Sohee Cho

Part 1:

1. **What are loops *great for*? Give *two reasons* why they are great.**
    a. It's great when you want to do the same task/run the same code over again, and each time with a different value
    b. It's 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.**
    a. The basic syntax
        i. for(initialization; condition; final-expression) { statement }
        ii. for (step 1; step 2; step 3) { // code block to be executed }
3. **What does the for loop *basically do*?**
    a. The "for loop" provides a concise way of writing the loop structure. Unlike a while loop, a for statement consumes the initialization, condition and increment/decrement in one line thereby providing a shorter, easy to debug structure of looping.
4. **Give an example of a for loop, and then *explain* what is going on with the code.**
    a. Example: let text = ''; for (i = 0; i < 5; i++) { text += "The number is " + i + "
"; }
        i. Statement 1 sets a variable before the loop starts (var i = 0)
        ii. Statement 2 defines the condition for the loop to run (i must be less than 5)
        iii. Statement 3 increases a value (i++) each time the code block in the loop had been executed
5. **What two characters (used in *code*) represent the for loop *code block*?**

a. The two characters used to represent the for loop code block are curly brackets { }

6. **After the for loop code block has been *executed*, which *step* does the for loop program *return* to?**
    a. It returns to step 2, in which we define the condition for executing the code block.
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.**
    a. Example:

```
let colorPantone = ' ';
for (let colorOptions = 1; colorOptions < 100;
colorOptions++) {
    if (colorOptions > 5) {
document.querySelector(".colorwheel").innerHTML =
colorPantone;
    break; }
    colorPantone += 'You have picked ${colorOptions}
color variations!'; }
Document.querySelector(".colorwheel").innerHTML =
colorPantone
```

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.

a. Example:

```
let farm = ' ';
for (potatoes = 1; potatoes <20; potatoes++) {
    if (potatoes === 5) {continue;}
    farm +='You have ${potatoes} planted in your farm!'; }
const potatoesPlanted = document.querySelector(".farm");
potatoesPlanted.innerHTML = field;
```

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.**
   a. Example: let statesArray = ["newyork", "pennsylvania", "california", "texas", "massachusettes"]

```
let states = [
{
    type: "newyork"
    location: "east",
    capital: "newyorkcity",
    population: "9 million",
},
{
    type: "pennsylvania",
    location: "east",
```

```
    capital: "philadelphia",
    population: "13million",
}
];
```

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.**

    a. Example: let cunyArray = ["citytech", "baruch", "brooklyn", "graduatecenter", "hunter", "citycollege"]

```
let cuny = [{
    type: "citytech"
    location: "brooklyn",
    graduationrate: "20.8%",
    enrollment: 16214,
}, {
    type: "baruch",
    location: "newyorkcity",
    graduationrate: "72%",
    enrollment: 14903,
}]
```

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**.

```
let fruit = [{
    type: "apple",
    color: "red",
    radio: true
}, {
    type: "banana",
    color: "yellow",
    radio: false
}]
for(let i = 0; i < fruit.types; i++) {
    const color = fruit[i].color;
    const type = fruit[i].type;
    const radio = fruit[i].radio;


    console.log(`${color} ${type} ${radio}`);
}
```

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.**

HTML:

```
<button onclick="myFunction()">Click Here</button>
```

Javascript:

```
function myFunction() {
  var person = {fname:"Sohee", lname:"Cho", age:26};
  var text = "";
  var x;

  for (x in person) {
    text += person[x] + " "; }
  document.getElementById("demo").innerHTML = text;}
```

In this example, var is the variable which iterates over the properties of an object. The "(x in person)" is the object which is iterated.

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.**

```
<html>
<body>

<script>
var students = ['Sohee', 'Dasom', 'Abundio'];
var x;

for (x of students) {
  document.write(x + "<br >");
}
```

```
</script>

</body>
</html>
```

In this example, "(x of students)" is the variable. For every iteration the value of the next property is assigned to the variable and is declared by var in this case "var students"

14. **Tell me what is the (*main*) difference *between* a for in loop and a for of loop.**
    a. The main difference a for in loop and for of loop is that "for in" is a method for iterating over enumerable properties of an object, which is any property that you add to an object.
    b. The "for of" loop only works through 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.**

```
<button onclick="myFunction()">Click Here</button>

<p id="demo"></p>

<script>
function myFunction() {
  var text = "Lucky numbers:"
  var i = 10;
  do {
    text += "<br>Your number is " + i;
```

```
    i++;
  }
  while (i < 20);
  document.getElementById("demo").innerHTML = text;
}
</script>

</body>
</html>
```

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

a. It terminates because the condition is no longer true.