1. **Design a Login Page with tag & implement Validation with the help of Traditional Method. If valid, then redirect to another HTML page with an alert message i.e. “Successfully Login” and, if not, then throw the specific alert message & focus the text cursor to that input box. (Attach the Output)**

```
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title> Document</title>
  </head>
 <body>
  <form action="form12.html" method="post" onsubmit="return log()">
    <label> User Name*:</label>
    <input type="text" id="us1">
    <br>
    <br>
    <label> Password*:</label>
    <input type="password" id="pw1">
    <br>
    <br>
    <input type="submit" value="LogIn">
  </form>
  <script src="FormValidation.js"></script>
 </body>
</html>
```

<u>Validation.js</u>

```
function log()
{
  var us=document.getElementById("us1");
  var ps=document.getElementById("pw1");
  if(us.value==''){
    alert("Please Enter your username");
    us.focus();
    return false;
  }
  else if(ps.value=='')
  {
    alert("Please enter your password");
    ps.focus();
    return false;
  }
  else
    {
    alert("Login Successfully");
    return true;
   }
}
```

2. **Design a Login Page with tag & implement Validation with the help of Regular Expression (RegEx). If valid, then redirect to another HTML page with an alert message i.e. “Successfully Login” and, if not, then throw the specific alert message & focus the text cursor to that input box. (Attach the Output) ➢ The Username should consist of Alpha-Numeric form like at least 1 Uppercase, 1 Lowercase & 1 Number. (Ex: AbcXyz123) ➢ The Password should consist of Alpha-Numeric with Special Symbol like at least 1 Uppercase, 1 Lowercase, 1 Number & 1 Special Character. (Ex: XyAb@12#)**

```
function log()
{
  var us=document.getElementById("us1");
  var ps=document.getElementById("pw1");


  // Regular Expression


  var ARegex=/[A-Z]+/;
```

```
   var aRegex=/[a-z]+/;

   var NRegex=/[0-9]+/;

   var spRegex=/\W+/;

   var uflag=0, pflag=0;

   if(us.value.match(ARegex)&&us.value.match(aRegex)&&us.value.match(NRegex))

   {

      uflag=1;

   }

   if(ps.value.match(ARegex)&&ps.value.match(aRegex)&&ps.value.match(NRegex)&&ps.value.match
(spRegex))


      if(ps.value.length>=8 && ps.value.length<=13)

      {

         pflag=1;

      }

      else

      {

         alert("PassWord should be in between 8 to 13 character!");

      }


   if(uflag==0||pflag==0)

   {

      if(uflag==0)

      {

         alert("User should contain at least one uppercase,one lowercase or one digit!");

      us.value="";

      us.focus();

      return false;

      }

   else

   {
      alert("Password should contain at least one uppercase,one lowercase,one digit and one spcharacter!");
        ps.value="";
        ps.focus();
        return false;
    }
}

else
{
    alert("Login Successfully");
    return true;
}


}
```

3. **Design a Login Page with tag & apply JavaScript to show or hide the Password by using the eye or eye-slash icon, whenever you will click on it. (Attach the Output) Use the following CSS file to apply the above-mentioned icon:**

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/4.7.0/css/font-awesome.min.css">

```
<html>
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title> Document</title>
        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
    </head>
   <body>
    <form action="" method="post">
```

```
        <label> User Name*:</label>
        <input type="text" id="us1">
        <br>
        <br>
        <label> Password*:</label>
        <input type="password" id="pw1">
        <i class="fa fa-eye-slash" id="bt1" onclick="abc()"></i>
        <br>
        <br>
        <input type="submit" value="LogIn">
        </form>
        <script src="cssclick.js"></script>
   </body>
</html>
```

cssclick.js

```
var us=document.getElementById("us1");
var ps=document.getElementById("pw1");
var bt=document.getElementById("bt1");

var hidebt=document.createElement("i");
hidebt.setAttribute("class", "fa fa-eye-slash");
hidebt.onclick=abc;
bt.onclick=abc;

function abc()
{
    if(ps.type=="password")
    {
        ps.type="text";
        bt.setAttribute("class", "fa fa-eye");
```

```
   }
   else
   {
      ps.type="password";
      bt.setAttribute("class", "fa fa-eye-slash");
   }
}
```

4. **Design a Registration Page with tag & apply JavaScript to preview the file (i.e. .jpg, .png, .jpeg, .pdf, .doc, etc) selected by yourself for Final Submission or Uploading purposes. Whenever you will click on the "preview" button, the selected file will be shown in a particular section of the webpage. (Attach the Output):**

```
<html>
   <head>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
      <title> Document</title>

   </head>
  <body>
   <form action="csslink.html" method="post">
      <label> User Name*:</label>
      <input type="text" id="us1">
      <br>
      <br>
      <label> Password*:</label>
      <input type="password" id="pw1">
      <br>
      <br>
      <input type="file" accept=".jpeg,.png,.jpg" id="img1">

      <input type="button" value="preview" id="btn" onclick="abc1()">
```

```
        <div id="d1">
            <!----Show the previewd image-------->
        </div>
        <br>
        <br>
        <input type="submit" value="LogIn">
        </form>
        <script src="Formpreview.js"></script>
    </body>
</html>
```

