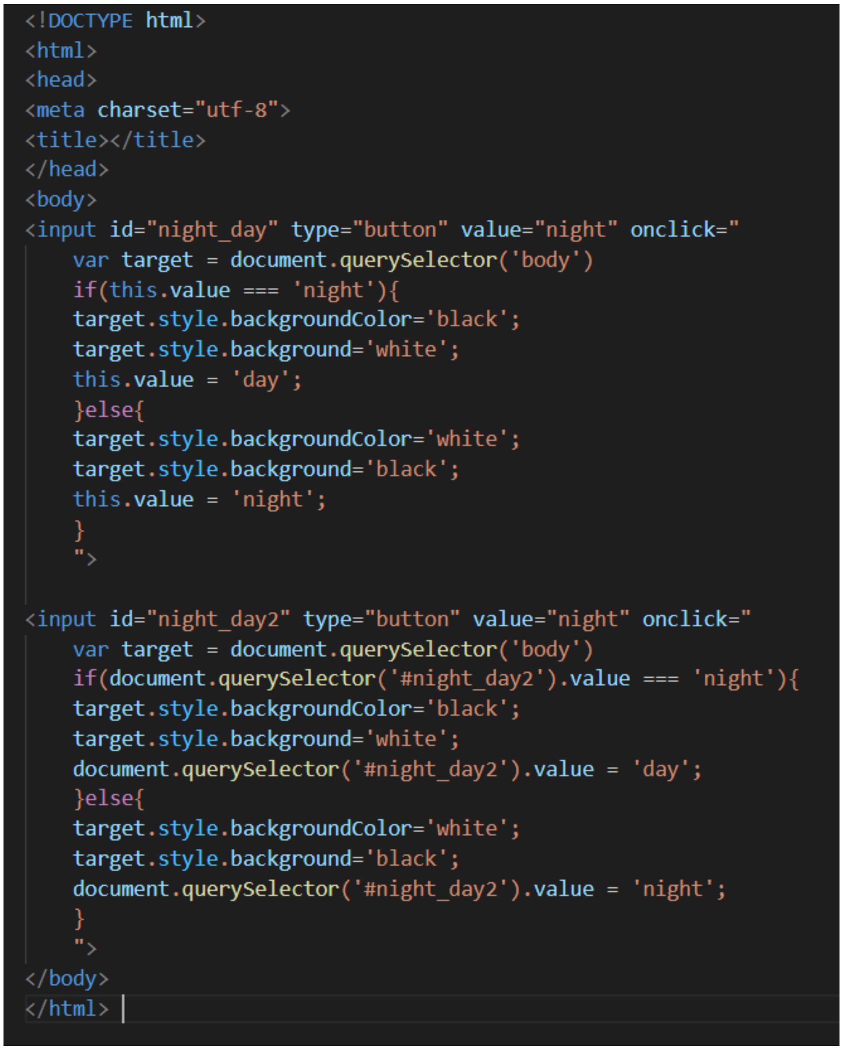
EAST TechSprint Log:

Use this document to log your work for the TechSprint period. Remember, your log should reflect adequate effort to achieve growth in the skills you are learning. This work is to be done **in addition** to any work that you are doing or skills you are developing for your EAST Project (although the work can be complementary). Each log entry should follow the following format:

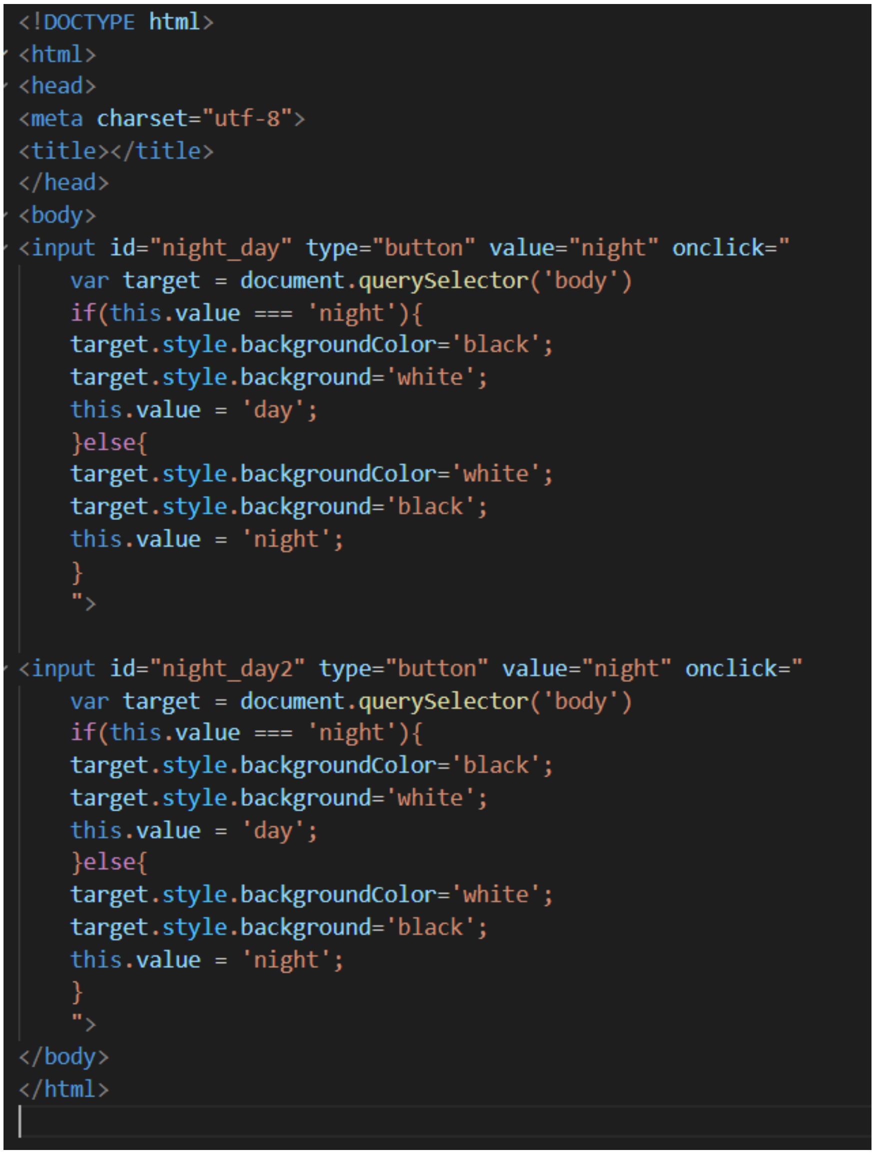
- Date
- Description of learning goal or task
- Link to tutorial resource(s)
- Example of **your work** based on the tutorial and goals (screenshot preferred), embedded in this document
- Reflection on what you learned, challenges or other important elements

This log will be due by 11:59 PM on the Sunday following the TechSprint period.

