Programming Language

Hyper Text Markup Language

HTML

Benner-Lee

Tag, Attribute, Entity

1.Tag: ជាឈ្មោះកំនត់រួចជាស្រេច សំរាប់ធ្វើអ្វីមួយ

មានពីរប្រភេទ៖

* Has content <tagname>Content</tagname>
* Empty: <tagname />

2.Attribute: ជាឈ្មោះកំនត់រួចជាស្រេច សំរាប់បំពេញបន្ថែមអោយ Tag

<tagname attributename=”value” …>Content</tagname>

Ex: <p align=”right”>This is my paragraph!</p>

<tagname attributename=”value” …./>

Ex: <hr color=”#FF0000” color=”blue”/>

3.Entity: ជាឈ្មោះកំនត់រួចជាស្រេច តាងអោយសញ្ញាពិសេសអ្វីមួយ

&entitiname; Ex: x &gt; y

4.Tag Form Text:

<i> (italic) ,<em> (emhasis),<small> (Make text smaller)

<b> (bold) ,<big>,<strong>

<u> (underline)

<strike>Hello</strike> (~~Hello~~)

x<sup>2</sup> (x2)

x<sub>1</sub> (x1)

<p>Hello sok</p>

<pre>hello sok</pre>

<span>Hello</span>

<font>

<marquee>

<center>

<br/>

<hr/>

+List

* OrderList

<ol type=”1,i,I,a,A” start=”n”>

<li>Item</li>

<li>item</li>

</ol>

* UnorderList

<ul type=”disc,square,circle”>

<li>Item</li>

<li>Item</li>

</ul>

DefinitionList

<dl>

<dt>

Item

<dd>Description</dd>

</dt>

……………………..

</dl>

+Table

**Tag**:

<table>

<tr>

<th>

<td>

**Attribute**:

border

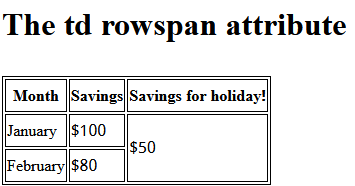
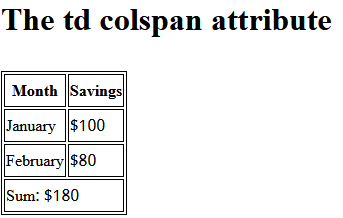
align

width

height

colspan

rowspan



<table border=”2px” >  
  <tr>  
    <th>Month</th>  
    <th>Savings</th>  
    <th>Savings for holiday!</th>  
  </tr>  
  <tr>  
    <td>January</td>  
    <td>$100</td>  
    <td rowspan="2">$50</td>  
  </tr>  
  <tr>  
    <td>February</td>  
    <td>$80</td>  
  </tr>  
</table>

<table>  
  <tr>  
    <th>Month</th>  
    <th>Savings</th>  
  </tr>  
  <tr>  
    <td>January</td>  
    <td>$100</td>  
  </tr>  
  <tr>  
    <td>February</td>  
    <td>$80</td>  
  </tr>  
  <tr>  
    <td colspan="2">Sum: $180</td>  
  </tr>  
</table>

5.Image

<img

src=”URL”

width=””

height=” “

align=” “

border=”. “

title=”. “ />

6.Sound

<audio controls>

<source src="horse.ogg" type="audio/ogg">

<source src="horse.mp3" type="audio/mpeg">

</audio>

7.Video

<video width="320" height="240" controls>

<source src="movie.mp4" type="video/mp4">

<source src="movie.ogg" type="video/ogg">

</video>

8.Link (ចុចឆ្លង) <a>

* link in page (anchor):

<a name=”name”>

<a href=”#name”>Label</a>

* link cross page

<a href=”url”>Label</a>

**\*All tag here:** [**1. html-tags-chart**](1.%20html-tags-chart.pdf)

Cascading Style Sheet

CSS

ជាភាសា ជួយបន្ថែមសំរស់ មានភាពទាក់ទាញ ភាពចាប់អារម្មណ៏ (style) បន្ថែមភាពស្អាតទៅអោយ Tag របស់ HTML។

