

Lab 3 - HTML, CSS, Bootstrap, JQuery

HTML

ให้ศึกษาพื้นฐาน html จาก <https://www.w3schools.com/html/default.asp> พร้อมตอบคำถามต่อไปนี้

3.1 จงทำหน้า web page แสดง profile ของตนเอง (ไม่มีการใช้ CSS) โดยมีข้อกำหนดดังนี้

1. ต้องมี title อธิบายหน้า web และ heading อย่างน้อย 2 ระดับ
2. ต้องมีรูปภาพ ที่สามารถ กด แล้ว เชื่อมโยงไปยัง google.com
3. ต้องมีตาราง ที่อยู่กึ่งกลางหน้า
4. ต้องมี bullet แสดงหัวข้อต่าง ๆ แบบตัวเลข หรือ ไม่ใช่ตัวเลข

* หน้า web ในข้อนี้จะต้องมีโครงสร้างของ html ที่ดี ตามโครงสร้างของ HTML5 รวมถึงพวก semantic element และ style (syntax) ที่ถูกต้อง ตาม guide line

ตอบ

3.2 จงอธิบายความแตกต่างระหว่าง <div> และ

ตอบ

Cascade Style Sheet (CSS)

CSS basic

การกำหนด css ให้กับ DOM ของ html มีด้วยกัน 3 แบบ คือ Element selector, Id selector และ class selector ดังนี้

Element selector:

```
p {  
  text-align: center;  
  color: red;  
}
```

Id selector:

```
#para1 {  
  text-align: center;  
  color: red;  
}
```

Class selector

```
.center {  
  text-align: center;  
  color: red;  
}
```

Grouping

```
h1, h2, p {  
  text-align: center;  
  /* This is a single-line comment */  
  color: red;  
}
```

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

การกำหนด CSS ให้กับ เอกสาร ทำได้ 3 รูปแบบ คือ Inline, Internal และ External ตามลำดับ

External:

```
<head>  
<link rel="stylesheet" type="text/css" href="mystyle.css">  
</head>
```

Internal:

```
<style>  
body {  
  background-color: linen;  
}
```

```
h1 {
```

```
color: maroon;
margin-left: 40px;
}
</style>

Inline:
<h1 style="color:blue;margin-left:30px;">This is a heading.</h1>
```

Priority:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section) (use the last defined)
3. Browser default

Example: (h1 will be orange)

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
h1 {
  color: orange;
}
</style>
</head>
```

```
p {
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
}
```

```
p {
  padding-top: 50px;
  padding-right: 30px;
  padding-bottom: 50px;
  padding-left: 80px;
}
```

```
p {
  padding: 50px 30px 50px 80px;
}
```

```
div {
  width: 500px;
  height: 100px;
  border: 3px solid #73AD21;
}
```

```
ul {
  background: #3399ff;
  padding: 20px;
}
```

```
ul li {
  background: #cce5ff;
  margin: 5px;
}
```

Less

เนื่องจากการเขียน CSS ในรุ่นใหม่ ๆ มีความซับซ้อนขึ้นเรื่อย ๆ จึงมีการพัฒนาโปรแกรม เพื่อใช้ในการสร้าง CSS ออกมาได้ แต่ Less, Sass ในตัวอย่างนี้ จะมีแนะนำการใช้ Less เบื้องต้น

ติดตั้ง: `npm install less -g`

Location after install: `C:\Users\comeng\AppData\Roaming\npm\lessc.cmd`

index.html

```
<html>
  <head>
    <title></title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <p> Test &lt;p>></p>
    <ul><li> list1 </li> <li> list2 </li> </ul>
  </body>
</html>
```

style.less

```
@background-color: grey;
@text-color: #ddf;

p{
  background-color: @background-color;
  color: @text-color;
  padding: 15px;
}

ul{ background-color: @background-color; }
li{ color: @text-color; }
```

มีการกำหนดสี โดยใช้ตัวแปร (ขึ้นด้วย @)

Lessc style.less > style.css

Output:

style.css

```
p{
  background-color: #ffffff;
  color: #1A237E;
  padding: 15px;
}
```

```

}
ul{
    background-color: #ffffff;
}
li{
    color: #1A237E;
}

```

Mixins (reusable)

สังเกตว่ามีการกำหนด #circle แล้ว สามารถนำมาใช้ใหม่ได้

```

#circle{
    background-color: #4CAF50;
    border-radius: 100%;
}

#small-circle{
    width: 50px;
    height: 50px;
    #circle
}

#big-circle{
    width: 100px;
    height: 100px;
    #circle
}

```

index.html

```

<div id="small-circle" > small circle </div> <br/>
<div id="big-circle" style="line-height: 90px;"> big circle </div><br/>
<div id="circle"> circle </div><br/>

```

Mixins : argument (default value of 50 pixels. This will create a small circle 50×50 and a big one 100×100 pixels)

