# Week 10 - "Where Are You?" Self-Check

**Self Evaluation Questions**

1. **FILE SYSTEM NAVIGATION - What would you type to create a folder called *my_awesome_folder* on the command line?**

2. **FILE SYSTEM NAVIGATION - What would you type to create multiple folders with the names *awesome_folder_one*, *awesomeness_two* and *awesome_sauce* on the command line?**

3. **FILE SYSTEM NAVIGATION - What would you type to create a single HTML file called *mainPage*?**

4. **FILE SYSTEM NAVIGATION - What would you type to create a single Javascript file called *myScript*?**

5. **FILE SYSTEM NAVIGATION - What would you type to create a single CSS file called *coolStyles*?**

6. **FILE SYSTEM NAVIGATION - What would you type to create those same files on the command line all at once?**

7. **FILE SYSTEM NAVIGATION - What would you type to print out what directory you're currently working in on the command line?**

8. **FILE SYSTEM NAVIGATION - What would you type to navigate to your home directory?**

9. **FILE SYSTEM NAVIGATION - What would you type to navigate one directory up?**

10. **FILE SYSTEM NAVIGATION - What would you type to navigate down into a folder called *my_first_app* in your current working directory?**

11. **FILE SYSTEM NAVIGATION - What would you type to print a list of the files and folders in your current working directory?**

12. **GIT COMMANDS - What would you type to initialize your current working directory as a git repository on the command line?**

13. **GIT COMMANDS - After initializing your current working directory for git and presuming you created a repository on Github, what would you type on the command line to add a remote and link your current working directory to the Github repository you just created? Let the link for the Github repository be: *https://github.com/zero-cool/gnarly-repository.git***

14. **GIT COMMANDS - What would you type to get the status of your local repository?**

15. **GIT COMMANDS - After making some changes to files and/or folders in your local repository, what would you type to stage your changes for commits on the command line?**

16. **GIT COMMANDS - To commit your changes with a message, what would you type on the command line?**

17. **GIT COMMANDS - What would you type on the command line to get a log of all your commits for the local repository you're currently working in?**

18. **GIT COMMANDS - What would you type on the command line to *unstage* or *reset* the already *staged* changes for a commit?**

19. **GIT COMMANDS - What would you type on the command line to push your commits to your remote master branch or *myFeatures* branch?**

20. **GIT COMMANDS - What would you type on the command line to fetch or download changes from a *master* branch without automatic merging?**

21. **GIT COMMANDS - What would you type on the command line to merge the fetched/downloaded changes from a *master* branch from the previous question?**

22. **GIT COMMANDS - What would you type on the command line to combine both *git fetch* and *git merge*?**

23. **HEROKU - APP DEPLOYMENT - What would you type to create a new Heroku app for your current local repository on the command line?**

24. **HEROKU - APP DEPLOYMENT - What would you type to deploy your github repository code to heroku on the command line?**

25. **HTML BASICS - List the four core DOM elements composing a web page. Fill in the blanks:**


(0) _____

(1) _____

(2) _____

(3) _____

(4) _____

(5) _____

26. **HTML BASICS - Write out the DOM element that allows you to apply inline CSS to your webpage. Fill in the blanks:**


(0) _____

(1) _____

27. **HTML BASICS - Write out the DOM element and its associated attributes that allow you to load an external stylesheet to your webpage. Fill in the blanks:**

```
<!-- Load external stylesheet -->
<__(0)__ rel="____(1)_____" href="styles.css" type="__(2)___">
```

(0) _____

(1) _____

(2) _____

28. **HTML BASICS - Write out the DOM element that allows you to execute inline Javascript on your webpage. Fill in the blanks:**

```
<!-- Inline script -->
<__(0)__ type="text/javascript">

  // A bunch of Javascript....

</__(1)__>
```

