



Ex;brainN

education

Ver 2.0

A Web Developer – II

Copyright @Ex;brainN Education

Web Developer II (Advance) Course

Content

1. Advance CSS3
2. Advance JavaScript
3. Bootstrap 5
4. Figma Design & Design Theory
5. jQuery
6. Vuejs
7. PHP (Pure + OOP)
8. API
9. JSON
10. Ajax
11. Database (MySQL)
12. Hosting
13. Laravel Framework

1. Advance CSS3

- 1) Gradients
- 2) Shadow
- 3) Text-Overflow & Wrap Text
- 4) Transform(2D,3D)
- 5) Transition
- 6) Animations
- 7) Column
- 8) Pseudo Class
- 9) Root & Var()
- 10) Media Queries
- 11) Flex Box
- 12) Viewport

1) Gradients

Background imageတွေကို color တရောင်တည်းမပေးပဲ အရောင်ရော၊ရောင်ပြေးတွေနဲ့ပေးချင်တဲ့အခါသုံးပါတယ်။

အမျိုးအစားနှစ်ခုရှိပါတယ်။

1. Linear Gradient
2. Radial Gradient

■ Linear Gradient

Syntax

background-image : linear-gradient(direction,color1,...,color(n))

background-image : repeating-linear-gradient(direction,color1,...,color(n))

Example:

background: linear-gradient(#e66465, #9198e5);



■ Radial Gradient

Syntax

background-image : radial-gradient(shape,color1,...,color(n))

Example:

background: radial-gradient(#e66465, #9198e5);



2) Shadow

Shadowကတော့စာသားတွေ၊ Boxတွေရဲ့အရိပ်တွေထည့်ပေးတဲ့အခါသုံးပါတယ်။

Shadowထည့်လိုက်တဲ့အတွက်ရုပ်လုံးကြွလာသလိုမြင်စေပါတယ်။

Shadow မှာတော့ နှစ်ခုရှိပါတယ်။

1. Text-Shadow
2. Box-Shadow

■ Text-Shadow

စာသားတွေကိုshadow ပေးတဲ့အခါtext-shadow property ကိုသုံးရပါတယ်။

- `text-shadow : horizontal(px) vertical(px);`
Horizontal နဲ့ Vertical ပေးပြီးshadowကိုပုံဖော်ပါတယ်။
- `text-shadow : horizontal(px) vertical(px) color;`
Default ကတော့ black ဖြစ်ပြီးတခြားအရောင်ပြောင်းချင်တဲ့အခါ
နောက်ဆုံးမှာ color valueကို ထည့်ပေးရပါတယ်။
- `text-shadow : horizontal(px) vertical(px) blur(px) color;`
ပုံမှန်အတိုင်းဆို shadow ထည့်လိုက်ရင် ထင်းထင်းကြီးဖြစ်နေပါတယ်။
အဲ့လိုမဟုတ်ပဲ အရိပ်နဲ့သဏ္ဌန်တူစေချင်တဲ့အခါ blurကို အရောင်
ရဲ့ရှေ့မှာထည့်ပေးရပါတယ်။blurဆိုတာဝါးတာကိုဆိုလိုတာပါအရောင်ကို
ဝါးလိုက်တဲ့သဘောပါပဲ။
- `text-shadow : (horizontal(px) vertical(px) blur color)n;`
Shadowအမျိုးအစားတခုတည်းမဟုတ်ပဲ multiple
ထည့်ချင်တဲ့အခါသုံးနိုင်ပါတယ်။
Example:
`text-shadow: 1px 1px 2px black, 0 0 25px blue, 0 0 5px darkblue;`
Shadowဟာ black blueနဲ့ darkblue multiple shadowဖြစ်နေပါလိမ့်မယ်။

■ Box-Shadow

စာသားမဟုတ်ပဲ elements (width,height ရှိပြီး လေးထောင်ပုံစံဖြစ်နေတဲ့) box
တွေကို shadow ထည့်ချင်တဲ့အခါမှာလည်း text-shadow
မှာသုံးခဲ့သလိုရေးပုံရေးနည်းတူမျိုးအတိုင်း အသုံးပြုနိုင်ပါတယ်။

3) Text-Overflow & Wrap Text

Overflowကိုhidden ပေးထားတဲ့အခါဘောင်ကျော်သွားတဲ့စာသားတွေကို မမြင်ရတော့ပါဘူး။ဒီအချိန်မှာကျော်သွားတဲ့စာသားတွေကို ... ဆိုတဲ့ပုံစံနဲ့မြင်ချင်တဲ့အခါ

■ Text-Overflow: ellipsis

Lorem ipsum dolor sit amet ...