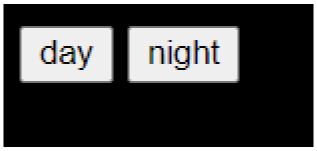
Date: 5/29
Task: Learning about Refactoring in JavaScript
Link to Resource:
https://www.youtube.com/watch?v=vwRkFRke7ls&t=11s
https://www.w3schools.com/js/js_this.asp



```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title></title>
</head>
<body>
<input id="night_day" type="button" value="night" onclick="
    var target = document.querySelector('body')
    if(this.value === 'night'){
    target.style.backgroundColor='black';
    target.style.background='white';
    this.value = 'day';
    }else{
    target.style.backgroundColor='white';
    target.style.background='black';
    this.value = 'night';
    }
    ">

<input id="night_day2" type="button" value="night" onclick="
    var target = document.querySelector('body')
    if(document.querySelector('#night_day2').value === 'night'){
    target.style.backgroundColor='black';
    target.style.background='white';
    document.querySelector('#night_day2').value = 'day';
    }else{
    target.style.backgroundColor='white';
    target.style.background='black';
    document.querySelector('#night_day2').value = 'night';
    }
    ">
</body>
</html>
```



```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title></title>
</head>
<body>
<input id="night_day" type="button" value="night" onclick="
    var target = document.querySelector('body')
    if(this.value === 'night'){
    target.style.backgroundColor='black';
    target.style.background='white';
    this.value = 'day';
    }else{
    target.style.backgroundColor='white';
    target.style.background='black';
    this.value = 'night';
    }
    ">

<input id="night_day2" type="button" value="night" onclick="
    var target = document.querySelector('body')
    if(this.value === 'night'){
    target.style.backgroundColor='black';
    target.style.background='white';
    this.value = 'day';
    }else{
    target.style.backgroundColor='white';
    target.style.background='black';
    this.value = 'night';
    }
    ">
</body>
</html>
```



Reflection:

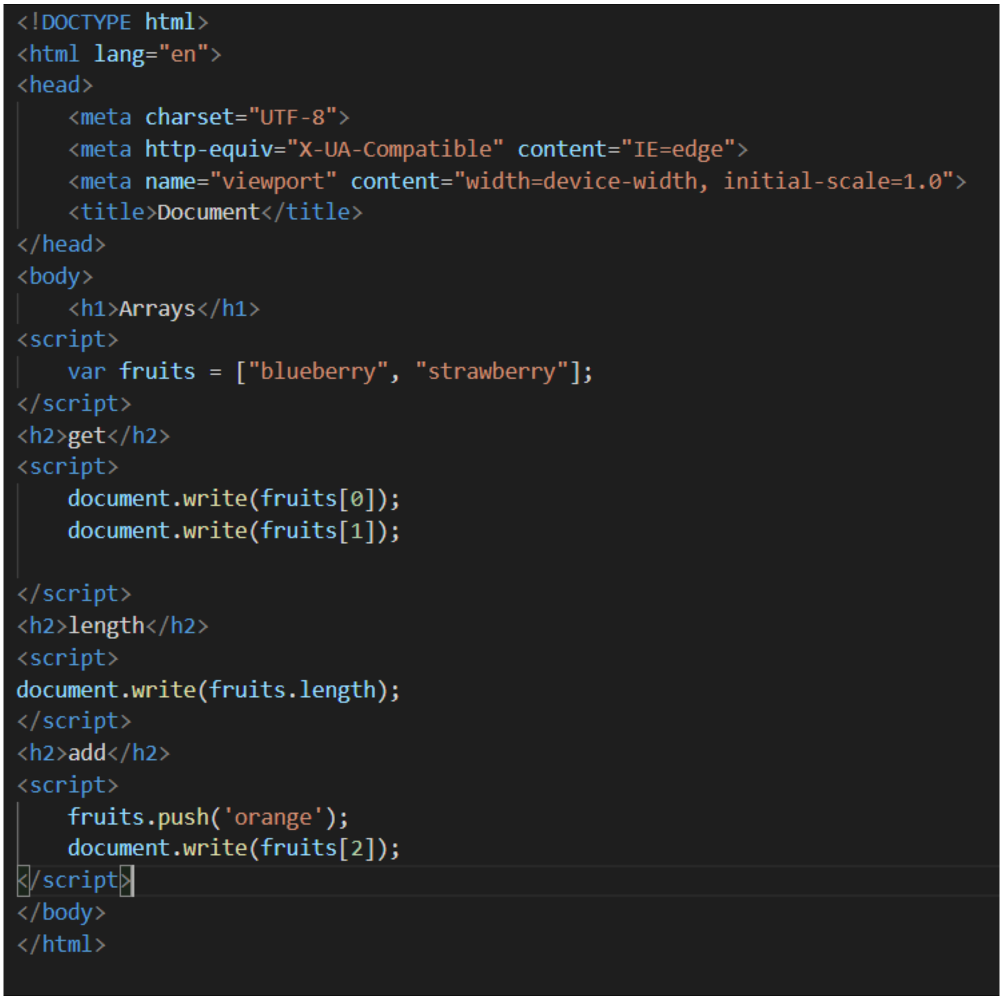
Refactoring is to restructure the source code in order to improve operation without altering functionality.

It is important because it improves the legibility and readability of the codes; other programmers might need to change the codes or take a look at the codes.
The code should be easy for everyone to understand, not only the person who writes it.
Therefore, the names of variables should be decided under consideration, and codes should be organized.

“this” is a JavaScript keyword which means “document.querySelector(#id of the upper tag)” in each tag in order to get the id value of the tags without repetition or mentioning them every time.
It avoids duplicates as well as enables the set of code to work properly when copying and pasting it.

With the less repetition, the better the program will work.