រៀន CSS មានន័យថារៀនយកTag HTML មកបន្ថែមភាពស្អាត (Property) តាមតំរូវកាយើងចង់បាន។

Tags

CSS

Tags modified

\* CSS អាចត្រូវបានសរសេរ 2 បែប៖ Internal & External

1. **Internal CSS:**

សរសេរផ្ទាល់ជាមួយ HTML File

2 បែប៖ Inline & Embedded

* **Inline:** យក Tag ណាដែលយើងចង់តែងមកដាក់ **attribute** style

រួចដាក់ **Property** អោយវា៖

**Syntax:**

<tagname style = “property:value; …”> Content</tagname>

Or <tagname style=”property:value; …”/>

* **Embedded:** យក Style មកប្រកាសនៅ block head ដោយដាក់នៅចន្លោះ Tag style

**Syntax:**

<style>

Declarations Style

</style>

ដើម្បីប្រកាស CSS style គេមាន ម៉ូតដូចខាងក្រោម៖

* **Selector with Tag**: យកTag ណាដែលយើងបន្ថែមភាពស្អាតមកប្រកាស

**Syntax**:

Tag { property:value; ……}

Tag1, Tag2, … { property:value; …….}

* **Selector with ID:** ធ្វើ Tag មួយអោយមាន style ច្រើន mode

ឬ ធ្វើ mode មួយអោយប្រើបានច្រើន Tag ។

**Syntax:**

Tag1#name{ property:value; }

Tag2#name { property:value; }

#name{ property:value; }

**Call**

<Tag id=”name”>Content</Tag>

<Tag id=”name” …/>

* **Selector with Class:** ធ្វើ Tag មួយអោយមាន style ច្រើន mode

ឬ ធ្វើ mode មួយអោយប្រើបានច្រើន Tag ។

**Syntax:**

Tag.name{ property:value; }

.name{ property:value; }

**Call**

<Tag class=”name”>Content</Tag>

<Tag class=”name” …/>

**+ Inherit:** តែងអោយ Tag មួយ លុះត្រាតែប្រើក្នុង Tag មួយទៀត

**Syntax:**

**Tag1 > Tag2 { property:value; …….}**

**Or**

**Tag1 Tag2 {property:value; }**

**+ Pseudo:** Pseudo-class and Pseudo-Element

**Syntax:**

**Tag::pseudo-class{. …… }**

1. **External CSS**

បង្កើត CSS File ដាច់ដោយឡែកពី HTML File ដោយក្នុងFile CSS នេះយើងអាចប្រកាស Mode អីក៏បាន selector, selector ID, selector CLASS, inherit, Pseudo ប៉ុន្តែ File ត្រូវ save យក extension.css ហើយហៅចូលទៅក្នុង HTML File block head បាន 2 បែបដូចខាងក្រោម៖

<**link** rel=”stylesheet” href=”url” />

Or

<**style**>

@import url(“url”);

</**style**>

Box:

width, min-width, max-width

height, min-height, max-height

border

border-top, border-bottom, border-left, border-right

margin

margin-left ,margin-right, margin-top, margin-bottom

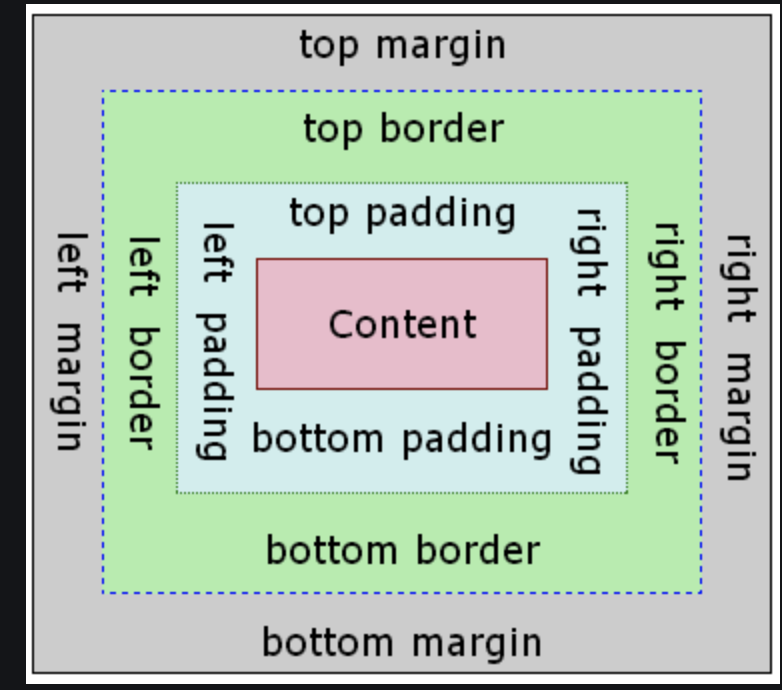
padding

padding-left, padding-right, padding-top, padding-bottom

float:

display:

clear:



p.dotted {outline-style: dotted;}  
p.dashed {outline-style: dashed;}  
p.solid {outline-style: solid;}  
p.double {outline-style: double;}  
p.groove {outline-style: groove;}  
p.ridge {outline-style: ridge;}  
p.inset {outline-style: inset;}  
p.outset {outline-style: outset;}

Layout:

ចូរសរសរប្រើ css ដើម្បីបង្កើត layout web page ដូចខាងក្រោម៖

1.

|  |  |
| --- | --- |
|  | |
|  |  |

2.

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |
|  | | |

3.

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |
|  |
|  |
|  | | |

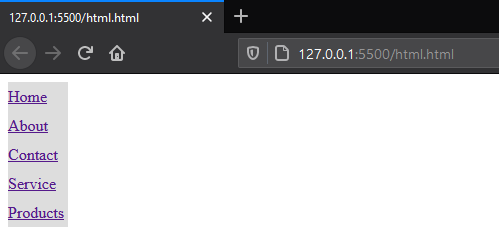
4.

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |
|  |
|  |
|  | | |

5.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
|  |  |  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  | | | | | |

**Menu bar** example:



**JavaScript**

ជាប្រភេទ script language មានប្រយោជន៏សំរាប់លាយ (embedded) ជាមួយ HTML ដើម្បីបង្កើត Web dynamic (input/output) ផ្នែក client site ។

Web Static = Web Read only

Web Dynamic = (user input/output) = ផ្លាស់ប្តូ Content របស់ Web Page បាន ។

ដើម្បីដាក់ JavaScript ចូល Web Page គេអាចធ្វើបានពីរបែបគឺ តាម Internal និង External។

1. **Internal JavaScript**

គឺ JavaScript និង HTML ស្ថិតក្នុង File តែមួយ ដោយប្រើ tag <script> ។

**Syntax**:

<script>

statements;

(declaration variable, input/output, operator, control structure)

</script>

statements: open source ប្រកាន់អក្សរតូច អក្សរធំហើយបញ្ចប់ដោយ ( ; )

1. **Declaration Variable:**

**Syntax**: var name1, name2, …, nameN;

1. **Input / Output**

**Input:**

- **Dialog**: prompt(label, default)

- **HTML Form**

- **HTML DOM**

**Output:**

- **Dialog**: alert(data),confirm(data)

- **building function**: document.write(data)

- **HTML Form**

- **HTML DOM**

1. **Operator:** ដូច ជាមួយ c, c++, Java, C#

- **Arithmethic**: +,-,\*,/,+=,-=,\*=,/=, \*\* (exponent)

- **Compare operator:** >,<,>=,<=,==, ===, !=, !==

- **Relational Operator:** &&, II, !

