

الله رب العالمين

Web Technologies and Programming

Lecture 19

Summary of Previous Lecture

- Dialog boxes in JavaScript
 - Alert box
 - Prompt box
 - Confirm box
- What is DOM?
- HTML DOM
- Retrieving HTML elements
 - `getElementById()`
 - `innerHTML`
 - `getAttribute()`
 - `setAttribute()`

Summary of Previous Lecture

- **HTML DOM Elements**
- **Finding HTML Elements**
 - **Find an element by element**
 - **Find elements by tag**
 - **Find elements by class name**

Today's Lecture Outline

- **Controlling the background dynamically**
- **Working with images**
- **Date Object**

1. Controlling the background

- The **body tag** has the following attributes
 - **Bgcolor**
 - It is Background Color
 - **Background**
 - It deals with the Background Image
 - **Text**
 - It is the text user sees in the browser

1.1 Setting the bgcolor

- We can **access** the bgcolor by following methods:
 - `document.body.bgColor`
 - `document.getElementById("id of the body tag").bgColor`

1.1 Setting the bgcolor...

```
<html>
<head>
    <title>BGcolor Setting</title>
    <script>
        function setBgcOLOR() ← Starting Function
        {
            var bg = prompt("Enter bgcolor") ← User Input

            document.getElementById("body") .bgColor = bg
        }
    </script>
</head>

<body id="body" onload="setBgcOLOR()">
    <h1>Hello Dear!</h1> ← Calling Function
</body>
</html>
```

1.1 Setting the bgcolor...

The image shows a screenshot of a web browser window. The address bar displays the URL: file:///C:/Users/Ahmad%20Kakakhail/Desktop/bbgcolor.html. The main content area of the browser shows the text "Hello Dear!".

A modal dialog box is overlaid on the browser window. The dialog has a title "This page says:" and contains the instruction "Enter bgcolor". Below this is a text input field containing the value "Yellow". At the bottom of the dialog are two buttons: "OK" and "Cancel".

A red arrow points from the text "User Inputs bgcolor in Prompt Box" to the "OK" button of the dialog box. Another red arrow points from the text "After Applying bgcolor" to the "Hello Dear!" text in the browser's main content area, indicating the result of the user's input.

User Inputs bgcolor in Prompt Box

After Applying bgcolor

1.2 Setting the text color

- Set a background color for a document:
- We can **access** the body text by following methods:
 - `document.body.style.color`
 - `document.getElementById("id of the body tag").style.color`

1.2 Setting the text color

- **Example**
- Set a background color of a specific <div> element:
- `document.getElementById("myDiv").style.backgroundColor = "lightblue";`

1.2 Setting the text color

- **Example**
- Return the background color of a specific `<div>` element:
`alert(document.getElementById("myDiv").style.backgroundColor);`
- **Example**
- Return the background color of a document:
`alert(document.body.style.backgroundColor);`

1.2 Setting the text color...

```
<html>
  <head>
    <title>BGcolor setting</title>
    <script>
      function setText(txtcolor) ← Set Text Function
      {
        document.body.style.color = txtcolor ← Setting Color
      }
    </script>
  </head>          Calling Function
  <body>          ← Passing Color as Parameter
    <p onClick="setText('blue')"> Click the Text to Make All the text Blue </p>
    <p onClick="setText('red')"> Click the Text to Make All the text Red </p>
    <p onClick="setText('orange')"> Click the Text to Make All the text Orange </p>
  </body>
</html>
```

1.2 Setting the text color...



Click the Text to Make All the text Blue

Click the Text to Make All the text Red

Click the Text to Make All the text Orange

← Before Clicking on Text



Click the Text to Make All the text Blue

Click the Text to Make All the text Red

Click the Text to Make All the text Orange

← After Clicking on Text
for Red Color

1.3 Setting the background

- The **background** property sets or returns up to eight separate background properties, in a shorthand form.
- With this property, you can set/return one or more of the following (in any order):
 - **background-color**
 - **background-image**
 - **background-repeat**
 - **background-attachment**
 - **background-position**
 - **background-size**
 - **background-origin**
 - **background-clip**

