Siobhan Lynch
WK4 Homework Answers

Part 1:

1. **What are loops *great for*? Give *two reasons* why they are great.**

   They are great for when you want the same task to run the same code per and over again where each time it has different values. They are also great when working with arrays.

2. **What is the basic syntax for a for loop? Give the higher level (more complexly worded syntax) and the more basic syntax.**

   Higher level syntax: for(initialization; condition, final-expression) {statement}
   Basic syntax: for (step 1; step 2; step 3){//code block to be executed}

3. **What does the for loop *basically do*?**

   The for loop loops through a block of code a specified number of times.

4. **Give an example of a for loop, and then *explain* what is going on with the code.**

```
var cars = ["BMW", "Volvo", "Saab", "Ford", "Fiat", "Audi"];
var text = "";
var i;
for (i = 0; i < cars.length; i++) {
  text = cars + ", ";
}
console.log(text)
```

   the variable cars is being defined with 6 values. the variable text has an undefined value. the variable i has no value. the for loop initializes i to equal 0, the condition is that i is less than the cars length, the final expression, i++, will add one (+1) to the operand, and then return a value. An operand is the quantity on which an operation is to be done. in this case it has no effect. The statement takes the variable text and makes it equal to cars and adds a comma and a space between each value of cars. console.log(text) executes the value of the variable text.

5. **What two characters (used in *code*) represent the for loop *code block*?**

The 2 characters to represent the fro loop code block are the curly brackets that hold the **{ statement }**

6. **After the for loop code block has been *executed*, which *step* does the for loop program *return* to?**

   it returns to step 2, the condition.

7. **Give an example of a for loop that *includes* an if statement and a break statement *within* the if statement. *Don't* give the exact example in the loops-arrays slide deck. I *will not give* you credit if you do. *Customize* it. *Make it* your own. I will *provide links* at the end of this document to *other places* where you can *get* more ideas for examples.**

```
for (let i = 10; i < 27; i++) {
   if (i > 20) {
      console.log(
         `The loop has stopped iterating because i reached the value of ${i}!`
      );
   break;
   }
   console.log(i);
}
```

   (I tried my best to mix it up from the code you had. but it was with numbers so there wasnt much to go on. but i think i did okay. let me know if its not.)

8. **Give an example of a for loop that *includes* an if statement and a continue statement *within* the if statement. *Don't* give the exact example in the loops-arrays slide deck. I *will not give* you credit if you do. *Customize* it. Style it with the *help* of CSS in JS (refer to the loops-arrays slide deck to follow how I do it; you can also reach out to me for help!). *Make it* your own. I will *provide links* at the end of this document to *other places* where you can *get* more ideas for examples.**

   number = 0 for number in range(10): if number == 5: continue print('Number is ' + str(number)) print('Out of loop')

```
function myFunction() {
 var text = "";
 var i;
```

```
for (i = 0; i < 5; i++) {
  if (i === 3) {
    continue;
  }
  text += "The number is " + i + "<br>";
}
```

9. **Give me an example of an *array*. You can use *whatever* data type you *want* inside. Use the examples in the slide deck as *guidelines*, but make up your own.**

```
let colorArray = ["red", "blue", "yellow", "green", "orange", "purple", "pink", "brown"]
console.log(colorArray[0]);
console.log(colorArray[1]);
console.log(colorArray[2]);
console.log(colorArray[3]);
console.log(colorArray[4]);
console.log(colorArray[5]);
console.log(colorArray[6]);
console.log(colorArray[7]);
```

10. **Give me another example of an *array* using a DIFFFERENT data type from number 9. You can use *more than one* if you like as well! Get creative. Use the examples in the slide deck as *guidelines*, but make up your own.**

```
let cats = [{
   type: "russian",
   color: "blue",
   eyes: 2,
   whiskers: 8,
   ears: "tipped"
}, {
   type: "calico",
   color: "orange and white",
   eyes: 1,
   whiskers: 6,
   ears: "intact"
}]

console.log(cats[0].type)
console.log(cats[1].eyes)
```

(poor cat has 1 eye :[ )

11. **Declare and *initialize* and array (remember, *initializing* means applying a value to the *variable*, in this case the value of the variable would be an array). Then *create* a for loop which *iterates* over that array. Use the loops-arrays slide deck as a *guide* to how to do this, and you can also visit the** JavaScript For Loop page on W3Schools, The JavaScript For Statement page on WsSchools, the Loops and Iteration page on MDN, and ***other links*** provided in the **helpful reading section** at the ***end*** of this **document**.

```
var dogs = ['Pug', 'Corgi', 'Laberdoodle'];
var x;

for (x of dogs) {
  document.write(x + ", ");
}
```

12. **Give an example of a *for in* loop. Explain what is *going on* with the code. Use the loops-arrays slide deck as an *inspiration* and/or guide to creating your for-in loop, the related resources I have included at the *end* of the slide deck there, along with the *other links* I have provided in the Helpful Reading section at the end of this document. but make it your own! Get *creative*. Again, somehow involve CSS in JS in a *similar way* to the way you did it for number 8.**

```
Object.prototype.objCustom = function() {};
Array.prototype.arrCustom = function() {};

const iterable = [3, 5, 7];
iterable.foo = 'hello';

for (const i in iterable) {
console.log(i); // logs 0, 1, 2, "foo", "arrCustom", "objCustom"
}
for (const i in iterable) {
if (iterable.hasOwnProperty(i)) {
          console.log(i); // logs 0, 1, 2, "foo"
                    }
          }
for (const i of iterable) {
 console.log(i); // logs 3, 5, 7
}
```

13. **Give an example of a for of loop. Explain what is *going on* with the code. Use the loops-arrays slide deck as an *inspiration* and/or guide to creating your for-in loop, the**

related resources I have included at the *end* of the slide deck there, along with the *other links* I have provided in the Helpful Reading section at the end of this document.

```
const array1 = ['a', 'b', 'c'];

for (const element of array1) {
  console.log(element);
}
```

14. Tell me what is the (*main*) difference *between* a for in loop and a for of loop.

The for in loop iterates over all the enumerable properties of an object that are keyed by Strings as opposed to keyed by Symbols which are ignored.
The for of statement creates a loop that iterates over iterable objects.

15. Give me an example of a do while loop. Use the loops-arrays slide deck as an *inspiration* and/or guide to creating your for-in loop, the related resources I have included at the *end* of the slide deck there, along with the *other links* I have provided in the Helpful Reading section at the end of this document.

```
let result = "";
let i = 0;

do {
  i = i + 1;
  result = result + i;
} while (i < 5);

console.log(result);
```

16. Why does a loop (in general) *terminate*?

it terminates when a break statement is used which halts the loops execution completely and the conditions previously stated is no longer true.