

អារម្ភកថា

សព្វថ្ងៃនេះការប្រើប្រាស់គេហទំព័រឬក៏ Website គឺជាមធ្យោបាយមួយយ៉ាងសំខាន់ក្នុងការដោះស្រាយបញ្ហា និងជួយសំរួលទៅដល់ការងារប្រចាំថ្ងៃរបស់មនុស្សដែលរស់នៅជុំវិញពិភពលោក។ ភាគច្រើនការងារប្រចាំថ្ងៃទាក់ទងនឹងការលក់ទំនិញនៅលើអ៊ីនធឺណែត ការផ្សព្វផ្សាយព័ត៌មានឬព្រឹត្តិការណ៍ផ្សេងៗត្រូវបានពឹងផ្អែកទៅលើការប្រើប្រាស់គេហទំព័រស្ទើរតែទាំងស្រុង ដែលជាហេតុផលដែលត្រូវទាមទារឱ្យអ្នកប្រើប្រាស់ត្រូវមានសមត្ថភាព និងចំណេះដឹងគ្រប់គ្រាន់ក្នុងការប្រើប្រាស់គេហទំព័រ។

ដើម្បីលើស្ទូចវិស័យបង្កើតគេហទំព័រនៅក្នុងព្រះរាជាណាចក្រកម្ពុជា **មជ្ឈមណ្ឌលបច្ចេកវិទ្យាព័ត៌មានអ៊ីតទ្រីម (Yttrium Information Technology Center)** បានបង្កើតសៀវភៅមូលដ្ឋានគ្រឹះក្នុងការបង្កើតគេហទំព័រដោយប្រើប្រាស់ភាសា HTML CSS jQuery និង JavaScript ឡើងក្នុងគោលបំណងបណ្តុះបណ្តាលចំណេះដឹងទៅដល់ សិស្ស និស្សិត និងអ្នកសិក្សាទាំងអស់ដែលមានបំណងចង់សិក្សាផ្នែកសរសេរគេហទំព័រ។ ដើម្បីឱ្យអ្នកសិក្សាមានចំណេះដឹងពិតប្រាកដជាពិសេសមានសមត្ថភាពគ្រប់គ្រាន់ក្នុងការបង្កើតគេហទំព័រដោយខ្លួនឯងឬក៏តាមអង្គភាព និងក្រុមហ៊ុននានាយកទៅធ្វើការងារផ្សេងៗតាមតម្រូវការនៅក្នុងព្រះរាជាណាចក្រកម្ពុជា ។

យើងខ្ញុំជាអ្នករៀបរៀងសៀវភៅមូលដ្ឋានគ្រឹះក្នុងការបង្កើតគេហទំព័រសង្ឃឹមថាអ្នកសិក្សាទាំងអស់នឹងខិតខំប្រឹងប្រែងរៀនសូត្រស្រាវជ្រាវដែលអាចយកទៅបំពេញតម្រូវការងារផ្សេងៗដែរខ្លួនចង់ធ្វើនៅពេលអនាគត។

ថ្ងៃទី ២៧ ខែ កញ្ញា ឆ្នាំ ២០២២

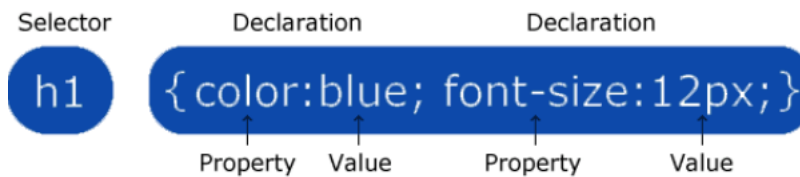
CSS

មេរៀនទី១	ការណែនាំពីភាសា CSS
មេរៀនទី២	CSS Width & Height.....
មេរៀនទី៣	CSS Background
មេរៀនទី៤	CSS Border
មេរៀនទី៥	CSS Padding
មេរៀនទី៦	CSS Box-Sizing.....
មេរៀនទី៧	CSS Margin.....
មេរៀនទី៨	CSS Outline.....
មេរៀនទី៩	CSS Text and Fonts.....
មេរៀនទី១០	CSS List and Table.....
មេរៀនទី១១	CSS Display.....
មេរៀនទី១២	CSS Position.....
មេរៀនទី១៣	CSS Float.....
មេរៀនទី១៤	CSS Overflow.....
មេរៀនទី១៥	CSS Pseudo.....
មេរៀនទី១៦	CSS Opacity.....
មេរៀនទី១៧	CSS Shadow.....
មេរៀនទី១៨	CSS Transform 2D.....
មេរៀនទី១៩	CSS Transform 3D.....
មេរៀនទី២០	CSS Transition.....
មេរៀនទី២១	CSS Animation
មេរៀនទី២២	CSS Flexbox.....
មេរៀនទី២៣	CSS Grid.....
មេរៀនទី២៤	CSS Media Query
មេរៀនទី២៥	CSS Variable.....

I. និយមន័យ

CSS (Cascading Style Sheet) គឺជាភាសារម្មួយប្រភេទដែលគេប្រើសម្រាប់ធ្វើការកែលំអរឬក៏តុបតែងទៅលើ HTML Element ។

A. Syntax



B. Selector

1. The Element Selector
2. Class Selector ប្រើដើម្បី apply Style ទៅលើ HTML element ណាដែលបានហៅ Class នោះទៅប្រើប្រាស់។ ជាទូទៅ class selector ត្រូវបានគេបង្កើតឡើងដើម្បីហៅទៅប្រើប្រាស់ច្រើនដង។ របៀបធ្វើការបង្កើត Class និងការហៅ class យកមកប្រើ។

Syntax

```
.class_name {
  property_name: value;
  property_name: value;
}
```

Example

```
<h1 class="class_name">Welcome to MasterIT Center</h1>
<p class="class_name">កិត្យានុវត្តន៍នេះផ្តល់ឱកាសដល់អ្នកនាំចេញ</p>
```

3. ID Selector ប្រើដើម្បី apply style ទៅលើ HTML Element ណាដែលបានហៅ id នោះទៅប្រើប្រាស់។ ជាទូទៅ id selector ត្រូវបានគេបង្កើតឡើងដើម្បីហៅទៅប្រើប្រាស់តែម្តងប៉ុណ្ណោះ។ របៀបធ្វើការបង្កើត id និងការហៅ id យកមកប្រើ

Syntax

```
#id_name {
  property_name: value;
  property_name: value;
}
```

Example

```
<h1 id="id_name">Welcome to MasterIT Center</h1>
<p id="id_name">កិត្យានុវត្តន៍នេះផ្តល់ឱកាសដល់អ្នកនាំចេញ</p>
```

C. Comments

CSS Comments គេប្រើប្រាស់សម្រាប់បិទ block code ណាមួយកុំឱ្យដំណើរការ។ CSS Comments មាន ២ ប្រភេទ

- Single-line comment
- Multi-line comment

Example

```
/* This is a single-line comment */
/* This is
a multi-line
comment */
```

D. ការបញ្ចូល style CSS មាន ៣ របៀប

- Inline Style: Style ជា attribute របស់ HTML Tag

Example

```
<h1 style="property: value;">MasterIT Center</h1>
```

- Internal Style Sheet Style CSS ត្រូវបានសរសេរដោយប្រើ tag <style> នៅក្នុង tag <head>

Example

```
<head>
  <style>
    property: value;
    property: value;
  </style>
</head>
```

- External Style Sheet: Style ត្រូវបានផ្គុំនៅក្នុង file ដែលមាន extension .css ហើយត្រូវបានហៅតាមរយៈ tag <link>

Example

```
<link rel="stylesheet" href="file-name.css">
```

លំហាត់អនុវត្ត

1. តើអ្វីទៅជា CSS?
2. តើ Class Selector and ID Selector គេប្រើសម្រាប់ធ្វើអ្វី?
3. តើ Comment គេប្រើប្រាស់សម្រាប់ធ្វើអ្វី? Comment មានអ្វីខ្លះ ចូលរៀនរាប់។
4. តើការ link Style CSS មានប៉ុន្មានប្រភេទ? អ្វីខ្លះ? ចូលរៀនរាប់និងពន្យល់ឲ្យបានក្លៀវក្លាយ។

មេរៀនទី២

CSS Width/Height

Width and height properties ត្រូវបានគេប្រើសម្រាប់កំណត់បណ្តោយ និង ទទឹងទៅឲ្យ THML element។ width and height properties ត្រូវបានគេប្រើដោយកំណត់ជាខ្នាតដូចខាងក្រោម៖

1. ខ្នាតមិនប្រែប្រួល: ខ្នាតប្រវែងដែលថេរ នៅគ្រប់មជ្ឈដ្ឋាន

- Cm: (centimeter) សង់ទីម៉ែត្រ
- Mm: (millimeter) មីលីម៉ែត្រ
- In: អ៊ីន(inch)
- Px: (pixel) ភីចសេល ខ្នាតដែលមានទំហំជាការតូចៗដែលបង្កើតរាល់element នៅលើ webpage
- Pt: (point) ភ័ញ
- Pc: (pica) ផែកា

2. ខ្នាតប្រវែងប្រែប្រួល: ខ្នាតប្រវែងដែលប្រែប្រួលទៅតាមមជ្ឈដ្ឋានដែលវាស្ថិតនៅក្នុង

- Em: គឺជាខ្នាតមួយសម្រាប់កំណត់ទំហំតាមdeviceផ្សេងៗគ្នា $1em = px * 2$
Or $1em = 16px$
 - vh: %នៃកម្ពស់នៃ viewport។ ex; $50vh = 50\%$ នៃកំពស់ viewport
 - vw: %នៃប្រវែងផ្នែកនៃ viewport។ ex; $50vw = 50\%$ នៃប្រវែងផ្នែក viewport
- percent: គឺជាខ្នាតមួយដែលទំហំរបស់វាគិតជារយ(%)នៃទំហំElement

3. Other Property

- Max-width: កំណត់ប្រវែងផ្នែកអតិបរមានៃធាតុមួយ
- Max-height: កំណត់កម្ពស់អតិបរមានៃធាតុមួយ
- Min-width: កំណត់ប្រវែងផ្នែកអប្បបរមានៃធាតុមួយ
- Min-height: កំណត់កម្ពស់អប្បបរមានៃធាតុមួយ

Example

```
div {  
width: 200px;  
height: 20%;  
background-color: blue;  
}
```

លំហាត់អនុវត្តន៍

1. តើ Properties width/height ត្រូវបានគេប្រើប្រាស់សម្រាប់ធ្វើអ្វី?
2. ចូររៀបរាប់ខ្នាតដែលប្រើប្រាស់ជាមួយ Properties width/heightដែលអ្នកស្គាល់ឲ្យបាន ៥ខ្នាត។

មេរៀនទី៣

CSS Background

CSS Background ប្រើសម្រាប់ឲ្យអ្នកសរសេរគេហទំព័រដាក់រូបភាពឬពណ៌ទៅឲ្យផ្ទៃខាងក្រោយនៃគេហទំព័រ

CSS Background មានដូចជា៖

- background-color: ប្រើសម្រាប់ដាក់ពណ៌ទៅឲ្យផ្ទៃខាងក្រោយនៃ website វាអាចជាឈ្មោះពណ៌ ឬលេខកូដពណ៌
- background-size: ប្រើសម្រាប់កំណត់ទំហំរូបភាពនៅផ្ទៃខាងក្រោយvalue មាន contain/cover
- background-image: ប្រើសម្រាប់ដាក់រូបភាពទៅឲ្យផ្ទៃខាងក្រោយនៃ website
Ex: background-image: url(" ");
- background-blend-mode: ប្រើសម្រាប់លាយពណ៌ផ្ទៃខាងក្រោយ។value: multiply / screen / overlay / darken / lighten /color-dodge / saturation / color / luminosity

- ## Example
- ```
.box {
background-color: red;
background-image: url(1.jpg);
background-position: center;
}
```

1. តើ CSS background គេប្រើសម្រាប់ធ្វើអ្វី?
2. តើ CSS background មានប៉ុន្មានចូលរៀបរាប់?
3. តើ property អ្វីដែលគេប្រើប្រាស់សម្រាប់ដាក់រូបភាពនៅផ្ទៃខាងក្រោយនៃ website?
4. ចូររក CSS background យ៉ាងតិចឲ្យបាន២ថែមទៀត?