အဲ့လိုမဟုတ်ပဲ ဘောင်ထဲမဝင်တဲ့ Textကို ဒီအတိုင်းထားချင်တဲ့အခါ

■ Text-Overflow: clip

Lorem ipsum dolor sit amet cor

ကိုသုံးပါတယ်။

4) Transform(2D,3D)

Transform propertyကတော့ element တွေရဲ့နဂိုမူလပုံစံကနေ rotateလုပ်တဲ့အခါx axis,y axis ပေါ်မူတည်ပြီး element ရဲ့position တွေ move လုပ်တာ၊scaleလုပ်တာ စတဲ့ animation တွေအတွက်အသုံးပြုပါတယ်။2Dနဲ့3Dဆိုပြီးထပ်ခွဲထားပါသေးတယ်။

-2Dမှာပါဝင်တဲ့ methods တွေကတော့

■translate()

Element ရဲ့position ပြောင်းချင်တဲ့အခါ သုံးပါတယ်။

■rotate()

Element ကို စောင်းချင်တာ၊လှည့်ချင်တဲ့ အခါသုံးပါ။deg နဲ့သုံးတာများပါတယ်။

■scale()

Element ရဲ့x y axisတွေကိုချဲ့ချင်တဲ့အခါ သုံးပါတယ်။

■scaleX()

Element ရဲ့x axis သီးသန့်ပြောင်းချင်တဲ့အခါသုံးပါတယ်။

■scaleY()

Element ရဲ့y axis သီးသန့်ပြောင်းချင်တဲ့အခါသုံးပါတယ်။

■skew()

Element ကို ယိုင်နေတဲ့ပုံစံဖြစ်ချင်တဲ့အခါ သုံးပါတယ်။

■skewX()

Element ရဲ့ x axis သီးသန့်ပြောင်းချင်တဲ့အခါသုံးပါတယ်။

■skewY()

Element ရဲ့ y axis သီးသန့်ပြောင်းချင်တဲ့အခါသုံးပါတယ်။

-3Dမှာပါဝင်တဲ့ methods တွေကတော့အပေါ်က 2D method တွေ အပြင်

■translateX

■translateY

■translateZ

■rotateX

■rotateY

■rotateZ တို့ပါဝင်ပါတယ်။

5) Transition

Transition ကတော့ element တွေရဲ့ property တွေကို smoothly ဖြစ်အောင် ပြောင်းချင်တဲ့အခါသုံးပါတယ်။ hover နဲ့တွဲသုံးတာများပါတယ်။ များသောအားဖြင့် delay တို့ duration တို့ကို second နဲ့သတ်မှတ်ကြပါတယ်။

သူမှာပါဝင်တဲ့ properties တွေကတော့

■ transition-delay

Transition delay ကိုပေးလိုက်တဲ့အခါ animation ဟာ ချက်ချင်းမစတော့ပဲ ပေးလိုက်တဲ့အချိန်ကြာသွားတဲ့အခါမှ စလုပ်တာပါ။

-transition-delay : 3s

■ transition-duration

Duration ကတော့ transition လုပ်နေတဲ့ကြာချိန်ကိုဆိုလိုပါတယ်။

ပေးလိုက်တဲ့ အချိန်ပေါ်မူတည်ပြီး transition လုပ်နေမှာဖြစ်ပါတယ်။

-transition-duration : 2s

■ transition-property

Property ကတော့ target ထားချင်တဲ့ point ဖြစ်ပါတယ်။ width target ထားတဲ့အခါ width ပြောင်းလဲတိုင်း transition ကအလုပ်လုပ်နေမှာဖြစ်ပါတယ်။