1. **Control Structure:** ដូចជាមួយ c,c++,Java,C#

if(..){. }

if(..){. } else {. …. }

exp? st1:st2;

switch (exp){

case cont1: st1; break;

……………………………

default: stn+1;

}

Do {…} while(exp);

while(exp) {. }

for()

1. **Function**

**Syntax:**

function name(){ ……………}

function name(){……………… return exp;}

function name(parameter){…………. …. }

function name(parameter){………………. return exp;}

1. **Input /Output with HTML Form:**

* **Form**

<form name = “formName”>

Element of Form

</form>

* **TextBox**

<input type= “text” name= “….” value= “…” />

<input type= “number” name= “…” value= “…” />

<input type= “email”. name= “….” / >

<input type= “password” name= “….” />

<textarea name = “…” rows = “…” cols= “…”></textarea>

* **Radio**

<input type=”radio” name=”….” value=” …”>Label

* **Checkbox**

<input type=”checkbox” name= “…” value=”…”>Label

* **ComboBox**

<select name=”…”>

<option value=”…”>Label</option>

<option value=”…”>Label</option>

………………..

</select>

* **list box**

<select name=”…” size=”…” multiple>

<option value=”…”>Label</option>

<option value=”…”>Label</option>

………………..

</select>

* **Button**

<input type=”button” name=”…” value=”…” onXX=”funtionName()” />

<input type=”submit” value=”….” />

<input type=”reset” value= “…”/>

* **Access Object**

document.formName.elementName.value

document.forms[index].elementName.value

document.formName.elemens[index].value

document.forms[index].elements[index].value

**// help to understand**

JTextField x=new JTextField();

x.setText("Hello");

String k=x.getText();

TextBox x=new TextBox();

x.Text="Hello";

String k=x.Text;

<**form** name="f">

<**input** type="text" name="x" />

</**form**>

document.f.x.value="Hello";

var k=document.f.x.value;

1. **HTML DOM (Document Object Model)**

វិធីសាស្រ្តធ្វើTag អោយទៅជា Object ដើម្បើយក ទិនិ្នន័យចូល និង ចេញ

**Syntax:**

<tag id=”objectname”></tag>

Or <tag id=”objectname” />

JTextField x=new JTextField();

x.setText(“Hello”);

String k=x.getText();

TextBox x=new TextBox();

x.Text=”Hello”;

String k=x.Text;

<**form** name=”f”>

<**input** type=”text”. name=”x”. />

</**form**>

document.f.x.value=”Hello”;

var k=document.f.x.value;

<**input** type=”text” id=”x” />

document.getElementById(“x”).value=”Hello”;

var k=document.getElementById(“x”).value;

**Accessor**

* **Text**

document.getElementById(“objectname”).innerHTML

**<input />, <textarea></textarea>, <select></select>**

document.getElementById(“objectname”).value

* **image, sound, video:**

document.getElementById(“objectname”).src

* **link**:

document.getElementById(“objectname”).href

* **style (css)**

document.getElementById(“objectname”).style.properyName

1. **របៀប Add Event**

មានបីបែប៖

**Syntax**:

1.

<tag onXX=”functionName()”></tag>

<tag onXX=”functionName()” />

<script>

function functionName() {…}

</script>

2.

<tag id=”objectname”></tag>

<tag id=”objectname” />

<script>

document.getElementById (“objectname”).onXX = function() {…}

</script>

3.

<tag id=”objectname”></tag>

<tag id=”objectname” />

<script>

document.getElementById(“objectname”)

. addEventListener(“xx”, functionName);

