

Week 2 Workshop

Web Development Fundamentals
HTML, CSS, and JavaScript





Agenda

Activity	Time	~Start
Get Prepared: Log in to Nucamp Learning Portal • Slack • Screenshare	10 minutes	9:00am
Check-In	10 minutes	9:10am
Week 2 Review	40 minutes	9:20am
Task 1	60 minutes	10:00am
BREAK	15 minutes	11:00am
Tasks 2 & 3	90 minutes	11:15am
Check-Out	15 minutes	12:45pm



Check-In

- How was this week for you? Any particular challenges or accomplishments?
- Did you understand the Exercises and were you able to complete them?
- How were the Challenges and Quiz this week?



Week 2 Review- Overview

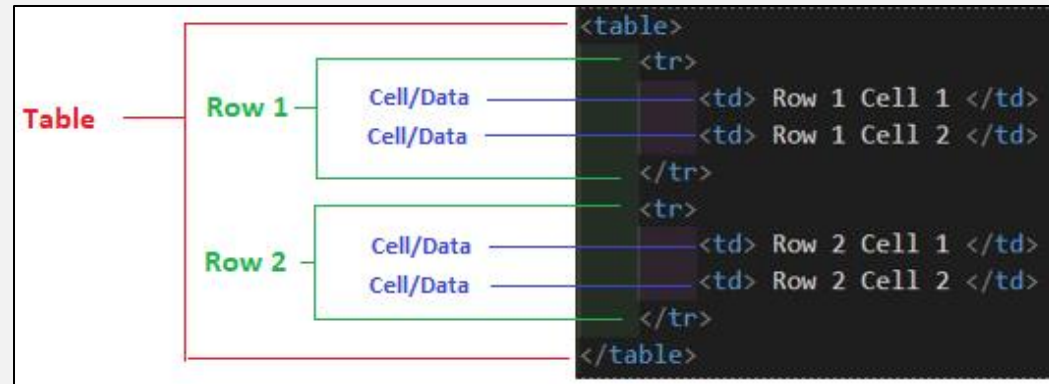
- | | |
|---|--|
| <ul style="list-style-type: none">• HTML Tables• HTML Forms• Cascading Style Sheets (CSS)• CSS Syntax• Adding CSS Three Ways• Element/Class/ID Selectors | <ul style="list-style-type: none">• The CSS Box Model• Shorthand Properties• Other Properties• CSS Specificity• Advanced Selectors• CSS Inheritance |
|---|--|



HTML Tables

- Use to display tabular data - not for page layout

- Simplest example:



Note: The example below includes a border applied using CSS, default style does not include a border

Row 1 Cell 1	Row 1 Cell 2
Row 2 Cell 1	Row 2 Cell 2

- Can use `<thead>` and `<tbody>` as **containers** (sections) for table header and body rows
- Can use `<th>` to create header cells with bold, centered text content

Discuss: Which of the elements listed above accept text content?

<th> & <td> contain actual content

Whenever you see "Discuss", that is an invitation for students to attempt to answer – please volunteer if you know or want to take a guess!



HTML Forms

- Use **action** attribute in the **<form>** start tag to provide a URL for a server-side script to process the form
- Set **method** attribute to "post" or "get"

```
<form action="/form_script.js" method="post">
```

Discuss:

What is the difference between the "post" and "get" methods?

GET: Request data
POST: Send data

Which one would you use to submit a search term to a search engine?

GET since you're requesting data. You can also bookmark or copy URL to share search criteria with others

Which one would you use to upload a file and what other attribute must you add to the **<form>** start tag?