1.3 Setting the background

- **Example**
- Set a **background color** for a document:
`document.body.style.backgroundColor = "red";`
- **Example**
- Set a **background image** for a document:
`document.body.style.backgroundImage = "url('img_tree.png')";`

1.3 Setting the background

- **Example**
- Set a background-image to no-repeat:
`document.body.style.backgroundRepeat = "repeat-y";`
- **Example**
- Set the background-image to be fixed (will not scroll):
`document.body.style.backgroundAttachment = "fixed";`

1.3 Setting the background

- **Example**
- Change the position of a background-image:
- `document.body.style.backgroundPosition = "top right";`

1.3 Setting the background

- We can **access** the background by following methods:
 - `document.body.background`
 - `document.getElementById("id of the body tag").background`

1.3 Setting background...

```
<html>
<head>
    <title>Background Setting</title>
    <script>
        function setBackground(bgSrc) ← Function for Receiving Src
        {
            document.body.background=bgSrc ← Setting Background
        }
    </script>
</head>           Image Added          Function Call          Src Passed
<body>
    
    
</body>
</html>
```

The diagram illustrates the flow of data and control between the HTML code and the JavaScript function. Red arrows point from the 'Image Added' section to the 'Function Call' section, and from the 'Function Call' section to the 'Src Passed' section. The 'Image Added' section points to the two tags in the body. The 'Function Call' section points to the onClick attribute of the first tag. The 'Src Passed' section points to the string 'bg1.jpg' within the onClick attribute.

1.3 Setting background...



← Before Clicking on Image



← After Clicking on Image

2. Working with images

- JavaScript has a **built-in** object called **Image**
- The **Image** object **allows** you to create **objects** that represent images created using the **** element
- **Properties:** name, border, height, width, hspace, vspace, src

2. Working with images...

- An image object can be created as
 - `var objname= new image()`
- The most important property of an image object is **src**
 - This **designates** the actual image file you will use as a source for your Image object
 - `objname.src= “abc.jpg”`
 - This property can be used to **dynamically change** an image on the webpage

2. Working with images....

- The **primary use** for an **Image** object is to **download** an image into the **cache** before it is actually needed for **display**
 - **Instance** of image object is created
 - **Src** property is set to the name of the image to be downloaded

2. Working with images....

- **Image Object**
- The Image object represents an **HTML ** element.
- **Access an Image Object**
- You can access an **** element by using **getElementById()**:

2.1 Image MouseMove and MouseOut Function

```
<html>
<head>
    <title>Image RollOver</title>
    <script language="javascript">
        function changePic(imgSrc) ← Function Start
        {
            document.getElementById("pic").src=imgSrc
        }
    </script>
</head>

<body bgcolor="Blue">
    <h1>Image Changing on Mouse Over and Mouse Out</h1>
    
</body> ↑
</html>
```

Image Addition **Calling MouseOver Function** **Calling MouseOut Function**

Setting Image Src

2.1 Image MouseMove and MouseOut Function...



← **On MouseOut**



← **On MouseOver**

2.2 Image Preview

```
<body bgcolor="Violet">
  <h1>Image Preview</h1>
  <table width="711" border="0" align="center">
    <tr>
      <td height="297">
        
      </td>
    </tr>
    <tr>
      <td>
        
        
        
        
        
      </td>
    </tr>
  </table>
</body>
```

Main Image

Small Images

Src Passed

Function Call

Diagram annotations:

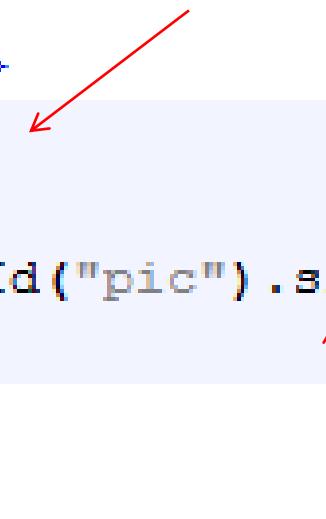
- A red bracket labeled "Main Image" groups the first `<td>` element under the `<tr>`.
- A red bracket labeled "Small Images" groups the five `` elements under the second `<td>` element.
- A red arrow labeled "Image Src" points from the `src` attribute of the large image to the `src` attribute of the first small image.
- A red arrow labeled "Src Passed" points from the `src` attribute of the large image to the `changePic` function call of the first small image.
- A red arrow labeled "Function Call" points from the `changePic` function call of the first small image to the `changePic` function call of the fifth small image.