## CSS Border

1. Border Style: ប្រើសម្រាប់ដាក់បន្ទាត់ទៅឱ្យ HTML element ដែលយើងចង់ដាក់ Border។ Property ដែលត្រូវប្រើគឺ border-style។ border-styleមានដូចជា
  - dotted
  - dashed
  - solid
  - double
  - none
  - groove

Border-style: dotted solid          dashed          double;  
 លើ          ស្នាំ          ក្រោម          នៃង

border-style: dotted          solid          dashed;

លើ                                  ឆ្វេងស្តាំ                  ក្រោម

border-style: dotted          solid;

លើក្រោម                  ឆ្វេងស្តាំ

border-style: dotted; ទាំងបួន

2. Border Width: ប្រើប្រាស់សម្រាប់ដាក់កម្រាស់ឲ្យប្រភេទបន្ទាត់ដែលយើងចង់ដាក់ border។ ខ្នាតមាន in, px, cm, em, etc ។ property ដែលត្រូវប្រើប្រាស់គឺ border-width
3. Border Color: ប្រើសម្រាប់ដាក់ពណ៌ទៅឲ្យបន្ទាត់ដែលយើងចង់ដាក់ border property ដែលត្រូវប្រើគឺ border-color វាអាចដាក់ដូចជា border-style បានផងដែរ
4. Rounded Borders: ត្រូវបានគេប្រើសម្រាប់ធ្វើការដាក់ border ឲ្យកោងជ្រុង។ property ដែលសត្រូវប្រើគឺ border-radius

## Example

```
p {
border-style: solid;
border-width: 10px;
border-color: black;
border-radius: 50px;
}
```

លំហាត់អនុវត្តន៍

1. តើ CSS Border ប្រើសម្រាប់ធ្វើអ្វី?
2. តើ property border-width គេប្រើសម្រាប់ធ្វើអ្វី?
3. តើ property របស់ border មានអ្វីខ្លះ? ចូលរៀបរាប់។
4. ចូលអនុវត្តន៍លំហាត់ខាងក្រោម។

## Exercise



មេរៀនទី៥

## CSS Padding

CSS Padding ប្រើដើម្បីកំណត់គំលាត HTML Element ពីខាងក្រៅបំផុតមកកាន់អ្វីដែលនៅខាងក្នុងវា។ padding មានដូចជា

- padding-top: ប្រើសម្រាប់កំណត់គំលាតពីខាងលើ HTML Element
- padding-right: ប្រើសម្រាប់កំណត់គំលាតពីខាងស្តាំ HTML Element
- padding-bottom: ប្រើសម្រាប់កំណត់គំលាតពីខាងក្រោម HTML Element
- padding-left: ប្រើសម្រាប់កំណត់គំលាតពីខាងឆ្វេង HTML Element

**ចំណាំ:** កាលណាយើងចង់ឲ្យ element ណាមួយចូលទៅកាន់កណ្តាល element មួយទៀតយើងដាក់

margin: 0 auto

- យើងអាចប្រើទម្រង់កាត់របស់វាបានផងដែរ៖  
Ex:
- padding: 10px; មានន័យថាដាក់ទៅក្រោមជ្រុង
- padding: 10px 20px; មានន័យថាដាក់ លើក្រោម 10px ស្តាំឆ្វេង 20px
- padding: 10px 20px 30px; មានន័យថា លើ 10px ស្តាំឆ្វេង 20px ក្រោម 30px
- padding: 10px 20px 30px 40px; មានន័យថា លើ ស្តាំ ក្រោម ឆ្វេង

## Example

```
.box {
 background-color: red;
 width: 200px;
 height: 200px;
 padding-top: 50px;
 padding-bottom: 50px;
}
```

### លំហាត់អនុវត្តន៍

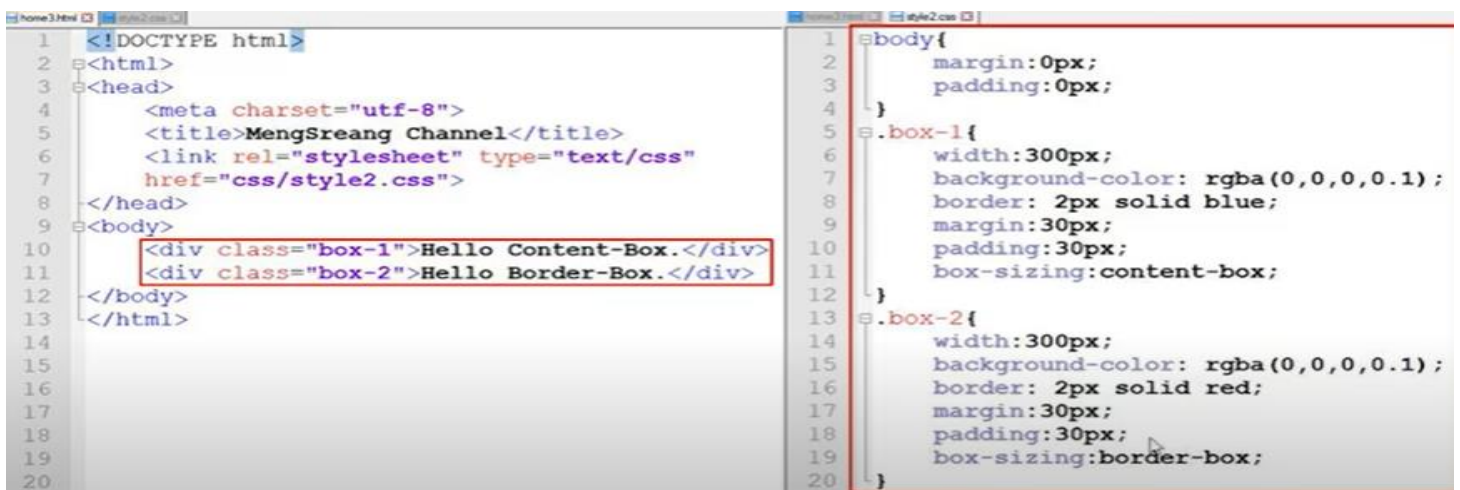
1. តើ CSS padding គេប្រើសម្រាប់ធ្វើអ្វី?
2. តើ Property របស់ padding មានប៉ុន្មានអ្វីខ្លះ? ចូររៀបរាប់។
3. ចូររៀបរាប់អំពីការប្រើប្រាស់ទម្រង់កាត់របស់ CSS padding។

### មេរៀនទី៦

### CSS Box-Sizing

យើងប្រើ box-sizing property ដើម្បីកំណត់ឱ្យ width អាចរាប់បញ្ចូលនូវទំហំរបស់padding និង border បាន។ វាមានពីរគឺ៖

- content-box: ជាប្រភេទដែលមិនរាប់បញ្ចូលនូវទំហំរបស់padding និង border (តម្លៃដើម)
- border-box: ជាប្រភេទដែលរាប់បញ្ចូលនូវទំហំរបស់ padding និង border មិនរាប់ margin



### មេរៀនទី៧

### CSS Margin

CSS Margin ប្រើសម្រាប់ដាក់គម្លាតពី Element មួយទៅកាន់ Element មួយទៀត។ CSS margin មានដូចជា៖



- margin-top: ប្រើសម្រាប់កំណត់គលាតខាងក្នុងនៅលើ HTML Element
- margin-right: ប្រើសម្រាប់កំណត់គម្រាតខាងក្នុងនៅស្តាំ HTML Element
- margin-bottom: ប្រើសម្រាប់កំណត់គលាតខាងក្នុងនៅក្រោម HTML Element
- margin-left: ប្រើសម្រាប់កំណត់គលាតខាងក្នុងនៅឆ្វេង HTML Element

**ចំណាំ:** កាលណាយើងចង់ឱ្យ element ណាមួយចូលទៅកាន់កណ្តាល element មួយទៀតយើងដាក់

margin: 0 auto

យើងអាចប្រើទម្រង់កាត់របស់វាបានផងដែរ៖

Ex:

- margin: 10px; មានន័យថាដាក់ទៅក្រោមជ្រុង
- margin: 10px 20px; មានន័យថាដាក់ លើក្រោម 10px ស្តាំឆ្វេង 20px
- margin: 10px 20px 30px; មានន័យថា លើ 10px ស្តាំឆ្វេង 20px ក្រោម 30px
- margin: 10px 20px 30px 40px; មានន័យថា លើ ស្តាំ ក្រោម ឆ្វេង

## Example

```
.box {
background-color: red;
width: 200px;
height: 200px;
margin-left: 20px;
margin-top: 20px;
}
```

### លំហាត់អនុវត្តន៍

1. តើ CSS margin គេប្រើប្រាស់សម្រាប់ធ្វើអ្វី?
2. តើ property របស់ marginមានអ្វីខ្លះ? ចូលរៀបរាប់។
3. បើចង់ឱ្យ HTML Element មួយចូលទៅកណ្តាលតើយើងត្រូវធ្វើយ៉ាងម៉េច?

មេរៀនទី៨

### CSS Outline

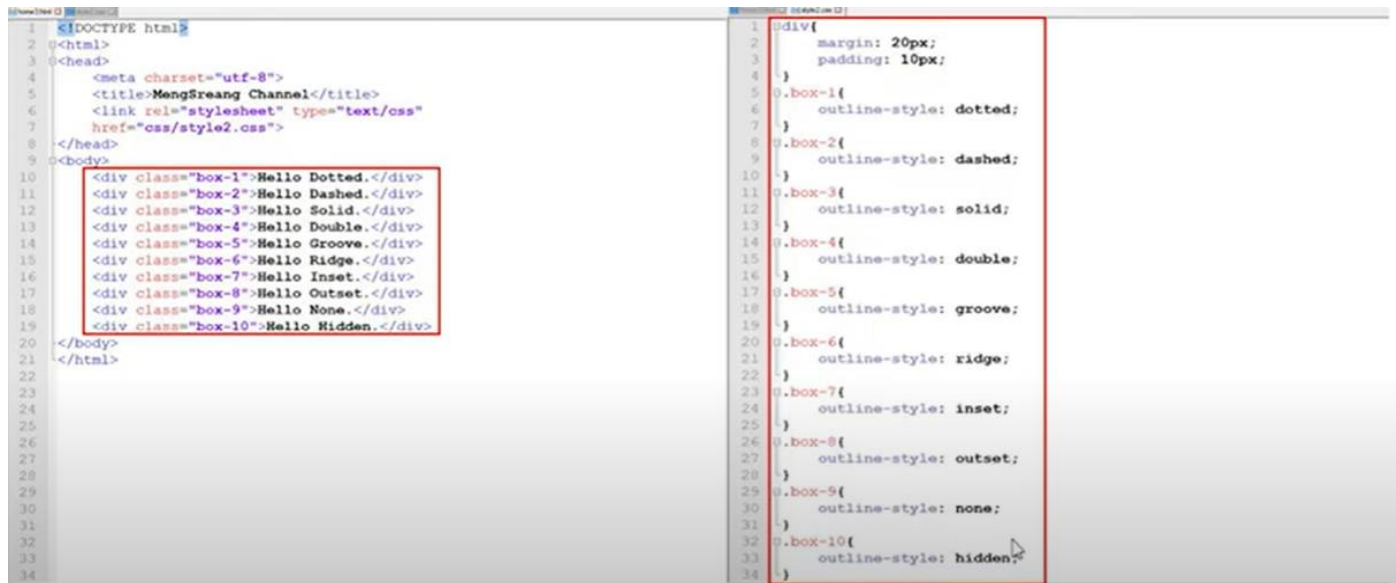
CSS Outline ជាបន្ទាត់ដែលត្រូវបានគូរជុំវិញធាតុទាំងអស់ ដែលធ្វើឱ្យធាតុនីមួយៗមើលទៅមានភាពលេចធ្លោ។

