}
function mup(obj) {
  obj.innerHTML = "Thank You";
  obj.style.backgroundColor = "blue";
  obj.style.color = "white";
}
</script>
</body>
</html>
```

Mouse ကိုလွှတ်လိုက်တဲ့အချိန်ဖြစ်လာတဲ့event ဖြစ်ပါတယ်။

onmousemove :

```
<!DOCTYPE html>
<html>
  <body>
    <div
      onmousemove="move()"
      style="background-color: red; width: 50px; height: 50px; padding: 50px"
    ></div>
    <p id="demo"></p>
    <script>
      function move() {
        var number = Math.floor(Math.random() * 1000000);
        document.getElementById("demo").innerHTML =
          "Random number is :" + number;
      }
    </script>
  </body>
</html>
```

Mouse လှပ်ရှားနေတဲ့အခိုင်တိုင်းဖြစ်နေတဲ့event ဖြစ်ပါတယ်။

- Keyboard Events

onkeydown :

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function myFunction(obj) {
        document.getElementById("demo").innerHTML = obj.value;
      }
    </script>
    <style>
      #demo {
        color: blue;
      }
    </style>
  </head>
  <body>
    <label>Name:</label>
    <input type="text" onkeydown="myFunction(this)" />
    <p id="demo"></p>
  </body>
</html>
```

Keyboard မှုkey တခုခုကိုရှိက်လိုက်တာနဲ့ဖြစ်လာတဲ့event ဖြစ်ပါတယ်။

onkeyup :

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function myFunction(obj) {
        document.getElementById("demo").innerHTML = obj.value;
      }
    </script>
    <style>
      #demo {
        color: blue;
      }
    </style>
  </head>
  <body>
    <label>Name:</label>
    <input type="text" onkeyup="myFunction(this)" />
    <p id="demo"></p>
  </body>
</html>
```

Keyboard ကနေkey တခုခုကို ရှိက်ပြီး ပြန်လွှတ်လိုက်တဲ့အခါန် ဖြစ်လာတဲ့event ဖြစ်ပါတယ်။

▪ Form Events

focus :

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function firstfocus() {
        document.getElementById("fname").focus();
      }

      function secondfocus() {
        document.getElementById("lname").focus();
      }
    </script>
  </head>
  <body>
    <label>First Name:</label>
    <input type="text" id="fname" />
  </body>
```

```

<br />
<label>Last Name:</label>
<input type="text" id="lname" />
<p>
<input
  type="button"
  onclick="firstfocus()"
  value="Set First Name Focus"
/>
<input
  type="button"
  onclick="secondfocus()"
  value="Set Last Name Focus"
/>
</p>
</body>
</html>

```

Html element တခုခုကို focus (cursor) ချုပ်ထဲအခါ သုံးပါတယ်။

blur :

```

<!DOCTYPE html>
<html>
<head>
<script>
function namefocus() {
  document.getElementById("fname").focus();
}

function nameblur() {
  document.getElementById("fname").blur();
}
</script>
</head>
<body>
<label> Name:</label>
<input type="text" id="fname" />
<br />
<p>
<input type="button" onclick="namefocus()" value="Set Name Focus" />
<input type="button" onclick="nameblur()" value="Lost Name Focus" />
</p>
</body>
</html>

```

Focus ဖြတ်ချုပ်ထဲအခါသုံးပါတယ်။

onchange :

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function myFunc() {
        var x = document.getElementById("courses").value;
        document.getElementById("demo").innerHTML =
          "You selected: " + x + " Course";
      }
    </script>
  </head>
  <body>
    <p>Select Courses:</p>

    <select id="courses" onchange="myFunc()">
      <option value="Web Developer I">Web Developer I</option>
      <option value="Web Developer II">Web Developer II</option>
      <option value="Japanese Basic">Japanese Basic</option>
      <option value="Japanese Advance">Japanese Advance</option>
    </select>

    <p id="demo"></p>
  </body>
</html>
```

Html form tag အတွင်းမှာရှိတဲ့ tag တွေမှာသုံးပါတယ်။ User input လုပ်လိုက်လို နိဂုံကနေ ပြောင်းသွားတဲ့အခါ ဖြစ်လာတဲ့eventဖြစ်ပါတယ်။

- **Window/Document Events**

load :

```
<!DOCTYPE html>
<html>
  <body onload="myFunction()">
    <script>
      function myFunction() {
        alert("Loading Page.Click OK");
      }
    </script>
  </body>
</html>
```

Web pageစစချင်းဖြစ်တဲ့event ဖြစ်ပါတယ်။

resize :

```
<!DOCTYPE html>
<html>
  <body onresize="myFunction()">
    <p id="demo"></p>

    <script>
      function myFunction() {
        var w = window.outerWidth;
        var h = window.outerHeight;
        var txt = "Window size: width=" + w + ", height=" + h;
        document.getElementById("demo").innerHTML = txt;
      }
    </script>
  </body>
</html>
```

Web page ကို size ချို့ချယ့် တဲ့အခါ ဖြစ်လာတဲ့event ဖြစ်ပါတယ်။