

Website Development

IT117 UNIT 8

Form Elements



RADIO GROUP



CHECKBOX



DROP MENU



RESET BUTTON

Radio Group

Two or more radio buttons grouped together

All buttons answer the same question

Only one option can be selected from the group

Use input tag

- type attribute
- name attribute
 - The name value must be the same in all radio inputs within a group
- value attribute
 - The value attribute identifies the individual radio inputs

```
<label>Pet Type:</label>
<input type="radio" name="pet" value="dog"><label>Dog</label>
<input type="radio" name="pet" value="cat"><label>Cat</label>
<input type="radio" name="pet" value="bird"><label>Bird</label>
<input type="radio" name="pet" value="fish"><label>Fish</label>
<input type="radio" name="pet" value="reptile"><label>Reptile</label>
```

Pet Type: Dog Cat Bird Fish Reptile

```
<label>Pet Type:</label>
<br><br>
<input type="radio" name="pet" value="dog"><label>Dog</label>
<br>
<input type="radio" name="pet" value="cat"><label>Cat</label>
<br>
<input type="radio" name="pet" value="bird"><label>Bird</label>
<br>
<input type="radio" name="pet" value="fish"><label>Fish</label>
<br>
<input type="radio" name="pet" value="reptile"><label>Reptile</label>
```

Pet Type:

- Dog
- Cat
- Bird
- Fish
- Reptile

Radio Group Examples

```
<h3>Pet Type </h3>
```

```
<input type="radio" name="pet" value="dog"><label>Dog</label>  
<br>  
<input type="radio" name="pet" value="cat"><label>Cat</label>  
<br>  
<input type="radio" name="pet" value="bird"><label>Bird</label>  
<br>  
<input type="radio" name="pet" value="fish"><label>Fish</label>  
<br>  
<input type="radio" name="pet" value="reptile"><label>Reptile</label>
```

Pet Type

- Dog
- Cat
- Bird
- Fish
- Reptile

Radio Group Examples

```
<label>Pet Type </label>
```

```
<br><br>
```

```
<input type="radio" name="pet" value="dog" id="dog"><label for="dog">Dog</label>  
<input type="radio" name="pet" value="cat" id="cat"><label for="cat">Cat</label>  
<input type="radio" name="pet" value="bird" id="bird"><label for="bird">Bird</label>  
<input type="radio" name="pet" value="fish" id="fish"><label for="fish">Fish</label>  
<input type="radio" name="pet" value="reptile" id="reptile"><label for="reptile">Reptile</label>
```

Pet Type

- Dog
- Cat
- Bird
- Fish
- Reptile

Checkbox

Can be used alone or in a group

If used in a group, all must answer the same question

Multiple options can be selected from the group

Use input tag

- type attribute
- name attribute
 - The name value must be the same in all checkbox inputs within a group
- value attribute
 - The value attribute identifies the individual checkbox inputs

```
<label>Pet Type</label>
<input type="checkbox" name="pet" value="dog"><label>Dog</label>
<input type="checkbox" name="pet" value="cat"><label>Cat</label>
<input type="checkbox" name="pet" value="bird"><label>Bird</label>
<input type="checkbox" name="pet" value="fish"><label>Fish</label>
<input type="checkbox" name="pet" value="reptile"><label>Reptile</label>
```

Pet Type Dog Cat Bird Fish Reptile

```
<label>Pet Type</label>
<br><br>
<input type="checkbox" name="pet" value="dog"><label>Dog</label>
<br>
<input type="checkbox" name="pet" value="cat"><label>Cat</label>
<br>
<input type="checkbox" name="pet" value="bird"><label>Bird</label>
<br>
<input type="checkbox" name="pet" value="fish"><label>Fish</label>
<br>
<input type="checkbox" name="pet" value="reptile"><label>Reptile</label>
```

Pet Type

Dog
 Cat
 Bird
 Fish
 Reptile

Checkbox Group Examples

```
<h3>Pet Type</h3>
```

```
<input type="checkbox" name="pet" value="dog"><label>Dog</label>
<br>
<input type="checkbox" name="pet" value="cat"><label>Cat</label>
<br>
<input type="checkbox" name="pet" value="bird"><label>Bird</label>
<br>
<input type="checkbox" name="pet" value="fish"><label>Fish</label>
<br>
<input type="checkbox" name="pet" value="reptile"><label>Reptile</label>
```

Pet Type

- Dog
- Cat
- Bird
- Fish
- Reptile

Checkbox Group Examples

```
<label>Pet Type </label>
```

```
<br><br>
<input type="checkbox" name="pet" value="dog" id="dog"><label for="dog">Dog</label>
<input type="checkbox" name="pet" value="cat" id="cat"><label for="cat">Cat</label>
<input type="checkbox" name="pet" value="bird" id="bird"><label for="bird">Bird</label>
<input type="checkbox" name="pet" value="fish" id="fish"><label for="fish">Fish</label>
<input type="checkbox" name="pet" value="reptile" id="reptile"><label for="reptile">Reptile</label>
```

