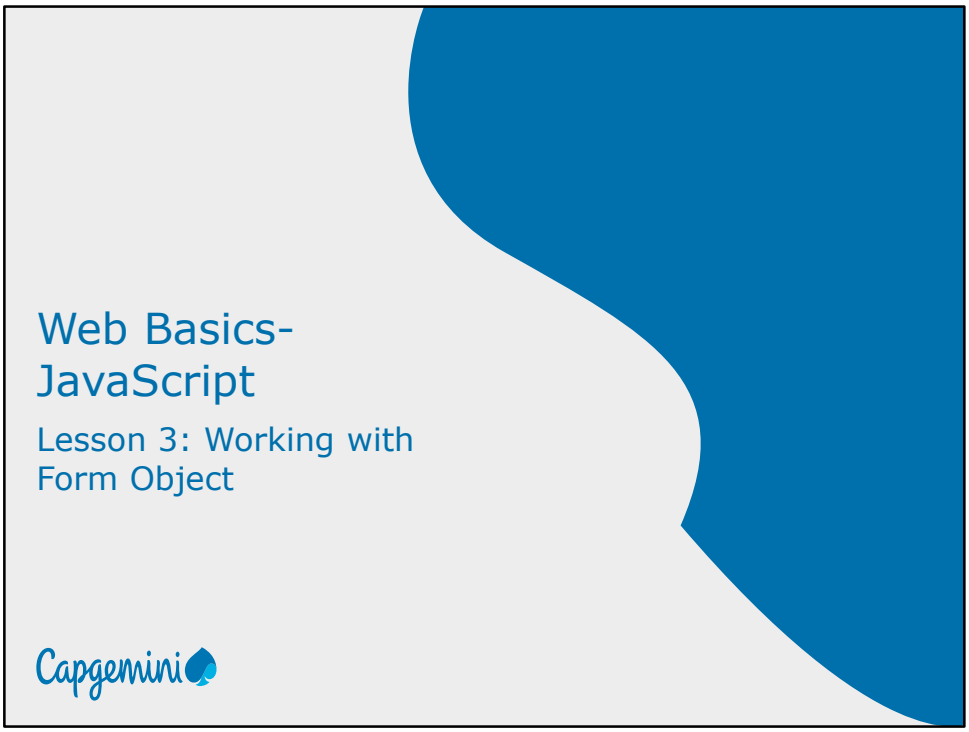
Instructor Notes:



Instructor Notes:

Lesson Objectives

- To understand the following topics:
  - Form Object Properties, Methods & Event Handlers
  - Text-Related Objects
  - Button Objects
  - Check Box and Radio Objects
  - Select Objects

Instructor Notes:

7.1: Form Object Properties, Methods and Event

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.

