

# **WEB PROGRAMMING**

**By Win Htut(GreenHackers)**

**Founder At National Cyber City**

## JWT(Json Web Token )

## HTML Tags

HTML သည် Hypertext Markup Language ကို ဆိုလိုချင်း ဖြစ်ပါတယ်။ ဒါဆို Hyper Text Markup Language ဆိုတာ ဘာကို ဆိုလို တာလဲ။ Markup Language တွေရဲ့ အဓိက ဝိသေသ လက္ဌဏာ တစ်ခုသည် tag တွေကို အသုံးပြုခြင်း ဖြစ်ပါတယ်။ ဥပမာ စာလုံး တစ်ခုကို ထင်ထင်ရှားရှား ပေါ်စေချင် တယ်ဆိုရင် <strong> ဆိုတဲ့ tag ကို သုံးရပါတယ် (<strong>Hello</strong>)။ Markup ဆိုတာကတော့ ဖော်ပြချင်တဲ့ documents တွေကို tags တွေနဲ့ ဝန်းရုထားတာကို ဆိုလို တာပါ ဥပမာ စာလုံး အကြီးနဲ့ ပေါ်စေချင်တယ်ဆိုရင် ကိုယ်ဖော်ပြချင်တဲ့ document ကို h1 tag နဲ့ ရေးရပါတယ် <h1> Hello </h1> ဒါဆိုရင် Hello ဆိုတဲ့ document ကို h1 tag နဲ့ ဝန်းရု ထားတာပါ။ Hypertext ထဲကမှ hyper အကြောင်းကို အရင် ရှင်းပါမယ် hyper ဆိုတာ hyperlinks တွေ လုပ်ဆောင် နိုင်စွမ်းကို ပြောတာပါ။ hyperlink or ( link for short) ဆိုတာကတော့ ကိုယ်က စာကြောင်း တစ်ကြောင်း သို့ မဟုတ်ပုံ တစ်ပုံကို click လုပ်တာနဲ့ တစ်ပြိုင်နက် တစ်ခြား page တစ်ခုကို ကူးပြောင်း နိုင်စွမ်း ရှုတာပါ။ ထိုကြောင့် HTML သည် markup tags တွေကိုသုံးနိုင်ပြီး အခြား သော web pages တွေကိုအလွယ်တကူ ကူးပြောင်းနိုင်တဲ့ language တစ်ခု ဖြစ်ပါတယ်။

HTML tags တွေဟာ အစုံလိုက် လာပါတယ်။ ဥပမာ <h1> Hello </h1> ဆိုတဲ့ နေရာမှာ <h1> သည် angled bracket ဖြစ်င့် အစ tag ဖြစ်ပြီး </h1> သည် အဆုံး tag ဖြစ်ပါတယ်။ tags တွေကို ရေးတဲ့ အခါမှာ အဆုံး tag ကိုပါ သေချာ တံတားကျကျ ရေးပေးရပါမယ်။ ဥပမာ <h1> hello </> ဆိုပြီး ရေးမယ် ဆိုလျှင်လည်း အလုပ် လုပ်မှာပါ သို့သော browser တိုင်းမှာတော့ လုပ်ချင်မှ လုပ်မှာပါ။ tags တွေထဲမှာ <h1> တင်မဟုတ်ဘူး h2 နှင့် h6 စတဲ့ tag တွေလည်း ရှုပါတယ် ထိုကြောင့် tag တွေကို ရေးတဲ့ အခါမှာ အစအဆုံးကို အတံအကျ ရေးပေးခြင်းဖြင့် error ဖြစ်ခြင်းကို ကာကွယ် နိုင်ပါတယ်။

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>Hello</title>
```

```
</head>
```

```
<body>
```

```
    <h6>Hello</h6>
```

```
<h1>MyBirthday</h1>
</body>
</html>
```

Tags တွေကို h1 tag and h2 tag စသာဖြင့် တံတိကျကျ ပြောတာမျိုး မဟုတ်ပဲ general အားဖြင့် elements တွေလို့ ခေါ်ပါတယ်။ ဥပမာ h1 element ပေါ့။ Element နှစ်မျိုး ရှိပါတယ် ပထမ တစ်မျိုးကတော့ container element လို့ ခေါ်ပြီး ဒုတိယ တစ်ခုကိုတော့ void elements လို့ ခေါ်ပါတယ်။ container elements တွေဟာ အစ နှင့် အဆုံး tag များ ပါဝင်ပြီး အလယ် မှာမှ ဖော်ပြချင်တဲ့ စာသားများ ပါဝင်ပါတယ်။ Void element ကတော့ tag တစ်ခု တည်း ပါဝင်ပါတယ်။ Void elements တွေ အကြောင်းကို နောက်ပိုင်ထပ်ပြီး ဆွဲးနွေးဖော်ပြသွားပါမယ်။

### Structural Elements

အထက်မှာ ရေးထားတဲ့ HTML program ကို source code တွေလို့ ခေါ်ပါတယ်။ ဘာလို့ source code လို့ ခေါ်သလဲ။ မည်သည့် programming languages ဖြင့် ရေးသည် ဖြစ်စေ program တစ်ပုဒ်မှာ code အမျိုး အစား နှစ်ခု ရှိပါတယ်။ ပထမ တစ်ခုသည် source code ဖြစ်ပြီး ဒုတိယ တတ်ခုသည် executable code ဖြစ်ပါတယ်။ Source code သည် programmer မှ ရေးလိုက်သော code များကို ဆိုလိုပြီး executable code သည် low-level code ဖြစ်ပြီး computer hardware တွေကသာ နားလည်ပြီး executes(runs) နိုင်ပါတယ်။ Executable code တွေကတော့ source code တွေမှ တစ်ဆင့် compiler ကိုသုံးပြီး ရရှိလာသော code များ ဖြစ်ပါတယ်။ ( ဤ နေရာတွင် compiler ဟု ဆိုလိုက်ခြင်းမှာ compile language တွေကို ဥပမာ ထားပြီး ဆိုလိုခြင်း ဖြစ်သည် ) Compiler ကို အသုံးပြုတဲ့ compile language နဲ့ interpreter ကို အသုံးပြုတဲ့ interpreted language ဆိုပြီး နှစ်မျိုး ရှိပါသေးတယ်။

HTML ကို လေ့လာတဲ့ အချိန်မှာတော့ compiled or interpreted တွေကို စိတ်ပူ စရာ မလိုပါဘူး ဘာကြောင့်လဲဆိုတော့ browser ကနေ html code တွေကို နောက်ကွယ်ကနေ ကိုင်တွေယ သွားတာ မို့ပါ။ Web pages တိုင်းမှာ HTML တည်ဆောက်ပုံနဲ့ ပတ်သက်ပြီး doctype , html , head and body ဆိုပြီး သွားပါတယ်။ သတိပြုရန် အချက်မှာ doctype သည် element မဟုတ်ပါဘူး သို့သော် ရေးတဲ့ အခါမှာတော့ ထိပ်ဆုံးမှာ အမြဲ ထည့်ရေးပါတယ် ဘာကြောင့် အမြဲ ထည့်ရေး ရသလဲ ဆိုတာ ကိုတော့ နောက်သင်ခန်းစာမှာ ရှင်းပြပါမယ်။ html , head , body စတဲ့ elements တွေကတော့ web page တစ်ခု ရဲ့ အခြေခံ ကျတဲ့ တည်ဆောက်သည့် ပုံစံ ဖြစ်ပါတယ် ထိုကြောင့် မို့လည်း ထို့ elements တွေကို structural elements တွေလို့ ခေါ်ဆိုခြင်း ဖြစ်ပါတယ်။

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="author" content="WinHtut">
<meta name="description" content="Pyin Oo Lwin conditions">
<title>Weather</title>
<style>
h1 {text-align: center;}
hr {width: 75%;}
</style>
</head>
<body>
<h1>PyinOoLwin City Weather</h1>
<hr>
<p>
It should be pleasant today with a high of 95 degrees.<br>
With a humidity reading of 30%, it should feel like 102 degrees.
</p>
<div>
Tomorrow's temperatures:<br>
high 96, low 65
</div>
</body>
</html>
```

အထက်ပါ program ကို run ကြည့်မည့် ဆိုလျင် အောက်ပါ အတိုင်း output ကို ရရှိမှာ ဖြစ်ပါတယ်။ file ကို run ရန် အရင်ဆုံး အထက်ပါ source code များ ကို မိမိ နှစ်သက်ရာ ide ထဲတွင် ရေးပါ ထို့နောက် file ကို test.html ဟု save ပါ။ ပြီးလျင် ထို file အား click နှစ်ချက် ခေါက်ပြီး မိမိ နှစ်သက်ရာ browser ဖြင့် run နိုင်ပါပြီ။

---

### PyinOoLwin City Weather

---

It should be pleasant today with a high of 95 degrees.  
With a humidity reading of 30%, it should feel like 102 degrees.

Tomorrow's temperatures:  
high 96, low 65

```
<!DOCTYPE html>
```

အထက်မှာ run လိုက်တဲ့ program ရဲ့ source code များကို တစ်ကြောင်းချင်းစီ step by step ရှင်းလင်းသွားပါမယ်။ ပထမစ ရေးတာကတော့ <!DOCTYPE html> ဖြစ်ပါတယ် !DOCTYPE သည် browser အား ယခု ရေးမည့် web page သည် ဘယ်လို့ document အမျိုးအစား ဖြစ်တယ် ဆိုတာ ကို ပြောလိုက်တာပါ။ DOCTYPE နောက်တွင် html ပါ ပါသည့် အတွက် HTML document အမျိုးအစားကို သုံးမယ်လို့ ပြော လိုက်တာပါ။ ပုံပြီး တိတိကျကျ ပြောမည့်ဆိုလျှင် HTML5 standard ကို သုံးမယ်လို့ ပြောလိုက်တာပါ။ HTML5 standard ဆိုတာက HTML5 မှာ ရေးသည့် ရေးထုံးရေးနည်း syntax တွေ အတိုင်း ရေးမယ်လို့ ပြောလိုက်တာပါ။

```
<html lang="en">
```

!DOCTYPE ပြီးသွားတဲ့ အချိန်မှာ html element ဆက်ရေးလာပါတယ်။ အရင်သင်ခေါ်စာမှာ ရေးခဲ့ပြီး ဖြစ်သလို ထို container ထဲမှာ lang="en" ဆိုပြီး ပါလာပါတယ်။ ဆိုလိုတာကတော့ web page ကို english ဖွင့် ရေးသားထားခြင်း ဖြစ်ကြောင်း browser ကို ပြောခြင်း ဖြစ်ပါတယ်။ ပြီးလျှင် နောက်ထပ် container တစ်ခုဖြစ်တဲ့ head လာပါတယ် head ထဲမှာတော့ web page တစ်ခုလုံးနဲ့ ပတ်သက်တဲ့ အချက်အလက်တွေကို ဖော်ပြပါတယ်။ ဘယ်လို့ အကြောင်းအရာတွေ ဖော်ပြလဲဆိုတာ အောက်မှာ ဆက်ကြည့်ကြပါစို့။

Container elements တွေဟာ nested တွေ ဖြစ်ပါတယ်။ ဘာကြောင့်လဲဆိုတော့ head ဆိုတဲ့ container ထဲမှာ နောက်ထပ် body ဆိုတဲ့ container ထပ်ပါ လာပါတယ် ထိုနည်းတယ် <b> , <p> စတဲ့ elements တွေလည်း ပါဝင်နေသေးတဲ့ အပြင် အထဲမှာ ရှိနေတဲ့ container element တွေဟာ အပြင်ဘက်ဆုံးမှာ ရှိနေတဲ့ container အပိတ် နေရာကို မရောက်ခင် ပြန်ပြတော့ အပိတ် tag ကို ရေးရပါမယ်။ ဥပမာ <body> Hello <h2>world</h2> </body> ယခု code မှာဆုံး ကြည့်ပါ body ဆိုတဲ့ nested container သည် အပြင်ဘက်ဆုံးမှာ ရှိနေပါတယ် ထို အထဲမှာ h2 ဆိုတဲ့ element ထပ်ရှိနေပါတယ် h2 ဆိုတဲ့ element သည် body container အပိတ် tag မရေးခင် သူ့ရေးမှု။ မှာ အပိတ် tag ရေးရပါမယ်။ အထက်တွင် အပြောရောင်ဖွင့် ဖော်ပြထားပါသည်။

### meta Element

```
<meta charset="utf-8">
<meta name="author" content="WinHtut">
<meta name="description" content="Pyin Oo Lwin conditions">
```

Head container ထဲမှာ ရှိတဲ့ meta elements web page နဲ့ပတ်သက်တဲ့ အချက်လက်  
တွေကို ဖော်ပြန့် သုံးပါတယ်။ meta element သည် void element ဖြစ်ပြီး container  
မဟုတ်ပါ အဘယ်ကြောင့် ဆိုသော် meta element တွင် end tag မပါ ဝင်ပါ။ web  
programmers အများစု သည် meta elements တွေကို ရေးတဲ့ အခါ အဆုံးတွင် space  
တစ်ခုနှင့် / တစ်ခု ကို ထည့်ရေးကြပါတယ်။ ဥပမာ-

```
<meta charset = "utf-8" />
```

Utf-8 encoding system တစ်ခု ဖြစ်ပါတယ် နောက်ပိုင်း သင်ခန်းစာတွေမှာ အသေးစိတ်  
ထပ်ရှင်းပြပါမယ်။ Charset attributes လို့ ခေါ်ပါတယ် ထို attributes တွေ  
အကြောင်းကို title element ပြီးလျှင် HTML attributes ခေါင်းစဉ်ဖြင့် ဆက်လက် ဆွဲးနွေး  
သွားပါမည်။

```
<title>Weather</title>
```

### title Element

Title element သည် head element ထဲတွင်ပါဝင်ပါတယ်။ title element ထဲမှာ data  
တစ်ခု ပါဝင်ပါတယ် ထို data သည် browser ရဲ့ title bar မှာ ပေါ်မှာ ဖြစ်ပါတယ်။ HTML  
standard အရ head container ထဲမှာ title element ပါဝင်ရပါမယ် ဘာကြောင့်လဲ ဆိုတော့ ထို  
title သည် web search engine ကို ကူညီပေးတဲ့ အတွက် ကြောင့် ဖြစ်ပါတယ်။ User ကနေ  
တစ်ဆင့် web search engine မှာ information တစ်ခုကို ရှာတဲ့ အခါမှ ထို web search engine မှ  
ထို user ထည့်ပေးလိုက်သည့် information နဲ့ ဆိုင်သည့် web page များစွာကို link များနှင့်  
တက္က ပြန်ပေးပါတယ်။



### HTML Attributes

Container elements တွေကြတော့ Information တွေကို သူတို့ရဲ့ start and end tag  
တွေ ကြားမှာ ဖော်ပြ ကြပါတယ် သုံးသော် meta element တွေလုံမျိုး void element  
တွေကြတော့ end tag မပါတဲ့ အတွက် Information တွေကို container elements တွေ အတိုင်း  
ဖော်ပြ ပေးလို့ မရပါဘူး။ ထိုကြောင့် ဖော်ပြချင်တဲ့ information တွေကို attributes တွေ  
သုံးပြီး ဖော်ပြ ပေးကြပါတယ်။ ဥပမာ <meta charset = "utf-8" /> charset သည် attribute

ဖြစ်ပါတယ်။ attributes တွေထကို value တွေ assigned လုပ်ပြီး ထည့်ပါတယ် utf-8 ဆိုတဲ့ value ကို charset ထဲသို့ = သုံးပြီး ထည့်တာ မျိုးပါ သို့သော value မထည့်တဲ့ attributes တွေလည်း ရှိပါသေးတယ် နောက်ပိုင်းမှာတော့ အသေးစိတ် ထပ်မံ့ပြပါဘီးမည်။ တစ်ခု သတ်ပြုရန်မှာ value တွေကို double quotes နဲ့ရေးပေးရပါတယ်။ Attributes တွေကို void elements တွေနဲ့ပဲ တွဲသုံးတာ မဟုတ်ပဲ container element တွေနဲ့လည်း တွဲသုံးပါတယ်။ ဥပမာ

```
<html lang="en">
.
.
</html>
```

Lang attribute က ကြတော့ browser ကို သူဖော်ပြထားတဲ့ language ဖြင့် သုံးမယ်ဆို ဆုံးလို လိုက်တာပါ။ ဥပမာ <html lang="en"> <h1> Hello World </h1> </html> Hello World ဆုံးတဲ့ စာသားကို english language နဲ့ပဲ ဖော်ပြ မယ်လို့ ပြောလိုက်တာပါ။ ကိုယ်တိုင် web page တွေ ရေးတဲ့ အခါကြရင် lang attribute ကိုပါ ထည့်ရေးသင့်ပါတယ် ဘာလို့ ထည့်ရေးသင့်တာလည်း ဆုံးတော့ ထိုသို့ ထည့်ရေးရင် ကောင်းတဲ့ အချက်တွေ ရှိပါတယ်။ ဘယ်လို့ ကောင်းတဲ့ အချက် ရှိလဲ ဆုံးတာကို W3C's Internationalization တွေမှာ ဖော်ပြထားပါတယ်။

<https://www.w3.org/International/questions/qa-lang-why.en>

- သီးခြား မိမိ အသုံးပြုလိုသည့် language များကို အသုံးပြုခြင်းဖြင့် web page ကိုရှာတဲ့ အခါမှာ ပိုပြီး လျှင်မြန်အောင် search engines တွေကို ကူညီပေးပါတယ်။
- စာလုံးပေါင်းတာ တွေ နဲ့ grammar စစ်ဆေးတာတွေကို ပိုမို လွယ်ကူစွာ လုပ်ဆောင် နိုင်ပါတယ်။
- Browser ကိုလည်း သင့်တော်တဲ့ fonts တွေ အသုံးပြုဖို့ ကူညီ ပေးနိုင်ပါတယ်။

#### meta charset Element

အထက်တွင် ရေးသားပဲ syntax များကို အသေးစိတ် လေ့လာပြီး ပြီဆုံးတော့ ယခု ဆက်ပြီး semantic အကြောင်းကို ဆက်ရှင်းပြ သွားပါမယ်။ syntax သည် code ရေးတဲ့ အခါမှာ လိုက်နာရမယ့် ရေးသားပဲ ရေးသားနည်းတွေကို ဆုံးလိုခြင်း ဖြစ်သလို semantic သည် code ရဲ့ အဓိပ္ပာယ်( meaning ) ဖြစ်ပါတယ်။ ပထမဆုံး အနေဖြင့် meta charset အကြောင်းကို အသေးစိတ် ရှင်းပြ သွားပါမယ်။

#### Web Server Encode System

Web server သည် အသုံးပြုနေသော user ရဲ့ web browser သို့ အချက် အလက်တွေ ပို့သော အခါတွင် source code တွေကို တိုက်ရှိက် ပို့တာ မဟုတ်ပါဘူး။ အချက် အလက်တွေကို coded လုပ်ပြီးမှ ပိုပါတယ် ဘယ်လို coded လုပ်တာလည်း ဆိုတော့ binary နဲ့ လုပ်ပါတယ်။ binary ဆုံး 0 and 1 တိုကို ဆုံးလိုခြင်း ဖြစ်ပြီး server မှ တစ်ဆင့် web browser ဆီသို့ ပိုပေးပါတယ်။ encoding လုပ်တဲ့ စနစ်တွေ အများကြီး ရှိပါတယ် ပိုတဲ့ဘက် ကနေ ပိုပေးလိုက်တဲ့ encoded data တွေကို လက်ခံမယ့် User ဘက်ကလည်း သံရှိထားဖို့ လိုအပ်ပါတယ် သို့မှာသာ အချက်လက်တွေကို လက်ခံ နိုင်မှာပါ။ Web page နဲ့ ပတ်သက်ပြီးတော့ meta charset ကို သုံးပါတယ် ပုံမှန်အားဖြင့် charset မှာ သူရဲ့ value ကို utf-8 ကိုသုံးကြပါတယ် ဘာကြောင့်လဲ ဆုံးတော့ ယခုခေတ် modern browsers အားလုံးက နားလည် ကြပါတယ်။ UTF-8 သည် UCS Transformation Format - 8 bit ကိုဆုံးလိုခြင်း ဖြစ်ပါတယ်။

### meta name Element

Meta elements အများစုကို name attribute နဲ့ တွဲသုံးပါတယ် ဘာအတွက်လို ဆုံးတော့ ဘယ်လို information အမျိုးအစား ဖြစ်သလဲဆုံးတာကို ဖော်ပြဖို့ပါ။ အထက်မှာ ရေးထားပြီးသား program အရဆိုလျှင် name and content နှစ်ခုသည် meta element ရဲ့ attribute တွေ ဖြစ်ပါတယ် သလို သူတို့ နှစ်ခုကို တွဲသုံးလေ့ ရှိပါတယ်။ name သည် ဘယ်လို information အမျိုးအစား ဖြစ်တယ်ဆုံးတာကို ဖော်ပြပြီး content သည် information ကို ဖော်ပြပါတယ်။

```
<meta name="author" content="WinHtut">
```

အဘယ်ကြောင့် ထိုကဲ့သို့ အချက် အလက်များကို မိမိတို့ program တွင် ထည့်ပေးရသနည်း။ ဘာကြောင့်လဲ ဆုံးတော့ program တစ်ပုံဒုက္ခက်တပ်သူတစ်ယောက်က ထပ်ရေးရတော့မယ် ဆုံးလျှင် meta ကို သုံးပြီး ပထမ သူသည် ဘယ်သူ ဖြစ်ကြောင်းကို အတိအကျ ဖော်ပြ ပေးထားတဲ့ အတွက် ဒုတိယ လူက program ကို ပြင်လိုသော အခါ ပထမလူဆီမှ အကူအညီကို အလွယ်တကူ တောင်းခံ နိုင်သည် သို့မဟုတ် program နှင့် ပတ်သက်ပြီး သံလို သော အချက်များကို အလွယ်တကူ မေးမြန်းနိုင်သည်။ နောက်ထပ် name attribute ရဲ့ description ကိုလည်း သုံးပါသေးတယ် သူ့ကိုတော့ web page မှာ ဘယ်လို information တွေ ဖော်ပြ မလဲ ဆုံးတာ သိနိုင်ဖို့ သုံးပါတယ်။ ဥပမာ ဆုံးလျှင် Pyin Oo Lwin မြို့ရဲ့ အခြေနေပါ့။ အကယ်၍ မိမိတို့ Program ရေးတဲ့ အခါမှာ meta ကိုရေးရတာ ရှုပ်တယ် အချိန်ကြာတယ် information တွေကို ဖော်ပြဖို့ အခြားနည်းလမ်းတွေ ရှိနေတယ် ဆုံးရင်တော့ meta description ကို မရေးပဲ ကျော်သွား နိုင်ပါတယ်။

```
<meta name="description" content="Pyin Oo Lwin conditions">
```

### Body Elements: hr , p , br ,div

ယခု သင်ခန်းစာမျက်တွေ body container မှာ ထည့်ရေးတဲ့ hr , p , br , div စတဲ့ elements တွေအကြောင်းကို ဖော်ပြသွားမှာပါ။ hr element တွင် h သည် horizontal ကို ဆိုလိုခြင်း ဖြစ်ပြီး r သည် rule ကို ဆိုလိုခြင်း ဖြစ်ပါသည်။ information တွေကို straight line ပုံစံဖြင့် ဖော်ပြလိုတဲ့ အခါမှာ သုံးပါတယ် ယခု web page မှာတွေ horizontal မျဉ်းလိုင်း တစ်ခု ဆွဲထားပါတယ်။ hr element သည် void element ဖြစ်ပြီး tag တစ်ခုသာ ပါဝင်ပါသည်။ p element သည် container element အမျိုးအစား ဖြစ်ပြီး စကားလုံးများကို စုပြုး paragraph (စာပိုဒ်) ပုံစံဖြင့် ဖော်ပြလိုတဲ့ အခါမှာ သုံးပါတယ်။ br tag ကတော့ နောက်တစ်ကြောင်းကို ဆင်းချင်တဲ့ အခါမှာ သုံးပါတယ်။ div ကတော့ container element အမျိုးအစား ဖြစ်ပြီး စုပြုး ဖော်ပြလိုတဲ့ အခါမှာ သုံးပါတယ် ထို့ပြင် project ကိုးလာတဲ့ အခါ div အများကြီးခဲ့ပြီး class များနှင့် တွဲပြီးတော့လည်း အသုံးပြုပါတယ်။ နောက်ပိုင်းမှာတော့ div နဲ့ ပတ်သက်ပြီး အသေးစိတ် ထပ်ဖော်ပြသွားပါဉီးမယ်။

<style>

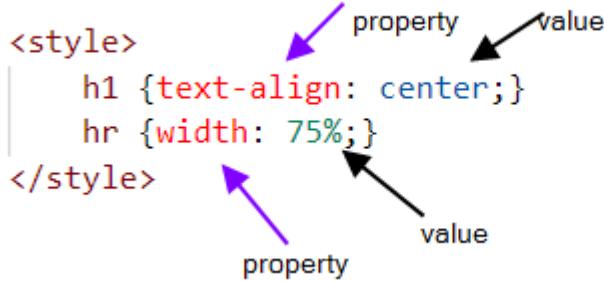
```
h1 {text-align: center;}
hr {width: 75%;}
</style>
```

အထက်က program ထဲမှာ css ကိုပါ ထည့်ထားတဲ့ အတွက် css များအကြောင်းကို ရှင်းသွားပါမယ်။

### Cascading Style Sheets

Cascading Style Sheets CSS သည် web page ကို မိမိတို့ လိုသလို ပုံဖော်နိုင်ရန် သုံးပါတယ်။ CSS နဲ့ ပတ်သက်တဲ့ code တွေကို .css ဆိုပြီး file တစ်ခု ထပ်ဆောက်ကာ အသုံးပြုသလို style ဆိုတဲ့ container ကိုသုံးပြီးတော့လည်း ထို container ထဲမှာ ရေးကြပါတယ်။ ယခု သင်ခန်းစာမျက်တွေ style container ကိုသုံးပြီး ရေးထားပါတယ်။ သတ်ပြုရန် အချက်မှာ style container သည် head container ထဲမှာ ရှိရမှာ ဖြစ်ပါတယ်။

ယခု program ထဲမှာတွေ style ထဲမှာ လိုင်းနှစ်လိုင်းသာ ရေးထားပါတယ် ထိုလိုင်းများကို rules လို့ ခေါ်ပါတယ်။ ဆိုလိုချင်တာကတော့ body ထဲမှာ ပါတဲ့ h1 and hr အားလုံးကို applied လုပ်မှာပါ။ CSS rule တစ်ခုစီတိုင်းမှာ CSS property and CSS value ရှိပါတယ် ထို property and value ကို colon နဲ့ ခြားထားပါတယ်။



First rule ဖြစ်တဲ့ h1 မှာ property ၏ text-align ဖြစ်ပြီး center ကတေသ့ value ဖြစ်ပါတယ် ဆိုလိုတာက h1 နဲ့ ရေးထားတဲ့ စာအားလုံးသည် web page ရဲ့ center မှာ ပေါ်မှာ ဖြစ်ပါတယ်။ second rule ဖြစ်တဲ့ hr မှာတေသ့ width ကို 75% ဟု ပေးထားတာပါ။

## PyinOoLwin City Weather

It should be pleasant today with a high of 95 degrees.  
With a humidity reading of 30%, it should feel like 102 degrees.

Tomorrow's temperatures:  
high 96, low 65

ပုံမှန်အားဖြင့် h1 သည် web page ရဲ့ left မှာ ပေါ်မှာ ဖြစ်သလို horizontal မျဉ်းလိုင်းသည်လည်း web page တစ်ခုလုံး အပြည့်ပေါ်နေမှာ ဖြစ်ပါတယ်။ သို့သော် style ထဲတွင် h1 text-align ကို value အနေဖြင့် center သို့ပိုလိုက်သလို hr width ကိုလည်း 75% သာ ပေးလိုက်တဲ့ အတွက် မျဉ်းလိုင်းသည် web page ရဲ့ လေးပုံ သုံးပုံသာ ပေါ်ခြင်း ဖြစ်ပါတယ်။ အောက်မှာ 50% or 100% အတွက် စမ်ပြထားပါတယ်။

Hr 50%

## PyinOoLwin City Weather

Hr 100%

## PyinOoLwin City Weather

အကယ်၍ အထက် ခေါင်းစဉ်းများ နေရာမှာ link များပါ တဲ့ ထည့်ချင်တယ် ဆိုရင်တော့ Attributes များကို ပါ တဲ့ ထည့်လို့ ရပါသေးတယ်။ link တွေ ချိတ်ဖို့ အတွက် ဆိုရင် href ဆိုတဲ့ attribute ကို သုံးပါတယ် ထို href နောက်ကမှ မိမိတို့ သွားစေချင်သော page ရဲ့ link ကို ထည့်ပေးရပါတယ်။ စာရေးသူ အနေဖြင့် ယခု မိမိတို့ ရေးနေသော html file ရှိသည့် နေရာမှာပင် နောက်တပ် 4-href.html ဆိုသည့် file တစ်ခု ထပ်ဆောက်လိုက်ပါတယ်။ html file များကို ချိတ်သည့် အခါတွင် ချိတ်ဆက်ချင်သော file များသည့် folder တစ်ခုအောက်ထဲမှာ

ရှိနေလျင် ပိုအဆင်ပြေပါသည် အကယ်၍ မရှိဘူး ဆိုလျင် file location အတိအကျကို href နောက်တွင် value အနေဖြင့် ဖော်ပြု ပေးရပါသည်။

```
<h1>PyinOoLwin City Weather</h1>
To New Link
<hr>
```

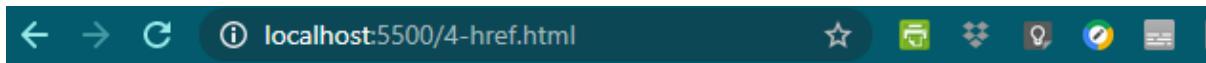
အထက်ပါ အတိုင်းရေးပြီး run ကြည့်ပါက အောက်ပါ အတိုင်း ပေါ်နပါမည် ထိုကဲ့သို့ ပေါ်နေသော link ကို နှိပ်ကြည့်လိုက်ပါက နောက်ထပ် page တစ်ခုသို့ ရောက်သွားမှာဖြစ်ပါတယ်။

## PyinOoLwin City W

[To New Link](#)

It should be pleasant today with a high of 95 degrees.  
With a humidity reading of 30% it should feel like 102 degrees.

To New Link ကို နှိပ်လိုက်လျင် အောက်ပါ အတိုင်း ပေါ်လာပါမည်။ စာရေးသူ အနေဖြင့် 4-href.html ထဲတွင် Hello this is from another page ဆိုတဲ့ စာသားကိုရေးထားပါသည်။



## Hello this is from another page

မိမိတို့ ရဲ့ web page မှာ image တစ်ခုပါ ထည့်ချင်ရင်တော့ img tag နှင့် src attribute တို့ကိုသုံးနိုင်ပါတယ်။ စာရေးသူ အနေဖြင့် body tag ထဲတွင် img tag တစ်ခုကို ထပ်ထည့်ပြီး ပုံတစ်ပုံကို ဖော်ပြုပါမည် ထိုသို့ ဖော်ပြု နိုင်ရန် src attribute နောက်တွင် မိမိတို့ပဲရဲ့ Location ကို အတိအကျ ထည့်ပေးရမှာ ဖြစ်ပါတယ်။ image ထည့်သော အခါတွင် image ရဲ့ name ကို အတိအကျ ရေးပေးရမှာ ဖြစ်သလို extension ကိုပါ ထည့်ပေးရမှာ ဖြစ်ပါတယ် အောက်တွင် စာရေးသူရဲ့ image name သည် pol ဖြစ်ပြီး extension name သည် jpg ဖြစ်ပါသည်။

```
<h1>PyinOoLwin City Weather</h1>
To New Link
<hr>
 ←
<p>
It should be pleasant today with a high of 95 degrees.<br>
```

အထက်ပါ အတိုင်းရေးပြီးလျှင် web page အားပြန်ကြည့်လိုက်ပါက image တစ်ခု ပေါ်နေသည်ကို တွေ့ရပါမည်။ image ထည့်ခြင်းသည်လည်း link ချိန်ခြင်းအတိုင်း folder တစ်ခုတည်းအောက်၍ ရှိနေလျှင် ပုံအဆင်ပြေပြီး အကယ်၍ မရှိလျှင် image ရှိတဲ့ Location ကို အတံအကျ ထည့်ပေးရမှာ ဖြစ်ပါတယ်။

## PyinOoLwin City Weather

[To New Link](#)



### URL(Uniform Resource Locator)

URL(Uniform Resource Locator) ဆိုတာ address တစ်ခု ဖြစ်ပါတယ် မိမိဖော်ပြချင့်တဲ့ document သို့မဟုတ် World Wide Web (WWW) ပေါ်မှ အချက်လက်များကို ဖော်ပြလိုတဲ့ အခါမှာ သုံးပါတယ် ဥပမာ <https://www.google.com> လိုမျိုးပါ။ Src attribute ကိုသုံးတဲ့ အခါမှာ URL နှစ်မျိုးရှိပါတယ် Absolute URL and Relative URL ပါ။ Absolute URL က ကြတော့ အခြားသော web site တစ်ခုကို လုမ်းညွှန်းခြင်း သို့မဟုတ် အခြားသော website တစ်ခုမှ ပုံ သို့မဟုတ် စာ အချက်အလက်တို့ကို အတံအကျ ဖော်ပြဖို့ အတွက်သုံးပါတယ် ဥပမာ <https://medium.com/@winhtutonline> လိုမျိုးပါ။ Relative URL က ကြတော့ အထက်မှာ ရေးခဲ့သည့် program အတိုင်းသုံးတဲ့ ပုံစံမျိုးပါ မိမိတို့ server သို့မဟုတ် မိမိတို့ hosted လုပ်ထားတဲ့ (current directory )နေရာက ပုံကို ပြန်သုံးတာပါ။

<img> tag ထဲမှာ အခြားသော attribute တွေလည်း ထပ်ထည့်နိုင်ပါသေးတယ် ဥပမာ image ရဲ့ width and height ပါ ကိုယ်ပေးချင့်တဲ့ value ကိုပေးနိုင်ပါတယ်။

```

```

အကယ်၍ အထက်ပါ အတိုင်းရေးလိုက်မည်ဆိုလျှင် Image ကို width 200 နှင့် height 200 တို့ဖြင့် size ချိန်းပြီး ဖော်ပြ ပေးမှာ ဖြစ်ပါတယ်။

[To New Link](#)



It should be pleasant today with a high of 95 degrees.

ပုံ တစ်ပုံ ကို ဖော်ပြတဲ့ အခါမှာ ထိပုံသည် ပျက်သွားတာမျိုးပဲ ဖြစ်ဖြစ် အကြောင်း တစ်ခုခုကြောင်း မရှိတော့ဘူးဆိုရင် ပုံကို ဖော်ပြလို့ မရတော့ပါဘူး ထိုကဲ့သို့သော အခြေနေမျိုးမှာ ဆိုလျှင် alt attribute ကိုသုံးနိုင်ပါတယ်။ Alt attribute က ဘာလုပ်ပေးလဲ ဆိုတော့ ကိုယ်ဖော်ပြချင်ပုံ မရှိတော့ဘူးဆိုလျှင် ထို ပုံနေရာမှာ စာသား ဖော်ပြ ပေးပါတယ်။ ဥပမာ စာရေးသူ အနေဖြင့် အထက်မှာ ပြင်ညီးလွှင်မြို့ပုံ ထည့်ထားတာ အကယ်၍ မရှိတော့ဘူးဆိုလျှင် ထိုနေရာ Pyin Oo Lwin City စသေဖြင့် မိမိဖော်ပြချင်တဲ့ စာသားများကို ဖော်ပြ နိုင်ပါတယ်။ alt attribute ကို အသုံးပြုလိုလျှင် img tag မှာထည့်ပိုး အောက်ပါ အတိုင်းအသုံးပြု နိုင်ပါတယ်။

```

```

### HTML Paragraph

ရှေ့ပိုင်း သင်ခန်းတာမှာ p သည် container element အမျိုးအစား ဖြစ်တယ်ဆိုတာကို ပြောခဲ့ပြီး ဖြစ်ပါတယ်။ p tag အစနဲ့ အဆုံးကြားမှာ ရှိတဲ့ စာသားများကို paragraph ပုံစံ ဖြင့် ဖော်ပြ ပေးပါတယ်။ p tag ကိုသုံးလိုက်တိုင်းမှာ လိုင်းအသစ် ကနေ ပြန်စ ပေးပါတယ်။ ထိုပြင် p tag ကိုသုံးသော အခါ browser ကနေ အလုံလျောက် space ခြားပေးပါတယ်။ p tag ကိုသုံးပြီး ရေးထားတဲ့ စာသားတွေထဲမှာ space တွေ အများကြီး ပါနေတယ် ဆိုရင်လည်း browser ကနေ အလုံလျောက် ဖျက်ပေးပါတယ်။ အောက်မှာ sample program လေး ရေးပြထားပါတယ်။

<p>

It should be pleasant today with a high of 95 degrees.<br>

With a humidity reading of 30%,



it should feel like 102 degrees.

</p>



30% ရဲ့ နောက်မှာ အောက်ကို တစ်ကြောင်း ဆင်းထားသလို space တွေလည်း အများကြီး ခြားထားပါတယ် သို့သော် p tag ကိုသုံးတာ ဖြစ်တဲ့ အတွက် space တွေသာယ်လောက် ခြားခြား line တွေ ဘယ်လောက် ကြဲထားပါစေ browser ကနေ အလိုလျောက် ပြုပြင်ပေးသွားမှာ ဖြစ်ပါတယ်။ အကယ်၍ မိမိ သဘော အရ တစ်ကြောင်း ဆင်းချင်တယ် ဆုံးရှင်တော့ br tag ကိုသုံးပြီး ဆင်းနိုင်ပါတယ်။ output ကို အောက်မှာ ဖော်ပြထားပါတယ်။

It should be pleasant today with a high of 95 degrees.  
With a humidity reading of 30%, it should feel like 102 degrees.

P tag သုံးပြီးရေးထားတဲ့ Paragraph ကိုပဲ သီးသန့် ပြင်ချင်တယ် ဆုံးရှင်တော့ style attribute ကိုသုံးနိုင်ပါတယ်။ Style အကြောင်းကို မပြောခင်မှာ paragraph ရဲ့ အားနည်းချက် သို့မဟုတ် အသုံးချမှုကိုပဲ ဆက်ပြောသွားပါမယ်။ Paragraph ကိုသုံးပြီး ကဗျာ တစ်ပုဒ်ရေးချင်တယ် ဆုံးပါစုံ မိမိထားချင်သလို spacing များကိုထားပြီး မိမိတို့ အောက်ကို တစ်ကြောင်းဆင်ချင်ရင်လည်း ဆင်းတဲ့ အတိုင်းပဲ ထားချင်တယ် ဆုံးလျှင် p tag သည် spacing များနှင့် Line break များကို အလိုလျောက် ပြုပြင်တဲ့ အတွက် အခက်ခဲ့ရှိနေပါတယ်။ ထိုပြုသနာကို ဖြေရှင်းဖို့ အတွက်ဆုံးလျှင် pre tag ကိုသုံး နိုင်ပါတယ်။ pre tag သည် မိမိတို့ ရေးလိုက်တဲ့ အတိုင်း spacing များနှင့် Line break များကိုပါ ပြုပြင်ခြင်း မရှိပဲ မူရင်း အတိုင်း ပြန်လည် ဖော်ပြပါ ပေးပါတယ်။

```

<pre>
    It should be pleasant
    today with a high of 95 degrees.           ←
    With a humidity reading of                 30%,           ←
                                                ←
        it should feel like                  102 degrees.
    </pre>
    ..

```

### Output

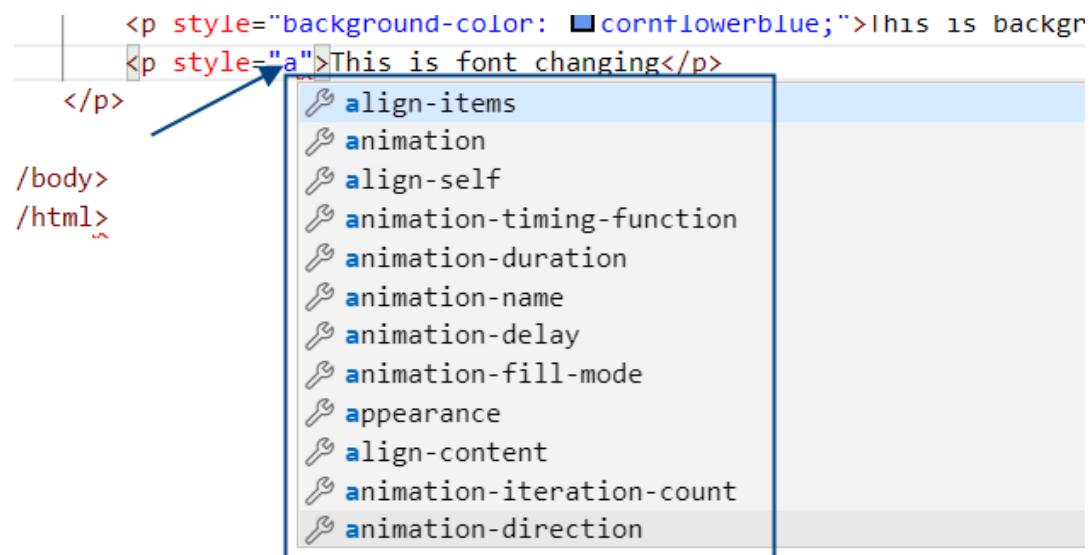
It should be pleasant  
today with a high of 95 degrees.  
With a humidity reading of 30%,  
  
it should feel like 102 degrees.