function functionName() {…}

</script>

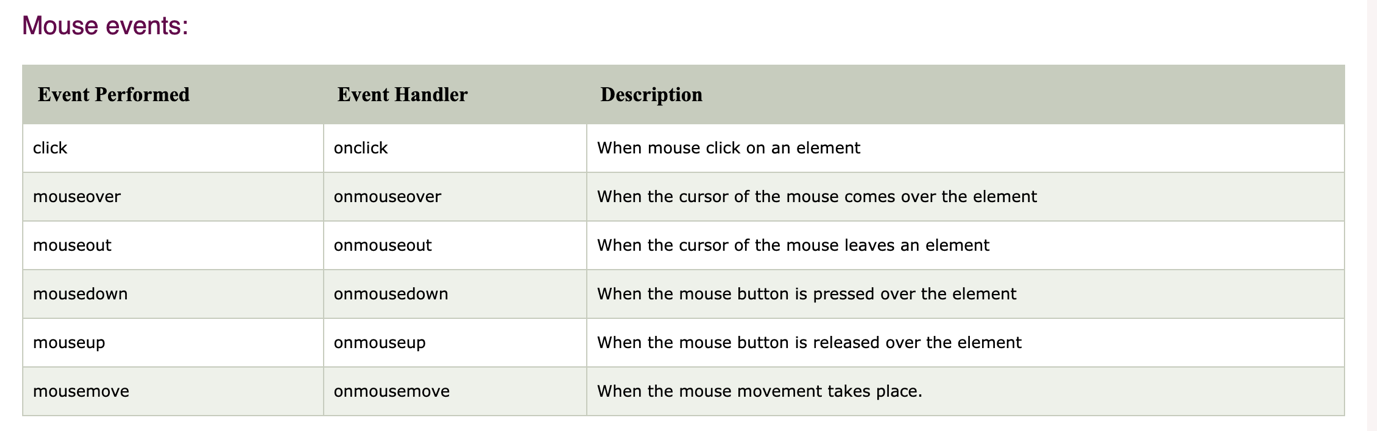
1. **External JavaScript**

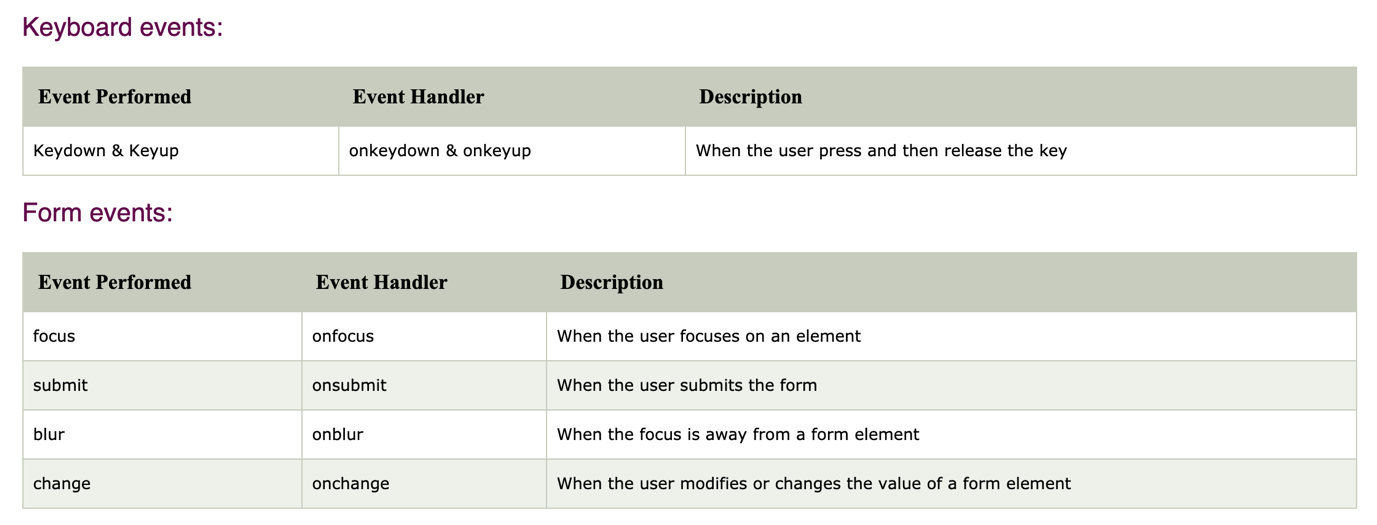
* បង្កីត JavaScript ផ្តាច់ចេញពី HTML
* ជាទូទៅក្នុង File JavaScript ផ្ទុក variable , function
* អាច save ឈ្មោះអីក៏បានដែរ ប៉ុន្តែត្រូវយក extension “.js” ហើយត្រូវហៅចូលក្នុង HTML File ។