Pet Type

- Dog
- Cat
- Bird
- Fish
- Reptile

Single Checkbox Example

```
<input type="checkbox" name="pet owner" value="yes"><label>Check here if you own a pet</label>
```

Check here if you own a pet

```
<input type="checkbox" name="pet owner" value="yes" id="pet-owner"><label for="pet-owner">Check here if you own a pet</label>
```

Check here if you own a pet

Drop Menu

One `<select></select>` tag pair surrounds the list

Creates a drop-down menu of options

Coded similar to other HTML lists

Each list item is coded between `<option>` and `</option>`

- The select tag uses the name attribute
- Each option tag uses the value attribute

Drop Menu Examples

```
<label>Please Select Pet Type:</label>
<select name="pet">
<option value="dog">Dog</option>
<option value="cat">Cat</option>
<option value="bird">Bird</option>
<option value="fish">Fish</option>
<option value="reptile">Reptile</option>
</select>
```

Please Select Pet Type:

Please Select Pet Type:

- Dog
- Cat
- Bird
- Fish
- Reptile

```
<select name="pet">
<option value="#">Please Select Pet Type</option>
<option value="dog">Dog</option>
<option value="cat">Cat</option>
<option value="bird">Bird</option>
<option value="fish">Fish</option>
<option value="reptile">Reptile</option>
</select>
```

Please Select Pet Type

Please Select Pet Type

- Please Select Pet Type
- Dog
- Cat
- Bird
- Fish
- Reptile

Drop Menu Examples

```
<label>Please Select Pet Type:</label>
<select name="pet">
<option value="#">Select One</option>
<option value="dog">Dog</option>
<option value="cat">Cat</option>
<option value="bird">Bird</option>
<option value="fish">Fish</option>
<option value="reptile">Reptile</option>
</select>
```

Please Select Pet Type:

Please Select Pet Type:

- Select One
- Dog
- Cat
- Bird
- Fish
- Reptile

```
<label>Please Select Pet Type:</label>
<br>
<select name="pet">
<option value="#">Select One</option>
<option value="dog">Dog</option>
<option value="cat">Cat</option>
<option value="bird">Bird</option>
<option value="fish">Fish</option>
<option value="reptile">Reptile</option>
</select>
```

Please Select Pet Type:

Please Select Pet Type:

- Select One
- Dog
- Cat
- Bird
- Fish
- Reptile

Drop Menu Examples

```
<label for="pet">Please Select Pet Type:</label>
<br>
<br>
<select name="pet" id="pet">
<option value="#">Select One</option>
<option value="dog">Dog</option>
<option value="cat">Cat</option>
<option value="bird">Bird</option>
<option value="fish">Fish</option>
<option value="reptile">Reptile</option>
</select>
```

Reset Button

Can use input or button code

Both options require the type attribute

- Set to reset

The entire form will clear when clicked

```
<button></button>
```

- Label is placed between tags

```
<input>
```

- Label is set with the value attribute

Reset Button Examples

```
<input type="reset" value="Clear Form">
```

Clear Form

```
<button type="reset">Clear Form</button>
```

Clear Form

**Remember the submit button must be coded using the <input> element*

JavaScript

An object-oriented programming language that enables you to create dynamic website content

Works in the Web browser

Uses object, methods, and functions to execute tasks

Syntax includes

- Curly brackets { }
- Equal and double equal signs = and ==
- Semi-colons ;
- Parenthesis ()

Case sensitive

- Uses camelCase

**Not the same as Java – Java is a desktop publishing programming language*

JS Variables

Must be declared before they can be used

A name given to a memory location

- All operations executed on the variable effect that memory location

Hold the data value

Can be changed at any time

```
var name1 = document.getElementById("id");
```

The Document Object (JS)

- Represents the HTML document that is displayed in that window
- Contains various properties that refer to other objects

```
document.documentElement.nodeName
```

```
var name1 = document.getElementById("id");
```



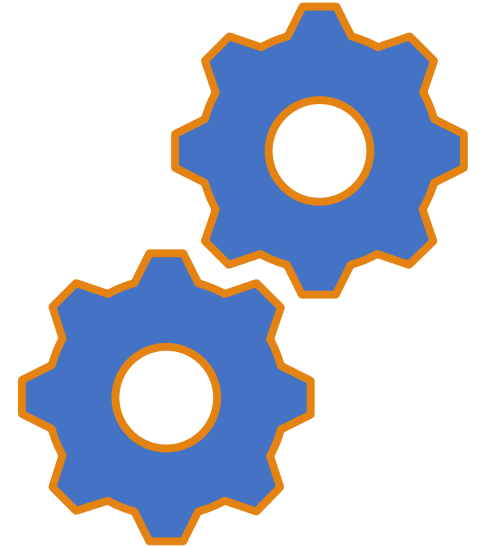
Methods

JS methods are actions that can be performed on objects

Essentially a property that contains a function definition

- Functions stored as object properties

```
var name1 = document.getElementById("id");
```