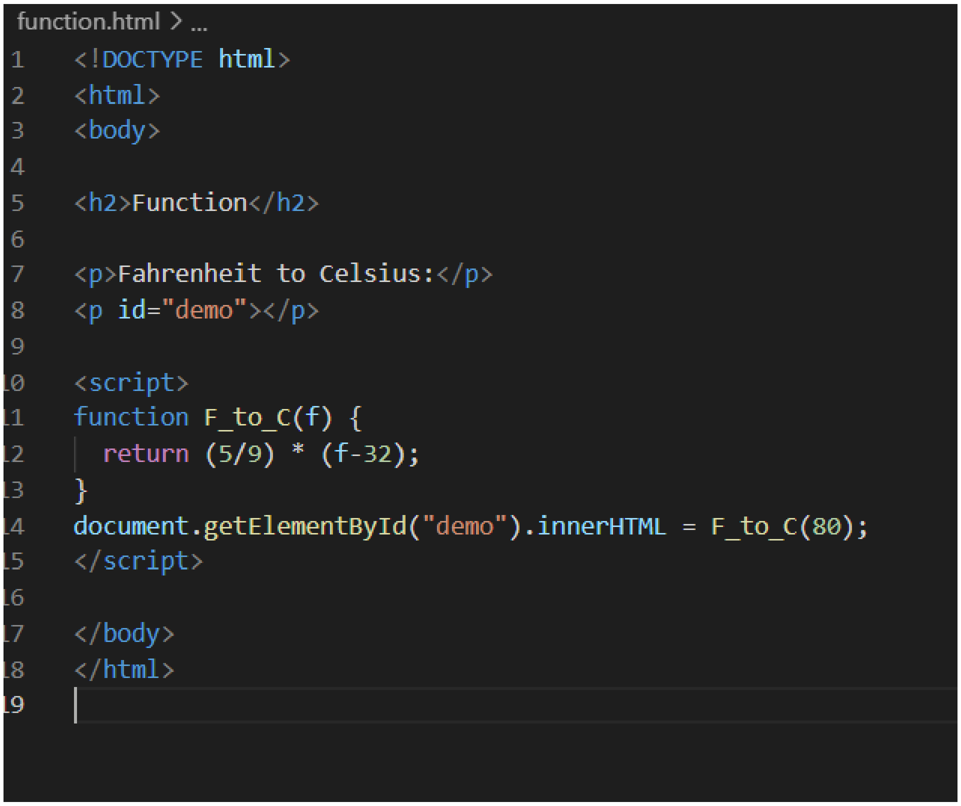
Date: 5/29
Task: Learning about Arrays in JavaScript
Link to Resource:
https://www.youtube.com/watch?v=ANDqF-NlWqQ&t=206s
https://www.w3schools.com/jsref/jsref_length_array.asp
https://www.w3schools.com/jsref/jsref_push.asp



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1>Arrays</h1>
<script>
    var fruits = ["blueberry", "strawberry"];
</script>
<h2>get</h2>
<script>
    document.write(fruits[0]);
    document.write(fruits[1]);

</script>
<h2>length</h2>
<script>
document.write(fruits.length);
</script>
<h2>add</h2>
<script>
    fruits.push('orange');
    document.write(fruits[2]);
</script>
</body>
</html>
```

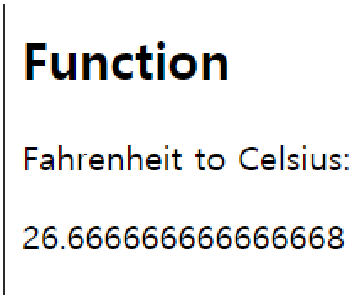
Reflection:
When using ‘document.write(),’ it is possible to bring elements in an array to the screen.
When adding elements in an array, “array name.push()” is used.
When counting elements in an array, “array name.length” is used.

Date: 5/29
Task: Learning about Function in JavaScript
Link to Resource:
https://www.youtube.com/watch?v=WpTdoEvmFxw
https://www.youtube.com/watch?v=IOuePUzLdnQ
https://www.w3schools.com/js/js_functions.asp



function.html > ...

```
<!DOCTYPE html>
<html>
<body>

<h2>Function</h2>

<p>Fahrenheit to Celsius:</p>
<p id="demo"></p>

<script>
function F_to_C(f) {
  return (5/9) * (f-32);
}
document.getElementById("demo").innerHTML = F_to_C(80);
</script>

</body>
</html>

```

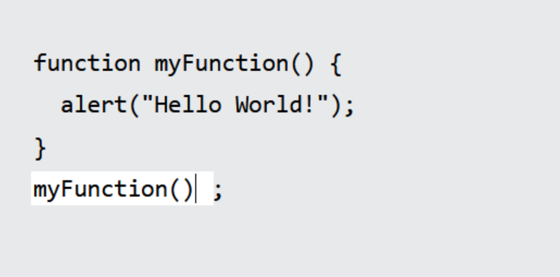


Function

Fahrenheit to Celsius:

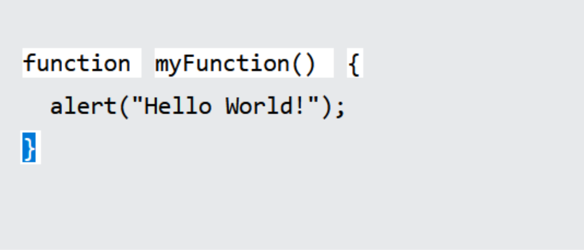
26.666666666666668

Execute the function named `myFunction`.



```
function myFunction() {
  alert("Hello World!");
}
myFunction() ;
```

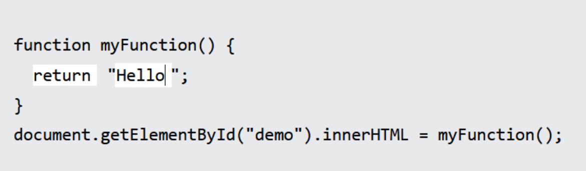
Create a function called "myFunction".