Outline មាន 4

- Outline-style: សម្រាប់កំណត់ style ឱ្យ outline
- Outline-color: សម្រាប់កំណត់ពណ៌ឱ្យ outline
- Outlined -width: សម្រាប់កំណត់ទំហំ outline
- Outline-offset: សម្រាប់បន្ថែមចន្លោះរវាង outline និង border



1. Outline-Style: នៅក្នុង outline-style មាន value ជាច្រើនដូចជា: dotted, solid, double, dashed, groove, ridge, inset, outset, none, & hidden។ យើងអាចកំណត់valueនៅក្នុង outline-styleបានតែមួយប៉ុណ្ណោះ។

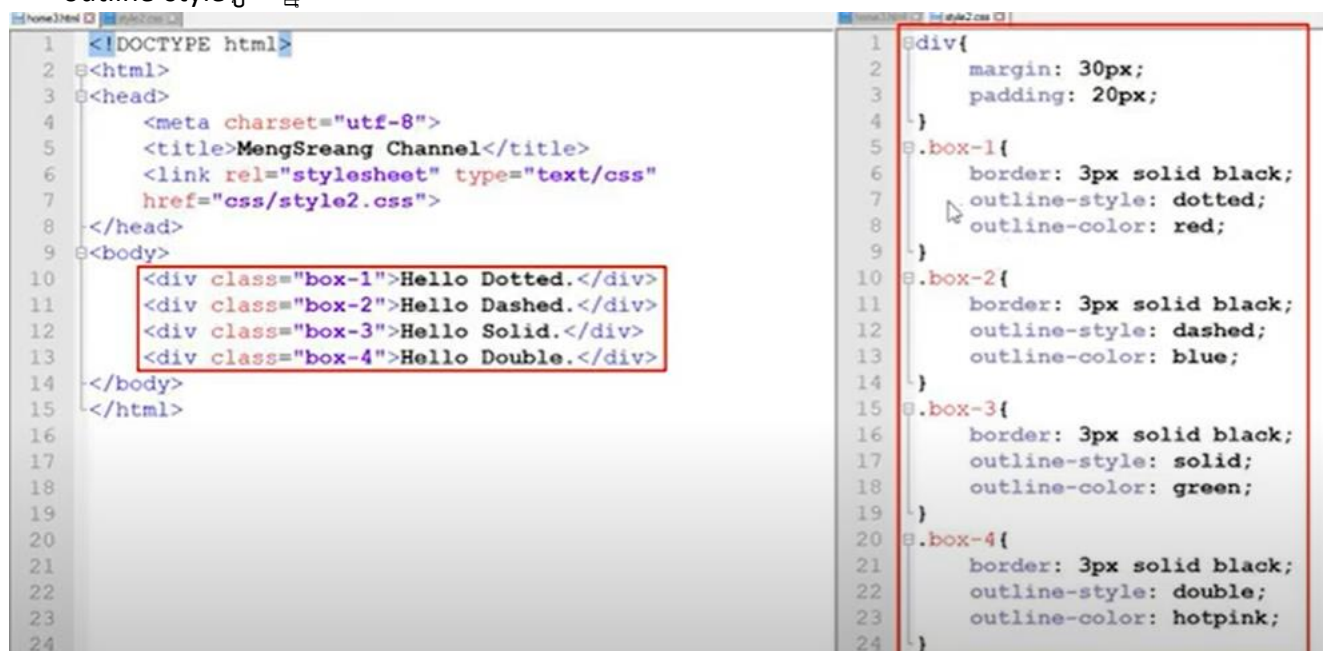


The screenshot shows a code editor with two panels. The left panel contains HTML code with ten div elements, each with a class from 'box-1' to 'box-10' and a text label. The right panel contains CSS code defining the 'outline-style' for each box class. The CSS rules are: box-1 (dotted), box-2 (dashed), box-3 (solid), box-4 (double), box-5 (groove), box-6 (ridge), box-7 (inset), box-8 (outset), box-9 (none), and box-10 (hidden). The HTML and CSS code are both highlighted with red boxes.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>MengSreang Channel</title>
6 <link rel="stylesheet" type="text/css"
7 href="css/style2.css">
8 </head>
9 <body>
10 <div class="box-1">Hello Dotted.</div>
11 <div class="box-2">Hello Dashed.</div>
12 <div class="box-3">Hello Solid.</div>
13 <div class="box-4">Hello Double.</div>
14 <div class="box-5">Hello Groove.</div>
15 <div class="box-6">Hello Ridge.</div>
16 <div class="box-7">Hello Inset.</div>
17 <div class="box-8">Hello Outset.</div>
18 <div class="box-9">Hello None.</div>
19 <div class="box-10">Hello Hidden.</div>
20 </body>
21 </html>
22
23
24
25
26
27
28
29
30
31
32
33
34
```

```
1 div{
2 margin: 20px;
3 padding: 10px;
4 }
5 .box-1{
6 outline-style: dotted;
7 }
8 .box-2{
9 outline-style: dashed;
10 }
11 .box-3{
12 outline-style: solid;
13 }
14 .box-4{
15 outline-style: double;
16 }
17 .box-5{
18 outline-style: groove;
19 }
20 .box-6{
21 outline-style: ridge;
22 }
23 .box-7{
24 outline-style: inset;
25 }
26 .box-8{
27 outline-style: outset;
28 }
29 .box-9{
30 outline-style: none;
31 }
32 .box-10{
33 outline-style: hidden;
34 }
```

2. Outline-Color: នៅក្នុង outline-color មាន value ជាច្រើនពណ៌និងផ្នែកផ្សេងៗទៀតនៅក្នុងមេរៀន Color។ ហើយការកំណត់valueគឺមានមួយដូចនឹង outline-styleដែរហើយយើងអាចកំណត់valueដូចទៅនឹងoutline-styleដូចគ្នាដែរ។



The screenshot shows a code editor with two panels. The left panel contains HTML code with four div elements, each with a class from 'box-1' to 'box-4' and a text label. The right panel contains CSS code defining the 'outline-color' for each box class. The CSS rules are: box-1 (red), box-2 (blue), box-3 (green), and box-4 (hotpink). The HTML and CSS code are both highlighted with red boxes.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>MengSreang Channel</title>
6 <link rel="stylesheet" type="text/css"
7 href="css/style2.css">
8 </head>
9 <body>
10 <div class="box-1">Hello Dotted.</div>
11 <div class="box-2">Hello Dashed.</div>
12 <div class="box-3">Hello Solid.</div>
13 <div class="box-4">Hello Double.</div>
14 </body>
15 </html>
16
17
18
19
20
21
22
23
24
```

```
1 div{
2 margin: 30px;
3 padding: 20px;
4 }
5 .box-1{
6 border: 3px solid black;
7 outline-style: dotted;
8 outline-color: red;
9 }
10 .box-2{
11 border: 3px solid black;
12 outline-style: dashed;
13 outline-color: blue;
14 }
15 .box-3{
16 border: 3px solid black;
17 outline-style: solid;
18 outline-color: green;
19 }
20 .box-4{
21 border: 3px solid black;
22 outline-style: double;
23 outline-color: hotpink;
24 }
```

3. Outline-Width: នៅក្នុង outline-widthមានvalueជាច្រើនដូចជា: thin, medium, thick និងខ្នាតប្រវែង។ ហើយការកំណត់ value គឺមានមួយដូចនឹង outline-styleដែរយើងអាចកំណត់valueដូចទៅនឹងoutline-styleដូចគ្នាដែរ។

យើងអាចកំណត់ property ៣ ឱ្យក្លាយជាមួយ property បាន

Ex: outline: 5px solid green;

4. Outline-Offset: នៅក្នុង outline-offsetមាន value ជាខ្នាតប្រវែងហើយការកំណត់valueគឺមានមួយដូចនឹងoutline-styleដែរហើយយើងអាចកំណត់valueដូចទៅនឹងoutline-styleដូចគ្នាដែរ។

### លំហាត់អនុវត្តន៍

1. តើគេប្រើOutlineសម្រាប់ធ្វើអ្វី?
2. តើoutlineមានអ្វីខ្លះ?ចូររៀបរាប់?
3. តើ outline-widthមានvalueអ្វីខ្លះ?

### មេរៀនទី៩

### CSS Text and Fonts

CSS Text ប្រើសម្រាប់ឱ្យអ្នកសរសេរគេហទំព័រធ្វើការតុបតែង និងធ្វើការផ្លាស់ប្តូរទៅអក្សរដែលបានបង្ហាញនៅលើគេហទំព័រ (website) ។ CSS Text មានដូចជា៖

- text-align: ប្រើសម្រាប់ធ្វើការតម្លៃអក្សរ។ value របស់វាមានដូចជា center, left, right, justify។
- text-indent: ប្រើប្រាស់សម្រាប់ចូលបន្ទាត់
- text-transform: ប្រើសម្រាប់បំប្លែងអក្សរ។ value របស់វាមានដូចជា lowercase, capitalize, uppercase។
- text-decoration: ប្រើសម្រាប់ដាក់បន្ទាត់ឬដោះបន្ទាត់ពីក្រោមអក្សរ។ value របស់វាមានដូចជា none, underline, line-through, overline, blink។
- line-height: គំលាតអក្សរពីមួយបន្ទាត់ទៅមួយបន្ទាត់px
- letter-spacing: គំលាតរវាងអក្សរនីមួយៗpx
- word-spacing: គំលាតរវាងពាក្យនីមួយៗpx
- text-shadow: ប្រើប្រាស់សម្រាប់ដាក់ស្រមោលទៅឲ្យអក្សរ។



- font-family: ប្រើប្រាស់សម្រាប់ប្តូរឈ្មោះឲ្យfont
- font-weight: ប្រើសម្រាប់ដាក់អក្សរដិត
- font-size: ប្រើសម្រាប់កំណត់ទំហំរបស់ font

## Example

```
p {
text-align: left;
text-indent: 10px;
text-transform: uppercase;
font-weight: bolder;
font-size: 20px;
font-family: Consolas;
}
```

### លំហាត់អនុវត្តន៍

1. តើ CSS Text គេប្រើសម្រាប់ធ្វើអ្វី?
2. តើ Property text-align គេប្រើសម្រាប់ធ្វើអ្វី?
3. តើ Property line-height គេប្រើសម្រាប់ធ្វើអ្វី?

