

Practical-5

Aim: JS Events handling, JavaScript Array, Callbacks & Timer Functions

- Demonstrate various Ways to handle Events

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <button id="addevent"> with add event listener</button>
  <button id="oc" onclick="r();"> with onclick</button>
  <!-- <button id="oc" onclick="r();"> with onclick</button> -->
  <button id="ocd"> with DOM onclick</button>
  <p id="p"></p>
  <script>
    let p = document.getElementById("p"); function r() {
    // document.write(`handle by onclick in html `) p.innerHTML = `handle by onclick in html`
    }
    let btn1 = document.getElementById("addevent") let btn2 = document.getElementById("ocd")
    btn1.onclick = () => {
    p.innerHTML = `handle with add event listner in js`
    // document.write(` `)
    }
    btn2.onclick = function () {
    p.innerHTML = `handle by with DOM Onclick property in js`
    // document.write(` `)
    }
  </script>
</body>
</html>
</html>
```

Output:

with add event listener

with onclick

with DOM onclick

handle with add event listner in js

Conclusion: In this practical we learn that how we can handle events in a single click.

- Write the Script to Change the background color randomly after every 1 minutes.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
</head>
<body>
<script>
function raj(){
let ff1 = Math.floor(Math.random() * 100) let ff2 = Math.floor(Math.random() * 100) let
ff3 = Math.floor(Math.random() * 100) console.log(ff1)
document.body.style.backgroundColor = "#" + ff1 + ff2 + ff3
}
// document.body.style.backgroundColor = "pink" setInterval(prachi, 1000);
</script>
</body>
</html>
```

Output:



Conclusion: In this practical we learn how change colors after specific period of time.

- Create Arrays of Colors and Apply these colors in order to tr background of table.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Document</title>
</head>
<body>
  <table border="2px solid gray" style="border-collapse:
collapse;border-spacing: 10px; font-size: 50px;">
    <tr id="tr0">
      <th>red</th>
      <th> background-color</th>
      <th> background-color</th>
    </tr>
    <tr id="tr2">
      <td>orange</td>
      <td>background-color</td>
      <td>background-color</td>
    </tr>
    <tr id="tr3">
      <td>yellow</td>
      <td>background-color</td>
      <td>background-color</td>
    </tr>
    <tr id="tr1">
      <td>pink</td>
      <td>background-color </td>
      <td> background-color</td>
    </tr>
  </table>
  <script>
    var arr = ["red","pink","orange","yellow"] for (let i = 0; i <
4; i++) {
      let x = document.getElementById("tr"+i)
      x.style.backgroundColor = arr[i];
    }
  </script>
</body>
</html>
```

Output:

red	background-color	background-color
orange	background-color	background-color
yellow	background-color	background-color
pink	background-color	background-color

Conclusion: In this practical we learn how colors change with its text

- Zoom image on mouse over and zoom out on mouse out

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Document</title>
</head>
<body>
  <div class="box">
    
  </div>
</button>
<style>
  button{
    position: relative;;
    margin: 5px;
    font-size: 30px;
    border: 2px solid black;
  }
  img{
    cursor: zoom-in;
  }
</style>
<script type="text/javascript">
  add
  function big() {
    var x = document.getElementById("pt");
    x.style.width = 1000 + "px";
  }
  function small() {
    var x = document.getElementById("pt");
    x.style.width = 500 + "px";
  }
</script>
</body>
</html>
```

Output:



Conclusion: In this practical we learn how to hover our mouse and picture zooms in and zoom out.

- Change the location of image based on arrow key of keyboard using callback function

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  
  <script>
    document.onkeydown = checkKey;
    function checkKey(e) {
      let i = document.getElementById("i");
      e = e || window.event;
      if (e.keyCode == '38') {
        let x = i.style.top;
        let y = parseInt(x, 10) + 10 ;
        i.style.top = y + "px"
        console.log(x,y)
      }
      else if (e.keyCode == '40') {
        let x = i.style.top;
        let y = parseInt(x, 10) - 10 ;
        i.style.top = y + "px"
      }
      else if (e.keyCode == '37') {
        let x = i.style.left;
        let y = parseInt(x, 10) + 10 ;
        i.style.left = y + "px"
      }
      else if (e.keyCode == '39') {
        let x = i.style.left;
        let y = parseInt(x, 10) - 10 ;
        i.style.left = y + "px"
      }
    }
  </script>
</body>
</html>
```

Output:



Conclusion: In this practical we learn how to change location of image using callback function.