

Bootstrap: Week 4

Workshop Presentation





Today's Agenda

Activity	Estimated Duration
Check-In & Set-Up	15 mins
Review	75 mins
Workshop Assignment Task 1	30 mins
Break	15 mins
Workshop Assignment Task 2	60 mins
Workshop Assignment Task 3	30 mins
Check-Out (Feedback & Wrap-Up)	15 mins

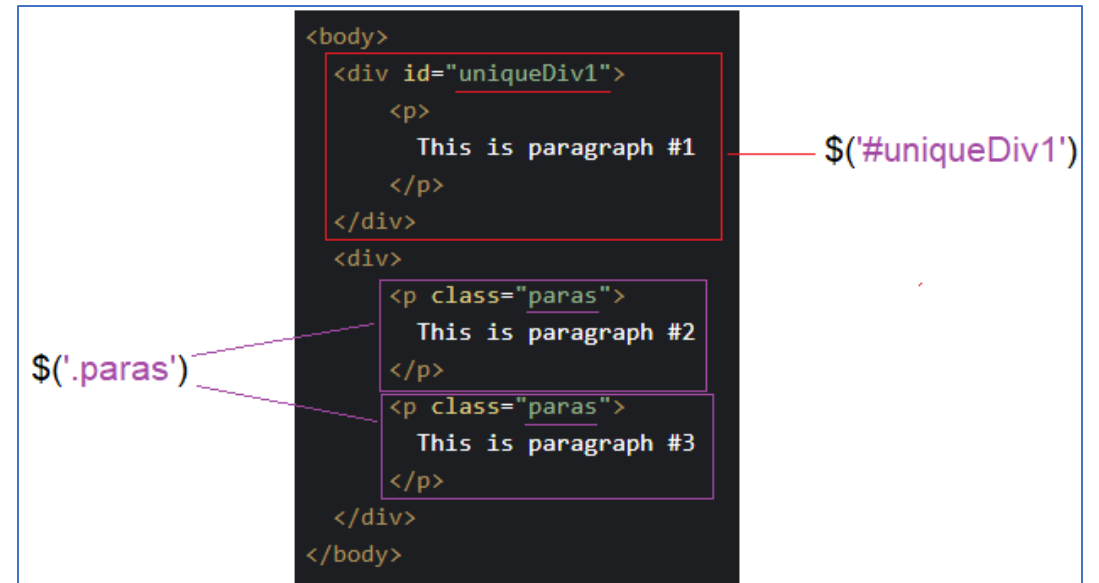
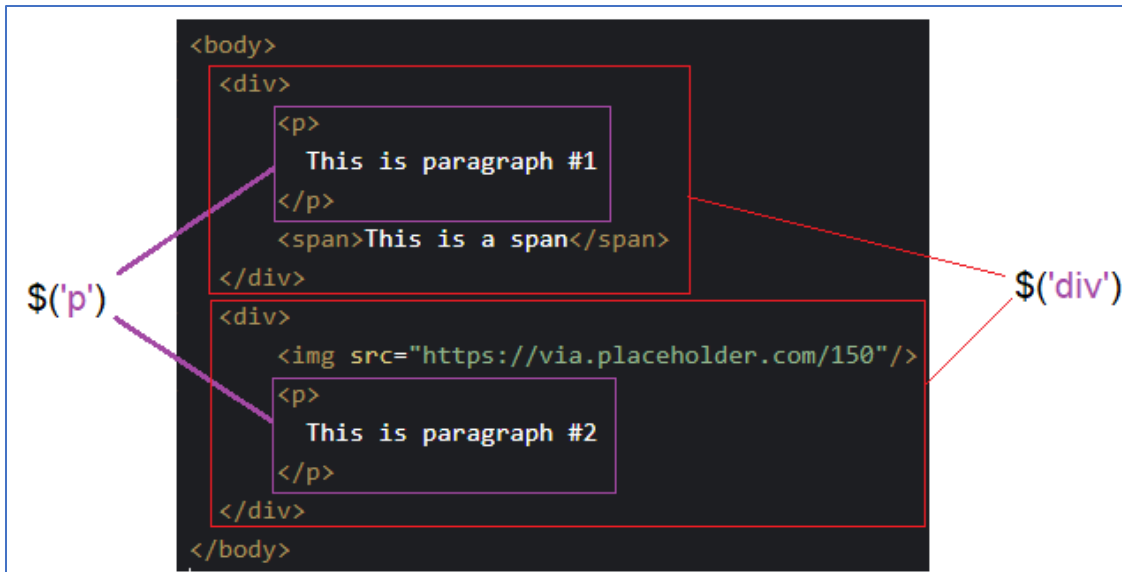


Review: jQuery

Discuss:

- How do you use jQuery to select a specific DOM node?