-transition-property : width

■ transition-timing-function

Transition အလုပ်လုပ်နေတဲ့အချိန်မှာ ပြောင်းလဲသွားတဲ့ effect တွေကို customization လုပ်ချင်တဲ့အခါ ထည့်သုံးပါတယ်။

ပြောင်းလဲလို့ရနိုင်တာတွေကတော့

■ ease

Default ဖြစ်ပြီးတော့ transition အစမှာ နှေးမယ်ပြီးရင်မြန်မယ်။
နောက်ဆုံးမှာနှေးသွားမယ်။

-transition-timing-function : ease

■ linear

အနှေးအမြန်မရှိပဲ တပြေးညီသွားမယ်။

-transition-timing-function : linear

■ ease-in

အစမှာနှေးမယ်။

-transition-timing-function : ease-in

■ ease-out

အဆုံးမှာနှေးမယ်။

-transition-timing-function : ease-out

■ ease-in-out

အစနဲ့ အဆုံးမှာနှေးမယ်။

-transition-timing-function : ease-in-out

Transition တွေကို shorten နည်းနဲ့ တကြောင်းထဲရေးချင်တဲ့အခါ

transition: properly duration timing delay
ဆိုပြီးရေးနိုင်ပါတယ်။

Example:

transition :width 2s linear 2s

6) Animation

Animation ကတော့ element tag တွေကိုလိုသလို animation property တွေသုံးပြီး စီမံတာဖြစ်ပါတယ်။ Animation flow တခုကို Manage လုပ်ဖို့အတွက် @keyframe ကိုအသုံးပြုပါတယ်။

Animation property ကိုမသွားခင် @keyframe ကိုအရင်ရှင်းပြပါမယ်။ @keyframe ကို animation ကခေါ်လို့ရအောင် name ပေးပေးရပါတယ်။

```
@keyframes slidein {  
  from {  
    transform: translateX(0%);  
  }  
  
  to {  
    transform: translateX(100%);  
  }  
}
```

■ from (0%)

From ကတော့ animation စတဲ့အခါ ဖြစ်ချင်တဲ့ block တခုဖြစ်ပါတယ်။

သူ့ထဲမှာတော့ css property တွေထည့်နိုင်ပါတယ်။ အရောင်ပြောင်းချင်တဲ့အခါ background color, အလင်းအမှောင်ချိန်ချင်တဲ့အခါ opacity အစရှိသလို ထည့်နိုင်ပါတယ်။

■ to (100%)

From နဲ့တူတူပါပဲ animation ကို ဘယ်လိုအဆုံးသတ်လဲဆိုတာရေးပေးရမှာပါ။

From to နဲ့မရေးပဲ ပိုအသေးစိတ်ကျချင်တဲ့အခါ % တွေနဲ့လည်းရေးနိုင်ပါတယ်။

```
@keyframes identifier {  
  0% { top: 0; left: 0; }  
  30% { top: 50px; }  
  68%, 72% { left: 50px; }  
  100% { top: 100px; left: 100%; }  
}
```

Animation ပိုင်းကိုသွားပါမယ်။ အသုံးများတဲ့method တွေကတော့

■ animation-name

@keyframe နဲ့ချိန်မှာဖြစ်တဲ့အတွက် @keyframe ရဲ့ name ကိုရေးပေးရပါမှာ။

■ animation-duration

Animation တခုရဲ့ကြာချိန် second နဲ့ရေးတာများပါတယ်

■ animation-delay

Animation မစခင်စောင့်နေတဲ့အချိန်ဖြစ်ပါတယ်။ second နဲ့ရေးတာများပါတယ်။

■ animation-iteration-count

အကြိမ်ရေဘယ်လောက်animation လုပ်မှာလဲဆိုတာသတ်မှတ်တာပါ။

1 ဆိုတကြိမ် 2 ဆို နှစ်ကြိမ် အမြဲတမ်းလုပ်နေမယ်ဆိုရင်တော့ infinite နဲ့သတ်မှတ်ပါတယ်။

■ animation-direction

Direction ကတော့ ၄ခုရှိပါတယ်။ normal,reverse,alternate,alternate-reverse ဖြစ်ပါတယ်။