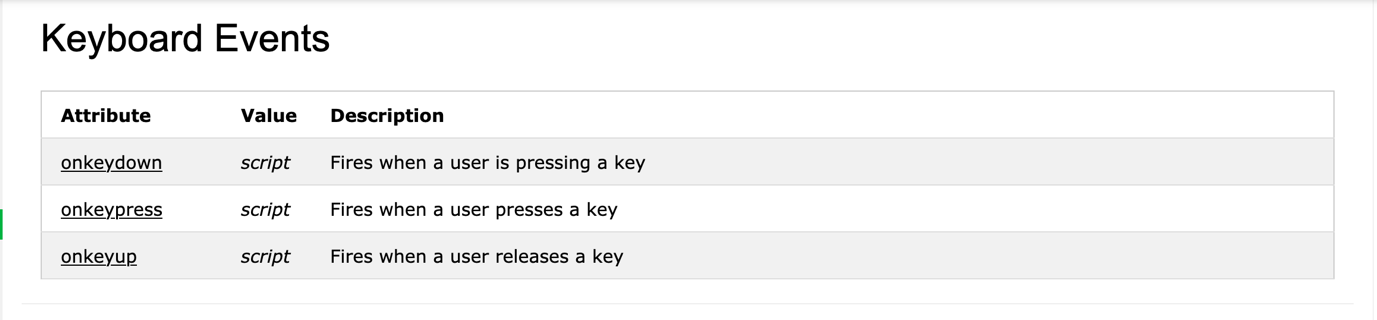
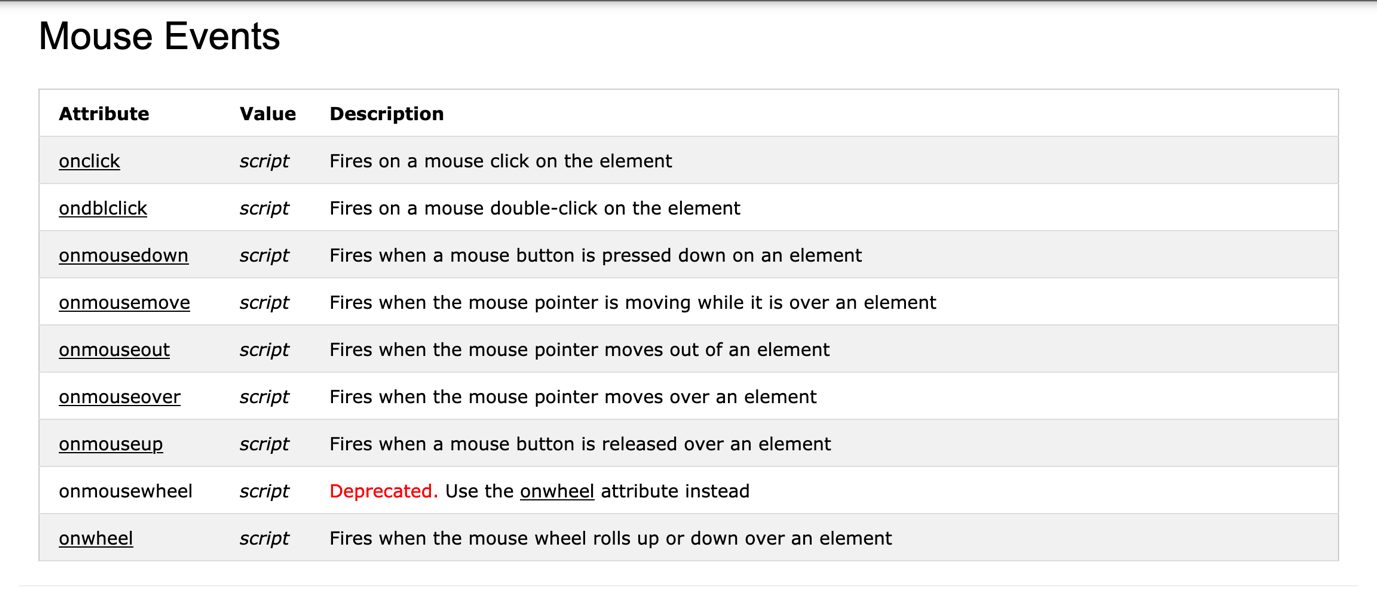
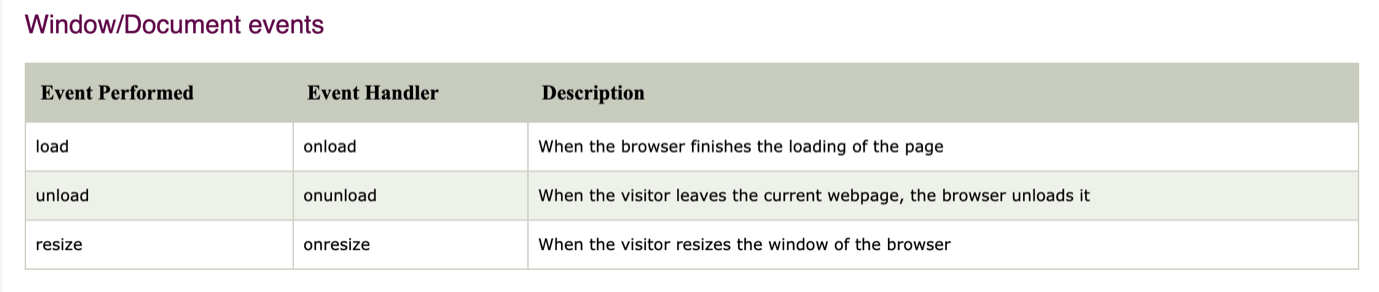