`$('css selector')`





Review: jQuery

Discuss:

- What is the way to activate a jQuery script in an HTML document when the document loads?

NOTE: This code that we are using in the beginning of the jQuery script:

```
$(function() { ... });
```

is the shorthand, recommended version for this code, called the jQuery *ready method* or *ready function*, which specifies a function to load when the webpage's DOM is ready:

```
$( document).ready(function() { ... });
```



Review: Scss

Discuss:

- What is the difference between Sass and Scss?

SCSS is a variant of Sass which is a CSS preprocessor (compiles Sass code to CSS syntax)

- You used variables and nesting in your Sass exercise. What are some of the other ways you can use Sass?

Creating Variables

`$color-dark: #60106b;`

Utilize custom functions

`@mixin`

Utilize built-in functions

`lighten(color, amount)`

Looping through elements

`@for @each`

Logically separate style code into multiple files for manageability

`@_partials`



Review: NPM Scripts

Discuss:

- Go through each of the scripts inside the scripts object in `package.json` and name what each one does.

```
"build": "npm run clean && npm run imagemin && npm run copyfonts && npm run usemin"
```

build: Runs "clean" then "imagemin" then "copyfonts" and finally the "usemin" npm scripts

```
"clean": "rimraf dist"
```

clean: Runs node package "rimraf" on the "dist" folder which will delete it

```
"copyfonts": "copyfiles -f node_modules/font-awesome/fonts/* dist/fonts"
```

copyfonts: Runs node package "copyfiles" and copies font-awesome fonts to "dist/fonts" folder

```
"imagemin": "imagemin img/* -o dist/img"
```

imagemin: Runs node package "imagemin" to compress all files in "img" and outputs compressed files to "dist/img" folder



Review: NPM Scripts - cont

```
"usemin": "usemin contactus.html -d dist --htmlmin -o dist/contactus.html && usemin aboutus.html  
-d dist --htmlmin -o dist/aboutus.html && usemin index.html -d dist --htmlmin -o dist/index.html"
```

usemin: Runs node package "**usemin**" to compress html & css files and output to "**dist**" folder

```
"lite": "lite-server"
```

lite: Runs node package "**lite-server**" which runs local web server for development purposes

```
"scss": "node-sass -o css/ css/"
```

scss: Runs node package "**node-sass**" which compiles all .scss files in "css" folder and outputs compressed version to "dist/css" folder

```
"start": "npm run watch:all"
```

start: Runs script "**watch:all**"

```
"watch:scss": "onchange \"css/*.scss\" -- npm run scss"
```

watch:scss: Runs node package "**onchange**" which watches all *.scss (SASS) files located in "**css**" folder. If a change is made, it will run "**scss**" script

```
"watch:all": "parallelshell \"npm run watch:scss\" \"npm run lite\""
```

watch:all: Runs node package "**parallelshell**" which allows multiple terminals to be open concurrently then runs "**watch:scss**" & "**lite**" scripts



Review: NPM Scripts

What does the **&&** symbol do in these scripts?

You learned that the **&&** operator returns with the first **falsy** value it runs into, or the last **truthy** value if there are no falsy values.

The principle is the same when applied to the scripts

```
"build": "npm run clean && npm run imagemin && npm run copyfonts && npm run usemin"
```

The "**build**" script will first try "npm run clean", then if it works it will go to the next script, "npm run imagemin", and so forth, but if "npm run clean" fails (is falsy) it will abort.



Review: Build Processes

Discuss:

- What is minification?

The process of removing unnecessary code in order to reduce the file size and increase load times

- What is uglification?

Compressing the files size by removing all white spaces, new lines, and other performance enhancements at the cost of making the code "ugly" and reduced readability

- What is file concatenation? (in the context of building your files for deployment)

The process of combining multiple files into one single file for reference



Review: JavaScript Loops

- What is wrong with this for loop and how would you fix it so that it prints the numbers 5-10 to the console?

WARNING: DO NOT type the code below into your console as-is. Discuss it first with your instructor and fix the code before testing it in your console.

```
for (let i = 11; i > 10; i++) {  
  console.log(i);  
}
```

```
for (let i = 5; i <= 10; i++) {  
  console.log(i);  
}
```

This code will cause an "infinite loop" and eventually crash when memory is filled



Review: JavaScript While Loops

Discuss:

- What is the difference between a **do ... while** loop and a **while** loop?

```
let i = 0;
```

```
while (i < 10) {  
  console.log("The iterator is", i);  
  i++;  
}
```

Runs AT LEAST 1 time

```
i = 0;
```

```
do {  
  console.log(i);  
  i++;  
} while (i > 0 && i < 10);
```



Review: JavaScript Arrays

Discuss:

- What is an array, and how do you create one?

