|  |  |  |
| --- | --- | --- |
| **Mobile Programming Laboratory**  **Week 2: JavaScript** |  | |
| Name:Atthapong Chooduang | ID:6231302022 | Section:1 |
| Date: | Due: | |

**Objective**

* To practice JavaScript

**References**

* <https://www.w3schools.com/js/default.asp>
* <https://developer.mozilla.org/en-US/docs/Learn/JavaScript>
* <https://github.com/davidshimjs/qrcodejs>

**Internal JavaScript**

We usually place JavaScript <script></script> at the bottom of the <body> to wait for all loading HTML tags to be complete.

Otherwise if we prefer placing JavaScript between <head></head> we can use an event “DOMContentLoaded”.

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Document</title>  <script>  document.addEventListener("DOMContentLoaded", function() {  document.querySelector("p").innerHTML = "Hi";  });  </script>  </head>  <body>  <p>Hello</p>  </body>  </html> |

The possible drawback of this method is that if our website has a large number of JavaScript codes and files, loading the page will be slow.

To solve such problem, we can extract our scripts to external files and load them concurrently with HTML.

**External JavaScript**

Assume we create two files: **index.html** and **hello.js** in the same folder.

**index.html**

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Document</title>  <script src="hello.js" async></script>  </head>  <body>  <p>Hello</p>  </body>  </html> |

**hello.js**

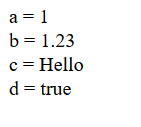
|  |
| --- |
| document.querySelector("p").innerHTML = "Hi"; |

Note that

* No need for loading event but we must always provide a keyword “async” or “defer” otherwise it will be the same as internal JS
* “async” means JavaScript file will be loading concurrently with HTML files. JS will act as soon as the both loading are complete.
* However, if you have many JS to load and they must be loaded in order, use “defer” instead.

**Data Types**

* Number, String, Boolean, Object



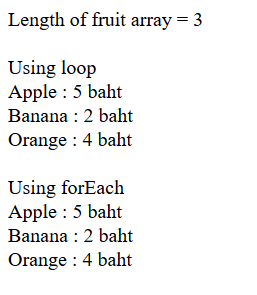
|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>JavaScript Example</title>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  </head>  <body>  <script>  var a = 1;  var b = 1.23;  var c = "Hello";  var d = true;  document.write("a = "+ a + "<br>");  document.write("b = "+ b + "<br>");  document.write("c = "+ c + "<br>");  document.write("d = "+ d + "<br>");  </script>  </body>  </html> |

Note that for ECMAScript6 or ES6 we can use “let” to declare local variables and “const” for constant declaration.

For ES5, we can add “use strict;” at the top of script to force the variable declaration.

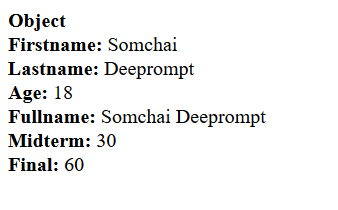
|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>JavaScript Example</title>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  </head>  <body>  <script>  "use strict";  a = 1; //fail  document.write(a); //no output  </script>  </body>  </html> |

**Array**



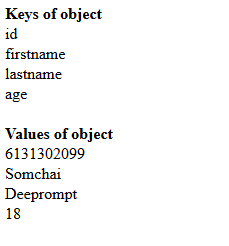
|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>JavaScript Example</title>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  </head>  <body>  <script>  var fruit = ["Apple", "Banana", "Orange"];  var price = [5, 2, 4];  document.write("Length of fruit array = " + fruit.length + "<br><br>");  document.write("Using loop <br>");  for(let i=0; i<fruit.length; i++) {  document.write(fruit[i] + " : " + price[i] + " baht<br>");  }  document.write("<br>");  //for ES5  document.write("Using forEach <br>");  //call a function for each array member, the function can have up to three parameters  fruit.forEach(function(value, index) {  document.write(value + " : " + price[index] + " baht<br>");  });  </script>  </body>  </html> |