2.2 Image Preview...

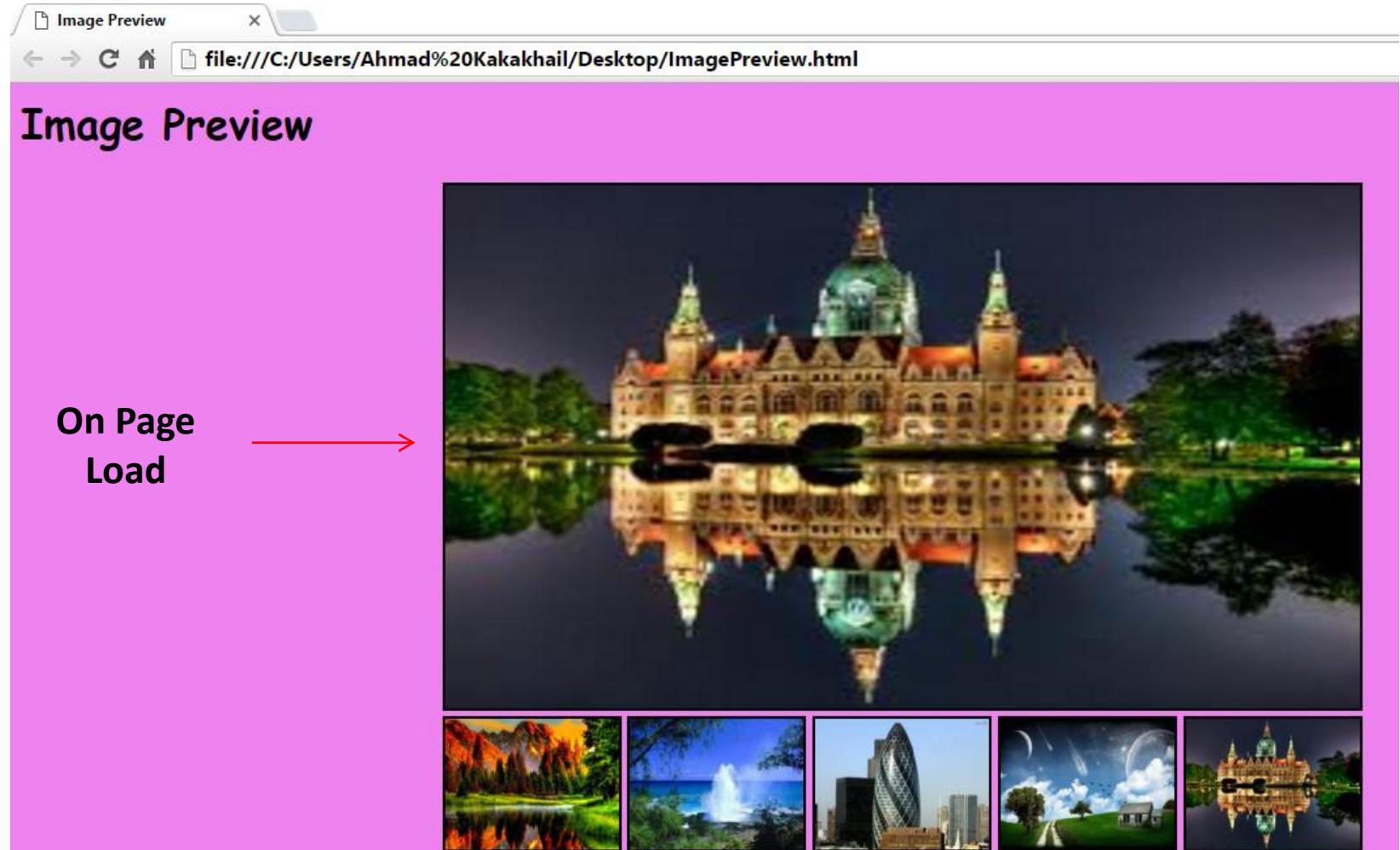
```
<head>
    <title>Image Preview</title>
    <script language="javascript">
        function changePic(imgSrc)
        {
            document.getElementById("pic").src=imgSrc
        }
    </script>
</head>
```