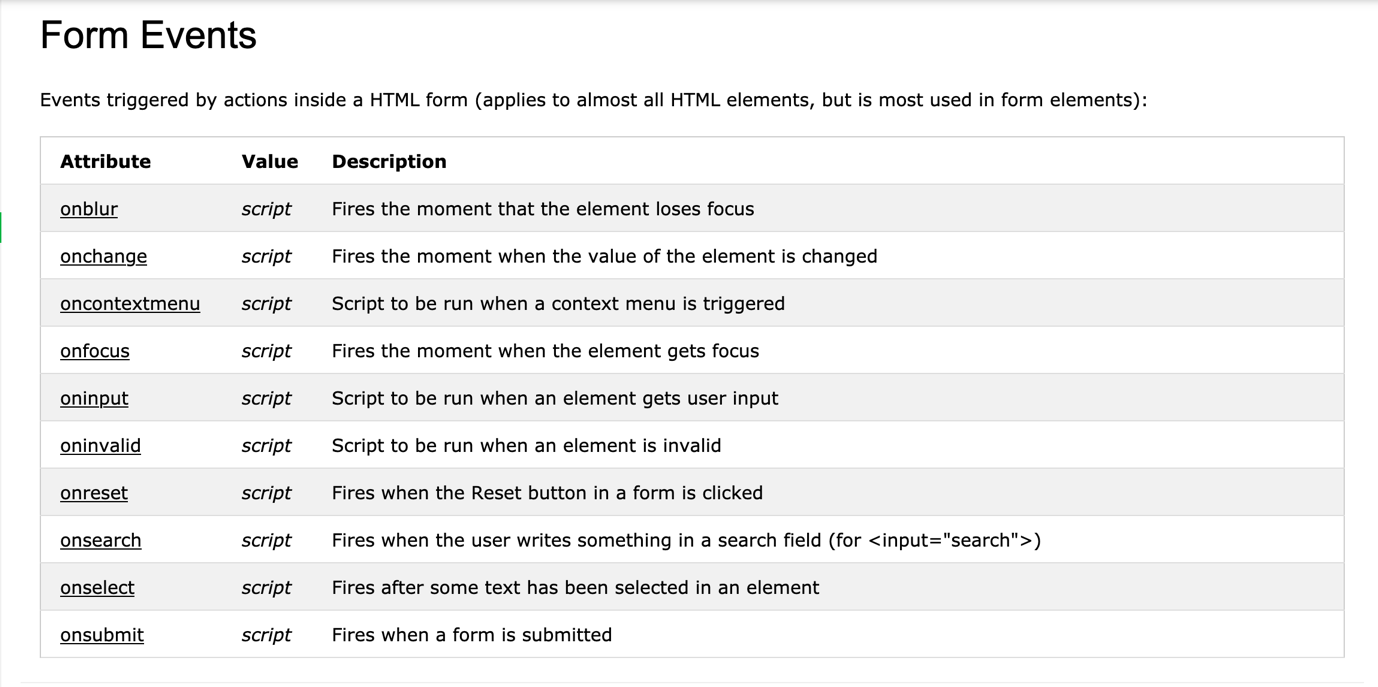
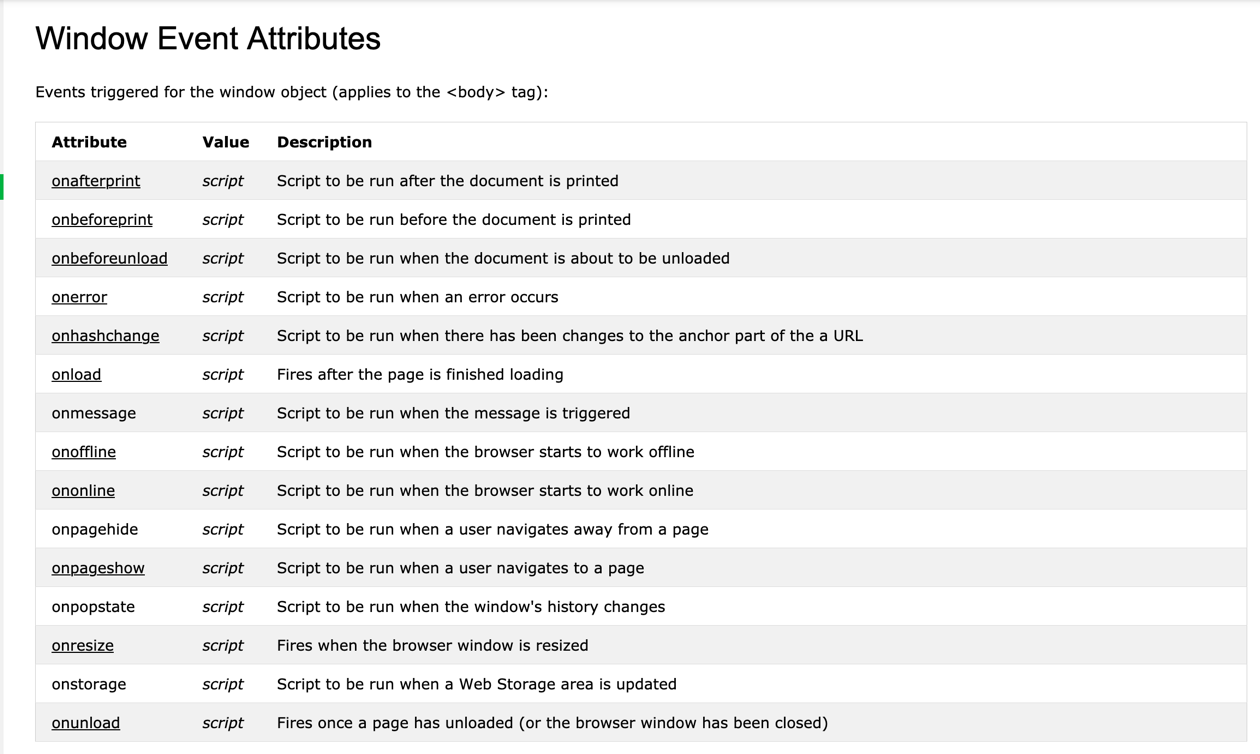
<script src=”URL” ></script>