**Object**



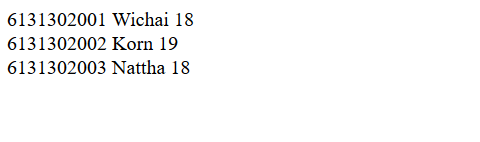
|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>JavaScript Example</title>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  </head>  <body>  <script>  let person = {  firstname: "Somchai",  lastname: "Deeprompt",  age: 18,  fullname: function() {  return this.firstname + " " + this.lastname;  },  score: {midterm: 30, final:60}  };  document.write("<b>Object</b> <br>");  document.write("<b>Firstname:</b> " + person.firstname + "<br>");  document.write("<b>Lastname:</b> " + person['lastname'] + "<br>");  document.write("<b>Age:</b> " + person.age + "<br>");  document.write("<b>Fullname:</b> " + person.fullname() + "<br>");  document.write("<b>Midterm:</b> " + person.score.midterm + "<br>");  document.write("<b>Final:</b> " + person.score.final + "<br>");  </script>  </body>  </html> |

We can also iterate through object’s keys using “for…in”.



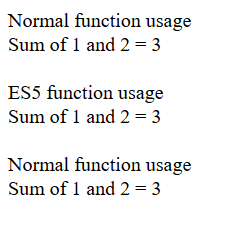
|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>JavaScript Example</title>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  </head>  <body>    <script>  let person = {  id: "6131302099",  firstname: "Somchai",  lastname: "Deeprompt",  age: 18  };  document.write("<b>Keys of object</b><br>")  for(key in person) {  document.write(key + "<br>");  }  document.write("<br>");  document.write("<b>Values of object</b><br>")  for(key in person) {  document.write(person[key] + "<br>");  }  </script>  </body>  </html> |

**Array of Objects**



|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>JavaScript Example</title>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  </head>  <body>  <script>  let person = [  { id: "6131302001", name: "Wichai", age: 18 },  { id: "6131302002", name: "Korn", age: 19 },  { id: "6131302003", name: "Nattha", age: 18 }  ];  //normal display  let result = "";  for (let i = 0; i < person.length; i++) {  result = result + person[i].id + " " + person[i].name + " " + person[i].age + "<br>";  }  document.write(result);  </script>  </body>  </html> |

**Function**



|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>JavaScript Example</title>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  </head>  <body>  <script>  //-------------- function -----------------  function sum(x, y) {  return x + y;  }  //ES5 function  let sum1 = function (x, y) {  return x + y;  }  //ES6 arrow function  let sum2 = (x, y) => x + y;  //--------------------------------------------  let a = 1, b = 2;  let result = sum(a, b);  document.write("Normal function usage<br>")  document.write("Sum of " + a + " and " + b + " = " + result + "<br><br>");  result = sum1(a, b);  document.write("ES5 function usage<br>")  document.write("Sum of " + a + " and " + b + " = " + result + "<br><br>");  result = sum2(a, b);  document.write("Normal function usage<br>")  document.write("Sum of " + a + " and " + b + " = " + result + "<br><br>");  </script>  </body>  </html> |

**DOM (Document Object Model)**

Components of a HTML file can be described as a Tree which is called Document Object Model (DOM).

Observe a basic HTML structure below.

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>JS Demo</title>  </head>  <body>      <p>Hello</p>      <input type="button" value="OK">  </body>  </html> |

Document

<head>

<body>

<title>

“JS Demo”

<p>

“Hello”

<input>

Attribute  
‘type’

Attribute  
‘value’

“button”

“OK”

**Finding HTML Elements**

|  |  |
| --- | --- |
| An element by id | //select an element whose id is "intro"  var x = document.getElementById("intro"); |
| All elements by tag name | //select all elements having tag <p>  var x = document.getElementsByTagName("p"); |
| All elements by class name | //select all elements having class "intro"  var x = document.getElementsByClassName("intro"); |
| By a way similar to CSS selectors | //select the first <p> having class "intro"  var x = document.querySelector("p.intro");  //select all <p> having class "intro"  var x = document.querySelectorAll("p.intro");  So “#intro” or “a[target]” or “div > p” are all valid |