### មេរៀនទី១០

#### CSS List & Table

List នៅក្នុង HTML យើងបានសិក្សាមួយចំនួនធំហើយ ហើយយើងនឹងសិក្សានៅក្នុង CSS បន្ថែមទៀត។ ក្នុងមេរៀនlistនេះគឺមាន propertier ដែលយើងត្រូវសិក្សាដែលមានដូចជាខាងក្រោម:

- List-style-type: កំណត់ប្រភេទនៃសញ្ញា
- List-style-image: កំណត់ប្រភេទនៃរូបភាព(Icon)ទៅឲ្យបញ្ជី។
- List-style-position: ប្រើសម្រាប់កំណត់ទីតាំងសម្រាប់ដាក់បញ្ជី។

1. List style type មាន <ul> & <ol>
  - <ul> មាន square, circle, disc
  - <ol> មាន Cambodia, decimal-leading-zero, lower-latin, upper-latin, lower-roman, upper-roman
2. List style Image ប្រើ url ដើម្បីកំណត់រូបភាពបាន។

EX: List-style-image: url("image");
3. List style position មានតម្លៃ២គឺ:

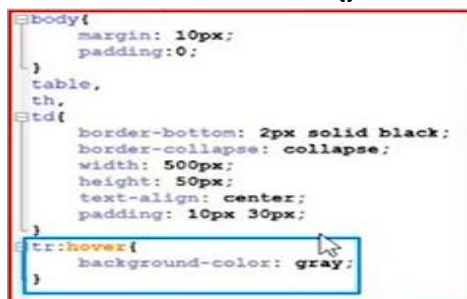
- Inside: សញ្ញាសម្គាល់នៅក្នុង។
  - Outside: សញ្ញាសម្គាល់នៅក្រៅ(វាដូចគម្លៃដើម)
- EX: list-style-position: inside;

ចំណាំ: ការសរសេរដាក់វាតើមួយដោយប្រើlist-style



Tableនៅក្នុងHTMLយើងបានសិក្សារួចហើយ ហើយយើងនឹងសិក្សាវានៅក្នុងcss បន្ថែមទៀត។

- Border: -----
- border-collapse: collapse;
- width: & height: -----
- text-align: -----
- vertical-align: top, metal, bottom;
- padding: -----
- hover table: -----tr:hover{}



- Striped Table: (even) ឬ (odd) ស៊េស



Ex: tr: nth-child(even){ background-color: gray;}

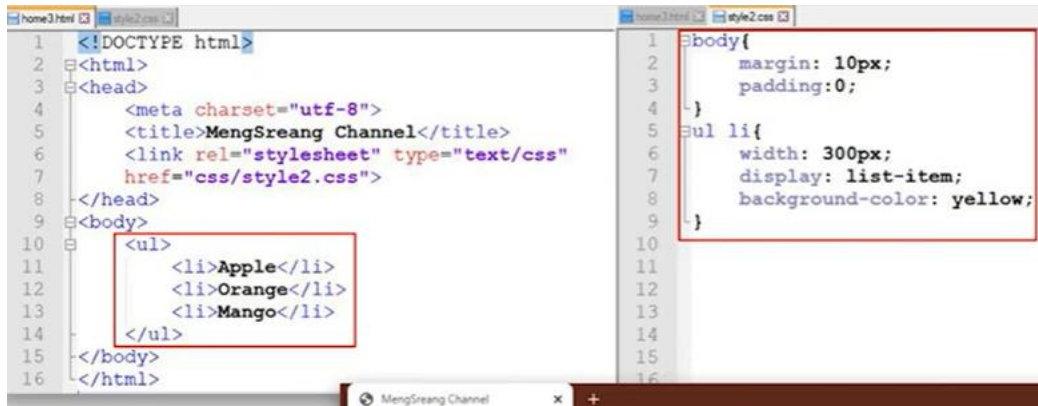


នៅក្នុងមេរៀនdisplayនេះមានpropertyសំខាន់ពីរសម្រាប់សិក្សា ទី១គឺdisplay property និងទី២ គឺvisibility property។

## 1.Display:

នៅក្នុងdisplayមានតម្លៃដូចជា:

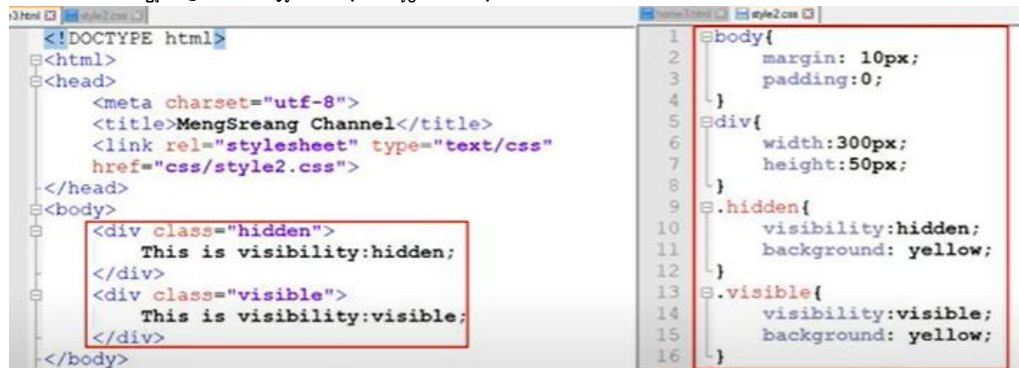
- None: មិនបង្ហាញអ្វីទាំងអស់។
- Block: បង្ហាញជាដុំ វាផ្តាច់ខ្លួនពីគេពីងង។ ប្រវែងផ្នែករបស់វាលាតសន្ធឹងពេញឆាកដែលផ្ទុកវា។
- Inline: បង្ហាញក្នុងបន្ទាត់។ មិនអាចកំណត់ប្រវែងទៅឲ្យវាបានទេ។
- Inline-block: បង្ហាញក្នុងបន្ទាត់ដូចinlineដែរ តែអាចកំណត់ប្រវែងទៅឲ្យវាបាន។
- List-item: បង្ហាញលក្ខណៈដូចនឹង<li>ដែរ



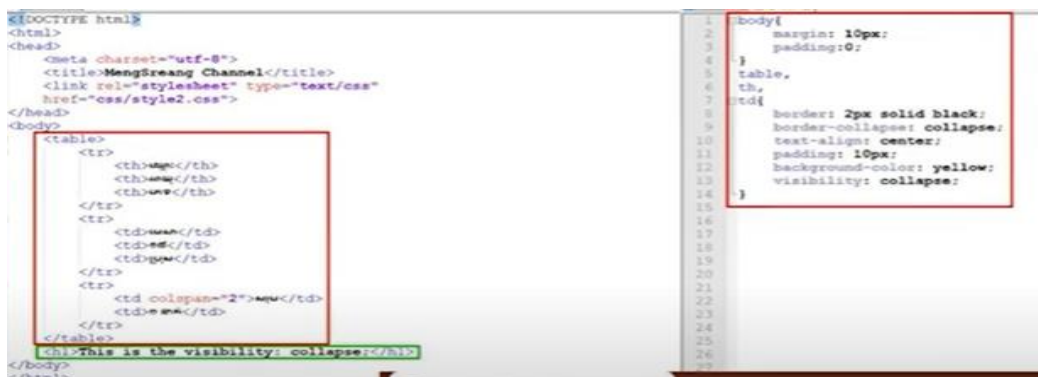
## 2.Visibility

វាប្រើដូចនឹងdisplayដែរ ហើយវាមានតម្លៃដូចជា:

- Hidden: មិនបង្ហាញអ្វីទាំងអស់ដូចនឹង តម្លៃ none in display ដែរ តែវាទុក space ដែលវាបានលុបឲ្យនៅទំនេរចោល។
- Visible: បង្ហាញជាធម្មតា។(តម្លៃដើម)



- Collapse: មិនបង្ហាញអ្វីទាំងអស់ដូចនឹង hidden ដែរ តែវាលុបបានតែតារាងមួយទេ។



## មេរៀនទី១២

### CSS Position

CSS Position ប្រើសម្រាប់ធ្វើការកំណត់លក្ខណៈនៃទីតាំងរបស់html elementណាមួយ។ CSS Position មានដូចជា៖

- Position: static; គឺជាdefaultរបស់position។
- Position: fixed; ប្រើសម្រាប់ឱ្យhtml elementណាមួយនៅនឹងថ្កល់
- Position: absolute; ប្រើសម្រាប់បញ្ជាក់ទៅលើhtml elementឱ្យទៅលើ ក្រោម ឆ្វេង ស្តាំ។ កាលណា យើងប្រើប្រាស់position: absoluteយើងអាចប្រើproperty: left, right, bottom, top
- Position: relative; ប្រើដើម្បីគ្រប់គ្រងទៅលើposition: absolute
- Position: sticky; ធ្វើឱ្យធាតុមានទីតាំងប្រែប្រួលទៅតាមការរំកិលរបស់ទំព័រ។

Z-Index: ធាតុដែលបង្កើតក្រោយ តែងនៅពីលើធាតុដែលបង្កើតមុន។ លក្ខណៈនេះបើសិនជាចង់ផ្លាស់ប្តូរ វា យើងត្រូវប្រើz-indexដែលមានតម្លៃជាលេខកំណត់ឱ្យជាន់របស់ធាតុមួយនេះ។

Ex: z-index: ជាតម្លៃលេខ;

### លំហាត់អនុវត្តន៍

1. តើ CSS Positionគេប្រើប្រាស់សម្រាប់ធ្វើអ្វី?
2. តើ Property position មាន value អ្វីខ្លះចូលរៀបរាប់ និងបកស្រាយឱ្យបានភ្លឺរក្សាយ?
3. តើ Position: absolute គេប្រើប្រាស់សម្រាប់ធ្វើអ្វី? ហើយអាចប្រើបាន property អ្វីខ្លះចូល រៀបរាប់។

## មេរៀនទី១៣

### CSS Float

CSS Float ប្រើសម្រាប់ឱ្យhtml elementអណ្តែតទៅឆ្វេងឬស្តាំនៅក្នុងគេហទំព័រ(website)។ ភាគ ច្រើនCSS Float គេតែងតែប្រើសម្រាប់ធ្វើការបង្កើត layout website។ CSS Float មាន

- Left: ប្រើសម្រាប់ឱ្យ html elementអណ្តែតទៅឆ្វេង
- Right: ប្រើសម្រាប់ឱ្យ html elementអណ្តែតទៅស្តាំ
- None: ប្រើសម្រាប់ឱ្យ html elementមិនអណ្តែត

### Example

```
.box {
width: 200px;
height: 200px;
background-color: red;
float: right;
}
```

### លំហាត់អនុវត្តន៍

1. តើCSS floatគេប្រើសម្រាប់ធ្វើអ្វី?
2. តើCSS Float មាន propertyអ្វីខ្លះ? ចូររៀបរាប់។
3. ចូរអនុវត្តន៍លំហាត់ដូចខាងក្រោម

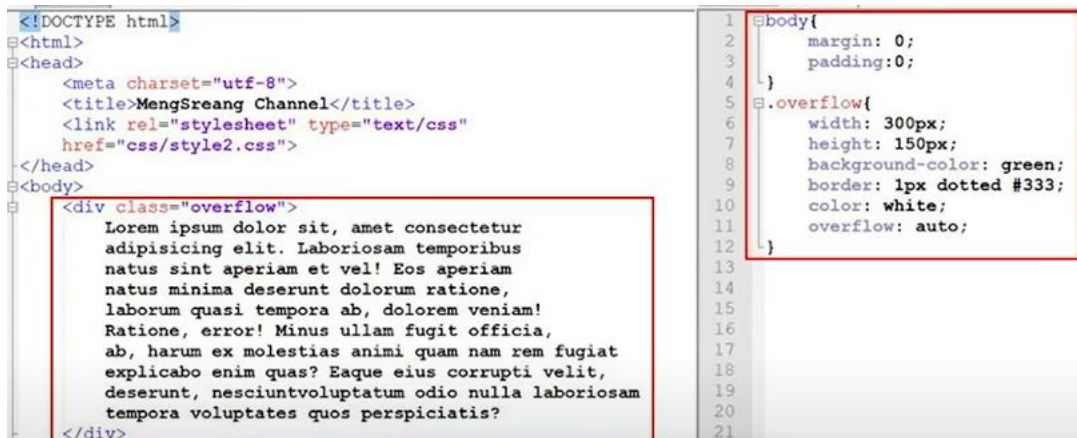


មេរៀនទី១៤

## CSS Overflow

ប្រើសម្រាប់កាត់នៅផ្ទៃប្រអប់ដែលធំជាងអ្វីដែលយើងបានកំណត់ឬក៏បន្ថែម scrollbars សម្រាប់អ្នកមើលទៅវិញទៅមក។ នៅក្នុង overflow មានតម្លៃដូចខាងក្រោម៖

- Visible: តម្លៃដើម គឺមិនកាត់អ្វីទាំងអស់។
- Hidden: កាត់ចោលទាំងអស់។
- Scroll: កាត់ចោលដែរ តែបន្ថែមនូវ scrollbar សម្រាប់អ្នកមើលទៅវិញទៅមក។
- Auto: ដូចscrollដែរ តែវាមានលក្ខណៈពិសេសជាងscrollទៅទៀត។