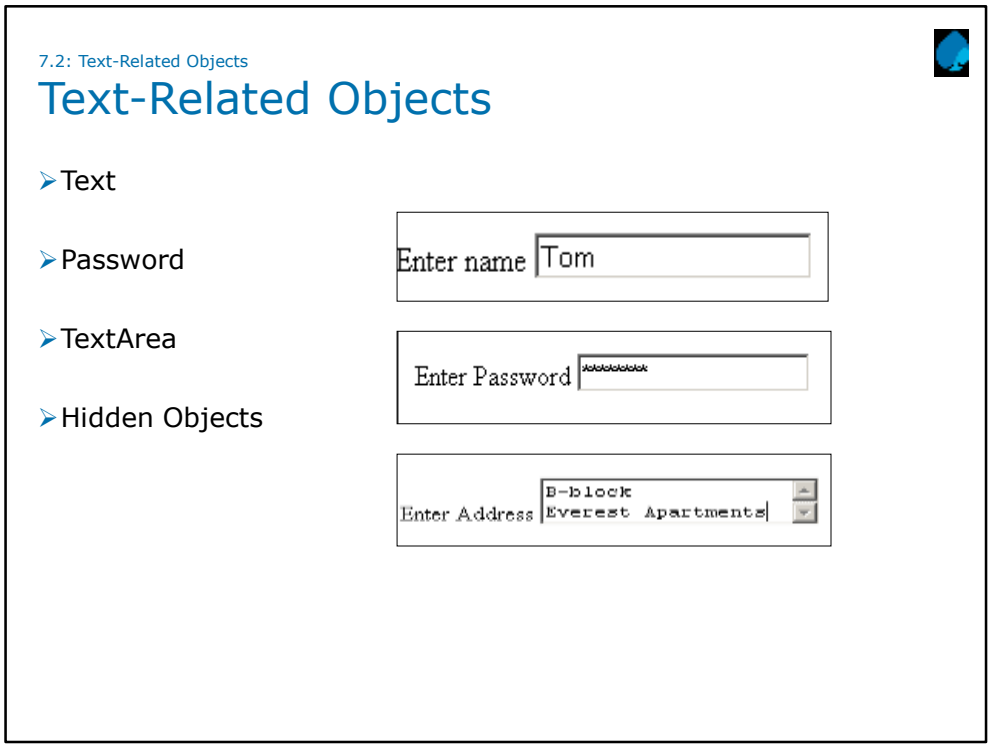
Property/ Method/ Events	Description
action	This property is the same as the value you assign to the ACTION attribute of a <FORM> tag. The value is typically a URL on the server where queries or postings are sent for submission.
elements[]	Returns an array of elements. It includes all the user interface 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:

7.2: Text-Related Objects

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.

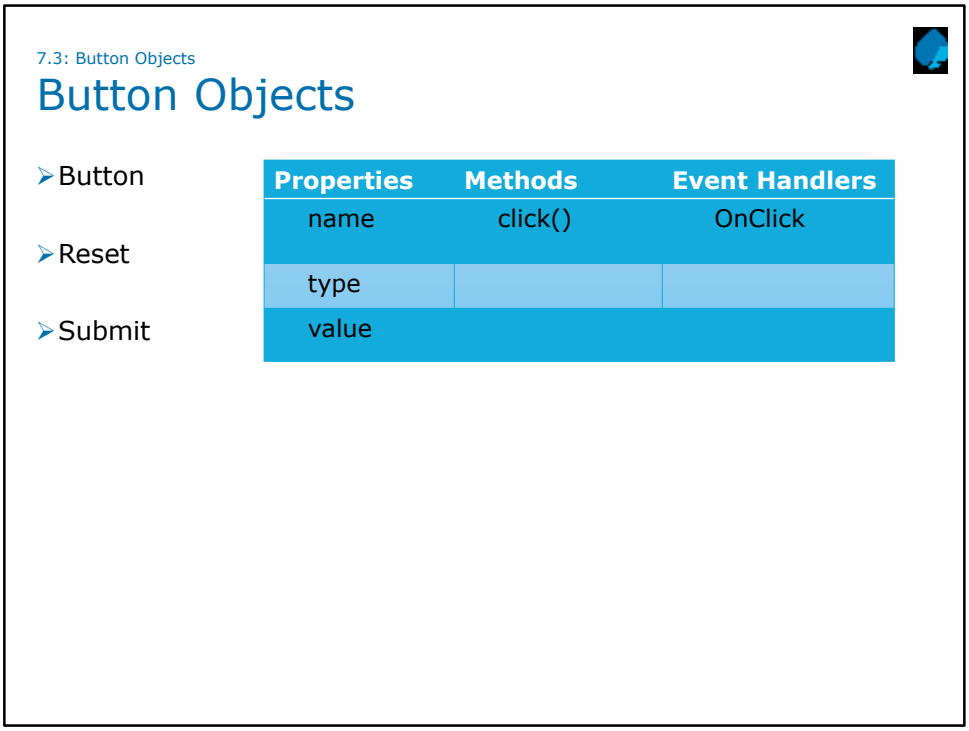
Property/ Events/ Methods	Description
defaultValue	Specifies or returns a defaultValue for a text related objects.
name	This property can be used to reference the text object in the script.
type	Returns the type of text related object
value	A reference to an object's value property returns the string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