P tag ဖြင့် ရေးထားသည့် paragraph များကို styling စလုပ်ကြည့်ပါမည်။ Styling လုပ်ရန် အတွက် style attribute ကိုသာ အသုံးပြုရန် လိုအပ်ပါသည်။ သူ့နောက်မှာမှ property and value တို့ လိုက်ပါသည်။ ရောသားရမည့် syntax မှာ အောက်ပါ အတိုင်းဖြစ်သည်။

```
<tagname style="property:value;">
```

```
<p style="color:blueviolet">Hello This is Formal Paragraph With Styling<br>
GreenCoders<br>
<p style="background-color:cornflowerblue;">This is background-color</p>
<p style="font-size: 70px;">This is font changing</p>
</p>
```

Style attribute နှင့် မိမိတို့ အသုံးပြု နိုင်သည့် properties များစွာ ရှိပါသည် ပါဝင်သည့် properties များကို သိရှိ နိုင်ရန် style နောက်တွင် a တစ်လုံးတည်း စပီး ရေးကြည့်ပါ ထို့နောက် မိမိတို့ အသုံးပြုလို ရနိုင်မည့် properties များစွာကို တွေ့မြင်ရမှာ ဖြစ်ပါတယ်။ ထိုနည်းတူ b ,c ,d e, f ... z ထို့ ရေးကြည့်ပြီး properties များကို စမ်းသုံးခြင်းဖြင့် လေ့လာနိုင်ပါသည်။



အကယ်ရှိ ထိုကဲ့သို့ မပေါ်လာပါက HTML CSS Support Extension ကို မိမိတို့ အသုံးပြုသည့် ide or editor တွင် Install လုပ်ပေးရန် လိုအပ်ပါသည်။

စာပိုဒ်တွေကို ဖော်ပြတဲ့ အခါမှာ အထဲမှာ ရေးသည့် text များကို လိုအပ်သလို ပြင်နိုင်ဖို့ HTML မှာ Formatings Elements တွေလည်း ရှိပါသေးတယ် တစ်နည်းအားဖြင့် Inline Text Semantics လိုလည်း ခေါ်ပါတယ်။

## Attributes and Elements

### Accept

Accept attribute ကို <form><input> စတဲ့ elements တွေနဲ့ တဲ့သုံးပါတယ်။ Accept ကို user တွေထည့်ပေးလိုက်တဲ့ file တွေကို accept လုပ်ဖို့ သုံးပါတယ်။ File တွေ အများကြီးကို accept လုပ်နိုင်ဖို့ အတွက်လည်း file type တွေကို, တွေခြားပြီး ဖော်ပြပေးထားလိုပါတယ်။ ဥပမာ microsoft word file တွေသို့မဟုတ် doc file တွေကို များစွာလက်ခံ နိုင်ဖို့ အတွက်ဆိုလျှင် accept attribute နှင့်အောက်ပါ အတိုင်း ဖော်ပြ နိုင်ပါတယ်။

```
accept=".doc, .docx, application/msword, application">
```

အကယ်၍ user ဆိုကနေ audio file, image file သို့မဟုတ် video file များကို လက်ခံ လိုတယ် ဆိုပါက အောက်ပါ အတိုင်းရေးနိုင်ပါတယ်။ /\* သည် မည်သည့် audio file မဆို လက်ခံ နိုင်တယ်လို့ ဆိုလိုတာပါ ဥပမာ audio/\* ဆိုလျှင် မည်သည့် audio file မဆို လက်ခံ နိုင်သည်။

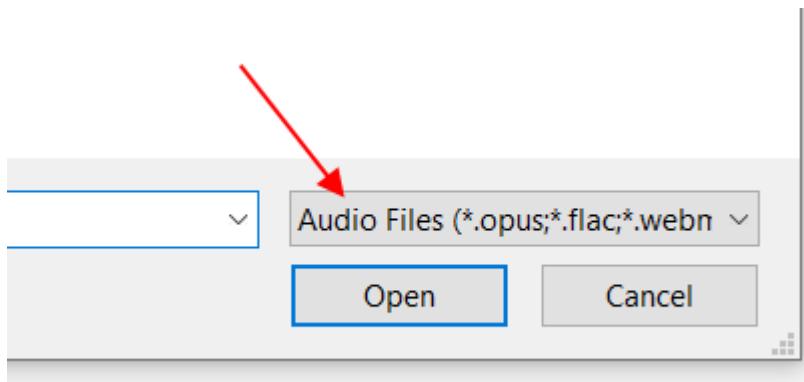
<body>

```
<input type="file" id="soundFile" accept="audio/*">
<input type="file" id="videoFile" accept="video/*">
<input type="file" id="imageFile" accept="image/*">
</body>
```

သတိပြုရန် အချက်မှာ accept attribute သည် file type များကို validate မလုပ်ဘူး ဆိုလိုချင်တာက user ထည့်လိုက်သည့် file သည် audio file ဖြစ်သလား video file ဖြစ်သလားဆိုတာကို စစ်ဆေးခြင်း မပြုလုပ်ပါဘူး။ သို့သော် user ကို မှန်ကန်တဲ့ file type ကို ရှေးနိုင်ဖို့ hints လေးတွေ လုပ်ပေးပါတယ်။ အထက်ပါ program ကို run လျှင် အောက်ပါ အတိုင်း ပေါ်လာပါမည်။ ထိုသို့ ပေါ်လာတဲ့ထဲကမှ ပထမဆုံး choose file ကို တစ်ချက် နှိပ်လိုက်ပါမည်။



အထက်ပါ အတိုင်း နှိပ်လိုက်သည့်နှင့် တစ်ပြီးနှင်း file ရွှေးရန် window form တစ်ခု ပေါ်လာပါမည်။ ထို window form ရဲ့ file types များကို ကြည့်ပါက audio file types များအား အလုံလျောက် ဖော်ပြန်သည်ကို တွေ့ရပါမည်။ ထို့ကဲ့သို့ ဖော်ပြ ရခြင်းမှာ မံမိတ္ထု accept နောက်တွင် ရေးပေးလိုက်သော properties များကြောင့် ဖြစ်သည်။



မည်သည့် file type ဖြစ်ကြင်းကို validate လုပ်နိုင်ဖို့ အတွက် server-side သိမဟုတ် javascript ကို သုံးပြီး စစ်ဆေးနိုင်ပါတယ်။

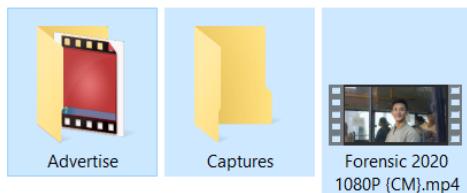
### Multiple Files

အထက်မှာ ဖော်ပြခဲ့တဲ့ input type သည် file တစ်ခုတည်းကိုသာ access စုစုပေါင်းကြော်မှုတဲ့ input လုပ်နိုင်ပါတယ် သို့သော် တစ်ကြိမ်တည်းမှာ file တွေအများကြီးကို input လုပ်ချင်တယ်ဆိုရင်တော့ multiple ကိုသုံးနိုင်ပါတယ်။ Multiple ကိုအသုံးပြုရန် input element ရဲ့ နောက်ဆုံးမှာ multiple ကို ထည့်ပေးရန်သာ ဖြစ်ပါတယ်။

### Sample program

```
<div>
    <label for="file">Choose file to upload</label>
    <input type="file" id="file" name="file" multiple>
</div>
<div>
    <button>Submit</button>
</div>
```

Output ကို အောက်မှာ ဖော်ပြထားပါတယ် File type မှာလည်း all File ဆိုပြီး ပေါ်နေတာကို မြင်ရမှာ ဖြစ်ပါတယ်။ Program ထဲမှာ button element ကိုသုံးပြီး submit button ကိုပါ ထည့်ထားပါတယ် Button တွေရဲ့ လုပ်ဆောင်ပုံကို JavaScript Lesson မှ အသေးစိတ် ထပ်မံ ဖော်ပြ ပေးသွားပါမည်။



### Limits Accept

Accept ကို အသုံးပြုခြင်းဖြင့် မိမိတို့ လက်ခံလိုတဲ့ File တွေကိုပဲသီးသန့် လက်ခံလိုကြောင်း limit လုပ်ထား နိုင်ပါသေးတယ် သို့သော် စစ်ဆေးတဲ့ အပိုင်းကိုတော့ server side programming တစ်ခုခု နဲ့ စစ်ဆေး ရမှာ ဖြစ်ပါတယ်။ ထိုသို့ Limit လုပ်နိုင်ရန် accept နောက်မှာ မိမိတို့ လက်ခံ လိုသော file type ရဲ့ extension များကို comma ခံပြီး ရေးသား နိုင်ပါတယ်။

```
<input type="file" id="pic" name="mypic" accept=".jpg, .jpeg, .png">
```

### ID (Sample Program )

Id ကို ပုံစံ အမျိုးမျိုးဖြင့် မိမိတို့ လိုအပ်သလို အသုံးပြု နိုင်ပါတယ်။ အောက်ပါ program မှာဆုံးလျှင် h1 element ထဲမှ စာသားများကို ပြင်ဖို့ အတွက် အသုံးပြု ထားပါတယ်။ h1 element ကိုသုံးပြီး WinHtut ဆိုတဲ့ စာသားတစ်ခုကို ရေးထားပါတယ် ထို စာသား အောက်မှာလည်း button တစ်ခု ထား ထားပါတယ် ထို button ကို တစ်ချက် နှုပ်လိုက် တာနဲ့ တစ်ပြိုင်နက် အပေါ်က WinHtut ဆိုတဲ့ စာသားကနောတစ်ခုင့် Id can use like this ဆုံးပြုး ပြောင်းသွားမှာ ဖြစ်ပါတယ်။ Sample program ကို အောက်မှာ ဖော်ပြထားပါတယ်။

```
<!DOCTYPE HTML>
<html>
<body>
<h1 id="myId">WinHtut</h1>
<button onclick="changeText()">Click Me!</button>
```

```

<script>
function changeText() {
    document.getElementById("myId").innerHTML = "Id can use like
this";
}
</script>
</body>
</html>
Output

```

## WinHtut



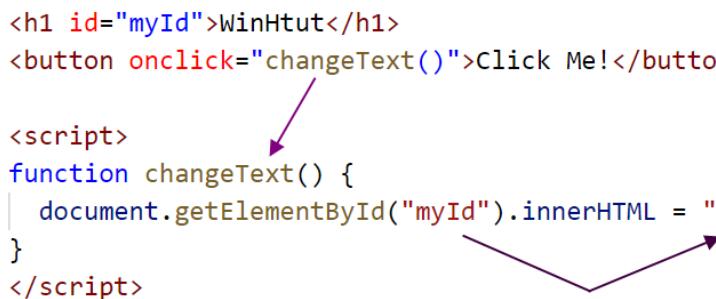
Button element ထဲမှာ onclick ဆိုတဲ့ attribute ကိုသုံးထားပါတယ် အကယ်၍ Click Me! ဆိုတဲ့ button ကို နှိပ်လိုက်လျှင် changeText() ဆိုတဲ့ function ကို သွားခေါ်ပေးပါတယ်။ changeText ဆိုတဲ့ function ကို script element ထဲမှာ ရေးထားပါတယ်။

```

<h1 id="myId">WinHtut</h1>
<button onclick="changeText()">Click Me!</button>

<script>
function changeText() {
    document.getElementById("myId").innerHTML = "Id can use like this";
}
</script>

```



Script ထဲမှာ function သည် ဘာလုပ်ဆောင်ပေးသလဲ။ Script ထဲမှာ changeText function သည် myID ဟုရေးထားသည့် id ရှိသည့် နေရာမှာ ရေးထားသော စာသားအစား မိမိတဲ့ အသစ် ဖော်ပြလိုသည့် စာသားကို ဖော်ပြ ပေးမှာ ဖြစ်ပါတယ်။ ဆုလိုချင်တာက စာသားကို ချိန်းဖို့ အတွက် id ကိုပဲ အသုံးချဖွားတာပါ။ ထိုကူးသို့ id ကို အသုံးချဖို့ ဆုလျှင်လည်း getElementById ဆိုသည့် function ကိုသာ အသုံးပြထားပါသည်။ ထို function ထဲမှာမှ parameter အနေဖြင့် "myId" ကို ပြန်ထည့်ပေးခြင်းဖြစ်ပါသည်။ ပုံမှိ နားလည်စေရန် အောက်တွင် sample program တစ်ပုဒ်ကို ထပ်မံ ရေးပြထားပါသည်။

```

<h1 id="myId">WinHtut</h1>
<button onclick="changeText()">Click Me!</button>
<h2 id="info">GreenHackers</h2> ←
<button onclick="changeInfo()">ChangeInfo</button>
<script>
function changeText() {
| document.getElementById("myId").innerHTML = "Id can use like this";
}
function changeInfo(){
| document.getElementById("info").innerHTML="Unity is Power";
}
</script>

```

နောက်ထပ် program တစ်ပုဒ်အနေဖြင့် Button and Paragraph element နှစ်ခုရေးထားပြီး button မှ တစ်ဆင့် paragraph မှ စာသားဖော်ပြချက်များကို ပြုပြင်ပါမည်။  
ပထမဆုံး အနေဖြင့် input button နှင့် paragraph တစ်ခုကို အောက်ပါ အတိုင်းရေးပါမည်။

```

<form>
| <input type="button" value="Start Motor" style="color: #crimson;">
</form>
| <p style="color: #blueviolet;">Motor is Stopped.</p>

```

သတိပြုရန် အချက်မှာ input ရဲ့ value ကို Start Motor ဟု ပေးခဲ့ပါသည်။  
ထိုသို့ရေးပြီးနောက် script element ကိုသုံးပြီး JavaScript code တိုကို စာရေးပါမည်။  
ပထမဆုံး အနေဖြင့် querySelector function ကိုသုံးပြီး input element ကိုလှမ်းယူပါမည်။  
ထိုသို့ ယူပြီးနောက် button ဆုံးတဲ့ variable ထဲသို့ ထည့်ထားပါမည် ထိုကြောင့် button သည်  
input element ကို ကိုင်ထားသော variable တစ်ခု ဖြစ်ပါသည် ထိုနည်းတူ p element အတွက်  
paragraph ဆုံးသည့် variable တစ်ခု တည်ဆောက်ထားပါသည်။

```

const button = document.querySelector('input');
const paragraph = document.querySelector('p');

```

နောက်တစ်ဆင့် အနေဖြင့် button အား click လုပ်လိုက်မည် ဆိုလျှင် buttonAction  
ဆုံးသည့် function အားလုမ်းခေါ်ပါမည်။ ထိုသို့ လုမ်းခေါ်နိုင်ရန် အတွက် addEventListener  
ဆုံးသည့် function ကိုသုံးရပါမည်။ ထို function ထဲတွင် parameter နှစ်ခု ပါ ပါသည် ပထမ

တစ်ခုသည် action ဖြစ်ပြီး ဒုတိယ တစ်ခုသည် function ဖြစ်ပါသည်။ Action ဆိတာကတော့ user မှ click လုပ်မည့် action ကို ပြောတာပါ။

buttonAction ကိုခေါ်မှာ ဖြစ်တဲ့ အတွက် buttonAction ထဲတွင်အလုပ်လုပ်မည့် program များကို ရေးသားရပါမည်။ အလုပ်လုပ်မည့် ပုံစံမှာ button ကို နိုပ်လိုက်သည့်နှင့် buttonAction ဆိုသည့် function ကိုလုမ်းခေါ်ပါမည် ထိုfunction ထဲတွင် button ထဲမှ value အား စစ်ဆေးပါမည် အကယ်၍ value သည် Start Motor ဖြစ်နေလျှင် သူ့အောက်မှ အလုပ်များကို ဆက်လုပ်ပါမည်။ အလုပ်လုပ်မည့် step နှစ်ခု ရှိပါသည် ပထမ တစ်ခုသည် button variable ထဲမှ value အား Stop Motor အဖြစ် ပြောင်းလဲမည် ဖြစ်ပြီး ဒုတိယ step အနေဖြင့် paragraph ထဲမှ စာသားအား Motor has started ဟုချိန်းပါမည် ထိုသို့ ချိန်းနိုင်ဖို့ textContent properteis ကိုသုံးရပါမည်။

```
button.addEventListener('click', buttonAction);
function buttonAction() {
    if (button.value === 'Start Motor') {
        button.value = 'Stop Motor';
        paragraph.textContent = 'Motor has started!';
    }
}
```

အထက်တွင် ရေးသားထားသော လုပ်ဆောင်ချက်များသည် button.value သည် Start Motor ဖြစ်နေလျှင် လုပ်ဆောင်မည့် လုပ်ဆောင်ချက်များဖြစ်သည် အကယ်၍ button.value သည် Start Motor မဟုတ်ခဲ့ပါက button.value ထဲသို့ Start Motor ဆိုတဲ့ value ကို ထည့်ပေးလိုက်မှာ ဖြစ်ပါတယ် ထိုနည်းတူ paragraph.textContent ကိုလည်း Motor is stopped ဟု value ထည့်ပေးလိုက်မှာ ဖြစ်ပါတယ်။

```
} else {
    button.value = 'Start Motor';
    paragraph.textContent = 'Motor is stopped.';
}
```

Program အပြည့်အစုံကို အောက်မှာ ဖော်ပြုပေးထားပါတယ်။

```

<body>
  <form>
    <input type="button" value="Start Motor" style="color: #crimson;">
  </form>
  <p style="color: #blueviolet;">Motor is Stopped.</p>
  <script>
    const button = document.querySelector('input');
    const paragraph = document.querySelector('p');

    button.addEventListener('click', buttonAction);
    function buttonAction() {
      if (button.value === 'Start Motor') {
        button.value = 'Stop Motor';
        paragraph.textContent = 'Motor has started!';
      } else {
        button.value = 'Start Motor';
        paragraph.textContent = 'Motor is stopped.';
      }
    }
  </script>
</body>

```

အထက်ပါ program အား run လိုက်ပါက အောက်ပါအတိုင်း button နှင့် paragraph တို့  
ပေါ်လာပါမည် button အား နှိပ်လိုက်ပါက button လည်း စာသားများ ပြောင်းလဲသွားမှာ  
ဖြစ်သလို paragraph မှ စာသာများလည်း ပြောင်းလဲ သွားမှာ ဖြစ်ပါတယ်။

Start Motor

Motor is Stopped.

**Next Photo**

Stop Motor

Motor has started!

**Input Type (Check Box)**

User တွေဟာ items သီးသန့် တစ်ခုချင်းစီကို ရွှေးချယ်တဲ့ အခါမှာ check box  
ကိုသုံးနိုင်ပါသေးတယ်။ Check box ကို အသုံးပြုနိုင်ဖို့ input element ထဲမှ type attribute  
ထဲတွင် checkbox ကိုရေးပေးခြင်းဖြင့် အသုံးပြုနိုင်ပါတယ်။ အကယ်၍ id များ name များ  
ထပ်ထည့် လိုပါကလည်း ထပ်ထည့်နိုင်သလို အလုံလျောက် checked လုပ်ထားလိုပါကလည်း

နောက်ဆုံးတွင်

checkedကိုထည့်ထားနိုင်ပါတယ်။

```
<p>What skill do you like in Dota?</p>

<div>
  <input type="checkbox" id="scales" name="GodStrength">
  <label for="scales">GodStrength</label>
</div>
```

### Output

What skill do you like in Dota?

GodStrength

နောက်ထပ်ရွေးချယ်စရာများ ထပ်ထည့်ပြီး အချို့သော ရွေးချယ်စရာများကို  
ကြိုးတင်ပြီး checked လုပ်ထားလိုပါက အောက်ပါ အတိုင်းရေးနိုင်ပါတယ်။

```
<p>What skill do you like in Dota?</p>
<div>
  <input type="checkbox" id="scales" name="GodStrength">
  <label for="scales">GodStrength</label>
</div>
<div>
  <input type="checkbox" id="Support" name="Support">
  <label for="skill">Support</label>
</div>
<div>
  <input type="checkbox" id="Carry" name="Carry" checked>
  <label for="Carry">Carry</label>
</div>
<div>
  <input type="checkbox" id="Safe" name="Safe" checked>
  <label for="Safe">SafeLane</label>
</div>
...
```

### Output

What skill do you like in Dota?

- GodStrength
- Support
- Carry
- SafeLane

output တွင် Carry and SafeLane နှစ်ခုအား

ကြိုတင် ရွေးချယ် checked လုပ်ထားသည်ကို တွေ့ရပါမည်။

#### Input Type(Color)

Color ကတေသူ user ကို color ရွေးချယ်ခွင့်ပေးတာပါ။ color တွေကို ကြိုင်တင်ပြီးတော့လည်း ဖော်ပြ ပေးထားလိုပါတယ် ထိုသို့ ဖော်ပြ နိုင်ရန် hexadecimal value တွေကိုသုံးရပါမည်။

```
<div>
    <input type="color" id="shirt" name="shirt" value="#e66465">
    <label for="shirt">Shirt</label>
</div>
<div>
    <input type="color" id="shoe" name="shoe" value="#f6b73c">
    <label for="shoe">Shoe</label>
</div>
```

#### Output



အထက်ပါ အတိုင်း user ကို color ရွေးချယ်ခွင့် ပေးမှာ ဖြစ်ပါတယ်။ Input element ရဲ့ color type မှာ value ကို 7-character string နဲ့ ဖော်ပြပါတယ်။ 7-character ဆိုတာ #000000 ပါ ရှုံးမှ # သည် hexadecimal ကို ဖော်ပြခြင်း ဖြစ်ပါတယ် ထို # နောက်က နေမှ rrggbb တို့ လိုက်ပါတယ်။ ဥပမာ #e66465 ယခု value မှာဆိုလျှင် e6 သည် r(red) တန်ဘိုး ဖြစ်ပြီး 64 သည် g(green) တန်ဘိုး 65 သည် b(blue) တန်ဘိုးဖြစ်ပါသည်။ color အတွက် hexadecimal value တွေကို ထည့်ပေးတဲ့ အခါမှာ uppercase or lowercase ကြိုက်သလို ထည့်လိုပါတယ် သို့သော် stored လုပ်တဲ့ အချိန်ကြရင်တော့ lower-case နဲ့ပဲ stored လုပ်ပါတယ်။ Stored

လုပ်လိုက်တဲ့ color value တွေကို သိချင်ရင်တော့ JavaScript ကိုသုံးပြီးထုတ်ကြည့်နိုင်ပါတယ်။

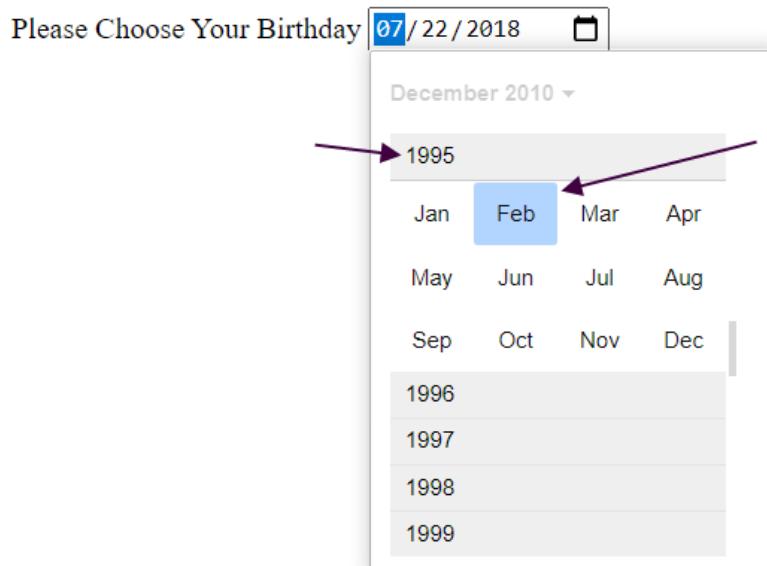
### Input Type(Date Time )

Date and Time ကို အလွယ်တကူ ရှုချယ်နိုင်ဖို့ အတွက် input type အနေနဲ့ date ကို အသုံး ပြုနိုင်ပါသေးတယ်။ Additional Attributes များအနေဖြင့် value , min , max တို့ကိုလည်း date နဲ့ အတူ တဲ့ သုံးနိုင်ပါတယ်။ value က တော့ default အနေဖြင့် မိမိတို့ သတ်မှတ်ပေးထားချင်တဲ့ date ကို ထားပေးလိုပါတယ်။ min ကတော့ အနည်းဆုံး သတ်မှတ်ထားချင်တဲ့ date ပါ ဥပမာ birthday ကို အနည်းဆုံး 1900 ခုနှစ်လောက်ကနေ ထားတာမျိုးပါ 1800 မှာ မွေးတဲ့ လူကတော့ အခုချိန်မှာ internet သုံးနေဖို့ မဖြစ်နိုင်တော့ပါဘူး ထိုကြောင့် min ကိုသုံးပြီး မိမိတို့ ထားချင်တဲ့ အနည်းဆုံး date ကို သတ်မှတ်ပေးထားနိုင်ပါတယ်။ max attribute ကတော့ အများဆုံး ရွှေးချယ်နိုင်တဲ့ date ပါ ဥပမာ user ကို 2030 ထံပဲ ရွှေးချယ်စေချင်တယ်ဆိုရင် မိမိတို့ လိုသလို အများဆုံး max ကို သတ်မှတ်ပေးထားလိုပါတယ်။

```

<label for="birth">Please Choose Your Birthday </label>
<input type="date" id="birth" name="birth"
       value="2018-07-22"
       min="1900-01-01" max="2010-12-31">
```

အထက်ပါ program မှာဆိုလျှင် default date ကို 2018-07-22 yy-mm-dd ဆိုပြီး သတ်မှတ်ပေးထားပါတယ်။ ရွှေးချယ်ရမဲ့ အနည်းဆုံး min date ကိုတော့ 1900 ခုနှစ်ကနေ စတင် သတ်မှတ် ပေးထားပြီး max date ကိုတော့ 2010 မှာ သတ်မှတ်ပေးထားပါတယ်။ ထိုကြောင့် အသုံးပြုသူ user သည် 1900 ခုနှစ်ကနေစတင်ပြီး 2010 ခုနှစ်ထံပဲ ပဲ ရွှေးချယ်နိုင်မှာ ဖြစ်ပါတယ်။



မြတ်စွာ အသုံးပြုသည့် browser ပေါ်မှုတည်ပြီး UI(User Interface) ပေါ်လာပုံ အနည်းငယ် ကွဲပြားနိုင်ပါတယ် စာရေးသူအနေဖြင့် chrome browser ကိုအသုံးပြုထားပါသည့် browser များ ကွဲပြားနိုင်သော်လည်း Feature များ ပျောက်နေတာမျိုး မပါတာမျိုး မဖြစ်နိုင်ပါဘူး input type date ကို browser အများစုံမှာ support လုပ်ပေးပါတယ်။ calendar ပုံ icon လေးကို နှိပ်ပြီး မြတ်စွာ ရွှေးချယ်လိုသည့် yy-mm-dd တို့ကို ရွှေးချယ် နိုင်ပါတယ်။

---

Please Choose Your Birthday

အထက်တွင် ဖော်ပြုခဲ့သော input type date သင်ခန်းစာကို သုံးပြီး program အသေးလေး စတင် ရေးကြည့် ကြပါစို့။ User ကနေပြီး မွေးနေ့ကို ရွှေးလိုက်မယ် ထို ရွှေးလိုက်တဲ့ မွေးနေ့ကို javascript နဲ့ပြန်ယူပြီး user ကို ပြန်လည် ဖော်ပြပေးပါမယ်။ အဲတော့ ပထမဆုံး အနေနဲ့ အချက် နှစ်ချက်ကို အရင်ဆုံး ပြုလုပ်ရပါမယ် ပထမ တစ်ချက်သည် user ထည့်လိုက်တဲ့ date ကို ယူရပါမည် ဒုတိယ တစ်ချက်သည် ထို date ကို ပြန်လည် ဖော်ပြပေးပါရပါမယ်။ ထို process နှစ်ခုကို လုပ်ဆောင်ဖို့ အတွက် function တစ်ခု တည်ဆောက်ပါမယ် ထို function အား button နှင့် ချိတ်ဆက်ထားပြီး button အား နှိပ်လိုက်သည်နှင့် တစ်ပြိုင်နှင့် function ကို အလုပ် စလုပ်ခိုင်းပါမည်။

```

<form>
<label for="birth">Please Choose Your Birthday </label>
<input type="date" id="birth" name="birth"
       value="1990-07-22"
       min="1900-01-01" max="2010-12-31">

<h3>Your Choosen Date:</h3>
<p>yy-mm-dd</p>
</form>
<button onclick="buttonAction()">GetDate</button>

```

ပထမဆုံး အနေဖြင့် အထက်ပါအတိုင်း design ပိုင်းနှင့် button တစ်ခုကို ရေးထားပါသည်။ အကယ်၍ button ကို အောက်ပါ javascript မှ အလုပ်များကို သွားရောက် လုပ်ဆောင်မှာ ဖြစ်ပါတယ်။

```

20   <script>
21     var button = document.querySelector('input[type="date"]');
22     var paragraph = document.querySelector('p');
23     button.addEventListener('click', buttonAction);
24     function buttonAction() {
25       paragraph.textContent = button.value;
26     }
27   </script>

```

Line21 သည် input element ထဲမှ date တွင်ရှိနေသော data ကို လုမ်းယူပြီး button ဆိုသည့် variable ထဲသို့ ထည့်ထားပါသည်။ line22 တွင် p element ကို paragraph ဆိုသည့် variable ဖြင့် လုမ်းထောက်ထားပါသည်။ line 23 သည် user မှ button ကို click နိုပ်လိုက်သော အခါတွင် buttonAction ဆိုသည့် function ကိုလုမ်းခေါ်မှာ ဖြစ်ပါတယ်။ line 25 buttonAction function ထဲတွင် button ထဲမှ value ကို paragraph ထဲမှ textContent ထဲသို့ ထည့်ပေးလိုက်ပါသည်။ သတိပြုရန် အချက်မှာ input element ထဲမှ value သည် default ပေးထားခြင်း ဖြစ်ပြီး user မှ အသစ်ရွေးလိုက်သော အခါတွင် ထို value သည်လည်း တန်ဘိုး အသစ် ပြောင်းလဲသွားမှာ ဖြစ်ပါတယ်။

The screenshot shows a web page with the following elements:

- A label "Please Choose Your Birthday" followed by a date input field containing "02/14/1995". An arrow points from the text "1995-02-14" to the input field.
- An h3 tag "Your Choosen Date:" followed by a paragraph tag containing "1995-02-14". An arrow points from the text "1995-02-14" to the paragraph.
- A button labeled "GetDate". An arrow points from the text "GetDate" to the button.

Program အပြည့်အစုံကို အောက်တွင် ဖော်ပြထားပါသည်။

```

<form>
  <label for="birth">Please Choose Your Birthday </label>
  <input type="date" id="birth" name="birth"
    value="1990-07-22" ←
    min="1900-01-01" max="2010-12-31">
  <h3>Your Chosen Date:</h3>
  <p>yy-mm-dd</p>
</form>
<button onclick="buttonAction()">GetDate</button>
<script>
  var button = document.querySelector('input[type="date"]');
  var paragraph = document.querySelector('p');
  button.addEventListener('click', buttonAction);
  function buttonAction() {
    paragraph.textContent = button.value;
  }
</script>

```

### Input Type(datetime-local)

ယခုလက်ရှိ မိမိတို့၏ local အချိန်နဲ့ နေ့ရက် သို့မဟုတ် meeting time သို့မဟုတ် အခြားသော သီးခြား သတ်မှတ်ထားသည့် အချိန်းများကို အတိအကျ ရယူလိုတဲ့ အခါမျိုးမှာ ဆုလျှင် input type ထဲမှ datetime-local ကိုအသုံးပြုနိုင်ပါတယ်။ အခြားသော attributes များအနေဖြင့် id , name , value , min , max စသည်တို့ကိုလည်း ထည့်သွင်းသုံးနိုင်ပါတယ်။ ထိုပြင် autocomplete , list , readonly and step စသည် attribute များကိုလည်း support လုပ်ပေးထားပါသေးတယ်။ အခြားသော attributes များကိုတော့ JavaScript အခန်းကြမှ သာလျှင် အသေးစိတ် ဆွေးနွေးသွားမှာ ဖြစ်ပါတယ်။

```

11  <input type="datetime-local" id="meeting-time"
12    name="meeting-time" value="2020-06-12T19:30"
13    min="2020-01-07T00:00" max="2020-12-31T00:00">
14    <p>yy-mm-dd</p>
15    <h3>About</h3>
16    <button onclick="buttonAction()">GetDate</button>

```

အထက်တွင် ဖော်ပြထားသော program ၏ အသုံးပြုထားသည့် attributes များသည် ယခင် သင်ခန်းစာများတွင်ဖော်ပြထားပြီး ဖြစ်သောကြောင့် အသေးစိတ်ထပ်မံ မဖော်ပြလိုတော့ပါ။ min and max value များကို မိမိတို့သတ်မှတ်လိုသည့် ခုနှစ်အတိုင်း သတ်မှတ်နိုင်ပါသည်။

```

● 17 <script>
18   var button = document.querySelector('input[type="datetime-local"]');
19   var paragraph = document.querySelector('p');
20   var about = document.querySelector('input[name="meeting-time"]')
21   var h3 = document.querySelector('h3')
22   button.addEventListener('click', buttonAction);
23   function buttonAction() {
24     paragraph.textContent = button.value;
25     about.name = "ForShowTime";
26     h3.textContent = about.name;
27   }
28 </script>

```

ယခု သင်ခန်းစာမျာတော့ input element ထဲမှ type နှင့် name နှစ်ခုလုံးကို ယူမှာဖြစ်သည့် အတွက် querySelector နှစ်ခု ထိုပြင် p tag နှင့် h3 tag တို့တွင် ရေးထားသောtextContent တို့ကိုပါ ပြင်မှာ ဖြစ်သည့်အတွက် စုစုပေါင်း querySelector 4 ခုရှိမှာဖြစ်ပါတယ်။ အထက်ပါ ပုံတွင် line 18,19,20,21 တို့တွင် ကြည့်ပါ။ input element ထဲမှ nameရဲ့ value သည် “meeting-time” ဖြစ်ပါသည်။ ထို meeting-time အား ForShowTime ဟုပြင်ပါမည် ပြင်ပီးသည့် value ကိုလည်း user မြင်အောင် ပြန်ပြ ပေးပါမည်။ line 20 မှာ name attribute အား about ဆိုသည့် variable ဖြင့် ကိုင်ထားပါသည်။ ထို့ကြောင့် line 25 တွင်about ထဲမှ name အား ForShowTime ဆိုသည့် value အသစ်ကို ထည့်ပေးလိုက်ပါသည်။ value အသစ်အား h3 tag ထဲတွင် ပြန်လည် ဖော်ပြလိုသည့် အတွက် line 26 တွင် h3.textContent ထဲသို့ ထည့်ပေးလိုက်ပါသည်။ ထို့ကြောင့် program run သော အခါတွင်အောက်ပါ အတိုင်း user ရွှေးချယ်ပေးလိုက်သည့် local time date အတိအကျတို့ကို ဖော်ပြုမှာ ဖြစ်သလို စာရေးသူတို့ ပြောင်းလဲ ပေးလိုက်သည့် ForShowTime ကိုလည်း ပြမှာ ဖြစ်ပါတယ်။

Choose a time for your appointment:

yy-mm-dd

## About

Before Button Click

Choose a time for your appointment:

2020-06-12T19:30

ForShowTime

After Button Click

### Input Type( email )

User တွေဆီကနေ email address တောင်းလိုတဲ့ အခါမှာ input type ထဲက email ကိုသုံးပါတယ်။ User ဆီက email တောင်းဖို့ အတွက် သာမန် textbox ကိုလည်း အသုံးပြုလို ရပါတယ် သို့သော် မတူပါဘူး။ email ကိုသုံးခြင်းအားဖြင့် validate လုပ်နိုင်တဲ့ attributes တွေပါ ဆက်ပြီး အသုံးပြုနိုင်ပါတယ်။ ဥပမာ pattern ပါ။

```
<label for="email">Enter your gmail.com email:</label>
<input type="email" id="email"
       pattern=".+@gmail.com" size="30" required>
```

Pattern Attribute ကတော့ user က ထည့်လိုက်တဲ့ text တွေကို စစ်ပေးဖို့ အတွက်သုံးတာပါ ထိုသို့ စစ်ခြင်းကို validate လုပ်ခြင်းလို့ ခေါ်ပါတယ်။ ဥပမာ user က ဘာမှ မထည့်ပဲ အလွတ်ထားလိုက်သလား သို့မဟုတ် format မကျသည့် ပုံစံများဖြင့် Input ကို ထည့်လိုက် သလားပေါ့။ User input လုပ်လိုက်သည့် data များကို စစ်ဆေးဖို့ အတွက်ဆိုလျှင် JavaScript ဖြင့် Regex ကိုသုံးကြပါတယ်။ Regex ဆိုတာ regular expression ကို ဆိုလိုတာပါ။ Regex ကိုသုံးပြီး username , email address , phone number စတဲ့ data များကိုလည်း စစ်ဆေးနိုင်ပါတယ်။ Regex ဖြင့် စစ်ဆေးပုံများကို JavaScript သင်ခန်းစာများတွင် ဆက်လက် ဖော်ပြု သွားပါမယ်။ နောက်ဆုံးမှာ ရေးထားတဲ့ required attribute ကတော့ boolean attribute ဖြစ်ပါတယ် boolean ဆိုတာ data type အမျိုးအစား တစ်ခုပါ ထို data type များအကြောင်းကို js သင်ခန်းစာများတွင် အသေးစိတ် သင်ကြားပေးသွားမှာ ဖြစ်ပါတယ်။

### Required Attribute

ကို user က data ဖြည့်ကို ဖြည့်ရမည့် နေရာများတွင် သုံးပါတယ် input type email အပြင် အခြားသော text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file စသည် တို့နှင့်လည်း တွဲဖက် အသုံးပြု နိုင်ပါတယ်။ required attribute ကို အောက်မှာ from sample program နှင့် ဖော်ပြု ပေးထားပါတယ်။

```
9   <form name="form" method="post">
10    <pre>           ↑          ↑
11    <label> Email Address </label><input type="text" name="name" required>
12    <input type="submit" value="Submit">
13    </pre>           ↑          ↑
14    </form>
15  <script>           ↓          ↓          ↓
16  if( document.forms['form']['name'].required = true)
17  label.innerHTML += " (required)" ←
18 </script>
```

Line number 9 မှ 14 ထိုသည် form တစ်ခုကို ရေးသားခြင်း ဖြစ်ပါတယ်။ Form တစ်ခု

ရေးပြီး javascript သိမဟုတ် အခြားသော backend programming များနှင့် အလုပ် တွဲလုပ် လုသော အခါတွင် ထိ form ရဲ့ name နှင့် ထိ form တွင် မည်သည့် method တိုကို သုံးမည် ဆုံးတာကိုပါ ထည့်ပေးရပါတယ်။ line 9 မှာတော့ form ရဲ့ name ကို form ဟုသာ ပေးထားခဲ့ပြီး form ရဲ့ method ကိုတော့ post ဟု ပေးထားခဲ့ပါတယ်။ post method အကြောင်းကို javascript သင်ခန်းစာများ အသေးစိတ် ရှင်းပြ သွားပါမယ်။ ယခု အချိန်မှာတော့ post method ဟာ data တွေ ပို့ချင်တဲ့ အခါမှာ သုံးတယ်လို့ အလွယ်တကူ မှတ်ထားနိုင်ပါတယ်။ line 11 ရဲ့ နောက်ဆုံးမှာတော့ required ကို ထည့်ထားပါတယ်။ ထိ required သည် user က data ထည့်သလား မထည့်သလား ဆုံးတာကို စစ်ဆေးရန် ဖြစ်ပါတယ်။ line 12 မှာ ပုံမှန်တိုင်း button ကို မသုံးထားပဲ submit ကိုသုံးထားပါတယ် form တွေကနေ data တွေ သွင်းချင်တဲ့ အခါမှာ submit ကိုသုံးပါတယ်။ submit အကြောင်းကိုလည်း ယခု html lesson မှာ ထပ်ပြီး ဖော်ပြသွားဦးမှာပါ။

Line 16 မှာတော့ Form မှ အချက်လက်များကို စစ်ဆေးဖို့အတွက် document.forms ကိုသုံးပါတယ်။ ထိုနောက်တွင် form ရဲ့ name နှင့် မိမိတို့ စစ်ချင်သော name ကို ထည့်ပေးရပါတယ် နောက်ဆုံးမှာ မှ .required ကိုရေးပြီး စစ်ဆေးပါတယ်။ အကယ်၍ form ထဲမှ name attribute ထဲတွင် data ရှိနေခဲ့လျှင်တော့ user မှ data ဖြည့်လိုက်သည်ဟု မှတ်ယူပြီး အခြားသော အလုပ်များကို ဆက်လုပ်မှာ ဖြစ်ပါတယ်။ သိမဟုတ် form ထဲမှ name ထဲတွင် data မရှိခဲ့ဘူး ဆုံးလျှင်တော့ အောက်ပါ အတိုင်း please fill out this field ဆုံးသည့် စာသားလေးကို ဖော်ပြ ပေးမှာ ဖြစ်ပါတယ်။

The screenshot shows a simple web form with an input field labeled "Email Address". Below the input field is a "Submit" button. A tooltip-like message box appears over the button, containing an exclamation mark icon and the text "Please fill out this field.".

