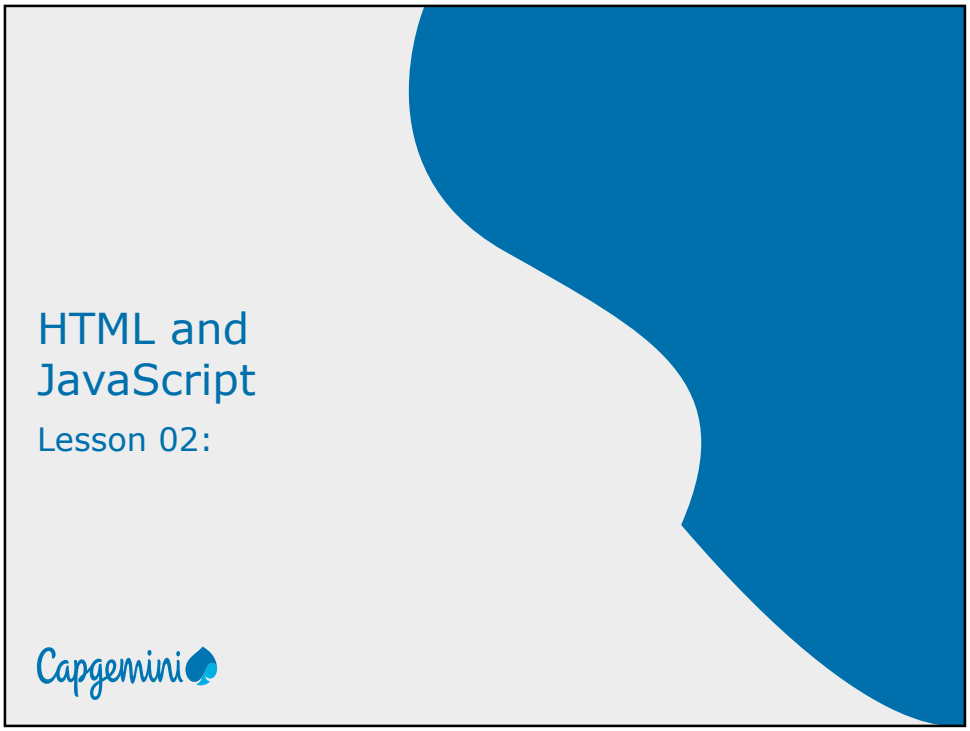
Instructor Notes:



Instructor Notes:

Lesson Objectives

- To understand the following concepts
  - HTML form element
  - HTML 5 new form elements (Number, Date and Email)
  - HTML 5 validations
  - DOM objects (Document and Form)
  - Event handling in JavaScript

Instructor Notes:

2.1 HTML form element

HTML Forms for User Input

- HTML forms are used to accept user inputs and then submit data for processing.
- A *form* is an area that contains form elements.
- Types of elements which can be included in a form are
  - Label
  - Single line text field
  - Password field
  - Text area
  - Drop down menu
  - Radio button
  - Checkbox
  - File selector box, etc..

Forms:

HTML forms are used to accept of user input.

A form is an area that contains form elements.

Form elements are elements that allow users to enter information (text fields, text area fields, drop-down menus, radio buttons, checkboxes, etc.) in a form.

Define a form with the <form> tag.</form>

Instructor Notes:

2.1 HTML form element

HTML Forms for User Input

- User input forms are created using <form> tag.
- Syntax:

```
<form method="get/post" action="URL"
  enctype="Encryption Type">
    Field definitions
</form>
```

- **action:** the URL of the script
- **method:** the HTTP request method to use, sometimes GET, but usually POST
- **enctype:** Specifies how the data is to be encoded.

Basic syntax for the <form> tag is:

<form method="Get or Post" action="URL" enctype="type">

Field definitions

</form>

<form> tag tells a browser that there is a fill-in-the-blank form in this HTML document.

Method:

method attribute states the method to use when you send the form to the server. Two acceptable methods are GET and POST.

GET sends information entered in the form to the server at the end of the URL.

POST sends information entered in the form to the server as a data body/document.

action attribute:

Gives the address of the script that processes the form.

Defines the name of the file to send the content to. File defined here typically does something with the received input.

enctype attribute:

Specifies how the data is to be encoded.

Applies only if you use the POST method. There is only one possible value, the default value "application/w-www/form-urlencoded".

Instructor Notes:

2.1 HTML form element

HTML Forms for User Input

- Some more attributes which can be used in <form> tag are

Attribute Name	Attribute Value	Description
name	Form name as a string	Mentions the name of a form.
autocomplete	On, off	Specifies whether a form should have autocomplete on or off
target	_blank, _self, _parent, _top	Specifies where to display the response that is received after submitting the form
novalidate	novalidate	Specifies that the form should not be validated during form submission.