Note: *The assignment instructions are written in Arial font so the capital letter “I” looks like the lowercase letter “L”. Be sure to type Id for the getElementById method:*

- Remember JS is case sensitive so the first letter in each word must be capitalized, therefore the I in Id cannot be a lowercase “L”
- Remember you are referencing the Id of an element in this method, so you get the element according to the Id, not the Ld.

Function

A block of code written to perform a particular task

Executed when called by something

- Values can be passed to a function
- Functions can return a value

Conditional Statements

The “if else” statement

- Checks the user input and returns a value dependent on the input

if condition a applies, return x
else condition b applies, return y

Unit 8 Assignment JavaScript

```
<h2>Enter your four-digit passcode, then click OK:</h2>  
<input id="id1" type="number" min="1000" max="9999" name="passcode">  
<button onclick="myFunction()" type="button">OK</button>  
<p id="passcode"></p>
```



Must use the "id1" id in the input

Unit 8 Assignment JavaScript

```
<h2>Enter your four-digit passcode, then click OK:</h2>  
<input id="id1" type="number" min="1000" max="9999" name="passcode" >  
<button onclick="myFunction()" type="button">OK</button>  
<p id="passcode"></p>
```

Must add the name attribute to the input



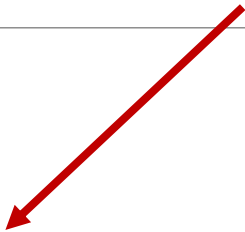
Unit 8 Assignment JavaScript

```
<h2>Enter your four-digit passcode, then click OK:</h2>  
<input id="id1" type="number" min="1000" max="9999" name="passcode">  
<button onclick="myFunction()" type="button">OK</button>  
<p id="passcode"></p>
```

Must add the type attribute to the button tag

Unit 8 Assignment JavaScript

**The HTML for the passcode must be coded inside the form*



```
<h2>Enter your four-digit passcode, then click OK:</h2>
```

```
<input id="id1" type="number" min="1000" max="9999" name="passcode">
```

```
<button onclick="myFunction()" type="button">OK</button>
```

```
<p id="passcode"></p>
```

```
<script>  
</script>
```

Note: *You can insert your JS immediately following the HTML inside the form, or you can add your JS at the end of the document, after </footer> and before </body>*

Unit 8 Assignment JavaScript

```
<h2>Enter your four-digit passcode, then click OK:</h2>
```

```
<input id="id1" type="number" min="1000" max="9999" name="passcode">
```

```
<button onclick="myFunction()" type="button">OK</button>
```

```
<p id="passcode"></p>
```

```
<script>
```

```
function myFunction() {
```

```
}  
</script>
```

The click "event" triggers the function –
When the user clicks the button, the
function is started

Unit 8 Assignment JavaScript

```
<h2>Enter your four-digit passcode, then click OK:</h2>
```

```
<input id="id1" type="number" min="1000" max="9999" name="passcode">
```

```
<button onclick="myFunction()" type="button">OK</button>
```

```
<p id="passcode"></p>
```

```
<script>  
function myFunction() {  
    var inpObj = document.getElementById("id1");  
}  
</script>
```

The `getElementById` method references a specific id and stores the value typed into the input (i.e. the passcode typed by the user)

Unit 8 Assignment JavaScript

```
<h2>Enter your four-digit passcode, then click OK:</h2>
```

```
<input id="id1" type="number" min="1000" max="9999" name="passcode">
```

```
<button onclick="myFunction()" type="button">OK</button>
```

```
<p id="passcode"></p>
```

```
<script>
```

```
function myFunction() {
```

```
    var inpObj = document.getElementById("id1");
```

```
    if (inpObj.checkValidity() == false ) {
```

```
        document.getElementById("passcode").innerHTML = inpObj.validationMessage;
```

```
    }
```

```
}  
</script>
```

The data typed into the input is checked to see if it is valid (is it from 1000 to 9999 and numeric). If it is not valid, the appropriate error message is displayed in the paragraph space with a specific id (in this case, the paragraph with the id of "passcode").