(0) _____

(1) _____

29. **HTML BASICS - Write out the DOM element and associated attributes that allows you to load an external Javascript file or source on your webpage. Fill in the blanks:**

```
<!-- Load external Javascript -->
<__(0)__ type="__(1)__/__(2)__" src="_example.js_">

  // A bunch of Javascript...

</__(3)__>
```

(0) _____

(1) _____

(2) _____

(3) _____

30. **JAVASCRIPT BASICS - Declare a variable named *ninja*.**

_____

31. **JAVASCRIPT BASICS - Using proper syntax, declare an empty anonymous function (no code to execute in the block) that takes in one argument named *coolness*.**

_____

32. **JAVASCRIPT BASICS - Using proper syntax, declare an empty named function (no code to execute in the block) with the name *beAwesome* that takes in one argument named *awesomeSauce*.**

33. **JAVASCRIPT BASICS - Using proper syntax, declare an empty anonymous function expression with a variable named *beAwesomer* that takes in one argument named *awesomerSauce*.**

34. **JAVASCRIPT BASICS - Using proper VanillaJS syntax, write the most widely supported function that loads the script(s) after all DOM elements and media content (pictures, videos, etc.) have been loaded:**

35. **JAVASCRIPT BASICS - Using proper jQuery syntax, write the function that you would put at the beginning of a Javascript file to load the script after just the DOM elements have loaded on the web page:**

36. **JAVASCRIPT BASICS - Why is $(document.).ready() better than window.onload?**

- No reason. They're both interchangeable.
- It's reverse. window.onload is better than $(document).ready().
- The .ready() loads scripts right after the DOM elements load but the .onload loads scripts after all DOM elements and media content have been loaded.

37. **JAVASCRIPT BASICS - Create an array called *myCoolArray* with four string elements of your choosing.**

38. **JAVASCRIPT BASICS - Write out how you would access the third element in the array from the previous question you created in the previous question.**

39. **JAVASCRIPT BASICS - Create an object called *myObject* with four properties:**
**- *myName* property with a string value of your name,**
**- *myBirthdayMonth* property with a string value of your birth month,**
**- *myAlertFunc* property whose value is a function that takes in one argument (string type) and invokes an alert with the argument text in it**
**- *myAdditionFunc* property whose value is a function that takes in two arguments (integer type), adds those two arguments, then returns the sum/answer).**

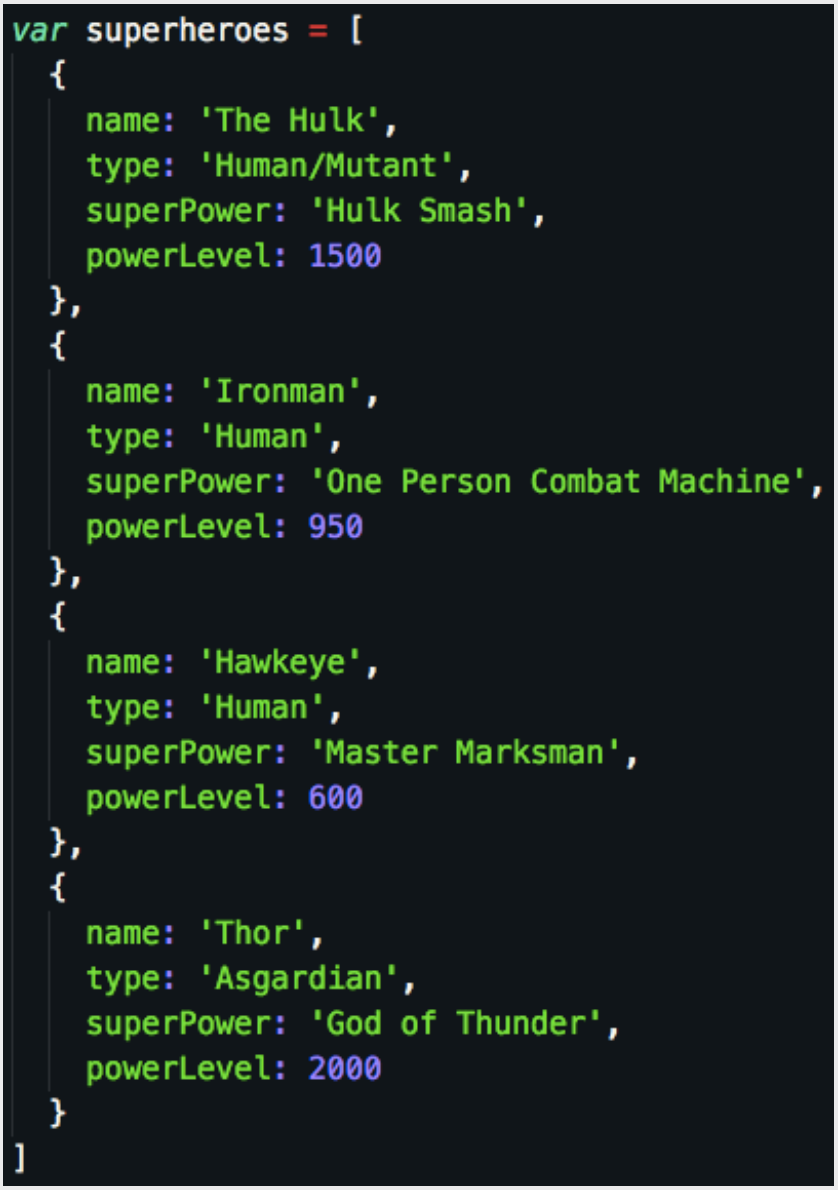
40. **JAVASCRIPT BASICS - Using the object that was just created, output the values of the four object properties to the browser console using console.log and dot notation.**