```
function myFunction() {
  alert("Hello World!");
}
```

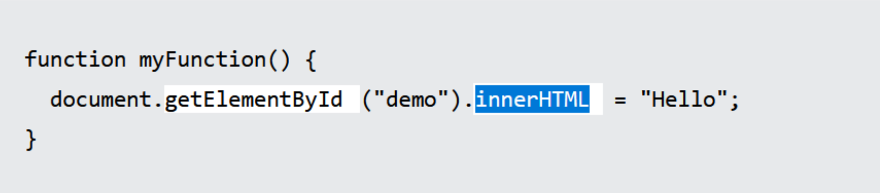
Exercise:

Make the function return "Hello".



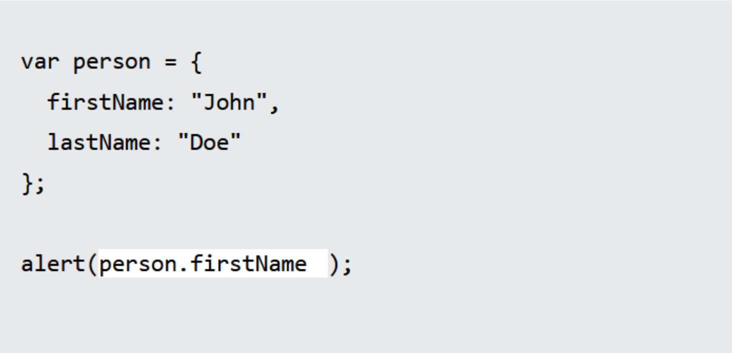
```
function myFunction() {
  return "Hello";
}
document.getElementById("demo").innerHTML = myFunction();
```

Make the function display "Hello" in the inner HTML of an element with the ID "demo".



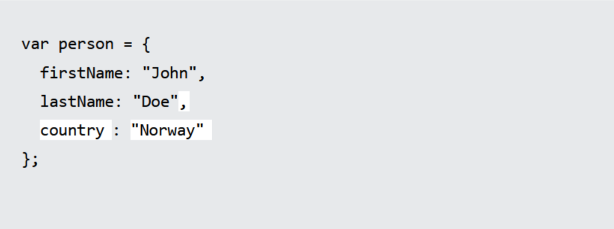
```
function myFunction() {
  document.getElementById("demo").innerHTML = "Hello";
}
```

Alert "John" by extracting information from the person object.

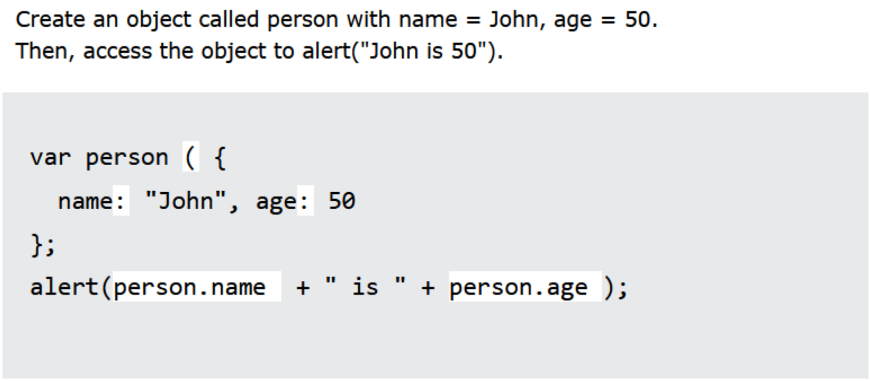


```
var person = {
  firstName: "John",
  lastName: "Doe"
};

alert(person.firstName);
```

Add the following property and value to the person object: country: Norway.



```
var person = {
  firstName: "John",
  lastName: "Doe",
  country : "Norway"
};
```



Create an object called person with name = John, age = 50.
Then, access the object to alert("John is 50").

```
var person ( {
  name: "John", age: 50
};
alert(person.name  + " is " + person.age );
```

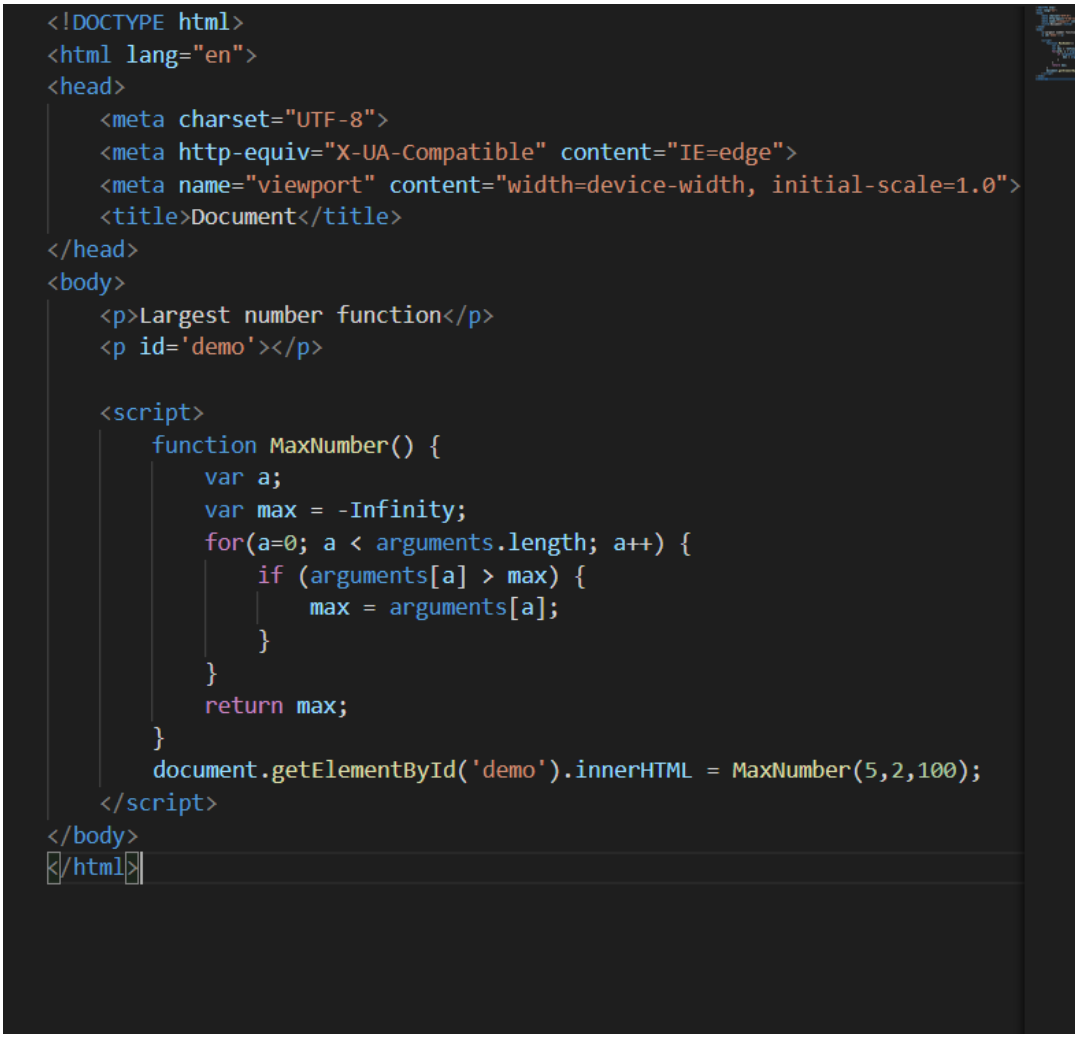
Reflection:

There are two ways to create a function in JavaScript:

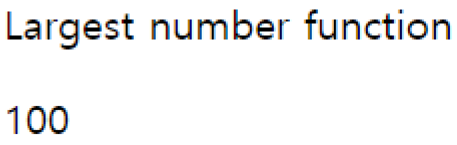
1) Function function_name() {}
2) Var variable_name({})

Functions make it possible to repeat the same pattern with changing values.

Date: 5/29
Task: Learning about Argument and Parameter of JavaScript Function
Link to Resource:
https://www.w3schools.com/js/js_function_parameters.asp
https://www.youtube.com/watch?v=6MzCHO8M3Uc
https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Global_Objects/Infinity
https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Functions/arguments



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <p>Largest number function</p>
    <p id='demo'></p>

    <script>
        function MaxNumber() {
            var a;
            var max = -Infinity;
            for(a=0; a < arguments.length; a++) {
                if (arguments[a] > max) {
                    max = arguments[a];
                }
            }
            return max;
        }
        document.getElementById('demo').innerHTML = MaxNumber(5,2,100);
    </script>
</body>
</html>
```



```
Largest number function

100
```