#### Input Type ( hidden )

Type hidden ကို မရှင်းခင်မှာ သူရဲ့ အသုံးပြုပုံလေးကို အရင်ဆုံး program နဲ့ ဖော်ပြ ပေးပါမယ်။

```
<body>
  <input id="secId" name="secId" type="hidden" value="vij123">
</body>
```

အထက်ပါ program ကိုရေးပြီး run ကြည့်မည်ဆုံးလျှင် output မှာ ဘာမှ မပေါ်ပဲ Plain ကြီးသာ မြင်ရမှာ ဖြစ်ပါတယ်။ ထိုကဲ့သို့ ဘာမှ မပေါ်တဲ့ input type တစ်ခုကို ဘာကြောင့် ထွင်ထားကြသလဲပေါ့။ developer တွေရော သုံးကြ သလားဆုံးတော့ သုံးကြပါတယ် ဘယ်လို နေရာတွေမှာ သုံးလဲ ဆုံးရင် security issue နှင့် ပတ်သက်သည့် နေရာတွေမှာ သုံးပါတယ်။

hidden type ကိုသုံးခြင်း အားဖြင့် user တွေဟာ မမြင်နိုင်သလို modified လည်း လုပ်လို မရပါဘူး။ ထိုပြင် web page မှာလုံးဝ ပျောက် နေမှာ ဖြစ်သလို ဘယ်လို နည်းနဲ့မှ မြင်ရအောင် လုပ်နိုင်မှာ မဟုတ်ပါဘူး။

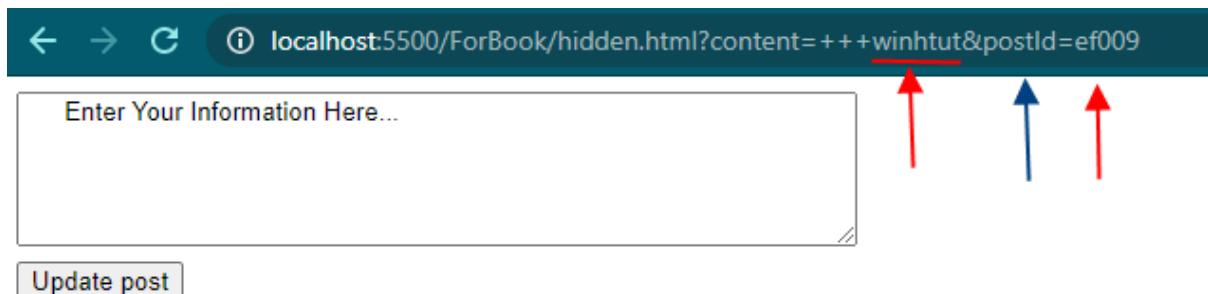
Hidden Input ကို security tokens or secrets စတဲ့ website security နဲ့ ပတ်သက်လိုလည်း သုံးပါတယ်။ ဥပမာ ငွေကြေးဆိုင်ရာ လွှဲပြောင်းမှု form တစ်ခုမှာ user မှ malicious form တစ်ခုကို server ဘက်သို့ တင်ပေးလိုက်မယ် ဆိုရင် server ကလည်း လက်ခံ မှာပါ ထိုသို့ လက်ခံ မယ်ဆိုရင် attack လုပ်ခံ လိုက်ရတာပါပဲ ထို attacking ကို Cross Site Request Forgery(CSRF) attack လိုခေါ်ပါတယ်။ ထိုကဲ့သို့သော ပြဿနာတွေကို ဖျဖော်ပြန်ဖို့ form ကို submit လုပ်တဲ့ အချိန်မှာ Hidden ကိုသုံးပြီး tokens or secrets key တွေထည့်ပေးလိုက်ခြင်းဖြင့် အဆင်ပြနိုင်ပါတယ်။

Value attribute မှာတော့ DOMString တစ်ခုပါဝင်ပါတယ်။ DOMString ထဲမှာမှ hidden data တွေ ပါဝင်ပါတယ် အဲဒီ hidden data တွေဟာ developer တွေကနေ user မသိအောင် server ဘက်ကို form submitted လုပ်တဲ့ အချိန်မှာ ပိုပေးလိုက်တဲ့ data တွေ ဖြစ်ပါတယ်။ DOMString ကတော့ UTF-16 string ဖြစ်ပါတယ်။ UTF -16 ဆိုတာ(UCS Transformation Format - 16 bit ) ပါ string type ဖြင့် 16 bit ထို stored လုပ်နိုင်တာကို ပြောတာပါ။ JavaScript သင်ခန်းစာမှာ stored လုပ်တဲ့ အကြောင်းကို အသေးစိတ် ထပ်မံ ရှင်းပြု ပေးပါမယ်။ hidden အကြောင်းကို ပိုမို နားလည် နိုင်ဖို့ အတွက် အောက်ပါမှာ sample program တစ်ပုဒ်ရေးပြထားပါတယ်။

```

9   <form>
10    |   <textarea id="content" name="content" cols="60" rows="5">
11    |   Enter Your Information Here...
12    |   </textarea>
13    |   </div>
14    |   <div>
15    |   |   <button type="submit">Update post</button>
16    |   </div>
17    |   <input type="hidden" id="postId" name="postId" value="ef009">
18  </form>
```

အထက်ပါ program မှာ textarea element ကိုသုံးထားပါတယ် textarea element သည် စာသားများ ပိုပေးချင်တဲ့ အခါမျိုးမှာ သုံးပါတယ် ထိုပြင် ထို textarea ရဲ့ name , cols , rows စတဲ့ attributes များကိုပါ ထည့်ပေးနိုင်ပါတယ်။ line 17 မှာတော့ input type hidden ကိုသုံးထားပြီး value ကို ef009 ဟု ပေးထားပါတယ်။ program ကို run ပြီး submit button ကို နှိပ်လိုက်တာနဲ့ တစ်ပြိုင်နှင်း user ထည့်ထားသော စာသားများနှင့် postId value တို့ကို server ဘက်သို့ တစ်ခါတည်း ပိုပေးမှာ ဖြစ်ပါတယ်။ အောက်က ပုံတဲ့မှာ အသေးစိတ် ဖော်ပြထားပါတယ်။



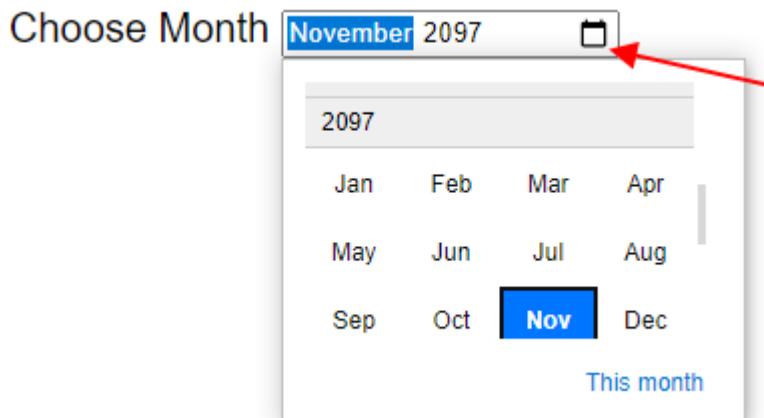
Server ဘက်သို့ ပိုပေးလိုက်သော data များကို browser URL တွင် ကြည့်နိုင်ပါတယ်။ စာရေးသူ အနေဖြင့် data ကို winhtut လို့ ထည့်ပေးလိုက်တာပါ။ value ကိုတော့ ef009 ဟု ထည့်ပေးလိုက်တာပါ။ ထို value အချက်လက်များကို server ဘက်က စစ်ဆေးပြီး မှန်ကန်မှ winhtut ဆိုတဲ့ data ကို database ထဲသို့ ထည့်သွင်းတာမျိုး သို့မဟုတ် ပြုပြင်တာမျိုး လက်ခံပေးမှာ ဖြစ်ပါတယ်။

#### Input Type( month )

Input type ထဲမှ month ကတော့ user ကို နှစ် နှင့် လ များ ရွှေးချယ်ခွဲင့်ပေးနိုင်ပါတယ်။ ရွှေးချယ်လိုက်တဲ့ value တွေကို stored လုပ်တဲ့ တဲ့အခါမှာ string type နဲ့ store လုပ်ပါတယ် format ကတော့ "YYYY-MM" ပုံစံနဲ့ပါ။ YYYY ကတော့ four-digit ပုံစံနဲ့နှစ်ကိုပြတာ ဖြစ်ပြီး MM ကတော့ လ number ကိုပြတာ ဖြစ်ပါတယ်။

```
<label for="start">Choose Month</label>
<input type="month" id="start" name="start"
       min="2020-10" value="2100-05">
```

အထက်ပါ program မှာ အနည်းဆုံး min ရွှေးချယ်နှင့် value ကိုတော့ 2020 10 လပိုင်းကို ရွှေးချယ် ပေးထားပြီး default value ကို 2100 5 လပိုင်းကို ပေးထားပါတယ်။ Output တွင် calendar icon ကို ရွှေးပြီး မိမိတို့ စိတ်ကြိုက် min မှ လွှဲပြီးရွှေးချယ် နှင့်ပေးနိုင်ပါတယ်။ browser များ အပေါ်တွင် မှတ်ညွှန်ပြီး ရွှေးချယ်ပုံများမှာ ကွဲပြား နိုင်ပါတယ်။ အထက်ပါ program မှာတော့ minimum date ကိုပဲ ထည့်ပေးထားပါတယ် အကယ်၍ စာဖတ်သူများ အနေဖြင့် maximum date ထည့်လိုပါကလည်း max="2020-12" စသည်ဖြင့် ရေးပြီး ထည့်နိုင်ပါတယ်။



### Input Type( number )

Input element ရဲ့ type number ကတေသာ user ကို number တွေ ထည့်ခိုင်းဖို့ အသုံးပြု နိုင်ပါတယ်။ number type မှာ တစ်ခု ထူးခြားတာက built-in validation လုပ်တာ ပါဝင်ပြီးသား ဖြစ်ပါတယ် ဆုံးလိုတာက number မဟုတ်တဲ့ အခြားသော character တွေ special character တွေ ထည့်မယ် ဆုံးရင် လက်ခံမှာ မဟုတ်ပါဘူး ထိုပြင် min and max ဆုံးတဲ့ step attribute တွေပါတယ့်ပြီး မိမိထည့်စေလိုသည့် number များကို control လုပ်ထားနိုင်ပါသေးတယ်။

```
<label for="number">Enter Your Lucky Number(1-1000):</label>
<input type="number" id="number" name="luckNumber"
       min="1" max="1000">
```

### Output

Enter Your Lucky Number(1-1000):

အကယ်၍ default value ပေးထားစေချင်တယ် ဆုံးရင်လည်း value attribute ကိုသုံးပြီးတော့ အောက်ပါ အတိုင်း ရေးထားနိုင်ပါသေးတယ်။ output ကို ကြည့်ပါက မိမိတို့ value မှာ ထည့်ပေးလိုက်သည့် အတိုင်း default အနေဖြင့် ပေါ်နေသည်ကို မြင်ရမှာပါ။

```
<label for="number">Enter Your Lucky Number(1-1000):</label>
<input type="number" id="number" name="luckNumber"
       min="1" max="1000" value="155">
```

### Input Type ( password )

Password input type ကတေသာ User ကို password တွေထည့်တဲ့ အခါမှာ လုံးခြုံအောင် ထည့်နိုင်ဖို့ လုပ်ဆောင် ပေးပါတယ်။ သာမန် text box ဆုံးလျှင် user ရိုက်ထည့်လိုက်တဲ့ number တွေကို ဘယ် number တွေ ရိုက်ထည့် နေတယ်ဆုံးတာ ပေါ်နေမှာပါ သို့သော်

password input type ကို သုံးမယ်ဆိုလျှင်တော့ asterisk ("\*") or dot (".") ပုံစံများဖြင့် ဖော်ပြုပေးမှာ ဖြစ်ပါတယ်။

```

10   <label for="username">Username:</label>
11   <input type="text" id="username" placeholder="username" name="username">
12 </div>
13 <div>
14   <label for="pass">Password (8-minimum):</label>
15   <input type="password" id="pass" name="password"
16   minlength="8" placeholder="password" ←
17   autocomplete="current-password" required>
18 </div>
```

Username:   
 Password (8-minimum):

အထက်ပါ program မှာဆိုလျှင် အသစ်အနေဖြင့် placeholder ဆိုတဲ့ attribute ကို ထည့်သုံးလာပါတယ် placeholder က ဘာလုပ်ပေးလဲဆိုတော့ user ကို username or password ရိုက်ထည့်ရမယ့် ပုံစံကို စာသားလေးတွေနဲ့ ဖော်ပြုပေးပါတယ် အထက်ပါ program မှာဆိုလျှင် line 11 and line 16 မှာ သုံးထားပါတယ်။ ထိုပြင် minlength နှင့် autocomplete စတဲ့ attribute များကိုပါ ထည့်သုံးပေးထားပါတယ်။ minlength က ကတော့ password ပေးတဲ့အချင်းမှာ အနည်းဆုံး ထည့်ရမယ့် number အရေအတွက် အတွက်ဖြစ်ပါတယ်။ autocomplete ကတော့ current-password လို့ရေးထားတဲ့ အတွက် ယခု လက်ရှိ password ကို auto ထည့်ပေးဖို့ သုံးနိုင်ပါတယ်။ Pattern attribute ကိုလည်း ထည့်နိုင်ပါတယ် သို့သော် regex အကြောင်းကို သိထားမှ အဆင်ပြုမှာ ဖြစ်ပါတယ်။

#### Input Type ( radio )



Radio input type ကို တစ်ခု သို့မဟုတ် တစ်ခု ရွှေးချယ်နိုင်ဖို့ အတွက် radio groups နဲ့ သုံးပါတယ်။ ဆိုလိုချင်တာကတော့ ရွှေးချယ်စရာ တွေ အများကြီးထဲမှာမှ တစ်ကြိမ်တည်းမှာ တစ်ခုပဲ ရွှေးချယ်နိုင်ပါတယ် ထိုကဲ့သို့ ဖြစ်အောင်လည်း group နဲ့

ရေးပေးရပါတယ်။ group ဆိုတာက သူရဲ့ name attribute ကို အတူတူ ရေးပေးရတာပါ။ အောက်ပါ program မှာ ဆိုလျှင် line 12 and line 17 မှာ ပြထားပါတယ်။ ထိုသို့ name မတူဘူးဆိုလျှင် တစ်ချိန်တည်းမှာပဲ အေးလုံးကို အတူတူ ရွေးချယ်ခွဲ့ရနုံမှာ ဖြစ်ပါတယ်။

```

12   <input type="radio" id="destktop" name="it" value="destktop"
13   | | | | checked> ↗
14   <label for="destktop">ComputerDesktop</label>
15   </div>
16   <div> ↗
17   <input type="radio" id="laptop" name="it" value="laptop">
18   <label for="laptop">Laptop</label>
19   </div>
```

## Select

- ComputerDesktop
- Laptop

### Input Type ( range )

Range input type က user ကို ကိုယ်လိုချင်တဲ့ value အား bar နဲ့ ဆွဲပြီး ရွေးချယ်နှင့်ဖွဲ့စေရန် ပေးပါတယ်။ range input type မှာ min and max စတဲ့ value တွေ သတ်မှတ်ပေးထားလိုရသလို default value ကိုလည်း သတ်မှတ်ပေးထားလိုရပါတယ် ထိုပြင် step attribute ကိုသုံးပြီး တစ်ခါ နှိပ်ရင် value ဘယ်လောက် တိုးသွားမလဲ ဆိုတာကိုပါ သတ်မှတ်ပေးထားလိုရပါတယ် အောက်ပါ program မှာ ဆိုလျှင် step="10" လိုရေးထားတာ ဖြစ်တဲ့ အတွက် တစ်ခါ နှိပ်တိုင်းမှာ value 10 တိုးသလို လျှော့မည်ဆိုလျှင်လည်း 10 လျှော့မှာ ဖြစ်ပါတယ်။

```

<p>Audio</p>
<div>
  <input type="range" id="audio" name="audio"
  | | | | min="0" max="100" value="90" step="10">
  <label for="cowbell">Sound</label>
</div>
..
```

Audio



Step ကို 10 မဟုတ်ပဲ any လို့ ပေးခဲ့မယ်ဆိုလည်း ရပါတယ် any ဆိုလျှင်တော့ click ပါပြီး မိမိရွေးချယ်တဲ့ ပုံစံ အတိုင်း ရွေ့သွားနိုင်ပါတယ်။

### Input Type ( reset )

Reset type ကိုသုံးတဲ့ အခါမှာ browser ကနေပြီး button ပုံစံဖြင့် render လုပ်ပေးပါတယ်။ reset ကို form များဖြင့် တွဲသုံးပြီး အကယ်၍ reset button ကိုနိပ်လိုက်လျှင် form ထဲတွင် ထည့်ထားသည့် value or data များကို ပျက်ပြစ်ပြီး initial values များကို ပြန်လည်ဖော်ပြ ပေးပါတယ်။ များသော အားဖြင့်တော့ reset button ကိုမသုံးကြပါဘူး ဘာလုံးမသုံးကြတာလဲ ဆိုတော့ user တွေဟာ submit button နဲ့ reset button ကို မှားပြီးနိုပ်တတ်ကြလိုပါ။ Program ရေးသားပုံကို အောက်တွင် ဖော်ပြထားပါတယ်။

```

<form>
  <div>
    <label for="name">Name:</label>
    <input type="text" id="name" placeholder="name">
  </div><div>
    <label for="class">Class:</label>
    <input type="text" id="class" placeholder="class">
  </div><div>
    <label for="town">Town :</label>
    <input type="text" id="town" placeholder="town">
  </div><div>
    <input type="reset" value="Reset">
    <input type="submit" value="Submit">
  </div>
</form>
  
```

အောက်ပါ ပုံတမ်း အတိုင်း data များဖြည့်ပြီးသည့် အခါမျိုးတွင် reset button ကိုနိပ်လိုက်သည်နှင့် တစ်ပြိုင်နက် ဖြည့်ထားသည့် data များအားလုံး ပျက်သွားမှာဖြစ်ပါတယ်။ ထို့ကြောင့် user များ မှားပြီး နိုပ်တာမျိုး ဖြစ်နိုင်သည့် အတွက်သိပ်မသုံးကြပါဘူး။

Name:	<input type="text" value="Win Htut"/>
Class:	<input type="text" value="ComputerScience"/>
Town :	<input type="text" value="PyinOoLwin"/>
<input type="button" value="Reset"/>	<input type="button" value="Submit"/>

Reset type သည် type and value attributes နှစ်ခုကိုပဲ support လုပ်ပြီး အခြားသော attributes များကို support မလုပ်ပါ။ ဘယ်အချိန်မှ အလုပ် လုပ် ပေးသလဲ ဆိုတော့ click ခေါက်လိုက်တဲ့ အချိန်မှာပါ။ value ကတော့ အခြားသော button attribute များ အတိုင်းပင်

DOMString ဖြင့် stored လုပ်ပြီး button မှာပေါ်ပေးမယ့် စာသား label ဖြစ်ပါတယ်။ သို့သော် user က value attribute ကို မထည့်ပေးပဲ type တစ်ခုတည်းကိုပဲ ရေးပေးပါက reset ဆိုသည့် စာသားကို သာ button ရဲ့ label အဖြစ် ဖော်ပြု ပေးပါတယ်။

### Input Type(Search)

Input type ထဲမှ search သည်လည်း text fields အမျိုးအစားထဲကပဲ ဖြစ်ပါတယ် search ကတော့ အချက်လက်တွေကို ရှာဖွေဖို့ သုံးပါတယ် ထိုပြင် မံမိရှာချင်တဲ့ အချက်လက်တွေကို ရေးပြီး နောက်မှာ cross sign လေး တစ်ခုပါဝင်ပါ ထို cross icon လေးကိုနှိပ်လိုက်တဲ့ အချိန်မှာ မံမိ ရေးထားသော အချက်လက် များကိုယ်ပြစ်မှ ဖြစ်ပါတယ်။

Search the site:

```
<label for="serach">Search Everythings</label>
<input type="search" id="serach" name="q">
<button>Search</button>
```

Change and input စုတဲ့ events တွေနဲ့ သုံး နိုင်ပါတယ်။ Change and input ကိုတော့ နောက်သင်ခန်းစာမှာ ရှင်းပြ သွားပါမယ်။ Common Attributes တွေ အနေနဲ့ autocomplete , list , maxlength , minlength , pattern , placeholder , required , size စသည်တို့ နဲ့ တွဲသုံးနိုင်ပါတယ်။ Methods တွေ အနေနဲ့ကတော့ select() , setRangeText() , setSelectionRange() စုတဲ့ methods တွေနဲ့ သုံးပါတယ် JavaScript သင်ခန်းစာ မှာ ယခု methods များအကြောင်းကို ဆေးနွေးပေးသွားပါမည်။

### Input Type( tel )

Input type မှ tel သည် user မှ telephone number များကို ထည့်သွင်းရန် အသုံးပြုပါ သည်။ user ထည့်ပေးလိုက်တဲ့ input value ကို automatically validated လုပ်ပေးရန် ခက်ခဲ ပါတယ် ဘာကြောင့်လည်း ဆိုတော့ ကဗ္ဗာတစ်ဝန်းမှာ telephone number အမျိုးမျိုး ရှိတဲ့ အတွက် ဖြစ်ပါတယ်။ အကော်ရှု browser က tel input type ကို support မလုပ်ဘူး ဆုံးလျှင်တော့ text input ကို အသုံးပြု သွားမှာ ဖြစ်ပါတယ်။ မံမိတို့သီးခြား အသုံးပြုလိုသည့် telephone number များကို maxlength , minlength and pattern စသည်တို့ကို သုံးပြီး validate လုပ်ထားနိုင်သလို placeholder attribute ကိုသုံးပြီးတော့လည်း phonenumbers format ကို ဖော်ပြု ထားနိုင်ပါတယ်။

```

<label for="tel">Telephone Number:</label>
<input type="tel" id="tel" name="phone" placeholder="+959" required>
<div>
<h6>Format: 09-453741976</h6>
</div>

```

Telephone Number:

Format: 09-453741976

#### Input Type(url)

Input type ထဲမှ Url သည် user မှ ရှာလိုသော address များအတွက် အသုံးပြုပါသည်။ Url input သည် user မှ ထည့်ပေးလိုက်သော input value သည် URL format နှင့်ကိုက်ညီခြင်းနှင့် user မှ input values များ ထည့်ခြင်း မထည့်ခြင်းတိုကို validate လုပ်ပေးပါသည်။ထိုပြင် url type ထဲ၌ pattern , placeholder အပြင် အခြားသော attributes များကိုလည်း ထည့်နိုင်ပါသည်။ Browser မှ url input type ကို support မလုပ်ဘူးဆိုလျှင် text input ပုံစံဖြင့် ငြောင်းလဲ အသုံးပြု သွားမှာ ဖြစ်ပါတယ်။

```

<label for="url">Enter an https:// URL:</label>
<input type="url" name="url" id="url"
       placeholder="https://winhtutonline.com"
       pattern="https://.*" spellcheck="true"
       required>

```

Enter an https:// URL:

List attribute ကိုလည်း support လုပ်ပေးတဲ့ အတွက် datalist element ကိုသုံးပြုး မိမိတို့ဘက်မှ suggest or offer လုပ်ပေးလိုသည့် site address or data list များကို ဖော်ပြပေးထား နိုင်ပါတယ်။ ထိုသို့ ဖော်ပြ ပေးနိုင်ရန် အတွက် datalist element ထဲတွင် option element ကိုအသုံးပြုရပါမည်။ Option element ထဲမှာမှ value ကို မိမိတို့ ပေးလိုသည့် site address or data list များ ထည့်ပေးထားနိုင်သည် sample program ကို အောက်တွင်ဖော်ပြထားပါသည်။

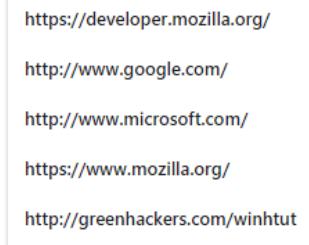
```

10  <input id="myURL" name="myURL" type="url"
11    |   list="urls" placeholder="https://winhtutonline.com/"
12  ↘   |   pattern="https://.*" spellcheck="true">
13  <datalist id="urls">
14    <option value="https://developer.mozilla.org/">
15  →  <option value="http://www.google.com/">
16    <option value="http://www.microsoft.com/">
17    <option value="https://www.mozilla.org/">
18  →  <option value="http://greenhackers.com/winhtut">
19  ↗</datalist>
~~

```

အထက်ပါ program တွင် line 13 မှာ အသုံးပြုထားသော datalist element ထဲမှ id attribute ရဲ့ value ဖြစ်သည့် urls အား line 11 မှ list attribute ဖြင့် ပြန်လည် အသုံးပြုထားပါသည်။ထိုကဲ့သို့ အသုံးပြုခြင်းဖြင့် datalist element ထဲမှ site or data အားလုံးကို input ထဲမှ access လုပ်လို ရသွားသည့်အတွက် မိမိတို့ ထည့်ပေးထားသည့် address အားလုံးကို အောက်ပါပုံအတိုင်းရှုံးချယ် နိုင်မှာ ဖြစ်ပါတယ်။

Enter an https:// URL or Choose :



## Lists & Tables

ယခု သင်ခန်းစာမျာတော့ html ရဲ့ list and table အကြောင်းကို ဆွေးနွေးသွားမှာ ဖြစ်ပါတယ်။ data တွေကို list အနေနဲ့ ဖော်ပြ လိုတဲ့ အခါမျိုးမှာ li tag ကိုသုံးပါတယ်။ ထို li tag ကိုမှ ul and ol စတဲ့ tag နှစ်ခုနဲ့ တွဲသုံးပါတယ်။ ul သည် unordered list ကို ဆိုလိုခြင်းဖြစ်ပြီး list များကို ဖော်ပြတဲ့ အခါ number or symbols တွေနဲ့ မဖော်ပြပါဘူး။ ol ( ordered list ) ကတော့ number များနဲ့ ဖော်ပြ ပေးပါတယ် ထိုပြင် symbol များဖြင့်လည်း ဖော်ပြ နိုင်ပါသေးတယ်။

```

</head>
<body>
    <ul>
        <li>name</li>
        <li>email</li>
        <li>phNumber</li>
    </ul>
    <ol>
        <li>name</li>
        <li>email</li>
        <li>phNumber</li>
    </ol>
</body>

```

**Output**

- name
  - email
  - phNumber
1. name
  2. email
  3. phNumber

List တွေ တဲ့မှာ style attribute ကိုသုံးပြီးတော့လည်း မိမိလိုချင်တဲ့ ပုံစံကို ချိန်းနိုင်ပါသေး တယ် သို့သော် CSS ကိုသုံးပြီးတော့ပြု style များကို လိုသလို ချိန်းသင့်ပါတယ်။

```

<body>
    <ul style="list-style-type:square;">
        <li>name</li>
        <li>email</li>
        <li>phNumber</li>
    </ul>
    <ol style="list-style-type: upper-alpha;">
        <li>name</li>
        <li>email</li>
        <li>phNumber</li>
    </ol>

```

**Output**

- name
  - email
  - phNumber
- A. name
  - B. email
  - C. phNumber

ထိုပြင် list တဲ့ဗျာမှု `dt`, `dd` တို့လည်း ရှိပါသေးတယ်။ `dt` သည် `description list` ဖြစ်ပြီး သူ့အတဲ့မှာမှ `dt` and `dd` တို့ကို ရေးပါတယ်။ `dt` သည် `name` ကို ဖော်ပြု ပြီး `dd` သည် အချက်အလက် များကို ဖော်ပြု ပေးပါတယ်။ ဥပမာ `dt` သည် မောင်မောင် ဖြစ်တယ်ဆုံး `dd`

သည် mgmg ရဲ့ အချက်လက်များကို ဖော်ပြု ပေးပါတယ်။

```
<dl>
    <dt>mgmg</dt>
        <dd>My name is mgmg i am 18 years old</dd>
    <dt>susu</dt>
        <dd>I am susu , 20 years old</dd>
</dl>
```

## Output

```
mgmg
    My name is mgmg i am 18 years old
susu
    I am susu , 20 years old
```

## Nested List

Nested list ဆိုတာကတော့ list ထဲမှာ နောက်ထပ် list တွေ ထပ်ထည့် တာကို ဆိုလိုတာပါ။

- name
- ◦ Nested1
- ◦ Nested2
- email
- phNumber

```
<ul style="list-style-type: none;">
    <li>name</li>
        <ul>
            <li>Nested1</li>
            <li>Nested2</li>
        </ul>
    <li>email</li>
    <li>phNumber</li>
</ul>
```

## Tables

Tables တွေကို data တွေ အစဉ်လိုက် store လုပ်နိုင်ဖို့ သုံးပါတယ်။ tables တွေကိုလည်း နှစ်ပိုင်းခဲ့ပြီး အသုံးပြုနိုင်ပါတယ်။ table ထဲမှာမှ thead ( table head ) နှင့် tbody ( table body ) ဆိုပြီး ခဲ့လိုက်ပါတယ် thead ထဲမှာတော့ table ရဲ့ head ပိုင်းမှာ ဖော်ပြချင်တဲ့ အချက်လက်တွေကို th tag သုံးပြီး ဖော်ပြ နိုင်ပါတယ်။

Name	email	Phumber
WinHtut	winhtut@gmail.com	09R100
Vijja	Vijja@gmail.com	09R100
MrHtut	MrHtut>@gmail.com	09R100

```
<!-- Tables-->
<table>
  <thead>
    <tr> ←
      <th>Name</th>
      <th>email</th>
      <th>Phumber</th>
    </tr>
  </thead>
  <tbody>
    <tr> ←
      <td>WinHtut</td>
      <td>winhtut@gmail.com</td>
      <td>09R100</td>
    </tr>
    <tr> ←
      <td>Vijja</td>
      <td>Vijja@gmail.com</td>
      <td>09R100</td>
    </tr>
    <tr> ←
      <td>MrHtut</td>
      <td>MrHtut>@gmail.com</td>
      <td>09R100</td>
    </tr>
  </tbody>
</table>
```

### Inline Vs Block-level elements

Inline element တွေဟာ များသာ အားဖြင့် text-based တွေဖြစ်ပြီး block level တွေကတော့ structural အတွက် အသုံးပြုပါတယ်။ inline and block levels တွေကို ကဲပြားစွာ သိထားဖို့ လိုအပ်ပါတယ် ထိုသို့ သိထားမှသာ design ကို မိမိတို့ လိုသလို ဖန်တီးနိုင်မှာ ဖြစ်သလို information များကိုလည်း သိသာထင်ရှားအောင် လုပ်ပေးနိုင်မှာ ဖြစ်ပါတယ်။ ဥပမာ span and div တွဲင် span သည် inline element ဖြစ်ပြီး div သည် block-level ဖြစ်ပါသည်။ inline element ကို စာသားများ ကြားတွင် ညှပ်သုံးသော်လည်း ထိုစာသားများကို ခြား ပြစ်လိုက်ခြင်း နောက်တစ်ကြောင်းသို့ ဆင်းသွားခြင်း စသည်တို့ကို လုပ်ဆောင်မှာ မဟုတ်ပါဘူး။ သို့သော် div ကဲသို့ block elements များကို စာသားများကြားတွင် ထည့်လိုက်မည် ဆုံးလျှင်တော့ စာသားများ ဗုံးအောက်တစ်ကြောင်းသို့

## ဆင်းသွားမှာ ဖြစ်သလို ခြားသွားမှာလည်း ဖြစ်ပါတယ်။

```
<body>
    <h3>Hello I am win htut From Green Hackers Team.<span>Unit is power </span> so
        we need to unit each other and need mutual respect.</h3>
</body>
</html>
```

အထက်ပါ ပုံထဲတွင် h3 tag ထဲ၌ span inline element ကိုထည့်ထားပါသည် သို့သော် Output တွင် စာသားများကို သီးသန့် ခွဲထုတ်လိုက်ခြင်း block တစ်ခု ပြုလုပ်လိုက်ခြင်း မရှိပါဘူး။

Hello I am win htut From Green Hackers Team.Unit is power so we need to unit each other and need mutual respect.

```
<h3>Hello I am win htut From Green Hackers Team.<div>Unit is power </div> so
    we need to unit each other and need mutual respect.</h3>
```

အထက်ပါ program တွင် div block element ကို အထဲ၌ ထည့်လိုက်သော အခါ output တွင် ခြားနားသွားသည်ကို တွေ့ရပါမည်။

**Hello I am win htut From Green Hackers Team.  
Unit is power  
so we need to unit each other and need mutual respect.**

အသုံးများတဲ့ inline elements တွေကတော့ အောက်ပါ အတိုင်း ဖြစ်ပါတယ်။

<a> <button> <cite> <code> <data> <em> <i> <select> <span> <time> <video>

အသုံးများတဲ့ block-level elements များ

<address><article> <div> <h1> <header> <li> <main> <nav> <ol> <p> <table> <ul>

### Class vs Id

Class and Id ကြားမှ အဓိက ခြားနားချက်သည် class သည် name တစ်ခုတည်းကို tag ပေါင်းများစွာ ထဲတွင် အသုံးပြုနိုင်သော်လည်း Id သည် tag တစ်ခုတွင် name တစ်ခု သာ အသုံးပြု နိုင်ပါသည်။ ဥပမာ css ကိုသုံးပြီး style ပြင်မည် ဆိုပါစို့ ထို style ရဲ့ class name ကို မိမိတို့ပြင်လို သည့် tag တိုင်းတွင် လိုက်ထည့် နိုင်ပါသည်။

```

<title>Document</title>
<style>
→ #h3{
    color: brown;
    background-color: black;
}
→ .p{
    color: darkmagenta;
    background-color: darkorange;
}
</style>
</head>
<body>
    <h3 id="h3">Hello I am win htut From Green Hackers Team.
    <div>Unit is power </div>
    so we need to unit each other and need mutual respect.</h3>
    <p class="p">Address</p>
    <p class="p">Myanmar</p>
    <p class="p">Cyber City</p>
</body>

```

အထက်ပါ Program တွင် h3 tag အား id attribute ကိုသုံးထားပါသည် p tag များတွင်တော့ class attribute ကို သုံးထားပါသည်။ ထို့ကြောင့် p class အား css ဖြင့် design ပြင်လိုက်သော အခါတွင် ထို့ design သည် p tag အားလုံးတွင် သက်ရောက်သွားမှာဖြစ်ပါတယ်။

Hello I am win htut From Green Hackers Team.  
 Unit is power  
 so we need to unit each other and need mutual respect.

Address

Myanmar

Cyber City

## iFrame

အခြား webpage တစ်ခုမှ အချက်လက်တွေကို မိမိတို့ ယခု လက်ရှိ ရှိနေတဲ့ webpage မှာ ဖော်ပြချင်တယ်ဆိုရင် iframe tag ကို သုံး နိုင်ပါတယ်။ iFrame tag တဲ့တွင် src attribute ကိုသုံးပြီး properties နေရာတွင် မိမိတို့ ဖော်ပြချင်သော webpage ရဲ့ address ကို ဖော်ပြပေးရပါမည်။

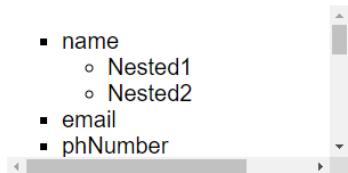
```

<body>
    <h1>Hello This is testing for iFrame</h1>
    <iframe src="app.html" frameborder="0"></iframe>
</body>

```

အထက်ပါ အတိုင်းရေးမည် ဆိုလျှင် app.html ထဲမှ အချက်လက်များကို အောက်ပါ အတိုင်း ဖော်ပြ ပေးမှာ ဖြစ်ပါတယ်။

## Hello This is testing for iFrame



ထိုပြင် iFrame ရဲ့ width and height တိုကိုလည်း မိမိတို့ လိုအပ်သလို ပြုပြင် နိုင်ပါတယ် ထိုသို့ ပြုပြင်ရာတွင် width and height attribute များကို တိုက်ရှိက်သုံးနိုင်သလို style attribute ကို သုံးပြီး width and height များကို style ရဲ့ properties များ အနေဖြင့်လည်း အသုံး ပြုနိုင်ပါတယ်။

```
<iframe src="app.html" width="400" height="300"></iframe>
```

အောက်တွင် style attribute ကိုသုံးပြီးရေးပြထားပါတယ်။

```
<iframe src="app.html" style="width: 400px; height:200px"></iframe>
```

iFrame ကို အသုံးပြုရင် output မှာ default အနေဖြင့် frameborder ပါ ပါတယ် ထို frameborder ကို ဖျက်ထုတ်ချင်တော့ ဖြစ်ဖြစ် ပြင်ချင်တော့ ဖြစ်ဖြစ် style attribute ကိုသုံးပြီးတော့လည်း ပြင်နိုင်ပါတယ်။ အကယ်၍ border မထားချင်ဘူးဆိုလျှင်တော့ border properties မှာ value ကို none ဆိုပြီး ထားခဲ့ နိုင်ပါတယ်။ ထိုသို့ မဟုတ်ပြ မိမိတို့ စိတ်ကြိုက် ပြင်လို ပါက အောက်ပါ အတိုင်းလည်း ရေးနိုင်ပါတယ်။

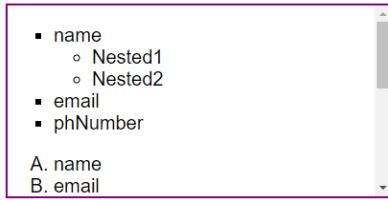
```
<iframe src="app.html" style="width: 400px; height:200px; border:2px solid #purple"></iframe>
```

iFrame ကို Link target အနေဖြင့်လည်းအသုံးပြု နိုင်ပါသေးတယ် ဆိုလိုချင်တာ iFrame ထဲမှာ မိမိတို့ အသေသတ်မှတ်ပေးထားတဲ့ page မဟုတ်ပဲ အခြား page တစ်ခုကိုပါ ထပ်မံ ဖော်ပြ ပေးတာမျိုးပါ ထိုသို့ ဖော်ပြ ပေးနိုင်ဖို့ iFrame မှာ name attribute ကိုအသုံးပြုပြီး name ပေးထားလိုက်ပါတယ်။ ထို name ကို အခြားတစ်နေရာကနေ target attribute ကိုသုံးပြီး လုမ်းခေါ်မယ်ဆိုရင် target attribute ပါဝင်နေတဲ့ tag ရဲ့ href link ကို iFrame ထဲမှာ ဖော်ပြ နိုင်မှာ ဖြစ်ပါတယ်။

```
<iframe src="app.html" style="width: 400px; height:200px;
```

```
border:2px solid #purple" name="myFrame"></iframe>
```

```
<p><a href="Inlinesvsblock.html" target="myFrame">Inlinesvsblock</a></p>
```

Inlinevsblock

**Hello I am win htut From Green Hackers Team.**  
**Unit is power**  
**so we need to unit each other**  
**and need mutual respect.**

AddressInlinevsblock

## CSS ( Cascading Style Sheets )

CSS ကို presentational technology လိုလည်း ပြောကြသလို CSS သည် standard style sheet language တစ်ခု ဖြစ်ပါတယ် သူ့ကို အဓိက အနေနဲ့ web page တွေရဲ့ layout တွေ formating တွေကို မိမိတဲ့ လိုသလို ပြုပြင်ပြောင်းလဲတဲ့ နေရာမှာ အသုံးပြုပါတယ်။ CSS မတိုင်ခင်တုန်းက font colors , background styles , element alignment , borders and sizes တွေ အားလုံးကို HTML tags တွေထဲမှာပဲ ထည့်ရေးခဲ့ကြပါတယ်။ ထိုကဲ့သို့ ပြုမှုခဲ့ခြင်းရဲ့ ရလဒ် အနေနဲ့ websites အကြိုးကြိုးတွေကို ရေးသားတဲ့ အချိန်မှာ အချိန်အရမ်းကြောပြီး code ရေးရတဲ့ လုပ်ငန်းစဉ်တွေဟာလည်း အရမ်းကို ကြောခဲ့ပါတယ်။ pages တိုင်းမှာ တူညီတဲ့ style တစ်ခု ပါနေခဲ့မယ် ဆုံးရင် pages တိုင်းအတွက် တူညီတဲ့ style တွေကို လိုက်ရေးရပါတယ် ထိုကဲ့သို့ ပြုသနတွေကို ဖြေရှင်းဖို့ 1996 မှာ World Wide Web Consortium(W3C) ကနေပြီး CSS နည်းပညာကို မိတ်ဆက်လာခဲ့ပါတယ်။ CSS ကြောင့်မို့ HTML ဟာ ပိုပြီး ရိုးရှင်းလာသလို maintain လုပ်ရတာလည်း လွှာယူလာပါတယ်။ CSS3 သည် CSS ရဲ့ နောက်ဆုံး version ဖြစ်ပြီး CSS3 မှာ features အသစ်တွေအများကြီး ပါဝင်လာသလို web page တွေရဲ့ လုပ်ဆောင်ချက်တွေကလည်း ပိုမို ကောင်းမွန်လာပါတယ်။

CSS ကိုသုံးခြင်းအားဖြင့် တူညီတဲ့ style တွေကို တစ်ခါတည်းရေးပြီး အခြားသော elements အများကြီးမှာပြန်သုံး နိုင်ပါတယ်။ Style sheet တစ်ခုတည်းကို ရေးသားထားပြီးအခြားသော pages တွေမှာလည်း ပြန်သုံးနိုင်ပါတယ် ထိုပြင် page တစ်ခုတည်းကိုပဲ devices မျိုးစုံမှာ ပုံစံ မျိုးစုံနဲ့ ပြပေး နိုင်ပါတယ်။ JavaScript ကို အသုံးမပြုပဲနဲ့ CSS ကိုပဲ အသုံးပြုပြီး animations and transitions effects တွေကို ပြုလုပ်

နိုင်ပါတယ်။ CSS ကို သုံးခြင်း အားဖြင့် အဓိက ရလာမယ့် ကောင်းကျိုးတွေကတော့ အောက်ပါ အတိုင်း ဖြစ်ပါတယ်။

- 1- CSS ကို တစ်ကြိမ်တည်းသာ ရေးသားထားပြီး HTML elements အများကြီး အတွက် ပြန်လည် အသုံးပြုနိုင်အတွက် codes တွေဖောင်းပွဲမှုကိုလည်း တားဆီးနိုင်သလို အချိန်ကုန်တာကိုလည်း အများကြီး သားတာ ပေါ်ပါတယ်။
- 2- web pages တွေ အများကြီးကို style sheets အနည်းငယ်နဲ့ control လုပ်လိုရတဲ့အတွက် လွှာယူကူစွာ maintenance လုပ်နိုင်ပါတယ်။
- 3- web pages တွေ အများကြီးအတွက် တူညီနေတဲ့ style တွေကို codes တွေ အများကြီး ထပ်ခါထပ်ခါ ရေးစရာ မလိုတဲ့ အတွက် အဲဒါ ရလဒ်ဟာ page ရဲ့ loading အချိန်ကို လျော့နည်း ပေါ်ပါတယ်။
- 4- CSS ကိုသုံးခြင်းအားဖြင့် မတူညီတဲ့ devices မျိုးစုံမှာ မတူညီတဲ့ styles မျိုးစုံကို ပြပေးနိုင်ပါတယ်။

### Three Methods

HTML documents တွေထဲမှာ CSS ကို နည်းလမ်းသုံးမျိုးနဲ့ embedded လုပ်နိုင်ပါတယ်။

- Inline styles - Inline styles ဆိုတာကတော့ HTML ရဲ့ tag တွေထဲမှာ style ဆိုတဲ့ attribute ကိုသုံးပြီး CSS features တွေကို ထည့်သုံးတာပါ။
- Embedded styles - Embedded styles ဆိုတာက HTML ရဲ့ head section ထဲမှာ <style> element ကို CSS code များကို ရေးသားတာပါ။
- External style sheets - External style sheet ဆိုတာက <link> element ကိုသုံးပြီး css file သီးသန့်တစ်ခု ရေးကာ ထို file နဲ့ html file ကို ချိတ်ပေးတာပါ။

Priority အနေဖြင့် ကြည့်မည် ဆုံးလျှင် inline styles ကို ပိုပြီး ဦးစားပေး ခံရပါတယ် external style sheets ကတော့ နောက်ဆုံးမှာပါ။ ဆုံးလုံချင်တာက inline styles နဲ့လည်း HTML tag တွေထဲမှာ ထည့်ရေးထားလို့ external style sheets အနေနဲ့လည်း HTML tag တွေကိုလမ်းပြီး နှစ်ခု တစ်ပြိုင်တည်း control လုပ်ထားမယ်ဆုံးရင် inline styles ကိုသာ အလုပ်လုပ်သွားမှာ ဖြစ်ပါတယ်။ အကယ်၍ embedded styles နဲ့ external styles နှစ်ခု တစ်ပြိုင်တည်း လုပ်ထားမယ်ဆုံးရင်လည်း embedded style ကိုသာ အသုံးပြုသွားမှာ ဖြစ်ပါတယ်။ External style sheet မှာမှ linking external style sheets နဲ့ importing external style sheets ဆုံးပြီး နှစ်မျိုး ရှိပါသေးတယ်။

### Inline

```

    inline style sheet      properties      value      semicolon
    ↓                         ↓             ↓           ↓
<iframe src="app.html" style="width: 400px; height:200px;
border:2px solid purple" name="myFrame"></iframe>
  
```

## Embedding

```

<title>iFrame</title>
<style>
  body{
    background-color: black;
    font-style: italic;
    font: medium;
  }
</style>
</head>
<body>
  <h1>Hello This is testing for Embedded Style Sheet</h1>
</body>
  
```

embedding

## External linking

### test.html

```

<title>iFrame</title>
<link rel="stylesheet" href="/Bok/css/style.css">
</head>
<body>
  <h1>Hello This is testing for Embedded Style Sheet</h1>
</body>
  
```

### style.css

```

body{
  background-color: black;
  font: 20px Arial, sans-serif;
  color: chartreuse;
}
  
```

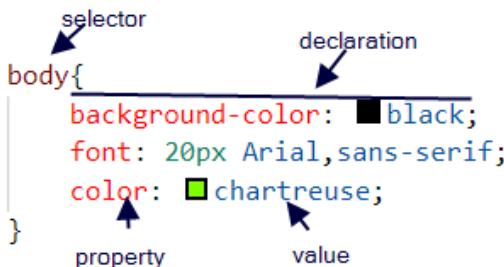
## Importing External Style Sheets

```

<style>
  @import url("/Bok/css/style.css");
</style>
</head>
<body>
  <h1>Hello This is testing for Embedded Style Sheet</h1>
</body>
  
```

### CSS syntax

CSS ကို အသုံးပြုဖို့ ဆိုရင် သူ့ရဲ့ rule တွေကိုလည်း သိဖို့ လိုပါတယ်။ CSS မှာ အရေးကြီးတဲ့ rule နှစ်ခု ရှိပါတယ် ပထမ တစ်ခုက selector ဖြစ်ပြီး ဒုတိယ တစ်ခုက declaration တွေ ဖြစ်ပါတယ်။



Selector ဆိုတာက HTML page ထဲက ဘယ် elements တွေကို CSS ရဲ့ rule တွေ အသုံးပြုမလဲ ဆိုတာကို သတ်မှတ် ပေးရတာပါ။ Declaration ကတော့ elements တွေဟာ web page ပေါ်မှာ ဘယ်လို ပုံစံနဲ့ ရှိမလဲ ဆိုတာကို ဖော်ပြ ပေးတာပါ။ Delaration တစ်ခုစီတိုင်းမှာ အပိုင်း နှစ်ပိုင်း ပါဝင်ပါတယ် ပထမ တစ်ခုက property ဖြစ်ပြီး ဒုတိယ တစ်ခုက value ဖြစ်ပါတယ်။ Declaration အပိုင်းတွေကိုတော့ curly braces တွေထဲမှာ ထည့်ရေးပါတယ်။ Declaration ရဲ့ properties အပိုင်းမှာ background-color တွေ font တွေ စသည်တို့ ပါဝင်ပြီး value အပိုင်းမှာတော့ blue စတဲ့ color တွေနဲ့ center စတဲ့ တည်နေရာတွေ ပါဝင်ပါတယ်။

### Comment in CSS

Comments တွေဟာ အမြဲတမ်း အရေးကြီးတဲ့ အပိုင်းမှာ ပါဝင်ပါတယ် comments တွေကို program က ထည့်ပြီး အလုပ် မလုပ်သော်လည်း comments တွေကြောင့်သာ program တွေကို ဖတ်ရ လွယ်ကူစေပြီး ထိန်းသီမ်းရတာ ပိုမို ထိရောက်စေတာပါ။ CSS ရဲ့ comment တွေအစကိုတော့ /\* နဲ့ \*/ ဖြစ်ပြီး အဆုံးကိုတော့ \*/ နဲ့ ဆုံးပါတယ်။

```

/*This is comment for CSS*/ ←
body{
    background-color: black;
    font: 20px Arial, sans-serif;
    color: chartreuse;
}
/*This is multiple ←
comment for CSS*/
  