បន្ថែមពីនេះទៀតគឺ វាមាន២ propertiesថែមទៀតគឺ៖

- Overflow-x: ប្រើសម្រាប់កាត់នៅផ្ទៃប្រអប់ធំជាងអ្វីដែលយើងបានកំណត់ពីឆ្វេងទៅស្តាំ។
- Overflow-y: ប្រើសម្រាប់កាត់នៅផ្ទៃប្រអប់ធំជាងអ្វីដែលយើងបានកំណត់ពីលើទៅក្រោម។ ហើយតម្លៃវាដូចនឹងoverflowដែរ។

មេរៀនទី១៤

## CSS Pseudo

Pseudo ប្រើសម្រាប់ចាប់យកនូវធាតុមួយឬផ្នែកណាមួយនៅស្ថានភាពពិសេសណាមួយ។ pseudo មានពីរប្រភេទគឺ៖

- Pseudo-class: សម្រាប់ចាប់យកនូវធាតុមួយនៅក្នុងស្ថានភាពពិសេសមួយ។
- Pseudo-element: សម្រាប់ចាប់យកនូវធាតុមួយនៅក្នុងស្ថានភាពពិសេសមួយ។

១. Pseudo-Class:



នៅក្នុងpseudo-classយើងត្រូវសរសេរដោយប្រើសញ្ញា ( : ) រួចដាក់ឈ្មោះនៃpseudoនោះ។ ហើយក្នុងpseudo-classនេះ យើងមានរបស់ត្រូវសិក្សាជាច្រើនដូចក្នុងតារាងខាងក្រោម៖

| Pseudo-class Selector | ឧទាហរណ៍             | អន្ថន័យ                                              |
|-----------------------|---------------------|------------------------------------------------------|
| :active               | p:active            | នៅពេលចុច Mouse ពីលើ(ចុចសង្កត់ទុក)                    |
| :hover                | h1: hover           | នៅពេលយក mouse រំលងលើ                                 |
| :link                 | a:link              | តំណភ្ជាប់ដែលមិនទាន់បានចូលមើល                         |
| :visited              | a:visited           | តំណភ្ជាប់ដែលបានចូលមើលរួចហើយ                          |
| :root                 | :root               | ចាប់យកឬសនៃទំព័រ html នឹង* ដែរ                        |
| :target               | #hello:target       | ចាប់យកធាតុដែលមានidហើយទំព័របច្ចុប្បន្នដំណើរការលើidនោះ |
| :empty                | p:empty             | ចាប់យកធាតុដែលគ្មានអ្វីក្នុងនោះសោះ                    |
| :lang                 | p:lang(hey)         | ចាប់យកធាតុដែលមាន attribute lang តម្លៃ hey            |
| :first-child          | p:first-child       | ចាប់យកធាតុដែលជាកូនដំបូងគេ                            |
| :first-of-type        | p:first-of-type     | ចាប់យកធាតុដែលជាកូនដំបូងនៅក្នុងចំណោមធាតុដូចគ្នា       |
| :last-child           | p:last-child        | ចាប់យកធាតុដែលជាកូនចុងក្រោយគេ                         |
| :last-of-type         | p:last-of-type      | ចាប់យកធាតុដែលជាកូនចុងក្រោយនៅក្នុងចំណោមធាតុដូចគ្នា    |
| :nth-child(n)         | p:nth-child(2)      | ចាប់យកធាតុដែលនៅលេខរៀងទី n                            |
| :nth-of-type(n)       | p:nth-of-type(2)    | ចាប់យកធាតុដែលនៅលេខរៀងទី n នៅក្នុងចំណោមធាតុដូចគ្នា    |
| :nth-last-child(n)    | p:nth-last-child(2) | ចាប់យកធាតុដែលនៅលេខរៀងទី n ដែលរាប់ពីក្រោយ             |
| :only-child           | P:only-child        | ចាប់យកធាតុដែលជាកូនទោល                                |
| :only-of-type         | P:only-of-type      | ចាប់យកធាតុដែលជាកូនទោលនៅក្នុងចំណោមធាតុដូចគ្នា         |
| :not(selector)        | :not(p)             | ចាប់យកធាតុដែលមិនមែនជាselector                        |
| :checked              | Input:checked       | ចាប់យកធាតុដែលត្រូវបានជ្រើសរើស                        |
| :disabled             | Input:disabled      | ចាប់យកធាតុដែលកំណត់ disabled attribute                |
| :enabled              | Input:enabled       | ចាប់យកធាតុដែលមិនបានកំណត់disabled attribute           |
| :optional             | Input:optional      | ចាប់យកធាតុដែលមិនបានកំណត់required attribute           |
| :required             | Input:required      | ចាប់យកធាតុដែលបានកំណត់ required attribute             |
| :read-only            | Input:read-only     | ចាប់យកធាតុដែលបានកំណត់readonly attribute              |
| :read-write           | Input:read-write    | ចាប់យកធាតុដែលមិនបានកំណត់readonly attribute           |
| :invalid              | Input:invalid       | ចាប់យកធាតុដែលបានកំណត់ invalid attribute              |
| :valid                | Input:valid         | ចាប់យកធាតុដែលបានកំណត់ valid attribute                |
| :in-range             | Input:in-range      | ចាប់យកធាតុដែលបានកំណត់ក្នុងព្រំដែន                    |
| :out-of-range         | Input:out-of-range  | ចាប់យកធាតុដែលមិនបានកំណត់នៅក្នុងព្រំដែនបានកំណត់       |
| :focus                | Input:focus         | ចាប់យកធាតុដែលបានផ្ដោតលើ                              |

Ex:

```
<title>Document</title>
<link rel="stylesheet" href="Test20.c
</head>
<body>
 <!-- <p></p> -->
 <p>Welcome</p>
 <p>To</p>
 <p lang="hey">Yttrium</p>
 <p>Information</p>
 <p>Technology</p>
 <p>Center</p>
</body>
</html>
```

```
1 body{
2 margin: 0;
3 padding: 0;
4 }
5
6 /* empty */
7 p:empty{
8 width: 50px;
9 border: 2px solid green;
10 padding: 10px;
11 }
12
13 /* :lang */
14 p:lang(hey){
15 width: 50px;
16 color: green;
17 background-color: pink;
18 padding: 10px;
19 }
```

```
<title>Document</title>
<link rel="stylesheet" href="Test20.c
/head>
body>
 <p>Welcome</p>
 <p>To</p>
 <p lang="hey">Yttrium</p>
 <p>Information</p>
 <p>Technology</p>
 <p>Center</p>
/body>
/html>
```

```
1 body{
2 margin: 10px;
3 padding: 0;
4 }
5
6 /* :first-child
7 p:first-child{
8 width: 50px;
9 color: green;
10 background-color: pink;
11 padding: 10px;
12 }
13 */
14 /* :last-child */
15
16 p:last-child{
17 width: 50px;
18 color: green;
19 background-color: green;
20 padding: 10px;
21 }
```

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta http-equiv="X-UA-Compatible" co
 <meta name="viewport" content="width=
 <title>Document</title>
 <link rel="stylesheet" href="Test20.c
</head>
<body>
 <h1>Hello</h1>
 <p>Welcome</p>
 <p>To</p>
 <p lang="hey">Yttrium</p>
 <p>Information</p>
 <p>Technology</p>
 <p>Center</p>
 <h1>Hello world</h1>
</body>
</html>
```

```
1 body{
2 margin: 10px;
3 padding: 0;
4 }
5
6 /*
7 p:first-of-type{
8 width: 50px;
9 color: green;
10 background-color: pink;
11 padding: 10px;
12 }
13 */
14 p:last-of-type{
15 width: 50px;
16 color: green;
17 background-color: pink;
18 padding: 10px;
19 }
20
```

```
est20.html x
ic > LessonCSS > <> Test20.html > html > body SITE > Static > LessonCSS > # Test20.css >
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta http-equiv="X-UA-Compatible" co
 <meta name="viewport" content="width=
 <title>Document</title>
 <link rel="stylesheet" href="Test20.c
</head>
<body>
 <p>Welcome</p>
 <p>To</p>
 <p>Yttrium</p>
 <p>Information</p>
 <p>Technology</p>
 <p>Center</p>
</body>
</html>
```

```
1 body{
2 margin: 10px;
3 padding: 0;
4 }
5
6 p:nth-child(4){
7 width: 80px;
8 color: green;
9 background-color: pink;
10 padding: 10px;
11 }
12
13 p:nth-last-child(1){
14 width: 80px;
15 color: green;
16 background-color: pink;
17 padding: 10px;
18 }
19
```

```

<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta http-equiv="X-UA-Compatible" content="width=device-width, initial-scale=1">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <title>Document</title>
 <link rel="stylesheet" href="Test20.css">
</head>
<body>
 <h1>Hello</h1>
 <p>Welcome</p>
 <p>To</p>
 <p>Yttrium</p>
 <p>Information</p>
 <p>Technology</p>
 <p>Center</p>
 <h1>Hello world</h1>
</body>
</html>

```

```

1 body{
2 margin: 10px;
3 padding: 0;
4 }
5
6 p:nth-of-type(3){
7 width: 80px;
8 color: green;
9 background-color: pink;
10 padding: 10px;
11 }
12
13 p:nth-last-of-type(3){
14 width: 80px;
15 color: green;
16 background-color: pink;
17 padding: 10px;
18 }
19

```

```

<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <title>MengSreang Channel</title>
 <link rel="stylesheet" type="text/css" href="css/style2.css">
</head>
<body>
 <div><p>Hello</p></div>
 <div>Hello Pseudo<p>Hello CSS</p></div>
</body>

```

```

1 body{
2 margin: 0;
3 padding: 0;
4 }
5 p:only-child{
6 width: 50px;
7 color: green;
8 background-color: pink;
9 padding: 10px;
10 }
11
12

```

```

<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <title>MengSreang Channel</title>
 <link rel="stylesheet" type="text/css" href="css/style2.css">
</head>
<body>
 <div><p>Hello</p></div>
 <div><p>Hello Pseudo</p><p>Hello CSS</p></div>
</body>

```

```

1 body{
2 margin: 0;
3 padding: 0;
4 }
5 p:only-of-type{
6 width: 50px;
7 color: green;
8 background-color: pink;
9 padding: 10px;
10 }
11
12

```

```

<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <title>MengSreang Channel</title>
 <link rel="stylesheet" type="text/css" href="css/style2.css">
</head>
<body>
 <h1>Hello Pseudo</h1>
 <p>Hello</p>
 <p>Hi</p>
 <p>Hey</p>
 <p>Bye</p>
 <h1>Hello Pseudo</h1>

```

```

1 body{
2 margin: 0;
3 padding: 0;
4 }
5 p{
6 color: red;
7 }
8 :not(p){
9 color: green;
10 }
11
12
13
14
15

```

```

<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <title>MengSreang Channel</title>
 <link rel="stylesheet" type="text/css"
 href="css/style2.css">
</head>
<body>
 <label>Are You:</label>

 <input type="checkbox" value="male">
 <label>Male</label>

 <input type="checkbox" value="female">
 <label>Female</label>

 <input type="checkbox" value="others">
 <label>Others</label>
</body>

```

```

1 body{
2 margin: 0;
3 padding: 0;
4 }
5 input:checked{
6 width: 30px;
7 height: 30px;
8 }
9
10
11
12
13
14
15
16
17

```

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>MengSreang Channel</title>
6 <link rel="stylesheet" type="text/css"
7 href="css/style2.css">
8 </head>
9 <body>
10 <label>Please Input Your Email & Password:</label>

11 <label>Email:</label>
12 <input type="email" value="email" id="email">

13 <label>Password:</label>
14 <input type="password" value="password" id="password" disabled="disabled">
15 </body>

```

```

1 body{
2 margin: 10px;
3 padding: 0;
4 }
5 #email:enabled{
6 background-color: orange;
7 }
8 #password:disabled{
9 background-color: #ddd;
10 }
11
12
13
14
15

```

```

<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <title>MengSreang Channel</title>
 <link rel="stylesheet" type="text/css"
 href="css/style2.css">
</head>
<body>
 <label>Read Write:</label>

 <input type="text" value="text" id="textWrite">

 <label>Read Only:</label>

 <input type="text" value="text" id="textOnly" readonly="readonly">
</body>