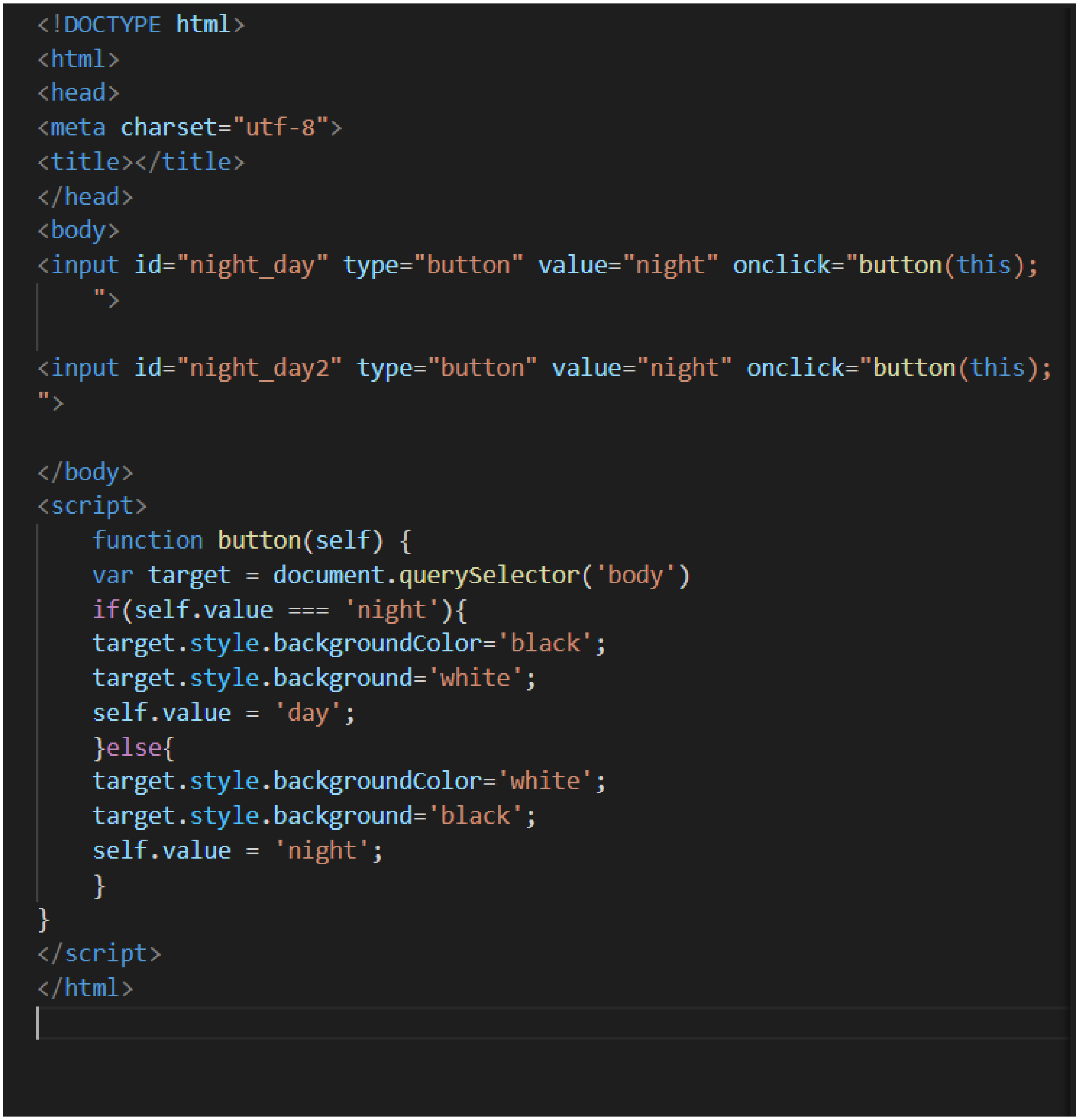
Reflection:
A parameter is a value that is given as an initial value for the function to work with.

‘Infinity’ means a number value of positive infinity as a default value.

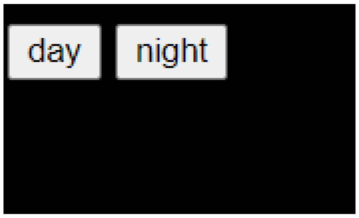
An argument is an array that constitutes a parameter.

When creating and using functions, it is efficient to utilize parameters to make it function as a logic or a formula that handles a variety set of values.