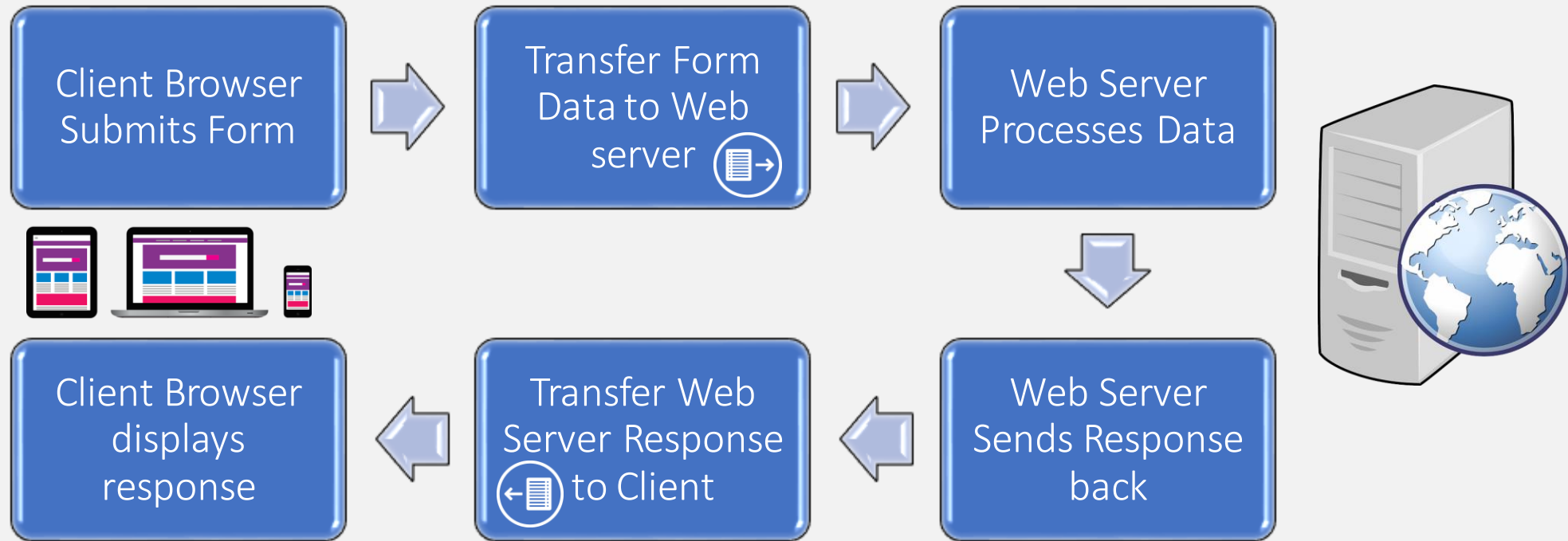
POST since you're sending data. For file uploads you must use the "**enctype**" attribute with the value "**multipart/form-data**" since this supports large binary data

Which one would you use to submit answers to a survey?

POST since you're sending data. Note that posts will conceal the data from the URL



Form Submission





HTML Forms – Input Element

- Most commonly used form control, highly flexible
- Use with **type** attribute to create

Form Input Element "Types"	Description
<code>type="text"</code>	single-line text input
<code>type="submit"</code>	submit button, runs the script given in form action attribute
<code>type="file"</code>	file chooser, allows selecting file from local filesystem to upload - must add <code>enctype="multipart/form-data"</code> to <code><form></code> start tag
<code>type="radio"</code>	create radio button, use in a group of 2 or more
<code>type="checkbox"</code>	create checkbox, can be single or multiple
... and more! https://www.w3schools.com/html/html_form_input_types.asp	

Discuss: Can you name any other types and what they do?

`type="email"` - Check if in email format

`type="month"` - Provides a calendar select

`type="color"` - Provides a Color Palette selector



HTML Forms – More Form Elements

- `<fieldset>` - use to group similar form controls together
 - `<legend>` - use to caption a fieldset
 - `<textarea>` - use to create a multi-line text input
 - `<option>` and `<select>` - use to create dropdown lists
-
- Give all form controls (elements which can have their value changed by the user) a **name** attribute.

Discuss: Why is the **name** attribute important?