```

CSS ရဲ့ property names နဲ့ values တွေဟာ case-sensitive မဖြစ်ပေမယ့် selectors တွေကတော့ case-sensitive ဖြစ်ပါတယ်။ ဥပမာ .hello နဲ့ .Hello က မတူပါဘူး။

## Selector

CSS ရဲ့ အရေးကြီးဆုံး အပိုင်းကို ပြပါ ဆုံးရင် selector တွေကို ပြရမှာပါ။ Selector တွေရဲ့ လုပ်ဆောင်ချက်ကတော့ ကိုယ်သတ်မှတ်ထားတဲ့ elements တွေကို အတိအကျ ကိုင်တွယ် နိုင်တာပါ။ CSS မှာ selector 9 ခုခန့်ရှိပါတယ် တစ်ခုချင်းရဲ့ လုပ်ဆောင်ပုံကိုလည်း အောက်မှာ ဖော်ပြ ပေးသွားပါမယ်။

### Universal Selector

Universal Selector ဆိုတာက web page မှာ ရှိတဲ့ elements အားလုံး select လုပ်လိုက်တာပါ။ Universal Selector ကို asterisk(\*) နဲ့ ဖော်ပြပါတယ် အများဆုံး အသုံးပြုတာကတော့ default ဖြစ်နေတဲ့ margins and paddings တွေကို ပြောင်းလဲ ဖို့ သုံးပါတယ်။

```
<style>
  *{
    margin:0;
    padding: 0;
  }
</style>
```

Universal selector ကိုတော့ အသုံးနည်းပါတယ် ဘာကြောင့်လည်း ဆိုတော့ web page ရဲ့ elements အားလုံးကို တစ်ပုံစံတည်း style rules များအတိုင်း ပြောင်းလိုက်လိုပါ။ Universal selector အစား element type or class selector တို့ကို သုံးနိုင်ပါတယ်။

### Element Type selector

Element type selector ကြတော့ web page ပေါ်မှာ ရှိတဲ့ သူနဲ့ သက်ဆိုင်ရာ အားလုံးကို style rules များအတိုင်း ပြောင်းပေးလိုက်တာပါ။ ဥပမာ ပ ဆိုတဲ့ element ကိုသုံးမယ်ဆုံးရင် web page ပေါ်မှာ ရှိတဲ့ p tag တွေ အားလုံးကို ပြောင်းလဲလိုက်မှာပါ။

```
p{
  margin:0;
  padding: 0;
}
```

### Id selectors

Id selector က ကြတော့ ကိုယ် style rules အသုံးပြုချင်တဲ့ element တစ်ခုချင်းစိတ် #color စသေဖြင့် # sign ကို ရှုံးက ခံပြီး အသုံးပြုလိုက်တာပါ။ ဥပမာ <p id="name"> မှာ ဆုံးရင် name သည် id ရဲ့ name ဖြစ်သွားပါတယ် ထို့ကြောင့် p tag ကို ပြုပြင်ချင်တယ် ဆုံးရင် #name ဆုံးပြီးတော့ style rules များကို နောက်မှာ ရေးပေးပါတယ်။

```

<style>
|   #name{
|     color: chartreuse;
|   }
</style>

```

### Class selectors

Class selector ကတေသာ အများဆုံး အသုံးပြုတဲ့ selector အမျိုးအစား ဖြစ်ပါတယ် class selector ကို သုံးဖို့ မိမိတို့ style rules အသုံးပြုချင်တဲ့ tag ထဲမှာ class attribute ကို သုံးပြီး name ရေးပေးလိုက်ရှုသာ ဖြစ်ပါတယ်။ style rules များရေးတဲ့ အခါက္ခရာင်တော့ class name ရဲ့ ရှေ့မှာ dot( . ) ကိုသုံးပြီး ရေးရပါတယ်။

```

.name{
  color: chartreuse;
}
...
```

Class selector ကို ဘာကြောင့် အသုံးများလဲ ဆိုတော့ style rules တစ်ခုတည်းကိုသာ ရေးထားပြီး မိမိတို့ သုံးချင်တဲ့ elements တွေထဲမှာ class name ကိုသာ ထည့်လိုက်ခြင်းဖြင့် အသုံးပြု နိုင်ပါတယ်။ အောက်ပါ program မှာ class name တစ်ခုတည်းကနော 4 ခုမှာ အသုံးပြုပုံကို ပြထားပါတယ်။

```

<style>
|   .name{ ←
|     color: chartreuse;
|   }
</style>
</head>
<body>
  <ul>
    <li>name</li>
    <ul>
      ←> <li class="name">Nested1</li>
      ←> <li class="name">Nested2</li>
    </ul>
    <li class="name">email</li> ←
    <li class="name">phNumber</li> ←
  
```

### Descendant Selectors

Element တစ်ခုထဲမှာ ရှိတဲ့ element တွေကို select လုပ်ချင်ရင် descendant selector ကိုသုံးပါတယ် ဥပမာ <h1>Hello <em>World</em></h1> ထဲကမှ em tag ကို ပြင်ချင်တယ်ဆိုရင် h1 em { style rules } ယခုပုံစံဖြင့် ပြင်နိုင်ပါတယ်။

```

h1 em {
  color: chartreuse;
}

```

Class နဲ့ name သတ်မှတ်ပေးထားပြီးသား element တွေကို ပြင်ချင်တယ်ဆိုရင်လည်း

descendant selector ကို အောက်ပါ အတိုင်းသုံးနိုင်ပါသေးတယ်။ ပထမဆုံး ကိုယ် class သုံးထားတဲ့ element ကိုရေးရပါမယ် ထိုနောက်မှ dot ခံပြီး class name ကိုရေးရပါမည် ပြီးလျှင် အထဲမှ မိမိပြင်လိုသော tag name ကိုရေးပေးရပါမည်။

```

        ul.contact li a{
            color: blueviolet;
        }
    
```

```

</style>
</head>
<body>
    <ul class="contact">
        <li><a href="#">name</a></li>
        <li>email</li>
        <li>phNumber</li>
    </ul>

```

အထက်ပါ program တွင် ဗျာတဲ့ li , li တဲ့ကမှ a tag ကို ပြင်ချင်တာ ဖြစ်တဲ့ အတွက် ul.className li a {} စသေဖြင့် ရေးသားပါသည်။

### Child Selectors

Child selector ဆိုတာကတော့ ကိုယ်ကိုင်တွယ်ချင်တဲ့ element ကိုပဲ တိုက်ရိုက်လျမ်းပြီး select လုပ်တာပါ အကယ်၍ select လုပ်ချင်တဲ့ element က အထဲမှာ ရှိနေတယ်ဆိုရင် greater than symbol ( > ) ကို အသုံးပြု နိုင်ပါတယ်။ အထဲဆုံးက child element ကိုပဲ style rules effect ဖြစ်မည် ဖြစ်ပြီး ကျွန်ုတ်သည့် elements များကို effect ထံရောက်မည် မဟုတ်ပါ။

```

ul > li a{
    color: blueviolet;
}
ul > li b{
    color: brown;
}

```

```

</style>
</head>
<body>
    <ul class="contact">
        <li><a href="#">name</a></li>
        <li><b>email</b></li>
        <li>phNumber</li>
    </ul>

```

### Adjacent Sibling Selectors

Adjacent sibling selector ဆိုတာကတော့ level အတူတူ ရှိနေတဲ့ element တွေထဲကမှ မိမိ လိုချင်တဲ့ element ကို ဆွဲထုတ်ပြီး style rules အသုံးပြုတာပါ။ ဥပမာ h1 + p ဟု ရေးသားခဲ့လျှင် h1 နှင့် p တို့သည် level အတူတူရှိနေပြီး ထို ထဲကမှ p ကို select လုပ်ချင်တာ ဖြစ်ပါတယ်။

```

<style>
  h1 + p {
    color: brown;
    font-size: xx-large;
  }
  ul.contact + h2{
    color: cadetblue;
    font-size: xx-large;
    font-weight: 300;
    font-family: 'Times New Roman', Times, serif;
  }
</style>
</head>
<body>
  <h1>Adjacnet sibling selectors</h1>
  <p>This p will apply style rules</p>
  <ul class="contact">
    <li><a href="#">name</a></li>
    <li><b>email</b></li>
    <li>phNumber</li>
  </ul>
  <h2>This h2 also apply</h2>

```

## Adjacnet sibling selectors

This p will apply style rules

- name
- email
- phNumber

This h2 also apply

ထိုပြင် General Sibling Selectors ဆိုတာ ရှိပါသေးတယ် `h1 + p` ဆိုရင် + နေရာမှာ ~ ကို  
သုံးပါတယ်။

### Grouping Selectors

Grouping Selector ဆိုတာကတော့ မိမိတို့ style rule သုံးချင်တဲ့ element တွေ  
အားလုံးကို comma ခြားပြီး ရေးလိုက်တာပါ။ ဥပမာ `h1, h2, h3` အားလုံးကို  
ပြင်ချင်တယ် ဆိုရင် အောက်ပါ အတိုင်း ရေးလိုက်လို့ရပါတယ်။

```

H1,h2,h3{
  /* style rules*/
}

```

### CSS Color Property

CSS ရဲ့ `color` property ကတော့ `text` တွေရဲ့ `color` အတွက် အသုံးပြုပါတယ် ဥပမာ  
`body` ကို `select` လုပ်ထားပြီး `color` property ကိုသုံးကာ `color` သတ်မှတ်ပေးထားမယ် ဆိုရင်  
ထို `color` သည် `html web page` တစ်ခုလုံးမှာ ရှိတဲ့ စာသားတွေရဲ့ `color` ကို `sytle rules`  
သုံးပေးမှာ ဖြစ်ပါတယ်။

```
Body{
color: blue;
}
```

CSS ရဲ့ color တွက် ပုံစံ ၃ မျိုးနဲ့ သတ်မှတ် ပါတယ် red , green , blue စတဲ့ name တွေရေးပြီး သတ်မှတ်ကြသလို #fe0000 , ef0000 စတဲ့ HEX value တွေနဲ့လည်း သတ်မှတ်ကြပါတယ်။ ထိုပြင် rgb( 255 , 255 , 255 ) စသဖို့လည်း သတ်မှတ်ကြပါတယ်။

```
h1{
color: ■chartreuse;
}
body{
color: □#ffffff;
}
li{
color: ■rgb(255, 100, 255);
}
```

HEX value တွေနဲ့ ပြတဲ့ အခါ six-digit representation ပုံစံတွေနဲ့ ပြပေးပါတယ် #rrggbb ဟူရေးပြီး rr , gg and bb တို့သည် red , green and blue တို့ကို သတ်မှတ်ပေးထားခြင်း ဖြစ်ပါတယ်။ #ffffff သည် white color အတွက် ဖြစ်ပြီး #000000 သည် black color အတွက် ဖြစ်ပါတယ်။

RGB ဖြင့် ဖော်ပြပေးနိုင်ဖို့ အတွက် rgb() ဆိုသည့် function ကိုသုံးပြီး ထို function ထဲတွင် parameters သုံးခု ပါဝင်ပါတယ် ပထမ တစ်ခုသည် red , green and blue တို့ စသဖို့ဖြင့် အစဉ်လိုက် ဖြစ်ပါတယ်။ ထို color တွက် value တွေနဲ့ ဖော်ပြပါတယ် 8-bit representation ပါ 8-bit မှာ 256 ဖြစ်ပြီး 1 လျှော့လိုက်တဲ့ အတွက် 255 ပါ ထိုကြောင့် color တစ်ခုစိုးရဲ့ တန်သုံးကို အနိမ်းဆုံး zero ကနေ အများဆုံး 255 ထို ဖော်ပြပေးနိုင်ပါတယ်။ ဥပမာ rgb ( 200 , 255 , 0 ) ဆိုရင် ထို color သုံးခု လုံးကို ပေါင်းလိုက်တဲ့ အရောင် ရလာမှာပါ။ rgb သုံးခုလုံးကို 255 စီအပြည့် ပေးထားမယ် ဆိုရင်တော့ white color ကို ရမှာ ဖြစ်ပြီး 0 တွေချည်း ပေးထားမယ်ဆိုရင်တော့ black ကိုရမှာ ဖြစ်ပါတယ်။

### Color on Borders and Outlines

Color property ကို text content တွေ အတွက်ချည်းပဲ အသုံးပြု နိုင်တာ မဟုတ်ပါဘူး border-color or outline-color အတွက် value သီးသန့် မသတ်မှတ် ပေးထားဘူးဆိုရင် color value ကိုပဲ အသုံးပြု သွားမှာ ဖြစ်ပါတယ်။

```
h1{
color: ■chartreuse;
outline: 2px solid;
}
```

## Outline Color

### Background

Background ရဲ့ web page တစ်ခုရဲ့ အရေးပါဆုံး အပိုင်းမှာ ပါဝင်ပါတယ်။ Background အတွက် CSS မှာ properties များစွာ ရှိပါတယ် background ရဲ့ color , background ရဲ့ image position စသည်ဖြင့် properties များစွာ ရှိပါတယ်။ background-color , background-image , background-repeat , background-attachment and background-position တို့ဖြစ်ပါတယ်။

background -color property ကိုတော့ element ရဲ့ background color အတွက် အသုံးပြုပါတယ်။ background color ကိုလည်း color name , HEX value , RGB value စသည်တို့နဲ့ဖော်ပြုနိုင်ပါတယ်။

```
h1{
    background-color: #chartreuse;
    outline: 2px solid;
}
```

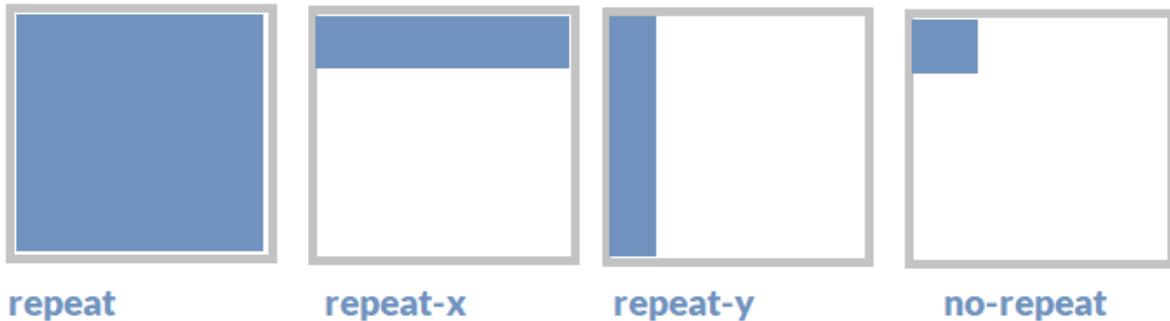
Background -image property ကိုတော့ HTML element ရဲ့ background မှာ image ထားနိုင်ဖို့ အတွက် သုံးပါတယ်။ အကယ်၍ page တစ်ခုလုံးမှာ image ထားချင်တယ်ဆုံးရင်တော့ body tag ကို သုံးနိုင်ပြီး image ထည့်ဖို့ အတွက် url function ကို အသုံးပြုရပါတယ်။

```
body{
    background-image: url("wall.jpg");
}
```

Background-image ကိုမှ ပုံသေးတယ် သို့မဟုတ် gradient color တွေ အတွက်ဆုံးရင် background-repeat properties နဲ့တွဲသုံးပါတယ်။

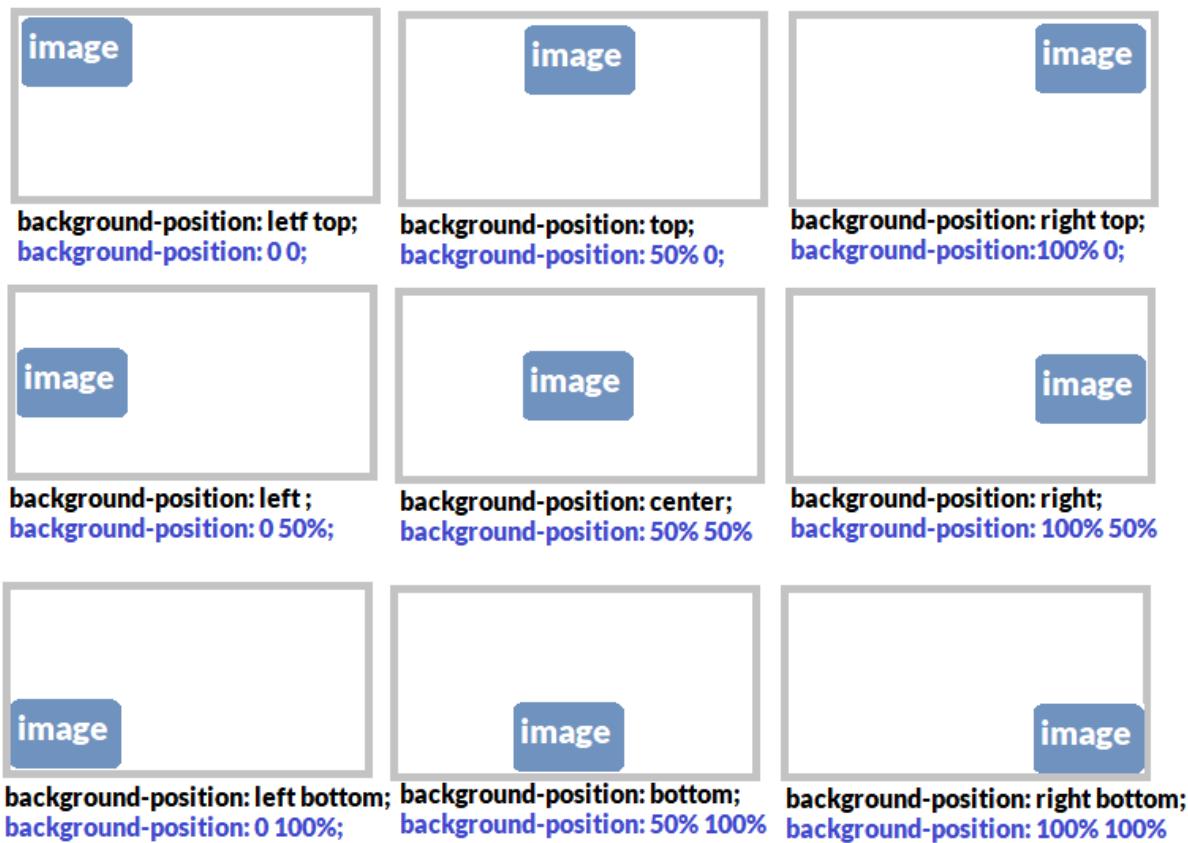
```
body{
    background-image: url("cy.png");
    background-repeat: repeat-y;
}
```

Background-repeat property နောက်မှာမှ value အနေဖြင့် repeat-y or repeat-x ကို သုံးပါတယ်။ repeat-y ကို အသုံးပြုမည် ဆုံးလျှင် Y ဝင်ရှိးအတိုင်း ပုံတွေကို ထပ်ထပ်ပြီး repeat လုပ်ပေးပါတယ်။ repeat-x ကို အသုံးပြုမည် ဆုံးလျှင်တော့ X ဝင်ရှိးအတိုင်း ပုံတွေကို ထပ်ထပ်ပြီး repeat လုပ်ပေးသွားမှာ ဖြစ်ပါတယ်။ ထိုပြင် repeat ကိုသာ အသုံးပြုခဲ့မည် ဆုံးလျှင် page တစ်ခု လုံး အပြည့် repeat လုပ်ပေးမှာ ဖြစ်သလို no-repeat ဆုံးရင်တော့ repeat လုပ်ပေးမှာ မဟုတ်ပဲ တစ်ပုံတည်းသာ ဖော်ပြုပေးမှာ ဖြစ်ပါတယ်။



Background-image ကို background-repeat အပြင် background-position တိန်ငါးလည်း  
တွဲပြီး အသုံးပြုပါတယ်။ အကယ်၍ background-position ကို သတ်မှတ် မပေးထားဘူး  
ဆုလျှင် default အနေနဲ့ top-left မှာ နေရာ ယူမှာ ဖြစ်ပါတယ်။ percent အနေဖြင့် ဖော်ပြ  
ရမည် ဆုလျှင် 0% 0% ဖြစ်ပါတယ်။

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



Background-attachment property ကိုတော့ image ကို လုံးဝ မရွှေ့လိုတဲ့ အခါမျိုးမှာ အသုံးပြုပါတယ်။

```
body{
    background-image: url("cy.png");
    background-repeat: no-repeat;
    background-attachment: fixed;
}
```

### Background Attachment

Background-attachment property ကေတော့ background image ကို fixed ထားမလား မထားဘူးလားဆိုတာ အတွက် အသုံးပြုပါတယ်။ ဥပမာ ပုံအသေး တစ်ပုံကို web page ရဲ့ အလယ်မှာ background-position property ကိုသုံးပြီး ထားလိုက်သည် ဆုပါစို့ mouse ကို scroll လုပ်သွားရင်တောင် ကိုယ်သတ်မှတ်ထားတဲ့ နေရာမှာပဲ image ကို ရှိစေချင်တဲ့ အခါမျိုးမှာ attachment ကိုသုံးနိုင်ပါတယ်။

```

body{
  background-image: url("cy.png");
  background-attachment: fixed;
  background-position: center;
  background-color: black;
  background-repeat: no-repeat;
}
p,h1{
  color: white;
}

```

## Hello This is testing for Embedded Style Sheet



Even though Israel is a small country it is always in the news. The news continues to report on Jews moving to Israel, on the technology invented there, but also on conflict, wars and tensions with surrounding people. Why? A look at Israel's history.

### CSS Fonts

Web page မှာ ဖော်ပြထားတဲ့ design နဲ့ လိုက်ဖက်ညီတဲ့ fonts တွေကို ရွှေးချယ်ခြင်း ဟာလည်း web design ရေးသားရာမှာ အရေးပါတဲ့ အရာ တစ်ခု ဖြစ်ပါတယ်။ CSS မှာ font property နဲ့ ပတ်သက်ပြီး properties များစွာကို support လုပ်ထားပါတယ် font-family , font-style , font-weight , font-size and font-variant တို့ ဖြစ်ပါတယ် တစ်ခုခြင်းစီရဲ့ အလုပ်လုပ်ပုဂ္ဂတော့ အောက်မှာ ဖော်ပြသွားပါမည်။

#### Font Family

Font-family ကတော့ web page မှာ ဖော်ပြချင်တဲ့ text တွေရဲ့ font style အတွက် အသုံးပြုပါတယ်။ Font-family property ဟာ comma တွေခံပြီး font များစွာကို အသုံးပြု နိုင်ပါတယ်။ ပထမဆုံး ရေးထားတဲ့ font သည် အဆင်မပြေတော့ဘူးဆိုရင် ဒုတိယ font ကို အလုပ် လုပ်ပါတယ် ဒုတိယလည်း အဆင်မပြေသေးဘူးဆိုရင် တတိယ ကို အလုပ် လုပ်ပေးပါတယ် ထို စနစ်ကို fallback system လိုလည်းခေါ်ပါတယ်။ ထိုကြောင့် ကိုယ်အသုံးပြုချင်တဲ့ font ကို ပထမဆုံး နေရာမှာ ထားပေးရမည် ဖြစ်ပြီး ထို font သည် user ရဲ့ system မှာ အဆင်မပြေခဲ့ရင် fallback system ကိုအသုံးပြုသွားမှာ ဖြစ်ပါတယ်။ Font-family ကိုသုံးတဲ့ အခါမှာ သတိပြုစရာ အချက် ရှိနေပါသေးတယ် အဲဒါကတော့ တွဲလျက် မဟုတ်တဲ့ font-name တွေကို ရေးတဲ့ အခါကြရင် ထို font-names များအား quotation marks တွေ ထည့်ပေးဖို့ လိုအပ်ပါတယ်။ “Times New Roman” , “Courier New”.

Web design မှာ အများဆုံး အသုံးပြုတဲ့ font families တွေကတော့ serif and sans-serif တို့ ဖြစ်ပါတယ် ဘာလို့ ထို fonts တွေကို အသုံးများရလဲဆိုတော့ စာဖတ်ရတာ ပိုပြီး အဆင်ပြလိုပါ။

## Font Style

Font-style property ဟာ text content element အတွက်ရေးထားတဲ့ font face style အတွက် အသုံးပြုပါတယ်။ Font-style တွေဟာ normal , italic or oblique စသည်တို့ဖြစ်ပြီး အကယ်၍ font-style ကို ဘာမှ မပေးရင်တော့ default အနေဖြင့် normal ကိုအသုံးပြုပါတယ်။

```
p,h1{
    font-style: italic;
    color: white;
}
```

## Font Size

(text) တာသားနဲ့ ဖော်ပြချက်တွေအတွက် font များရဲ့ size တွေကို ပြင်ဆင်နိုင်ဖို့ font-size ဆိုတဲ့ properties ကို သုံးနိုင်ပါတယ်။ Font-size တွေကို မိမိတို့ လိုသလို သတ်မှတ်နိုင်ဖို့ CSS မှာ နည်းလမ်းများစွာကို support ပေးထားပါတယ် ထိုတဲ့ကမှ keywords , percentage , pixels and ems တို့ကတော့ အသုံးများပါတယ်။

### Font-size with Pixels

Font-size တွေကို တိတိကျကျ အသေးစိတ်ဖော်ပြလိုတဲ့ အခါမှာ pixels တွေကိုသုံးပြီးဖော်ပြပါတယ် ဘာကြောင့်လည်း ဆိုတော့ pixel သည် digital image or graphic တွေကို ဖော်ပြတဲ့ နေရာမှာအသေးဆုံး unit ဖြစ်လိုပါ။

```

<head>
    <style>
        h1{
            font-size: 14px;
        }
        h6{
            font-size: 36px;
        }
    </style>
</head>
<body>
    <h1>Hello this is h1</h1>
    <h6>hello this is h6</h6>

```

သတိပြုရန် အချက်မှာ pixels များကို သုံးသော အခါတွင် User တွေဘက်မှာ သေးနေတာမျိုး ဖြစ်နေနိုင်သလို ကြိုးနေတာလည်း ဖြစ်နိုင်ပါတယ် ဘာကြောင့်လည်းဆိုတော့ user တွေဟာ pixel နဲ့ ပေးထားတဲ့ အတွက်font-size ပြန်ချိန်းနှင့် အဆင်မပြေပါဘူး။ ထိုကြောင့် pixels များ အသုံးပြုခြင်းကို မပြုပဲ ထိုအတား value or percentage , em တို့ကို အသုံးပြု သင့်ပါတယ်လို့ W3C recommend ပေးထားပါတယ်။

### Font-size with EM

Em unit သည် Typographyရဲ့ unit တစ်ခုဖြစ်ပြီး မူရင်း element ရဲ့ font-size နဲ့ တူညီတယ်လို့ ဆိုလိုချင်တာပါ။ ဥပမာ font-size ကို 1em လို့ သတ်မှတ် လိုက်မည် ဆိုလျှင် ထို

font size သည် ယခု လက်ရှိ အသုံးပြုထားသော element ရဲ့ font-size နဲ့ အတူတူပဲဖြစ်ပါတယ်။ ဆုံလိုချင်တေက h1 element ရဲ့ font-size ကို 20px ဟု သတ်မှတ်လိုက်မည်ဆုံလျှင် 1em=20px ဖြစ်သွားမှာပါ 2em ဆုံလျှင် 40px ပါ။ အကေယ်၏ font-size ကို မသတ်မှတ် ပေးထားဘူးဆုံလျှင် normally အားဖြင့် font-size သည် 16px ဖြစ်ပါတယ်ထို့ကြောင့် 1em=16px ဖြစ်သွားမှာ ထိုနည်းတူ 2em=32px ပါ။

```

<style>
  div{
    font-size: 1em;
  }
</style>
</head>
<body>
  <span>hello this is h1<span>
  <div>hello this is h6<div>

```

အထက်ဖော်ပြပါ အတိုင်း စမ်းရေးကြည့်ပါ သူတို့ နှစ်ခုရဲ့ font-size အတူတူထွက်လေတာကို မြင်ရမှာပါ။ Chrome ရဲ့ Normal font-size သည် 16px ဖြစ်ပါတယ်ထို့ကြောင့် div သည်လည်း 1em=16px ဖြစ်နေတဲ့ အတွက် သူတို့ နှစ်ခုရဲ့ font-size က အတူတူဖြစ်နေတာပါ။

hello this is h1  
hello this is h6

### Percentage and EM

Em value တွေကို px value တွေနဲ့ တွဲမှတ်နိုင်ဖို့ ဆိုတာမလွယ် လှပါဘူး သို့သော်နည်းလမ်းတော့ ရှိပါတယ် ဘာကြောင့်လည်း ဆုံတော့ browser အများစုံမှာ 16px ဖြစ်ပါတယ်။ အဲ တော့ body တစ်ခုလုံးရဲ့ font size ကို 62.5% ထားလို့ရပါတယ်။ 62.5 သည် 16px ရဲ့ ရာခိုင်နှုန်းကို ပြောတာပါ။ ဘာကြောင့်လည်းဆုံတော့ 62.5% သည် 10px နဲ့ အတူတူပင် ဖြစ်သလို 0.625em လည်း ဖြစ်ပါတယ်။ body ရဲ့ element အားလုံး size သည် 10px ဖြစ်သွားတဲ့ အတွက် 10px သည် 1 em နဲ့ တူညီ သွားမှာပါ 12px=1.2 em ဖြစ်သွားသလို 16px=1.6em ဖြစ်သွားမှာပါ။

```

<style>
  body{
    font-size: 62.5% /*16px ရဲ့ 62.5% ဖြစ်သည်*/
  }
  p{
    font-size: 1.4em; /*1.4em = 14px*/
  }
</style>
</head>
<body>
  <span>hello this is span</span>
  <p>This is paragraphx</p>

```

## Root EM

Rem unit ကတော့ CSS3 မှာပါလာတဲ့ feature တစ်ခုပါ rem ဆိတာ root em ကို ဆိုလိုခြင်း ဖြစ်ပြီး သူကတော့ html ရဲ့ font-size နဲ့ သက်ဆိုင်ပါတယ်။ ဒီနေရာမှာ ရှုပ်ထွေးနိုင်တာ တစ်ခုက rem and em ပါ em သည် သူ့ရဲ့ parent emement နဲ့ပဲ သက်ဆိုင်တာပါ html တစ်ခုလုံးနဲ့ သက်ဆိုင်တာ မဟုတ်ပါဘူး ဥပမာ p tag or h1 tag ပေါ့။ rem ကတော့ html တစ်ခုလုံးနဲ့ သက်ဆိုင်ပါတယ် ထိုကြောင့် html ရဲ့ font-size ကို ပြင်မယ် ဆိုရင် အခြားသော font-size တွေလည်း ကွဲပြားသွားမှာပါ။ Browser တွေရဲ့ default font-size က em lesson မှာ ဆွဲးနွေးခဲ့သလို 16px ပါ ထိုကြောင့် html ရဲ့ font-size သည်လည်း 16px ဖြစ်ပါတယ်။ ထိုကြောင့် မိမိတို့ ရဲ့ web page မှာ font-size များကို မှတ်ရလွယ်ကူစေရန် html tag တစ်ခုလုံးကို font-size သတ်မှတ်ပေးထားလိုဂုဏ်ပါတယ်။ em lesson တွင် body ရဲ့ font-size ကို 62.5% or 10px ဟု သတ်မှတ်ခဲ့သလို ယခု rem အတွက်လည်း html ကို သတ်မှတ်ပြီး အသုံးပြုနိုင်ပါတယ်။

```

html{
  font-size: 62.5% /*16px ရဲ့ 62.5% ဖြစ်သည် ထိုကြောင့် 10px*/
}
p{
  font-size: 1.4rem; /*1.4rem = 14px*/
}

```

## Keywords

CSS မှာ font-size တွေကို keywords တွေနဲ့ အသုံးပြုနိုင်ဖို့ အတွက်လည်း support ပေးထားပါသေးတယ်။ Keywords တွေကတော့ xx-small , x-small , small , medium , large , x-large , xx-large တို့ဖြစ်တယ်။ Keywords တွေထဲက medium ရဲ့ font-size ကတော့ browser တွေရဲ့ default font-size နဲ့ တူပါတယ် ပုံမှန်အားဖြင့် 16px ပါပဲ xx-small ကတော့ 9px နဲ့ တူညီသလို x-small ကတော့ 10px နဲ့ တူညီပါတယ် small = 13px ,large = 18px , x-large=24px and xx-large=32 px !!

## Viewport Units

Font-size တွေကို viewport unit ကို သုံးပြီးတော့လည်း သတ်မှတ်နိုင်ပါတယ် viewport အတွက်တော့ vw or vh ကို သုံးပါတယ်။ viewport unit သည် browser ရဲ့ viewport dimension ကို percentage ဖြင့်ဖော်ပြခြင်းဖြစ်ပြီး 1vw သည် viewport width ရဲ့ 1% ကိုဆိုလိုခြင်းဖြစ် ပါတယ်။ ထိုနည်းတူ 1vh သည်လည်း viewport height ရဲ့ 1% ကိုဆိုလိုခြင်း ဖြစ်ပါတယ်။ အကယ်၍သာ viewport wide သည် 1600px ဖြစ်နေခဲ့မည် ဆိုလျှင် 1vw သည် 16px ဖြစ်ပါတယ်။ view port ကို ပိုမို အသေးစိတ် သိနိုင်ရန် <https://viewportsizer.com/> ယခု site များကြည့်မည်ဆိုလျှင် စာဖတ်သူတို့ ရဲ့ viewport ကို width and height ကို pixel နဲ့ သိနိုင်ပါတယ်။ viewport ကတော့ မိမိတူ့ browser ကို အကျဉ်းအကျယ် လုပ်တဲ့ အပေါ်မှာ မူတည်ပြီး pixel value တွေ ပြောင်းလဲ နေမှာပါ။

```
body{
    font-size: 1vw;
}
p{
    font-size: 3vw;
}
```

အထက်ပါ အတိုင်း viewport unit ကိုအသုံးပြုမည်ဆိုလျှင် small screen မှာရှိတဲ့ fonts တွေဟာ ပိုသေးလားပြီး ဖတ်ရတာ ခေါ်ခဲ့ နေမှာပါ ထိုပြုသနကို ဖြေရှင်းဖို့ အတွက် CSS ရဲ့ calc() ဆိုတဲ့ function ကို အသုံးပြု နိုင်ပါတယ်။ အသုံးပြုပုံကတော့ ထို function ထဲမှာ font-size ကို parameter တစ်ခု အနေဖြင့် ထည့်ပေးလိုက်မှာပါ result အနေနဲ့ small screen တွေပေါ်မှာ viewport value အနေဖြင့် အရမ်းသေးငယ် သွားရင်တောင် font-size သည် 1em or 16px ရှိနေမှာပါ။ ပိုပြီးတော့ responsive ဖြစ်တဲ့ web page တွေ ဖန်တီးဖို့ အတွက်ဆိုရင်တော့ CSS3 ရဲ့ feature တွေထဲကမှ media queries တွေကို အသုံးပြု နိုင်ပါတယ် ထို သင်ခန်းစာတွေကို Media Queries lesson မှာ ဆွေးနွေးသွားမှာ ဖြစ်ပါတယ်။  
calc(1em + 1vw)

## Font Weight

Font weight ဆိုတာက font တွေကို ထင်ထင်ရှားရှား (boldness ) သို့မဟုတ် မထင်မရှား မပေါ်စေချင်တဲ့ အချိန်မျိုးမှာ သုံးနိုင်ပါတယ်။ font-weight property ရဲ့ values များ အနေဖြင့် normal , bold , bolder , lighter , 100 , 200 , 300 , 400 , 500 , ... တို့အပြင် inherit ကိုလည်း သုံးနိုင်ပါတယ်။ inherit ဆိုတာ CSS ရဲ့ property တွေနဲ့တွဲသုံးနိုင်တဲ့ feature တစ်ခု ဖြစ်ပါတယ် သူကတော့ parent element ကနေပြီး value ကို ယူတာပါ။ font-weight တွေထဲက 400 ဆိုလျှင် normal နှင့်တူပြီး 700 ဆိုလျှင်တော့ bold နဲ့တူပါတယ်။

```

body{
    font-size: 1vw;
}
p{
    font-size: 3vw;
    font-weight: bold;
}

```

### Font Variant

Font-variant property သည် စာလုံးတွေကို special small-caps ပုံစံမျိုးဖြင့် ဖော်ပြန့် အတွက် သုံးနိုင်ပါတယ်။ small-caps or small capital letter သည် normal value နဲ့တော့ ကွဲပြားပါတယ် အကယ်၍သာ font-variant တွင် small-caps ကိုမသုံးပဲ normal ကိုသာသုံးမည် ဆိုလျှင် normal သည် မည်သိမှု ပြောင်းလဲ ခြင်း မရှိပဲ ပုံမှန် အတိုင်းသာ ဖော်ပြပေးပါသည်။

```

p{
    font-variant: normal;
}
span{
    font-variant: small-caps;
}
</style>
</head>
<body>
    <span>Font variant small-caps</span>
    <p>Font variant normal </p>

```

### Output

FONT VARIANT SMALL-CAPS

Font variant normal

### CSS Text

CSS မှာ text တွေနဲ့ ပတ်သက်ပြီး properties များစွာကို support လုပ်ထားပါသေးတယ်။ အထူးသဖြင့် text တွေရဲ့ color တွေ alignment , spacing , decoration , transformation တွေဖြစ်ပါတယ်။ အများဆုံး အသုံးပြုတဲ့ properties တွေကတော့ text-align , text-decoration , text-transform , text-indent , line-height , letter-spacing , word-spacing တို့ဖြစ်ပါတယ်။

### Text Alignment

Text-align property ကိုတော့ text တွေရဲ့ horizontal alignment အတွက်သုံးပါတယ်။ text-align တွေကို ပုံစံ လေးမျိုးနဲ့ အသုံးပြု နိုင်ပါတယ် left , right , center or justified တို့ဖြစ်ပါတယ်။ text-align ကိုသုံးတဲ့ ပုံပြီး သိသာ လွှာယက္ခဇာန် width နဲ့လည်း တွဲသုံးနိုင်ပါတယ် ဥပမာ horizontal ရဲ့ px ဘယ်လောက်ပဲ ယူမယ် ဆိုတာမျိုးပါ။

```

    p{
        font-variant: small-caps;
        width: 300px;
        text-align: justify;
    }
    h1{
        text-align: center;
        font-variant: normal;
    }

```

</style>

</head>

<body>

<h1>Heading Text</h1>

<p>Above is your current viewport size in pixels.

Width by Height. You can read more about viewport sizing here, or you can view our list of viewport dimensions by device. This is a free tool for measuring current viewport sizes for websites and apps as well as documentation of viewport and visible CSS dimensions of popular devices.

Some stats about your current browser are included below. </p>

## Output

### Heading Text

ABOVE IS YOUR CURRENT VIEWPORT SIZE IN PIXELS. WIDTH BY HEIGHT. YOU CAN READ MORE ABOUT VIEWPORT SIZING HERE, OR YOU CAN VIEW OUR LIST OF VIEWPORT DIMENSIONS BY DEVICE. THIS IS A FREE TOOL FOR MEASURING CURRENT VIEWPORT SIZES FOR WEBSITES AND APPS AS WELL AS DOCUMENTATION OF VIEWPORT AND VISIBLE CSS DIMENSIONS OF POPULAR DEVICES. SOME STATS ABOUT YOUR CURRENT BROWSER ARE INCLUDED BELOW.