myName

myBirthMonth

myAdditionFunc(3,4)

Get the alert to appear in the browser with myAlertFunc("hello there")

41. 

```
var superheroes = [
  {
    name: 'The Hulk',
    type: 'Human/Mutant',
    superPower: 'Hulk Smash',
    powerLevel: 1500
  },
  {
    name: 'Ironman',
    type: 'Human',
    superPower: 'One Person Combat Machine',
    powerLevel: 950
  },
  {
    name: 'Hawkeye',
    type: 'Human',
    superPower: 'Master Marksman',
    powerLevel: 600
  },
  {
    name: 'Thor',
    type: 'Asgardian',
    superPower: 'God of Thunder',
    powerLevel: 2000
  }
]
```

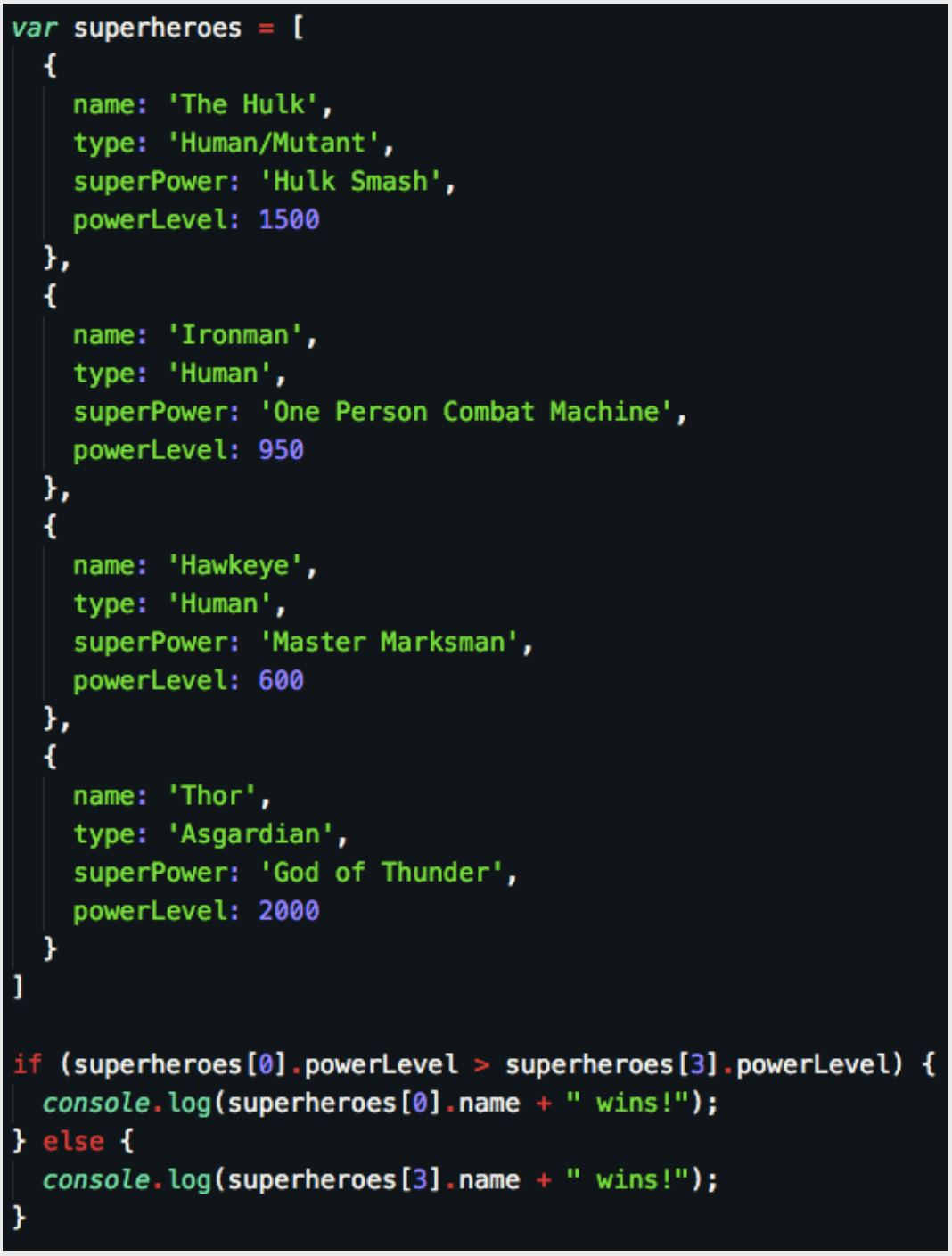
**JAVASCRIPT BASICS - Given the following array of objects in the image above, write out the syntax using dot notation to access the second element in the array and print out all of its properties:**

name

type

superPower

42.



```
var superheroes = [
  {
    name: 'The Hulk',
    type: 'Human/Mutant',
    superPower: 'Hulk Smash',
    powerLevel: 1500
  },
  {
    name: 'Ironman',
    type: 'Human',
    superPower: 'One Person Combat Machine',
    powerLevel: 950
  },
  {
    name: 'Hawkeye',
    type: 'Human',
    superPower: 'Master Marksman',
    powerLevel: 600
  },
  {
    name: 'Thor',
    type: 'Asgardian',
    superPower: 'God of Thunder',
    powerLevel: 2000
  }
]

if (superheroes[0].powerLevel > superheroes[3].powerLevel) {
  console.log(superheroes[0].name + " wins!");
} else {
  console.log(superheroes[3].name + " wins!");
}
```

**JAVASCRIPT BASICS - Given the array and if-else conditional above, what would be printed out to the console?**

43.

```
var universe = 'Star Wars';

switch (universe) {
  case 'DC':
    console.log('Spiderman lives here!');
    break;
  case 'Marvel':
    console.log('The X-Mean live here!');
    break;
  case 'Star Wars':
    console.log('The Jedi live here!');
    break;
  case 'Star Trek':
    console.log('The Federation lives here!');
    break;
  default:
    console.log('Everyone lives here!');
}
```

