

LAB 04 – JAVASCRIPT

1/. Javascript Overview 1/.

What is Javascript?

Javascript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

2/. Javascript Syntax

JavaScript can be implemented using JavaScript statements that are placed within the `<script>... </script>` HTML tags in a web page.

```
<script ...>  
    JavaScript code  
</script>
```

- Whitespace and Line Breaks: You can use spaces, tabs, and newlines freely in your program and you are free to format and indent your programs in a neat and consistent way that makes the code easy to read and understand.
- Semicolons are Optional: JavaScript allows you to omit semicolon if each of your statements are placed on a separate line.
- Case Sensitivity: JavaScript is a case-sensitive language.
- Comments in JavaScript: use `//` or `/*...*/`

3/. Javascript Placement

There is a flexibility given to include JavaScript code anywhere in an HTML document. However the most preferred ways to include JavaScript in an HTML file are as follows:

- Script in `<head>...</head>` section.
- Script in `<body>...</body>` section.
- Script in `<body>...</body>` and `<head>...</head>` sections.
- Script in an external file and then include in `<head>...</head>` section.

II/. Javascript DOM

With the HTML DOM, JavaScript can access and change all the elements of an HTML document.

The HTML DOM (Document Object Model): When a web page is loaded, the browser creates a Document Object Model of the page.

With the object model, JavaScript gets all the power it needs to create dynamic HTML:

- JavaScript can change all the HTML elements in the page
- JavaScript can change all the HTML attributes in the page
- JavaScript can change all the CSS styles in the page
- JavaScript can remove existing HTML elements and attributes
- JavaScript can add new HTML elements and attributes
- JavaScript can react to all existing HTML events in the page
- JavaScript can create new HTML events in the page

1/. JavaScript - HTML DOM Methods

HTML DOM methods are actions you can perform (on HTML Elements).

HTML DOM properties are values (of HTML Elements) that you can set or change.

2/. JavaScript HTML DOM Document

The HTML DOM document object is the owner of all other objects in your web page. a/. Finding HTML Elements

Method	Description
<code>document.getElementById(<i>id</i>)</code>	Find an element by element id
<code>document.getElementsByTagName(<i>name</i>)</code>	Find elements by tag name
<code>document.getElementsByClassName(<i>name</i>)</code>	Find elements by class name

- Finding HTML elements by id
- Finding HTML elements by tag name
- Finding HTML elements by class name
- Finding HTML elements by CSS selectors
- Finding HTML elements by HTML object collections

b/. Changing HTML Elements

Property	Description
<code>element.innerHTML = new html content</code>	Change the inner HTML of an element
<code>element.attribute = new value</code>	Change the attribute value of an HTML element
<code>element.style.property = new style</code>	Change the style of an HTML element
Method	Description
<code>element.setAttribute(attribute, value)</code>	Change the attribute value of an HTML element

c/. Adding and Deleting Elements

Method	Description
<code>document.createElement(element)</code>	Create an HTML element
<code>document.removeChild(element)</code>	Remove an HTML element
<code>document.appendChild(element)</code>	Add an HTML element
<code>document.replaceChild(new, old)</code>	Replace an HTML element
<code>document.write(text)</code>	Write into the HTML output stream

d/. Adding Events Handlers

Method	Description
<code>document.getElementById(id).onclick = function(){code}</code>	Adding event handler code to an onclick event

III/. Javascript Events

HTML events are "things" that happen to HTML elements.

When JavaScript is used in HTML pages, JavaScript can "react" on these events.

An HTML event can be something the browser does, or something a user does.

Here are some examples of HTML events:

- An HTML web page has finished loading
- An HTML input field was changed
- An HTML button was clicked

Often, when events happen, you may want to do something.

JavaScript lets you execute code when events are detected.