အထက်ပါ output တွင် စာသားများသည် အစီအစဉ် အလိုက်လေး ဖြစ်နေပါတယ် ဘယ်လည်း မဟုတ်သလို ညာဘက်လည်း မဟုတ်ပါဘူး အလယ်မှာလည်း မဟုတ်ပါဘူး သူ့ဟာနဲ့ သူ သင့်တင့်သလို အစီအစဉ် တကျလေး ပေါ်ပေးပါတယ်။

### Text Decoration

Text-decoration property ကတော့ text တွေမှာ ထည့်ထားတဲ့ decoration တွေကို ဖယ်ထုတ်ဖို့ သို့မဟုတ် text တွေထဲကို ထပ်ထည့်ဖို့ ပြုပြင်ဖို့ သုံးပါတယ်။ text-decoration property သည် values များ အနေဖြင့် underline , overline , line-through and none တို့ ဖြစ်ပါတယ်။

```

.first{
  font-variant: small-caps;
  text-decoration: underline;
}
h1{
  font-variant: normal;
  text-decoration: overline;
}
.second{
  text-align: justify;
  text-decoration: line-through;
}

```

## Heading Text

ABOVE IS YOUR CURRENT VIEWPORT SIZE IN PIXELS

~~Some stats about your current browser are included below.~~

### Remove Underline

HTML မှာ link တွေကို ရေးပြီဆိုလျှင် ထို link စာသားရဲ့ အောက်မှာ underline လေးပါလာ ပါတယ်။ ထို Underline လေးကို ဖျောက်ဖို့ ဆိုရင်တော့ text-decoration ကို သုံးနိုင်ပြီး property အနေဖြင့် none ကိုသုံးနိုင်ပါတယ်။ ထိုပြင် border-bottom property ကိုသုံးပြီး underline style ကိုလည်း ပြောင်းလဲ နိုင်ပါတယ်။ သို့တည်း မဟုတ်ပဲ hover ကိုသုံးပြီး mouse pointer က ထို Link ပေါ်ကို ရောက်လာလျှင် အောက်မှာ ပေါ်နေတဲ့ underline လေးပျောက်သွားအောင်လည်း ရေးနိုင်ပါသေးတယ်။

```

<style>
  a{
    text-decoration: none;
    border-bottom: 1px dotted;
  }
  a:hover{
    border-bottom: none;
  }
</style>
</head>
<body>
  <p>Hello this is testing for link <a href="#">click here</a></p>

```

အထက်ပါ program မှာတော့ a tag ရဲ့ underline ကို ဖျောက်လိုက်ပြီး underline နေရာမှာ dot များဖြင့် ဖော်ပြထားပါတယ်။ ထိုပြင် hover ကိုပါ သုံးထားပြီး Link ပေါ်သို့ mouse ထောက်လိုက်တာနဲ့ တစ်ပြိုင်နှင်း dot ကလေးတွေပါ ပျောက်သွားမှာဖြစ်ပါတယ်။

Hello this is testing for link [click here](#)



### Text Transformation

Text-transform property ကတေသာ text တွေကို cases တွေ လုပ်ပေးဖို့ သုံးပါတယ်။ cases တွေဆိုတာက uppercase , capitalize , lowercase တို့ကို ပြောတာပါ။ Text တွေ တော်တော်များကို ရေးသားဖော်ပြတဲ့ အခါ mixed case တွေနဲ့ ရေးကြပါတယ် မိမိတို့ အနေဖြင့် စာလုံး အားလုံး အကြီး သို့မဟုတ် စာလုံးအားလုံး အသေး သို့မဟုတ် စာလုံး အားလုံးရဲ့ အစသည် အကြီး ကျိန်တာ အားလုံးသည် အသေး စသဖြင့် ပြုလုပ်လိုလျှင် text-transform ကို အသုံးပြု နိုင်ပါတယ်။

```

<style>
  a{
    text-decoration: none;
    border-bottom: 1px dotted;
  }
  a:hover{
    border-bottom: none;
  }
  p{
    text-transform: uppercase;
  }
  span{
    text-transform: capitalize;
  }
  div{
    text-transform: lowercase;
  }
</style>
<head>
<body>
  <p>This is for upper case <a href="#">click here</a></p>
  <span>This is for capitalize</span>
  <div>This is for Lower CASE</div>

```

### Output

THIS IS FOR UPPER CASE [CLICK HERE](#)

This Is For Capitalize  
this is for lower case

### Text Indentation

Text-indent property ကတေသာ indent လုပ်နိုင်ဖို့ အတွက် သုံးပါတယ် ။ indent လုပ်တယ် ဆိုတာ စာပိုဒ်တစ်ခုရဲ့ ပထမဆုံး စာ လိုင်းကို space အနည်းငယ် or tab တစ်ချက် ခြားပေးထားတာ မျိုးကို ဆိုလိုတာပါ။ text-indent property ကိုသုံးပြီး မိမိတို့ ပေးချင်သလို pixels သတ်မှတ်ပေးပြီး ပေးထားလို့ ရပါတယ် ထိုပြင် % percentage or ems တို့ကိုလည်းသုံးနိုင်ပါတယ်။

p{

text-indent: 100px;

```
}
```

### Letter Spacing

letter-spacing property ကတေသာ စာလုံး တစ်လုံးနဲ့တစ်လုံးကြားမှာ spacing ခြားဖို့အသုံးပြုတာပါ။ letter-spacing ကိုအသုံးပြုရာတွင် units များ အနေဖြင့် pixel , ems စသည်တို့ကိုလည်း သုံးနိုင်သလို pixel ကိုတေသာ အများဆုံး အသုံးပြုကြပါတယ်။ ထိုပြင် letter-spacing လုပ်ရာတွင် 10px စသုဖြင့်ရေးနိုင်သလို -2px ဖြင့်လည်း negative value ကိုရေးနိုင်ပါတယ်။

```

<head>
  <style>
    p{
      letter-spacing: -3px;
    }
    span{
      letter-spacing: 10px;
    }
  </style>
</head>
<body>
  <p>This is for upper case <a href="#">click here</a></p>
  <span>This is for capitalize</span>
  <div>This is for Lower CASE</div>

```

### Output

This is for uppercase [click here](#)

This is for capitalize  
This is for Lower CASE

### Word spacing

Word-spacing property ကတေသာ letter-spacing property နဲ့ နဲ့ ကွာခြားပါတယ် letter-spacing က စာလုံးတွေအားလုံး ကြားကို space တွေ လိုက်ခြားတာပါ word-spacing ကတေသာ အဲလို မဟုတ်ပါဘူး စကားစု တစ်စု နဲ့ တစ်စု ကြားကို ခြားပေးတာပါ။ Units တွေ အနေနဲ့ pixels , ems တို့ကို သုံးပြီး positive value တင်မက negative value တွေကိုပါ support လုပ်ပေးပါတယ်။

```

<head>
  <style>
    p{
      letter-spacing: 10px;
    }
    span{
      word-spacing: 100px;
    }
  </style>
</head>
<body>
  <p>This is letter spacing </p>
  <span>This is words spacing </span>

```

This is letter spacing

This is words spacing  
This is for Lower CASE

### Line Height

အထက်မှာ စကားလုံး တစ်လုံးကြား၊ စကားစုံ တစ်စုနဲ့ တစ်စုကြား spacing များကို မိမိတို့ လိုသလို control လုပ်နိုင်သလို စာကြောင်း တစ်ကြောင်းနဲ့ တစ်ကြောင်းကြားမှာ လည်း မိမိတို့ လိုသလို control လုပ်နိုင်ပါသေးတယ် ထို အတွက် property ကတော့ line-height ဖြစ်ပါတယ်။ Units အားဖြင့် percentage နဲ့ သတ်မှတ် နိုင်သလို pixel နဲ့လည်း သတ်မှတ် နိုင်ပါတယ်။ သတိပြုရင် အချက်မှာ line-height property အတွက် negative value တွေ သတ်မှတ် ပေးလို့ မရပါဘူး။

```
p{
    letter-spacing: 10px;
}
span{
    word-spacing: 100px;
    line-height: 100px;
}
```

This is letter spacing

This is words spacing

This is for Lower CASE

### Links

မိမိတို့ရဲ့ web page ကို user တွေ အသုံးပြုရ လွယ်ကူစေရန် သို့မဟုတ် အဆင်ပြုစေရန် user-friendly website တစ်ခု ဖြစ်စေရန် အတွက်ဆုံးလျှင် link တွေဟာလည်း အရေးပါပါတယ်။ ဥပမာ user တစ်ယောက်က Link တစ်ခုကို သွားပြီး ပြီလား မသွားရသေးဘူးဆုံးလာ သိနိုင် အတွက်ကို color လေးတွေ သုံးပြီး ဖော်ပြု ပေးတာမျိုး user က link တစ်ခုပေါ် Mouse ချွေးလိုက်ရင် color ပြောင်းပေးတာမျိုး စသဖြင့် ပြုလုပ်ပေးနိုင်ပါတယ်။ ထိုသို့ ပြုလုပ်ပေးနိုင်ဖို့ အတွက် link နဲ့ ပတ်သက်ပြီး state လေးခု ရှုပါတယ် link , visited , active and hover တို့ ဖြစ်ပါတယ်။ link ကတော့ normal link တစ်ခု ဖြစ်သလို unvisited link လည်း ဖြစ်ပါတယ်။ visited ကတော့ user က ယခု Link ကို သွားခဲ့ပြီးသားဆုံးပြီး ဖော်ပြုချင်တာပါ။ hover ကတော့ mouse pointer က link ပေါ်မှာ ရောက်လာတဲ့ အချိန် style တစ်ခုခု ပြန့် အသုံးပြု ပါတယ်။ active ကတော့ user က ယခု link ကို click လုပ်ပြီ ဆုံးတဲ့ အကြောင်းကို ပြပေးချင်တဲ့ အချိန်မှာ သုံးပါတယ်။

```

a:link{
    color: black;
}
a:hover{ /*mouse on link*/
    color: #rgb(0, 34, 128);
}
a:visited{
    color: #rgb(255, 0, 200);
}
a:active{
    color:#rgb(17, 17, 17);
}

```

သတိပြုရန် အချက်မှာအထက်ပါ အတိုင်း ရေးပြီး program ကို run ချိန်တွင်  
ကျွန်ုပ်သည့် value မှာ အလုပ်လုပ်သော်လည်း hover value သည် အလုပ် မလုပ်ပဲ  
ဖြစ်တတ်ပါတယ်။ ထိပြုသာနကို ဖြေရှင်းဖို့ အတွက် ဆိုလျှင် color များကို အောက်ပါ  
အတိုင်း ကွဲပြားစွာ ပေးဖို့လိုပါတယ်။

```

a:link {
    color: #1ebba3;
}
a:visited {
    color: #ff00f4;
}
a:hover {
    color: #11db43;
}
a:active {
    color: #15ff00;
}

```

အကယ်၍ မိမိတို့ရဲ့ link ကို button ပုံစံဖြင့် ပြလိုပါက အောက်ပါ အတိုင်း ရေးသား  
နိုင်ပါတယ်။

```

a:link, a:visited {
    color: white;
    background-color: #1e8ebb;
    display: inline-block;
    padding: 10px 20px;
    border: 2px solid #2b0999;
    text-decoration: none;
    text-align: center;
    font: 16px arial;
}
a:hover, a:active {
    background-color: #6350ce;
    border-color: #7443b6;
}

```

Display , padding properties တွေကိုတော့ နောက်သင်ခန်းတော့မှာ  
ဆွဲးနွဲးပေးသွားမှာ ဖြစ်ပါတယ်။

Lists

ယခု သင်ခန်းစာမျာတော့ CSS ကိုသုံးပြီး html list တွေကို style ပြုလုပ်သွားတာကို လေ့လာရမှာ ဖြစ်ပါတယ်။ ပထမဆုံး အနေဖြင့် list မှာ ordered list and unordered list ဆုံးပြီး နှစ်ခု ရှိပါတယ် ordered list မှာတော့ 1,2,3 စသေဖြင့် အခြားသော နည်းလမ်းများသုံးကာ နံပါတ်စဉ်များ ပါသော်လည်းပဲ unordered list မှာတော့ မပါ ပါဘူး။ lists များအကြောင်းကို html lesson တွင်သင်ကြားပေးခဲ့ပြီး html ထဲမှာပင် list တွေကို မိမိတို့စိတ်ကြိုက် style များပြုလုပ်နိုင် ကြောင်းကို ဆွဲးနွေးပေးခဲ့ပြီး ဖြစ်ပါတယ်။

```

        ul{
            list-style-type: disc;
        }
        ol{
            list-style: lower-alpha;
        }
    </style>
</head>
<body>
    <p>Unordered List</p>
    <ul>
        <li>Apple</li>
        <li>Orange</li>
        <li>Banana</li>
    </ul>
    <p>Ordered List</p>
    <ol>
        <li>Apple</li>
        <li>Orange</li>
        <li>Banana</li>
    </ol>

```

### List Markers

List markers ဆိုတာက list တွေရှေ့မှာ ရေးထားတဲ့ number or style တွေကို ဆိုလိုခြင်းဖြစ်ပြီး ထို number or style တွေကို list's data တွေနဲ့ အတူ အထဲမှာ ထည့်ရေးနိုင်သလို အပြင်မှာလည်း ရေးနိုင်ပါတယ်။ ထိုသို့ ရေးသားရာတွင် list-style-position ကိုသုံးပြီး inside or outside value များကို သုံးနိုင်ပါတယ်။

```

<style>
    ol li{
        background-color: #rgba(38, 185, 230, 0.548);
        padding: 5px;
        margin: 5px;
    }
    ol.inside li{
        list-style-position: inside;
    }
    ol.outside li{
        list-style-position: outside;
    }
</style>
</head>
<body>
    <p>Unordered List</p>
    <ol class="inside">
        <li>Apple</li>
        <li>Orange</li>
        <li>Banana</li>
    </ol>
    <p>Ordered List</p>
    <ol class="outside">
        <li>Apple</li>
        <li>Orange</li>
        <li>Banana</li>
    </ol>
</body>

```

## Unordered List

- 1. Apple
- 2. Orange
- 3. Banana

## Ordered List

1. Apple
2. Orange
3. Banana

အကယ်၍ list marker နေရမှာ ပုံတွေ ထည့်ချင်တယ် ဆိုလည်း ရပါတယ် သုံးရမယ့် properties ကတော့ list-style-image ဖြစ်ပြီး value အနေဖြင့် url function ကိုသုံးရပါမည် ထိ function ထဲတွင် မိမိတို့ ထည့်လိုသည့် image url ကိုထည့်ပေးရပါမည်။

`list-style-image: url("images/myimage.jpg");`

**List-style**

List-style properties နဲ့ ပတ်သက်ပြီး properties လေးခု ရှိပါတယ် list-style-type , list-style-image နှင့် list-style-position တို့ ဖြစ်ပါတယ် အကယ်၍ ထိုသုံးခုကို မသုံးလိုလျှင် list-style ဆိုသည့် properties တစ်ခုတည်းကို သုံးပြီး သုံးခုလုံးစာ အတွက် ရေးနိုင်ပါတယ်။

```

    ul{
      | list-style: square inside url("listphoto.png");
    }
  </style>
</head>
<body>
  <ul>
    <li>Apple</li>
    <li>Orange</li>
    <li>Banana</li>
  </ul>

```

## Navigation Menus

ယခု သင်ခန်းစာမျာတော့ nav element ကိုအသုံးပြုပြီး navigation menus တစ်ခုကို ရေးသား သွားမှာ ဖြစ်ပါတယ်။ program ရေးသားပုံကိုတော့ အောက်မှာ ဖော်ပြထားပါတယ်။ အောက်ပါ သင်ခန်းစာတွင် display ကိုသုံးထားသည့် အတွက် display အကြောင်း အနည်းငယ်ကို အရင်ဦးစွာ ဖော်ပြပါမည် display properties နောက်တွင် ပုံမှန်အားဖြင့် value လေးခု လိုက်ပါသည်။ ပထမ တစ်ခုသည် inline ဖြစ်ပြီး line တစ်ခု တည်းမှာတင် မိမိတို့ ဖော်ပြချင်တဲ့ အချက်လက်တွေကို ဖော်ပြ ပေးပါတယ်။ ဒုတိယ တစ်ခုသည် inline-block ဖြစ်ပြီး သူသည်လည်း Line တစ်ခု တည်းမှာတင် မိမိတို့ ဖော်ပြလိုတဲ့ အချက်လက်တွေကို ဖော်ပြ နိုင်ပါတယ် သို့သော် ထို block ကို ပေးထားတဲ့ width and height properties value များအတိုင်း block size ကို ပြောင်းလဲ နိုင်ပါတယ်။ နောက်ဆုံး တစ်ခု ဖြစ်တဲ့ block ကတော့ နောက် တစ်ကြောင်းကို ဆင်းသွားပါတယ်။

## Display

Elements တွေ အားလုံးအတွက် display properties ရဲ့ value တွေကို default အတိုင်း ထားပေး ထားပါတယ် ဆုံးလိုချင်တာ div element ဆုံးလျှင် display သည် block ပုံစံ ဖြစ်ပါတယ် ဆုံးလိုချင်တာက အောက်တစ်ကြောင်းဆင်းသွားပြီး horizontal တစ်ခုလုံးကို သူအပိုင်းသိမ်းလိုက် ပါတယ် span element ကတော့ ထိုကဲ့သို့ မဟုတ်ပါဘူး block လုပ်သွားတာမျိုး မဟုတ်ပဲ သူနဲ့ သက်ဆိုင်သလောက် နေရာကို horizontal line တစ်ခုတည်းမှာ နေရာ ယူပါတယ်။

## Display Block

Display properties မှ block value သည် သူနဲ့ သက်ဆိုင်ရာ select လုပ်ခံရတဲ့ element ကို div, p element တိုကဲ့သို့ block ပုံစံ ဖန်တီး ပစ်လိုက်ပါတယ်။ အောက်ပါ program တွင် ပုံမှန်အားဖြင့် block level မဟုတ်သည့် element များအား block value ကို သုံးကာ block ဖြစ်အောင် ဖန်တီးပါမည်။

```

<style>
  span{
    display: block;
    background-color: #rgb(120, 128, 126);
  }
  a{
    display: block;
    background-color: #rgb(127, 202, 104);
  }
</style>
</head>
<body>
  <span>Hello this is span</span>
  <a href="#">link</a>
</body>

```

### Display inline

Inline value သည် element များအား inline-level ကဲသို့ လုပ်ဆောင်ပေးပါသည် ဥပမာ span and a စတဲ့ element များကဲသို့ဖြစ်ပါသည်။ ပုံမှန် အားဖြင့် ပါထွင် ရှိသည့် list များသည် block-level ဖြင့် ဖော်ပြကပါသည် ထို့ list များအား Inline ပုံစံဖြင့် အောက်ပါ အတိုင်းရေးသားနိုင်ပါသည်။

```

  ul li{
    display: inline;
  }
</style>
</head>
<body>
  <ul>
    <li>programming</li>
    <li>networking</li>
    <li>hacking</li>
  </ul>

```

programming networking hacking

### Display inline-block

Display value ရဲ့ inline-block value ဟာ သူနဲ့ သက်ဆိုင်ရာ element တွေကို block box ပုံစံသို့ ပြောင်းလဲ ပစ်လိုက်ပါတယ် ဥပမာ အောက်ပါ program တွင် block level ဖြစ်တဲ့ div ထဲသို့ span ကို ထည့်ပြီး inline-block box ပုံစံ ဖန်တီးထားပါတယ်။

```


display: inline-block;
    background-color: #rgb(105, 227, 236);
    padding: 10px;
}
span{
    display: inline-block;
    background-color: #rgb(182, 95, 95);
    padding: 10px;
}


```

</style>

</head>

<body>

<div>

<span>

This is block-box

</span>

</div>

### Output



This is block-box

### Padding

Padding properties ကတေသူ border နှင့် content ကြားမှာ ရှိတဲ့ spacing ကို ပြုလုပ်ရာ တွင် အသုံးပြုပါတယ်။ element မှာ background color သတ်မှတ်ပေးထားမည့်အို လျင် ထို color သည် padding မှာလည်း effect ဖြစ်သွားမှပါ။ padding properties မှ value များကို top , right ,bottom and left စသည့် sides လေးခုလုံးကို သတ်မှတ်ပေးနိုင်ပါတယ်။ properties တွေကတေသူ padding-top , padding-right , padding-bottom နှင့် padding-left တို့ ဖြစ်ပါတယ်။

```

<style>
    h1{
        padding-top: 50px;
        background-color: #rgb(149, 170, 169);
    }
</style>
```

</head>

<body>

<h1>Padding Lesson</h1>

</body>

အထက်ပါ program ကတေသူ padding ကို top တစ်ခုတည်း ယူထားတာပါ။ ထို့ကြောင့် output အနေဖြင့်အောက်ပါ အတိုင်း မြင်ရပါမည်။

## Padding Lesson

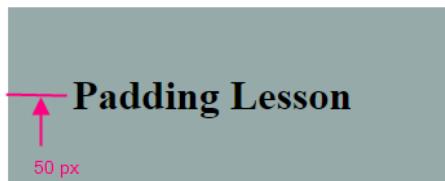
### Padding Lesson

Padding-bottom ကိုတော့အောက်ပါ program မှာ ဖော်ပြထားပါတယ်။

```
h1{  
    padding-top: 50px;  
    padding-bottom: 50px;  
    background-color: ■rgb(149, 170, 169);  
}
```



Padding-left ကိုတော့အောက်မှာ ဖော်ပြထားပါတယ်။



```
h1{  
    padding-top: 50px;  
    padding-bottom: 50px;  
    padding-left: 50px;  
    background-color: ■rgb(149, 170, 169);  
}
```

Padding-right ကိုတော့အောက်မှာ ဖော်ပြထားပါတယ်။

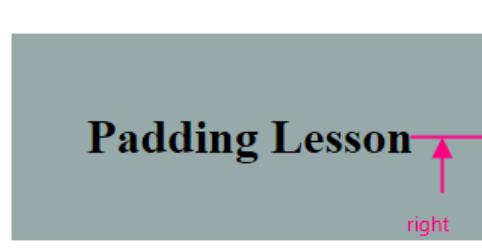
```
padding-top: 50px;  
padding-bottom: 50px;  
padding-left: 50px;  
padding-right: 50px;  
background-color: ■rgb(149, 170, 169);  
}
```

အထက်ပါ အတိုင်း padding-right ကို ထည့်လိုက်သော်လည်း ပုံစံက မပြောင်းသေးပါဘူး ဘာကြောင့်လည်း ဆိုတော့ block-level ဖြစ်နေလို့။



Inline-level ဖြစ်စေရန် display properties ကို အောက်ပါ အတိုင်း ရေးနိုင်ပါတယ်။

```
h1{
    padding-top: 50px;
    padding-bottom: 50px;
    padding-left: 50px;
    padding-right: 50px;
    background-color: #rgb(149, 170, 169);
    display: inline-block;
}
```



သတိပြုရန် အချက်မှာ padding ဆိုသည့် properties တစ်ခုတည်းကို သုံးပြီးတော့လည်း value တွေကို shorthand နည်းဖြင့် ဖော်ပြနိုင်ပါတယ်။

`padding: 50px;` ယခု အတိုင်း ရေးသားလိုက်မည် ဆိုလျှင် top , bottom , left ,right စသည့် နေရာ အားလုံးကို 50px apply လုပ်ပေးသွားမှာ ဖြစ်ပါတယ်။ ဥပမာ နောက် တစ်ခု အနေဖြင့် အောက်ပါ အတိုင်း ရေးသား လိုက်မည်ဆိုလျှင် vertical and horizontal အတိုင်း apply လုပ် သွားမှာ ဖြစ်ပါတယ်။

```
h1{
    padding: 25px 100px;
    background-color: #rgb(149, 170, 169);
    display: inline-block;
}
```



`padding: 25px 50px 75px; /* top | horizontal | bottom */`

`padding: 25px 50px 75px 100px; /* top | right | bottom | left */`

Value တစ်ခုတည်းသာ ထည့်ပေးလျှင် sides လေးခုလုံးကို apply လုပ်ပြီး နှစ်ခုဆိုလျှင် top and bottom sides တွေကို apply လုပ်ပါတယ်။ ထိုနည်းတူ values 3 ခု ထည့်ပေးရင်တော့ top , right and left , bottom တို့ကို apply လုပ်ပါတယ်။ Values လေးခု ထည့်ပေးရင်တော့ top , right ,bottom , left လေးကို အစဉ်လိုက် apply လုပ်ပါတယ်။ အထက်ပါ အတိုင်း shorthand နည်းလမ်းများကို သုံးခြင့် အားဖြင့် code တွေအများကြီး မရေးရတော့ သလို အချိန်ကုန်တာ လည်း သက်သာ ပေါ်တယ်။

Units တွေကို သတ်မှတ် ပေးတဲ့အ ခါမှာ px တွေ တင် မဟုတ်ပဲ px ,em ,rem , pt , cm , % , inherit တို့ဖြင့်လည်း သတ်မှတ်နိုင်ပါတယ်။ အောက်မှာတော့ padding and display တို့ကို naviagtion program လေး တစ်ခု ရေးပြထားပါတယ်။

```

<style>
    body{
        font-size: 16px;
        font-family: Arial, Helvetica, sans-serif;
    }
    ul{
        padding:0;
        list-style: none;
        background-color: #rgb(67, 195, 174);
    }
    ul li{
        display: inline-block;
    }
    ul li a{
        display: block;
        padding: 15px 30px;
        color: #rgb(238, 246, 246);
        text-decoration: none;
    }
    ul li a:hover{
        color: black;
        background-color: #rgb(212, 206, 219);
    }
</style>
</head>
<body>
    <nav>
        <ul>
            <li><a href="#">Home</a></li>
            <li><a href="#">Hacking</a></li>
            <li><a href="#">Programming</a></li>
            <li><a href="#">IoT</a></li>
            <li><a href="#">ComputerScience</a></li>
        </ul>
    </nav>

```

## Tables with CSS

HTML နဲ့တည်ဆောက်ထားတဲ့ table တွေကို မိမိတို့ လိုချင်သည့် အတိုင်း လုပွဲ ဖန်တီးနိုင်ဖို့ CSS မှာ properties များစွာကို support လုပ်ပေးထားပါတယ်။ပထမဆုံး အင်ဖြင့် table တစ်ခု တည်ဆောက်ပြီး ထို table မှ head ပိုင်းတွင် ID , First name , Last

Name , Email တိုကို ထည့်ထားပါမည် ထိန္ဒေက် tr tag ထဲတွင် td tag ကို ထပ်သံးကာ table data များကို ဖြည့်ထားပါမည်။

```



```

ID	Name	Ph number	Email
1	WinHtut	0990000009	winhtutonline@gmail.com
2	Htut	0990000009	winhtutonline@gmail.com
3	Win	0990000009	winhtutonline@gmail.com
4	VijjA	0990000009	winhtutonline@gmail.com

```

<table>
    <tr>
        <th>ID</th>
        <th>Name</th>
        <th>Ph number</th>
        <th>Email</th>
    </tr>
    <tr>
        <td>1</td>
        <td>WinHtut</td>
        <td>0990000009</td>
        <td>winhtutonline@gmail.com </td>
    </tr>
    <tr>
        <td>2</td>
        <td>Htut</td>
        <td>0990000009</td>
        <td>winhtutonline@gmail.com </td>
    </tr>
    <tr>
        <td>3</td>
        <td>Win</td>
        <td>0990000009</td>
        <td>winhtutonline@gmail.com </td>
    </tr>
    <tr>
        <td>4</td>
        <td>VijjA</td>
        <td>0990000009</td>
        <td>winhtutonline@gmail.com </td>
    </tr>
</table>

```

အထက်ပါ program မှာတော့ table ရဲ့ border စွဲ style ကို နှစ်ထပ် separate ပုံစံဖြင့် ဖော်ပြထားပါတယ်။ အကယ်၍ ထိကဲသို့ပုံ မလိချင်ပါက border-collapse ဆိတ့ property

ကိုသုံးပြီး value အနေဖြင့် collapse ကို သုံးနိုင်ပါတယ်။

ID	Name	Ph number	Email
1	WinHtut	099000009	winhtutonline@gmail.com
2	Htut	099000009	winhtutonline@gmail.com
3	Win	099000009	winhtutonline@gmail.com
4	VijjA	099000009	winhtutonline@gmail.com

```
table{
    border-collapse: collapse;
}
```

### Adjusting Space

Table header နှင့် table data တို့ကလည်း spacing တွေ မိမိတို့လိုသလို ပြောင်းလဲ နိုင်ပါသေးတယ်။

```
<style>
    table,th,td{
        border: 2px solid #rgb(80, 188, 207);
    }
    table{
        border-collapse: collapse;
    }
    th,td{
        padding: 10px;
    }
</style>
```

ID	Name	Ph number	Email
1	WinHtut	099000009	winhtutonline@gmail.com
2	Htut	099000009	winhtutonline@gmail.com
3	Win	099000009	winhtutonline@gmail.com
4	VijjA	099000009	winhtutonline@gmail.com

ထိုပြင် border တစ်ခုနဲ့တစ်ခုကြားမှာလည်း spacing တွေကို သတ်မှတ် ပေးနိုင်ပါသေးတယ် ထိုသို့ သတ်မှတ် ပေးနိုင်ဖို့ border-spacing properties ကိုသုံးပြီး value အနေဖြင့် မိမိတို့ အဆင်ပြုသည့် ၁၃ ကိုထည့်နိုင်ပါသည်။ တရေးသူ အနေဖြင့် အောက်ပါ table တွင် border တစ်ခုနဲ့တစ်ခုကြား 10 px ကိုသုံးထားပါသည်။

```
<style>
  table,th,td{
    border: 2px solid #rgb(80, 188, 207);
  }
  table{
    border-spacing: 10px;
  }
  th,td{
    padding: 10px;
  }
</style>
```

ID	Name	Ph number	Email
1	WinHtut	099000009	winhtutonline@gmail.com
2	Htut	099000009	winhtutonline@gmail.com
3	Win	099000009	winhtutonline@gmail.com
4	Vijja	099000009	winhtutonline@gmail.com

အထက်ပါ လုပ်ဆောင် မှူးများ အပြင် table ကို width and height တို့ ထည့်ပေးပြီး values များ သတ်မှတ် ပေးနိုင်ပါသေးတယ်။

```
<style>
  table,th,td{
    border: 2px solid #rgb(80, 188, 207);
  }
  table{
    width: 100%;
    border-spacing: 10px;
    color: #rgb(209, 27, 139);
    background-color: #rgb(226, 243, 245);
  }
  th,td{
    height: 50px;
    padding: 10px;
  }
</style>
```

ID	Name	Ph number	Email
1	WinHtut	099000009	winhtutonline@gmail.com
2	Htut	099000009	winhtutonline@gmail.com
3	Win	099000009	winhtutonline@gmail.com
4	VijjA	099000009	winhtutonline@gmail.com

နောက်ဆုံး အနေဖြင့် table body ပိုင်းမှ row များကို တစ်ခုနဲ့ တစ်ခု background color မတူအောင် ပြောင်းလဲ နိုင်ပါတယ်။ ထိုသို့ ပြောင်းလဲ ခြင်းဖြင့် များပြားလှတဲ့ data တွေကို ဖတ်ရတာ လွယ်ကူ ပေါ်ပါတယ် ထိုနည်းလမ်းကိုတော့ zebra-striping လိုလည်း ခေါ်ပါတယ်။ အသုံးပြုရမည့် function သည် nth-child() ဖြစ်ပြီး pseudo-class selector လို့ ခေါ်ပါတယ်။ parameter အနေဖြင့် odd or even စသေဖြင့် မိမိတို့ စိတ်ကြိုက် ထည့်နိုင်ပါတယ်။ အကယ်၍ odd ကိုထည့်ပေးလိုက်မည် ဆိုလျှင် 1 , 3 , 5 .. စသေဖြင့် color တွေ ချိန်းသွားမှာ ဖြစ်ပါတယ်။ even ဆိုရင်တော့ 2,4, 6 .... စသေဖြင့် အစဉ်လိုက် color တွေ ချိန်းသွားမှာပါ။

```

<style>
  table,th,td{
    border: 2px solid #rgb(80, 188, 207);
  }
  table{
    width: 100%;
    border-spacing: 10px;
    color: #rgb(209, 27, 139);
    background-color: #rgb(226, 243, 245);
  }
  th,td{
    height: 50px;
    padding: 10px;
  }
  tbody tr:nth-child(odd){
    background-color: #rgb(195, 196, 144);
  }
</style>

```

အထက်ပါ program တွင် tbody ကို သုံးထားသည့် အတွက် table head ပိုင်းမှ လွှဲ၍ ကျော် tr များအားလုံးကို tbody ထဲတွင် ထည့်ပေးရပါမည်။

```

-----+
<table>
  <thead>
    <tr>
      <th>ID</th>
      <th>Name</th>
      <th>Ph number</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>WinHtut</td>
      <td>099000009</td>
      <td>winhtutonline@gmail.com </td>
    </tr>
    <tr>
      <td>2</td>
      <td>Htut</td>
      <td>099000009</td>
      <td>winhtutonline@gmail.com </td>
    </tr>
    <tr>
      <td>3</td>
      <td>Win</td>
      <td>099000009</td>
      <td>winhtutonline@gmail.com </td>
    </tr>
    <tr>
      <td>4</td>
      <td>VijjA</td>
      <td>099000009</td>
      <td>winhtutonline@gmail.com </td>
    </tr>
  </tbody>
</table>

```

### Table responsive

နောက်ဆုံး အနေဖြင့် table ကို responsive ပြုလုပ်ဖို့ လိုအပ်ပါတယ် ဘာကြောင့်လည်း ဆုံးတော့ small screen တွေမှာ အဆင်ပြုစေဖို့ အတွက်ပါ။ ထိုသို့ responsive ဖြစ်ဖို့ အတွက် css property ကို overflow-x: auto ဟု သုံးရန် လိုအပ်ပါသည်။ ထိုပြင် မိမိတို့ ရဲ့ table တစ်ခု လုံးကို div ထဲတွင် ထည့်ပြီး ထို div အား class တစ်ခု သတ်မှတ် ပေးကာ ထို class ကို css style rules အသုံးပြု ပေးရပါမည်။

```

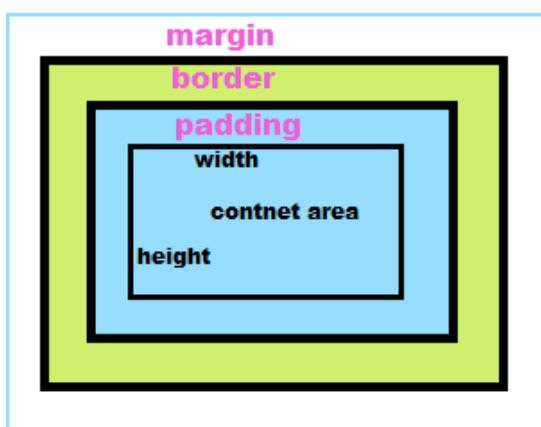
        tbody tr:nth-child(odd){
            background-color: #rgb(195, 196, 144);
        }

    </style>
</head>
<body>
    <div class="responsive"></div>
    <table>
        table data
    </table>
</div>

```

### BOX MODEL

Elements တွေကို web page ပေါ်မှာ ဖော်ပြရာတွင် rectangular boxes ပုံစံဖြင့် ဖော်ပြလေလာ ရှိပါတယ်။ CSS box model တွေကနေ တစ်ခင့် rectangular boxes တွေကို web page ပေါ်မှာ ဘယ်လို ပုံစံဖြင့် တည်ရှိရမလဲ ဆိုတာကို ဆုံးဖြတ် ပေးပါတယ်။ Boxes တစ်ခုစီတိုင်းမှာ မတူညီတဲ့ properties တွေ များစွာ ရှိနိုင်ပါတယ် သို့သော် box တစ်ခုမှာတော့ သေချာပေါက် information တွေ ဖော်ပြပို့ content area ရှိရပါတယ် ထို content area ဘေးမှာမှ padding , border နှင့် margin areas တို့ တည်ရှိပါတယ်။ အောက်မှာ diagram တစ်ခုကို ဖော်ပြ ထားပါတယ် ထို diagram သည် box model တစ်ခုကို ဘယ်လို တည်ဆောက်ထားသလဲ ဆိုတာ ဖော်ပြထားပြီး css properties တွေဖြစ်တဲ့ width , height , padding , border နှင့် margin စတဲ့ values တွေကို ဘယ်လိုအသုံးချေသလဲ ဆိုတာပါ ဖော်ပြထားပါတယ်။

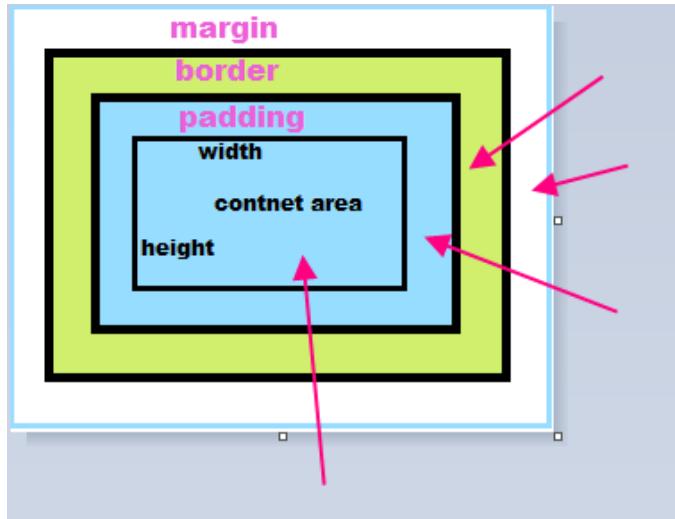


padding သည် element ခဲ့ content နဲ့ border

ကြားမှာ ရှိတဲ့ နေရာ တစ်ခု ဖြစ်ပါသည်။ margin ကတော့ border ခဲ့ အပြင်က နေရာ တစ်ခု ဖြစ်ပါတယ်။ element ခဲ့ color အတိုင်း padding နေရာမှာလည်း color ပေါ်နေမှာပါ ထို့ကြောင့် padding နဲ့ content area နဲ့သည် စပ်ဆက် နေပါသည်။ သို့သော် margin area မှာတော့ များသော အားဖြင့် transparent ဖြစ်နေမှာ ဖြစ်ပြီး element ခဲ့ background color ကင်း affect ဖြစ်နေမှာ မဟုတ်ပါဘူး။ border ကတော့ margin နဲ့ padding ကြားမှာ ရှိနေပြီး

သူကိုတော့ မိမိတို့စိတ်ကြိုက် colorထားပေးနိုင်သလို px ကိုလည်း ထားနိုင်ပါတယ်။

ဒါဆိုရင် box တစ်ခုရဲ့ size ဖြစ်တဲ့ width and height ကို ဘယ်လို သိနိုင်မလဲ။ box တစ်ခုရဲ့ width ကို လိုချင်ရင် element width + padding-left + padding-right + border-left + border-right + margin-left + margin-right စသည်တို့ ပေါင်းလိုက်လျှင် box တစ်ခုရဲ့ width ကို ရပါမည်။

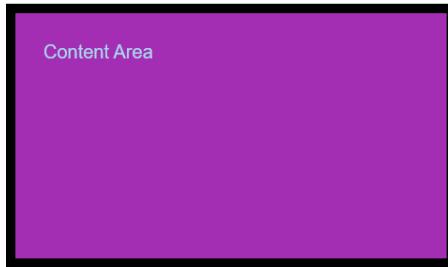


Box တစ်ခုရဲ့ height ကို လိုချင်ရင်လည်း ထိုနည်း အတိုင်းပါပဲ element height + padding-top + padding-bottom + border-top + border-bottom + margin-top + margin-bottom စသည်တို့ကို ပေါင်းခြင်းဖြင့် height ကို ရနိုင်ပါတယ်။ ဒီလောက်ဆိုရင် box တစ်ခုရဲ့ ဖွံ့ဖြိုးပုံကို နားလည် သွားလိမ့်မယ်လို့ မျှော်လင့်ပါတယ်။ အောက်မှာတော့ box model တစ်ခုကို တည်ဆောက် ပြထားပါတယ်။ margin အကြောင်းကိုတော့ နောက် lesson မှာ ဆေးနွေးပါမည်။

```

<style>
div{
    width: 400px;
    height: 200px;
    padding: 30px;
    border: 10px solid black;
    margin: 20px auto;
    color: #rgb(171, 217, 238);
    background-color: #rgb(163, 45, 179);
}
</style>
</head>
<body>
<div>Content Area</div>

```



## Margin

Element box တွေရဲ့ ဘေးပတ်လည်မှာ spacing တွေထားဖို့ အတွက် margin property ကို အသုံးပြုပါတယ်။ element တွေရဲ့ margin သည် ထို element တွေရဲ့ background color ဖြင့် affected မဖြစ်ပါဘူး။ margin ကို ပိုမို နားလည် စေရန် side တစ်ခုချင်းစီရဲ့ margin တွေကို မြိမ်တို့ လုံသလို ကတားပြီး ပြောင်းလဲပါမည်။



```

<style>
    body{
        background-color: ■rgb(1, 14, 1);
    }
    h1{
        width: 100%;
        height: 50px;
        margin: 20px auto;
        padding-left: 700px;
        background-color: □rgb(194, 165, 210);
    }
    h3{
        margin-top: 30px;
        margin-left: 350px;
        background-color: □rgb(184, 148, 210);
        margin-right: 350px;
        height: 100px;
        color: ■green;
    }
</style>
</head>
<body>
    <h1>Navigation Bar</h1>
    <h3>Contents Place</h3>

```

Margin မှာလည်း margin-top , margin-right , margin-bottom , margin-left စိတ်

properties တွေ ရှိပါတယ်။ Margin ခဲ့ values တွေကို ဖော်ပြ တဲ့ နေရာမှာ length , % , auto , inherit စတာတွေနဲ့ သတ်မှတ်ပါတယ်။ auto ဆိုတာကတော့ browser က သူဘာသာ အဆင်ပြောမယ့် ပုံစံ တစ်ခုကို ဖန်တီးပေးတာပါ။ ပုံမှန် အားဖြင့်တော့ element ကို အလယ်ကို ပိုမို သုံးပါတယ်။

### Margin Shorthand

Margin တွေကိုလည်း shorthand ပုံစံဖြင့် အသုံးပြု နိုင်ပါသေးတယ်။ ဥပမာ margin: 30px လို့ ရေးမည် ဆိုလျှင် top , bottom , left , right စတဲ့ sides လေးခုလုံးကို အသုံးပြုမယ်လို့ ဆိုလိုတာပါ။

Margin: 25px 25px ဆိုလျှင် vertical and horizontal ဖြစ်ပါတယ်။

Margin: 25px 25px 25px ဆိုလျှင် top , horizontal , and bottom တို့ ဖြစ်ပါတယ်။

Margin: 25px 25px 25px 25px ဆိုလျှင် top , right , bottom , left တို့ ဖြစ်ပါတယ်။

```

<style>
  div{
    width: 500px;
    height: 250px;
    background-color: green;
    margin:0px auto;
  }
</style>
</head>
<body>
  <div>
    <h1>Navigation Bar</h1>
    <h3>Contents Place</h3>
  </div>

```

## Navigation Bar

Contents Place

### Dimension

Element တွေရဲ့ width and height တွေ သတ်မှတ် ပေးလိုတဲ့အ ခါမှာ css ကို အသုံးပြုပါတယ်။ Dimension properties တွေကတော့ width , height , max-width , min-width , max-height နှင့် min-height တို့ ဖြစ်ပါတယ်။ Width and height ကတော့ ရှေ့ပိုင်း သင်ခန်းစာ တွေမှာလည်း ဆွေးနွေး ခဲ့ပြီး ဖြစ်သလို length , % , auto , initial နှင့် inherit တို့ကို အသုံးပြု နိုင်ပါတယ်။ length ဆိုတာက px , em , rem , pt , cm စသည်တို့ ကို ဆိုလိုတာပါ။ initial ကတော့ width and height ကို default value နဲ့ ဖော်ပြ ပေးတာပါ။ inherit

ကတေသာ သူရဲ့ parent element ကနေတစ်ဆင့် ရယူ လိုက်တာပါ။

Maximum width and height ကတေသာ content ရဲ့ area ကို သတ်မှတ်ပေးဖို့ သုံးပါတယ် သတိပြုရန် အချက်သည် ထို maximum width and height ထဲမှာ padding, border and margin တို့ မပါ ပါဘူး။ element တွေရဲ့ size သည် max-width value တွေ အပေါ်မှာသာ မူတည်ပြီး max-width တွေထက် ကျော်လို့မရပါဘူး။ ဥပမာ width ကို 300px လို့ သတ်မှတ်ထားရင် max-width value သည် 300px ကို ကျော်ပြီး ပေးလို့ မရပါဘူး။ အောက်မှာ sample program ဖြင့် နှိုင်းယူဉ်ပြထားပါတယ်။

```



```



အထက်ပါ ပုံတွင် div tag သည် width ကို 300px ဟုပေးထားပါသည် ထိုနည်းတူ max-width ကို 280px ဟု ပေးထားပါသည်။ အကယ်၍ max-width ကို 310 ဟု ပြောင်းပေးလျှင် ရမည် မဟုတ်ပါ အဘယ်ကြောင့် ဆိုသော် max-width ကို မူလ သတ်မှတ်ထားသည့် width ထက် ကျော်လွန်းပြီး ပေးလို့ မရလိုပါ။

Max-height သည်လည်း ထိုနည်းတူပါပဲ သူ့ကို သတ်မှ တ် ပေးထားတဲ့ height value ထက် မကျော်လွန်ပါဘူး ထိုပြင် height ရဲ့ value ကို 300px ဟု ပေးထားပြီး max-height ရဲ့ value ကို 150px ဟု ပေးထားခဲ့မည်ဆိုလျှင် ထို element ရဲ့ value သည်လည်း 150px သာ ဖြစ်ပါမည်။ ထိုကြောင့် value သည် အမြင့်ဆုံးကို သတ်မှတ် ပေးထားရှုသာ ဖြစ်ပြီး max value ကတေသာ အမှန်တကယ် သတ်မှတ် ပေးမည့် value တစ်ခု ဖြစ်ပါသည်။

### Minimum Width and Height

Min-width and min-height စာတဲ့ properties တွေကတေသာ content area ရဲ့ minimum width and height တို့ကို သတ်မှတ် ပေးပါတယ်။ minwidth and height မှာလည်း

padding , borders and margins တို့ မပါဝင်ပါဘူး။ Minimum and maximum က အသုံးပြုပုံချင်း မတူသလို ရေးသားပုံချင်းလည်းကွာပါ ဥပမာ width ကို 200px လိုပေးထားသော်လည်းပဲ min-width ကို 400px ဆိုပြီး ပေးထားလိုခုပါတယ်။ content area ကိုလည်း min-width ရဲ့ value အတိုင်းပဲ အသုံးပြုသွားမှာပါ။

```



```

min တွင် မူရင်းထက် ကျော်လို့ရသည်

နှင့်ယူဉ် ရန် ဖြစ်သည်

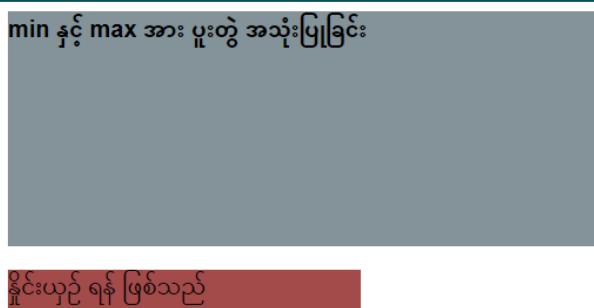
Min-height သည်လည်း ထိုနည်း အတိုင်းပင် ဖြစ်ပါတယ်။ မူရင်း value ထက် ကျော်လွှာနိုင်သလို မူရင်းထက် သည် content area အတွက် အသုံးပြုမည့် value ဖြစ်ပါတယ်။