■ animation-timing-function

transition-timing-function နဲ့တူတူပါပဲ။

7) Column

Column ကတော့ စာသားတွေကို တဆက်တည်းရေးလိုက်တယ်ပြီးတဲ့နောက် group လေးတွေထပ်ခွဲချင်တယ်ဆိုတဲ့အခါသုံးပါတယ်။သူ့မှာရှိတဲ့Methodsတွေကတော့

■ column-count

Group ဘယ်နှစ်ခုခွဲချင်တာလဲဆိုတာကိုသတ်မှတ်ပေးရပါတယ်။

`column-count : 4;`

■ column-gap

Group တခုနဲ့တခုကြားအကွားအဝေးကို သတ်မှတ်ချင်တဲ့အခါ သုံးပါတယ်။

`column-gap : 20px;`

■ column-rule-style

Group တွေကိုပိုင်းတဲ့အခါ ကြားက မျဉ်းကြောင်းရဲ့style ကိုသတ်မှတ်ချင်တဲ့အခါ သုံးပါတယ်။သူ့မှာရှိတဲ့ styleတွေကတော့ Border style ရဲ့value တွေနဲ့တူတူပါပဲ။

`column-style : solid;`

■ column-rule-width

မျဉ်းကြောင်းရဲ့အထူဖြစ်ပါတယ်။

`column-rule-width : 20px;`

■ column-rule-color

မျဉ်းကြောင်းရဲ့အရောင်ဖြစ်ပါတယ်။

`column-rule-color : blue;`

■ column-rule

Border လိုမျိုးတကြောင်းတည်းနဲ့ရေးချင်တဲ့အခါ

`column-rule : 5px solid blue;`

■ column-width

Group အနေနဲ့ပိုင်းလိုက်တဲ့ column ရဲ့widthဖြစ်ပါတယ်။

`column-width : 12px;`

8) Pseudo Class

Pseudo Classဆိုတာကတော့ ရွေးချယ်လိုက်တဲ့ Element ရဲ့special state ကိုသတ်မှတ်တာဖြစ်ပါတယ်။ဥပမာ user က link တခုကို clickနှိပ်တဲ့အခါ Linkရဲ့colorကvisitedဖြစ်သွားတဲ့အခါ နောက်တရောင်ဖြစ်သွားစေချင်တဲ့အခါ pseudo Class ကိုသုံးပါတယ်။

အသုံးများတဲ့ Pseudo Class တွေကတော့

:active user ကနေ mouse နဲ့ဖြစ်စေ element ကိုactiveဖြစ်အောင် လုပ်လိုက်တဲ့အခါအလုပ်လုပ်ပေးမယ် class ဖြစ်ပါတယ်။ a tag မှာအသုံးများပါတယ်။

:checked

Input element မှာ checked ဖြစ်နေတဲ့ elementအတွက်အလုပ်လုပ်ပေးမယ် class ဖြစ်ပါတယ်။

:disabled

Selector element က disable ဖြစ်တဲ့အခါ အလုပ်လုပ်ပေးမယ် class ဖြစ်ပါတယ်။

:empty

Selector element က empty ဖြစ်တဲ့အခါ အလုပ်လုပ်ပေးမယ် class ဖြစ်ပါတယ်။

:enabled

Selector element က enabled ဖြစ်တဲ့အခါ အလုပ်လုပ်ပေးမယ် class ဖြစ်ပါတယ်။

:first-child

Selector element မှာ child element တွေထက်ရှိသေးတဲ့အခါ အကုန်လုံးကို Css effect တွေ မသက်ရောက်စေချင်ပဲ ပထမဆုံးတခုကိုပဲ effect ဖြစ်စေချင်တဲ့အခါ သုံးပါတယ်။

:focus

Selector element ကို user က cursor select လုပ်လိုက်တဲ့အခါ effect ဖြစ်စေချင်တဲ့အခါသုံးပါတယ်။

:hover

Selector element ကို user က hover လုပ်လိုက်တဲ့အခါ effect ဖြစ်စေချင်တဲ့အခါသုံးပါတယ်။