**Changing HTML Content**

Use the innerHTML property

|  |
| --- |
| document.getElementById(id).innerHTML = "new HTML"  or  document.querySelector(id).innerHTML = "new HTML" |

**Changing the Value of an Attribute**

Use the attribute property

|  |
| --- |
| document.getElementById(id).attribute = "new value"  or  document.querySelector(id).setAttribute("new value"); |

**Changing HTML Style**

Use the style.property

|  |
| --- |
| document.getElementById(id).style.property = "new style"  or  document.querySelector(id).style.property = "new style" |

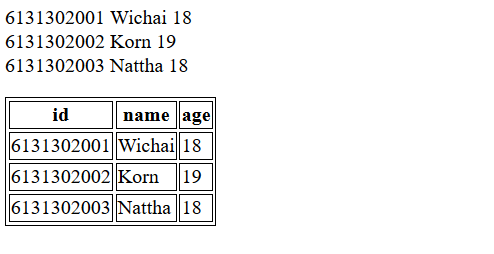
**Click event**

At least three possible choices to bind a click event to a DOM element like a button.

1. Use html property “onclick”
2. Use JS “.onclick”
3. Use JS “addEventListener()”

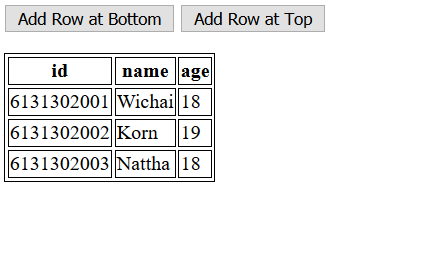
|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Click event</title>  </head>  <body>  <!-- first method -->  <p id="one" onclick="hello()">One</p>  <p id="two">Two</p>  <p id="three">Three</p>  <script>  function hello() {  alert("Hello");  }  //second method  document.getElementById("two").onclick = hello;  //third method  document.getElementById("three").addEventListener("click", hello);  </script>  </body>  </html> |

**Exercise:** Display JS array of object in a table

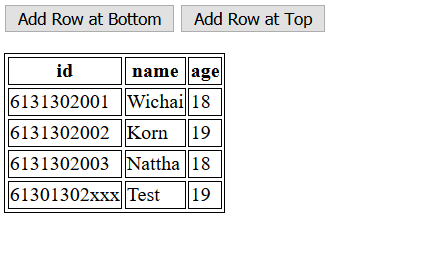


|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>JavaScript Example</title>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <style>  table, th, td {  border: 1px solid black;  }  </style>  </head>  <body>  <p id="pDisplay"></p>  <table id="tDisplay"></table>  <script>  let person = [  { id: "6131302001", name: "Wichai", age: 18 },  { id: "6131302002", name: "Korn", age: 19 },  { id: "6131302003", name: "Nattha", age: 18 }  ];  //normal display  let result = "";  for (let i = 0; i < person.length; i++) {  result = result + person[i].id + " " + person[i].name + " " + person[i].age + "<br>";  }  document.getElementById("pDisplay").innerHTML = result;  //table display  result = "<tr><th>id</th> <th>name</th> <th>age</th></tr>";  for (let i = 0; i < person.length; i++) {  result = result + "<tr><td>" + person[i].id + "</td><td>" + person[i].name + "</td><td>" + person[i].age + "</td></tr>";  }  document.getElementById("tDisplay").innerHTML = result;  </script>  </body>  </html> |

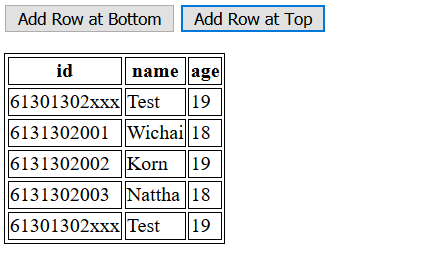
For a HTML table, we have a method to insert or delete row.



Clicking the first button.



Clicking the second button.