Min-width and min-height properties ကိုတော့ max-width and max-height value တွေနဲ့ ပေါင်းပြီး အသုံးပြုပါတယ် ဥပမာ box တစ်ခု ဖန်တီးလိုတဲ့ အခါမျိုးတွင် min-width and max-height တို့ကို သုံးနိုင်ပြီး မိမိတို့ စိတ်ကြိုက် min and max တို့ကို သတ်မှတ်ပေးထားနိုင်ပါတယ်။

```

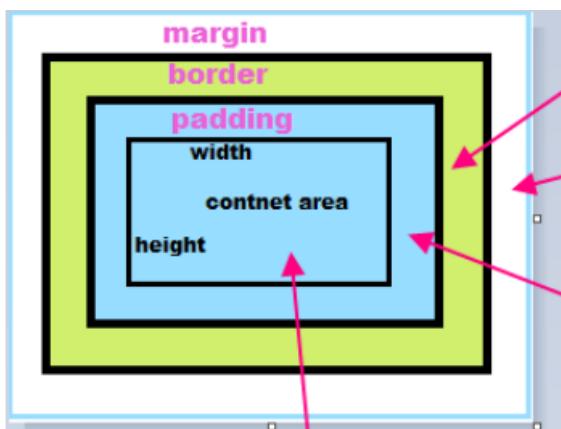
        div{
            min-width: 400px;
            max-width: 500px;
            height: 200px;
            background-color: #rgb(131, 147, 153);
        }
        p{
            width: 300px;
            background-color: #rgb(163, 74, 74);
        }
    </style>
</head>
<body>
    <div><b>min နှင့် max အား ပူးတဲ့ အသုံးပြုခြင်း</b></div>
    <p>နိုင်းယူး ရန် ဖြစ်သည်</p>

```



## Border

Border property ကို ဘယ်နေရာမှာ သုံးလဲ ဆိုတော့ element တစ်ခုရဲ့ border နေရာတွေကို မိမိတို့ သတ်မှတ်ပေးချင်တဲ့ အချိန်မှာ သုံးနိုင်ပါတယ်။ ရှေ့ပိုင်း သင်ခန်းစာတွေများ ဆွေးနွေးခဲ့သလိုပဲ border ရဲ့ နေရာသည် padding and margin ကြားထဲမှာ ဖြစ်ပါတယ်။ border တွေရဲ့ style သည် solid line , dotted line , double line or image စသည်တို့ ဖြစ်နိုင်ပါတယ်။



## Border Styles

Border တွေကိုလည်း style အမျိုးမျိုးသုံးနိုင်ဖို့ CSS က support လုပ်ထားပါတယ် border-style ရဲ့ value တွေကတော့ none , hidden , solid , dashed , dotted , double , inset ,

outset ,groove နဲ့ ridge တို့ ဖြစ်ပါတယ်။ none and hidden value တွေကတော့ no border ပါ။ inset , outset , groove and ride value တွေကိုတော့ 3D ပုံစံမျိုးဖြင့် ဖော်ပြ လိုတဲ့ အခါတွေမှာ အသုံးပြုပြီး border-color နဲ့ လည်း သက်ဆိုင်နေပါတယ်။



```
<style>
p {
    width: 500px;
    display: inline-block;
    border-width: 20px;
    background: #rgb(143, 206, 107);
    padding: 20px;
    margin: 20px;
}

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

```
<body>
<p class="none">No border Style.</p>
<p class="hidden">Hidden border Style.</p>
<p class="dotted">Dotted border Style.</p>
<p class="dashed">Dashed border Style.</p>
<p class="solid">Solid border Style.</p>
<p class="double">Double border Style.</p>
<p class="inset">Inset border Style.</p>
<p class="outset">Outset border Style.</p>
<p class="groove">Groove border Style.</p>
<p class="ridge">Ridge border Style.</p>
</body>
```

## Border Width

Border-width property වෙති



## JavaScript

JavaScript သည် scripting programming language တစ်ခု ဖြစ်ပါတယ်။ JavaScript နဲ့ ဘာတွေ လုပ်လိုရသလဲ သူက ဘာတွေလုပ်ဆောင် ပေးသလဲ။ Web pages တွေပေါ်မှာ မိမိလုပ်ဆောင် လိုသော လုပ်ငန်းများ သို့မဟုတ် ပေါင်းခြင်း နှစ်ခြင်း စတဲ့ arithmetic လုပ်ခြင်းများ ထံပြင် database ပေါ်သူ့ data တွင်သွင်းခြင်းများ database မှ data ထုတ်ယူခြင်းများ အပြင် အခြားသော ရှုပ်ထွေးသည့် process များကိုလုပ်ဆောင်

နိုင်ပါတယ်။ ဥပမာ webpages များပေါ်တွင် seconds နှင့်အမျှ ပြောင်းလဲ နေသော data များကို database မှ ရယူပြီး ဖော်ပြခြင်းများ 2D/3D စုတဲ့ games များကို JavaScript ကိုအသုံးပြုပြီး ဖန်တီးနိုင်ပါတယ်။ JavaScript webpages များရဲ့ third layer ဖြစ်သလို ပထမ layer နှင့် ဒုတိယ layer တို့တွင် html နှင့် css တို့ရှိကြသည်။ တို့ကြောင့် JavaScript ကို လေ့လာမည်ဆိုလျှင် html နှင့် css တို့ကို basic level အနေဖြင့် ကြိုတင် သံထားသင့်သည်။

ဥပမာ အားဖြင့် JavaScript ရဲ့ လုပ်ဆောင်ပုံကို ဖော်ပြ ရမည်ဆိုလျှင် user မှ input အနေဖြင့် number နှစ်ခုကို ထည့်လိုက်သည့် ဆိုပါစို့ ထို number နှစ်ခုအား ပေါင်းပြီး output ကို ပြပေး စေလိုလျှင် ထို number နှစ် ပေါင်းသည့် နေရာနှင့် ပြန်လည် ပြပေးသည့် နေရာ၏ JavaScript ကို အသုံးပြုရမည်။ JavaScript ကို မိမိတို့ရဲ့ html file ထဲမှာ add မည်ဆိုလျှင် နည်းလမ်းနှစ်ခုနှင့် add နိုင်ပါတယ်။ ပထမ တစ်ခုသည် internal javascript ဖြစ်ပြီး ဒုတိယ တစ်ခုသည် external javascript ဖြစ်ပါတယ်။ Internal javascript မှာတော့ html file ထဲတွင် script tag ကိုသုံးပြီး add ပါတယ်။ ထို script tag ထဲမှာမှ javascript code များကို ထည့်ရေးပါတယ်။ External javascript မှာတော့ .js ဖြင့် javascript file တစ်ခု သီးသန် ဆောက်ပြီး ထို file ထဲမှာမှ source code များကို ရေးသားပါတယ် ထို့နောက် ထို js file အား html file ထဲမှာ head tag ထဲတွင် script tag ကို သုံးပြီး html file နှင့် javascript file ကို တွဲပါတယ်။ script tag ကို သုံးပြီး javascript file ကို add တဲ့ အခါမှ src attribute ကိုသုံးပါတယ်။ Internal and external ချိတ်ဆက်ပုံတို့ကို အောက်ပါ ပုံတဲ့မှာ ကြည့်နိုင်ပါတယ်။

```

14 |     <h1>Hello this is h1</h1>
15 |     <script> ←
16 |     const button =document.querySelector('input');
17 |     button.addEventListener('click',buttonAction);
18 |     function buttonAction(){
19 |     }
20 |     </script> ←

```

အထက်ပါ ပုံသည် script tag ကို သုံးပြီး javascript code များအား html file ထဲတွင် ထည့်ထားသောကြောင့် internal ချိတ်ဆက်မှု ဖြစ်ပါတယ်။

```

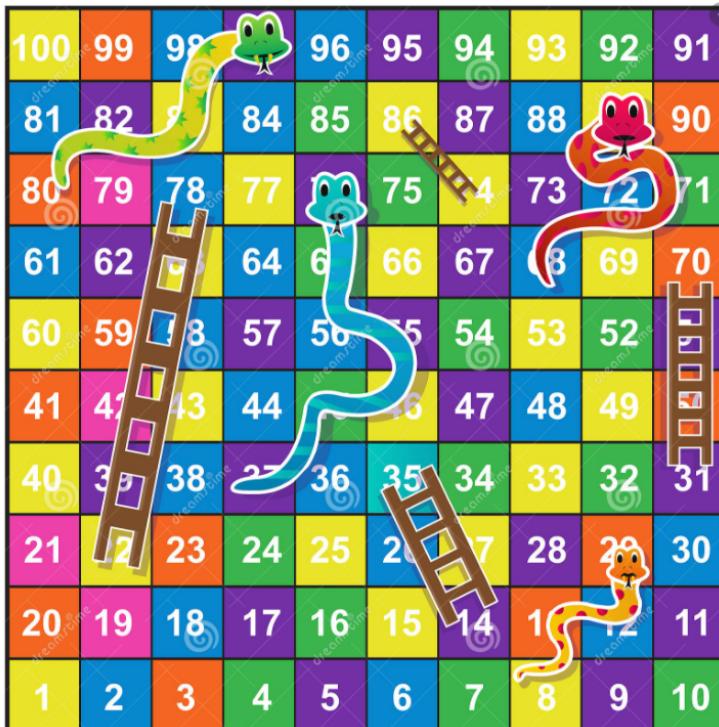
3 |     <head>
4 |         <meta charset="UTF-8">
5 |         <meta name="viewport" content="width=device-wi
6 |         <title>EventListener</title>
7 |         <script src="hello.js"></script>
8 |     </head> ←

```

အထက်မှာ ဖော်ပြထားတဲ့ ပုံကတော့ hello.js ဆိုတဲ့ javascript file ကို src attribute အားသုံးပြီး html file ရဲ့ head tag ထဲတွင် add ထားခြင်း ဖြစ်ပါတယ်။

### Control Structure

User မှ ထည့်ပေးလိုက်သော input အမျိုးမျိုးတို့ စဉ်းစား ဆုံး ဖြတ်ပြီး decision ချ ပေးဖို့ အတွက် မည့်သည့် programming မှာ မဆို လိုအပ်ပါတယ်။ ဥပမာ ပေးရမည့်ဆိုလျှင် ကျွန်တော် တို့ ငယ်ငယ် တုန်းကို အန်စာတိုး ခေါက်ပြီး snake game ကစားခဲ့ကြသလိုပါပဲ။



ထို game ထဲမှာဆိုလျှင် ဆော့တဲ့ သူများသည် အန်စာတိုးကို လှိမ့်လိုက် ရပါတယ် ထိုနောက် အန်စာတိုးက ကျလာတဲ့ number အတိုင်း number စဉ်အတိုင်း လိုက်ပြီး ရွှေ့ရပါတယ်။ အကယ်၍ အန်စာတိုး လှိမ့်လိုက်လို့ 5 ကို ကျရင် လှိမ့်တဲ့ သူရဲ့ နံပါတ်က 5 မှာ ရောက်သွားမှာပါ။ Programming နည်း ဖြင့် စဉ်းစား ကြည့်ကြပါစို့။ ဆော့ တဲ့ သူ တစ်ယောက်ပဲ ထားပါမယ် playerA ပေါ့။

```

1 let playerA=0;
2
3 if( playerA==5){
4     playerA သည် နံပါတ် ၅ သို့ရောက်နေပါသည်;
5
6 }else{
7     playerA သည် အမြား နေရာတွင် ဖြစ်ပါသည်;
8 }
9

```

အထက်ပါ program မှာဆိုလျှင် အန်စာတိုး ခေါက်လိုက်လို့ playerA သည် number 5 ကို ရောက်သွားခဲ့တယ်ဆုံး၍ line 3 မှာ if ( playerA == 5 ) ဆိုပြီး စစ်ထားပါတယ်။ playerA သည် number 5 သာ တကယ် ကျခဲ့တယ် ဆုံး၍ playerA သည် နံပါတ် ၅

သို့ရောက်နေပါသည် ဆိတဲ့ output ကို ထုတ်ပေးမှာ ဖြစ်ပါတယ်( line 4 ) || သို့မဟုတ်( else ) playerA သည် အခြားနေရာသို့ ရောက်နေပါသည်။ အဲတော့ ဒီ program လေးမှာ ဆိုရင် အခြေနှင့် ရှုပါတယ် ပထမ တစ်ခုကတော့ playerA သည် 5 နေရာသို့ ရောက်နေသလားနဲ့ ဒုတိယ အခြေနေသည် 5 မဟုတဲ့ အခြားနေရာသို့ရောက် နေရာသို့ ရောက် နေသလားဆိတဲ့ အခြေနှင့် ဖြစ်ပါတယ်။ Programming တွေမှာ ထဲ အခြေတွေ ပေါ်မှတည်ပြီး မံမိတ္ထာ လုပ်ဆောင် လိုသည့် လုပ်ငန်းစဉ်များကို ကွဲပြားစွာ လုပ်ဆောင်ကြပါတယ်။ ဥပမာ အထက်ပါ program မှာ ဆိုရင် အခြေနှင့် နှစ်ခု ပေါ်မှာ မှတည်ပြီး လုပ်ဆောင် ချက် နှစ်ခု လုပ်ဆောင် ပါတယ် ပထမ တစ်ခုသည် 5 ဖြစ်လျှင် playerA သည် 5 သို့ရောက်နေပါသည်ဆိတဲ့ အခြေနှင့် ဒုတိယမှာ playerA သည် အခြားနေရာသို့ ရောက်နေပါသည်ဆိတဲ့ အခြေနှင့် ဖြစ်ကြပါတယ်။

### Write Your JavaScript Program

JavaScript Program စတင် ရေးသားဖို့ အတွက် visual studio code နှင့် မံမိတ္ထာ နှစ်သက်သည့် browser ( chrome recommended ) တို့ လိုအပ်ပါတယ်။

- 1- <https://code.visualstudio.com/download> ( ယခု link ဖြင့် vscode အား download ဆဲနိုင်သည်)
- 2- <https://www.google.com/chrome/> ( ယခု link ဖြင့် chrome browser အား download ဆဲ နိုင်သည်)

ပထမဆုံး အနေဖြင့် folder တစ်ခု အောက်တွင် test.html နှင့် test.js စသည့် file နှစ်ခု တည်ဆောက်ပါမည်။ html file ထဲတွင် မည်သည့် code မျှ ရေးထားခြင်း မရှိသေးပဲ test.js file အား external အနေဖြင့် ချိတ်ဆက်ထားပါမည်။ test.html ထဲတွင် အောက်ပါ အတိုင်း သာ ဖြစ်သည်။

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>JavaScript</title>
7      <script src="/test.js"></script>
8  </head>
9  <body>
10 
11 
12 
13 </html>
```



ထို့နောက် test.js file ထဲတွင် နောက်ကွယ်မှာ အလုပ်လုပ်မည့် code များကို စတင် ရေးသားပါမည်။

```

1
2 let playerA=5;←
3
4 if( playerA==5){←
5   document.writeln("PlayerA is at",playerA);
6
7 }else{←
8   document.writeln("PlayerA is at another place");
9
10 }

```

### If else statement

အထက်ပါ program မှာ ဆိုလျှင် if else statement ကို စသုံးထားပါတယ်။ if else statement ကို ရေးတဲ့ အခါမှာ if နောက်တွင် parentheses လိုက်ပါတယ်။ ထို parentheses ထဲတွင်မှ condition ကို ရေးပါတယ်။ Condition ဆိုတာက မိမိ စစ်ဆေးချင်တဲ့ အခြေနောက်ပြောတာပါ အထက်ပါ Program မှာ ဆိုလျှင် playerA နဲ့ 5 နဲ့ တူသလား ဆိုပြီး စစ်ဆေးတာပါ။ သူတဲ့ နှစ်ခု ကြေားမှာ သုံးထားတဲ့ === ကိုတော့ comparison operator လို့ ခေါ်ပါတယ်။ Comparison operator တွေ အများကြီး ရှိပါတယ်။ ထိုတဲ့ကမှ === သည် strict equality operator လို့ ခေါ်ပါတယ် သူသည် သူရဲ့ Left and right တွင် ရှိသော value နှစ်ခုကို တူသလား ဆိုပြီး စစ်ဆေးတာပါ အကယ်၍ တူတယ် ဆိုရင် if ( တူသည် ) { အလုပ် ဆက်လုပ်မည် }။ အထက်ပါ program မှာ ဆိုလျှင် line 4 တွင် စစ်ဆေးသော condition မှန်လျှင် line 6 ကို အလုပ် ဆက်လုပ်ပါတယ် တနည်း အားဖြင့် ( condition ) နောက်တွင် ရှိသော curly braces ထဲမှ code များကို အလုပ် ဆက်လုပ်တာ ဖြစ်ပါတယ်။ အကယ်၍ line 4 တွင် စစ်ဆေးသော condition မှ မှန်ဘူး ဆိုလျှင်တော့ line 8 ကို ရောက်သွားပြီး else နောက်မှ curly brace {} ထဲတွင် ရှိသော line 9 မှ code များ အတိုင်း အလုပ် လုပ်ပါတယ်။ ထို နှစ်ခု ထဲမှ တစ်ခုပဲ လုပ်တာပါ။ if and else စတဲ့ စကားလုံးတွေကိုတော့ keyword လို့ ခေါ်ပါတယ်။ Operators တွေ အကြောင်းကိုတော့ JavaScript ရဲ့ operators သင်ခန်းစာများသီးသန့် ထပ်ဆွဲးနွေးပေးပါမယ်။

### Output

#### Else if

ရှေ့က သင်ခန်းစာ ဖြစ်တဲ့ if else lesson မှာ နှစ်ခုတည်းကနေပဲ တစ်ခုကို လုပ်ဆောင်ပါတယ်။ သို့သော် လုပ်ဆောင်ချက် အများကြီး ရှိတဲ့ ထဲကနေ တစ်ခုကို လုပ်ဆောင် ချင်တယ် ဆိုရင် else if ကို သုံးရပါတယ်။ ရေးသားတဲ့ syntax ပုံစံ ကတော့ အောက်ပါအတိုင်း ဖြစ်ပါတယ်။

```

1
2   if(condition){ ←
3     //အပေါက အခြေနေ မှန်ခဲ့ရင် ယခု code များ အလုပ် လုပ်မည်။
4   }else if(condition){ ←
5     //အပေါက အခြေနေ မှန်ခဲ့ရင် ယခု code များ အလုပ် လုပ်မည်။
6   }else if(condition){ ←
7     //အပေါက အခြေနေ မှန်ခဲ့ရင် ယခု code များ အလုပ် လုပ်မည်။
8   }else{ ←
9     //အပေါက အခြေနေ တစ်ခုမှာ မမှန်ခဲ့လျှင် ယခု code များ အလုပ် လုပ်မည်။
10 }

```

အထက်ပါ ပုံ အတိုင်းပင် else if ကို if and else ကြားမှ တွေ့သုံးပါတယ်။ if နောက်မှာ နှင့် else if များနောက်တွင် condition များ ရေးပေးရပါတယ် ထို condition များ မှန်ခဲ့လျှင် condition ရဲ့ နောက်မှ curly braces ထဲတွင် ရှိသော code များကို အလုပ် ဆက်လုပ်မှာ ဖြစ်ပါတယ်။ else if ကို သုံးပြီး real world program လေးတစ်ပုဒ် စရေးပါမယ်။ ကျွန်တော်တို့ အနေနဲ့ html web page တစ်ခုမှာ select and option tag များကို သုံးပြီး user အတွက် ရွှေးချယ်လို့ ရမယ့် option box လေး တစ်ခု ဖန်တီးပါမယ်။ ထို option box ထဲမှာလည်း user က ဘယ် အစား အစာကို မှာလို သလဲ ဆိတာကို မေးပါမယ်။ ထိုနောက်မှာယူလိုတဲ့ အစားအစာ ကို အရေအတွက် ဘယ်လောက် လိုချင်သလဲ ဆိတာ ရွှေးချယ် ခိုင်းပါမယ်။

အောက်ပါ program တွင် Line number 8 မှ 22 ထိ html ရဲ့ body ပိုင်းဖြစ်ပြီး ထို body ပိုင်းထဲတွင် select tag နှင့် input tag တိုကို သုံးထားပါသည်။ select tag သည် user မှ မှာချင်သော အစားအစာများကို ရွှေးချယ်ရန် ဖြစ်ပြီး input tag သည် number type နှင့်ဖြစ်ပြီး user မှာ အစားအသောက် အရေအတွက်ကို မှာရန် အတွက် ဖြစ်သည်။

Please select your favourite Foods:  Number:

Program ကို run ကြည့်သော အခါတွင် အထက်ပါ အတိုင်း မြင်ရမှာ ဖြစ်ပါတယ်။ ယခု စတင်ရေးသားသော Project အတွက် နောက်ကွယ်မှ အလုပ် လုပ်ပေးသော code များသည် line 23 မှ line number 47 အထိ ဖြစ်ပါသည်။

```

8   <body>
9     <label for="food">Please select your favourite Foods:</label>
10    <select id="toEat">
11      <option value="">Please select Here</option>
12      <option value="coffee">Coffee</option>
13      <option value="snack">Snack</option>
14      <option value="juice">Juice</option>
15      <option value="chicken">Chicken</option>
16    </select>
17    <label for="count"> Number:</label>
18    <input type="number" id="count" name="countNumber" value="0"
19      min="1" max="1000">
20    <p></p>
21    <h3> </h3>
22  </body>
23  <script>
24    const select=document.querySelector('select');
25    const para = document.querySelector('p');
26    const count = document.querySelector('input');
27    const number = document.querySelector('h3');
28    select.addEventListener('change',foods);
29    function foods(){
30      const choice = select.value;
31      if(choice ==='coffee'){
32        para.textContent='It is so cool.Coffee is Good';
33        number.textContent='count is'+count.value;
34      }else if(choice ==='snack'){
35        para.textContent='Snack will good for you.';
36        number.textContent='count is'+count.value;
37      }else if(choice ==='juice'){
38        para.textContent ='Juice is so good.';
39        number.textContent='count is'+count.value;;
340      }else if(choice ==='chicken'){
341        para.textContent='Chicken is comming wait a min:';
342        number.textContent='count is'+count.value;
343      }else{
344        para.textContent='Please choose something to eat or drink';
345      }
346    }
347  </script>

```

Line 24 သည် select tag ထဲမှ data များကို ရယူရန် အတွက်ဖြစ်ပါသည်။ line 25 သည် p tag ထဲသို့ data များ ပြန်လည် ဖော်ပြရန် အတွက် ဖြစ်ပါသည်။ line 26 သည် input tag ထဲမှ data များကို ရယူ ရန် အတွက် ဖြစ်သည်။ line 27 သည် h3 tag ထဲသို့ စာသားများ ထည့်ပေးရန် အတွက် ဖြစ်သည်။ line 28 သည် select tag ထဲမှ အဖြစ်အပျက်များကို စောင့်ကြည့်ရန် အတွက် ဖြစ်ပါသည် ထိုသို့ စောင့်ကြည့် ရန် အတွက် addEventListener ကို သုံးပါသည်။ select tag ထဲမှ changes များကို စောင့်ကြည့်မှာ ဖြစ်တဲ့အတွက် select.addEventListener ဟု ရေးထားခြင်း ဖြစ်ပါသည်။ addEventListener သည် method တစ်ခု ဖြစ်ပြီး ထို method သည် parameter နှစ်ခု ယူ ပါသည် ပထမ တစ်ခုသည် change ဖြစ်ပြီး ဒုတိယ တစ်ခုသည် changes ဖြစ်လျှင် လုမ်းခေါ်မည့် function ဖြစ်ပါသည်။ Method

များအကြောင်းကို function သင်ခန်းစာနှင့် OOP သင်ခန်းစာများ တွင် ထပ်မံဆွဲးနွေးသွားပါမည်။

foods() function ထဲသို့ရောက်သွားသော အခါတွင် line 30 ၌ select tag ထဲမှရလာသော value ကို choice ဆိုသည့် variable ထဲသို့ထည့်ပါသည်။ ထို choice ဆိုသည့် variable အား စစ်ဆေးပါသည်။ အကယ်၍ choice သည် coffee ဖြစ်နေလျှင် line 32 and 33 တို့ကို အလုပ်လုပ်ပြီး choice သည် snack ဖြစ်နေလျှင် line 35 and 36 တို့ကို အလုပ်လုပ်ပါသည်။ choice သည် juice ဖြစ်နေလျှင် line 38 and 39 တို့ကို အလုပ်လုပ်ပြီး choice သည် chicken ဖြစ်နေလျှင် line 41 and 42 တို့ကို အလုပ်လုပ်ပါသည်။ choice သည် coffee , snack , juice , chicken တစ်ခုမှ ဖြစ်မနေဘူးဆိုလျှင် line 44 ကို အလုပ်လုပ်ပါသည်။ အခုခက်ပြီး line 32 and 33 မှာ သာယ်လိုအလုပ်လုပ်သလဲဆိုတာကို ရှင်းပြပါမယ်။

Line32 မှာ para.textContent ဆိုပြီးရေးထားပါတယ် ထို ထဲမှ para သည် line 25 တွင် ထောက်ထားသော p tag မှ para ဆိုသည့် variable ဖြစ်ပါတယ်။ textContent သည် ထို paragraph ထဲမှ စာသားများသို့မဟုတ် content များကို ကိုင်တွယ် ဖို့အတွက်သုံးပါတယ်။ line32 တွင် It is so cool.Coffee is Good ဆိုတဲ့စာသားကို paragraph ထဲမှ textContent ထဲသို့ထည့်မည်ဟု ဆိုလိုခြင်း ဖြစ်ပါတယ်။ ထိုကြောင့် web page တွင် စားစရာများကို select လုပ်ပြီးချိန်းလိုက်တဲ့ အခါမှာ line 20 မှ p tag ( paragraph ) နေရာတွင် စာသားများ changes ဖြစ်သွားခြင်း ဖြစ်ပါတယ်။ line 33 မှာလည်း ထိနည်း အတိုင်းပင် ဖြစ်ပါတယ် line 33 မှာတော့ number variable နှင့် သက်ဆိုင်သည့် h3 tag သည် line 21 မှ ဖြစ်သည့် အတွက် အရေအတွက် number ကို ထို h3 tag နေရာတွင် ဖော်ပြခြင်း ဖြစ်ပါတယ်။ ထိုပြင် line 33 ၌ 'count is'+count.value တွင် + ကို သုံးထားပါသည် ထို + sign သည် addition operator ဖြစ်ပါသည် ငါး Operators များ အကြောင်းကို operator lesson များတွင် အသေးစိတ် ထပ်ပြီးဆွဲးသွားပါမည်။

#### Nesting if.....else

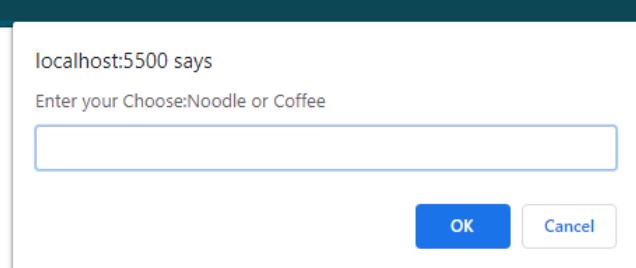
ကျွန်တော်တို့ အနေနဲ့ ခေါက်ဆွဲ ဆိုင် တစ်ဆိုင်ကို သွားကြပြီ ဆိုကြပါစို့ ခေါက်ဆွဲ ဆိုင် ဆိုပေမယ့် ခေါက်ဆွဲ အမျိုး အစားတွေ အများကြီး ထပ်ခွဲ ထားပါသေးတယ်။ ထိုနည်းတူပါပဲ coffee ဆိုင် ရောက်တဲ့ အချိန်မှာလည်း coffee အမျိုးအစားတွေ အများကြီး ထပ်ခွဲထားပါသေး တယ်။ အဲတော့ ခေါက်ဆွဲ ဆိုင်ရောက်ရင် ရှမ်းခေါက်ဆွဲ စားမယ် ရှမ်းခေါက်ဆွဲမှာမှ အစပ် မထည့်ဘူး ရှမ်းခေါက်ဆွဲ သုတေပဲ စားမယ်ပေါ့။ ထိုကဲ့သို့သော ဖြစ်စဉ်ကို Programming အနေနဲ့ သာယ်လို စဉ်းစားမလဲ သာယ်လို statement တွေ သုံးမလဲပေါ့။ ထိုကဲ့သို့သော အခြေနှေမှုဆိုလျှင် nested if....else ကိုသုံးနိုင်ပါတယ်။

ယခု သင်ခန်းစာမှာတော့ user input ကို html ကနေ မတောင်းတော့ပဲ javascript ကနေပဲ တောင်းပါမယ်။ ထိုသို့ တောင်းနိုင်ဖို့ အတွက်ဆိုလျှင် window object ထဲမှ prompt

method ကို သုံးပါမည်။ အသုံးပြုရမည့် ပုံစံမှာ အောက်ပါ အတိုင်း ဖြစ်ပါတယ်။

```
var name = window.prompt("Enter your Choose:Noodle or Coffee ");
```

Window.prompt သည် မိမိတို့ web page တွင် user ဆီမှ data များကို တောင်းခံ ရန် ဖြစ်ပါတယ် ထိုသို့ တောင်းခံ ရာတွင် မိမိတို့ ပေါ်စေချင်သော စာသားများကို prompt method နောက်တွင် parentheses ကိုသုံးပြီး double quote ဖြင့် ရေးပေးရပါတယ်။ အထက်ပါ code အား run ကြည့်လျှင် Output သည် အောက်ပါ အတိုင်း ထွက်လာပါမည်။



```
8 <body>
9   <h3> </h3>
10 </body>
11 <script>
12 var name = window.prompt("Enter your Choose:Noodle or Coffee ");
13 alert("Your has choosen " + name);
14 const number = document.querySelector('h3');
15 if(name==='noodle'){
16   var noodleType = window.prompt("normal or special");
17   if(noodleType === 'normal'){
18     number.textContent='This customer is normal';
19   }else if(noodleType == 'special'){ ←
20     number.textContent='This customer is special';
21   }else{
22     number.textContent='This customer is always Customer';
23   }
24 }else{
25   number.textContent='Coffee Shop';
26 }
27 </script>
28 </html>
```

အထက်ပါ program တွင် ရေးသားထားပုံမှာ ပထမဆုံး အနေဖြင့် line 12 တွင် user ဆီမှ noodle or coffee ဆိုသည့် input ကိုတောင်ပါတယ်။ ထိုကဲ့သို့ တောင်းပြီး ရလာသော data ကို name ဆိုသည့် variable ထဲသို့ ထည့်ပါတယ်။ ထိုသို့ ထည့်ပြီးနောက် alert function ကို သုံးပြီး user မှ ဘာကို ရွှေးချယ်လိုက်တယ် ဆိုတာကို ပြန်ပြပေးပါတယ်(line13)။

Line 14 မှာတော့ h3 tag ကိုလုမ်းပြီး select လုပ်ထားပါတယ်။ line 15 တွင် user ထည့်လိုက်သော data သည် noodle ဖြစ်သလား ဆိုတာကို စစ်ပါတယ် အကယ်၍ noodle ဖြစ်ခဲ့တယ် ဆိုရင် line 16 ကို အလုပ် ဆက်လုပ်မည် ဖြစ်ပြီး noodle မဖြစ် ခဲ့ဘူး

ဆိုလျင်တော့ line 24 သို့ရောက်သွားမည် ဖြစ်ပြီး line 25 မှ code များကို အလုပ် ဆက်လုပ်ပေးမှာ ဖြစ်ပါတယ်။ user မှ noodle ကို ရွေးလိုက် သောကြား Line 16 ကိုသာ အလုပ် ဆက်လုပ်ခဲ့မည် ဆိုလျင် user ဆီမှ နေပြီး normal or special ဆိုသည့် input data ကို ထပ်တောင်းပါတယ်။ ထိုသို့ ထပ်တောင်း လိုက်လို့ User မှ normal ဆိုတာကို ထည့်ပေးလျှင် line 18 ကို အလုပ် ဆက်လုပ်ပါတယ်။ သို့မဟုတ် special ကိုသာရွေးချယ် ခဲ့တယ် ဆိုလျင်တော့ line 20 ကို အလုပ် ဆက်လုပ်ပေးပါတယ်။ ဒီနေရာမှာ if statement ထဲတွင် နောက်ထပ် if statement များကို ထပ်သုံးထားသောကြား nesting if....else ဟု ခေါ်ဆိုခြင်း ဖြစ်ပါတယ်။ ဒီလောက်ဆိုရင်တော့ nesting သဘောတရားကို နားလည် လိမ့်မယ်လို့ မျှော်လင့်သလို if else statement ကိုလည်း ကောင်းကောင်း ရသွားလိမ့်မယ်လို့ ထင်ပါတယ်။

Program လေးကိုပြန်ဆက်ကြည့်ရအောင် line 16 မှာ user မှ special ကို ထည့်လိုက်တယ် ဆိုရင် special အတွက် special တစ်ခု လုပ်ပေးဖို့ လိုအပ်တာပေါ့ ဥပမာ အခြား page တစ်ခုကို ပြပေးတာမျိုးပါ။ ယခု program မှာလည်း user က special ကိုသာ ရွေးချယ်ခဲ့မယ် ဆိုလျင် နောက်ထပ် page တစ်ခု ကို ပြပေးပါမယ်။ ထိုသို့ ပြပေးဖို့အတွက် html ကို မသုံးပဲ javascript ကိုသာ အသုံးပြုသွားပါမယ်။ JavaScript မှာ အခြား page ကို သွားမည် ဆိုလျင် နည်းလမ်း နှစ်ခု ရှိပါတယ်။

#### Redirect a Webpage

အခြား webpage ကိုသွားဖို့အတွက် နည်းလမ်း နှစ်ခုမှာ window.location.href နှင့် window.location.replace တို့ ဖြစ်ပါတယ်။ သူတို့ နှစ်ခုရဲ့ ခြားနားချက်မှာ replace() method သည် လက်ရှိ နေရာမှ အခြားသော webpage တစ်ခုသို့ သွားခြင်း ဖြစ်သလို အခြား webpage သို့ရောက်ပြီးသွားလျှင် မူရင်း နေရာကို ပြန်သွားလို့ မရပါဘူး ဘာကြားလည်းဆိုတော့ replace () method သည် မူရင်း webpage ရဲ့ URL ကို remove လုပ်လိုက်သောကြား ဖြစ်ပါတယ်။

```

1  if(noodleType=='special'){
2      document.writeln("This customer is normal");
3      window.location.replace("https://special.html");
4  }else{
5      document.writeln("This customer is normal");
6      window.location.href="https://www.google.com";
7
8 }
```

အထက်တွင် ရေးထားသော if else statement အား မိမိတို့ main program ထဲတွင်လည်း ထည့်သွင်းရေးသား နိုင်ပါသည်။ စာဖတ်သူများ အနေဖြင့် ထည့်သွင်းရေးကာ စမ်းကြည့်သင့်ပါ သည်။

## Switch Statements

Switch statements ကလည်း if else statements ကဲ့သို့ပင် ရွှေးချယ် စရာတွေ အများကြီး ထဲက တစ်ခုကို ရွှေးချယ်တဲ့ statements တွေ ဖြစ်ကြပါတယ်။ သို့သော ရွှေးချယ်ရမဲ့ ပမာဏ အရမ်းများတယ် ဆိုရင်တော့ if else ကိုပဲ သုံးပါတယ်။ ဥပမာ လူတစ်ယောက် အသက်သည် 20 ကနေ ၅၀ ကြားမှာ ရှိတဲ့ လူတွေကို လူကြီး ဟု သတ်မှတ်မည် ဆိုပါစို့။ 20 - 50 ကြားမှာ ဖြစ်နိုင်ချေရှိတဲ့ ပမာဏသည် ၃၀ တောင် ဖြစ်နေပါတယ်။ ထိုကဲ့သို့သော အခြေနေ မျိုးမှာ ဆိုရင် if else ကို operators တွေနဲ့ တွဲသုံးပါတယ်။ operators တွေဆိုတာ + , - , \* , / , < , > , == ဒါတွေကို ပြောတာပါ အခုံ signs လေးတွေက ငယ်ငယ်က သချုပ်မှာ သင်ခဲ့ကြပြီးသား ဖြစ်ပါတယ်။ အဲတော့ အထက်က စဉ်းစားပုံကို ပြန်ဆက်ရမယ်ဆိုရင် if( age> 20 && age<50 ){ လူကြီး } ဒီလိုပုံစံမျိုး ရေးရမှာ ဖြစ်ပါတယ်။ age နဲ့ 20 နဲ့ ကြားမှာ ရှိတဲ့ > ကို greater than ဟုခေါ်ပြီး age နဲ့ 50 ကြားမှာ ရှိတဲ့ sign ကို less than ဟု ခေါ်ပါတယ်။ && ကိုတော့ AND operator ဟု ခေါ်ပါတယ်။

AND operator ကို condition နှစ်ခုနှင့် နှစ်ခုထက် ပိုပြီး စစ်ဆေးလိုတဲ့ အခါမှာ သုံးပါတယ်။ ရှေ့က lesson ကို ပြန်လှန်ကြည့်မည် ဆိုလျှင် noodleType === 'normal' ယခုလိုမျိုး condition တစ်ခုတည်းကိုပဲ စစ်ဆေးခဲ့ကြတာပါ။ ယခု အခါမှာတော့ if( age> 20 && age<50 ) condition နှစ်ခုကို စစ်ချင်တာ ဖြစ်တဲ့ အတွက် && operator ကို အသုံးပြုခြင်း ဖြစ်ပါတယ်။ && operator ကို နှစ်ခုတည်း စစ်လို့ရတာမဟုတ်ပါဘူး သုံးခု သို့မဟုတ် လေးခု မိမိတို့လိုချင်သလို စစ်ဆေး နိုင်ပါတယ်။ if( age> 20 && age<50 && name==='mgmg' && type === 'boy' ) ယခုကဲ့သို့ condition လေးခုကို စစ်ဆေးလိုသော အခါတွင်လည်း AND operator 3 ခုကိုသုံးပြီး စစ်ဆေး နိုင်ပါတယ်။ ထိုကဲ့သို့ if else statements က conditions အများကြီး ဖြစ်နိုင်တာကို စစ်ပြီးတော့ switch statements ကတော့ conditions နည်းတာကို စစ်ဆေးပါတယ်။ ဥပမာ ဖုန်း ဝယ်မလား laptop ဝယ်မလား ဆိုင်ကယ်ဝယ်မလား ဒီလို conditions နည်းတာမျိုးတွေပါ။ switch statement ရဲ့ ရေးသားပုံ(syntax) ကတော့ အောက်ပါ အတိုင်းဖြစ်ပါတယ်။

```

1 ➤switch(expression){
2   | case 'noodle':
3   |   ခေါက်ဆွဲမှာသည်;
4   | break;
5   | case 'coffee':
6   |   ကော်ဖီမှာသည် ;
7   | break;
8   | case 'juice':
9   |   ဖျော်ရည်မှာသည် ;
10  | break;
11
12 ➤ default:
13   |   အပေါက်ဟာတစ်ခုမှုမလုပ်လျှင်
14   |   ယခုအောက်ဆုံးကအလုပ်ကိုလုပ်ပါ;
15 }
16

```

ပထမဆုံး အနေဖြင့် switch ဆိုတဲ့ keyword ကို စာရေးပါတယ် ထို keyword နောက်ကနေမှ parentheses ကို အသုံးပြုပြီး အထဲတွင် expression ကိုစာရေးပါတယ်။ expression ဆိုတာ variable or မသိကိန်း တစ်ခုပါ။ ထို မသိကိန်း variable သည် noodle or coffee or juice or တစ်ခုမှ မဟုတ်တာလည်း ဖြစ်နိုင်ပါတယ်။ ထို ပြီးလျှင်မှ နောက်ကနေ curly braces လိုက်ပါတယ် ထို curly braces ထဲတွင်မှ မိမိတို့ ရေးချင်သော source code များကို ရေးသား ပါတယ်။ မိမိတို့ စစ်ချင်တဲ့ variable or data ရဲ့ ရှေ့မှာ case ဆိုတဲ့ keyword ကို ထည့်ပေးရပါတယ် ထိုပြင် variable နောက်၌ colon : ကိုထည့်ပေးရပါတယ်။ ပြီးမှ မိမိတို့ ရေးချင်သော expression များကိုရေးသားပြီး အားလုံးရေးသားလို့ ပြီးလျှင် break; keyword ဖြင့် အဆုံးသတ်ပေးရမည်။ အသေးစိတ်ကို အောက်ပါ ပုံတွင်လည်း တစ်ခုချင်းစီ ဖော်ပြပေးထားပါသည်။

```

1     switch keyword          မိမိတို့ စစ်ခေါ်လိုသည် အခြေနေ
2     switch(expression){      မပါ မဖြစ်ပါရမည့် case keyword
3       | case 'noodle':      colon
4       |   ခေါက်ဆွဲမှာသည်;    မိမိတို့ စစ်လိုသည် variable or data
5       | break;              မိမိတို့ ဝေရေးသားလိုသည် code များ
6     }                      semi colon
7
8
9
10
11 အဆုံးတွင် မပါမဖြစ် ပါရမည့် break keyword

```

သို့ဖြစ်၍ switch statements နားလည် သွားရင် အရင် သင်ခန်းစာမှ Program အား if else မသုံးတော့ပဲ switch ကိုသုံးပြီး အောက်ပါ ပုံအတိုင်း ပြောင်းလဲ ရေးသား နိုင်ပါတယ်။ Operators တွေ မပါတော့ပဲ အရမ်းလွယ်ကူ ရုံးရှင်းတာကို တွေ့ရမှာပါ။ သတိပြုရန်

အချက်မှာ အောက်ပါ ပုံထဲမှ line number 50 တွင် default ဖြစ်တဲ့ special customer တွေအတွက် အခြားသော page တစ်ခုကို သွားခိုင်ထားပါတယ်။ စာရေးသူ အနေဖြင့် special.html ဆိုတဲ့ html file အသစ် တစ်ခု တည်ဆောက်ထားပြီး ထို file ထဲတွင် this is special ဆိုတဲ့စာကြားတွင် ကိုသာရေးထား ပါတယ်။

```
<body>
    this is special
</body>
```

နောက်ထပ် သတိပြုရန် အချက်မှာ အောက်ပါ ပုံထဲတွင် line number 51 ခြား break ကိုသုံးထားပါသည် ထို break မှာ မပါလည်း ရပါသည် default တွင် break ထည့်ပေးရန် မလိုအပ်ပါ။ အဘယ်ကြား ဆိုသော် default သည် switch statement ရဲ့ နောက်ဆုံး condition ဖြစ်တဲ့ အတွက် break မပါလည်း program က ရပ်သွားမှာ ဖြစ်ပါတယ်။

```
28     select.addEventListener('change',foods);
29     function foods(){
30     let choosen=select.value;
31     switch(choosen){
32     case 'coffee': ←
33         paragraph.textContent= choosen+" is so cool";
34         h3.textContent ='Quantity is'+ number.value;
35         break; ←
36     case 'snack': ←
37         paragraph.textContent= choosen+" is so cool";
38         h3.textContent ='Quantity is'+ number.value;
39         break; ←
40     case 'juice': ←
41         paragraph.textContent= choosen+" is so cool";
42         h3.textContent ='Quantity is'+ number.value;
43         break; ←
44     case 'chicken': ←
45         paragraph.textContent= choosen+" is so cool";
46         h3.textContent ='Quantity is'+ number.value;
47         break; ←
48     default: ←
49         paragraph.textContent="This is special customer";
50         window.location.href="special.html";
51         break;
52     }
53 }
```

### Ternary operator

Ternary operator ကတော့ အရမ်းရှိုးရှင်းပြီး အလွယ်ကူးဆုံး operator တစ်ခု ဖြစ်ပါတယ်။ if ....else တို့ switch statements တို့လို syntax မရှုပ်ထွေးသလို code line လည်း နည်းပါတယ်။ သူရဲ့ syntax ကတော့ ပထမဆုံး parentheses ထဲမှာ condition ရှိပါတယ် ထို

parentheses နောက်မှာမှ ? လိုက်ပါတယ် ထိုနောက်မှ မိမိတို့ အလုပ်လုပ် လိုသော လုပ်ဆောင် ချက် နှစ်ခုကို : colon [ြားပြီး ရေးပါတယ်။

#### Syntax

(condition)? Run this code1 : run this code instead 2

ဆုလိုချင်တာကတော့ parentheses ထဲက condition က True ဖြစ်ခဲ့မယ်ဆုရင် run this code 1 ဆုတာကို အလုပ်လုပ်ပြီး မ မှန်ခဲ့ဘူး(False) ဆုရင်တော့ run this code instead 2 ဆုတာကို အလုပ်လုပ်ပါတယ်။