Function Gets Src

Main Image Src Changed



2.2 Image Preview...



2.2 Image Preview...

Image Preview

file:///C:/Users/Ahmad%20Kakakhail/Desktop/ImagePreview.html

After
Changing
Source

→

The image shows a landscape with a dirt road leading towards a small house on a hill under a sky with two large planets and a meteor. A red arrow points from the text 'After Changing Source' to the main image. Below the main image is a row of five smaller images: a forest scene, a beach scene, a city skyline, a landscape with a crescent moon, and a building reflected in water.

2.3 Slide Show

- **Array of images**
- **Main image**
- **Next and previous image buttons**
- **Next image and previous image functions**

2.3 Slide Show...

```
<head>
    <title>Slide Show</title>
    <script language="javascript">
        pics=new Array("img1.jpg","img2.jpg","img3.jpg","img4.jpg","img5.jpg")
        count=0
        pcount=pics.length
        function npic() Increases Pics Index
        {
            count++
            if(count < pcount) Checks for Last Image
                document.pic.src=pics[count]
            else Set count to zero incase of last image
                count=0
                document.pic.src=pics[count]
        }
    </script>
</head>
```

Next Pic

Array of Images

Length of Pics Array

Increases Pics Index

Checks for Last Image

Set count to zero incase of last image

Changes the Image

2.3 Slide Show...

```
function ppic()
{
    count-- ← Count Decrement
    if(count>-1) ← Checks for First Image
    {
        document.pic.src=pics[count]
    }
    else
    {
        count=pcount-1 ← Sets Count incase of First Image
        document.pic.src=pics[count]
    }
}
</script>
</head>
```

Count Decrement

Checks for First Image

Changes The Image

Sets Count incase of First Image

2.3 Slide Show...

```
<body bgcolor="violet">
  <table width="200" border="0" align="center">
    <tr>
      <td>
        
      </td>
    </tr>
    <tr>
      <td align="center">
        
        
      </td>
    </tr>
  </table>
</body>
```