```

```

1 body{
2 margin: 10px;
3 padding: 0;
4 }
5 #textWrite:read-write{
6 background-color: orange;
7 }
8 #textOnly:read-only{
9 background-color: #ddd;
10 }
11
12
13

```

```

<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <title>MengSreang Channel</title>
 <link rel="stylesheet" type="text/css"
 href="css/style2.css">
</head>
<body>
 <label>In-Range:</label>

 <input type="number" value="7" id="numInRange" max="10" min="1">

 <label>Out-Of-Range:</label>

 <input type="text" value="15" id="numOutOfRange" max="10" min="1">
</body>

```

```

1 body{
2 margin: 10px;
3 padding: 0;
4 }
5 #numInRange:in-range{
6 background-color: orange;
7 }
8 #numOutOfRange:out-of-range{
9 background-color: #ddd;
10 }
11
12
13
14

```

```

<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <title>MengSreang Channel</title>
 <link rel="stylesheet" type="text/css"
 href="css/style2.css">
</head>
<body>
 <label>Please Input Your Email & Password:</label>

 <label>Email:</label>
 <input type="email" value="email" id="email">

 <label>Password:</label>
 <input type="password" value="password" id="password">
</body>

```

```

1 body{
2 margin: 10px;
3 padding: 0;
4 }
5 input:focus{
6 background-color: orange;
7 }
8
9
10
11
12
13
14
15

```

□. Pseudo-Element:



នៅក្នុងPseudo-Element នេះ យើងត្រូវសរសេរដោយប្រើសញ្ញា ( :: ) រួចដាក់ឈ្មោះនៃpseudoនោះ។ ហើយក្នុង pseudo -element នេះ យើងមានរបស់ត្រូវសិក្សាជាច្រើនដូចក្នុងតារាងខាងក្រោម៖

Pseudo-Element Selector	ឧទាហរណ៍	អត្ថន័យ
::after	p::after	ចាប់យកផ្នែកក្នុងប្រព័ន្ធនៃធាតុបន្ទាប់ធាតុបច្ចុប្បន្ន
::before	p::before	ចាប់យកផ្នែកក្នុងប្រព័ន្ធនៃធាតុមុនធាតុបច្ចុប្បន្ន
::first-letter	p::first-letter	ចាប់យកផ្នែកក្នុងប្រព័ន្ធនៃធាតុនៅក្នុងអក្សរដំបូង
::first-line	p::first-line	ចាប់យកផ្នែកក្នុងប្រព័ន្ធនៃធាតុនៅបន្ទាត់ដំបូង
::selection	p::selection	ចាប់យកផ្នែកក្នុងប្រព័ន្ធនៃធាតុនៅពេលselectនៅលើធាតុណាមួយ

```

<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <title>MengSreang Channel</title>
 <link rel="stylesheet" type="text/css" href="css/style2.css">
</head>
<body>
 <div>Hello Pseudo-Element.</div>
</body>
</html>

```

```

1 body{
2 margin: 0;
3 padding:0;
4 }
5 div{
6 width: 340px;
7 height: 50px;
8 background-color: orange;
9 }
10 div::before{
11 content:"Pseudo-Element: ";
12 background-color: white;
13 }
14 div::after{
15 content:" Thank!";
16 background-color: pink;
17 }

```

```

<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <title>MengSreang Channel</title>
 <link rel="stylesheet" type="text/css" href="css/style2.css">
</head>
<body>
 <div>Hello Pseudo-Element.Lorem ipsum dolor sit amet consectetur adipisicing elit. Ullam labore,consequuntur optio ratione eum quasi dolore quod excepturi exercitationem temporibus.
</div>
</body>
</html>

```

```

1 body{
2 margin: 0;
3 padding:0;
4 }
5 div{
6 width: 500px;
7 background-color: orange;
8 }
9 div::first-letter{
10 background-color: green;
11 }
12 div::first-line{
13 background-color: pink;
14 }
15
16

```

```

<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <title>MengSreang Channel</title>
 <link rel="stylesheet" type="text/css" href="css/style2.css">
</head>
<body>
 <div>Hello Pseudo-Element.Lorem ipsum dolor sit amet consectetur adipisicing elit. Ullam labore,consequuntur optio ratione eum quasi dolore quod excepturi exercitationem temporibus.
</div>
</body>
</html>

```

```

1 body{
2 margin: 0;
3 padding:0;
4 }
5 div{
6 width: 500px;
7 background-color: orange;
8 }
9 div::selection{
10 background-color: yellow;
11 color: red;
12 }
13
14
15

```

## មេរៀនទី១៦ CSS Opacity

ប្រើសម្រាប់កំណត់ភាពស្រអាប់ទៅឱ្យធាតុនីមួយៗ។ ជាធម្មតា opacity property in css គឺគេចូលចិត្តប្រើមួយ pseudo-class គឺប្រើជាមួយ :hover ដែលនៅពេលដែល user យក mouse អូសពីលើធាតុដែលយើងបានកំណត់នោះ :hover នឹងមានសកម្មភាព។ opacity property មានតម្លៃចាប់ពី 0 to 1។



```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>MengSreang Channel</title>
6 <link rel="stylesheet" type="text/css" href="css/style2.css">
7 </head>
8 <body>
9 <div class="opacityOne">
10
11 Hello Opacity: 0.2
12 </div>
13 <div class="opacityTwo">
14
15 Hello Opacity: 0.7
16 </div>
17 </body>
18 </html>

```

```

1 body{
2 margin: 0;
3 padding:0;
4 }
5 .div{
6 width: 200px;
7 height: 120px;
8 float: left;
9 }
10 .div img{
11 width: 100%;
12 height: 100%;
13 }
14 .opacityOne{
15 opacity: 0.2;
16 }
17 .opacityOne:hover{
18 opacity: 1;
19 }
20 .opacityTwo{
21 opacity: 0.7;
22 margin-left: 10px;
23 }
24 .opacityTwo:hover{
25 opacity: 1;
26 }

```

មេរៀនទី៧ CSS Shadow

Shadow នៅក្នុង CSS គឺប្រើសម្រាប់ដាក់ស្រមោលទៅឲ្យអក្សរឬក៏ធាតុផ្សេងៗទៀត។ ហើយវាមាន ២ property :

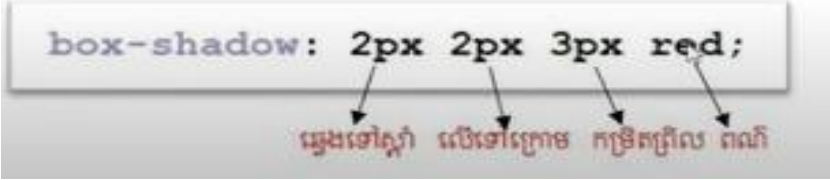
- text-shadow: ប្រើសម្រាប់ដាក់ស្រមោលទៅឲ្យអក្សរ។
- Box-shadow: ប្រើសម្រាប់ដាក់ស្រមោលទៅឲ្យធាតុផ្សេងៗទៀត។



1. Text-Shadow: យើងអាចកំណត់តម្លៃបាន ៤ តម្លៃ មាន៖ ទំហំពីរន្ទងទៅស្តាំ ទំហំពីរលើទៅក្រោម ទំហំនៃកម្រិតព្រិល និង ពណ៌។



2. Box-Shadow: យើងអាចកំណត់តម្លៃបាន ៤ តម្លៃ មាន៖ ទំហំពីរន្ទងទៅស្តាំ ទំហំពីរលើទៅក្រោម ទំហំនៃកម្រិតព្រិល និង ពណ៌។



```
OCTYPE html>
ml lang="en">
ad>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE
<meta name="viewport" content="width=device-wi
<title>
| Shadow
</title>
<link rel="stylesheet" href="Test21.css">
ead>
dy>
<h1>Welcome to Yttrium IT.</h1>

ody>
tml>

1 body{
2 margin: 10px;
3 padding: 0;
4 box-sizing: border-box;
5 font-family: monospace;
6 }
7 h1{
8 text-align: center;
9 font-size: 2rem;
10 text-shadow: 3px 5px 2px gray;
11 }
12 img{
13 width: 300px;
14 display: block;
15 margin: auto;
16 box-shadow: -5px 5px 2px gray, 3px 5px 2px
17 }
```

## មេរៀនទី១៨

### CSS Transform 2D

ដើម្បីផ្លាស់ប្តូរទ្រង់ទ្រាយអ្វីមួយនៅក្នុង css យើង ត្រូវប្រើ transform property។ ដែលនៅក្នុង transform មាន property ដូចជា៖

- translate(x,y): ប្រើសម្រាប់រកិលធាតុ។
- translateX(n): ប្រើសម្រាប់រកិលធាតុទៅអក្ស X។
- translateY(n): ប្រើសម្រាប់រកិលធាតុទៅអក្ស Y។
- scale(x,y): ប្រើសម្រាប់ពង្រីក ឬ បង្រួមធាតុ។
- scaleX(n): ប្រើសម្រាប់ពង្រីក ឬ បង្រួមធាតុនៅអក្ស X។
- scaleY(n): ប្រើសម្រាប់ពង្រីក ឬ បង្រួមធាតុនៅអក្ស Y។
- rotate(angle): ប្រើសម្រាប់បង្វិលធាតុ។
- skew(x,y): ប្រើសម្រាប់បញ្ជ្រកធាតុ។
- skewX(angle): ប្រើសម្រាប់បញ្ជ្រកធាតុទៅអក្ស X។
- skewY(angle): ប្រើសម្រាប់បញ្ជ្រកធាតុទៅអក្ស Y។

```
WEBSITE > Static > LessonCSS > <> Test22.html > html > head
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <link rel="stylesheet" href="Test22.css">
8 </head>
9 <body>
10 <div class="normal">normal box</div>
11 <div class="translate">Hello Translate box</div>
12 <div class="scale">Hello scale box</div>
13 <div class="rotate">Hello rotate box</div>
14 <div class="skew">Hello skew box</div>
15 </body>
16 </html>

WEBSITE > Static > LessonCSS > # Test22.css
1 body{
2 margin: 10px;
3 padding: 0;
4 box-sizing: border-box;
5 font-family: monospace;
6 }
7 h1{
8 text-align: center;
9 font-size: 2rem;
10 text-shadow: 3px 5px 2px gray;
11 }
12 div{
13 width: 200px;
14 height: 200px;
15 float: left;
16 text-align: center;
17 }
18 .normal{
19 background-color: brown;
20 }
21 .translate{
22 margin-top: 150px;
23 background-color: chocolate;
24 transform: translate(30px, 40px);
25 }
26 .scale{
27 text-align: center;
28 margin-top: 150px;
29 background-color: rgb(14, 13, 12);
30 color: white;
31 transform: scale(2.1);
32 }
33 .rotate{
34 text-align: center;
35 margin-top: 550px;
36 background-color: yellowgreen;
37 color: white;
38 transform: rotate(90deg, 30deg);
39 }
40 .skew{
41 text-align: center;
42 background-color: yellowgreen;
43 color: white;
44 transform: skew(10deg, 10deg);
45 }
```

## មេរៀនទី១៩

### CSS Transform 3D

ចំណុច transform 3D នេះគឺយើងមានរបស់ត្រូវសិក្សាច្រើន ហើយនៅតែយក Transform property មកប្រើ ដដែល សម្រាប់ property មាន៖

- translate(x,y,z): ប្រើសម្រាប់រកិលធាតុ។
- translateX(n): ប្រើសម្រាប់រកិលធាតុទៅអក្ស X។
- translateY(n): ប្រើសម្រាប់រកិលធាតុទៅអក្ស Y។
- translateZ(n): ប្រើសម្រាប់រកិលធាតុទៅអក្ស Z។