Instructor Notes:

2.1 HTML form element

HTML Form Elements

- <input> element is the most used form tag.
- An <input> tag includes the following attributes
  - name: Name of the field which is required to send data(Key/Value pair) during form submission
  - id: A unique identified of the field
  - value: Sets a default value of the field
  - maxlength: Specifies the maximum number of characters allowed in an <input> element
  - readonly: Specifies that an input field is read-only
  - size: Specifies the width, in characters, of an <input> element
  - Disabled: specified that an input element should be displayed.
  - type attribute of <input> tag specifies the field type

Input:

The most used form tag is the <input> tag. Input type is specified with the type attribute. Type attributes are:

Text

Password

Hidden

Radio

Checkbox

File

Button

Submit/reset

Instructor Notes:

2.1 HTML form element

Text-related Elements

- Text related elements can be created as shown below:

Code	Element
<input type="text">	Single line text box
<input type="password">	Password field
<input type="hidden">	Hidden field

- Multiple line text input control
  - If input exceeds more than one line, then create Multi-line input control using HTML <textarea> tag
  - Syntax:

    <textarea rows=" " cols=" " name=" ">
    - Rows : Number of rows of text area box
    - Cols: Number of columns of text area box
    - Name: name of the element

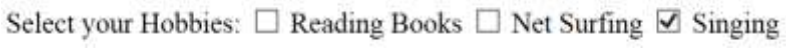
Instructor Notes:

2.1 HTML form element

Checkbox Element

- If more than one option is required to be selected from multiple options, then create checkbox as shown below:
  - <input type="checkbox">
  - Use checked attribute for selecting any checkbox to be selected by default
- Example:

```
<input type="checkbox" name="hobbies" value="Reading Books"> Reading Books
<input type="checkbox" name="hobbies" value="Net Surfing"> Net Surfing
<input type="checkbox" name="hobbies" value="Singing" checked>Singing
```



Instructor Notes:

2.1 HTML form element

Radio Button

- If only one option is required to be selected from multiple options, then create radio button as shown below:
  - <input type="radio">
  - Use checked attribute for making a radio button to be selected by default
- Example:

```
<input type="radio" name="sector" value="Public"> Public
<input type="radio" name="sector" value="Private"> Private
```

Select your sector in which you are working: ○ Public ○ Private

Instructor Notes:

2.1 HTML form element

Drop down list

- Drop down list allow the user to select one or more values from a pre-determined options
- Tags for creating drop down list with options are:
  - <select> : Creates drop down list
  - <option>: Defines an option in a select list.

Tag name	Attribute	Description
<select>	Name	Defines a name for the drop down list
	Size	Defines the number of visible options in a drop down list
	Multiple	Allow to select multiple options at once
	Disabled	Disable drop down list
<option>	value	Specifies the value to be sent to a server
	Selected	Makes option to be selected by default

Instructor Notes:

2.1 HTML form element

Drop down list - Example

```
<!DOCTYPE html>
<html>
<body>Select a country:
      <select name="country">
          <option value="Germany">Germany</option>
          <option value="India" selected>India</option>
          <option value="China">China</option>
          <option value="Japan" >Japan</option>
      </select>
</body>
</html>
```

Instructor Notes:

2.1 HTML form element

File Upload

- File upload will allow the user to upload a file from the desktop to an application in browser.
- The below code is used to define a file-select field and a "Browse..." button (for file uploads):
  - <input type="file"/>

Attribute	Description
Name	Defines a name for the file upload dialog box
Disabled	Disable element
Accept	Specify MIME type to describe the file type which accepts by a server

- In <form> tag, use enctype="multipart/form-data" if file need to be uploaded using file selector form element.

Instructor Notes:

2.1 HTML form element

File Upload- Example

```
<!DOCTYPE html>
<html>
<body>
      <form method="post" action="success.html"
        enctype="multipart/formdata">
        Select a photo to upload:
        <input type="file" name="photo"/>
       </form>
</body>
</html>
```

Instructor Notes:

2.1 HTML form element

Button

- Different types of button which is possible to be created in HTML5 are as shown below:

Field type	Element	
<input type="button">	Button	A clickable button, that activates a JavaScript when it is clicked
<input type="submit">	Submit button	Defines a button for submitting a form
<input type="reset">	Reset button	Define a reset button (resets all form values to default values)

Instructor Notes:

2.2 : HTML 5 new form elements

Number

- Up and down button provided to increase and decrease the value.
- Min and max parameters provided to limit the values.
- Browser will treat it as simple textfield if it doesn't support this type.

```
<input id="movie" type="number" value="0"/>
```

- Syntax is

```
< input id="user_lic" type="number" min="5" max="30" step="5" value =""/>
```

- Limiting the values for this field…

Number:

Asking for a number is trickier than asking for an email address or web address.

First of all, numbers are more complicated than you might think. You don't often

ask for "just a number." It's more likely that you'll ask for a number in a

particular range. You may only want certain kinds of numbers within that range-

maybe whole numbers but not fractions or decimals.

Example:

<input type="number" min="0" max="10" step="2" value="6">

Let's take that one attribute at a time.

1. type="number" means that this is a number field.
2. min="0" specifies the minimum acceptable value for this field.
3. max="10" is the maximum acceptable value.

4. step="2", combined with the min value, defines the acceptable numbers in the range: 0, 2, 4, and so on, up to the maxvalue.
5. value="6" is the default value.

Instructor Notes:

2.2 : HTML 5 new form elements

Date

- Date
  - Important and mostly used element
  - Simple to implement
  - Before HTML5, programmers used to write lines of JavaScript code for date picker
  - Input type for date:- date, week, month, time, datetime (gives UTC time), datetime-local (local time)
- Syntax is

```
<input id="meeting" type="date" value=""/>
```

Date:

HTML 4 did not include a date picker control. JavaScript frameworks have picked up

the slack (Dojo, jQuery UI, YUI, Closure Library), but of course each of these

solutions requires "buying into" the framework on which the date picker is built.