Only form elements with a **name attribute** will have their values passed when submitting a **form**



HTML Forms - <label> element

- Form labels are helpful for accessibility in two ways:
 - Used by screen readers
 - Creates a larger target area for mouse clicks or touchscreen finger presses
- The label's *for* attribute value must match the *id* attribute of the form element it's paired with:

Example:

```
<label for="name"> Name: </label>  
<input type="text" id="name">
```

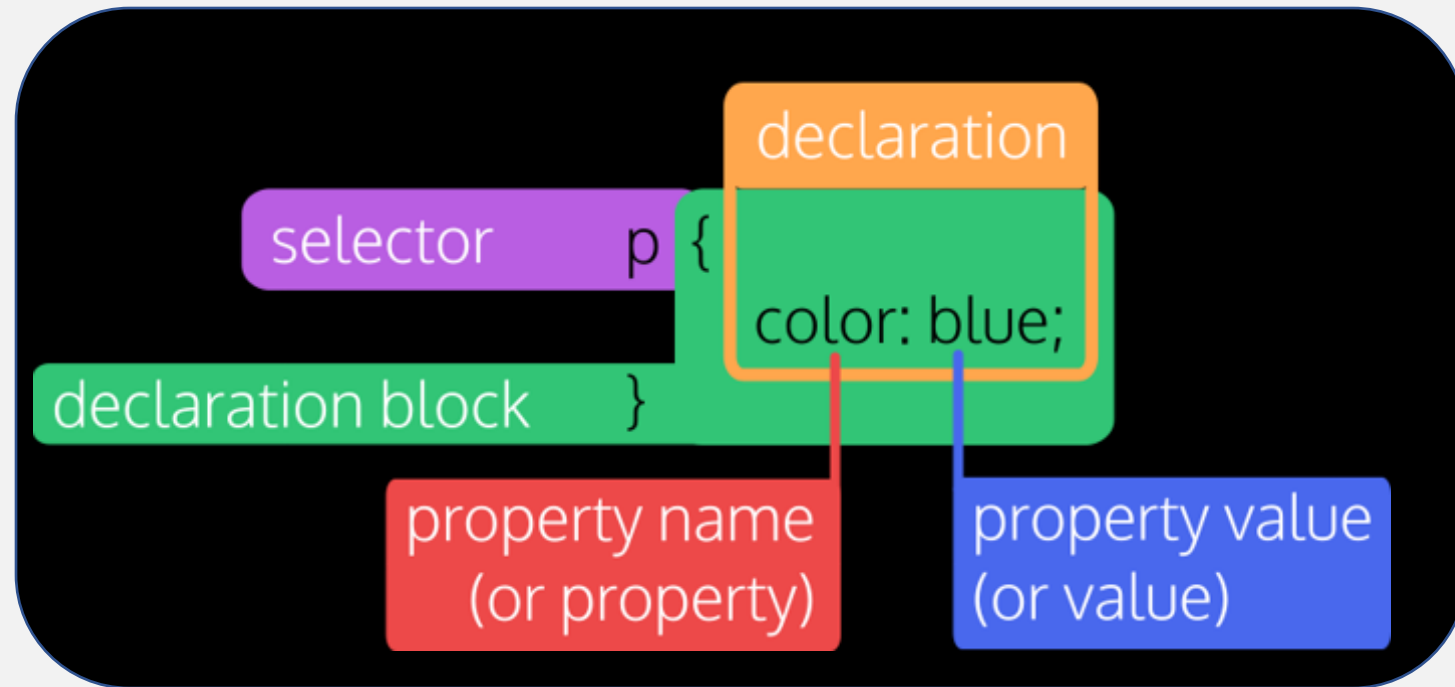


CSS Syntax

- Current implementation is called CSS3

- Syntax:

```
p{  
  color: blue;  
}
```





Adding CSS Three Ways

- **Inline Style**

- Add directly to an element with the **style** attribute

```
<p style="margin:20px">
```

- **Internal Stylesheet**

- Add to the **<head>** of an HTML document using **<style>** element with CSS rulesets inside it:

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport">
  <title>Document</title>
  <style>
    p {
      margin: 20px;
    }
  </style>
</head>
```

- **External Stylesheet**

- Add reference in the **<head>** of an HTML document by using void **<link>** element with **rel** attribute set to "stylesheet" and **href** attribute pointing to an external .css file with CSS rulesets inside it