**JAVASCRIPT BASICS - Given the switch-case statement above, what would be printed to the console?**

44. **JAVASCRIPT BASICS - What boolean value does 2 === “2” evaluate to?**

45. **JAVASCRIPT BASICS - What boolean value does 3 !== 3 evaluate to?**

46.

```
var superheroes = [
  {
    name: 'The Hulk',
    type: 'Human/Mutant',
    superPower: 'Hulk Smash',
    powerLevel: 1500
  },
  {
    name: 'Ironman',
    type: 'Human',
    superPower: 'One Person Combat Machine',
    powerLevel: 950
  },
  {
    name: 'Hawkeye',
    type: 'Human',
    superPower: 'Master Marksman',
    powerLevel: 600
  },
  {
    name: 'Thor',
    type: 'Asgardian',
    superPower: 'God of Thunder',
    powerLevel: 2000
  },
  {
    name: 'Loki',
    type: 'Asgardian',
    superPower: 'Mischief',
    powerLevel: 2000
  }
]
```

```
superheroes[3].powerLevel <= superheroes[4].powerLevel ? console.log('The conflict ensues') : console.log('The battle stops here!');
```

**JAVASCRIPT BASICS - Given the array and ternary if-else conditional above, what would be printed to the console?**

47. **JAVASCRIPT BASICS - What would 2 + 5 evaluate to?**

48. **JAVASCRIPT BASICS - What would 200 - 100 evaluate to?**

49. **JAVASCRIPT BASICS - What would 10 * 50 evaluate to?**

50. **JAVASCRIPT BASICS - What would 60 / 12 evaluate to?**

51. **JAVASCRIPT BASICS - What would 22 % 10 evaluate to?**

52.
```
var counter = 10;

counter++

console.log(counter)
```

**JAVASCRIPT BASICS - Given the following scenario above, what would be printed to the console?**

53.
```
var counter = 20;

counter--;

console.log(counter);
```

**JAVASCRIPT BASICS - Given the scenario above, what would be printed to the console?**

54. **JAVASCRIPT BASICS - Given the following *for* loop:**

```
var jedi = ['Yoda', 'Mace Windu', 'Qui-Gon', 'Obi-Wan', 'Luke'];

for (____;_____;_____) {
  console.log(jedi[k]);
};
```

**What would be inside the parentheses of the *for* loop?**

55. **JAVASCRIPT BASICS - Using VanillaJS, what would you type to add an *Event Listener* to an element with *id="myButton"* on the DOM for a click event and using an anonymous function as a callback, have it output an alert with the text "Hello World!"?**

56. **JAVASCRIPT BASICS - Using jQuery, what would you type to add an *Event Listener* to an element with *id="myButton"* on the DOM for a click event and using an anonymous function as a callback, have it output an alert with the text "Hello World!"?**

57. **JAVASCRIPT BASICS - Traversing the DOM with VanillaJS: Given elements on the DOM with *class="images"*, write out how you would select those elements with VanillaJS and assign it to a variable named *elementz*.**

58. **JAVASCRIPT BASICS - Traversing the DOM with VanillaJS: Given an element on the DOM with an *id="menu-slider"*, write out how you would select that element with VanillaJS and assign it to a variable named *coolSlider*.**

59. **JAVASCRIPT BASICS - Traversing the DOM with jQuery: Given an element on the DOM with *class="article-section"*, write out how you would get all the elements contained within it.**

60. **JAVASCRIPT BASICS - Traversing the DOM with jQuery: Given an element on the DOM with an *id="win-count"*, write out how you would get the single *p* tag contained within it if you don't know where it's positioned within it.**

61. **JAVASCRIPT BASICS - Manipulating DOM element attributes:**

```
<a class="link" href="http://cnn.com">CNN</a>
```

Using VanillaJS and given the element above, write out how you would change the href attribute to the value "http://inc.com".

Using jQuery and given the same anchor tag element above, write out how you would change the href attribute to the value "http://inc.com".