```

#circle(@size: 50px){
    background-color: #4CAF50;
    border-radius: 100%;

    width: @size;
    height: @size;
}

```

```
#small-circle{
  #circle
}

#big-circle{
  #circle(100px)
}
```

Output: (no #circle anymore)

```
#small-circle {
  background-color: #4CAF50;
  border-radius: 100%;
  width: 50px;
  height: 50px;
}
#big-circle {
  background-color: #4CAF50;
  border-radius: 100%;
  width: 100px;
  height: 100px;
}
```

Reference: <http://tutorialzine.com/2015/07/learn-less-in-10-minutes-or-less/>

3.3 จาก web ในข้อ 3.1 จงใช้ CSS จัดรูปแบบให้สวยงาม

ตอบ

CSS Framework: Bootstrap

ให้ศึกษาการใช้ bootstrap จาก <https://getbootstrap.com/>

3.4 จาก web ในข้อ 3.1 จงใช้ bootstrap จัดรูปแบบให้สวยงาม

ตอบ

Traditional Javascript

Javascript Intro

การใช้ javascript แบบเก่า เพื่อควบคุม พฤติกรรมต่าง ๆ ผ่าน web browser

```
<html>
  <head><script> alert('Hello'); </script> </head>
  <body>
  </body>
</html>
```

หมายเหตุ: Case sensitive alert(); != Alert()
+ - */ var... (similar as nodejs since it's javascript)

Function

```
<script>
  for(var i=1;i<2;i++)
    alert('Hello' + ' word' + i);
  function foo() {
    alert('foo');
  }
  foo();
</script>
```

Loop และ object

```
<script>
  y = 8; // global variable
  console.log('y =' + y)
  var myarray = [5, "foo", 'bar'];
  for(var i in myarray)
    console.log(myarray[i]);

  var myObj = { "name": "John",
    "surname": "Denver",
    "age": 22,
    "Link": { "web1": "http://web1.com", "web2": "http://web2.com" },
    "ma" : [ 'red', 'green', 'blue' ]
  };
  console.log(myObj.ma[1]);
</script>
```

ตัวอย่างการเข้าถึง DOM element

เข้าถึง DOM element โดย Id หรือ Tag จาก document object จากนั้น สามารถปรับเปลี่ยนผ่าน innerHTML

```
<body>
<p id="myId1" align="center"> Hello id1 </p>
<p id="myId2"> Hello id2 </p>

<ul id="myul">
  <li><a href="#"><b>Menu1</b></a></li>
  <li><a href="#">Menu2</a></li>
</ul>

<script>
  var p1 = document.getElementById("myId1");
  console.log(p1);
  console.log(p1.getAttribute("align"));
  p1.setAttribute("align", "right");
  console.log(p1.getAttribute("align"));

  var li = document.getElementsByTagName("li");
  console.log("li.length:" + li.length);
  console.log("li[0].innerHTML:" + li[0].innerHTML);
  console.log("li[1].innerHTML:" + li[1].innerHTML);

  li[0].innerHTML = "New Menu1";

  var ul = document.getElementById("myul");
  var newLi = document.createElement("li");
  ul.appendChild(newLi);
  newLi.innerHTML = 'Menu3';
</script>
</body>
```


Javascript event

ใช้ onclick() event ในการเปลี่ยนค่าใน html element เช่น <p> ผ่าน innerHTML

```
<body>
<p id="id1"> Hello world</p>
<p id="id2"> My P ID2 </p>
<p id="id3"> My P ID3 </p>

<button onclick="alert('Hello');"> Alert </button>
<button onclick="changeP();" > Change P </button>

<script>
  var p1 = document.getElementById("id1");
  p1.onclick = function () {
    p1.innerHTML = "Foo bar";
  };

  function changeP() {
    p1.innerHTML = "Change p1";
  };

  window.onload = function start() {
    setTimeout( function () {
      p1.innerHTML = "Start";
    }, 2000);
  };

  document.getElementById("id3").style.borderStyle = "dashed";
</script>
</body>
```

jQuery

เนื่องจาก Javascript มีความซับซ้อน และไม่เป็นโครงสร้าง จึงมีการพัฒนา Library ที่ชื่อว่า jQuery เพื่อให้การเขียน javascript สั้นลง (แต่ก็ยังไม่เป็นโครงสร้างแบบ OOP) การใช้ jQuery เพียงแค่ load script jquery.min.js มา ก็สามารถใช้งานได้เลย

- จาก document.getElementById('id') สามารถอ้างอิงสั้น ๆ เป็น jQuery('#id') และ ย่อให้สั้นกว่าเดิมเป็น \$('#id')
- innerHTML ก็เปลี่ยนเป็น text() ทำให้ดูง่ายมากขึ้น (ถ้าเข้าใจความหมายที่ย่อ)