```
<link rel="stylesheet" href="temp.css" />
```



Element Selector

- Also known as type selector
- The element selector `p { ... }`
- Will select all elements of that name (paragraph elements) in the HTML document `<p>`



Class Selector

- In a CSS ruleset, use `.` in front of the class name

```
.my-class {  
  color: blue;  
}
```

- For the class name: Use hyphens for multiple words (no spaces), don't start with a number, use lower case
- Apply this ruleset on a HTML element using `class` attribute. You can set multiple classes on one element by separating with spaces. Order does not matter

```
<p class="my-class1 my-class2 my-class3">
```

- Can set the same class on multiple elements, ex.:

```
<p class="my-class1 my-class2 my-class3">  
<img class="my-class1"/>
```



ID Selector

- In a CSS ruleset, use **#** in front of the **id** name
- For the ID name: Use hyphens for multiple words (no spaces), don't start with a number, use lower case
- Only set one ID per element
- Apply this ruleset on a HTML element using **id** attribute. **Only use each ID once per document**, do not use it on more than one element
- Generally, avoid using IDs as a selector if you can use a class instead

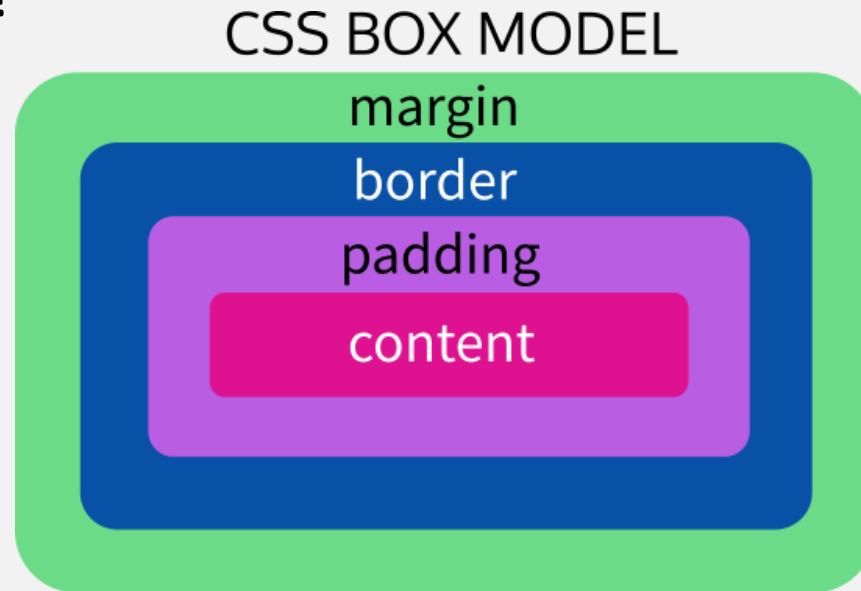
```
#my-id-name {  
  color: blue;  
}
```

```
<p id="my-id-name">
```



CSS Box Model

- Element content, padding, border, margin
- Element's width/height does not include the length of the padding, border, and margin – so be careful or your element will take up more room than you expect!





Shorthand Properties & Margin

- Many CSS properties (but not all) come in sets that can be combined together into a **shorthand property**
- Example: The **margin** shorthand property is a shorthand for the **margin-top**, **margin-right**, **margin-bottom**, and **margin-left** properties
- You can set each separately, e.g.:
 - **margin-top**: 5px;
 - **margin-right**: 10px;
 - **margin-bottom**: 15px;
 - **margin-left**: 20px;
- Or you can set them all at once with the margin shorthand (**margin**: top, right, bottom, left)
 - **margin**: 5px 10px 15px 20px;
- Convenient - but sometimes you may only want to set one side!



Margin & Padding

- **margin** is the space outside of an element's border, **padding** is the space inside it
- **margin** and **padding** shorthand properties set lengths in clockwise direction
 - Example: **margin**: 5px 10px 15px 20px; will apply 5px to **top margin**, 10px to **right margin**, 15px to **bottom margin**, 20px to **left margin** – padding works the same way
- If you provide 2 values to the **margin** or **padding** shorthand, the first will be applied to the **top & bottom**, the second will be applied to the **right & left**
 - Example: **padding**: 5px 10px; will apply 5px to **top** and **bottom** padding, 10px to **left** and **right**
- If you provide **1 value** to the margin or padding shorthand, it will be applied to **all sides**

Discuss: The challenge question on the **margin** shorthand with **three values** -- What was your answer?