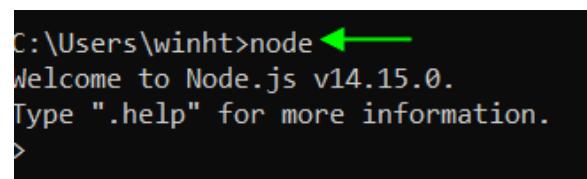
```
1 let men=true;
2 let gender = (men)? 'He is a man':'we dont know';
```

အထက်ပါ code နှစ်လိုင်းတွင် men ရဲ့ value ကို true ဆုတဲ့ boolean value တစ်ခု ပေးထားပါတယ်။ ဒါဆုရင် men ဆုတဲ့ variable ရဲ့ value က true ပါ။ line 2 မှာ ( men ) ဆုပြီး parentheses ထဲမှာ men ကိုထည့်စွစ်လိုက်ပါတယ် အကယ်၍ men ရဲ့ value သည် true ဖြစ်ခဲ့မည် ဆုလျှင် He is a man ဆုတဲ့ output ကို ရမှာပါ false ဖြစ်နေခဲ့ရင် we dont know ဆုတဲ့ output ကို ရမှာပါ။ ထိုပြင် ရှေ့မှာ let gender ကိုပါ ထည့်ထား သေးတဲ့ အတွက် ရလေတဲ့ output ကို gender ဆုတဲ့ variable ထဲမှာ ထည့်ပေးလိုက်မှာ ဖြစ်ပါတယ်။ အထက်ပါ program ကို run ကြည့်မည် ဆုလျှင် He is a man ဆုတဲ့ output ကို ရမှာပါ။

Program တွေကို မြန်မြန် ဆန်ဆန် ကောင်းမွန်စွာ run နိုင်ဖို့ ယခု သင်ခန်းစာ ကင့် စတင်ပြီး node.js ကို စသုံးပါမယ်။ Node.js ကို အကျဉ်းချုပ်ပြောရမည် ဆုလျှင် ပုံမှန် အားဖြင့် javascript သည် web browser ပေါ်မှာ သာလျှင် run နိုင်ပါတယ် သို့သော် node.js ကို install လုပ်ပြီးသည်နှင့် တစ်ပြိုင်နက် မိမိတို့ စက်ထဲမှာပါ run နိုင်ပါတယ်။ Node.js ကို install လုပ်နိုင်ဖို့ အောက်ပါ link ကိုအရင်သွားပါ။

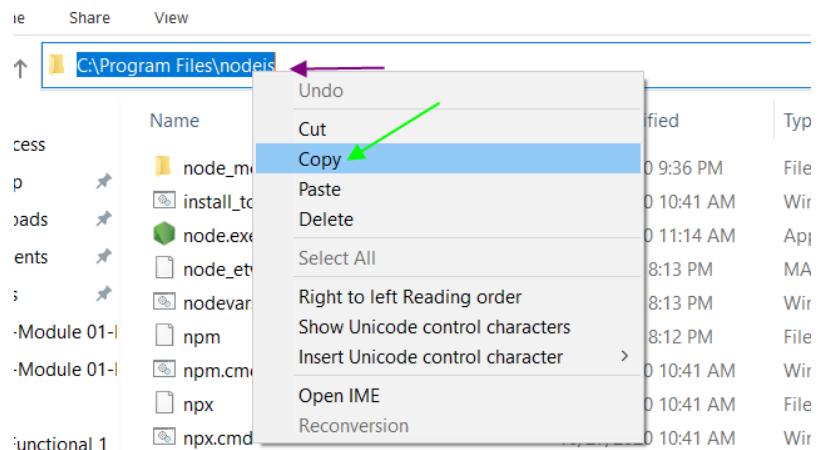
<https://nodejs.org/en/download/>

မိမိတို့ computer နှင့် ကိုက်ညီမည့် file ကို download ဆွဲပါ စာရေးသူ အနေဖြင့် windows ဖြစ်သည့် အတွက် windows ကိုသာ download ဆွဲပါတယ်။ download ဆွဲလို ရလေသော installer file ကို install လုပ်ပေးပြီး node.js ကို စတင် အသုံးပြု နိုင်ပါပြီ။ မိမိတို့ စက်တွင် node.js ကို အသုံးပြုလို ရနိုင် မရနိုင်ကို သိနိုင်ဖို့ cmd သို့ သွားပြီး node ဟုရေးကြည့်ပါ။ အောက်ပါ ပုံထဲက အတိုင်း node version ပေါ်လာတယ် ဆုရင် မိမိတို့ စက်မှာ node.js ကို အသုံးပြု နိုင်ပါပြီ။

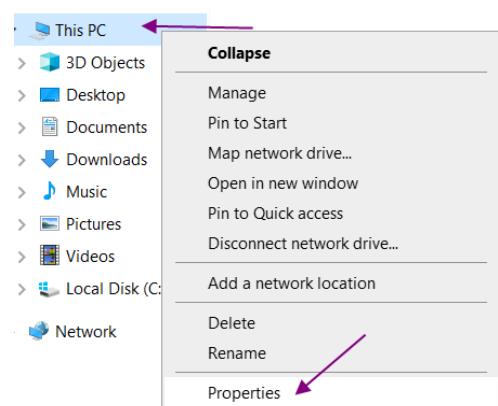


```
C:\Users\winht>node
Welcome to Node.js v14.15.0.
Type ".help" for more information.
>
```

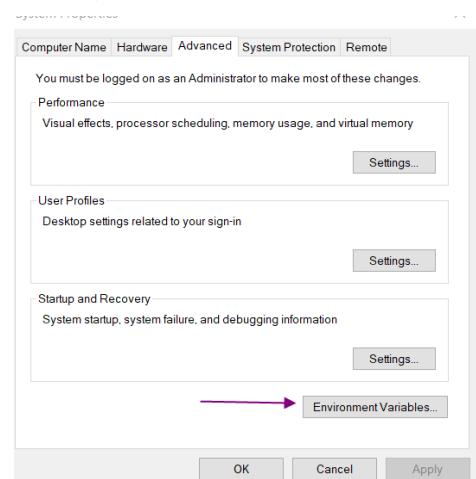
အကယ်၍ windows သမား ပျော်မှုများ အနေဖြင့် ထိကဲသို့ မပေါ်ပဲ error တစ်ခုခဲ့  
တက်ခဲ့ပါက C:\Program Files\nodejs ယခု အတိုင်းသွားပြီး select မှတ်ကာ copy ကူးခဲ့ပါ။

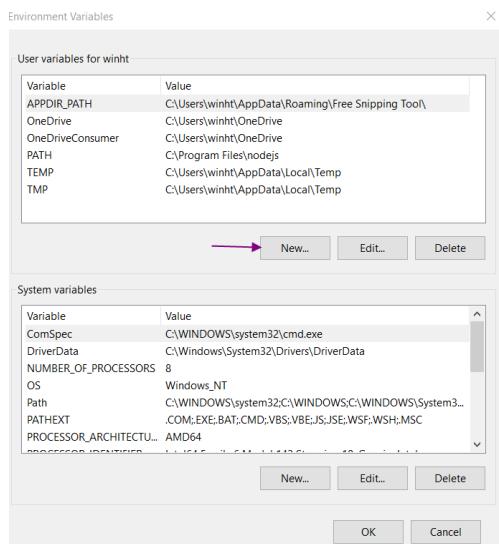


ထို့နောက် mycomputer or This PC အား right click ထောက်ပြုး properties ထဲသို့  
သွားပါ။

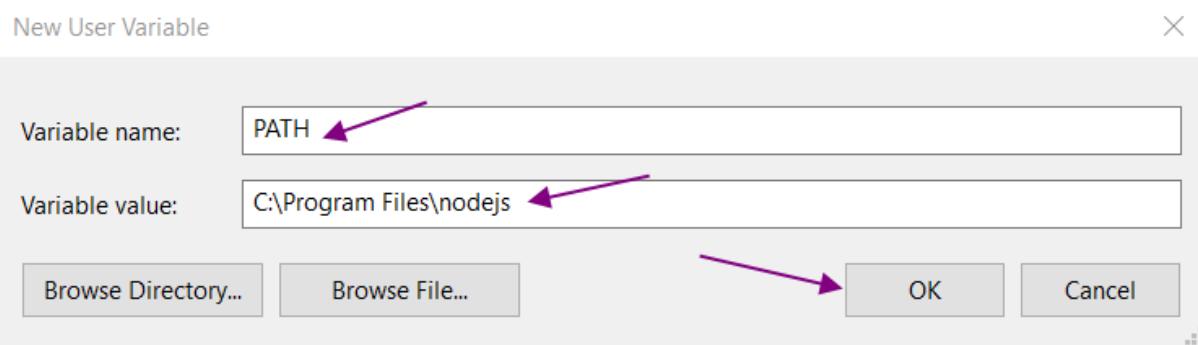


ထို့နောက်ပေါ်လာသော window from တွင် Advance system settings ကို နှိပ်ပါ  
ပြီးလျှင်အောက်ပါ ပုံအတွင်း Environment Variables ကို နှိပ်ပါ။





အထက်ပါ ပုံတွင် မြားပြထားသည့် အတိုင်း New ကို ဆက်နိုပ်ပါ။



အထက်ပါ ပုံတဲ့မှ အတိုင်း Variable name နေရာ တွင် PATH ကို ရေးပါ ထိုနောက် Variable value နေရာတွင် မိမိတို့ copy ကူးလာသော စာသားများ ကိုထည့်ပါ။ ပြီးလျှင် OK ကို နိုပ်ပါ ကျွန်ုတ်သည့် window forms များအားလုံးတွင်လည်း OK များကိုသာ နိုပ်လိုက်ပါ။ အထက်ပါ အဆင့်များ ပြီးပါက cmd တွင် node ဟု ပြန်ရှိကြဖို့ပါက version ပေါ်လာသည်ကို တွေ့ရပါမည်။

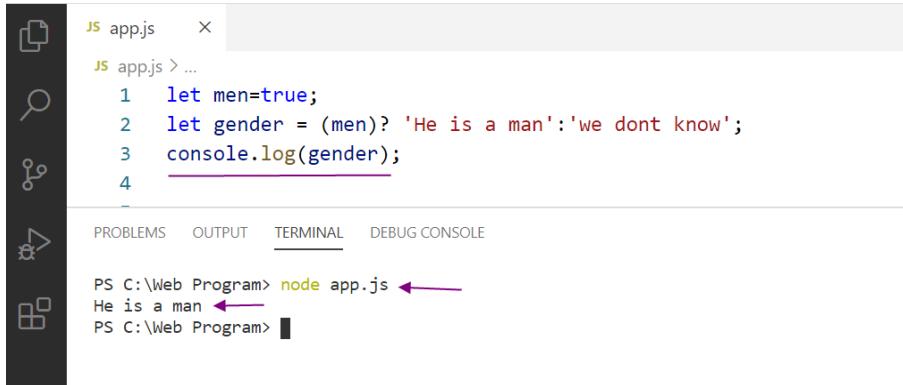
အထက်ပါ ဖော်ပြချက်များသည် windows os အသုံးပြုသူများ error တက်လျှင် ပြရှင်းရန် နည်းလမ်း ဖြစ်သည်။

အဆင့်များ အားလုံး ပြီးပါက visual studio code ကို ဖွင့်ပါ ထိုနောက် ctrl+ ` ကို နိုပ်လိုက်ပါက အောက်ပါ အတိုင်း terminal ပေါ်လာပါမည်။



ထိုနေရာတွင် မိမိတို့ program များကို run နိုင်ပါသည်။ program run ရန် ပထမဆုံး

အနေဖြင့် app.js ဆိုသည့် file တစ်ခုကို စဖွင့်ပါ။ ထို app.js file ထဲတွင် အောက်ပါ code များကို ရေးသားပါ။



```
JS app.js  X
JS app.js > ...
1 let men=true;
2 let gender = (men)? 'He is a man': 'we dont know';
3 console.log(gender);
4

PROBLEMS   OUTPUT   TERMINAL   DEBUG CONSOLE

PS C:\Web Program> node app.js ←
He is a man ←
PS C:\Web Program> █
```

အထက်ပါ အတိုင်းရေးသားပြီးလျှင် `ctrl + `` ( esc အောက်နားက ) ကိုနိပ်လိုက်လျှင် terminal တစ်ခု ပေါ်လာပါမည် ထို terminal တွင် node app.js ဟု ရေးပြီး enter ခေါက်ကြည့်ပါက output ထွက်လာသည်ကို မြင်ရပါမည်။ ထိုသို့ မပေါ်လာလျှင် မိမိတို့ program အား save ခဲ့သလား ပြန်လည် စစ်ဆေးပါ။

အထက်ပါ အဆင့်များ အတိုင်း အဆင်မပြေပါက youtube or google တွင် ကြည့်ရန် အကြံပေးလိုပါသည်။ အခုခက်ပြီးတော့ မိမိတို့ရဲ့ web page မှာ theme ပြောင်းတဲ့ beginner project လေးတစ်ခုကို စတင် ရေးသား သွားပါမယ်။

```
<body>
  <label for="theme">Select Your Theme </label>
  <select id="theme">
    <option value="white">White</option>
    <option value="black">Black</option>
  </select>

  <h1>This is my website</h1>
</body>
<script>
```

အထက်မှာ ရေးထားတဲ့ program ကတော့ body ပိုင်းမှာရေးမယ့် program ဖြစ်ပါတယ်။ သည် နေရာမှာ select tag and body tag ကို တွဲသုံးထားပါတယ် မိမိတို့ ရွှေးချယ် လိုတာတွေ ကို ရွှေးချယ် နိုင်ဖိုပါ။

```

<script>
const select = document.querySelector('select');
const html = document.querySelector('html');
document.body.style.padding = '20px';

function update(bgColor, textColor) {
    html.style.backgroundColor = bgColor;
    html.style.color = textColor;
}

select.onchange = function() {
    (select.value === 'black') ? update('black','white') :
    update('white','black');
}

</script>

```

ပထမဆုံး အနေဖြင့် user ရွေးချယ်လိုက်သည့် select tag ထဲမှ color ကို သိနိုင်ရန် အတွက် querySelector သုံးပြီး select tag ကို select ဆိုသည့် varaible တစ်ခု အသေး သိမ်းထား ပါတယ်။ ထိုပြင့် html တစ်ခုလုံးကို ပြင်ချင်သည့် အတွက် querySelector ကိုသုံးပြီး html ကိုလည်း html variable တစ်ခုအသေး သိမ်းထားပါတယ်။ document.body.style.padding ဆိုတာကတော့ ဘေးဘောင်ကနေ 20px ကွာ နိုင်ဖို့ အတွက် ဖြစ်ပါတယ်။

ဒုတိယ အနေဖြင့် အရမ်း ရိုးရှင်းတဲ့ function တစ်ခုကို ရေးသားပါတယ် ထို function မှာ parameter နှစ်ခု ပါ ပါတယ် ပထမ တစ်ခုသည် bgColor ဖြစ်ပြီး background color ကို ချိန်းရန် အတွက် ဖြစ်ပါတယ် ဒုတိယ parameter သည် textColor ဖြစ်ပြီး စာသားများရဲ့ color ကို ချိန်းရန် အတွက် ဖြစ်ပါတယ်။

```

22  select.onchange = function() {
23      (select.value === 'black') ? update('black','white') : update('white','black');
24  }
25
26  function update(bgColor, textColor) {
27      html.style.backgroundColor = bgColor;
28      html.style.color = textColor;
29  }

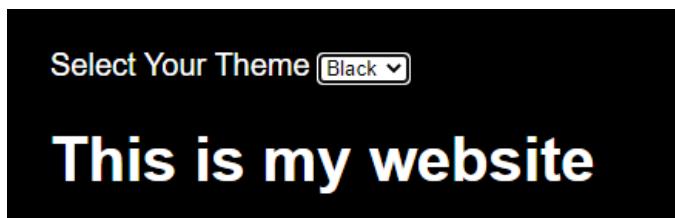
```

Select.onchange သည် select tag ထဲတွင် တစ်ခုခဲ့ ချိန်းသွားလျှင် သိနိုင်ရန် အတွက်ဖြစ်ဖြစ်ပြီး တစ်ခုခဲ့ change သွားသည်နှင့် တစ်ပြီးပြီး သူ့အောက်မှ code များကို အလုပ် ဆက်လုပ်ပါတယ်။ အကယ်၍ user ကနေပြီး black ကိုသော်လိုက်မည် ဆိုလျှင် update function ကို black and white parameter နှစ်ခုနဲ့လှမ်းခေါ်ပါတယ်။ line 23 ကနေပြီး line 26

ကို လုမ်းခေါ်လိုက်တာပါ ထို့ကြောင့် bgColor နေရာတွင် black ဝင်လာပြီး textColor နေရာတွင် white ဝင်လာပါတယ်။ ထို့သို့ ဝင်လာပြီးနောက် bgColor ဖြစ်သည့် black ကို html.style.backgroundColor ထဲသို့ ထည့်ပေးလိုက်ပါတယ်။ ထို့ကြောင့် line 27 ကို အလုပ်လုပ်ပြီးတဲ့ အချိန်မှာ ကျွန်ုပ်တို့ web page တွင် background color သည် black ဖြစ်သွားမှာ ဖြစ်ပါတယ်။ အကယ်၍ web page တွင် background color သည် black ဖြစ်သွားလျှင် စာသားများပါ black ဖြစ်နေသည့် အတွက် စာသားများကို မမြင် ရနိုင်ပါဘူး ထို့ကြောင့် စာသားများကို ပြင်ရန် textColor ထဲမှ white အား html.style.color ထဲသို့ ထည့်ပေးလိုက်ပါသည်။ ထို့ကြောင့် background color black ဖြစ်လျှင် စာသားများသည် white ဖြစ်ပြီး background color white ဖြစ်လျှင် textColor သည် black ဖြစ်ပါမည် ထို့သို့ ဖြစ်ရန် အတွက်လည်း program ကို Ternary operator သုံးပြီး အောက်ပါ အတိုင်း ရေးထားခြင်း ဖြစ်ပါသည်။

```
22 select.onchange = function() {
23   | ( select.value === 'black' ) ? update('black','white') : update('white','black');
24 }
25
```

Program ရေးထားပဲ အပြည့်အစုံနှင့် output ကို အောက်တွင် ဖော်ပြထားပါတယ်။



Select Your Theme

**This is my website**

Select Your Theme

**This is my website**

```

8   <body>
9     <label for="theme">Select Your Theme </label>
10    <select id="theme">
11      <option value="white">White</option>
12      <option value="black">Black</option>
13    </select>
14
15  <h1>This is my website</h1>
16  </body>
17  <script>
18  const select = document.querySelector('select');
19  const html = document.querySelector('html');
20  document.body.style.padding = '20px';
21
22  select.onchange = function() {
23    ( select.value === 'black' ) ? update('black','white') : update('white','black');
24  }
25
26  function update(bgColor, textColor) {
27    html.style.backgroundColor = bgColor;
28    html.style.color = textColor;
29  }
30  </script>
31 </html>

```

အထက်ပါ program မှာ ဆိုရင် black and white theme နှစ်ခုကိုပဲ User ကိုရွေးချယ်ခွင့်ပေးထားပါတယ် အကယ်၍ user က အဲဓိတက်ပိုပြီး color တွေကို ရွေးချယ်လိုတယ် ဆိုရင် ရော ကျွန်ုပ်တို့ ဘယ်လို ရေးနိုင်သလဲ။

#### Exercise

နောက်ထပ် program တစ်ပုဒ်အနေဖြင့် user ကို white , black , pink , purple စတဲ့ themes လေးမျိုးကို ရွေးချယ် ခိုင်းပါမယ် ထို ကဲ့သို့ ရွေးချယ်လိုက်တဲ့ theme များအတိုင်း သူနဲ့ လိုက်ဖက်တဲ့ ဖတ်လိုရနိုင်တဲ့ စာလုံးအရောင်ကိုလည်း ပြောင်းပေးရပါမယ်။ ပြီးရင် user ရွေးချယ်လိုက်တဲ့ အရောင်ကိုလည်း ပြန်လည် ဖော်ပြ ပေးနိုင်ရပါမယ်။ Sample Program ကိုတော့ အောက်မှာ ရေးပြထားပါတယ် sample program ကို မကြည့်ခင်မှာ မြမ်တို့ ကိုယ်တိုင် program ကို ရေးကြည့် သင့်ပါတယ်။



```

17  <h1>Your Favourite Theme(' - ')</h1>
18  <p></p>
19  </body>
20  <script>
21  const select = document.querySelector('select');
22  const html = document.querySelector('html');
23  let ptext = document.querySelector('p');
24  document.body.style.padding = '20px';
25  select.onchange = function() {
26    switch(select.value){
27      case 'white':
28        update('white','black');
29        ptext.textContent= select.value + ' is so amazing';
30        break;
31      case 'black':
32        update('black','green');
33        ptext.textContent= select.value + ' is so amazing';
34        break;
35      case 'pink':
36        update('pink','white');
37        ptext.textContent= select.value + ' is so amazing';
38        break;
39      case 'purple':
40        update('purple','yellow');
41        ptext.textContent= select.value + ' is so amazing';
42        break;
43      default:
44        update('white','black');
45        ptext.textContent= select.value + ' is so amazing';
46    }
47  }
48  function update(bgColor, textColor) {
49    html.style.backgroundColor = bgColor;
50    html.style.color = textColor;
51  }
52  </script>
-- 

```

## Variables

Variable ဆိုတာ data တွေကို stored(သိမ်း) ဖို့ အတွက်ပါ။ ဥပမာ ကျဉ်တော်တို့ အိမ်ကို လူတစ်ယောက် အလည်လာတယ် ဆိုပါစို့ သူထိုင်ဖို့ အတွက် ထိုင်ခံ တစ်ခု ဖန်တီးပေးရပါတယ်။ ထိုနည်း အတိုင်းပါပဲ ကျဉ်တော်တို့ ရေးနေတဲ့ program ထဲကို data အသစ် တစ်ခု ရောက်လာမယ် ဆိုရင် ထို data သိမ်းဖို့ အတွက် variable တစ်ခု ဖန်တီးပေးရပါတယ်။ ဥပမာ ကျဉ်တော်တို့ web page ကို အသုံးပြုသည့် user ဆီကနေ သူရဲ့ နံမည်ကို ရယူ ချင်တယ် ဆိုပါစို့ user ကနေပြီး နံမည် ရိုက်ထည့် ပေးလိုက်တယ် အဲတော့ program ရေးသူ programmer ကလည်း အဲဒီ variable ကို သိမ်းဖို့ နေရာ တစ်ခု လိုလာပါတယ်

အဲဒိုလို သိမ်းတဲ့ နေရာကို variable လို့ ခေါ်တာပါ။ ဒါဆုံးရင် ထပ်ပြီး စဉ်းစားကြည့်ကြရအောင် user ကနေ name , age , address စတဲ့ မတူညီတဲ့ data ပုံစံတွေနဲ့ ထည့်သွင်းသိမ်းဆည်းကြပါတယ်။ ထိုကဲ့သို့ မတူညီတဲ့ data တွေ အတွက်လည်း မတူညီတဲ့ variable types တွေရှုပါတယ်။

1. Number
2. String
3. Boolean
4. Array
6. Object

### Number

Program ထဲမှာ number တွေ stored လုပ်မည် ဆုပါစ္စ 10 ကဲ့သို့ ကိန်းပြည့် integer တွေရှိသလို 1.3435 စတဲ့ ဒသမကိန်း decimal number တွေ သိမ်းဆည်းတာမျိုးလည်းဖြစ်နိုင်ပါတယ်။ အခြားသော programming တွေမှာ ဆုရင်တော့ ဥပမာ c programming မှာ ဆုရင် integer ကိန်းပြည့် variable ကြော်မယ် ဆုရင် အောက်ပါ အတိုင်းကြော်ရပါတယ်။

```
int myAge=27;
```

အထက်ပါ အတိုင်း integer ကို ကြော်ရပါတယ် ထိုကဲ့သို့ မကြော်ဘူး ဆုလျှင် error တက်ပါတယ်။ ထိုပြင် ဒသမကိန်း decimal numbers တွေ အတွက်ဆုလျှင်လည်းအောက်ပါ အတိုင်း float or double စတဲ့ type တွေကို ကြော်ပြီး သုံးရပါတယ်။ ကိန်းဂဏ်း သေးတဲ့ ဒသမ ကိန်းတွေ stored လုပ်ချင်ရင် float ကိုသုံးပြီး ကိန်းဂဏ်းကြီးတဲ့ ဒသမ ကိန်းတွေ stored လုပ်ချင်ရင် double ကို သုံးပါတယ်။

```
float height=1.35;
```

```
double height=11111.35555;
```

စသည်ဖြင့် အထက်ပါ အတိုင်း စနစ်တကျ သိမ်းဆည်း ရပါတယ်။ ထိုသို့ မသိမ်းဆည်းလျှင် data loss ဖြစ်ခြင်း error တက်ခြင်း များ ဖြစ်နိုင်ပါတယ်။ သို့သော JavaScript မှာတော့ အဲလို မဟုတ်ပါဘူး double quote မပါပဲ number တစ်လုံး variable ထဲကို ထည့်လိုက်မည် ဆုလျှင် အလုံလျောက် ထို variable ရဲ့ type ကို သိပါတယ်။ ဥပမာ စာသားဆုရင် string or char , number ဆုရင် int , float , double စတဲ့ type တွေ အတိအကျ ထည့်နေစရာ မလိုပါဘူး။ JavaScript မှာ integer or decimal number တစ်ခု ကြော်လိုလျှင် အောက်ပါ အတိုင်း ကြော်လိုက်ရုံပါပဲ။

```
let myAge=27;
```

```
let myHeight=5.9;
```

အထက်ပါ အတိုင်း number တွေကို “ ” double quotes မပါပဲ ကြော်လိုက်မယ် ဆိုရင် JavaScript က myAge သည် number ဖြစ်ကြာင်း သံပါတယ်။ သို့သော် let myAge="27"; ဟု ရေးခဲ့မည် ဆိုလျှင်တော့ double quotes ပါသည့် အတွက် string(စာသား) ဟု JavaScript က မှတ်ယူ လိုက်မှာ ဖြစ်ပါတယ်။

### String

String ဆိုတာကတော့ program ထဲမှာ စာသားတွေ stored လုပ်တဲ့ အချိန် သုံးတာပါ။ ဥပမာ နံမည် တစ်ခု 'mgm' ပေါ့။ ထိကဲသို့ ကြော်တဲ့ အချိန်မှာ data ကို single quote or double quote ကို သုံးနိုင်ပါတယ်။

```
let myName = 'WinHtut' ;
```

### Boolean

Boolean type ဆိုတာကတော့ true or false စတဲ့ data တွေကို stored လုပ်ဖိုပါ။ အခြေနေ တစ်ခုအား မှားသည် မှန်သည် စသဖြင့် သတ်မှတ်လိုသော အခါများတွင် true or false ကို သုံးနိုင်ပါတယ်။

```
let male = true;
```

### Arrays

ပုံမှန် variable တစ်လုံးမှာ ဆိုရင် name တစ်ခု or value တစ်ခုပဲ သိမ်းနိုင်ပါတယ် သို့သော် array တော့ square bracket တစ်ခုရေးပြီး ထိ bracket ထဲမှာ မိမိတို့ store လုပ်ချင်တဲ့ data တွေကို comma ခြားပြီး ရေးသား နိုင်ပါတယ်။

```
let students=['mgm', 'aungaung', 'koko'];
```

အထက်ပါ အတိုင်း students ဆိုတဲ့ array တစ်ခုထဲမှာ data များစွာကို ထည့်နိုင်ပါတယ်။ သတိပြုရန် အချက် မှာ array ကို ရေးမည် ဆိုလျှင် [ ] square bracket ကို သုံးရမှာ ဖြစ်ပါတယ်။ ထို array ထဲမှ data တွေကို ပြန်ထုတ် ချင်တယ် ဆိုရင်တော့ index number ကို သုံးပြီး ထုတ်ရမှာ ဖြစ်ပါတယ်။ index number တွေက ရှုံးဆုံးတွင် 1 ကနေ မစပဲ 0 ကနေပဲ စပါတယ်။ ဥပမာ အထက်မှာ ကြော်ထားတဲ့ students ဆိုတဲ့ array ထဲက mgm ဆိုတဲ့ data ကို ရယူ ချင်တယ် ဆိုရင် အောက်ပါ အတိုင်း ရေးရမှာ ဖြစ်ပါတယ်။

```
console.log( students[0] ) ;
```

အထက်ပါ အတိုင်းရေးလိုက်လျှင် mgm ဆိုတဲ့ index 0 နေရာက data ကို ပြန်လည် ရှုံးရမှာ ဖြစ်ပါတယ်။ ထိုနည်းတူ နောက်ဆုံးက koko ဆိုတဲ့ data ကို ထုတ်ချင်တယ် ဆိုရင်တော့ console.log( students[2] ) ; ဟု ယခု အတိုင်း ရေးရမှာ ဖြစ်ပါတယ်။

### Object

Object ဆိတာကတော့ data တစ်ခုတည်း မဟုတ်ပဲ သူနဲ့ သက်ဆိုင်တဲ့ data တွေကိုပါ stored လုပ်ဖို့ အတွက်သုံးပါတယ်။ ဥပမာ လူ တစ်ယောက် ဆိုပါစို့ သူမှာ name ,age ,height , hobby စိတဲ့ အချက်လက်တွေ ရှိနေ ပါသေးတယ်။ ထိုကဲ့သို့သော အချက်လက်တွေ stored လုပ်ဖို့ အတွက်ဆိုလျှင် Object type ကိုသုံးပါတယ်။ Object type ကို သုံးဖို့အ တွက်ဆိုလျှင် curly brace ကိုသုံးပါတယ်။ ထို curly brace ထဲမှာမှ မိမိတို့ ထည့်လိုသော data များကို key and value ပုံစံဖြင့် အတွဲလိုက် ထည့်ပေးရပါတယ်။

```
let human={ name:'WinHtut',age:'27',hobby:'BuildingComputer'};
```

```
console.log(human.)
```

The screenshot shows a code editor with the following code:

```
let human={ name:'WinHtut',age:'27',hobby:'BuildingComputer'};
```

Below the code, there is a call to `console.log(human.)` followed by a tooltip-like box. The box contains three entries: "age" with a small icon, "(property) age: string" to its right, and "hobby" and "name" below it, each with a small icon.

အထက်ပါ program ၏ object သည် human ဖြစ်သည် ထို object ထဲ၌ data သုံးခါ ပါဝင်ပါသည်။ ထို data များ ထည့်သော အခါတွင် key : value ပုံစံဖြင့် ထည့်ရမည် ဖြစ်ပြီး name သည် key ဖြစ်ပြီး 'WinHtut' သည် value ဖြစ်ပါသည်။ ဒုတိယ Line တွင် human ဟု ရေးလိုက် သည်နှင့် တစ်ပြိုင်နက် human object ထဲတွင် ရှိသော keys or properties များကို တစ်ခါတွင် ပြပေးပါသည်။ ဥပမာ 'WinHtut' ဆိုတဲ့ data ကို ထုတ်ချင်တာပဲ ဖြစ်ဖြစ် ပြုပြင် ပြောင်းလဲ ချင်တာပဲ ဖြစ်ဖြစ် human.name ဟုရေးပြီး ပြုပြင် ပြောင်းလဲ နိုင်သည်။

```
console.log( human.name );
```

### Dynamic Typing

JavaScript သည် dynamically typed language တစ်ခု ဖြစ်ပါတယ်။ ဥပမာ variable တစ်လုံးကို ကြော်လှုံးလိုက်သည် ဆိုပါစို့ let keyword ကိုသုံးပြီး ကြော်လှုံးလိုက်ရှုပဲ ဖြစ်ပါတယ်။ int , float , boolean , string စသည့် မည်သည့် data type ကိုမှ ရှုံးက ထည့်ပေးစရာ မလိုပါဘူး။ JavaScript, Objective-C, PHP, Python, Ruby, Lisp, and Tcl စသည့် languages များသည်လည်း dynamic typed language များ ဖြစ်ကြပါတယ်။ variable တွေရဲ့ type တွေကို စစ်ဆေးနိုင်ရန် Programming language တော်တော်များများမှာ typeof ဆိုတဲ့ Operator တစ်ခု ရှိပါတယ် သူကို သုံးပြီး စစ်ဆေး နိုင်ပါတယ်။

```
let age='10';
console.log(typeof(age));
```

အထက်ပါ အတိုင်း program ကို ရေးပြီး run ကြည့်ပါ Output အနေဖြင့် string ဆိတာကို ပြန်ရပါလိမ့်မယ် အဘယ်ကြောင့်ဆိုသော '10' ဆိုသည့် value သည် single quote ပါနေလိုပါ ထိုကဲ့သို့ ပါလာခြင်းကို JavaScript သည် အထဲမှာ ဘာdata ပဲ ထည့်ထား ထည့်ထား

single quote ပါလာရင် string အနေဖြင့် သတ်မှတ်ပါတယ်။ ထိုကြောင့် နံပါတ်များ ဖြစ်စေ စာသားများ ဖြစ်စေ String အနေဖြင့် အသုံးပြုလို ပါက single quote ကိုထည့်ပေးရမှာ ဖြစ်ပါတယ်။ ထိုနည်းတူ number များကို number data type ပုံစံဖြင့် အသုံးပြုလိုပါက single quote မပါပဲ အသုံးပြုရမှာ ဖြစ်ပါတယ်။

```
let age=10;
console.log(typeof(age));
```

အထက်ပါ အတိုင်း ရေးပြီး program run ကြည့်ပါက number ဆိုသည့် output ကို ရမှာ ဖြစ်ပါတယ်။

### Constant

Programming languages တော်တော်များများမှာလည်း constant data type ရှိပါတယ်။ value တစ်ခုခု ကို program တစ်ပုဒ်မှာ မပြောင်းလဲ လိုပဲ အသေထားချင်သော အခါတွင် အသုံးပြုပါတယ်။

```
const passcode= 562789 ;
```

အထက်ပါ အတိုင်း သာ ရေးလိုက်မည် ဆိုလျှင် passcode ဆိုတဲ့ variable ကို အခြား value တွေ ထပ်ထည့်လို့ မရတော့ပါဘူး။

```
3 const passcode= 123456;
4 passcode=234;
5 console.log(passcode);
6
```

Program ကို အထက်ပါ အတိုင်းရေးပြီး run လိုက်မည် ဆိုလျှင် TypeError တက်မှာ ဖြစ်ပါတယ် ဘာကြောင့်လည်း ဆိုတော့ line number 4 မှာ 234 ဆိုသည့် data အသစ်ကို passcode variable ထဲ ထပ်ထည့် သောကြောင့် ဖြစ်ပါတယ်။ passcode ထဲမှ value အား ပြုပြင် ပြောင်းလဲ ခွင့် မပေးပါဘူး အဘယ်ကြောင့်ဆိုသော် line 3 တွင် passcode variable ကို const type ဖြင့် ကြော်ဖြော်ပြုခဲ့သောကြောင့် ဖြစ်ပါတယ်။

### Operators

Programming language တွေကို လေ့လာတဲ့ အခါမှာ operators တွေကိုလည်း သေချာ အချိန်ပေးပြီး လေ့လာရပါတယ် ဘာကြောင့်လည်း ဆိုတော့ conditions တွေကြားထဲမှာ operators တွေရဲ့ role ဟာ အဓိက အနေနဲ့ပါဝင် နေလို့ပါ။ ဥပမာ a < b ဆိုတဲ့ နေရာမှာ b သည် a ထက်ကြီးမှ true ဖြစ်မယ် ထိုကဲ့သို့သော အခြေနေတွေဟာ program ရေးတိုင်း အမြဲတမ်း ပါဝင် နေပါတယ်။

### Arithmetic Operators

Arithmetic Operators တွေ ဆိတာ ပေါင်းခြင်း နှုတ်ခြင်း မြောက်ခြင်း စားခြင်း အကြောင်းရာခြင်း ထပ်ကိန်း တင်ခြင်း စသည့် လုပ်ဆောင် ချက်တွေ အားလုံးကို arithmetic လုပ်တယ်လို့ ခေါ်ပါတယ်။

```

1 let a=10;
2 let b=5;
3
4 let add= a + b; //Addition Operator ပေါင်းရန် သုံးသည်
5 let sub= a - b; //Subtraction Operator နှုတ်ရန် သုံးသည်
6 let mul= a * b; //Multiplication Operator မြောက်ရန်သုံးသည်
7 let div= a / b; // Division Operator စားရန် သုံးသည်
8 let rem= a % b; // Remainder Operator အကြောင်းရာရန် သုံးသည်
9 let exp= a** b; // Exponent Operator နှစ်ထပ်ကိန်းကို ရှာရန် သုံးသည်
10

```

အထက်ပါ program တွင် line 8 ရဲ့ output သည် 0 ရမှာ ဖြစ်ပါတယ်။ ထိန်ည်းတူ line 9 ရဲ့ အဖြော်သည်လည်း 1000000 ရမှာ ဖြစ်ပါတယ်  $a^{**} b$  ( $10 * 10 * 10 * 10 * 10$ ) အတိုင်းဖြစ်တဲ့ 10 ကို 5 ခါ မြောက်သောကြာင့် ဖြစ်ပါတယ်။

### Operator Precedence

Operator precedence ဆိတာ ကျွန်ုပ်တို့ ရေးလိုက်တဲ့ equation or statement တစ်ခုမှာ operators ပေါင်း များစွာ ပါဝင် နိုင်ပါတယ်။ ထို operators များစွာ ထဲကမှ ဘယ် operator သည် ပထမဆုံး အလုပ် လုပ်မလဲပေါ့ ထိုကဲ့သို့ သတ်မှတ်ချင်းကို operator precedence လို့ခေါ်ပါတယ် တစ်နည်းအားဖြင့် ဦးစားပေး လုပ်ဆောင်ချက်ပေါ့။

### Grouping Operator()

Operators တွေထဲမှာ အမြင့်ဆုံး ပထမဆုံး ဦးစားပေး လုပ်ဆောင်ရမည့် operator သည် grouping operator ဖြစ်ပါတယ်။ Grouping Operator ကို အလုပ်လုပ်မည့် ပုံစံသည် မြိမ်တို့ ပထမဆုံး အလုပ် လုပ်စေလိုသည့် နေရာတွင် ( ) group လုပ်ပေးလိုက်ခြင်း ဖြစ်ပါတယ်။ ဥပမာ  $1+2+3$  တွင်  $2+3$  ကို အရင် ပေါင်းစေချင်တယ်ဆုံးရင်  $1+(2+3)$  ဟု ရေးပေး လိုက်မည် ဆုံးလျှင်  $2+3$  ကို အရင်ပေါင်းပေးပြီးမှ 1 ဖြင့် ထပ်ပေါင်းမှာ ဖြစ်ပါတယ်။

```
-- myjs.js ...
1 let a=10;
2 let b=5;
3 let c=3;
4
5 let answer= a+b *c ;
6 console.log(answer); //output 25
7
8 let answer1= a+(b*c);
9 console.log(answer1); //output 25
10
11 let answer2 = (a+b)*c ;
12 console.log(answer2); //output 45
13
```

အထက်ပါ program တွင် line 5 ရဲ့ output သည် 25 ရပါတယ် အဘယ်ကြောင့် ဆိုသော် + ထက် \* သည် ပိုပြီး operator precedence မြင့်ပါတယ်  $a+b*c$  တွင်  $b*c$  ကို အရင် အလုပ်လုပ်ပါတယ်  $a$  နှင့်ပေါင်းသည့် အတွက် output 25 ကို ရခြင်း ဖြစ်ပါတယ် line 8 သည်လည်း ထိနည်း အတိုင်းပင် ဖြစ်ပါတယ်။ သို့သော် line 11 မှာတော့ ပုံစံ ပြောင်းသွားပါပြီ  $a$  နဲ့  $b$  ကို group လုပ်ပေးထားတဲ့ အတွက် သူတို့ နှစ်ခု အရင် အလုပ် လုပ်ပါတယ်  $a+b*c$  နဲ့ အလုပ် ဆက်လုပ်ပါတယ်။ လုပ်ဆောင်ပုံတွေ မတူ တော့တဲ့ အတွက် output တွေလည်း မတူတော့ပါဘူး။

### Property Accessors

Property Accessors တွေဟာလည်း operator precedence ရဲ့ ဒုတိယ နေရာမှာ ရှုပါတယ်။ Property Accessor နှစ်မျိုး ရှုပါတယ် ပထမ တစ်ခုက (.) Dot notation ဖြစ်ပြီး [ ] Bracket notation ဖြစ်ပါတယ်။ Dot notation နဲ့ [ ] Bracket notation ကို ဘာအတွက် သုံးလဲ ဆိုတော့ object တစ်ခုရဲ့ Properties တွေကို ရယူနိုင်ဖို့ ( ဆွဲထုတ်နိုင်ဖို့ ) သုံးပါတယ်။ Object တစ်ခု ဘယ်လို့ ကြော်လဲ properties and values တွေကို ဘယ်လို့ ထည့်ရလဲ ဆိုတာ ကိုတော့ Data Type lesson တွေထဲက Object သင်ခန်းစာမှာ ပြောခဲ့ပြီး ဖြစ်ပါတယ်။

#### Dot property accessor

ရောဥယျာဉ် အားဖြင့် object တစ်ခုရဲ့ properties တွေကို access လုပ်နိုင်ဖို့ dot property ကို သုံးပါတယ်။

Syntax:

**expression.identifier**

Expression နေရာတွင် object ကိုရေးပြီး identifier နေရာတွင် property name ကိုရေးပါတယ်။ Sample program ကို အောက်မှာ ဖော်ပြထားပါတယ်။

```
let car = { property:'driving'};
console.log(car.property);
// output driving
```

Dot กို အသံးပြုပြီး properties ထဲမှ properties ကိုလည်း ထုတ်ယူလို ရပါသေးတယ်။ ရေးသားရမယ့်ပုံစံကတော့ object.property1.property2 ဖြစ်ပါတယ်။

```
let car = {
    male:{ driver:'mgmg' } ,
    female:{ driver:'susu' }
};
console.log(car.male.driver);
// output mgmg
```

အထက်ပါ program တွင် male သည် car object ရဲ့ property ဖြစ်ပြီး driver သည် male ရဲ့ property ဖြစ်ပါတယ်။ Object များထဲမှ properties များကို access လုပ်ရာတွင် properties များသည် valid identifier ဖြစ်ခဲ့မည် ဆိုလျှင် access လုပ်နိုင်သော်လည်း valid identifier မဟုတ်ဘူး ဆိုလျှင်တော့ dot ကိုသုံးပြီး access လုပ်နိုင်မှာ မဟုတ်ပါဘူး။

```
let car = {
    'name-3':'markII',
    '5':'type'
};
console.log(car.name-3);
// output NaN
```

အထက်ပါ program တွင် 'name-3' သည် valid identifier မဟုတ်ပါဘူး ထိုနည်းတူ 5 သည် လည်း valid identifier မဟုတ်ပါဘူး။ JavaScript identifier သည် case-sensitive ဖြစ်ပါတယ်။ Unicode letter , \$ , \_ and digits တွေ ပါဝင်နိုင်သော်လည်းပဲ digit တွေနဲ့တော့ စလို မရပါဘူး။ အထက်ပါ program တွင် output ထုတ်ကြည့်သည့် အခါ NaN ဆိုသည့် output ကိုရပါလိမ့်မည် NaN ဆိုတာ Not a number ကို ပြောတာပါ။ ထိုပြောနာ များကို ပြောရင်းဖို့ အတွက် square brackets property accessor ကို သုံးနိုင်ပါတယ်။

```
document.createElement('pre');
```

Document object ထဲကမှ createElement ဆိုတဲ့ method ကို dot သုံးပြီး ခေါ်စားတာပါ။ သတိပြုရန် အချက်မှာ document သည် object ဖြစ်ပြီး createElement သည် method ဖြစ်ပါသည်။