Date: 5/29
Task: Utilizing Functions in JavaScript
Link to Resource:
https://www.youtube.com/watch?v=WsPJ8FsoMcU

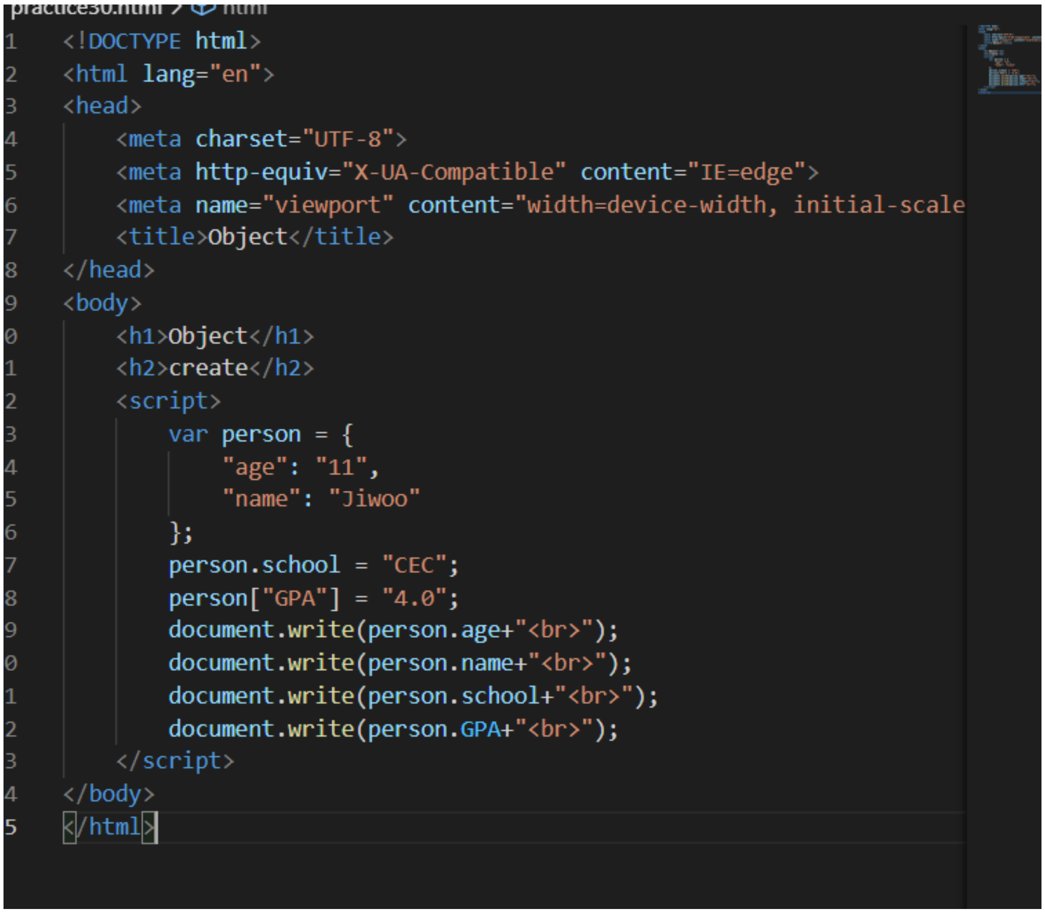


```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title></title>
</head>
<body>
<input id="night_day" type="button" value="night" onclick="button(this);
    ">

<input id="night_day2" type="button" value="night" onclick="button(this);
">

</body>
<script>
    function button(self) {
    var target = document.querySelector('body')
    if(self.value === 'night'){
    target.style.backgroundColor='black';
    target.style.background='white';
    self.value = 'day';
    }else{
    target.style.backgroundColor='white';
    target.style.background='black';
    self.value = 'night';
    }
}
</script>
</html>
```

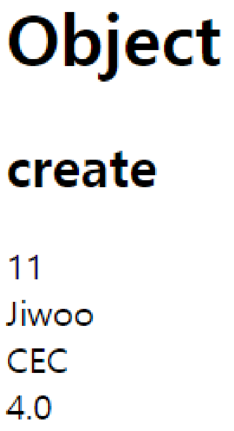


Reflection:
When using a function in which a parameter is required, it is possible to put ‘this’ as a parameter to a function.

Date: 5/29
Task: Learning about How to Write and Read an Object in JavaScript
LInk to Resource:
https://www.youtube.com/watch?v=2j04s1erzTs
https://www.w3schools.com/js/js_objects.asp



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale
    <title>Object</title>
</head>
<body>
    <h1>Object</h1>
    <h2>create</h2>
    <script>
        var person = {
            "age": "11",
            "name": "Jiwoo"
        };
        person.school = "CEC";
        person["GPA"] = "4.0";
        document.write(person.age+"<br>");
        document.write(person.name+"<br>");
        document.write(person.school+"<br>");
        document.write(person.GPA+"<br>");
    </script>
</body>
</html>
```



Object

create