HTML5 finally defines a way to include a native date picker control without having to

script it yourself. In fact, it defines six input types: date, month, week, time, date +

time, and date + time - timezone.

So far, support is… sparse.

Instructor Notes:

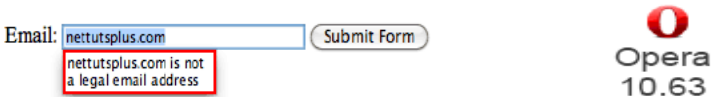
2.2 : HTML 5 new form elements

Email

- Email - This field is used to check whether the string entered by the user is valid email id or not.
- Syntax is –

```
<input id="email" name="email" type="email" />
```

- Browser's that don't support this field will treat this as a simple text field
- This is how it looks like on form



Email:

The email type is used for input fields that should contain an e-mail address. The

value of the email field is automatically validated when the form is submitted.

Example

E-mail: <input type="email" name="user_email" />

Safari on the iPhone recognizes the email input type, and changes the on-screen

keyboard to match it (adds @ and .com options).

Instructor Notes:

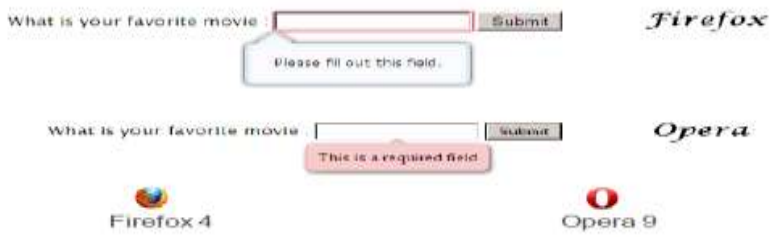
2.3 HTML 5 validations

Required

- Required - A field with "required " attribute must be filled in with value before submission of a form
- Syntax is –

```
<input name="name" type="text" required />
```

- The picture below shows us how Firefox and Opera prompt user to fill in value if a "Required field" is left blank upon submission



Required:

If the **required** attribute is present, then the field must contain a value when the form

is submitted. This informs the (*HTML5*-aware) web browser that the field is to be considered mandatory. Different browsers may mark the input box in some way (Firefox 4 Beta adds a red box-shadow by default), display a warning (Opera) or even prevent the form from being submitted if this field has no value. Hopefully these

behaviours will converge in future releases.

Here's an example of an input field for a required email address that ensures that the

field has a value and that the value is a valid email address.

Example:

<input type="email" id="email_addr" name="email_addr" required />

Instructor Notes:

2.3 HTML 5 validations

Pattern

- Pattern - A value filled in the field must be checked against the regular expression specified in pattern attribute.
- Syntax is –

```
Pincode: <input type="text" name="pin_code" pattern="[0-9]{6}" title="666666">
```

- The picture below shows us how browser prompt user to fill in valid value if a "field with Pattern attribute" is filled with invalid value upon submission

Pattern:

The pattern attribute specifies a regular expression that the <input> element's value is checked against.

Note: The pattern attribute works with the following input types: text, date, search, url, tel, email, and password.

Tip: Use the global title attribute to describe the pattern to help the user.

Example:

```
<input type="text" name="country_code"
  pattern="[A-Za-z]{3}" title="Three letter country code">
<input type="password" name="pw" pattern=".{6,}" title="Six or more
  characters">
<input type="password" name="pw" pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-
  Z]).{8,}" title="Must contain at least one number and one uppercase and
  lowercase letter, and at least 8 or more characters">
<input type="email" name="email" pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-
  z]{2,3}$">
<input type="url" name="website" pattern="https?://.+" title="Include
  http://">
```

Instructo Refer **http://www.html5pattern.com** to explore some more patterns.

Instructor Notes:

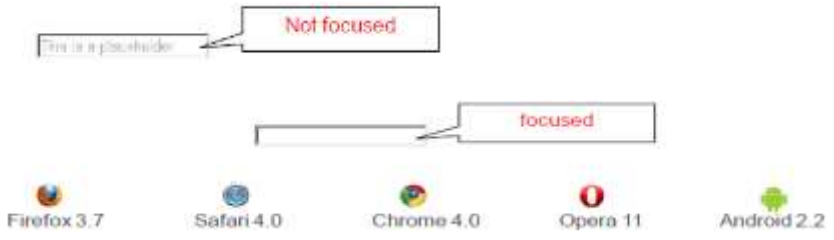
2.3 : HTML 5 validations

Placeholder

- Place Holder - A placeholder is a textbox that hold a text in lighter shade when there is no value and not focused
- Syntax is –

```
<input id="first_name" placeholder="This is a placeholder">
```

- This is how place holder looks like on supporting browser
- Once the textbox gets focus, the text goes off and you shall input your own text



Place Holder:

The first improvement HTML5 brings to web forms is the ability to set placeholder text in an input field. Placeholder text is displayed inside the input field as long as the field is empty and not focused. As soon as you click on (or tab to) the input field, the placeholder text disappears.