### Square Brackets Property Accessor

Square Brackets Property Accessor ကို ရေးတဲ့ ပုံစံကတော့ expression['expression'] ဖြစ်ပါတယ်။ ပထမ expression နေရာတွင် object ကိုရေးရမည် ဖြစ်ပြီး ဒုတိယ expression

နေရာတွင် string ပုံစံဖြင့် property ကိုရေးရမှာ ဖြစ်ပါတယ်။ တစ်နည်းအားဖြင့် object[property\_name] ဖြစ်သည်။ ထိုပြင် property\_name နေရာတွင် string သာမက symbol များကိုလည်း string ပုံစံဖြင့် '' ကို သုံးပြီး ရေးသား နိုင်ပါတယ်။ အောက်ပါ program တွင် အသေးစိတ် ဖော်ပြထားပါတယ်။

```
let car = {
    'name-3': 'markII',
    '5': 'type'
};
console.log(car["name-3"]);
// output markII
console.log(car["5"]);
// output type
```

## Looping

မိမိတို့ program ရေးတဲ့ အခါမှ ထပ်ခါ ထပ်ခါ ဖော်ပြရမယ့် အချက်အလက်တွေ ရှိတတ်ပါတယ် ထိုကဲ့သို့သော အခြေနေမျိုးမှာ ဆုံးလျှင် looping ကို အသုံးပြုရပါတယ်။ ပထမဆုံး အနေဖြင့် for loop ကို ဖော်ပြ ပေးသွားပါမယ်။ အောက်က program လေးကို run ကြည့်မည် ဆုံးလျှင် I love you ဆုံးသည့် output အကြောင်း 101 ထွက်ပါလိမ့်မယ်။

```
for(let i=0; i<=100 ; i++){
    console.log('I love you');
}
```

Program မှာ အချက် သုံးချက် ရှိပါတယ် initializer , condition , final-expression ဆုံးပြီး ဖြစ်ပါတယ်။ i=0 သည် initializer ဖြစ်ပါတယ် i<=100 သည် condition ဖြစ်ပါတယ် i++ သည် final-expression ဖြစ်ပါတယ်။

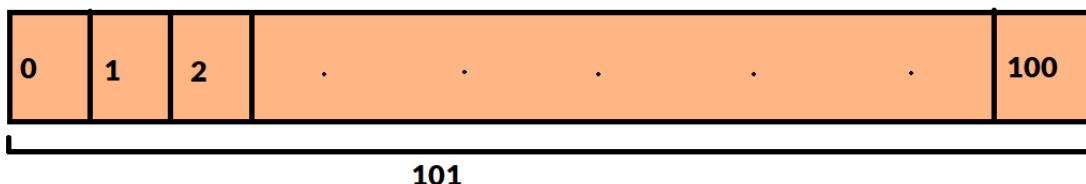
program ကို နားလည်ဖို့အတွက် အပြေးပြုပြုပဲ တစ်ခုနဲ့ နှိုင်းယူည်ပြီး ရှင်းပြ ပေးပါမယ်။ အပြေးပြုပြုပဲ တစ်ခုမှာ ဘေးလုံးကွင်းကို ပတ်ရမည် ဆုံးပါစို့ ဘယ်နှစ်ပတ် ပတ်ပြေးရမလဲ ဆုံးတာကိုတော့ i=0 initializer က ဖြတ်ပေးပါတယ်။ ဘယ်နားကနေ စပြေးရမလဲ ဆုံးတာကိုတော့ i=0 initializer က ဖော်ပြ ပေးပါတယ်။ i++ သည် final-expression ဖြစ်ပြီး တစ်ကြိမ် ပတ်တိုင်း i value ထက် 1 ပေါင်းပေးသွားတာ ဖြစ်ပါတယ်။

```

    စတဲ့နေရာ      ခံးမဲ့နေရာ
for(let i=0; i<=100 ; i++){
    console.log('I love you');
}
    တို့ဘွား မဲ လျှော့ဘွား

```

အထက်တွင် ဖော်ပြု ထားတဲ့ ပုံအရ I love you ဆိုသည့် အကြိမ် အရေအတွက် 101 ပေါ်ပေးမှာပါ။ i=0 ဆိုတော့ စမယ့်နေရာက 0 , ဆုံးမယ့် အကြိမ်အရေအတွက်က i <=100 ဆိုတော့ 100 ဖြစ်ပါတယ်။ 0- 100 ထိ ဆိုတော့ ပုံမှန် အားဖြင့် ကြည့်လျှင် အကြိမ်အရေအတွက် 100 ပဲ ပေါ်မည်ဟု ထင်နိုင်သော်လည်း 0 အတွက်ပါ ရေတွက် ကြည့်မည်ဆိုလျှင် 101 ဖြစ်သွားပါပြီ Program သည် 0 ကိုပါ ထည့်တွက်ပြီး အလုပ်လုပ်ပါတယ်။ ဥပမာ တိုက်ခန်း နံပါတ် လေးတွေကို ဥပမာ ထားပြီး စဉ်းစားကြည့်ပါ 0 ဆိုတဲ့ အခန်းက တစ်ခန်း 1 ဆိုတဲ့ အခန်းက တစ်ခန်းပါ။



နောက်ထပ် ဥက္ကစား အနေဖြင့် အောက်ပါ အတိုင်း i=5 ကနေ စမယ် ဆိုရင် I love you ဆိုသည့် စာသား ဘယ်နှစ်ကြိမ်တွက်မလဲပေါ့။

```

    ↗
for(let i=5; i<=100 ; i++){
    console.log('I love you');
}

```

အထက်ပါ အတိုင်း ရေးမည် ဆိုလျှင် i သည် 5 ကစတာ ဖြစ်ပြီး 100 မှာ ဆုံးတာ ဖြစ်သောကြောင့် အကြိမ်ရေအတွက် 96 ကြိမ် ထွက်မှာ ဖြစ်ပါတယ် အကယ်၍ 6 က စမည် ဆိုရင်တော့ 95 ကြိမ်တွက်မှာ ဖြစ်ပါတယ်။ ထိုကဲ့သို့သော စာကြောင်းရေ အရေအတွက်များကို လူကိုယ်တိုင် ရေတွက်ဖို့ ဆိုတာ ခေါ်ခဲ့လှပါတယ်။ ထို့ကြောင့် variable များကို သုံးပြီး ရေတွက် နိုင်ပါတယ်။ အထက်ပါ program မှာ ဆိုလျှင် looping တစ်ကြိမ် ပတ်သွား တိုင်း ပြောင်းလဲ နေသည့် variable တစ်ခု ရှိပါတယ် ငါးသည် i ဖြစ်ပါတယ်။ အောက်ပါ ပုံအတိုင်း i တန်ဖိုးကို ထည့်ပြီး Print ထုတ်ကြည့်မည်ဆိုလျှင် number များကိုတွေ့ရပါမည်။

```
for(let i=5; i<=100 ; i++){
    console.log('I love you',i);
}
```



**output :**

```
- - - - -
PS C:\Pro\Bok\js> node loop.js
I love you 5
I love you 6
I love you 7
I love you 8
I love you 9
I love you 96
I love you 97
I love you 98
I love you 99
I love you 100
```

Output သည် 100 မှာ ဆုံးသွားပါတယ် ထိန်ပါတ်များသည် looping တစ်ကြိမ်  
ပတ်တိုင်းမှာ တိုးသွားတဲ့ i ရဲ့ တန်ဘိုးတွေ ဖြစ်ပါတယ် တိုးသွားရခြင်းမှာလည်း i++  
ကြောင့် ဖြစ်ပါတယ်။ အကြိမ်အရေအတွက် ဘယ်လောက် အလုပ် လုပ်သွားလဲ ဆိုတာကို  
အောက်ပါ အတိုင်း variable တစ်လုံးကို အသုံးပြုပြီး ဖော်ပြု နိုင်ပါတယ်။

```
let count=0;
for(let i=5; i<=100 ; i++){
    count+=1; ← looping တစ်ကြိမ်ပတ်တိုင်း 1 ပေါင်းစပ်
    console.log('I love you',i,' Counting',count);
}
```



Count += 1 သည် count = count +1 နှင့် အတူတူပင် ဖြစ်သည် count ဆိတဲ့ variable ထဲကို 1  
ပေါင်းထည့်ပေးတာပါ။ ဥပမာ count ထဲမှာ မူရင်း value အနေနဲ့ 1 ရှိနေပြီးသားဆိုလျှင် count  
+=1 ကို run ပြီးနောက် count value သည် 2 ဖြစ်သွားမှာပါ။ အထက်ပါ program မှာလည်း  
ထိန်ည်းအတိုင်း ဖြစ်ပါတယ်။

```
I love you 5 Counting 1
I love you 6 Counting 2
I love you 7 Counting 3
I love you 8 Counting 4
I love you 9 Counting 5
```

```

1 love you 94 Counting 90
I love you 95 Counting 91
I love you 96 Counting 92
I love you 97 Counting 93
I love you 98 Counting 94
I love you 99 Counting 95
I love you 100 Counting 96

```

အထက်ပါ နည်းလမ်းကို အသုံးပြုပြီးနောက်ထပ် program  
တစ်ပုဒ်ထပ်ရေးကြည့်ပါမည်။

```

1 let city=['ygn','mdy','pol','tg','kalaw'];
2 let cities='Cities are ';
3
4 for( let i=0; i<city.length ; i++){
5   cities += city[i] + ' , ';
6 }
7 console.log(cities);
8

```

အထက်ပါ program တွင် line 1 ၏ list တစ်ခုကို ကြော်ပြုထားပါတယ် ထို့ list ထဲတွင် မြှုများနံမည်ကို ရေးထားပါတယ်။ line 2 တွင် cities ဆိုတဲ့ variable တစ်ခုကို ကြော်ပြုထားပြီး ထို့ variable ထဲတွင် Cities are ဆိုတဲ့ value ကို ထည့်ထားပါတယ်။ line 4 တွင် looping ပေတ်ပါတယ် looping ထဲမှ condition တွင် city.length ကိုသုံးထားပါတယ် city array ရဲ့ length ကို ရယူ လိုက်တာပါ။ city array ထဲတွင် data 5 ခု ရှိတယ် ဆိုတော့ city.length သည် 5 ဖြစ်မှာပါ ထို့ကြောင့် condition သည်  $i < 5$  ဆိုပြီး ရမှာပါ။ နောက်က expression နေရာတွင်  $i++$  လို့ ရေးထားတာ ဖြစ်တဲ့ အတွက်  $i$  တန်ခိုးကို 1 ပေါင်းပြီး အလုပ်လုပ်သွားမှာပါ။ line 5 တွင် ညီမျှခြင်းရဲ့ ညာဘက်ကို စကြည့်ပါမယ်  $city[i]$  သည် ပထမ အကြိမ် looping ပတ်တဲ့ အချင်း၌  $i$  တန်ခိုးသည် 0 ဖြစ်နေတဲ့ အတွက်  $city[ 0 ]$  ဆိုပြီး ဖြစ်နေမှာပါ။ city array ထဲတွင် 0 နေရာသည်  $ygn$  ဖြစ်ပါသည် ထို့ကြောင့်  $city[ 0 ] = ygn$  ဟု မှတ်ထား နိုင်ပါတယ်။  $city[i]$  နောက်တွင်  $+ ' , '$  ကို ထပ်တွေ့ရတဲ့ အတွက်  $ygn$  , ဆိုပြီး ဖြစ်နေမှာပါ ထို  $ygn$  , ကို cities ဆိုတဲ့ variable ထဲသို့  $+=$  ကိုသုံးပြီးပေါင်းထည့်ပါတယ်။ ထို့ကြောင့် ပထမ တစ်ကြိမ် looping ပတ်ပြီး အချင်းတွင် output အနေဖြင့် **Cities are ygn ,** ဆိုသည့် output ကို ရရှိမှာပါ။

ဒုတိယ အကြိမ် မှာလည်း ထိနည်းအတိုင်းပဲ လုပ်ဆောင်သွားမှာပါ။ condition သည်  $i < city.length$  ဟုရေးထားပြီး less than operator ကိုသုံးထားသည့်အတွက်  $i$  တန်ခိုး 4 ရောက်တဲ့ အထိ အလုပ် လုပ်မှာပါ။  $i$  တန်ခိုးသည် 0 ကနေ စပြီး 4 ထိ အလုပ်လုပ်မှာ ဖြစ်တဲ့ အတွက် စုစုပေါင်း looping အကြိမ်ရေအတွက် 5 ကြိမ် အလုပ်လုပ်မှာ ဖြစ်ပါတယ်။ အောက်ပါ အတိုင်း ရေးကြည့်ပြီး output ဖြင့် ယုံကြည့်ပါက ပုံမှိ နားလည် လာမှာ ဖြစ်ပါတယ်။

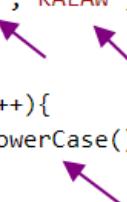
```
for( let i=0; i<city.length ; i++){
    cities += city[i] + ' , ';
    console.log(cities);
}
```

```
PS C:\Pro\Bok\js> node loop.js
Cities are ygn ,
Cities are ygn , mdy ,
Cities are ygn , mdy , pol ,
Cities are ygn , mdy , pol , tg ,
Cities are ygn , mdy , pol , tg , kalaw ,
```

### toLowerCase()

`toLowerCase()` method ဟာ စာသားတွေကို `toUpperCase` မှ တစ်ဆင့် `lowerCase` သို့  
ပြောင်းပေးပါတယ်။ ရေးသားတဲ့ ပုံစံကတော့ `string.toLowerCase()` အတိုင်း ဖြစ်ပါတယ် ထို့  
method အတွင်းသို့ မည်သည့် parameter မှ ထည့်စရာ မလိုပါဘူး။

```
let city=[ 'YGN' , 'MDY' , 'POL' , 'TG' , 'KALAW' ];
let cities='Cities are ';
```



```
for( let i=0; i<city.length ; i++){
    cities += city[i].toLocaleLowerCase() + ' , ';
    console.log(cities);
}
```

အထက်ပါ program ထဲတွင် city array ထဲ၏ data များကို `UpperCase` ဖြင့် ဖော်ပြထား  
ပါတယ် ထို့ array ထဲတွင်ရှိသော data အားလုံးကို `lowerCase` သို့ပြောင်းပြီး output ပြန်  
ထုတ်ပြ ထားပါတယ်။

```
Cities are ygn ,
Cities are ygn , mdy ,
Cities are ygn , mdy , pol ,
Cities are ygn , mdy , pol , tg ,
Cities are ygn , mdy , pol , tg , kalaw ,
```

### split()

`split()` method ကတော့ `string` ကနေပြီး ကိုယ်မလိုချင်တဲ့ စာသားများ or symbol  
များကို ဖျက်ထုတ်ကာ ကျိန်သည့် စာသားများကို array တစ်ခု အနေဖြင့် ပြန်လည်  
သိမ်းဆည်းထား ပါတယ်။ `split()` method ထဲမှာ parameter နှစ်ခု ထည့်နိုင်ပါတယ် ပထမ  
တစ်ခုသည် မိမိ ဖျက် လုံသည့် စာလုံးဖြစ်ပြီး ဒုတိယ တစ်ခုသည် limiter ဖြစ်သည်  
မိမိလိုချင်သည့် data အရေအတွက်ကို ကန့်သတ်ထားတာပါ။ ဥပမာ data တွေ အများကြီး  
ထဲမှ 3 ခုပဲ လိုချင်တယ် ဆုံးရင် ဒုတိယ Parameter နေရာတွင် 3 လုံးရေးပေး နိုင်ပါတယ်။

```

let str = 'WinHtut GreenHackers VijjA';
let arr = str.split(' ');
console.log(arr);
//output
//[ 'WinHtut', 'GreenHackers', 'VijjA' ]

```

အထက်ပါ program တွင် စာလုံးစု တစ်ခုနဲ့ တစ်ခုကြား space များကို  
ဖျက်ထုတ်ထားပြီး data များကို ရယူကာ array တစ်ခုတည်းမှာ ထည့်ပေးထားတာ  
ဖြစ်ပါတယ်။ space and special character တွေကို ဖြတ်ထုတ်တဲ့ အခါမှာတော့ split method  
သည် အဆင့်ပြုပြီး string အထူးစာရွက်များကို ဖြတ်ထုတ်တဲ့ အခါကြရင်တော့ string output  
ပုံစံ ချိန်းသွားပါတယ်။ အောက်ပါ program မှာတော့ limit ကို အသုံးပြုပြု ထားပါတယ်။

```

let str = 'WinHtut VijjA GreenHackers Myanmar programmer';
let arr = str.split(' ',4);
console.log(arr);
//output
//[ 'WinHtut', 'VijjA', 'GreenHackers', 'Myanmar' ]

```

အောက်ပါ Program လေးကိုလည်း စမ်းရေးကြည့်သင့်ပါတယ် space ကို ဖျက်ထုတ်တုန်းက နဲ့  
ပုံစံ မတူတာကို ကွဲပြားစွာ မြင်ရမှာပါ။

```

let str = 'WinHtut VijjA GreenHackers Myanmar programmer hello';
let arr = str.split('VijjA',4);
console.log(arr);
//output
//[ 'WinHtut ', ' GreenHackers Myanmar programmer hello' ]

```

အထက်တွင် သင်ကြားပြီးသော for loop , toLowerCase() and Split တို့ကို သုံးပြီး  
project လေး တစ်ခု ကို ရေးပြု ပေးပါမယ်။

Program ရဲ့ အဓိက အလုပ်လုပ်မယ့် အပိုင်းက အထက်ပါ အတိုင်း line 3 တွင် user  
ဆိုက ထည့်ပေးလိုက်တဲ့ data ကို အရင်ခုံး lowercase ပြောင်းလိုက်ပါတယ် line 5 တွင်  
contact ထဲမှ အချက်အလက်များကို ဆွဲထုတ်ဖို့အတွက် looping ပတ်ပါတယ် contact array  
ထဲမှ ပထမဆုံး data ကို ဆွဲထုတ်ပြီး ထို data ကို split လုပ်ပါတယ် split လုပ်လိုက်တဲ့ အတွက်  
data နှစ်ခု ထွက်လာပါ ထွက်လာတဲ့ data တွေကို splitContact ဆိုတဲ့ array ထဲသို့  
ထည့်လိုက်ပါတယ်။

```
splitContact = contacts[i].split( ':' );
```

ထိုနောက် user ထည့်ပေးလိုက်သော data နှင့် splitContact ထဲမှ ပထမဆုံး data နှင့်  
တူညီသလား တိုက်စစ်ပါတယ်။ တူညီတယ်ဆုံးရင် အချက်လက်များကို ပြန်လည် ဖော်ပြ

ပေးပြီး မတူညီသေးဘူး ဆိုလျင် နောက်ထပ် တစ်ကြိမ် ထပ်စစ်ပါတယ်။ နောက်ဆုံး လုံး၊ မတွေ့တော့ဘူး ဆိုလျင်တော့ မတွေဘူး ဆိုတဲ့ အကြောင်းကို ပြန် ပြ ပေးလိုက်ပါမယ်။

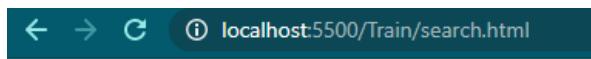
```
const contacts = ['win:2232322', 'htut:3453456', 'vij:7654322', 'ja:9998769', 'mr:9384975'];
let input='winwin';
let searchName = input.toLowerCase();

for (let i = 0; i < contacts.length; i++) {
    let splitContact = contacts[i].split(':');
    if (splitContact[0].toLowerCase() === searchName) {
        console.log( splitContact[0] + '\'s number is ' + splitContact[1] + '.');
        break;
    }
    if( i === contacts.length-1){
        console.log('Contact not found');
    }
}
```

အထက်ပါ program ကို html page တစ်ခုနဲ့ ချိတ်ဆက်ပြီး မြိမ်တို့ ဘာသာ program ကို ရေးကြည့် သင့်ပါတယ်။

```
<body>
    <label for="serach">Search Here:</label>
    <input type="text" id="serach">
    <button>Search</button>
    <p></p>
</body>
<script>
let data=['win:12345','htut:345622','Vij:@#@#$$', 'Ja:78yu'];
let para = document.querySelector('p');
let input=document.querySelector('input');
let button=document.querySelector('button');
button.addEventListener('click',function(){
    let searchData = input.value.toLowerCase();
    for( let i=0 ; i <data.length ; i++ ){
        let splitContact=data[i].split(':');
        //splitContat=['win','12345']
        if(splitContact[0].toLowerCase() === searchData){
            para.textContent=splitContact[0] + ' Number is ' +splitContact[1];
            break;
        }
        if( i==data.length-1){
            para.textContent= searchData+' Is not found!';
        }
    }
});
</script>
</html>
```

**Output**



Search Here:

Vij Number is @#@#\$\$

### While and do.....while

While loop ကိုလည်း for loop အတိုင်းပင် အကြမ်းကြံမ်း အလုပ်လုပ် နိုင်ဖို့ အတွက် အသုံးပြုပါတယ်။ သို့သော် while(boolean condition) while နောက်က parentheses ထဲတွင် boolean condition တစ်ခုသာ ပါဝင်ပါတယ်။ အကယ်၍ ထို boolean condition true ဖြစ်နေခဲ့မည် ဆိုလျှင် {} curly braces ထဲမှာ program များ အလုပ်လုပ်ပြီး false ဖြစ်နေခဲ့မည် ဆိုလျှင်တော့ while loop ထဲက code များ အလုပ်လုပ်မှာ မဟုတ်ပါဘူး။

Syntax:

```
while( boolean condition ) {
    //program
}
```

looping.js

```
let i=10;
while( i < 100){
    console.log('i value is lessthan 100');
}
```

အထက်ပါ program လေးသည် while loop ရဲ့ အလုပ်လုပ်တဲ့ ပုံစံ ဖြစ်ပါတယ်။ program ကို run ကြည့်မည် ဆိုလျှင် ရပ်မှာ မဟုတ်ပါဘူး အဘယ်ကြောင့် ဆိုသော် while ထဲတွင် အဆုံး မှတ် မပါ ပါဘူး ထို့ကြောင့် အဆုံးမှတ်ကို programmer က ထည့်ပေးရမှာပါ။ ဥပမာ အထက်ပါ program ကို အဆုံးမှတ် ထည့်ချင်တယ် ဆုံးရင် အောက်ပါ အတိုင်း ရေးသား ရိုင်ပါတယ်။

```
let i=10;
while( i < 100){
    console.log('i value is lessthan 100');
    i+=10;
}
```

ယခု အတိုင်းရေးလိုက်မည် ဆိုလျှင် looping တစ်ကြိမ် ပတ်တိုင်းမှာ । တန်ဖိုး ထဲကို 10 ပေါင်းထည့် သွားမှာ ဖြစ်ပါတယ်။ ထို့ကြောင့် । တန်ဖိုး 90 ရောက်ပြီ ဆုံးရင် while နောက်က condition false ဖြစ်သွားတဲ့ အတွက် looping က အလုပ် မလုပ်တော့ပဲ ရပ်သွားမှာ ဖြစ်ပါတယ်။ while loop ကို ပိုမို ကျွမ်းကျင်စေရန် အောက်ပါ program ကဲ့သို့ program

ယောက်ပါ program မှာတော့ JavaScript က data လေးတွေကို looping ပတ်ပြီး html ဆီသို့ လှမ်းပို့ပေးထားပါတယ် tag ရဲ့ id ကိုသုံးပြီးပို့တာ ဖြစ်တဲ့အတွက် getElementById ကို သုံးထားပါတယ်။

```

<body>
    <p id='while'></p>
</body>
<script>
    let i=0;
    let data='';
    while(i<5){
        data+=`First Time ${i+1}<br>`;
        i++;
    }
    document.getElementById('while').innerHTML=data;
</script>
</html>

```

Do...while loop ကတော့ while loop နဲ့ အနည်းငယ် တူပါတယ် သို့သော် do...while loop သည် condition ကို နောက်ဆုံးမှာ ရေးပါတယ်။ do...while loop ရဲ့ ထူးခြားချက်သည် condition မ မှန်ခဲ့ရင်တောင် တစ်ကြိမ်တော့ အလုပ် လုပ်ပါသေးတယ်။ ဘာကြောင့်လဲ ဆုံးတော့ သူရဲ့ syntax ရေးသားပုံကြောင့်ပါ။

Syntax:

```

do{
    //program
}while(condition);

```

အောက်ပါ program တွင် do...while loop ရဲ့ အလုပ်လုပ်ပုံကို သေချာ မြင်အောင် ပြပေးထားပါတယ်။ i တန်သိုးသည် ၆ ဖြစ်နေပြီး condition မ မှန်တာ တောင် program တစ်ကြိမ် အလုပ်လုပ်သွားပါသေးတယ်။

```

<body>
    <p id='while'></p>
</body>
<script>
    let i=6;
    let data='';
    do{
        data+=`First Time ${i+1}<br>`;
        i++;
    }
    document.getElementById('while').innerHTML=data;
}while(i<5);
</script>
</html>

```

Output:

## First Time 6

### Function()

Function ဆိတ် လုပ်ဆောင် ချက် တစ်ခုပဲ။ ဥပမာ ထမင်းပေါင်းအိုး function ဆိုပါစို့ သူသည် ထမင်း ချက်ပေးတယ်။ ထမင်းပေါင်းအိုးသည် ထမင်းချက်ပေးတဲ့ function(လုပ်ဆောင်ချက်) ဖြစ်တယ်။ ဥပမာ dog function ဆိုပါစို့ dog သည် ဟောင်တယ် ဟောင်ခြင်းသည် သူရဲ့ လုပ်ဆောင် ချက် function ဖြစ်တယ်။ Function မှာ နှစ်မျိုးရှိပါတယ် standard library function နဲ့ programmer defined function တို့ဖြစ်တယ် JavaScript မှာဆိုရင်တော့ Standard library function လို့မပြောပဲ built-in browser function လို့ပြောနိုင်ပါတယ်။ Programmer defined function ဆိုတာ ကတော့ ကျဉ်တော်တို့လို့ program ရေးသားတဲ့ သူတွေ ရေးထားတဲ့ function တွေ ကို ပြောတာပါ။ ဒါဆိုရင် built-in browser function တွေကို ဘယ်သူတွေ ရေးသလဲပေါ့ ထို function တွေကိုလည်း programmer or programming language ကို ထွင်တဲ့ သူတွေကပဲ ကြံးတင် ရေးပေးခဲ့ကြတာပါ။ ဥပမာ math.random() ပေါ့ random ကို programming language ထွင်တဲ့ သူတွေက ကြံးတင် ရေးပေးခဲ့ကြတာပါ ကျဉ်တော်တို့က random number လိုချင်ရင် ကိုယ်တိုင် ရေးနေ စရာ မလိုပဲ သူတို့ ရေးထားတဲ့ function တွေကို ခေါ်သုံး လိုက်ရှုပါပဲ ထို function တွေကို standard library function သို့မဟုတ် built-in browser functions တွေလို့ ခေါ်ဆိုခြင်း ဖြစ်ပါတယ်။ ဥပမာ replace function ပေါ့ သူက parameter နှစ်ခု ယူတယ် ပထမ တစ်ခုက အစားထိုးချင်တဲ့ စာတန်း နေရာ ဒုတိယ တစ်ခုကတော့ အစားထိုးမယ် စာ အသစ်။ replace ရဲ့ လုပ်ဆောင်ချက်ကလည်း data တွေကို အသစ်နဲ့ အစားထိုးပေးတာပါ။

```
let contact='information data';
let newContact = contact.replace('data','JSON');
console.log(newContact);
//output
//information JSON
```



အထက်ပါ program မှာဆိုရင် replace သည် built-in function ဖြစ်ပြီး information data ဆိုတဲ့ စာသားထဲက data ဆိုတဲ့ စာသား နေရာမှာ JSON ဆိုတဲ့ စာသား နဲ့ အစားထိုးပေးတာပါ။ ထိုနောက် replace function ရဲ့ return အနေနဲ့ new string အသစ် တစ်ခုကိုပြန်ပေးပါတယ်။

### Own Function

ကိုယ်တိုင် function တစ်ခုကို create လုပ်ဖို့ အတွက်ဆိုရင် function တစ်ခုကို အရင်ဆုံး ကြော်ပေးဖို့ လိုအပ်ပါတယ်။ ထိုသို့ ကြော်ရာမှာလည်း ရှေ့ဆုံးကန် function ဆိုတဲ့

keyword ကို အသုံးပြုပါတယ် ထို့နောက်မှ function name ကိုရေးပေးရပါတယ်။ function name နောက်တွင် parenthesis တစ်ခု ရှိပါတယ် ပြီးမှ curly brace ဖြင့် မိမိတို့ရေးသားလိုသာ program ကို ရေးသား ပါတယ်။

Syntax:

```
function myFun( ){
    //my program
}
```

myFun နောက်က parenthesis ထဲတွင် parameter တွေကိုထည့်နိုင်ပါတယ်။ နောက်သင်ခန်း စာများ example program ရေးပြု သွားပါမယ်။ function တွေကို ကြော်ပြီးရှုသူထဲမှာ program တွေ ရေးသားထားရှုနဲ့ function က အလုပ် လုပ်မှာ မဟုတ်ပါဘူး။ Function အလုပ်လုပ် ဖို့ active ဖြစ်ဖို့အတွက် function name ကို ပြန်ခေါ်ပေးရပါတယ်။

```
1 let contact='information data';
2
3 function myFun(){
4     console.log(contact);
5 }
6 myFun();
7 //output
8 //information data
```

အထက်ပါ program မှာ ဆိုလျှင် myFun သည် function ရဲ့ name ဖြစ်ပါတယ်။ line number 5 ထိ function က သာ အလုပ်လုပ်မှ လုပ်းမှာ မဟုတ်ပါဘူး။ line 6 ကြော် function name ကို ပြန်ခေါ်တဲ့ အချိန်မှသာလျှင် function က အလုပ် ထလုပ်ပါတယ်။ Function ကို ပြန်ခေါ်ဖို့ အတွက်လည်း function name ကို ရေးလိုက်ရှုပဲ ဖြစ်ပါတယ် အကယ်၍ အခေါ်ခံရမယ့် function က parameter တွေ ရှိနေမယ်ဆိုရင် function call ကို ခေါ်တဲ့ နေရာကနေလည်း function name အပြင် parameters တွေကို ထည့်ပေးလိုက်ရမှာ ဖြစ်ပါတယ်။

```
1 function myFun(){ ←
2     console.log('Parameter not included function');
3 }
4 function fun2(a,b){ ←
5     let c=a+b;
6     return c; ←
7 }
8 myFun(); ←
9 console.log(fun2(10,20)); ←
10
```

အထက်ပါ program မှာဆိုလျင် function နှစ်ခုရေးထားပါတယ် ပထမ တစ်ခုသည် myFun() ဖြစ်ပြီး ထို function စာကြောင်း တစ်ကြောင်း ဖော်ပေးတဲ့ လုပ်ဆောင်ချက်က တစ်ခုတည်းပါပဲ။ ဒုတိယ အနေဖြင့် fun2 ဆိုတဲ့ function တစ်ခုကို ထပ်ရေးထားပါတယ်။ ထို function ထဲမှာတော့ parameter နှစ်ခုပါ ထည့်ကြော်ပေးတယ်။ line 9 မှာ fun2 function ကို ခေါ်လိုက်ပါတယ်။ ခေါ်တဲ့ အချိန်မှာလည်း 10 and 20 ကို ထည့်ပေးလိုက်ပါတယ်။ line 9 ကနေ ခေါ်တဲ့ အတွက် line 4 က fun2 ကိုသွားခေါ်ပါတယ် a , b နေရာတို့တွင် 10 , 20 စတဲ့ values တွေ အစဉ်လိုက် နေရာယူပါတယ်။ ထို့ကြောင့် a=10 ဖြစ်ပြီး b=20 ဖြစ်သွားပါတယ်။ line 5 မှာ a and b ကိုပေါင်းပြီး ရလဒ်ကို c ထဲသို့ ထည့်ပေးလိုက်ပါတယ်။ ထို့နောက် line 6 တွင် return keyword ကို သုံးပြီး c တန်သုံးကို return ပြန်ပေးလိုက်ပါတယ်။ ထို့ကြောင့် line 9 က function call ခေါ်တဲ့ နေရာတွင် c တန်သုံးကို ပြန်ရလေမှာ ဖြစ်ပါတယ်။ ခေါ်တဲ့ နေရာကို return value ပြန်ပေးလိုက်တဲ့ သဘောပါ။ ထို့ကြောင့် line 9 ရဲ့ Output သည် 30 ဖြစ်နေသည်ကိုမြင်ရမှာပါ။

#### Anonymous Function()

Anonymous function ဆိုတာ ကတော့ function name မပါ ပါဘူး။ function name မပါတဲ့ အတွက် သူ့ကို ခေါ်သုံးလို့ မရပါဘူး ထိုကဲ့သို့ ခေါ်သုံးလို့ မရတဲ့ အတွက် သူဟာ တစ်ခြား function တွေရဲ့ ပုံစံနဲ့ လုပ်ဆောင်ချက်ချင်း မတူညီပါဘူး။ အောက်က ပုံမှာ သူတို့ နှစ်ခုရဲ့ ကဲ့ပြားချက်ကို ဖော်ပြထားပါတယ်။

```
function myFunction() { //normal function
    alert('hello');
}

function() { //anonymous function
    alert('hello');
}
```

အပေါ်က function သည် function name ပါတဲ့ အတွက် normal function ဖြစ်ပြီး အောက်က function သည် function name မပါတဲ့ အတွက် anonymous function ဖြစ်ပါတယ်။ anonymous function ကို ရှေ့ပိုင်း သင်ခန်းစာတွေမှာ အသုံးပြုခဲ့ပြီး ဖြစ်ပါတယ်။

```

let but = document.querySelector('button');
but.onclick = function() {
  console.log('User was clicked button');
}

```

Button ကို click ခေါက်လိုက်တဲ့ function ကို အလုပ် စလုပ် စေပါတယ်။ သူမှာ name မရှိတဲ့ အတွက် ခေါ်သုံးလိုတော့ မရပါဘူး။

### Arrow Function

Arrow function ကိုသုံးခြင်း အားဖြင့် code အနဲ့ဆုံးနဲ့ အရိုးရှင်းဆုံး ပုံစံ ဖြစ်အောင် ရေးသား နိုင်ပါတယ်။

```

//normal function vs arrow function
let a=10;
let b=20;
function myFun(a,b) {
  return a+b;
}
//1
(a,b)=>{           function keyword , function name နှစ်ခုလုံး မပါပါဘူး
  return a+b;
}
//2
(a)=>a+b;         return ဆိုတဲ့ keyword ပါ မပါတော့ပါဘူး
                   curly brace လည်း မပါတော့ပါဘူး
//3
a =>a+b;          parameter အတွက် parentheses လည်း မပါတော့ပါဘူး

```

## Objects

Objects တွေဟာ JavaScript အတွက်အရေးကြီးဆုံးနဲ့ အခြေခံ အကျခုံး datatype ဖြစ်ပါတယ်။ JavaScript objects တွေဟာ dynamic - properties တွေဖြစ်ပြီး ထပ်ထည့်တာတွေ ဖျက်တာတွေ ပြုပြင်တာတွေ လုပ်လို့ ရပါတယ်။ Object ဆိုတာ အချက်အလက်တွေနဲ့ လုပ်ဆောင် ချက်တွေကို စုပေါင်းထားတာလည်း ဖြစ်ပါတယ်။ အချက်အလက်တွေနဲ့ လုပ်ဆောင်ချက်တွေကို variables and functions တွဲလို့ ခေါ်ပြီး တစ်နည်းအားဖြင့် properties and methods တွဲလို့လည်း ခေါ်ပါတယ်။

## Object Literals

JavaScript မှာ Object တစ်ခု ဖန်တီးဖို့ဆိုတာ အလွန်လွယ်ကူပြီး object ဖန်တီးရန် JavaScript code ထဲမှာ object literal တွေ ထည့်ပေးလိုက်ရင် ဖန်တီးပြီးသား ဖြစ်ပါပြီ။ Object literal ဆိုတာက object တစ်ခုအတွင်းမှာ ပါဝင်မယ့် list တွေကို comma-separated လုပ်ထားခြင်း၊ name:value pairs တွေကြားမှာ colon ခြားထားခြင်း အပြင် list အားလုံးကို curly braces ထဲတွင် ထည့်ထားခြင်းတို့ ဖြစ်ပါတယ်။ JavaScript မှာ အခြား သော programming အချို့နဲ့ မတူတာက property name တွေနေရာမှာ string literal တွေ အပြင် JavaScript identifier name တွေကိုပါ ပေးလို့ ရပါတယ်။ ထိုပြင် object တွေ

တည်ဆောက်ရာမှာ အထဲတွင် ပါဝင်မည့် properties များ မထည့်ပဲလည်း တည်ဆောက်လို ရနိုင်ပါသည်။ Object ထဲမှာ ရှိတဲ့ properties တွေရဲ့ value များအား ရရှိလိုလျှင် dot(.) or square bracket[] တိုကိုအသုံးပြု၍ နိုင်ပါတယ်။ အောက်တွင် example ကို အသေးစိတ် ဖော်ပြပေးထားသည်။

### Object Example

```
let emptyObject = {}; // object without properties

let numPro= {x:0 , y:0};// object with numeric properties

let num = {x:numPro.x , y:numPro.y};

let jsObject = {

    "booktile":"JavaScript" ,

    "author":"anonymous" ,

    for:"beginner" ,

    author:{

        firstname:"anon" ,

        sirname:"ymous"

    }

};

//using JS identifier "for" with dot()

console.log(jsObject.for);

//using JS identifier "for" with bracket[ ]

console.log(jsObject["for"]);
```

```
console.log(jsObject.author.firstname);

console.log(jsObject["author"]["firstname"]);
```

### Object Example 2

```
const ncc = {

    name:["national","cyber","city"],

    year:2022,

    ambition:function(){

        console.log(` ${this.name[0]} ${this.name[1]} is ${this.year} years
old`);

    },

    subject:function(){

        console.log(`Hi i'm ${this.name[0]}`)

    },

};

ncc.ambition();

ncc.subject();
```

အထက်ပါ program ကို ကြည့်မည်ဆုံလျှင် properties များလည်း ပါဝင်သလို ambition and subject စသည့် methods များလည်း ပါဝင်နေသည်ကို တွေ့ရမည်။ ambition method ထဲမှ this.name[0] တွင် this သည် ယခု object ကိုပင် ပြန်လည် ရည်ညွှန်းတာဖြစ်ပါတယ်။ အကယ်၍ method များရေးရာတွင် shorter syntax ပုံစံဖြင့် ရေးလိုပါကလည်း အောက်ပါ အတိုင်း ရေးနိုင်ပါသေးတယ်။

### Object Example 3

```

const ncc = {

    name:["national","cyber","city"],

    year:2022,

    ambition(){

        console.log(` ${this.name[0]} ${this.name[1]} is ${this.year} years
old`);

    },

    subject(){

        console.log(`Hi i'm ${this.name[0]}`)

    },

};

ncc.ambition();

ncc.subject();

```

**Output**

```

national cyber is 2022 years old
Hi i'm national

```

အထက်ပါ program တွင် dot notation ကိုသုံးထားပါတယ် dot notation ( . ) ဆိုတာက object တစ်ခုတဲ့မှာ ရှိတဲ့ properties and method တွေကို ခေါ်သုံးချင်တဲ့ အခါမှ အသုံးပြုပါတယ်။ အထက်ပါ program တွင်မူ ncc namespace ပုံစံဖြင့် လုပ်ဆောင်ပါတယ်။ ဆုလိုသည်မှာ object ထဲမှ properties or method တစ်ခုခုကို access ရယူလိုလျှင် ထိ ncc ကို ရှုံးက အရင် ရေးရပါတယ်။

JavaScript ရဲ့ object သည် object တစ်ခုရဲ့ properties အဖြစ်လည်း လုပ်ဆောင် နိုင်ပါသေးတယ်။ အောက်တွင်ကြည့်ပါ။

```
const ncc = {
    name: {
        firstname: "National",
        secondname: "CyberCity",
    },
};

console.log(ncc.name.firstname);
```

အထက်ပါ program တွင် output အနေဖြင့် National ကို ပြန်ရပါမည်။ အထက်ပါ program များတွင်လည်း dot(.) နှင့် bracket notation များကို သင်ခဲ့ပြီး ဖြစ်ပါတယ် ပုံမှန်အားဖြင့် ထို နှစ်ခုသည် တူသည်ဟု ထင်ရသော်လည်း ပိုမိုသံရမည့် အသုံးပြုပုံ အချို့ ရှိနေသေးသည်။ အချို့နေရာများတွင် dot ကို သုံးလို့ မရပဲ bracket ကိုသာ မသုံးမဖြစ် အသုံးပြုရမည့် နေရာများလည်း ရှိနေသေးသည်။

#### Object Example 4

```
// essential bracket notation

const ncc = {
    name: ["natioanl", "CyberCity"],
    year: 2022,
};

function property(propertyName){
    console.log(ncc[propertyName]);
```

```

}

property("name");

property("year");

```

**Output**

```

[ 'natioanl', 'CyberCity' ]
2022

```

အထက်ပါ Program output သည် အပေါ်ပုံအတိုင်း ထွက်ပါသည်။ မှတ်သားရန်  
တစ်ခုမှာ object တစ်ခုရဲ့ property name သည် variable အဖြစ် အသုံးပြုခဲ့ရသော  
အချိန်မျိုးတွင် dot notation ကို အသုံးပြုမရနိုင်တော့ပါ ထိုကဲ့သို့သော အခြေနေမျိုးတွင်  
bracket notation ကိုသာ property ရဲ့ value များကို access လုပ်ရန် သုံးရ ပါတော့သည်။

**new**

JavaScript မှာ new ဆိုသည့် keyword ကို သုံးပြီး objects တွေ ဖန်တီးလို  
ရပါသေးတယ်။ new ဆိုတဲ့ keyword နောက်မှာ ဖန်တီးလိုသည့် object function လာပါတယ်။

**ဥပမာ** let ob = new Object();

Object() အား အထက်ပါ နည်းလမ်းအတိုင်း အသုံးပြုခြင်းကို function ဟု မသုံးပဲ  
constructor ဟု သုံးပါတယ်။ JavaScript မှာ built-in constructors များစွာ ရှိပါတယ်။ String() ,

`Number()` , `Boolean()` , `Object()` , `Array()` , `RegExp()` , `Function()` , `Date()` စသည်။  
ဖြစ်ပါတယ်။

### new Example

```
let num = new Number();

let ob = new Object();

let fun = new Function();

console.log(typeof(num));

console.log(typeof(ob));
```

အထက်ပါ program အား run ကြည့်မည် ဆိုလျှင် output အနေဖြင့် object များသာ ထွက်လာသည်ကိုတွေ့ရမည်။

### Express.js

Express.js အကြောင်းကို မလေ့လာခင် ဦးစွာ node.js အကြောင်းကို သိသင့် ပေသေးသည်။ JavaScript language သည် ရုံးရှင်းသော scripting language အဖြစ် စတင် လာခဲ့ပြီး ကန်ဥုးတွင် browser ထဲတွင်သာ အလုပ်လုပ်ကာ web pages တွေ ဖြစ်ပေါ် လာစေရန် လိုအပ်သော အချက်လက်များကို ကူညီ လုပ်ဆောင် ပေးပါတယ်။ သို့သော် JavaScript သည် server ပေါ်မှာတော့ တိုက်ရှိက် run နိုင်တာမျိုး မရှိသေးပါဘူး။

V8 engine သည် Google Chrome web browser ထဲတွင် JavaScript အလုပ်လုပ် နိုင်စေရန် Google မှ တီထွင်ထားခြင်း ဖြစ်ပါတယ်။ ထို V8 engine အား c++ programming ကို အသုံးပြုပြီး ရေးထားခြင်း ဖြစ်ပါတယ်။ Google သည် 2008 တွင် ထို V8 engine အား open-sourced အဖြစ် ချု ပေးခဲ့ပါတယ်။ V8 engine သည် JavaScript source code တွေကို runtime မှာ native machine code အဖြစ် compile လုပ်ပေးခြင်း ဖြစ်ပါတယ်။