```
11
Jiwoo
CEC
4.0
```

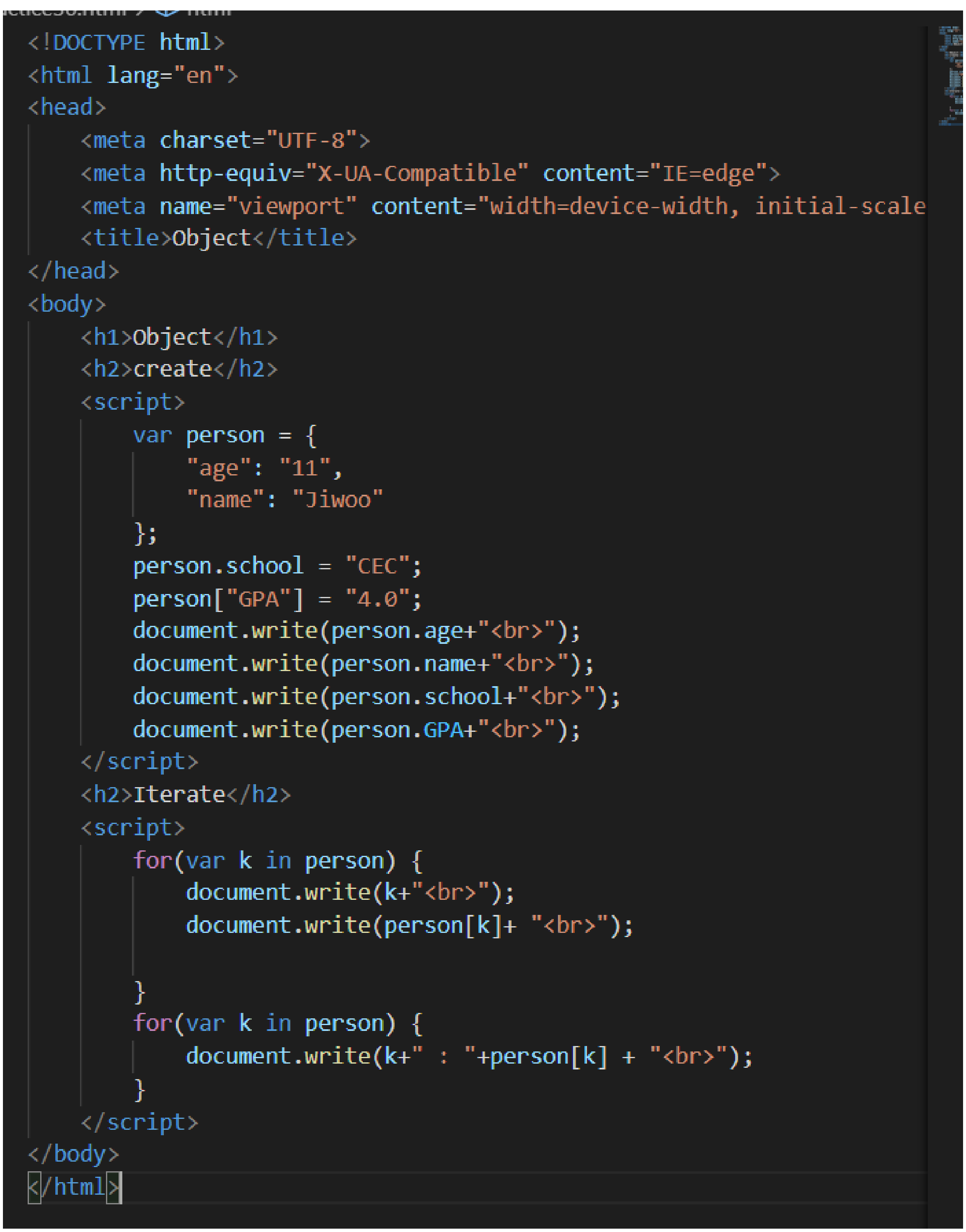
Reflection:
When creating a variable, it is effective to include properties and their corresponding values, making a variable into an object.

When writing adding a property, there are two ways:

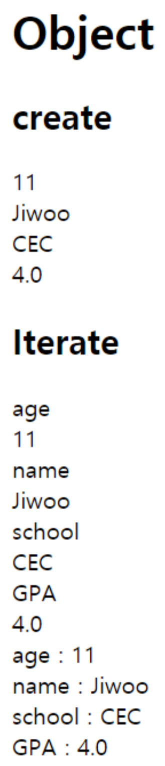
1) Variable.property_name = Value;
2) Variable[“property”] = value;

A method is a function stored as a property.

Date: 5/29
Task: Learning How to Create Loop Regarding Objects in JavaScript
Link to Resource:
https://www.youtube.com/watch?v=GIB8O6R3wko



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale
    <title>Object</title>
</head>
<body>
    <h1>Object</h1>
    <h2>create</h2>
    <script>
        var person = {
            "age": "11",
            "name": "Jiwoo"
        };
        person.school = "CEC";
        person["GPA"] = "4.0";
        document.write(person.age+"<br>");
        document.write(person.name+"<br>");
        document.write(person.school+"<br>");
        document.write(person.GPA+"<br>");
    </script>
    <h2>Iterate</h2>
    <script>
        for(var k in person) {
            document.write(k+"<br>");
            document.write(person[k]+ "<br>");

        }
        for(var k in person) {
            document.write(k+" : "+person[k] + "<br>");
        }
    </script>
</body>
</html>
```



Object

create

11
Jiwoo
CEC
4.0

Iterate

```
age
11
name
Jiwoo
school
CEC
GPA
4.0
age : 11
name : Jiwoo
school : CEC
GPA : 4.0
```

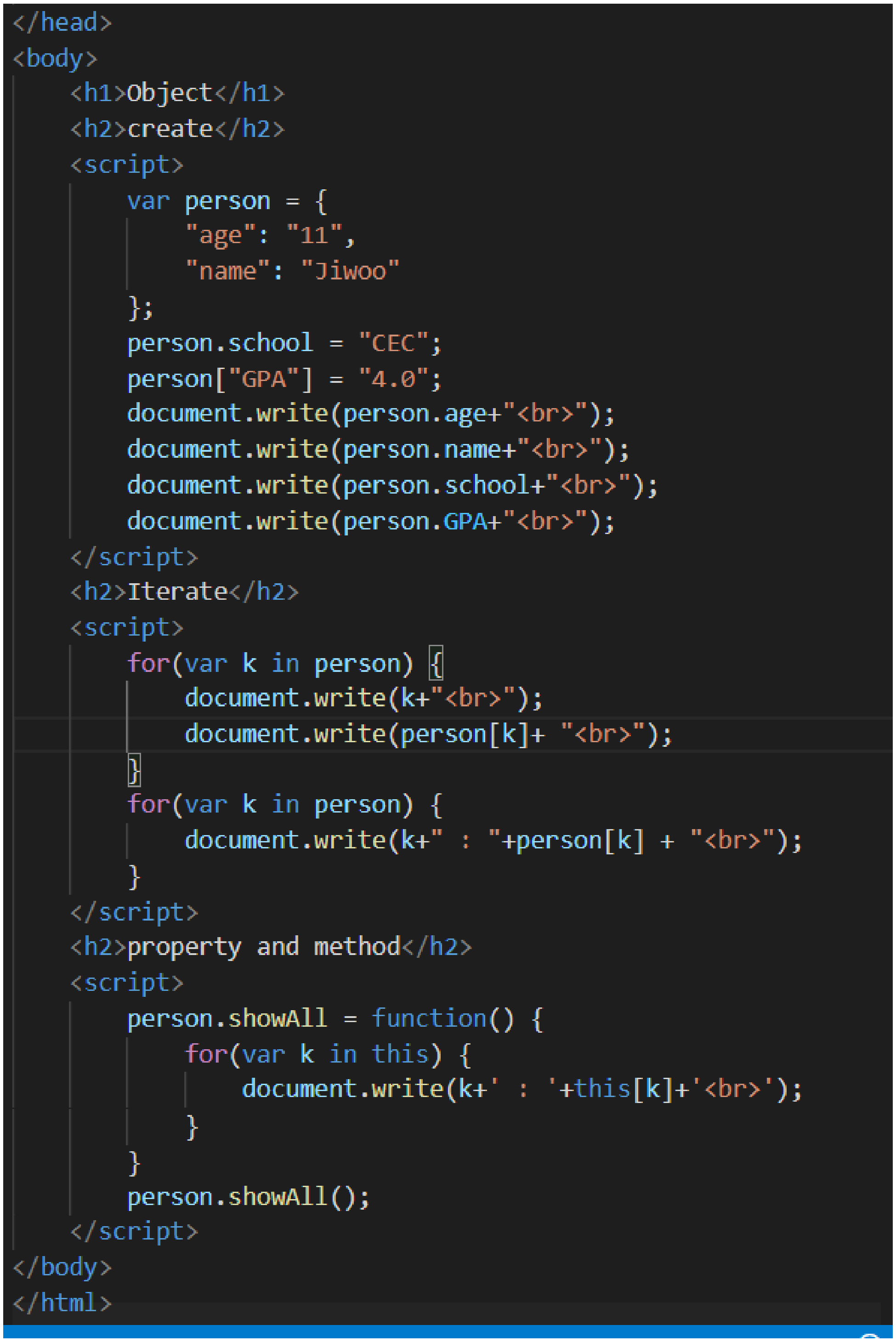
Reflection:
The properties in an object are called ‘keys,’ since they are the keys to approach each value in an object. On the other hand, they are called an index in arrays.