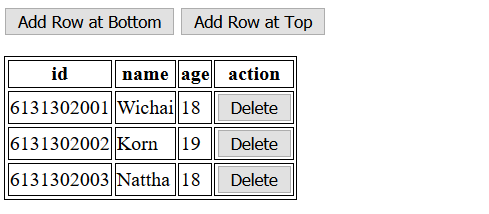
|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>JavaScript Example</title>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <style>  table, th, td {  border: 1px solid black;  }  </style>  </head>  <body>  <button onclick="addBottom()">Add Row at Bottom</button>  <button onclick="addTop()">Add Row at Top</button>  <p id="pDisplay"></p>  <table id="tDisplay"></table>  <script>  let person = [  { id: "6131302001", name: "Wichai", age: 18 },  { id: "6131302002", name: "Korn", age: 19 },  { id: "6131302003", name: "Nattha", age: 18 }  ];  //table display  let result = "<tr><th>id</th> <th>name</th> <th>age</th></tr>";  for (let i = 0; i < person.length; i++) {  result = result + "<tr><td>" + person[i].id + "</td><td>" + person[i].name + "</td><td>" + person[i].age + "</td></tr>";  }  let tDisplay = document.getElementById("tDisplay");  tDisplay.innerHTML = result;  function addBottom() {  let newRow = "<tr><td>61301302xxx</td><td>Test</td><td>19</td></tr>";  //add at the end of the row, use index -1  let row = tDisplay.insertRow(-1);  row.innerHTML = newRow;  }  function addTop() {  let newRow = "<tr><td>61301302xxx</td><td>Test</td><td>19</td></tr>";  //add at the top of the row, head row is index 0  let row = tDisplay.insertRow(1);  row.innerHTML = newRow;  }  </script>  </body>  </html> |

Another approach is to update the data itself and redraw the table.

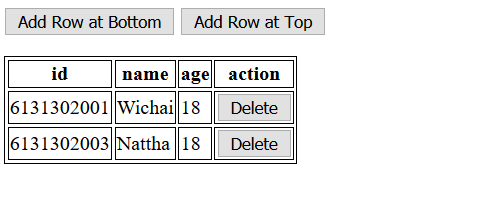
|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>JavaScript Example</title>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <style>  table,  th,  td {  border: 1px solid black;  }  </style>  </head>  <body>  <button onclick="addBottom()">Add Row at Bottom</button>  <button onclick="addTop()">Add Row at Top</button>  <p id="pDisplay"></p>  <table id="tDisplay"></table>  <script>  let person = [  { id: "6131302001", name: "Wichai", age: 18 },  { id: "6131302002", name: "Korn", age: 19 },  { id: "6131302003", name: "Nattha", age: 18 }  ];  let tDisplay = document.getElementById("tDisplay");  //create table and display  createTable();    function createTable() {  let result = "<tr><th>id</th> <th>name</th> <th>age</th></tr>";  for (let i = 0; i < person.length; i++) {  result = result + "<tr><td>" + person[i].id + "</td><td>" + person[i].name + "</td><td>" + person[i].age + "</td></tr>";  }  tDisplay.innerHTML = result;  }  function addBottom() {  //add at the end of the array  person.push({id: "61301302xxx", name: "Test", age:19});  createTable();  }  function addTop() {  //add at the top of the array  person.unshift({id: "61301302xxx", name: "Test", age:19});  createTable();  }  </script>  </body>  </html> |

Using the latest technique has a benefit that we can get the updated index of each row.

**Assignment 1:** Try to delete a selected row.



Try to delete the middle row.



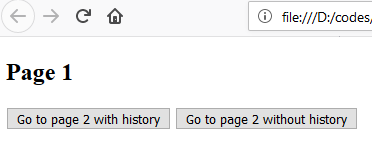
Paste your code here.