62. **JAVASCRIPT BASICS - Manipulating DOM element styles**

```
<div id="like-counter"></div>
```

Using VanillaJS and given the element above, write out how you would change the width of the div to 100px.

Using jQuery and given the same element above, write out how you would change the width of the div to 100px.

63. **JAVASCRIPT BASICS - Callback functions:**

```
var foo = function() {
  // ...some JS code
};

$(document).on("click", "button", _____);
```

Fill in the blank line below to complete the callback function using a function expression.

64. **JAVASCRIPT BASICS - Callback functions:**

```
$(document).on("click", "button", _____);
```

Given the event listener, fill in the blank line to complete the callback function using an anonymous function (just make it an empty block).

65. **APPLICATION PROGRAMMING INTERFACE - CONSUMING - Using jQuery to make an ajax call to the Github API to retrieve data, fill in the blank: (Read blank lines left to right, top-down).**

```
$.ajax({
  method: _____,
  url: 'http://api.github.com/username/repos',
})._____(function(data, status, jqXHR) {
  // some JS code to access manipulate the data variable
})._____(function(jqXHR, status, error) {
  // some JS code to show the error
});
```

Blank Line 1

Blank Line 2

Blank Line 3

66. **FIREBASE - Given the code in the image below, complete the blanks to create a reference to a Firebase database so we can store data:**

```
<!-- Firebase with saving data -->
<!DOCTYPE html>
<html>
  <head>
    <title>My Ninja App</title>
    <script src="https://cdn.firebase.com/js/client/2.4.1/firebase.js"></script>
  </head>
  <body>
    <!-- A bunch of HTML -->
    <script type="text/javascript">
      var ref = __(0)__ _____(1)_____("ninja-catalog.firebaseIO.com");

      // Initial Values
      var nickName = "";
      var weapon = "";
      var powerLevel = 0;

      // Capture Button Click
      $("#addUser").on("click", function() {

        // Grabbed values from text boxes
        nickName = $('#nickNameInput').val().trim();
        weapon = $('#weaponInput').val().trim();
        powerLevel = $('#powerLevelInput').val().trim();

        // Code for pushing our data to our Firebase database
        ref.__(2)__({
          nickName: nickName,
          weapon: weapon,
          powerLevel: powerLevel
        })

        // Don't refresh the page!
        return false;
      });
    </script>
  </body>
</html>
```

(0) _____

(1) _____

(2) _____

67. **FIREBASE - Given the code in the image below, complete the blanks to listen to your Firebase database when your data changes and have it reflect on the DOM:**

```
  // Don't refresh the page!
  return false;
});

//Firebase watcher + initial loader
ref._(3)_("___(4)___", function(snapshot) {

  // Log everything that's coming out of snapshot
  console.log(snapshot.val());
  console.log(snapshot.val().nickName);
  console.log(snapshot.val().weapon);
  console.log(snapshot.val().powerLevel);

  // Change the HTML to reflect using jQuery
  (5)_("#nickNameDisplay").html(snapshot.val().nickName);
  (6)_("#weaponDisplay").html(snapshot.val().weapon);
  (7)_("#powerLevelDisplay").html(snapshot.val().powerLevel);


// Error handling
}, function(errorObject) {

  console.log("Errors handled: " + errorObject.code)
});
```

(3) _____

(4) _____

(5) _____

(6) _____

(7) _____

68. **FIREBASE - Given the code in the image below, complete the blanks to listen to your Firebase database when a new child is added:**

```
//Firebase watcher + initial loader
ref.__(8)__("___(9)___", function(childSnapshot) {
  // Log everything that's coming out of snapshot
  console.log(childSnapshot.val().nickName);
  console.log(childSnapshot.val().weapon);
  console.log(childSnapshot.val().powerLevel);


// Handle the errors
}, function(errorObject){
  console.log("Errors handled: " + errorObject.code);
});
```

(8) _____

(9) _____

69. Please check this when you're ready to submit. *

- [ ] I'm done :)