```
margin: 10px 20px 5px;
```

This is the same as stating


- **margin-top: 10px**
- **margin-right: 20px**
- **margin-bottom: 5px**
- ▣ **margin-left: 20px** (right/left and top/bottom are linked when using shorthand unless explicitly defined)



Border Property

- The **border** shorthand property includes these border properties:

- border-width
- border-style
- border-color

```
border: 2px solid  red;
```

- You must include the **border style (solid, dotted, double, etc)** to set the border; otherwise it will default to none and the border will not be visible



Font Property

- The **font** shorthand property includes these font properties:
 - font-size
 - font-style
 - font-weight
 - font-family
 - and others
- When using the font shorthand property, font-family and font-size are required, and **font-family must come last in order**


```
font: 18px italic Verdana, Arial, sans-serif;
```



Background Property

- The **background** shorthand property includes all CSS background-properties:

- background-color
- background-image
- background-repeat
- background-size
- Many others

```
background:  red url('strawberry.jpg') repeat;
```

- To only set the background color, use: **background: red;**
 - This will have the same result as writing: **background-color: red;**



Display Property

- Some common values for the display property:
 - **Block:** will **begin on a new line and take up the full width** of the line, can set width & height properties on element with **display: block;**
 - **Inline:** will **stay on the same line; cannot set width/height** properties on element with display: inline;
 - **Inline-block:** will **stay on the same line, but CAN set width & height** properties as if it were a block element

Invalid – Can't set width/height

```
span {  
  display: inline;  
  width: 500px;  
  height: 50%;  
}
```

Block-Level

This is a paragraph

This is another paragraph

Paragraphs are block-level elements, so they stack vertically

Inline

Links are inline elements, so they fit side-by-side



CSS Color Values & Color Property

- Ways to encode color values in CSS:
 - Color names – `blue`, `green`, `violet`, etc
 - Hex color codes - `#ffffff`, `#fa8072`, etc
 - Other ways include `RGBA`, `HSL`, more...
- The `color` property in CSS is used mainly to set **text color**
- **Discuss:** Did everyone get the color picker in VS Code to work by hovering over an existing color value?



CSS Specificity

- An HTML file can have CSS from many different sources:
 - Inline styles
 - Internal stylesheet
 - External stylesheet
 - User-agent stylesheet (browser defaults)
 - User stylesheet (**custom** stylesheet set by user in their browser)
- The cascading algorithm of CSS figures out how to determine which rules to apply



CSS Specificity (continued)

- **Inline styles** have **highest specificity** and will override styles set in stylesheets

```
<p style="margin:20px">
```

- Stylesheet CSS rules are applied in order of appearance, whether internal or external
 - When multiple rules target the exact same selector and property, the **last one is used** since it will overwrite all the previous rules
- If multiple, different selectors target the same element, then the **order does not matter**, instead, rules of specificity are used to figure out which one takes precedence



CSS Specificity and Selectors

- Point value system to determine which selectors have higher specificity:
- Inline styles: **1000**
- ID: **100**
- Class, attribute selector, pseudo-class: **10**
- Element/type: **1**

Discuss: How many points is this selector? (answer in next slide)

.my-class + div:hover

21 points : .my-class is 10 points, :hover is 10 points, div is 1 point



CSS Specificity and Selectors

!important

- The value **!important** can be added to a CSS declaration to give it **highest precedence**, even over inline styles – this is not so much a selector as a **last resort trick** that should be used carefully and sparingly



Some More Advanced Selectors

- Grouping selectors/selector list (`h1, div, .my-class`)