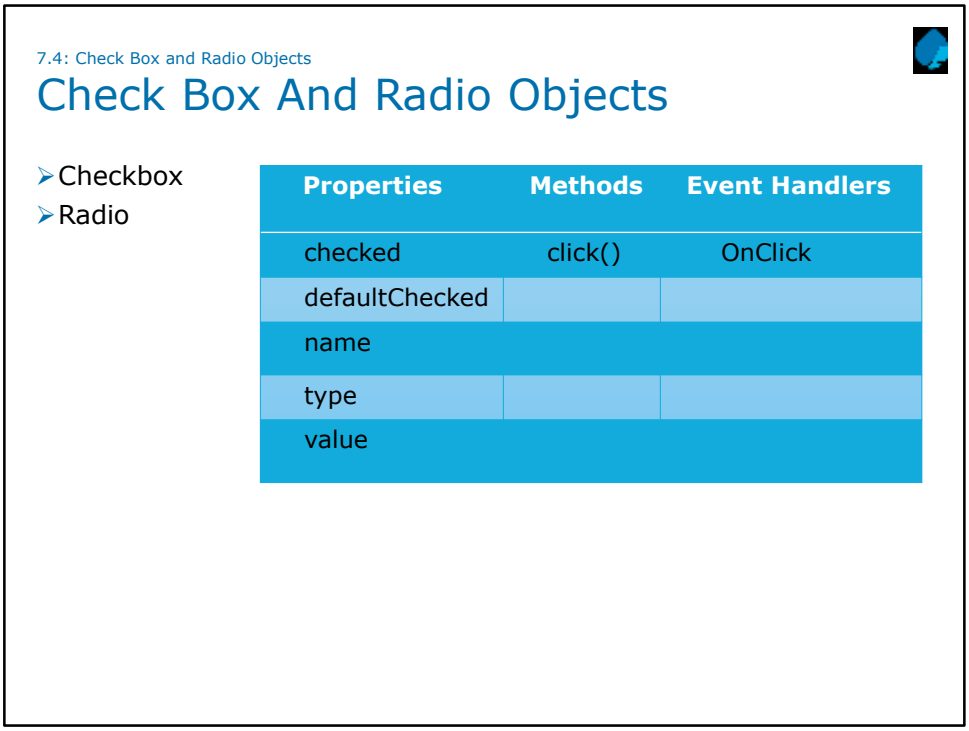
Instructor Notes:



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:



Checkbox object:

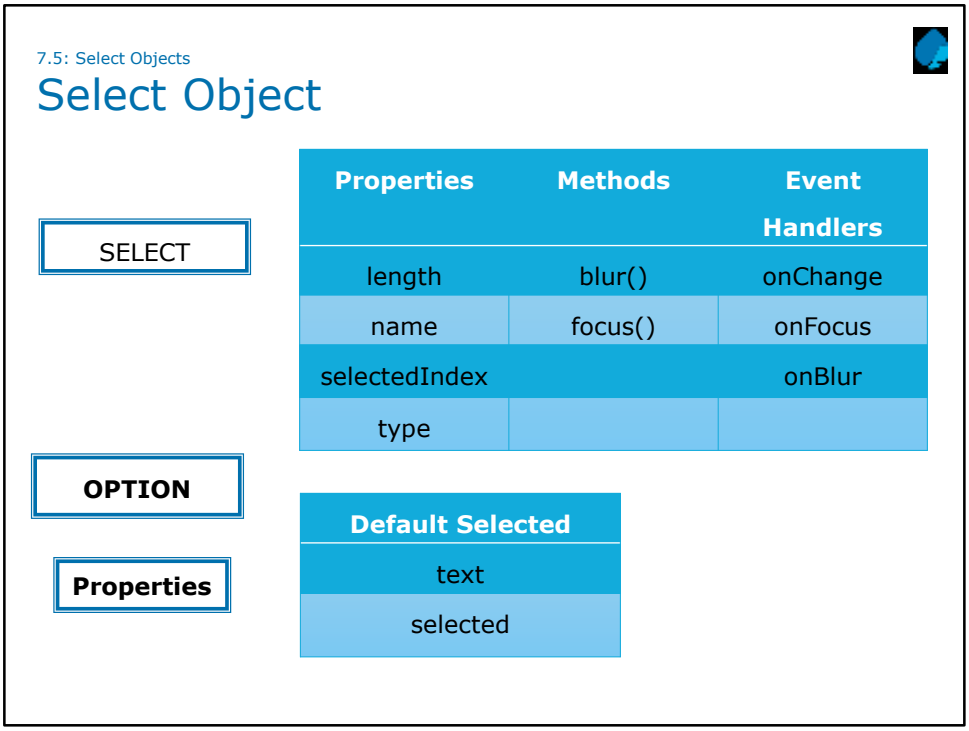
Property/ Events/ Methods	Description
checked	The simplest property of a checkbox gets or lets you set whether or not a checkbox is checked. The value is true for a checked box and false for an unchecked box. Only one radio button in a group can be highlighted checked) at a time. That one button's checked 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.

Refer to Appendix for some more properties

Instructor Notes:

Using 'this' keyword
The 'this' keyword can be used to reference the object which called the function. It can be used within a function scope or global scope and it receives a different value in each scope. Depending on which object has called the function the value of 'this' will differ. The 'this' keyword always points to the object that is calling a particular method.
Consider the example given below:

The 'this' keyword is used in the showColor() function of an object. In this context, this is equal to car, making this code functionality equivalent to the following code snippet

```
var oCar = new Object;
oCar.color = "red";
oCar.showColor = function () {
    alert(this.color); //outputs "red"
};
```

So the reason for using 'this' is you never know what kind of variable names you will use to instantiate an object. By using 'this' you are sure to invoke the correct function with the correct value. Also it allows you to use the same function any number of times. To understand this consider the following code:

```
var oCar = new Object;
oCar.color = "red";
oCar.showColor = function () {
    alert(oCar.color); //outputs "red"
};
```

In the above snippet both oCar1 and oCar2 refer to the same function. The function gives the output according to the object which called the function.

```
function showColor() {
        alert(this.color);
}
var oCar1 = new Object;
oCar1.color = "red";
oCar1.showColor = showColor;
var oCar2 = new Object;
oCar2.color = "blue";
oCar2.showColor = showColor;
oCar1.showColor(); //outputs "red"
oCar2.showColor(); //outputs "blue"
```

Instructor Notes:

Demo

- Form_Object.html
- Select_option.html
- Element_array.html
- Enctype.html
- Hidden_value.html

Instructor Notes:

Lab

- Lab 8:
- Working with Form Object

Instructor Notes:

Summary

- Form Object corresponds to an HTML input form constructed with the FORM tag
- Forms have their own properties, objects, methods & events
- A form can be submitted by calling the JavaScript submit method or clicking the form submit button
- JavaScript can do entry-level validation & do it very easily

Summary
This module provided an understanding of:
Form object and its components.
How to create form objects.
How to handle events.
How to validate data.
How to submit a form.

Instructor Notes:

Review Questions

- Question 1: A form's _____ property is either the GET or POST values assigned to the METHOD attribute in a <FORM> definition.
  - Option 1: Method
  - Option 2: Class
  - Option 3: Object
- Question 2: The intention of the click() method is to enact, via a script, the physical act of clicking a radio button.
  - True / False
- Question 3: A button's _____method should replicate, via scripting, the human action of clicking that button.