HTML allows event handler attributes, with JavaScript code, to be added to HTML elements. 1/. Input Events

onblur - When a user leaves an input field
onchange - When a user changes the content of an input field
onchange - When a user selects a dropdown value
onfocus - When an input field gets focus
onselect - When input text is selected
onsubmit - When a user clicks the submit button
onreset - When a user clicks the reset button
onkeydown - When a user is pressing/holding down a key
onkeypress - When a user is pressing/holding down a key
onkeyup - When the user releases a key
onkeyup - When the user releases a key
onkeydown vs onkeyup - Both

2/. Mouse Events

onmouseover/onmouseout - When the mouse passes over an element
onmousedown/onmouseup - When pressing/releasing a mouse button
onmousedown - When mouse is clicked: Alert which element
onmousedown - When mouse is clicked: Alert which button
onmousemove/onmouseout - When moving the mouse pointer over/out of an image
onmouseover/onmouseout - When moving the mouse over/out of an image
onmouseover an image map

3/. Click Events

Acting to the onclick event
onclick - When button is clicked
ondblclick - When a text is double-clicked

4/. Load events

onload - When an image has been loaded
onerror - When an error occurs when loading an image
onunload - When the browser closes the document
onresize - When the browser window is resized

IV/. Practice Javascript

Exercise 01: (Using Popup Boxes)

Create a website that performs the following functions:

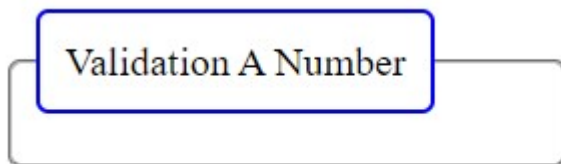
- Displays a dialog box asking for a username.
- If the user does not enter a name, an error message is displayed.
- Otherwise, display a dialog asking the user's gender. Then report back the information that the user has entered.

You can try:

```
<script>
  var fullname = prompt("Please enter your name: ", "Enter your full name here...");
  if (fullname == '' || fullname == null) {
    alert("You must enter your name!")
  }else{
    var gender = confirm("Are you male?" +
      "\nClick 'OK' if you are male" +
      "\nClick 'Cancel' if you are female");
    if (gender) {
      alert("Welcome Mr " + fullname);
    }else{
      alert("Welcome Mrs " + fullname);
    }
  }
</script>
```

Exercise 02: (Validate number) Create

an interface like this:



- Displays a dialog box to enter an integer.
- If the input value is not an integer, an error is displayed as shown.

Validate A Number

You must enter an integer number

- If the input value is an integer and this number is even. A message will be displayed as shown.

Validate A Number

Entered number is even

- If the input value is an integer and this number is odd. A message will be displayed as shown.

Validate A Number

Entered number is odd

You can try:

```

<style>
    legend {
        border: 2px solid blue;
        border-radius: 5px;
        padding: 10px;
    }
    fieldset {
        width: 250px;
        padding: 10px;
        border-radius: 5px;
    }
</style>
</head>
<body>
    <fieldset>
        <legend>Validate A Number</legend>
        <label id="lbValidate"></label>
    </fieldset>
    <script>
        var input = prompt("Please enter an integer number");
        if(input == '' || input == null || !Number.isInteger(Number(input))){
            document.getElementById("lbValidate").innerHTML = "<font color='red'>You must enter an integer number</font>";
            document.getElementById("lbValidate").style.fontWeight = "bolder";
        }else{
            if(input % 2 == 0) {
                document.getElementById("lbValidate").innerHTML = "<font color='blue'>Entered number is even</font>";
                document.getElementById("lbValidate").style.fontWeight = "bolder";
            }else{
                document.getElementById("lbValidate").innerHTML = "<font color='green'>Entered number is odd</font>";
                document.getElementById("lbValidate").style.fontWeight = "bolder";
            }
        }
    </script>

```

Exercise 03: (Draw Rectangle) Create

an interface like this:

- Enter value of width and height into input text. These values must be numbers.
- When user click to button “Generate”, a rectangle will show with these configurations.