```
h1, div, .myclass{  
  margin: 5px;  
}
```

- Pseudo-classes (`a:hover`)

```
a:hover{  
  margin: 5px;  
}
```

- Attribute selectors (`input[type="text"]`)

```
input[type="text"]{  
  margin: 5px;  
}
```

- Combinators

- Descendant selector (`div h1`)

- Adjacent sibling selector (`div + h1`)

```
<div>  <!--Ancestor-->  
  <form>  
    <h1></h1> <!-- Descendant-->  
  </form>  
</div>
```

- Direct Child selector (`div > h1`)

- General sibling selector (`div ~ h1`)

```
<div>  <!--Parent-->  
  <h1></h1> <!--Child-->  
</div>
```

```
div h1 {  
  margin: 5px;  
}
```

```
div + h1 {  
  margin: 5px;  
}
```

```
div > h1 {  
  margin: 5px;  
}
```

```
div ~ h1 {  
  margin: 5px;  
}
```

Discuss: Did anyone answer the Challenge Question about **pseudo-elements**? Can you give an example?

::first-line - Applies style to the first line of a block-level element

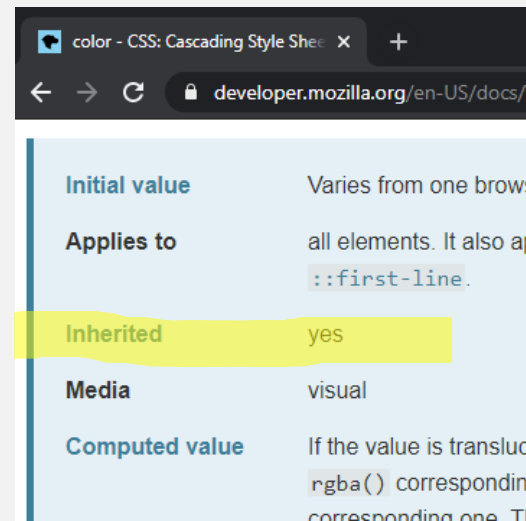


CSS Inheritance

- Some CSS properties are inherited – passed down to descendant elements if they do not have the property set. Others are not.

Examples of properties that are inherited by default:	Examples of properties that are not inherited by default:
color	opacity
font-family	padding
text-align	border

- You can look up each element on MDN for information on its default inheritance:





Questions during this week tasks?

- If we have extra time before the Workshop then feel free to bring up any unresolved questions, and to discuss any Challenge Questions or Code Challenges.
- Otherwise, please start the Workshop Assignment and save the discussion for after the assignment is finished, or online.



Workshop Assignment

- It's time to start the workshop assignment!
- Break out into groups of 2-3.
 - Sit near your workshop partner(s) in person
 - For online Workshops your instructor may break you out into different virtual rooms
- Work closely with each other.
 - Don't forget that the 20-minute rule becomes the 10-minute rule during workshops!
 - 10-minute rule does *not* apply to talking to your partner(s). Work together throughout. This will be useful practice for working with teams in real life.
- Follow the workshop instructions very closely.
 - Talk to your instructor if any of the instructions are unclear to you.



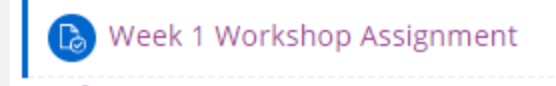
Assignment Submission & Check-Out

- Submit the **datingform.html** page at the bottom of the assignment page in the learning portal.
- Example instruction on the next slide



Submitting Your Assignment

- Go to <https://learn.nucamp.co>
 - Click "Workshop Assignment: Students' Work"
 - Upload your work by clicking "Add Submission", select the file, and then click "save"



- Note that your work is in Draft status
 - Click "submit assignment" to submit it

Submission status	
Attempt number	This is attempt 1.
Submission status	Draft (not submitted)
Grading status	Not graded
Last modified	Sunday, June 2, 2019, 5:29 PM
File submissions	Week 1 Solution (Part 1+2+3).html
Make changes to your submission	
<input type="button" value="Submit assignment"/>	

Submission status	
Attempt number	This is attempt 1.
Submission status	No attempt
Grading status	Not graded
Last modified	-
<input type="button" value="Add submission"/>	



Review: Week 2 Quiz

The following file is viewed in a modern browser. What color is the paragraph text?