A single variable that stores multiple items
`const fruits = ['mangos', 'bananas', 'pineapples']`

You can declare an array with **'let'** as well

- How would you access the second item in an array?

`fruits[1]`

Remember Arrays are zero based index (starts at 0)

- How do you find the length of an array?

`fruits.length`



Review: JavaScript Array Methods

Discuss:

- What does the word *method* mean in JavaScript?

A method is a function that is tied to an object

- What does the syntax to call a method typically look like?

```
Object.method(argument);
```

- There is more than one way to check if an item exists inside an array. Name two array methods that can do this. What are the differences?

indexOf()

Takes an argument and returns its index # if exists
or **-1** if not in the array

```
myArr.indexOf('foo');
```

includes()

Takes an argument and returns a boolean **true** if exists
or **false** if not in the array

```
myArr.includes('bar');
```



Review: JavaScript Array Methods

Discuss:

- What is a destructive/mutating method versus a non-mutating method?

A destructive/mutating method will modify the original Array (i.e. **push()**)

A non-mutating method will not modify the original Array and create a new array in memory (i.e. **concat()**)

- What are the differences between push/pop and shift/unshift?

push/pop (add/remove) will mutate the end of the original array

shift/unshift (add/remove) will mutate the beginning of the original array AND shift the index value of the items in the array down

- What are the return values of push, pop, shift, and unshift?

push - returns the new length of the array (array.length)

pop - returns the value that was removed from the end of the array

shift - returns the value that was removed from the beginning of the array

unshift - returns the new length of the array (array.length)



Review: In Class Example

- Have students open Chrome Developer Console type along:
 - `let myArr = [1,2,3,4];`
 - `console.log(myArr);`
 - `myArr.push(5,6,7);` [Returns the new length of the array]
 - `console.log(myArr);`
 - `myArr.pop();` [Returns `7` which was in the last index of the array]
 - `console.log(myArr);` [Notice that the `7` is now gone from the array]
 - `myArr.shift();` [Returns the `1` which was at the first index of the array]
 - `console.log(myArr);` [Notice that the `1` is now gone from the array]
 - `myArr.unshift(0,1);` [Returns the new length of the array]
 - `console.log(myArr);` [Notice that `0` and `1` are added to the beginning of the array]



Week 4 Workshop Assignment

- All students should aim to finish and submit your assignment before you leave today.
- Work in pairs, or groups of three. Talk to each other and figure things out together!
- 10-minute rule during workshops: If you and your paired partner have spent more than 10 minutes trying to figure something out, ask your instructor for help.



Week 4 Workshop Assignment – Format

Task 1

(Approximate End Time: **11:00am**)

Break Time

(Approximate End Time: **11:15am**)

Task 2

(Approximate End Time: **12:15pm**)

Task 3

(Approximate End Time: **12:45pm**)

Happy learning!



Review: Challenges/Quiz

- It is important that students have ample time to complete the assignment during the workshop.
- If there is time left *after* students have completed the Workshop Assignment, review the Week 4 challenges and quiz together.



Review: Week 4 Quiz

Sass (.sass) and Scss (.scss) are both valid implementation of the Sass preprocessor language. Sass was the original implementation, and Scss (which stands for Sassy CSS) came after.

Select one:

- ☒ True ✓
- ☐ False

In jQuery, the code:

```
$(function() { ... })
```

is a valid and recommended shortcut for the code:

```
$(document).ready(function() { ... })
```

Select one:

- ☒ True ✓
- ☐ False

```
let pastriesArr = ['muffin', 'cookie', 'pie', 'cupcake', 'strudel'];
```

What would you type to retrieve the value of the third element ("pie") in this *pastriesArr* array?

Answer:





Review: Week 4 Quiz

What is the return value from this function:

```
function applyAdmissionDiscounts(fee, isChild, isMember) {  
  if (isChild)  
    fee *= .3;  
  else if (isMember)  
    fee /= 2;  
  return fee;  
}
```

...if invoked in this way?:

```
applyAdmissionDiscounts(10, true, false);
```

Answer: 3





Review: Week 4 Quiz

Which of these are *not* DOM Events? There are TWO answers.

Select one or more:

- ☐ paste
- ☐ dragenter
- ☒ onclick ✓ **onclick** is not itself a valid DOM event - it is a keyword used to create an event handler for the event **click**.
- ☒ mouseoff ✓ While there are valid mouse events such as **mouseout** and **mouseleave**, **mouseout** is not a valid event.
- ☐ click

https://www.w3schools.com/jsref/dom_obj_event.asp