- scale(x,y,z): ប្រើសម្រាប់ពង្រីក ឬ បង្រួមធាតុ។
- scaleX(n): ប្រើសម្រាប់ពង្រីក ឬ បង្រួមធាតុនៅអ័ក្ស X។
- scaleY(n): ប្រើសម្រាប់ពង្រីក ឬ បង្រួមធាតុនៅអ័ក្ស Y។
- scaleZ(n): ប្រើសម្រាប់ពង្រីក ឬ បង្រួមធាតុនៅអ័ក្ស Z។
- rotate(x,y,z,angle): ប្រើសម្រាប់បង្វិលធាតុ។
- rotateX(angle): ប្រើសម្រាប់បង្វិលធាតុទៅអ័ក្ស X។
- rotateY(angle): ប្រើសម្រាប់បង្វិលធាតុទៅអ័ក្ស Y។
- rotateZ(angle): ប្រើសម្រាប់បង្វិលធាតុទៅអ័ក្ស Z។

## មេរៀនទី២០

### CSS Transition

Transition ប្រើសម្រាប់ផ្តល់នូវអន្តរភាពទៅឲ្យធាតុអ្វីមួយដែលអន្តរភាពនេះគឺមានលក្ខណៈយ៉ាងរលូន។

transition property:

- transition-duration: សម្រាប់កំណត់រយៈពេលនៃអន្តរភាព ដែលតម្លៃវាជា វិនាទីឬមីលីវិនាទី។
- transition-delay: សម្រាប់ពន្យារពេលដំណើរការនៃអន្តរភាព ដែលតម្លៃវាជា វិនាទីឬមីលីវិនាទី។
- transition-property: សម្រាប់ចាប់យកpropertyមួយណាដែលយើងចង់កំណត់អន្តរភាពទៅឲ្យវា ឬបើចង់ចាប់ទាំងអស់គឺត្រូវប្រើ all ដើម្បីឲ្យ property ទាំងអស់មានអន្តរភាព។
- transition-timing-function: សម្រាប់កំណត់ទម្រង់ល្បឿន ដែលតម្លៃវាមានដូចជា ease, linear, ease-in, ease-out, ease-in-out។

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>transform 2D</title>
 <link rel="stylesheet" href="Test22.css">
</head>
<body>
 <div class="box-1">Linear</div>
 <div class="box-2">Ease</div>
 <div class="box-3">Ease-in</div>
 <div class="box-4">Ease-out</div>
 <div class="box-5">Ease-in-out</div>
</body>
</html>
```

```
1 body{
2 margin: 10px;
3 padding: 0;
4 box-sizing: border-box;
5 font-family: monospace;
6 }
7 div{
8 width: 100px;
9 text-align: center;
10 background-color: pink;
11 padding: 50px;
12 font-weight: bold;
13 cursor:pointer;
14 margin-bottom: 10px;
15 opacity: 0.5;
16 transition-duration: .5s ;
17 transition-property: opacity,border-radius;
18 }
19
20 div:hover{
21 background-color: black;
22 color: aquamarine;
23 opacity: 1;
24 border-radius: 20px 40px;
25 }
26 .box-1{
27 transition-timing-function: linear;
28 }
29 .box-2{
30 transition-timing-function: ease;
31 }
32 .box-3{
33 transition-timing-function: ease-in;
34 }
35 .box-4{
36 transition-timing-function: ease-out;
37 }
38 .box-5{
39 transition-timing-function: ease-in-out;
40 }
```



## ការប្រើ Transition តែមួយ Property:

```
transition: all 0.5s linear 2s;
```

transition-property transition-duration transition-timing-function transition-delay

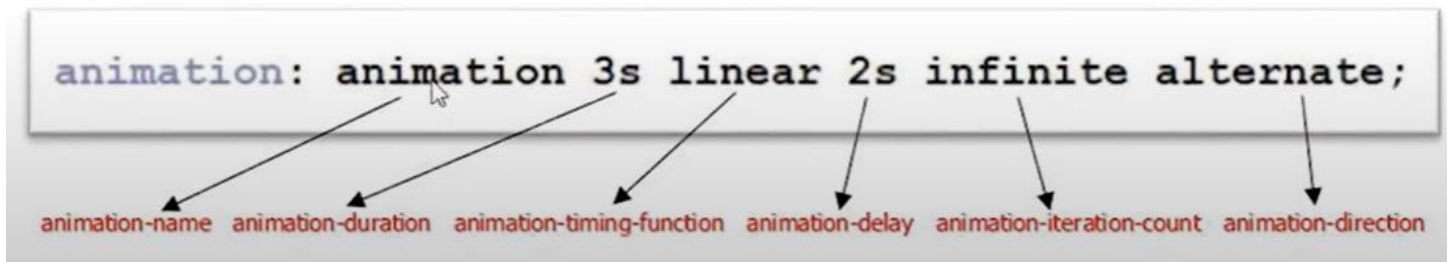
មេរៀនទី២១

### CSS Animation

Animation ប្រើសម្រាប់កំណត់ឲ្យធាតុនីមួយៗមានចលនាវិលទៅវិញទៅមក។ វាមានPropertyជាច្រើនដូចជា៖

- animation-name: ប្រើសម្រាប់កំណត់ឈ្មោះឲ្យ @keyframes។
- animation-duration: សម្រាប់កំណត់ពេលវេលានៃការធ្វើចលនា ដែលតម្លៃវាជា វិនាទីឬមីលីវិនាទី។
- animation-delay: សម្រាប់ពន្យារពេលនៃការធ្វើចលនា ដែលតម្លៃវាជា វិនាទីឬមីលីវិនាទី។
- animation-iteration-count: សម្រាប់កំណត់ចំនួនដងនាំការធ្វើចលនា ដែលតម្លៃយើងអាចកំណត់បានពីរគឺចំនួនដងជាលេខ និង infiniteដំណើរច្រើនដែលមិនឈប់។
- animation-direction: សម្រាប់កំណត់ទិសដៅនៃការធ្វើចលនា ដែលតម្លៃវាមានបីគឺ
  1. normal: ធម្មតា
  2. reverse: ធ្វើចលនាថយក្រោយ
  3. alternate: ធ្វើចលនាទៅមុខហើយថយក្រោយ
  4. alternate-reverse: ធ្វើចលនាថយក្រោយហើយទៅមុខ
- animation-timing-function: សម្រាប់កំណត់ល្បឿននៃការធ្វើចលនា ដែលតម្លៃមានដូចជា linear, ease, ease-in, ease-out, ease-in-out។
- animation-fill-mode: ប្រើសម្រាប់កំណត់ស្ដាយឲ្យធាតុ នៅពេលដែលចលនាមិនទាន់ចាប់ផ្ដើមឬនៅពេលបញ្ចប់ ដែលតម្លៃវាមានបួនគឺ
  1. none: មិនបានកំណត់ស្ដាយ
  2. forwards: នៅពេលធ្វើចលនាចប់ ស្ដាយរបស់វាដូចទៅនឹងពេលដែលចប់នេះឯង
  3. backwards: នៅពេលមិនទាន់ធ្វើចលនា ស្ដាយរបស់វាដូចទៅនឹងនៅពេលចាប់ផ្ដើមធ្វើចលនា
  4. both: មានន័យថាដំណើរការទាំងforwards & backwards។
- @keyframes: ប្រើសម្រាប់កំណត់ទម្រង់ចលនា ដែលតម្លៃយើងអាចប្រើជា ភាគរយ(%) ក៏បានឬចង់ប្រើជាអក្សរក៏បាន ដែលមានពីរគឺ from និង to ដែល from ស្មើនឹងតម្លៃ 0% ហើយ to ស្មើនឹងតម្លៃ100%។

## ការប្រើ Animation តែមួយ Property:



Ex:

```
WEBSITE > Static > LessonCSS > <> Test22.html > html
9 </head>
10 <body>
11 <div class="box-1">Linear</div>
12 <div class="box-2">Ease</div>
13 <div class="box-3">Ease-in</div>
14 <div class="box-4">Ease-out</div>
15 <div class="box-5">Ease-in-out</div>

16 <p class="box-6">hello all this is css animation</p>
17 </body>
18 </html>

WEBSITE > Static > LessonCSS > # Test22.css > ...
41 /* css animation tool */
42 .box-6{
43 width: 300px;
44 height: 300px;
45 background-color: cadetblue;
46 text-align: center;
47 line-height: 300px;
48 font-size: 15px;
49 /* animation: name;
50 animation-duration: 5s;
51 animation-iteration-count: infinite;
52 animation-timing-function: linear;
53 animation-direction: alternate; */
54 animation: name 5s linear infinite alternate;
55 }
56 /* យើងអាចដាក់ជាការយល់បាននៅក្នុងkeyframes */
57 @keyframes name {
58 from{
59 background-color: blue;
60 color: chartreuse;
61 }
62 to{
63 background-color: crimson;
64 color: cornsilk;
65 }
66 }
67
```

## មេរៀនទី២២

### CSS Flexbox

Flexbox ប្រើសម្រាប់កំណត់ធាតុនីមួយៗឲ្យមានភាពស្របគ្នា និងងាយស្រួលក្នុងការកំណត់ទីតាំង។ នៅក្នុង flexbox នេះគឺយើងមានរបស់ពីរដែលត្រូវសិក្សាគឺ Parent Element និង Child Element។



1. Parent Element: មាន properties ជាច្រើនដែលយើងត្រូវសិក្សា ដែលមានដូចខាងក្រោម៖
  - Flex-direction: សម្រាប់កំណត់ទិសដៅនៃធាតុដែលវាផ្ទុក។  
=>column, row, column-reverse, row-reverse
  - Flex-wrap: សម្រាប់កំណត់ថា ធាតុដែលវាផ្ទុកត្រូវចុះបន្ទាត់ឬអត់ នៅពេលដល់កន្លែងទាល់។

=> nowrap, wrap, wrap-reverse

- Flex-flow: សម្រាប់សរសេរតែក្នុង property រវាង flex-direction & flex-wrap។

=> row wrap.....

- Justify-content: សម្រាប់តម្រៀបទៅតាមទិសដៅគោល។

=> center, flex-start, flex-end, space-around, space-between, space-evenly

- Align-content: សម្រាប់តម្រៀបទៅតាមទិសដៅដែលកាត់ទិសដៅគោល (សម្រាប់ច្រើនបន្ទាត់)។

=> center, flex-start, flex-end, space-around, space-between, stretch

- Place-content: សម្រាប់សរសេរតែក្នុង property រវាង justify-content និង align-content។

=> center center .....

- Align-items: សម្រាប់តម្រៀបទៅតាមទិសដៅកាត់ទិសដៅគោល (សម្រាប់មួយបន្ទាត់)។

```
SITE > Static > LessonCSS > flex-box > home.html > html > body > div.flex-box
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Flexbox</title>
<link rel="stylesheet" href="flex.css">
</head>
<body>
 <div class="flex-box">
 <div class="box-1">1</div>
 <div class="box-2">2</div>
 <div class="box-3">3</div>
 <div class="box-4">4</div>
 <div class="box-5">5</div>
 <div class="box-6">6</div>
 <div class="box-7">7</div>
 <div class="box-8">8</div>
 <div class="box-9">9</div>
 <div class="box-10">10</div>
 </div>
</body>
</html>