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Week2-Assign1</title>  <style>  table,  th,  td {  border: 1px solid black;  }  </style>  </head>  <body>  <button onclick="addBottom()">Add Row at Bottom</button>  <button onclick="addTop()">Add Row at Top</button>  <p id="pDisplay"></p>  <table id="Display"></table>  <script>  let person = [  { id: "6331302001", name: "Piti", age: 19 },  { id: "6331302002", name: "Aden", age: 18 },  { id: "6331302003", name: "Jiraya", age: 17 }  ];  let Display = document.getElementById("Display");  //create table and display  createTable();  function createTable() {  let result = "<tr><th>id</th> <th>name</th> <th>age</th> <th>action</th></tr>";  for (let i = 0; i < person.length; i++) {  result = result += "<tr> <td> "+person[i].id+" </td> <td>"+person[i].name+"</td> <td>"+person[i].age+"</td> <td> <button onclick='delet(this)' >Delete</td></tr>";  }  Display.innerHTML = result;  }  function addBottom() {  //add at the end of the array  person.push({id: "63301302xx", name: "AA", age:19});  createTable();  }  function addTop() {  //add at the top of the array  person.unshift({id: "63301302xx", name: "Test", age:23});  createTable();  }  function delet(r){  var i = r.parentNode.parentNode.rowIndex;  document.getElementById("Display").deleteRow(i);  }  </script>  </body>  </html> |

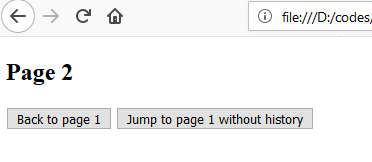
**BOM (Browser Object Model)**

Navigation

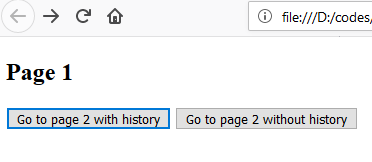
|  |
| --- |
| //jump with history  window.location.href = "./page2.html";  //jump without history  window.location.replace("./page2.html"); |



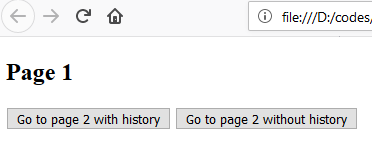
When click the first button, we will jump to page 2. Observe the back button is active now.



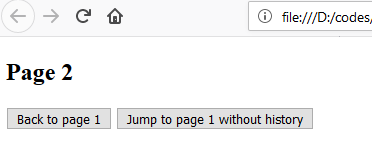
Now try click the first button to go back. Observe that the forward button is currently active.



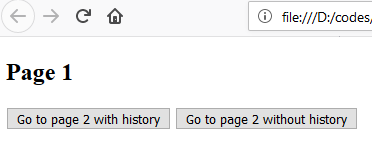
Close this window and relaunch it.



Now click the second button, we will jump to page 2 but the back arrow icon is not useable.



If you click the first button, nothing happens. Try clicking the second one. Observe that both backward and forward buttons are not possible.



page1.html

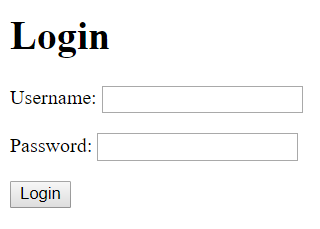
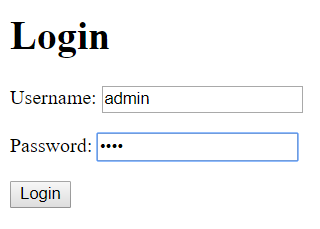
|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>JavaScript Example</title>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  </head>  <body>  <h2>Page 1</h2>  <button onclick="jump1()">Go to page 2 with history</button>  <button onclick="jump2()">Go to page 2 without history</button>  <script>  //jump with history  function jump1() {  window.location.href = "./page2.html";  }  //jump without history  function jump2() {  window.location.replace("./page2.html");  }  </script>  </body>  </html> |

page2.html

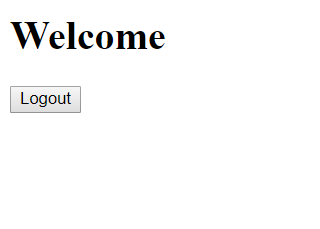
|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>JavaScript Example</title>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  </head>  <body>  <h2>Page 2</h2>  <button onclick="jump1()">Back to page 1</button>  <button onclick="jump2()">Jump to page 1 without history</button>  <script>  //jump backward  function jump1() {  window.history.back();  // window.history.go(-1);  }  //jump without history  function jump2() {  window.location.replace("./page1.html");  }  </script>  </body>  </html> |