Width:

Height:

Color:

Generate



You can try:

```

<style>
  #info {
    width: 250px;
    height: 100px;
    border: 2px solid blue;
    padding: 10px;
  }
</style>
</head>
<body>
  <div id="info">
    <table>
      <tr>
        <td align="right">Width:</td>
        <td><input id="txtWidth" type="text" placeholder="Please enter a number"></td>
      </tr>
      <tr>
        <td align="right">Height:</td>
        <td><input id="txtHeight" type="text" placeholder="Please enter a number"></td>
      </tr>
      <tr>
        <td align="right">Color:</td>
        <td><input id="colorVal" type="color"></td>
      </tr>
      <tr>
        <td colspan="2" align="center">
          <input onclick="Generate()" type="button" value="Generate">
        </td>
      </tr>
    </table>
  </div>
  <div id="rectangle"></div>

```



```

<script>
function Generate(){
  if (document.getElementById("txtwidth").value == '' || document.getElementById("txtwidth").value == null
  || isNaN(Number(document.getElementById("txtwidth").value)) || document.getElementById("txtHeight").value == ''
  || document.getElementById("txtHeight").value == null || isNaN(Number(document.getElementById("txtHeight").value))){
    alert("Please enter correct format of width or height!");
  }else{
    document.getElementById("rectangle").style.width = document.getElementById("txtwidth").value + 'px';
    document.getElementById("rectangle").style.height = document.getElementById("txtHeight").value + 'px';
    document.getElementById("rectangle").style.backgroundColor = document.getElementById("colorVal").value;
    document.getElementById("rectangle").style.margin = "5px";
  }
}
</script>

```

Exercise 04: (Day in month)

Create a form that allows the year and month to be entered and then displays the number of days of that month.

Create an interface like this:

DAY IN MONTH

Year:

Month ▼

Number of days:

If you enter an invalid year, you will get an error message like this:

DAY IN MONTH

Year:

Month ▼

Number of days:

If the input data is valid, the output will be displayed as shown:

DAY IN MONTH

Year:	<input type="text" value="2023"/>
Month	<input type="text" value="02"/>
Number of days:	<input type="text" value="28"/>

You can try:

```
<table>
  <tr>
    <td colspan="2" align="center">
      <h2>DAY IN MONTH</h2>
    </td>
  </tr>
  <tr>
    <td align="right">Year: </td>
    <td>
      <input type="text" id="txtYear" value="2023" onblur="updateInfo()">
    </td>
  </tr>
  <tr>
    <td align="right">Month</td>
    <td>
      <select id="slMonth" onblur="updateInfo()">
        <option value="1" selected>01</option>
        <option value="2">02</option>
        <option value="3">03</option>
        <option value="4">04</option>
        <option value="5">05</option>
        <option value="6">06</option>
        <option value="7">07</option>
        <option value="8">08</option>
        <option value="9">09</option>
        <option value="10">10</option>
        <option value="11">11</option>
        <option value="12">12</option>
      </select>
    </td>
  </tr>
  <tr>
    <td align="right">Number of days: </td>
    <td>
      <input type="text" id="txtNumDays" readonly>
    </td>
  </tr>
</table>
```

```

<script>
function isLeapYear(n){
    return (n%4==0 && n%100!=0) || (n%400==0);
}
function updateInfo(){
    if(document.getElementById("txtYear").value == '' || document.getElementById("txtYear").value == null
    || !Number.isInteger(Number(document.getElementById("txtYear").value))){
        document.getElementById("txtNumDays").value = "Year is not valid";
    }else{
        let month = Number(document.getElementById("slMonth").value);
        switch(month) {
            case 1:
            case 3:
            case 5:
            case 7:
            case 8:
            case 10:
            case 12:
                document.getElementById("txtNumDays").value = 31;
                break;
            case 4:
            case 6:
            case 9:
            case 11:
                document.getElementById("txtNumDays").value = 30;
                break;
            case 2:
                document.getElementById("txtNumDays").value = (isLeapYear(document.getElementById("txtYear").value)) ? 29 : 28;
                break;
        }
    }
}
</script>

```

V/. More Exercises

1/. Create a web page that allows input of a positive integer n and then displays an n-line triangle with the following form:

```

1
1 2
1 2 3
1 2 3 4

```

.....

2/. Create a web page that allows input of a positive integer n and then displays an n-line triangle with the following form: *

```

* * *
* * * * *
* * * * * *

```

.....