

# Lecture 16

## History, Navigator, Screen and Form Objects

**Mr. Mubashir Ali**

**Lecturer (Dept. of Computer Science)**

[dr.mubashirali1@gmail.com](mailto:dr.mubashirali1@gmail.com)

# Summary of the previous lecture

- **Controlling the background dynamically**
  - **Bgcolor**
  - **Text**
  - **background**
- **Working with images**
  - **Image rollover**
  - **Image preview**
  - **Image slide show**
- **Date object**
  - **Digital clock**

# Outline

- **History object**
- **Navigator object**
- **Screen object**
- **Form object**

# 1. The History object

- The **history object** contains the **URLs** visited by the user (within a browser window)
- The history object is **part** of the **window object** and is accessed through the **window.history** property
- Used to move **forward and backward** through the visitor's browsing history

# 1. The History object...

- **History object properties:**
- **Length:** Returns the number of URLs in the history list
- **History object methods:**
- **back():** Loads the previous URL in the history list
- **forward():** Loads the next URL in the history list
- **go():** Loads a specific URL from the history list

# 1. The History object...

```
history - Notepad
File Edit Format View Help
<html>
<head>
<title>History Object</title>
<script language="javascript">
document.write(history.length)
function goBack()
{
window.history.back()
}
function goForward()
{
window.history.forward()
}
</script>
</head>
```

Writes history length on webpage

Go back function

Go forward functions

# 1. The History object...

```
history - Notepad
File Edit Format View Help
window.history.forward()
}
</script>
</head>

<body>
<h1>This is the first Page</h1>
<a href="history1.html">Go to next page</a><br>
<input type="button" value="Go Back!"
onclick="goBack()" >
<input type="button" value="Go Forward!"
onclick="goForward()" >
</body>
</html>
```

Body contents

Call to go back

Call to goforward

## 2. The Navigator object

- The **navigator** object contains information about the **browser**
- provides **several properties** that assist in the **detection of various elements** of the visitor's **browser and environment**



## 2. The Navigator object...

- **Navigator object properties:**
  - **appName:** Returns the **code name** of the browser
  - **appVersion:** Returns the **name** of the browser
  - **appCodeName:** Returns the **version information** of the browser
- **Navigator object methods:**
  - **javaEnabled():** Specifies whether or not the browser has **Java enabled**

## 2.1 Detecting Users browser

- Used to write **browser specific** code
- Can also be used to **restrict users** to use a specific browser

# 2.1 Detecting Users browser...

```
navigator - Notepad
File Edit Format View Help
<HTML>
<HEAD>
<TITLE>Browser Detection</TITLE>
</HEAD>
<BODY BGCOLOR=white>
<SCRIPT LANGUAGE="JavaScript">
var browsername= navigator.appName
var browserversion = navigator.appVersion
document.write("You are using"+ browsername +
"version " + browserversion)
</SCRIPT>
</BODY>
</HTML>
```

**Finding browser name**

**Finding browser version**

**Writing browser information**

# 3. The Screen object

- The screen object contains information about the **visitor's screen**
- You might need this information to determine **which** images to display or **how large** the page can be

# 3. The Screen object...

- The screen object properties:
- **availHeight**: Returns the height of the screen (excluding the Windows Taskbar)
- **availWidth**: Returns the width of the screen (excluding the Windows Taskbar)
- **colorDepth**: Returns the bit depth of the color palette for displaying images

## 3. The Screen object...

- **height:** Returns the total height of the screen
- **width:** Returns the total width of the screen

# 3. The Screen object...

```
screen - Notepad
File Edit Format View Help
<html>
<head>
<title>Screen Object</title>
<script language="javascript">
document.write("Available Height:"
+ screen.availHeight)
document.write("<br>Available width:"
+ screen.availWidth)
document.write("<br>Height:", screen.height)
document.write("<br>width:", screen.width)
</script>
</head>

<body>
</body>
```

**Finding available height**

**Finding available height**

**Finding actual height**

**Finding actual width**

# 4. Form Object

- The **Form object** represents an HTML form
- For **each** `<form>` tag in an HTML document, a **Form object** is created
- The browser creates a **'forms array'** which keeps the number of **form objects** in the HTML program
- The **first form** object in the HTML file being held as array index `[0]`, the second as index `[1]` and so on



## 4. Form Object...

- The **'forms array'** also holds information about each **element** used within `<FORM>` and `</FORM>` tags
- **elements** array keeps information about form elements