```
<head><script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script></h
ead>
<body>
<p id="id1"> Hello world</p>
<p id="id2"> My P ID2 </p>
<p id="id3"> My P ID3 </p>

<button onclick="alert('Hello');"> Alert </button>
<button onclick="changeP();" > Change P </button>

<script>
    jQuery("#id2").click( function () {
        jQuery("#id2").text("Foo bar");
    });

    $("#id3").click( function () {
        $(this).text("Short form");
    });

    $("#id3").css("border-style","dashed");

    $("#id3").click(function() {
        $(this).fadeOut(3000);
        $(this).fadeIn(3000);
    });
</script>
```

window.onload

สามารถใช้ window.onload เพื่อเรียกให้ ฟังก์ชันทำงานตั้งแต่แรก แต่หากเราใส่ onload ไว้ก่อนที่ html element จะถูก render มา จะทำให้ คำสั่งนั้นไม่สามารถทำงานได้ ดังนี้

```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script>
    window.onload = function() {
        $("#id1").text("successfully Load Id1");
    }
</script>
```

```

    $("#id2").text("successfully Load Id2"); // Not execute since DOM is not
    ready yet (under head, before <p> not work too)
    $("#id3").text("successfully Load Id3");
    $("#div").text("New DIV area");
</script>

</head>
<body>
<p id="id1"> Hello world</p>
<p id="id2"> Hello world</p>
<p id="id3"> Hello world</p>
<div> DIV Area </div>
<div id="d1"> DIV Area </div>

</body>
</html>

```

การแก้ปัญหาคือ ให้นำ script ไปไว้ส่วนท้ายของ html เพื่อให้ html element ถูก load เสร็จเรียบร้อยก่อน จึงจะใช้ jQuery จัดการได้ ดังนี้

```

<html>
<head>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
</head>
<body>
<p id="id1"> Hello world</p>
<p id="id2"> Hello world</p>
<p id="id3"> Hello world</p>
<div> DIV Area </div>
<div id="d1"> DIV Area </div>

<script>
  window.onload = function() {
    $("#id1").text("successfully Load Id1");
  }

  $("#id2").text("successfully Load Id2");
  $("#id3").text("successfully Load Id3");
  $("#div").text("New DIV area");
</script>
</body>
</html>

```

ready()

วิธีที่ดีกว่า คือ ใช้ฟังก์ชัน *ready* ของ jQuery เพื่อรอให้ html element ถูก load มาเรียบร้อยแล้ว จึงจะใช้คำสั่งในการจัดการ

```

<html>
<head>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
  <script>
    window.onload = function() {
      $("#id1").text("successfully Load Id1");
    }

    $(document).ready(function () {
      $("#id2").text("successfully Load Id2");
    });

    $(function () {
      $("#id3").text("successfully Load Id3");
    });

    $(function () {
      $("#div").click( function () {
        $(this).text("New DIV area");
      });
    });

    $(function () {
      $("#d1").click( function () {
        $(this).html("<h1>New HTML DIV area<h1>");
      });
    });
  </script>
</head>
<body>
<p id="id1"> Hello world</p>
<p id="id2"> Hello world</p>
<p id="id3"> Hello world</p>
<div> DIV Area </div>
<div id="d1"> DIV Area </div>
</body>
</html>

```

ฟังก์ชัน (แสดง ซ่อน สลับ)

```

<html>
<head>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
  <script>
    $(function () {
      $("#b1").click( function () {
        $("#id3").hide(2000);
        $("#id3").show('normal'); // slow, normal, fast
      });
    });

    $(function () {
      $("#b2").click( function () {
        $("#id4").toggle('slow', function() {

```

```

        var test = $(this).css('display');
        if (test == 'none')
            alert('hide');
        else
            alert('show')
    })
});

</script>
</head>
<body>
<p id="id1"> Hello world</p>
<p id="id2"> Hello world</p>
<p id="id3"> Hello world</p>
<p id="id4"> Hello world</p>
<div> DIV Area </div>
<div id="d1"> DIV Area </div>
<button id="b1"> Show/Hide </button>
<button id="b2"> Toggle </button>
</body>
</html>

```

3.5 จาก web ในข้อ 3.4 จงใช้ jQuery เพิ่มเข้าไป เพื่อให้ web มีการตอบสนองต่อเหตุการณ์ และเขียนฟังก์ชันใหม่ ในรูปแบบของ arrow function

ตอบ