:last-child

Selector element မှာ child element တွေထက်ရှိသေးတဲ့အခါ အကုန်လုံးကို Css effect တွေ မသက်ရောက်စေချင်ပဲ နောက်ဆုံးတခုကိုပဲ effect ဖြစ်စေချင်တဲ့အခါ သုံးပါတယ်။

:link

a tag မှာအသုံးများပါတယ်။ User က link ကိုမနှိပ်ရသေးတဲ့ အခါ effect ဖြစ်မယ့် class ဖြစ်ပါတယ်။

:not(selector)

Element တွေအများကြီးရှိတဲ့အခါ တခုတည်းသော element ကလွဲလို့ကျန်တဲ့ element တွေကို effect ဖြစ်စေချင်တဲ့အခါသုံးပါတယ်။

:read-only

Selector element က read-only ဖြစ်တဲ့အခါ အလုပ်လုပ်ပေးမယ် class ဖြစ်ပါတယ်။

:required

Selector element က required attributes ဖြစ်တဲ့အခါ အလုပ်လုပ်ပေးမယ် class ဖြစ်ပါတယ်။

:visited

a tag မှာအသုံးများပါတယ်။ Userက link ကိုနှိပ်ပြီးတဲ့အခါ Css effect ဖြစ်စေတဲ့အခါ သုံးပါတယ်။

:placeholder-shown

Input tagမှာ placeholder ပြနေတဲ့အချိန်မှာအလုပ်လုပ်လုပ်ပေးမယ့် class ဖြစ်ပါတယ်။

:root

Css မှာ global variableတွေကြေညာပြီးလိုသလိုခေါ်သုံးချင်တဲ့အခါ အသုံးပြုပါတယ်။ နောက်သင်ခန်းစာမှာအသေးစိတ်လေ့လာပါ။

9) Root & Var()

■ Root

```
:root {  
  css declarations;  
}
```

Root ကတော့ Css declaration တွေကို global ကြေညာပြီးလိုအပ်သလိုယူသုံးတာဖြစ်ပါတယ်။ Declaration တွေကြေညာတဲ့အခါနာမည်ပေးရပါတယ်။နာမည်ရဲ့အရှေ့မှာတော့ - - ထည့်ပြီးကြေညာရပါတယ်။အသုံးပြုမယ်ဆို Var နဲ့ခေါ် ပြီးသုံးရပါတယ်။

```
:root {  
  --background-color: coral;  
  --text-color: blue;  
  --font-size: 15px;  
}
```

```
#div1 {  
  background-color: var(--background-color);  
  color: var(--text-color);  
  font-size: var(--font-size);  
}
```

■ Var()

```
var(--name, value)
```

အပေါ်ကလေ့လာခဲ့တဲ့အတိုင်း pseudo class ::root မှာ Css declaration တွေကိုကြေညာထားပြီး၊လိုအပ်သလိုခေါ်သုံးနိုင်အောင် Var ကိုအသုံးပြုပါတယ်။

Var မှာတော့ root မှာကြေညာထားတဲ့ name ကိုထည့်ပေးရပါတယ်။

အကယ်၍:: root မှာ ခေါ်ချင်တဲ့name မရှိရင် name ရဲ့နောက်မှာ ကိုယ်ထည့်ချင်တဲ့valueကို ထည့်ပေးရပါတယ်။

*(ဆိုလိုတာကတော့ ::rootမှာအဲ့nameရှိရင် အဲ့ဒီvalueကိုယူမယ်မရှိရင်၊ သတ်မှတ်လိုက်တဲ့valueကိုသုံးဆိုတဲ့သဘောပါပဲ။)

10) Media Queries

```
@media not|only mediatype and (expressions) {  
  CSS-Code;  
}
```

Desktop တွေ mobile တွေ size မတူတဲ့ screen တွေနဲ့ webpage တွေကို ကြည့်တဲ့အခါ Design တွေဟာသတ်မှတ်ထားတဲ့ပုံစံတွေနဲ့လွဲချော်သွားတာ ၊ သို့မဟုတ် mobile view နဲ့ကြည့်တဲ့အခါ Desktop view ရဲ့ design အတိုင်းမဟုတ်ပဲ customization လုပ်ချင်တဲ့အခါ @media queries ကိုသုံးပါတယ်။

```
Example:  
body {  
  background-color: red;  
}  
  
@media screen and (min-width: 720px) {  
  body {  
    background-color: green;  
  }  
}
```