1. **JavaScript Event**

Event សំខាន់ៗក្នុង JavaScript រួមមាន៖







JQuery

គឺជាសំណុំ Java Script Function មានស្រាប់ ដែល Developer បានបង្កើតទុកអោយស្រាប់ ដើម្បីអោយយើងប្រើប្រាស់បន្ត ងាយស្រួលក្នុងការ Developer កូដបានលឿន វាមានប្រយោជន៏សំរាប់ធ្វើអោយ Web Page Dynamic ផ្នែក client side។

គឺជា Front-End Framework.

គឺជា Client-Side Framework.

គឺជា JavaScript Framework.

គឺជា external JavaScript library

ដើម្បីប្រើប្រាស់ វាគេអាចធ្វើបាន 2 របៀប៖

* Download library connection locale.
* CDN

**1.Connection Locale:**

-Download. <https://jquery.com/download/>

ដាក់ library jquery: jquery-3.5.1.min.js ចូលក្នុង Folder js

បន្ទាប់មកភា្ចប់ទៅកាន់វា ៖

**<script type=”text/javascript” src=”../js/jquery-3.5.1.min.js”></script>**

រួចហៅ function jquery ប្រើ៖

<script>

$(function () {

statements;

});

</script>

ជ្រើសរើសធាតុ (Selector): $(selector)

$(“tag”)

$(“tag#id”)

$(“#id”)

$(“tag.classname”)

$(“.classname”)

$(“[attributename]”);

Action (Event) ដូច Javascript ទាំងអស់ តែ អត់មាន on ទេ

click, dblick, mouseover, hover, …

$(selector).action (function () {

});

Access Data:

* text: val(), text(), html()
* images, sound, video, link: attr()
* style: css(), addClass(), removeClass(), ….

**\* For more detail of JQuery:** [20. jQuery\_pptx](20.%20jQuery_pptx.pptx)**Example 1: keyup, val(), text(v)**

<html>

  <head>

    <script src="../js/jquery-3.5.1.min.js"></script>

    <script>

      $(document).ready(function () {

        $("input").keyup(function () {

          var v;

          v = $("input").val();

          $("h1").text(v);

        });

      });

    </script>

  </head>

  <body>

    <input type="text" />

    <h1></h1>

  </body>

</html>

**Example 2: click, val(s)**

<html>

  <head>

    <script src="../js/jquery-3.5.1.min.js"></script>

    <script

      $(document).ready(function () {

        $("#bt1").click(function () {

          var a, b, s;

          a = parseFloat($("#tfa").val());

          b = parseFloat($("#tfb").val());

          s = a + b;

          $("#tfr").val(s);

        });

        $("#bt2").click(function () {

          var a, b, s;

          a = parseFloat($("#tfa").val());

          b = parseFloat($("#tfb").val());

          s = a \* b;

          $("#tfr").val(s);

        });

      });

    </script>

  </head>

  <body>

    <center>

<h1>Calcultator</h1>

      <hr />

      <table>

        <tr>

          <td>Value A</td>

          <td><input type="text" id="tfa" /></td>

        </tr>

        <tr>

          <td>Value B</td>

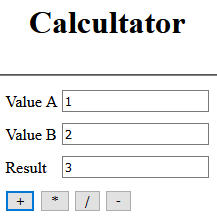
          <td><input type="text" id="tfb" /></td>

        </tr>

        <tr>

          <td>Result</td>

          <td><input type="text" id="tfr" /></td>

        </tr>

        <tr>

          <td colspan="2">

            <button id="bt1">+</button>

            <button id="bt2">\*</button>

            <button id="bt3">/</button>

            <button id="bt4">-</button>

          </td>

        </tr>

      </table>

  </body>

</html>

**Example 3: mouseover, mouseout, css(“color”, “red”)**

<html>

  <script src="../js/jquery-3.5.1.min.js"></script>

  <script>

    $(function () {

      $("h1").mouseover(function () {

        $(this).css("color", "red");

      });

      $("h1").mouseout(function () {

        $(this).css("color", "black");

      });

    });

  </script>

  <body>

    <h1>Welcome to see you!</h1>

  </body>

</html>

**Example 4: mouseover, mouseout, atrr(“src”, “"../images/a.jpeg”)**

<html>

  <head>

    <style>

      img {

        width: 200px;

        height: 200px;

        border-radius: 50%;

      }

    </style>

    <script src="../js/jquery-3.5.1.min.js"></script>

    <script>

      $(function () {

        $("img").mouseover(function () {

          $(this).attr("src", "../images/b.jpeg");

        });

        $("img").mouseout(function () {

          $("img").attr("src", "../images/a.jpeg");

        });

      });

    </script>

  </head>

  <body>

    <img src="../images/a.jpeg" />

  </body>

</html>

**Example 5: (Missing myscript.js)**

<html>

  <head>

    <style>

      img {

        width: 150px;

        height: 150px;

        border-radius: 50%;

      }

    </style>

    <script src="../js/jquery-3.5.1.min.js"></script>

    <script src="../js/myscript.js"></script>

    <script>

      $(function () {

        $("img").mouseover(function () {

          $(this).attr("src", "../images/" + pics[getRandom(0, 9)]);

        });

      });

    </script>

  </head>

  <body>

    <img src="../images/a.jpeg" />

  </body>

</html>

**Example 6: mouseover, mouseout, addClass(...), removeClass(…)**

<html>

  <head>

    <style>

      .model1 {

        width: 400px;

        height: 400px;

        border-radius: 50%;

        background-color: teal;

      }

    </style>

    <script src="../js/jquery-3.5.1.min.js"></script>

    <script>

      $(document).ready(function () {

        $("h1").mouseover(function () {

          $(this).addClass("model1");

        });

        $("h1").mouseout(function () {

          $(this).removeClass("model1");

        });

      });

    </script>

  </head>

  <body>

    <h1>Welcome to see you !</h1>

  </body>

</html>

**Example 7:**

<html>

  <head>

    <style>

      .box {

        width: 25%;

        height: 50px;

        border: 4px groove red;

        background-color: teal;

      }

    </style>

    <script src="../js/jquery-3.5.1.min.js"></script>

    <script>

      $(function () {

        $("h1").mouseover(function () {

          $(this).addClass("box");

        });

        $("h1").mouseout(function () {

          $(this).removeClass("box");

        });

      });

    </script>

  </head>

  <body>

    <h1>Welcome to see you !</h1>

  </body>

</html>

**Example 8:**

<html>

  <style>

    div {

      width: 300px;

      height: 300px;

      border: 5px groove red;

      background-color: teal;

