

Homework(Easy-level)

1. **HTML** ทำหน้าที่ในการจัดการโครงสร้างและรูปร่างของเว็บไซต์ จะประกอบไปด้วย tag ต่าง ๆ

```
<html>

  <head>

    <title></title>

  </head>

  <body>

    <div>

      <p></p>

    </div>

  </body>

</html>
```

CSS ทำหน้าที่ตกแต่งหน้าเว็บไซต์ให้สวยงาม

```
h1 {

  display: block ;

  font-size: 2em ;

  margin-top: 0.67em ;

  margin-bottom: 0.67em ;

  margin-left: 0 ;

  margin-right: 0 ;

  color:#FFF ;

}
```

JavaScript ทำหน้าที่เพิ่มความสามารถหรือเพิ่มคุณสมบัติพิเศษให้กับเว็บไซต์ ทำให้เว็บไซต์สามารถ interactive กับผู้ใช้งาน หรือ user ได้ดีขึ้น

```
<script>

var cars = ["BMW", "Volvo", "Saab", "Ford", "Fiat", "Audi"];

var text = "";

var i;

for (i = 0; i < cars.length; i++) {

    text += cars[i] + "<br>";

}

document.getElementById("demo").innerHTML = text;

</script>
```

2. 3 วิธี **Inline** <button onclick="document.getElementById('myImage').src='pic_bulbon.gif'">

```
Internal function onLight(){

    document.getElementById('myImage').src='pic_bulbon.gif' ;}
```

```
External <head>    <script src='myScript.js'></script>    </head>
```

External file: myScript.js

```
<script>

function onLight(){

    document.getElementById('myImage').src='pic_bulbon.gif' ;}

}</script>
```

3. `<script> </script>`

4.

```
<body>
  <h2>What Can JavaScript Do?</h2>
  <p id="demo">JavaScript can change HTML content</p>
  <button type="button"
    onclick="document.getElementById('demo').innerText='Hello JavaScript!'">
    Click Me!</button>
</body>
```

5. `log` ค่า 1 และ -1 (F12→console)

6. ไม่ต้องกำหนด `type` ในการประกาศตัวแปร

7. ตัวแปร `Arrays` คือ ชุดของข้อมูลที่บรรจุค่าข้อมูลชนิดเดียวกันเอาไว้ ในลักษณะของแถวจำนวนแบบคงที่

ตัวแปร `ออบเจ็กต์ (object)` คือสิ่งที่ถูกสร้างขึ้นมาจากคลาส ก่อนที่จะสร้างออบเจ็กต์ได้ ต้องสร้างคลาสขึ้นมาก่อน เมื่อคลาสแปลงร่างกลายเป็นออบเจ็กต์แล้ว จะกำหนดค่าข้อมูลได้หรือใช้งานความสามารถต่าง ๆ

8.

```
<body>
  <p id="test">
    <script>
      function add(a,b){
        return a + b ;
      }
      document.getElementById("test").innerHTML = add(20,10);
    </script>
  </p>
</body>
```

9.

```
var arr = [1,2,3,4];
function sum(arr){
    var sum = 0 ;
    for (let i = 0; i < arr.length; i++) {
        sum =sum + arr[i];
    }
    return sum ;
}
document.getElementById("test").innerHTML = sum(arr);
```

10. รูปแบบอย่างหนึ่งที่ใช้ในการเก็บและการส่งข้อมูลโดยจะมีการเก็บข้อมูลในลักษณะของ Array of Object ที่ถูกแปลงเป็นข้อความธรรมดาแล้วใช้ส่งข้อมูล

```
"employees":[
  {"firstName":"John", "lastName":"Doe"},
  {"firstName":"Anna", "lastName":"Smith"},
  {"firstName":"Peter", "lastName":"Jones"}
]
```

Homework(Mid-level)

11.

```
<body>
  <p id="test">
    <script>
      var people = {
        title: 'Mr',
        firstname: 'Sawasdee',
        lastname: 'Klambu'
      }
      function getFullname(people){
        return people.title+' '+people.firstname+' '+people.lastname;
      }
      document.getElementById("test").innerHTML = getFullname(people);
    </script>
  </body>
```

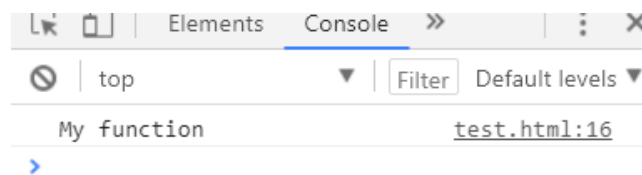
12. เครื่องหมาย == หรือ != จะมีการเปรียบเทียบให้ตรงกันทั้งชนิดของข้อมูล (เช่นต้องเป็นตัวเลขเหมือนกัน) และค่าของมันด้วย โดยจะมีการเปรียบเทียบชนิดของตัวแปรก่อน ถ้าตัวแปรทั้งสองตัวเป็นชนิดเดียวกัน ถึงจะมีการเปรียบเทียบค่าของมัน

ส่วนแบบ == หรือ != จะมีการเปรียบเทียบเฉพาะค่าของมันเท่านั้น โดยจะมีการตรวจสอบชนิดของตัวแปรก่อน ถ้าชนิดของตัวแปรไม่ตรงกัน javascript จะทำการแปลงตัวแปรทางด้านขวามือ ให้เหมือนกับตัวแปรทางด้านซ้ายมือ แล้วถึงนำไปเปรียบเทียบกัน

13.

```
<body>
  <h2>JavaScript</h2>
  <p id="demo"></p>
  <script>
    var s = "hello";
    function reverse(s){
      return s.split("").reverse().join("");
    }
    document.getElementById("demo").innerHTML = reverse(s);
  </script>
  ...
</body>
```

14. ไม่ error เพราะ console.log เป็นคำสั่งที่เมื่อเขียนฟังก์ชันแล้วไม่ต้องการให้ User เห็นไม่ต้องการใช้ Alert



15. ไม่ error เหมือนข้อ 14 แต่เป็นการเรียกใช้ function ซ้ำๆ