**Assignment 2:** Create a sample login page. If a user inputs the correct login (username: admin, password: 1234), the welcome page is shown. Also the user can logout and returns to the login page.

login.html

welcome.html



**Paste your code here.**

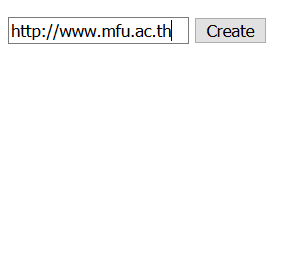
**login.html**

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Login</title>  </head>  <body>  <h1>Login</h1>  <label for="un">Username: </label><input id="txtUsername" type="text"><br><br>  <label for="pw">Password: </label> <input id="txtPassword" type="password"><br><br>  <input id="btnLogin" type="button" value ="Login" >  <script>  const btnLogin = document.querySelector("#btnLogin");  btnLogin.onclick = next;  function next(){  //window.location.href = "./Assignment2\_logout.html";  const txtUsername = document.querySelector("#txtUsername");  const username = txtUsername.value;  const txtPassword =document.querySelector("#txtPassword");  const password = txtPassword.value;    if(username == "admin" && password == "1234"){  window.location.replace ("./Assignment2\_logout.html");  }else if(username == "" && password == ""){  alert("Incomplete Input");  }else{  alert("Login Fail");  }    }  </script>  </body>  </html> |

**welcome.html**

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Welcome</title>  </head>  <body>  <h1>Welcome</h1>  <input id="btnLogout"type="button" value ="Logout">  <script>  const btnLogout = document.querySelector("#btnLogout");  btnLogout.onclick = next;  function next(){  window.location.href = "./Assignment2\_login.html";    }  </script>  </body>  </html> |

**Assignment 3:** Use the following JavaScript library (<https://github.com/davidshimjs/qrcodejs>) to generate QR code from text input.

After clicking a “Create” button,



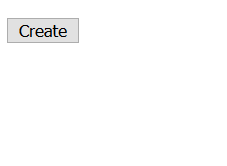
Try to validate your QR code with any mobile app.

Paste your code below.

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Assign3</title>  </head>  <body>  <input id="txtsearch" type="text"><input id="btncreate" type="button" value="CREATE" onclick="ceqr()"><br><br>  <script src="https://cdnjs.cloudflare.com/ajax/libs/qrious/4.0.2/qrious.min.js"></script>  <canvas id="qrcode"></canvas>  <script>    function ceqr() {  var qrce = document.getElementById("txtsearch").value;  var qr;  qr = new QRious({  element: document.getElementById('qrcode'),  size: 200,  });  qr.set({  foreground: 'black',  size: 200,  value: qrce  });  }  </script>  </body>  </html> |

**Assignment 4:** Assume that we have a set of web URLs in the following array. Generate all QR codes from the array’s members after clicking the button. Also show the URL of each QR code.

|  |
| --- |
| const urls = ["http://www.mfu.ac.th", "http://itschool.mfu.ac.th", "http://reg.mfu.ac.th"]; |



After clicking the button,



Paste your code below.

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Assignment4</title>  <script src="./qrcode.min.js" async></script>  </head>  <body>  <input type="button" id="create" value="CREATE">  <table id="table"></table>    <script>  const urls = ["http://www.mfu.ac.th","http://itschool.mfu.ac.th","http://reg.mfu.ac.th"];  const table = document.querySelector("#table");  const btnCreate = document.querySelector("#create");  btnCreate.addEventListener("click",makeQr);  function makeQr(){  let tr ="";  for(i in urls){  tr = `<br><tr id="qr\_${i}"></tr><p>${urls[i]}</p>`  table.innerHTML += tr;  }  for(qrs in urls){  new QRCode(document.querySelector(`#qr\_${qrs}`),urls[qrs]);  }  }  </script>  </body>  </html> |