Formpreview.js

```
function abc1()
{
    var f1=document.getElementById("img1");
    var file=f1.files[0];
    var imgurl=URL.createObjectURL(file);

    var dv1=document.getElementById("d1");
    dv1.innerHTML="";
    var img1=document.createElement("img");
    img1.src=imgurl;
    img1.height=300;
    img1.width=350;

    dv1.append(img1);

}
```

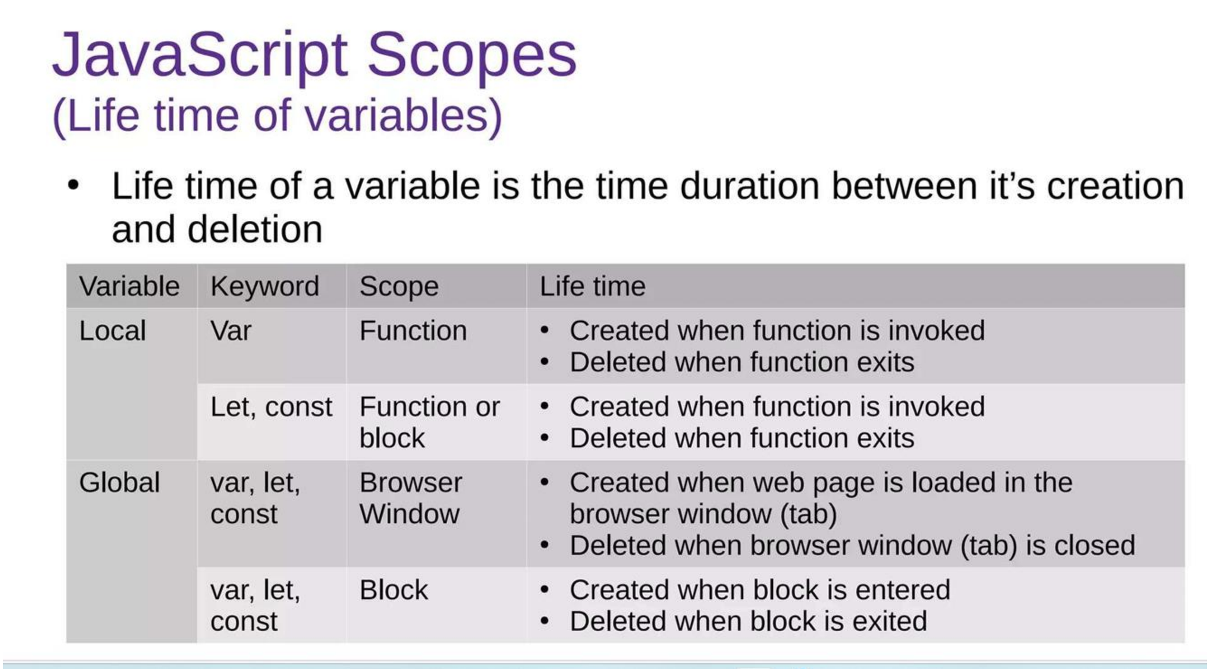
11.

```
<html>
    <head>
        <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title> Document</title>


 </head>
<body>
 <p id="P1"> FSWD-1 Lab</p>
 <p id="P2">
    <b> Bold 1</b>
    <i> Italic1</i>
    <!-- JS Programe -->
 </p>
 <b id="P3">Hello</b>
 <script src="DoumentInsert.js"></script>
</body>
</html>
```

5. **Example for insertBefore() similarly do yourself for removeChild() and replaceChild()**