```
<!DOCTYPE html>
<html lang="en">
<head>
  <style>
    p {
      color: red;
    }
    p {
      color: blue;
    }
    body {
      color: yellow;
    }
  </style>
</head>
<body>
  <p>Welcome!</p>
</body>
</html>
```

Select one:

- ☒ a. blue ✓ Correct! This would overwrite the other ruleset with the same selector and property name because it comes after it.
- ☐ b. yellow
- ☐ c. red

Your answer is correct.

The correct answer is: blue



Review: Week 2 Quiz

The following file is viewed in a modern browser. What color is the paragraph text?

```
<!DOCTYPE html>
<html lang="en">
<head>
  <style>
    p {
      color: red;
    }
    body > .heading {
      color: blue;
    }
  </style>
</head>
<body>
  <p style="color: green;" class="heading">Welcome!</p>
</body>
</html>
```

Select one:

- ☐ a. blue
- ☒ b. green ✓ Correct! Inline styles have a higher specificity than any other selectors.
- ☐ c. red

Your answer is correct.

The correct answer is: green



Review: Week 2 Quiz

You are tasked with creating a webpage that people can use to join a fitness gym. The webpage must contain a form that asks the user to enter their current weight. Which one of the following set of form controls is **more appropriate** for that purpose?

Select one:

☐ a.

```
<input type="checkbox" name="weight" value="under 100"/> Under 100 lbs
<input type="checkbox" name="weight" value="100-120"/> 100-119 lbs
<input type="checkbox" name="weight" value="121-140"/> 120-139 lbs
<input type="checkbox" name="weight" value="141-160"/> 140-159 lbs
<input type="checkbox" name="weight" value="161-180"/> 160-179 lbs
<input type="checkbox" name="weight" value="181-200"/> 180-199 lbs
<input type="checkbox" name="weight" value="over 200"/> Over 200 lbs
```

☒ b.

```
<input type="radio" name="weight" value="under 100"/> Under 100 lbs
<input type="radio" name="weight" value="100-120"/> 100-119 lbs
<input type="radio" name="weight" value="121-140"/> 120-139 lbs
<input type="radio" name="weight" value="141-160"/> 140-159 lbs
<input type="radio" name="weight" value="161-180"/> 160-179 lbs
<input type="radio" name="weight" value="181-200"/> 180-199 lbs
<input type="radio" name="weight" value="over 200"/> Over 200 lbs
```

✓ Correct! Radio buttons would be more appropriate than checkboxes, because a person cannot be in more than one weight range at the same time. There are other form controls that would be appropriate here as well because they only allow for a single answer - option/select, for example, or a text input element.

Your answer is correct.

The correct answer is:

```
<input type="radio" name="weight" value="under 100"/> Under 100 lbs
<input type="radio" name="weight" value="100-120"/> 100-119 lbs
<input type="radio" name="weight" value="121-140"/> 120-139 lbs
<input type="radio" name="weight" value="141-160"/> 140-159 lbs
<input type="radio" name="weight" value="161-180"/> 160-179 lbs
<input type="radio" name="weight" value="181-200"/> 180-199 lbs
<input type="radio" name="weight" value="over 200"/> Over 200 lbs
```



Review: Week 2 Quiz

Question 4

Correct

1.00 points out of 1.00

Flag question

The following HTML file is used to let students submit a programming assignment. There are three errors in the form. What are they?