It is effective to use objects in order to repeat the same function with different values.

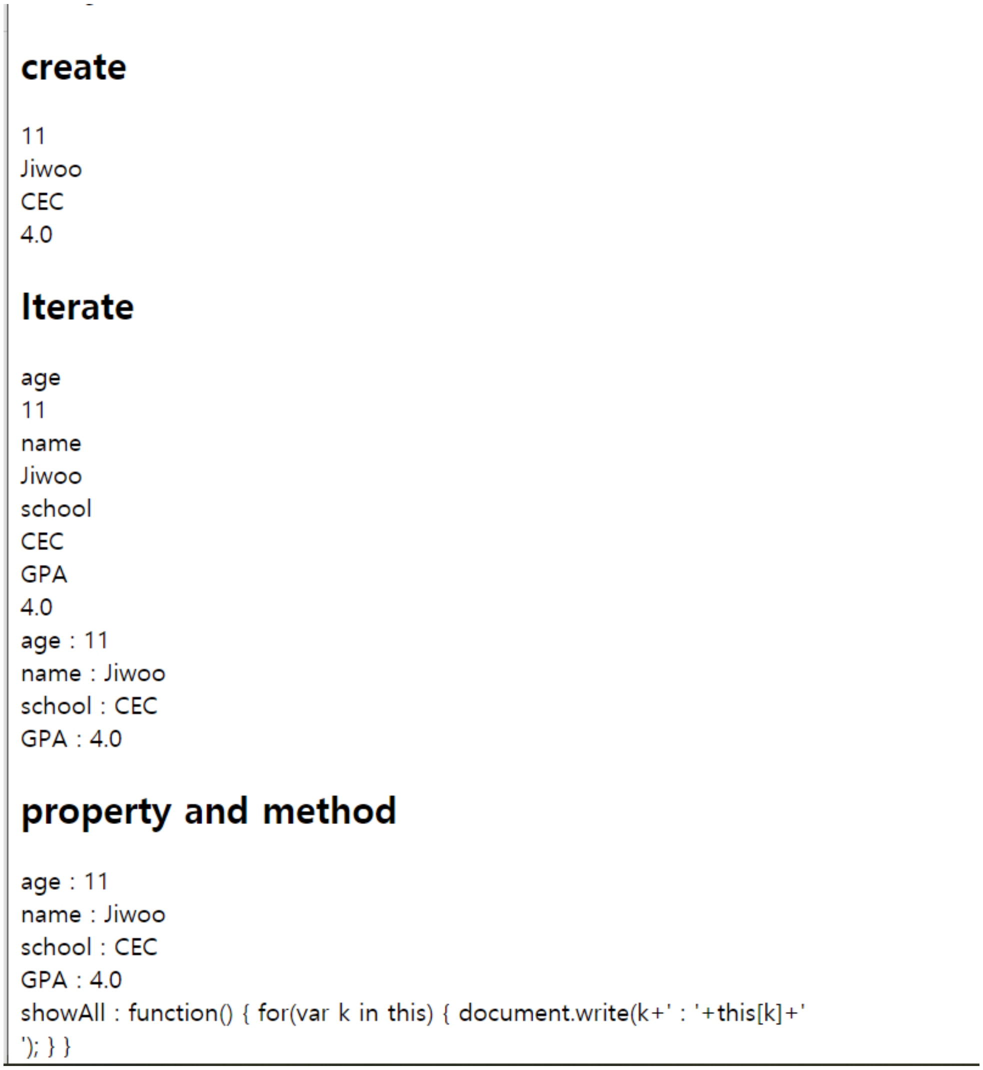
By using the loop, it is possible to write or manage every property and its corresponding value within a single code.

The combination of for loop and objects makes it easier and efficient to make the code changeable in different circumstances.

Date: 5/29
Task: Learning about Object Properties and Methods
Link to resource:
https://www.youtube.com/watch?v=mjzx1qbyPwU



```
</head>
<body>
    <h1>Object</h1>
    <h2>create</h2>
    <script>
        var person = {
            "age": "11",
            "name": "Jiwoo"
        };
        person.school = "CEC";
        person["GPA"] = "4.0";
        document.write(person.age+"<br>");
        document.write(person.name+"<br>");
        document.write(person.school+"<br>");
        document.write(person.GPA+"<br>");
    </script>
    <h2>Iterate</h2>
    <script>
        for(var k in person) {
            document.write(k+"<br>");
            document.write(person[k]+ "<br>");
        }
        for(var k in person) {
            document.write(k+" : "+person[k] + "<br>");
        }
    </script>
    <h2>property and method</h2>
    <script>
        person.showAll = function() {
            for(var k in this) {
                document.write(k+' : '+this[k]+'<br>');
            }
        }
        person.showAll();
    </script>
</body>
</html>
```



create

```
11
Jiwoo
CEC
4.0
```

Iterate

```
age
11
name
Jiwoo
school
CEC
GPA
4.0
age : 11
name : Jiwoo
school : CEC
GPA : 4.0
```

property and method

```
age : 11
name : Jiwoo
school : CEC
GPA : 4.0
showAll : function() { for(var k in this) { document.write(k+' : '+this[k]+'
'); } }
```

Reflection:
A function can be set as a property value in an object.

A function belonging to an object is called a 'method,' while a variable belonging to an object is called a 'property.'

When calling a method in an object, the method itself is also included because the function also belongs to the object as a property.
This can be solved by using conditional statements in the method.

By using the method, it is possible to manage all the properties at once by creating another property in the object.