Here's how you can include placeholder text in your own web forms:

```
<form>
  <input name="name" placeholder="Enter your name">
  <input type="submit" value="Search">
</form>
```

Browser's that don't support placeholder attribute will simply ignore it. But if you want to make it work in other browsers, you can use some **JavaScript** to create The same behavior. There is an excellent **jQuery plugin** called HTML5 Placeholder Plugin that will go through all input fields with placeholders attached to them and make them work in all browsers.

Instructor Notes:

2.4 DOM objects (Document and Form)

Working With Document Object

- Container for all HTML HEAD and BODY objects associated within tags
- Provides access to page elements from your script
  - This includes form, link, anchor, as well as global Document properties such as background and foreground colors

Document object is part of the Window object. It is used to access all elements in a page. It provides access to the elements in an HTML page from within the script.

This includes the properties of every form, link and anchor (and, where applicable, any sub-elements), as well as global document properties such as background and foreground colors.

Instructor Notes:

2.4 DOM objects (Document and Form)

Document Object Properties

- document.anchors
- document.forms
- document.links
- document.title

document.anchors - Returns a collection of all <a> elements in the document that have a name attribute

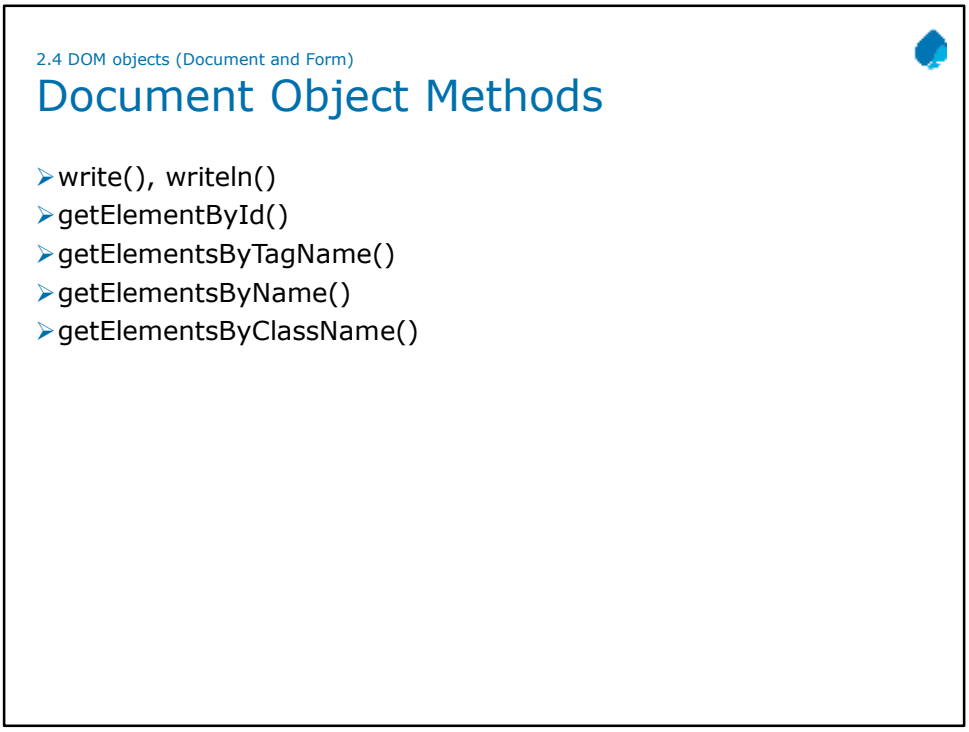
document.forms - An HTMLCollection Object, representing all <form> elements in the document. The elements in the collection are sorted as they appear in the source code.

document.links - Returns a collection of all <a> and <area> elements in the document that have a href attribute

document.title - Sets or returns the title of the document

Property	Description
alinkColor vlinkColor bgColor fgColor linkColor	Get and set the properties of document – activated link, visited link, background color, foreground color (text) and hyperlink color.
anchors[], forms[], links[]	These properties retrieve array of values respectively as present in the document object
title	Gets the title of the document which occurs between the TITLE tags.

Instructor Notes:



write() - Writes HTML expressions or JavaScript code to a document
writeln() - Same as write(), but adds a newline character after each statement
getElementById() - Returns the element that has the ID attribute with the specified value
getElementsByTagName() - Returns a NodeList containing all elements with the specified tag name
getElementsByName() - Returns a NodeList containing all elements with a specified name
getElementsByClassName() - Returns a NodeList containing all elements with the specified class name

Property	Description
write("string1", …) writeln("string1", ..)	Both of these methods send text to a document for display in its window. The only difference between the two methods is that *document.writeln()* appends a carriage return to the end of the string it sends to the document (you must still write a to insert a line break).
getElementById("#para1")	This method locates the element whose id has been passed. The text within this element can then be accessed using properties innerHTML or innerText
getElementsByTagName("p")	This method locates all the elements which match the tagname passed. Each element of this type of tag can then be accessed in an array like manner
getElementsByName()	This method locates all the elements which match the name passed. Same name to many elements is usually given for radio buttons.
getElementsByClass()	This method locates all the elements which match the class name passed.