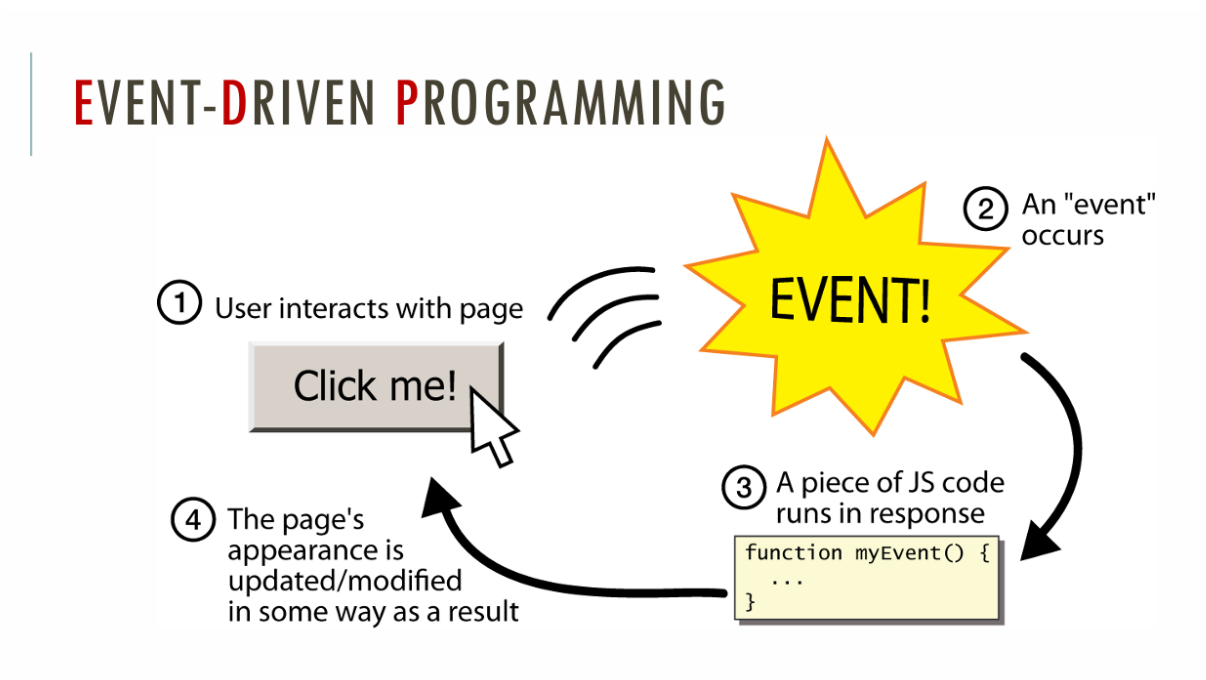
```
var p1=document.getElementById("P2");
var b1=document.createElement("b");
b1.innerText="Bold2";
p1.insertBefore(b1,p1.children[1]);
var b2=document.createElement("b");
b2.innerText="Bold 3";
p1.insertBefore(b2,p1.children[1]);
```



JavaScript Scopes
(Life time of variables)

- Life time of a variable is the time duration between it's creation and deletion

Variable	Keyword	Scope	Life time
Local	Var	Function	• Created when function is invoked • Deleted when function exits
	Let, const	Function or block	• Created when function is invoked • Deleted when function exits
Global	var, let, const	Browser Window	• Created when web page is loaded in the browser window (tab) • Deleted when browser window (tab) is closed
	var, let, const	Block	• Created when block is entered • Deleted when block is exited



EVENT-DRIVEN PROGRAMMING

① User interacts with page
Click me!
EVENT!
② An "event" occurs
③ A piece of JS code runs in response
④ The page's appearance is updated/modified in some way as a result

```
function myEvent() {
  ...
}
```

Events

- An event is some notable action to which a script can respond. It may be
  - Click
  - Mouseover
  - Keystroke etc.
- When a function is assigned to an event handler, that function is run when that event occurs.
- An Event handler is JavaScript code associated with a particular part of the document and a particular event.

Event Handler

- An Event handler is JavaScript code associated with a particular part of the document and a particular event.

- For example, an event handler associated with a button could open a new window when the button is clicked.

- A handler to the click event is called onclick.

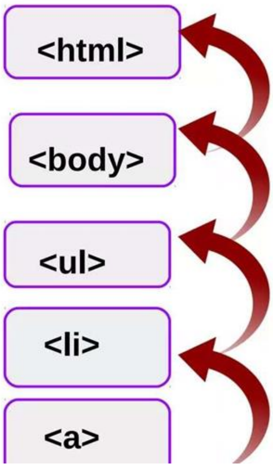
Event Flow

- The order in which events are received on the web page are described by event flow.

- An event has three phases :
  - Cycle
  - Target
  - Bubbling

Event Bubbling

- Event Bubbling , the event is first captured and handled by innermost element and then propagated to outer element.



<html>

<body>

<ul>

<li>

<a>

Mouse Events

Event Attribute	Description
`onclick`	Occurs when the mouse button is clicked
`ondblclick`	Occurs when the mouse button is double clicked
`onmousedown`	Occurs when the mouse button is pressed
`onmouseup`	Occurs when the mouse button is released
`onmousemove`	Occurs when mouse has moved while over an element.
`onmouseover`	Occurs when mouse has moved over an element.
`onmouseout`	Occurs when mouse has moved away from an element.

Keyboard Events

Event Attribute	Description
`onkeypress`	Occurs when a key pressed and released with focus on element
`onkeydown`	Occurs when a key pressed down
`onkeyup`	Releases a key

Loading Events

Event Attribute	Description
`onload`	Occurs when element has loaded
`onunload`	Indicates that browser is leaving the current document
`onabort`	Occurs when the user abort the loading of an image

Selection and Focus Event

Event Attribute	Description
`onselect`	Occurs after some text has been selected in an element
`onchange`	Occurs when text input has been changed
`onfocus`	Indicates that an element has received focus
`onblur`	Occurs when an element losses focus

Loading Events

Event Attribute	Description
`onload`	Occurs when element has loaded
`onunload`	Indicates that browser is leaving the current document
`onabort`	Occurs when the user abort the loading of an image

Other Events

Event Attribute	Description
`onresize`	User resizes a window or a frame
`onsubmit`	Indicates form submission by clicking a submit button
`onreset`	Indicates that form is being reset by clicking reset button