Main Image

Back Image

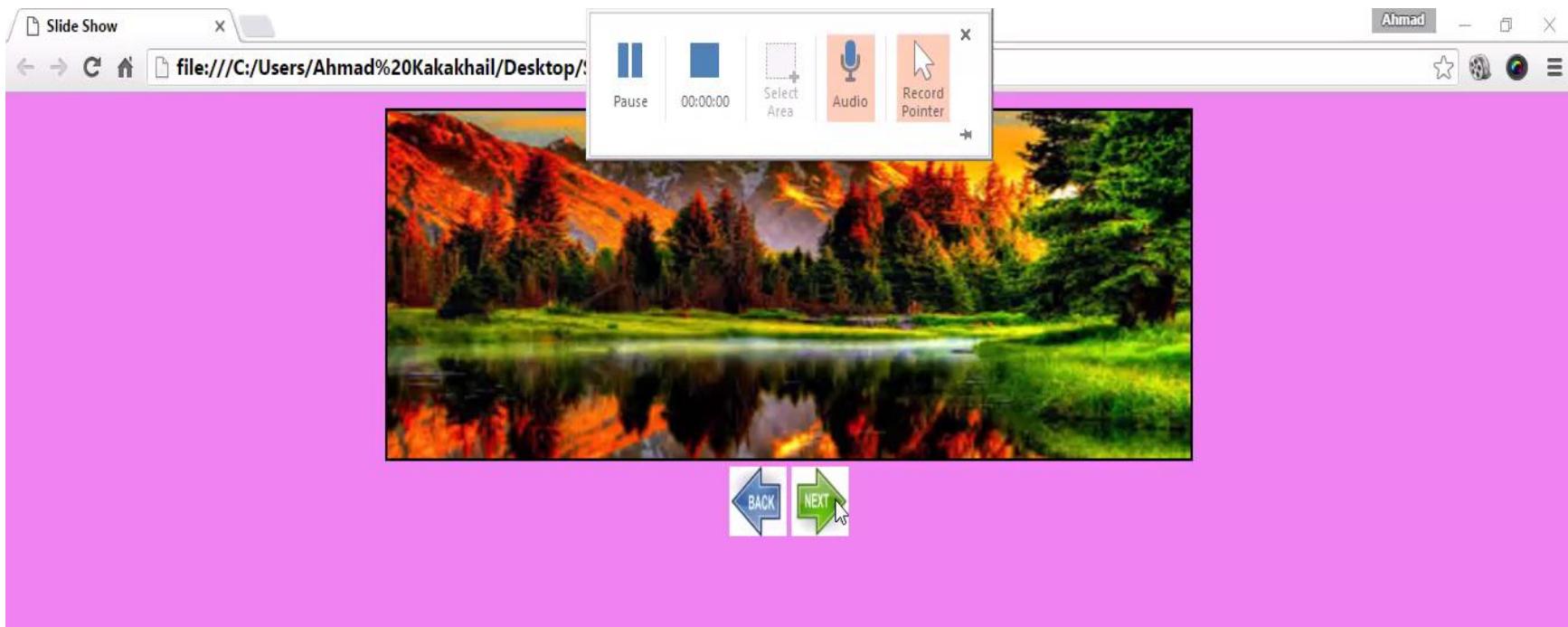
Next Image

Call to Previous Pic Function

Call to Next Pic Function

The diagram illustrates a slide show interface. It features a main image (img1.jpg) centered in a table cell. Below the main image are two smaller images (img6.jpg and img7.jpg) aligned horizontally, also centered in a table cell. Red annotations identify the main image and the two smaller images as "Back Image" and "Next Image". Red arrows point from the text labels to their corresponding HTML elements. A red bracket on the left side groups the first two rows of the table. Two red arrows originate from the text labels "Call to Previous Pic Function" and "Call to Next Pic Function", pointing to the onclick attributes of the images in the second table row.

2.3 Slide Show...



2.4 Slide Show (with Self Changing Images)

- **Array of images**
- **Main image**
- **Next image function**

2.4 Slide Show (with Self Changing Images)...

```
<head>
    <title>Slide Show with Self Changing Images</title>
    <script language="javascript">
        pics=new Array("img1.jpg","img2.jpg","img3.jpg","img4.jpg","img5.jpg")
        count=0
        pcount=pics.length

        function npic()      Increment the count
        {
            count++          Checks for the last image
            if(count<pcount)
                document.pic.src=pics[count]           Image changes
            else
            {
                count=0           Set Count to Zero incase of Last Image
                document.pic.src=pics[count]
            }

            setTimeout("npic()",2000)           Time Based
        }                                    Function Call

    </script>
</head>
```

2.4 Slide Show (with Self Changing Images)...

```
<body bgcolor="violet" onload="npic()>
  <table width="200" border="0" align="center">
    <tr>
      <td>
        
      </td>
    </tr>
  </table>
</body>
```

← Next Pic Function Call

Main Image

2.4 Slide Show (with Self Changing Images)...



3. Working with date object

- The **Date Object** is used to work with dates and times
- It can be used to find the **Current Time** and **Date**
- Many time based tasks can be performed
- Date objects are created as
 - **var today=new Date()**

3. Working with date object...

- **Date object methods:**
 - **getDate():**
 - Returns the day of the month (from 1-31)
 - **getDay():**
 - Returns the day of the week (from 0-6)
 - **getFullYear():**
 - Returns the year (four digits)
 - **getMonth():**
 - Returns the month (from 0-11)

3. Working with date object...

- **getHours():**

- Returns the hour (from 0-23)

- **getMilliseconds():**

- Returns the milliseconds (from 0-999)

- **getMinutes():**

- Returns the minutes (from 0-59)

- **getSeconds():**

- Returns the seconds (from 0-59)