Instructor Notes:

2.4 DOM objects (Document and Form)

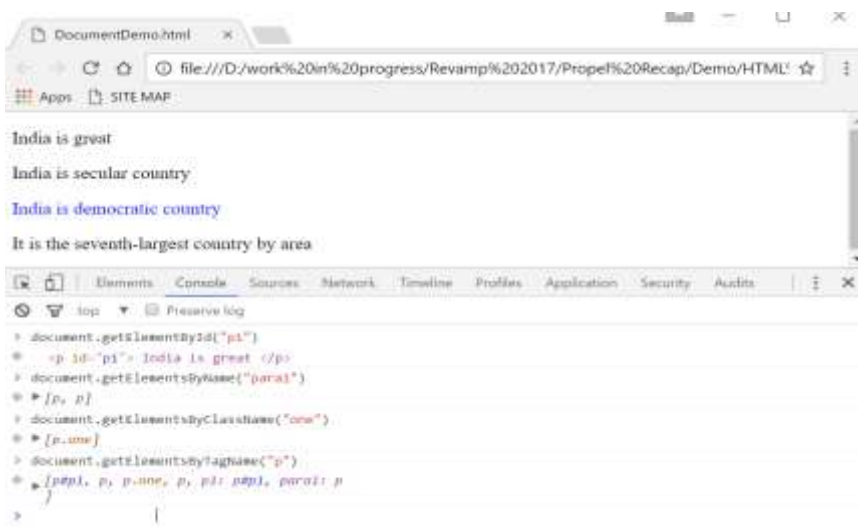
Document Object Methods

```
<!DOCTYPE html>
<html>
<style>
	.one{color: blue}
</style>
	<body>
		<p id="p1"> India is great </p>
		<p name="para1"> India is secular country </p>
		<p class="one"> India is democratic country</p>
		<p name="para1"> It is the seventh-largest country by area</p>
	</body>
</html>
```

Instructor Notes:

2.4 DOM objects (Document and Form)

Document Object Methods



Instructor Notes:

2.4 DOM objects (Document and Form)

Handlers Form Object

Properties	Methods	Event Handlers
action	reset()	onReset
elements[ ]	submit()	onSubmit
enctype		
length		
method		
name		
target		

Working with Form Objects: Form Object Properties:

A form element provides the only way that users can enter textual information or make a selection from a predetermined set of choices, whether those choices appear in the form of an on/off checkbox, one of a set of mutually exclusive radio buttons, or a selection from a list.

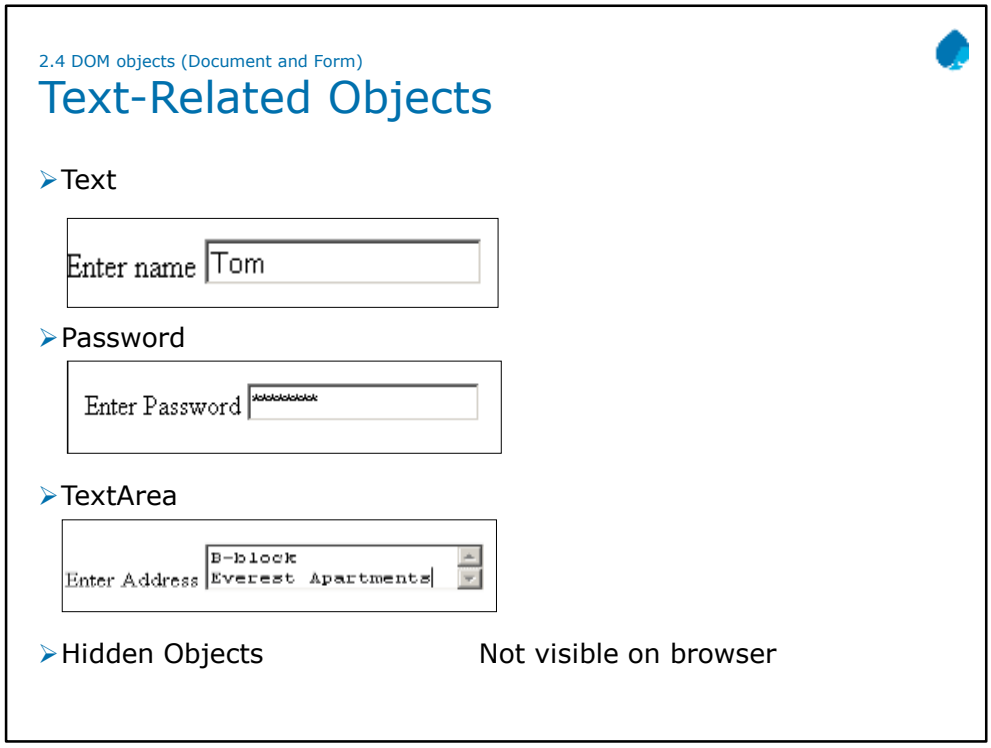
Property/ Method/ Events	Description
action	This property is the same as the value you assign to the ACTION attribute of a <FORM> tag. The value is typically a URL on the server where queries or postings are sent for submission.
elements[]	Returns an array of elements. It includes all the user interface elements defined for a form: text fields, buttons, radio buttons, checkboxes, selection lists, and more.
encoding	You can define a form to alert a server that the data being submitted is in a MIME type. This property reflects the setting of the ENCTYPE attribute in the form definition. The default value is an empty string.
method	A form's method property is either the GET or POST values assigned to the METHOD attribute in a <FORM> tag.

