

Practical – 5

AIM:

Demonstrate various Ways to handle Events

HTML Source Code:

```
<!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>
  <h1>Change Message</h1>
  <input type="text" id="myInput" onclick="alert('Input was
  clicked!')"onblur="alert('Input lost focus!')"
                                onmouseover="alert('Mous
  eover on Input!')" onkeypress="alert('key pressed!')" >
```

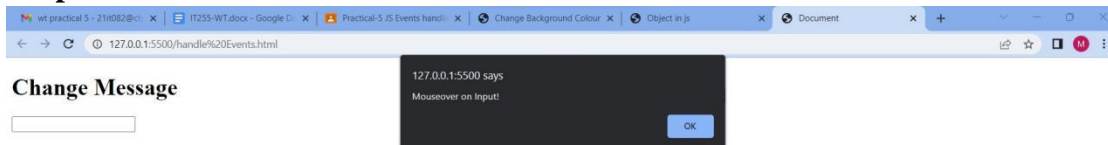
Javascript Source code:

```
const input =
document.querySelector("#myInput");
input.addEventListener("click",
handleClick);
input.addEventListener("blur",
handleBlur);
input.addEventListener("mouseover",
handleMouseOver);
input.addEventListener("keypress",
handleKeyPress);

function handleClick() {
```

```
}  
  
function handleKeyPress(e)  
{ console.log("Key  
pressed:" + e.key);  
}
```

Output:



AIM:

Write the Script to Change the background color randomly after every 1 mins

HTML Source Code:

```
<!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>Change Background Colour</title>
```

```
<body>

</body>
</html>
```

Javascript Source code:

```
let colors = ['red', 'orange', 'yellow', 'green',
'blue', 'indigo', 'violet'];

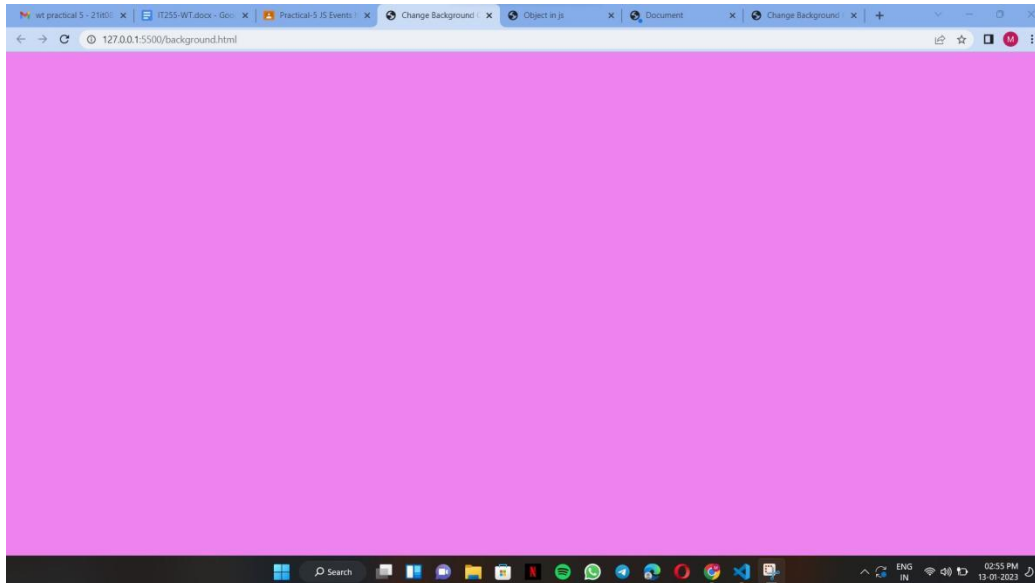
let currentIndex = 0;

function changeColor() {
    let body = document.querySelector('body');

    body.style.backgroundColor =

    colors[currentIndex];

    currentIndex = (currentIndex + 1) %
```

Output:

AIM:

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

HTML Source Code:

```
<!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>Change Table Colour</title>
  <
    s
    t
    y
    l
    e
    >

    t
    a
    b
    l
    e
    ,
    t
    d
    {
      border: solid black;
    }
  </style>
</head>
```

Javascript Source code:

```
var colors = ["red", "orange", "yellow", "green",  
"blue", "indigo", "violet"];  
var currentIndex = 0;  
  
function changeTableColor() {  
    var table = document.getElementById("myTable");  
    var colorInput =  
    document.getElementById("colorInput").value; var  
    color = colors[colorInput - 1];  
    table.style.backgroundColor = color;  
}
```

Output:

Change Table Colour

Colour Number:

ID	Name	Age
1	abc	19

AIM:

Zoom image on mouseover and zoom out on mouse out

HTML Source Code:

```
<!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>
```

```


<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>Move Image</title>

</head>
```

```


<script src="./locImg.js"></script>

</body>
```

Javascript Source code:

```
let img =

document.getElementById("myImage");

document.addEventListener("keydown",

moveImage);

function

moveImage(event

nt) {switch

(event.keyCode

e) {

case 37:

img.style.left = (img.offsetLeft -
```



```
case 40:  
  
    img.style.top = (img.offsetTop +  
  
    10) + "px";break;  
  
}  
  
event.preventDefault();
```

Output:

**Conclusion:**

A callback function is a function passed into another function as an argument, which is then invoked inside the outer function to complete some kind of routine or action. A timer is used to execute some task after a particular time interval.