Unit 8 Assignment JavaScript

```
<h2>Enter your four-digit passcode, then click OK:</h2>
<input id="id1" type="number" min="1000" max="9999" name="passcode">
<button onclick="myFunction()" type="button">OK</button>
<p id="passcode"></p>

<script>
function myFunction() {
    var inpObj = document.getElementById("id1");
    if (inpObj.checkValidity() == false ) {
        document.getElementById("passcode").innerHTML = inpObj.validationMessage;
    } else {
        document.getElementById("passcode").innerHTML = "Passcode Valid";
    }
}
</script>
```

The data typed into the input is checked to see if it is valid (is it from 1000 to 9999 and numeric). If it is valid, the appropriate confirmation message is displayed in the paragraph space with a specific id (in this case, the paragraph with the id of "passcode").

Formatting a Specific Input Type

To format a specific type of input tag:
`input[type=xxx] { }`

**Replace xxx with the type you wish to format*

For example:

```
input[type=email] {  
background-color: #800000;  
}
```

```
input[type=radio], input[type=checkbox] {  
padding: 5px;  
display: inline-block;  
}
```

Note: *You must use this method to change the background color of the passcode input to be different than all other form elements.*

Formatting a Specific Input Type

```
input[type=radio], input[type=checkbox] {  
  display: inline-block;  
}
```

If you have

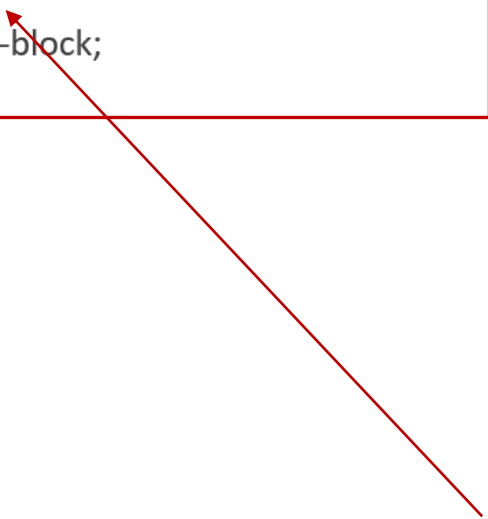
```
input {  
  display: block;  
}
```

in your existing internal CSS, you will need to add this code for the radio and checkbox inputs to display correctly.

Note: This does **NOT** satisfy the required styling of a unit 8 form element since `display` is not on the list of required styles.

Formatting a Specific Input Type

```
input[type=radio], input[type=checkbox] {  
padding: 5px;  
display: inline-block;  
}
```



*If you add padding, this will satisfy one of the requirements.

Note: You cannot apply border, background color, width, and border-radius to the radio and checkbox inputs the same way you do with other form elements.

Extra

You can divide your form into sections using the fieldset element.

- Code `<fieldset>` and `</fieldset>` around the elements you want to group.
 - Style the fieldset element in CSS:
 - `fieldset { }`
- Apply a heading to the section using the `<legend>` and `</legend>` tag pair
 - Style the legend in CSS:
 - `legend { }`

Personalia

First name:

Last name:

Email:

Birthday:

Survey: Poor Average Great

Feedback:

Subscribe to our newsletter

Extra

```
<fieldset>
  <legend>Personalia</legend>
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email"><br><br>
  <label for="birthday">Birthday:</label>
  <input type="date" id="birthday" name="birthday"><br><br>
</fieldset>
```

Unit 8 Assignment HTML and JavaScript

- In the existing contactus.html file, within the existing form, add the following:
 - A radio group (2 or more radio inputs with the same name)
 - A checkbox (at least one)
 - A drop menu (using select and option elements)
 - A reset button (should be at the end of the form with the submit)
 - The passcode input, button, label, and JavaScript (type what is given in the instructions)
 - Add the required attribute to at least three form elements (correct method is getElementById)

**Be sure to code all form elements inside the form, between `<form action="#" method="post" autocomplete="on">` and `</form>`*

*****Ensure the submit element and reset element are the last items within the form.***

Unit 8 Assignment CSS

Code an internal style to change the background color of the passcode input

- You will NOT use a class or id for this
- Ensure the background color for the passcode is not used as a background color for any other form elements.

Add at least two styles to format the new unit 8 form elements (radio, checkbox, drop menu, reset button)

- Choose two styles from the list:
 - Background color
 - Border color
 - Border rounded corners (border-radius)
 - Padding
 - Width
- *Format radio and checkbox inputs to display inline-block (not block) if you have all inputs set to display block.*
- ***Background color, text color, border size, border color, rounded corners, and widths cannot be applied to radio and checkbox elements like the other items.***
- You can apply padding to all inputs
- You can apply all styles from the list to the drop menu and reset elements.

Unit 8 Assignment Submit

Save and upload contactus.html to your server account

Attach any file to the Dropbox (it will not be opened)

Type or copy/paste your URL for the contact us page in the comments section of the Dropbox

- Add /contactus.html to the end of your base URL.
- *Be sure your URL displays, not a text link. If you avoid pressing the space bar or Enter this should prevent the embedded text link from displaying.*

Wrap-Up

Questions?