Instructor Notes:

name	Assigning a name to a form via the NAME attribute is optional but highly recommended when your scripts need to reference a form or its elements. This attribute's value is retrievable as the name property of a form.
target	The purpose of the TARGET attribute of a <FORM> definition is to enable you to specify where the output from the server's query should be displayed. The value of the target property is the name of the window or frame.
reset()	If you want to clear the form i.e. return the form elements to its default settings using script control, you must do so by invoking the reset() method for the form.
submit()	Invoking this method is almost the same as a user clicking a form's Submit button
onReset	Immediately before a Reset button returns a form to its default settings, JavaScript sends a reset event to the form. By including an onReset event handler in the form definition, you can trap that event before the reset takes place.
onSubmit	When you define an onSubmit handler as an attribute of a <FORM> definition, JavaScript sends the submit event to the form just before it dashes off the data to the server. Therefore, any script or function that is the parameter of the onSubmit attribute executes before the data is actually submitted. Note that this event handler fires only in response to a genuine Submit-style button, and not from a form.submit() method.

Table 9.1 Form object properties, methods and event handlers

Instructor Notes:



Text-Related Objects:

Text Objects : The text object is the primary medium for capturing user-entered text.

Password Object: A password-style field looks like a text object, but when the user types something into the field, only asterisks or bullets (depending on your operating system) appears in the field.

Textarea Object: A textarea object closely resembles a text object, except for attributes that define its physical appearance on the page.

Hidden object: A hidden object is a simple string holder within a form object whose contents are not visible to the user of your Web page. With no methods or event handlers, the hidden object's value to your scripting is as a delivery vehicle for strings that your scripts need for reference values or other hard-wired data.

Instructor Notes:

2.4 DOM objects (Document and Form)

Text-Related Objects (Contd..)

Properties	Methods	Event Handlers
defaultValue	blur()	OnBlur
name	focus()	OnChange
type	select()	OnFocus
value		

The properties, methods and event handlers are same for text object, text area and Password. For hidden object the properties are same but no methods and event handlers are associated with this object.

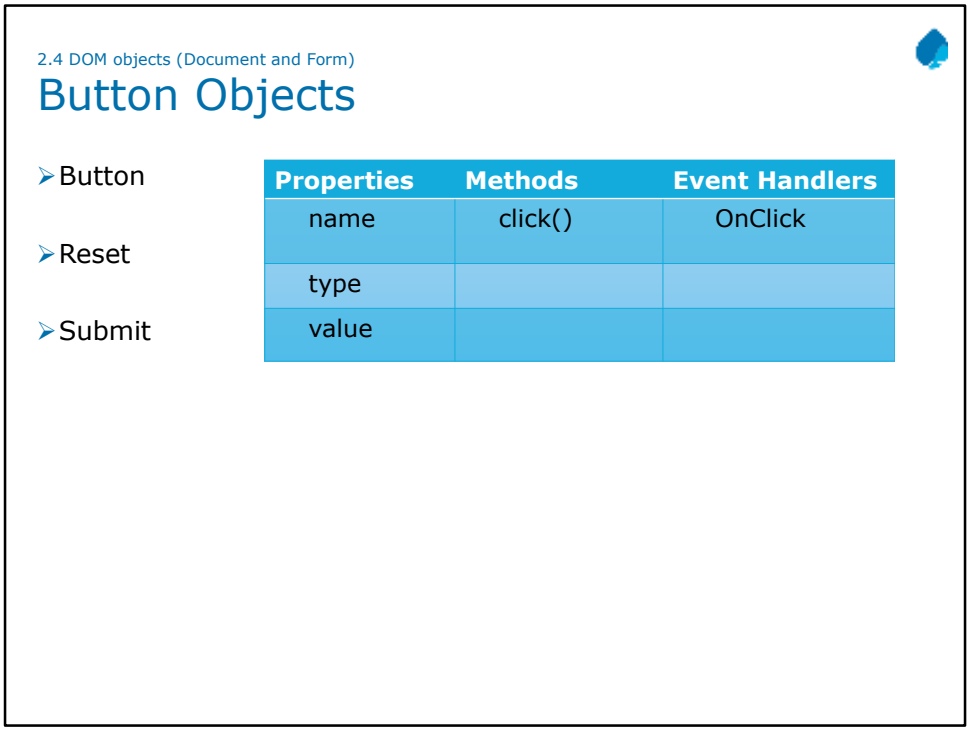
Property/ Events/ Methods	Description
defaultValue	Specifies or returns a defaultValue for a text related objects.
name	This property can be used to reference the text object in the script.
type	Returns the type of text related object
value	A reference to an object's value property returns the string currently displayed in the field.

Instructor Notes:

blur()	blur() deselects whatever may be selected in the field, and the text insertion pointer leaves the field. The pointer does not proceed to the next field in tabbing order, as it does if you perform a blur by tabbing out of the field manually.
focus()	For a text object, having focus means that the text insertion pointer is flashing in that text object's field. The cursor usually appears at the beginning of the text. To prepare a field for entry to remove the existing text, use both the focus() and select() methods.
select()	Selecting a field under script control means selecting all text within the text object.
onBlur onFocus	The onBlur event is fired when a text field loses focus because user has clicked somewhere outside the text field. The onFocus event is fired when the user clicks inside the text field.
onChange	This event is fired when the user changes the value in the text field.