    }

  </style>

  <script src="../js/jquery-3.5.1.min.js"></script>

  <script>

    $(function () {

      $(".btHide").click(function () {

        $("div").hide(500);

        // $("div").css("display", "none");

      });

      $(".btShow").click(function () {

        $("div").show(500);

        // $("div").css("display", "block");

      });

      $("#bt1").click(function () {

        $("div").slideUp(500);

      });

      $("#bt2").click(function () {

        $("div").slideDown(500);

      });

      $("#bt3").click(function () {

        $("div").slideToggle(1000);

      });

    });

  </script>

  <head> </head>

  <body>

    <button class="btHide">Hide</button>

    <button class="btShow">Show</button>

    <button id="bt1">slideUp</button>

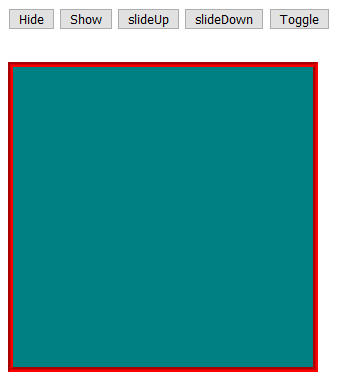
    <button id="bt2">slideDown</button>

    <button id="bt3">Toggle</button>

    <br /><br />

    <div></div>

  </body>

****</html>

១. ផ្នែកសំណួរ

ក. + HTML មានប្រយោជន៍ដូចជា ៖

- ប្រើសម្រាប់បង្កើត Content នៃ Web pages

- ប្រាប់ Browser ពីរបៀបត្រូវបង្ហាញ Content

+ សមាសធាតុសំខាន់ៗ ៖

- head : ផ្ទុក metadata រួមមាន document title, style និង script ជាដើម

- body : ជាកន្លែងផ្ទុក Content ទាំងអស់នៃ HTML document

- footer : ផ្ទុក Footer នៃ HTML document ដូចជា author info, link ជាដើម

- tag: ជាឈ្មោះកំនត់រួចជាស្រេច សំរាប់ធ្វើអ្វីមួយ

<tagname>content</tagname>

<tagname />

- attribute: ជាឈ្មោះកំនត់រួចជាស្រេច សំរាប់បំពេញបន្ថែមអោយ Tag

<tagname attributeName=”value”>content</tagname>

<tagname attributeName=”value” />

- entity: ជាឈ្មោះកំនត់រួចជាស្រេច តាងអោយសញ្ញាពិសេសអ្វីមួយ

&entitiname;

+ ទម្រង់ទូទៅ ៖

<html>

<head></head>

<body></body>

<footer></footer>

<html>

ខ. + CSS មានប្រយោជន៍ដូចជា ៖

- កំណត់ Layout នៃ Web pages

- ដាក់ Style ឱយ Web page មានសោភ័ណភាព

+ គេអាចធ្វើវាបាន 2 របៀប ៖

- Internal មាន 2 បែបទៀត ៖

- Inline:

<tagname style=”property:value;…”> Content</tagname>

- Embedded:

<style>

Declarations Style

</style>

- External:

<link rel= “stylesheet” href= “url” />

គ. + JS មានប្រយោជន៍ដូចជា ៖

- ផ្លាស់ប្ដូរ Content ឬ Behavior នៃ Web pages

- សម្រាប់ embed ជាមួយ HTML ដើម្បីបង្កើត Web dynamic ផ្នែកclient site

+ គេអាចធ្វើវាបាន 2 របៀប ៖

- Internal

<script>

statements;

</script>

- External

<script src = “URL”></script>

ឃ. + JQuery មានប្រយោជន៍ដូចជា ៖

- ធ្វើឱយមានភាពងាយស្រួល ក្នុងការ Develop កូដ JavaScript

- សម្រាប់ ធ្វើអោយ Web Page Dynamic ផ្នែក client side

+ ភាពខុសគ្នា រវាង JS និង JQuery ៖

JS

- មិនមាន Library ដើម្បិងាយ Develop

- ប្រើ កូដ វែង

- ពិបាក ក្នុងការ Develop បើប្រៀបនឹង JQuery

JQuery

- ជា Libray ឬ Framework របស់ JS

- ប្រើ កូដ ខ្លីជាង

- ងាយស្រួល ក្នុងការ Develop កូដ JS

ង. + Static Web page គឺ Read Only មិនអាចផ្លាស់ប្ដូរ Content នៃ Web page បាន។

+ Dynamic Web page គឺ (user input / output ) អ្នកប្រើប្រាស់អាចផ្លាស់ប្ដូ Content របស់ Web Page បាន។

២. ផ្នែកលំហាត់

ក. សរសេរ HTML Tag ដើម្បីប្រើ Style mode1, mode2

<div id="mode1"></div>

<div class="mode2"></div>

ខ. Code ការប្រើប្រាស់ slideup, slidedown, slideToggle

<html>

  <style>

    div {

      width: 300px;

      height: 300px;

      border: 5px groove red;

      background-color: teal;

    }

  </style>

  <script src="../js/jquery-3.5.1.min.js"></script>

  <script>

    $(function () {

      $("#bt1").click(function () {

        $("div").slideUp();

      });

      $("#bt2").click(function () {

        $("div").slideDown();

      });

      $("#bt3").click(function () {

        $("div").slideToggle();

      });

    });

  </script>

  <body>

    <button id="bt1">slideUp</button>

    <button id="bt2">slideDown</button>

    <button id="bt3">slideToggle</button>

    <div></div>

  </body>

</html>