```
<!DOCTYPE html>
<html lang="en">
<body>
  <form method="get" action="upload.php" enctype="text/plain">
    <div>
      <label for="studentId">Student ID: </label>
      <input type="text" name="studentId" id="studentId"
        placeholder="Enter your student ID here" />
    </div>
    <div>
      <label for="fileToUpload">Select a file to upload: </label>
      <input type="file" id="fileToUpload" />
    </div>
    <div>
      <input type="submit" value="Upload" />
    </div>
  </form>
</body>
</html>
```

Select one or more:

- ☐ a. You need to add the attribute **value="fileToUpload"** to the second **input** element.
- ☒ b. Using the attribute **method="get"** in the form element is inappropriate. ✓ Correct! The form element must have the attribute **method="post"** if you want to use file upload. You cannot use **method="get"** for file upload.
- ☒ c. In the form start tag, the value for the attribute **enctype** should contain **"multipart/form-data"** instead of **"text/plain"** for uploading a file. ✓ Correct! The form element must have the attribute **enctype="multipart/form-data"** if you want to use an input element with the type of **"file"**.
- ☒ d. The attribute **name="fileToUpload"** (or a similar name) must be added to the second **input** element. ✓ Correct! The **name** attribute is required in order for the server to process the data correctly.

Your answer is correct.

The correct answers are: In the form start tag, the value for the attribute **enctype** should contain **"multipart/form-data"** instead of **"text/plain"** for uploading a file., Using the attribute **method="get"** in the form element is inappropriate., The attribute **name="fileToUpload"** (or a similar name) must be added to the second **input** element.



Review: Week 2 Quiz

Question 5

Correct

1.00 points out of 1.00

Flag question

Within an HTML document, you can use the same class attribute on multiple elements, but you should only use an id once.

Select one:

- ☒ True ✓
- ☐ False

Correct! Classes can be reused, an id is a unique identifier that can only be used once.

The correct answer is 'True'.

Question 6

Correct

1.00 points out of 1.00

Flag question

To only select **div** elements that are children of **form** elements, but not necessarily direct children, as in the example below, what selector would you use?

```
<form >
  <fieldset >
    <div > How would you select this div? </div >
  </fieldset >
</form >
<div > But not this div? </div >
```

Select one:

- ☒ a. form div ✓ Correct. A space between two selectors creates a descendant selector, which selects for all elements that match the second selector if it is a descendant of the first selector).
- ☐ b. form > div
- ☐ c. form, div
- ☐ d. div

Your answer is correct.

The correct answer is: form div



Review: Week 2 Quiz

Question 7

Correct

1.00 points out of 1.00

Flag question

Match the following CSS keywords to their definitions.

An element with its display property set to this value will not start on a new line.

inline ✓

This property sets the width of the space outside of the border of an element.

margin ✓

This property sets the width of the space inside the border of an element.

padding ✓

Your answer is correct.

The correct answer is: An element with its display property set to this value will not start on a new line. → inline, This property sets the width of the space outside of the border of an element. → margin, This property sets the width of the space inside the border of an element. → padding

Question 8

Correct

1.00 points out of 1.00

Flag question

Width and height CSS properties are inherited by default.

Select one:

☐ True

☒ False ✓

Incorrect. Width and height CSS properties are not inherited by default.

The correct answer is 'False'.



Review: Week 2 Quiz

Question 9

Correct

1.00 points out of 1.00

Flag question

In the HTML below, what attribute needs to be set on the label element so that the label will be associated with its corresponding input?

```
<label>Your IQ</label>: <input type="number" id="iq" name="myIQ" placeholder="Enter your IQ here" />
```

Select one:

- ☐ a. value="myIQ"
- ☒ b. for="iq" ✓ Correct! A for attribute should be set in the label element that matches the id attribute in the input element.
- ☐ c. for="myIQ"
- ☐ d. name="iq"
- ☐ e. name="myIQ"
- ☐ f. id="myIQ"
- ☐ g. value="iq"
- ☐ h. id="iq"

Your answer is correct.

The correct answer is: for="iq"

Question 10

Correct

1.00 points out of 1.00

Flag question

You can have multiple external stylesheets linked in one HTML file.

Select one:

- ☒ True ✓
- ☐ False

Correct! Multiple external CSS files can be linked in one HTML document, but you need to be careful about how they are ordered in case there are conflicts.

The correct answer is 'True'.

Happy learning!