# 4. Form Object...

**<body>**

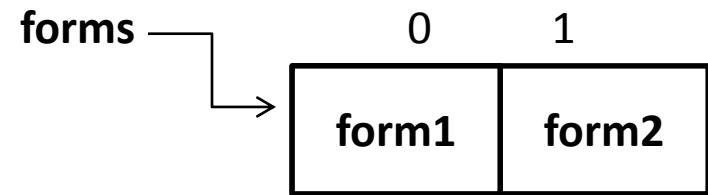
**<form name="form1">**

**</form>**

**<form name="form2">**

**</form>**

**</body>**



# 4.1 Accessing form elements

**<body>**

**<form name="form1">**

**<input type="text" name="name">**

**<input type="text" name="email">**

**</form>**

**</body>**

**document.forms[0].name.value**

**or**

**document.form1.elements[0].value**

**document.forms[0].email.value**

**or**

**document.form1.elements[1].value**

## 4.2 Setting form elements

**<body>**

**<form name="form1">**

**<input type="text" name="name">**

**<input type="text" name="email">**

**</form>**

**</body>**

**document.forms[0].name.value = "ali"**

**or**

**document.form1.element[0].value = "ali"**

**document.forms[0].email.value = "ali@gmail.com"**

**or**

**document.form1.elements[1].value = "ali@gmail.com"**

# 4.2 Setting form elements

Untitled Document - Google Chrome  
file:///C:/Users/Administrator/Desktop/Web%20Engineering%20sp14/Lectures/Lecture%2026%20(Working%20with%20form)/billing.html

Billing System	
Previous Reading	<input type="text"/>
Current Reading	<input type="text"/>
Units Consumed	<input type="text"/>
Your Bill	<input type="text"/>
<input type="submit" value="Submit"/>	

Previous reading

Current reading

Consumed=current - previous

Bill= consumed\*5

# 4.2 Setting form elements

```
function ver()  
{  
var cur,per,ut,bill  
cur=form1.elements[1].value  
per=form1.elements[0].value  
if(cur<per)  
{  
alert("Current reading can not be less than the prev  
}  
else  
{  
form1.elements[2].value=cur-per  
form1.elements[3].value=form1.elements[2].value*5  
}  
}
```

**Variables declared**

**Getting the value of elements**

**Setting values**

## 4.2 Setting form elements

```
billing - Notepad
File Edit Format View Help
</label>
</span></td>
</tr>
<tr>
<td><span class="style5">Current Reading </span></td>
<td><span class="style5">
<label>
<input type="text" name="textfield2" onchange="ver()" />
</label>
</span></td>
</tr>
<tr>
<td><span class="style5">Units Consumed </span></td>
<td><span class="style5">
<label>
<input type="text" name="textfield3" onfocus="this.blur()"
</label>
```

**Current reading field**

**Calling the function**

# 4.3 Validating form data

The image shows a screenshot of a web browser window displaying a form titled "emptyvalues.html". The form contains four input fields and a submit button. Red arrows point to the "Your Name" and "Your Email" fields, with the text "Must fill" and "Valid email" respectively. The "Your Gender" field has two radio buttons labeled "Male" and "Female".

Your Name	<input type="text"/>	← <b>Must fill</b>
Your Email	<input type="text"/>	← <b>Valid email</b>
Your Age	<input type="text"/>	
Your Gender	<input type="radio"/> Male <input type="radio"/> Female	
<input type="submit" value="Submit"/>		



# 4.3 Validating form data...

```
emptyvalues - Notepad
File Edit Format View Help
<script language="javascript">
function validateForm()
{
var x=document.form1.elements[0].value;
if (x==null || x=="")
{
alert("First name must be filled out");
return false;
}
}
</script>
</head>
<body>
<form name="form1" onsubmit="validateForm()" >
  <table width="364" border="1" align="center">
```

Getting first element value

Checking for emptiness

Calling the function

# 4.3 Validating form data...

```
email - Notepad
File Edit Format View Help
<title>Email</title>
<script language="javascript">
function validateEmail()
{
var x=document.form1.elements['email'].value;
var atpos=x.indexOf("@");
var dotpos=x.lastIndexOf(".");
if (atpos<1 || dotpos<atpos+2 || dotpos+2>=x.length)
{
alert("Not a valid e-mail address");
return false;
}
}
</script>
</head>
```

Getting value of email

Finding position of '@'

Finding last position of '.'

Checking conditions

# Summary

- **The history object**
- **The navigator object**
- **The screen object**
- **The form object**

# References

- **Chapter 11.** Beginning HTML, XHTML, CSS, and JavaScript, by Jon Duckett, Wiley Publishing; 2009, ISBN: 978-0-470-54070-1.
- **Chapter 6,11.** Learn JavaScript, by Chuck Easttom, Wordware Publishing; 2002, ISBN 1-55622-856-2