WEBSITE > Static > LessonCSS > flex-box > flex.css
1 body{
2 margin: 10px;
3 padding: 0;
4 box-sizing: border-box;
5 font-family: 'Lucida Sans', 'Lucida Sans Reg
6 }
7 .flex-box {
8 background-color: #ghostwhite;
9 height: 100vh;
10 display: flex;
11 /* flex-direction: column; */
12 /* flex-direction: column-reverse; */
13 flex-direction: row;
14 /* flex-direction: row-reverse; */
15 flex-wrap: wrap;
16 /* flex-wrap: wrap-reverse; */
17 /* flex-flow: row wrap; */
18 /* justify-content: center; */
19 /* justify-content: space-between; */
20 align-content: center; /*
21 /* place-content: center space-between; */
22 justify-content: space-around;
23 align-items: center;
24 }
25 .flex-box div{
26 padding: 20px;
27 text-align: center;
28 width: 100px;
29 height: 20px;
30 margin: 10px;
31 background-color: #chocolate;
32 border-radius: 6px;
33 }
```

## 2. Child Element: មាន properties ជាច្រើនដែលយើងត្រូវសិក្សា ដែរមានដូចខាងក្រោម៖

- Order: សម្រាប់កំណត់លំដាប់
  - Flex-grow: សម្រាប់ពង្រីកទំហំទៅឲ្យធាតុនូវទំហំដែលនៅសល់
  - Flex-shrink: សម្រាប់បង្រួមធាតុ
  - Flex-basis: សម្រាប់កំណត់ទំហំទៅឲ្យធាតុ
  - Flex: សម្រាប់សរសេរតែក្នុងមួយ property រវាង flex-grow, flex-shrink & flex-basis។
  - Align-self: សម្រាប់កំណត់តម្លៃដូច align-items ប៉ុន្តែសម្រាប់តែមួយធាតុបង្កបង្កើននេះទេ។
- => center, flex-start, flex-end, stretch, baseline

```
SITE > Static > LessonCSS > flex-box > <> home.html > html
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Flexbox</title>
<link rel="stylesheet" href="flex.css">
</head>
<body>
 <div class="flex-box">
 <div class="box-1">1</div>
 <div class="box-2">2</div>
 <div class="box-3">3</div>
 <div class="box-4">4</div>
 </div>
</body>
</html>

WEBSITE > Static > LessonCSS > flex-box > # flex.css
7 .flex-box {
8 background-color: #ghostwhite;
9 height: 100vh;
10 display: flex;
11 flex-direction: row;
12 flex-wrap: wrap;
13 }
14
15 .flex-box div{
16 padding: 20px;
17 text-align: center;
18 width: 100px;
19 height: 20px;
20 margin: 10px;
21 background-color: #chocolate;
22 border-radius: 6px;
23 }
24
25 .box-1{
26 /* order: 4;
27 flex-grow: 1; */
28 /* flex-basis: 200px; */
29 /* flex: 0; this is flex-shrink */
30 /* flex: 1; */
31 }
32 .box-2{
33 /* order: 3;
34 flex-grow: 2; */
35 /* flex: 2; */
36 align-self: flex-end;
37 }
38 .box-3{
39 /* order: 2;
40 flex-grow: 3; */
41 /* flex: 3; */
42 }
43 .box-4{
44 /* order: 1;
45 flex-grow: 4; */
46 /* flex-basis: 200px; */
47 /* flex: 3; */
48 }
```

មេរៀនទី២៣

CSS Grid

អ្នកគ្រូសិក្សារមៀនflex-boxជាមុនសិន មុននឹងអ្នកសិក្សារមៀនមួយនេះព្រោះថា grid មានសមត្ថភាពលើសពី flex boxទៅទៀត។

Grid ប្រើ property display ដូច flexbox ដែរវាមានpropertiesជាច្រើនដូចជា៖

- Grid-template-columns: ប្រើសម្រាប់កំណត់ជួរឈរ។ តម្លៃវាមាន fr(fractional), auto, max-content, min-content, minmax(min,max), repeat(ចំនួនប្រវែង)។
- Grid-auto-row: ប្រើសម្រាប់កំណត់ទំហំឱ្យជួរដេក។
- Grid-gap: សម្រាប់សរសេរតែមួយproperty រវាង grid-row-gap & grid-column-gap។
- Grid-column-start: ប្រើសម្រាប់កំណត់ការចាប់ផ្តើមនៃបន្ទាត់ជួរឈរ។
- Grid-column-end: សម្រាប់កំណត់ការបញ្ចប់នៃបន្ទាត់ជួរឈរ។
- Grid-row-start: សម្រាប់កំណត់ការចាប់ផ្តើមនៃបន្ទាត់ជួរដេក។
- Grid-row-end: សម្រាប់កំណត់ការបញ្ចប់នៃបន្ទាត់ជួរដេក។
- Grid-column: សម្រាប់កំណត់ការចាប់ផ្តើមនិងបញ្ចប់នៃបន្ទាត់ជួរឈរ។
- Grid-row: សម្រាប់កំណត់ការចាប់ផ្តើមនិងបញ្ចប់នៃបន្ទាត់ជួរដេក។
- Grid-area: សម្រាប់កំណត់ជាការassignតម្លៃជាឈ្មោះនៅក្នុងធាតុនីមួយៗហើយយើងនឹងបង្ហាញតម្លៃទាំងអស់នោះនៅក្នុងgrid-template-area
- Justify-content: សម្រាប់តម្រៀបទៅតាមទិសដៅផ្នែក(សម្រាប់ធ្វើច្រើនបន្ទាត់)។ Center, start, end, space-around, space-between, space-evenly, stretch
- Align-content: សម្រាប់តម្រៀបទិសដៅដែលកាត់ទិសដៅបញ្ឈរ (សម្រាប់ធ្វើច្រើនបន្ទាត់)។ តម្លៃរបស់វាមានដូចជា center, start, end, space-around, space-between, space-evenly, stretch
- Justify-items: សម្រាប់តម្រៀបទៅតាមទិសដៅដែលកាត់ទិសដៅផ្នែក (សម្រាប់មួយបន្ទាត់)។ តម្លៃរបស់វាមានដូចជាcenter, start, end, stretch។

```
TE > Static > LessonCSS > grid > <> grid.html > html > head > link
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-
 <title>Grid</title>
 <link rel="stylesheet" type="text/css" href="grid.css">
</head>
<body>
 <div class="grid-template">
 <div class="box-0">0</div>
 <div class="box-1">1</div>
 <div class="box-2">2</div>
 <div class="box-3">3</div>
 <div class="box-4">4</div>
 <div class="box-5">5</div>
 <div class="box-6">6</div>
 <div class="box-7">7</div>
 <div class="box-8">8</div>
 <div class="box-9">9</div>
 </div>
</body>
</html>

WEBSITE > Static > LessonCSS > grid > # grid.css > .box-1
1 body{
2 margin: 10px;
3 padding: 0;
4 box-sizing: border-box;
5 font-family: sans-serif;
6 }
7 .grid-template{
8 display: grid;
9 grid-template-columns: repeat(5, 1fr);
10 grid-gap: 10px;
11 background-color: #rgb(173, 173, 221);
12 }
13 .grid-template div{
14 background-color: #crimson;
15 padding: 10px;
16 text-align: center;
17 }
18 .box-1{
19 /* grid-column-start: 1;
20 grid-column-end: 4;
21 grid-row-start: 1;
22 grid-row-end: 3; */
23 /* grid-column-start: span 2;
24 grid-row-start: span 2; */
25 /* grid-column: 1/4;
26 grid-row: 1/3; */
27 grid-column: 1 / span 3;
28 grid-row: 1 / span 2;
29 }
```

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Grid</title>
 <link rel="stylesheet" type="text/css" href="grid.css">
</head>
<body>
 <div class="grid-template">
 <div class="box-0">0</div>
 <div class="box-1">1</div>
 <div class="box-2">2</div>
 <div class="box-3">3</div>
 <div class="box-4">4</div>
 <div class="box-5">5</div>
 <div class="box-6">6</div>
 <div class="box-7">7</div>
 <div class="box-8">8</div>
 <div class="box-9">9</div>
 </div>
</body>
</html>

1 body{
2 margin: 10px;
3 padding: 0;
4 box-sizing: border-box;
5 font-family: sans-serif;
6 }
7 .grid-template{
8 display: grid;
9 grid-template-columns: repeat(5, 1fr);
10 grid-gap: 10px;
11 background-color: #rgb(173, 173, 221);
12 grid-template-areas: "box-0 box-0 box-2 box-3 box-3";
13 "box-5 box-6 box-7 box-8 box-9";
14 }
15 .grid-template div{
16 background-color: #crimson;
17 padding: 10px;
18 text-align: center;
19 }
20 .box-0{
21 grid-area: box-0;
22 }
23 .box-1{
24 grid-area: box-1;
25 }
26 .box-2{
27 grid-area: box-2;
28 }
29 .box-3{
30 grid-area: box-3;
31 }
32 .box-4{
33 grid-area: box-4;
34 }
35 .box-5{
36 grid-area: box-5;
37 }
38 .box-6{
39 grid-area: box-6;
40 }
41 .box-7{
42 grid-area: box-7;
43 }
44 .box-8{
45 grid-area: box-8;
46 }
47 .box-9{
48 grid-area: box-9;
49 }
50 }
```

## មេរៀនទី២៤

## CSS Media Query

@media mediatype and (expresssions) {

CSS-Code;

}

- ⇒ Mediatype : all, screen
- ⇒ And : សម្រាប់ភ្ជាប់រវាងរូបសម្ព័ន្ធ 2 ឬច្រើនគ្នា។
- ⇒ Expressions : max-width, min-width, orientation: portrait and landscape

```
WEBSITE > Static > LessonCSS > media > <> index.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>Media</title>
8 <link rel="stylesheet" href="style.css"/>
9 </head>
10 <body>
11 <h1>Responsive "Yttrium IT" Media</h1>
12 </body>
13 </html>

WEBSITE > Static > LessonCSS > media > # style.css > {} @media {
1 body{
2 margin: 0;
3 padding: 0;
4 box-sizing: border-box;
5 font-family: Verdana, Geneva, Tahoma, sans-serif;
6 }
7 h1{
8 color: red;
9 text-align: center;
10 }
11 /* @media (min-width: 500px) and (max-width: 600px){
12 h1{
13 color:black;
14 background-color: blue;
15 }
16 }
17 @media (min-width: 700px) and (max-width: 800px){
18 h1{
19 color: chocolate;
20 background-color: deepskyblue;
21 }
22 } */
23 /* @media (orientation: portrait){
24 h1{
25 color: blue;
26 background-color: crimson;
27 }
28 } */
29 @media (orientation: landscape){
30 h1{
31 color: blue;
32 background-color: rgb(212, 172, 180);
33 }
34 }
```

## មេរៀនទី២៥

## CSS Variable

ដើម្បីបង្កើតvariable នៅក្នុង CSS បានយើងត្រូវប្រើfunction var() ។ variable នៅក្នុង CSS ភាគច្រើនគេប្រកាស វាជា global scope នៅក្នុង :root & body ។ ដើម្បីបង្កើត variable បានយើងត្រូវដាក់សញ្ញាទ្រូពីរ(--)ហើយឈ្មោះ អញ្ជាតដែលយើងចង់ដាក់ ហើយពេលហៅប្រើយើងត្រូវដាក់ var(--variable\_name) ។

```
WEBSITE > Static > LessonCSS > <> Test24.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>variable</title>
8 <link rel="stylesheet" href="Test24.css">
9 </head>
10 <body>
11 <div class="box-1">Hello Blue</div>
12 <div class="box-2">Hello Green</div>
13 <div class="box-3">Hello Orange</div>
14 <div class="box-4">Hello Yellow</div>
15 <div class="box-5">Hello Red</div>
16 </body>
17 </html>

WEBSITE > Static > LessonCSS > # Test24.css > <> div
1 :root{
2 --mainBlue: #8E44AD;
3 --mainGreen: #F39C12;
4 --mainOrange: #F1C40F;
5 --mainYellow: #DFF000;
6 --mainRed: #FF7F50;
7 }
8 body{
9 padding: 0;
10 margin: 0;
11 box-sizing: border-box;
12 font-family: Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;
13 }
14 div{
15 margin: 10px;
16 }
17 .box-1{
18 color: var(--mainBlue);
19 background: var(--mainRed);
20 }
21 .box-2{
22 color: var(--mainGreen);
23 background: var(--mainYellow);
24 }
25 .box-3{
26 color: var(--mainOrange);
27 background: var(--mainRed);
28 }
29 .box-4{
30 color: var(--mainYellow);
31 background-color: var(--mainGreen);
32 }
33 .box-5{
34 color: var(--mainRed);
35 background-color: var(--mainBlue);
36 }
```