3.1 Digital clock

```
<script>
    function startTime()
    {
        var today = new Date();
        var h = today.getHours();
        var m = today.getMinutes();
        var s = today.getSeconds();
        m = checkTime(m);
        s = checkTime(s);
        document.getElementById('txt').innerHTML = h ":" + m ":" + s;
        var t = setTimeout(function() {startTime()}, 500);
    }

```

Date Object is Created

Gets Hours

Gets Minutes

Gets Seconds

Calls Check-Time Function

Self Function Call

Writes Time on txt Div

3. Working with date object...

```
function checkTime(i)
{
    if (i<10)
        {i = "0" + i};

    return i;
}
</script>
</head>

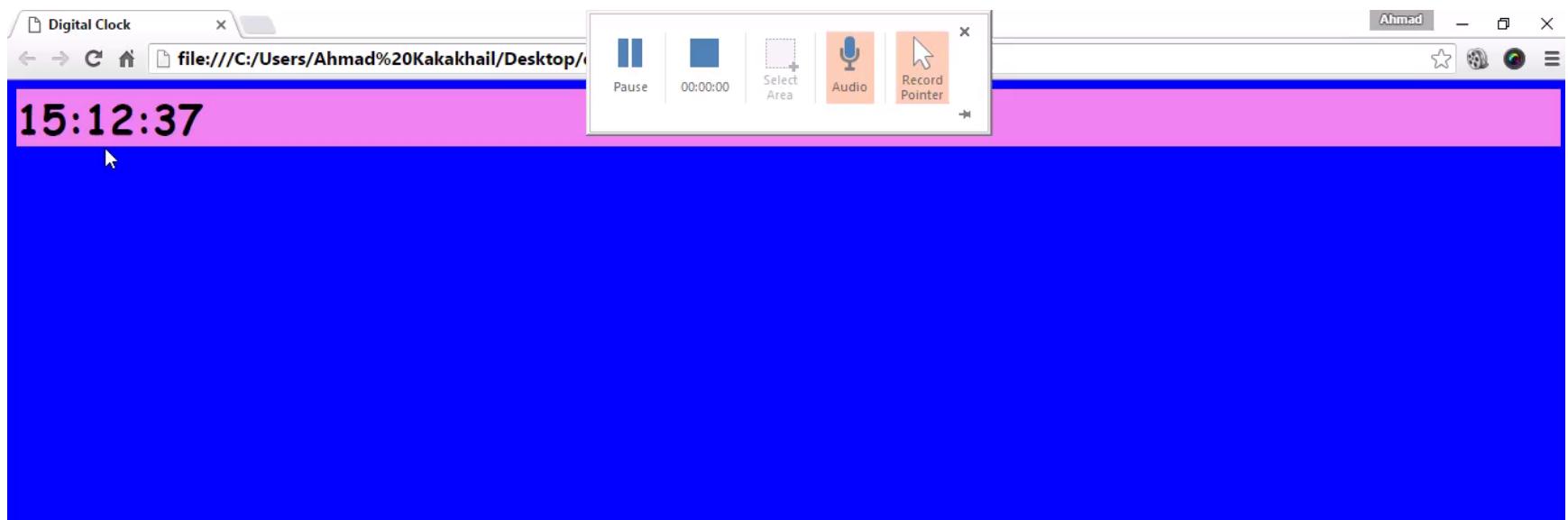
<body bgcolor="blue" onload="startTime()">
    <div id="txt" style="font-size:36px; font-weight:bold; background-color:violet"> </div>
</body>
```

Concatenate '0' If Time is Less Than 10

Start Time Function Call

Empty div to Display Time

3. Working with date object...



Summary of Today's Lecture

- Controlling the background dynamically
 - **Bgcolor**
 - SET For a document
 - SET For a specific <div>
 - SET the Text Color
 - Return background color of a specific <div> element
 - Return background color of a document:
 - **Background**
 - background color for a document
 - background image for a document
 - Set a background-image to no-repeat
 - Set the background-image to be fixed
 - Change the position of a background-image

Summary of Today's Lecture

- Working with images
 - Access an Image Object
 - On MouseOut
 - On MouseOver
 - Image Rollover
 - Image Preview
 - Image Slide Show
- Date object
 - Digital Clock

THANK YOU