အပေါ်က code ကိုကြည့်မယ်ဆိုရင် screen size က 720pxနဲ့အထက် ဆို green လို့ပေါ် ပြီး 720px အောက်ဆို redလို့ပေါ်ပါမယ်။ အဲလိုပဲ @media မှာဘယ် screen sizeဆို ဘယ်design နဲ့ရေးမယ်ဆိုတာ ကိုယ်ပိုင်သတ်မှတ်လို့ရပါတယ်။
@media queries ကို screen size အလိုက် Multipleရေးနိုင်ပါတယ်။

11) Flex Box

Display ရဲ့ value တခုဖြစ်ပါတယ်။ Layout တွေကို flexible ဖြစ်အောင် လိုသလိုချိန်ညှိပြီး design လုပ်တာဖြစ်ပါတယ်။

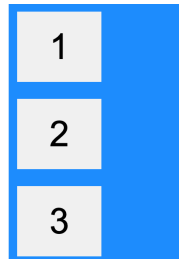
Flex Container

■ flex-direction

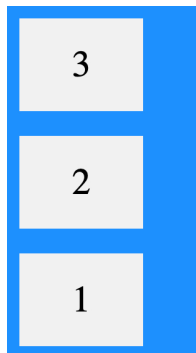
Div block တွေကို ဘယ်လိုပုံစံနဲ့ပြမှာလဲဆိုတာကို သတ်မှတ်ပေးပါတယ်။

Valueတွေကတော့

```
.flex-container {  
  display: flex;  
  flex-direction: column;  
}
```



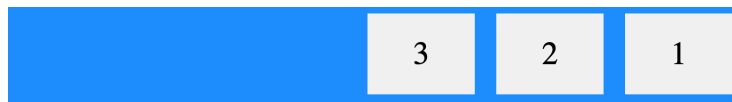
```
.flex-container {  
  display: flex;  
  flex-direction: column-reverse;  
}
```



```
.flex-container {
  display: flex;
  flex-direction: row;
}
```



```
.flex-container {
  display: flex;
  flex-direction: row-reverse;
}
```



■ flex-wrap

Web page ရဲ့ site ကျဉ်းသွားခဲ့ရင် အလိုအလျောက် block တွေကို ဖြတ်ဖြတ် ပေးပြီး ပြချင်တဲ့အခါ သုံးပါတယ်။

Value တွေကတော့ wrap, no-wrap, wrap-reverse

```
.flex-container {
  display: flex;
  flex-wrap: wrap;
}
```

■ justify-content

Block တွေကို horizontal show position ထားချင်တဲ့အခါ သုံးပါတယ်။

Value ကိုတော့ center, flex-start, flex-end ဖြစ်ပါတယ်။ center ကတော့ အသုံးများပါတယ်။

```
.flex-container {
  display: flex;
  justify-content: center;
}
```

■ align-items

Block တွေကို vertical show position ထားချင်တဲ့အခါသုံးပါတယ်။

Value ကိုတော့ center, flex-start, flex-end, stretch, baseline ဖြစ်ပါတယ်။ center ကတော့ အသုံးများပါတယ်။ justify-content: center နဲ့တွဲသုံးမယ်ဆိုရင်တော့ Web page ရဲ့ အလယ်တည့်တည့်ကျနေမှာဖြစ်ပါတယ်။

```
.flex-container {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}
```

For Detail Cheat sheet

<https://yoksel.github.io/flex-cheatsheet/>

12) Viewport

Desktop view နဲ့ရေးပြီး screen size မတူတဲ့ mobile view တို့နဲ့ကြည့်တဲ့ အခါ အလွန် သေးတဲ့ ပုံစံနဲ့ webpage တွေကိုမြင်ရပါတယ်။ အဲ့လိုမဖြစ်အောင် viewport ကိုသုံးပါတယ်။ ရေးတဲ့ပုံစံကတော့

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```