Refer to Appendix for more event handlers

Instructor Notes:



Button Objects: Button, Submit and Reset

Property	Description
name	You may need to retrieve this property in a general-purpose function handler called by multiple buttons in a document. The function can test for a button name and perform the necessary statements for that button.
type	The precise value of the type property echoes the setting of the TYPE attribute of the <INPUT> tag that defined the object: button; submit; or reset.
value	A button's visible label is determined by the VALUE property.
click()	A button's click() method should replicate, via scripting, the human action of clicking that button.
onClick	Virtually all button action takes place in response to the onClick event handler. A click is defined as a press and release of the mouse button while the screen pointer rests atop the button.

Instructor Notes:

2.4 DOM objects (Document and Form)

Check Box And Radio Objects

- Checkbox
- Radio

Properties	Methods	Event Handlers
checked	click()	OnClick
defaultChecked		
name		
type		
value		

Checkbox object:

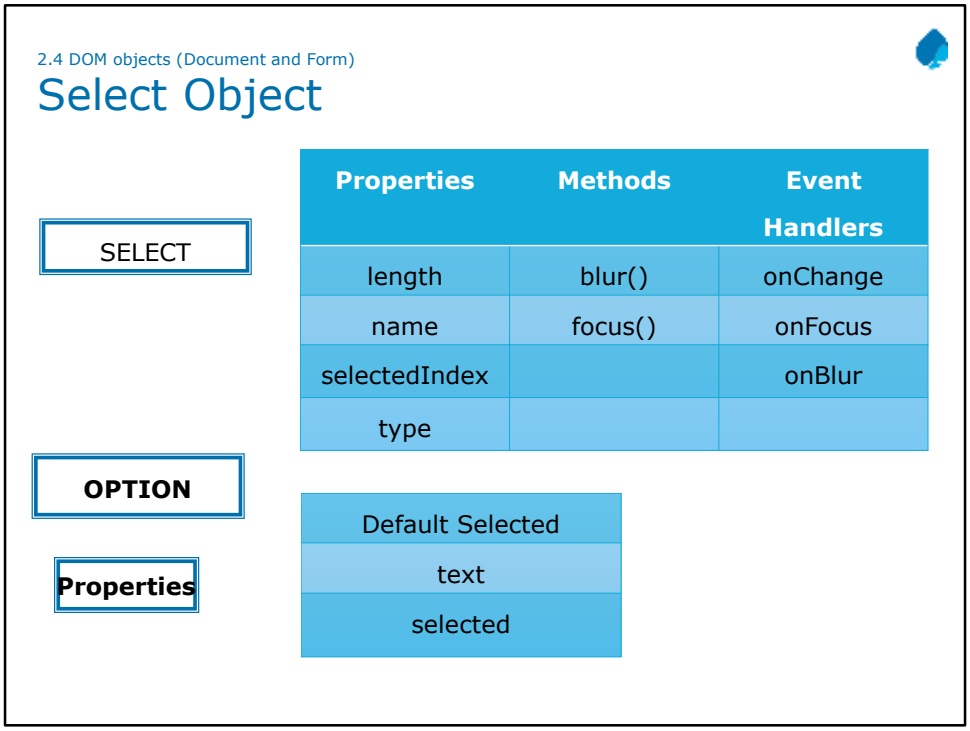
Property/ Events/ Methods	Description
checked	The simplest property of a checkbox gets or lets you set whether or not a checkbox is checked. The value is true for a checked box and false for an unchecked box. Only one radio button in a group can be highlighted checked) at a time. That one button's checked property is set to true, whereas all others in the group are set to false.
defaultChecked	If you add the CHECKED attribute to the <INPUT> definition for a checkbox or radio button, the defaultChecked property for that object is true; otherwise, false.
name	The name property allows user to access name for the checkbox or radio button through script.
type	Use the type property to help you identify a checkbox object or a radio button object from an unknown group of form elements.

Instructor Notes:

value	A checkbox or radio button object's value property is a string of any text you want to associate with. Either you can set or retrieve the value
click()	The intention of the click() method is to enact, via script, the physical act of checking a checkbox or selecting a radio button
onClick	The onClick event of checkboxes or radiobuttons should be handled when through script you need to handle a specific task

Table 9.4 Checkbox object properties, methods and event handlers

Instructor Notes:

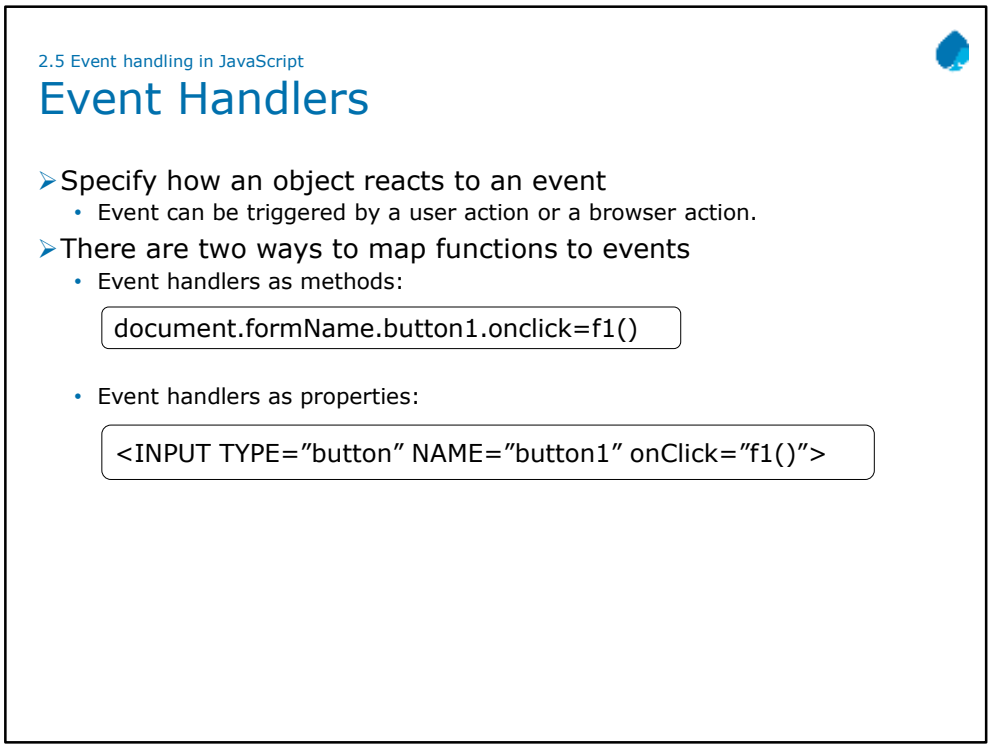


Property/ Methods/ Events	Description
length	Returns the number of items available in the list. A select object with three choices in it has a length property of 3.
Name	A select object's name property is the string you assign to the object by way of its NAME attribute in the object's <SELECT> tag which can be
selectedIndex	When a user clicks on a choice in a selection list, the selectedIndex property changes to a number corresponding to that item in the list.
type	Use the type property to help you identify a select object from an unknown group of form elements.
blur() focus()	Your scripts can bring focus to a select object by invoking the object's focus() method. To remove focus from an object, invoke its blur() method. These methods work identically with their counterparts in the text object.
onChange	As a user clicks on a new choice in a select object, the object receives a change event that can be captured by the onChange event handler.

Instructor Notes:

options[index]. defaultSelected	If your select object definition includes one option whose SELECTED attribute is included, that option's defaultSelected property is set to true. The defaultSelected property for all other options is false.
options[index]. selected	To determine which option a user has selected from a list than looping through all options and examining the selected property this property can be used.
options[index]. text	The text property of an option is the text of the item as it appears in the list.

Instructor Notes:



Object Event Handlers

Event handlers specify how an object reacts to an event, whether the event is triggered by a user action (for example, a button click) or a browser action (for example, the completion of a document load). Event Handlers can be specified as methods or they can be specified using attributes in tags.

Instructor Notes:

2.5 Event handling in JavaScript

Event Handlers

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

Instructor Notes:

Demo

- Demo on:
  - SampleForm.html (html File)
  - Validation.js (JavaScript file)

Instructor Notes:

Case Study

- Create a **prob3.html** page as shown in the below figure. The page should be submitted on clicking the **Submit** button when all the form fields are properly validated.

Instructor Notes:

Case Study

- None of the fields should be empty (Use HTML 5 required attributes)
- Name field should be between 3 to 10 characters (Use HTML 5 pattern attributes)
- For Date of Birth format use HTML 5 date control
- Phone Number should be in xxx-xxxx-xxxx format (Use HTML 5 pattern attributes)
- Email ID should be valid. (Use HTML 5 email control)
- Use placeholder attribute to denote the expected pattern format to the user
- Based on graduation level selected, qualification need to be populated automatically. For an example, if graduation level selected is UG, then qualification should be B.Sc., B.A, B.Com, etc... If graduation level selected is PG, then qualification should be M.A, M.Tech, MCA, MBA, etc.... (Call function on onChange event)
- Calculate age of the person and display all the details in a new popup window when "Submit" button is clicked. Details should be printed in the specified format as given below:
  - Name:
  - Age:
  - Phone Number:
  - Email:
  - Graduation Level:
  - Qualification:

  Display the appropriate error message when the validation condition fails.

Hint: Use below code for age calculation:

```
today = new Date();
dob=document.frm.dob.value;
var age = today.getYear()-dob.getYear();
```

Instructor Notes:

Summary

- In this lesson, you have learnt about:
  - HTML form element
  - HTML 5 new form elements (Number, Date and Email)
  - HTML 5 validations
  - DOM objects (Document and Form)
  - Event handling in JavaScript

Instructor Notes:

Question 1: A,B,C
Question 2: B

Review Questions

- Question 1: A _____ is a textbox that hold a text in lighter shade when there is no value and not focused
- Question 2: Radio Buttons are used when you want the user to select:
  - Option 1: one of a limited number of choices.
  - Option 2: one or more options of a limited number of choices.
  - Option 3: many of unlimited number of choices.
- Question 3: METHOD attribute states the method to use when you send the form to the server.
  - True/ False
- Question 4:The _____ attribute of form specifies how the data is to be encoded.