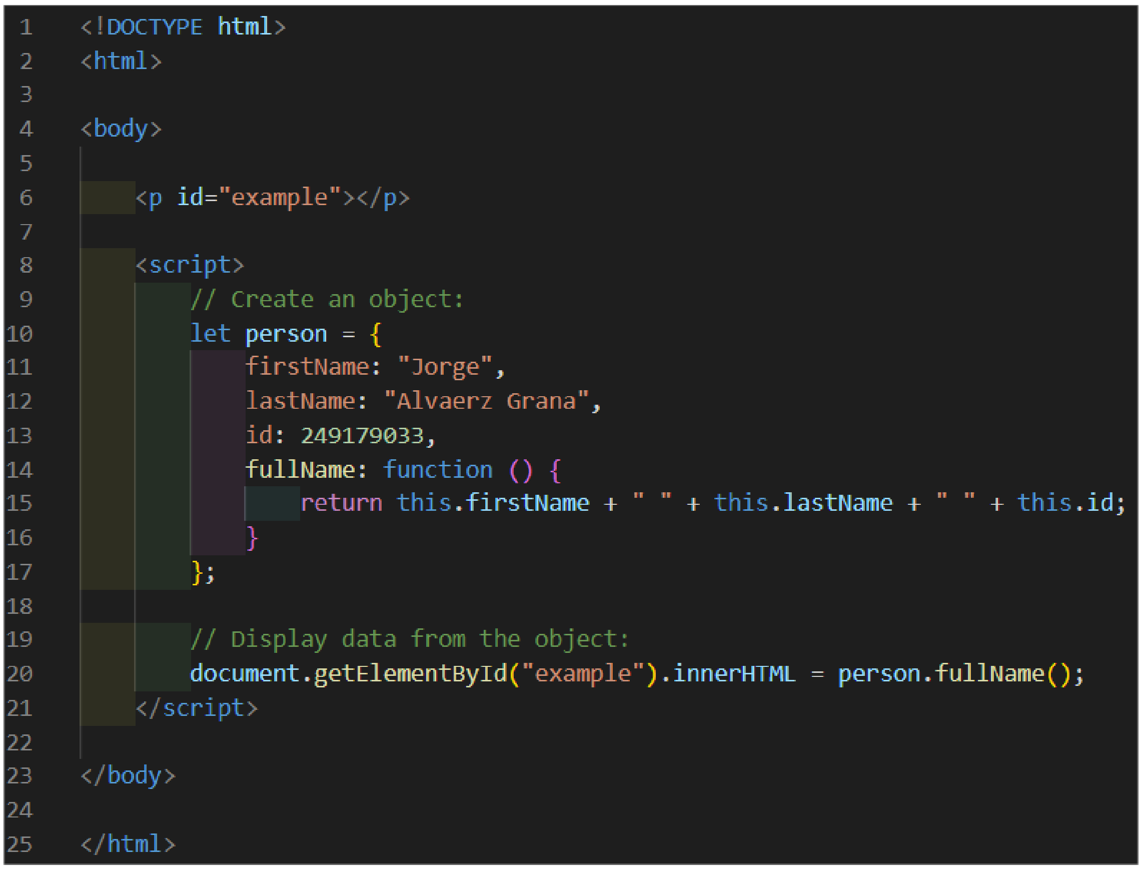
Student: Jorge Alvarez Grana (#249179033)
Class: WDD330
Term: Fall 2021
Subject: W03 Readings

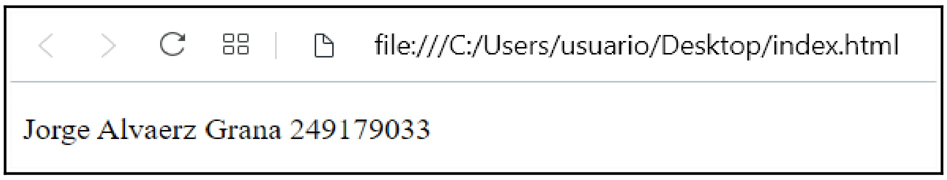
Object Methods: this

I've found interesting the concepts of *function*, *object* and *method*. Although it is just names, when hearing or reding about JS at times I get a bit lost in the nomeclature. So in this case a *function* is a procedure (executes a task). And *object* is a way of storing entities (related data like a user). And a *method* is a function inside and object. Objects use {}, functions use (), so it would look like this:



```
<!DOCTYPE html>
<html>

<body>

    <p id="example"></p>

    <script>
        // Create an object:
        let person = {
            firstName: "Jorge",
            lastName: "Alvaerz Grana",
            id: 249179033,
            fullName: function () {
                return this.firstName + " " + this.lastName + " " + this.id;
            }
        };

        // Display data from the object:
        document.getElementById("example").innerHTML = person.fullName();
    </script>

</body>

</html>
```

In the example I made a person *object*, given some data (firstname…) and made a *function* inside the object (a *method*). In order to access the data of the object the function uses *this,* displaying the following content when using *return*:

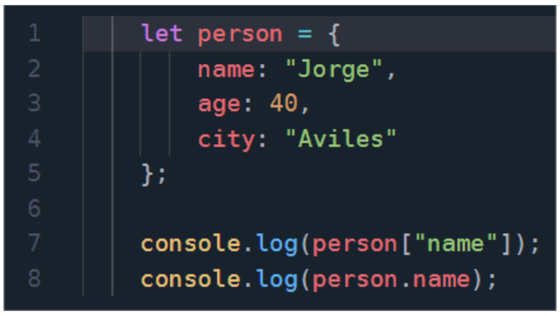


```
file:///C:/Users/usuario/Desktop/index.html

Jorge Alvaerz Grana 249179033
```

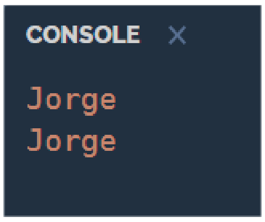
On the other hand, I've read that *this* is rather complicated concepts that even professionals still struggle to graps the full meaning of it. That made me realize that still there is much to learn because when reading the articles there were some gaps still. But the core concept as I understood it is as explained above.

Ch5: Objects

Interesting to me is the idea of linking *arrays* to lists and *objects* are used to store collections of related data. Unlike arrays, *objects* have no built-in order. So, an example of object is:

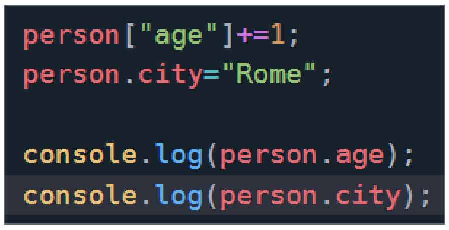


```
let person = {
    name: "Jorge",
    age: 40,
    city: "Aviles"
};

console.log(person["name"]);
console.log(person.name);
```

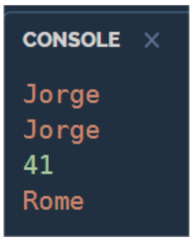


```
Jorge
Jorge
```

And here also we can see the two ways of accessing data in an *object*. I've found the second alternative easier to remember because of the dot notation, but I read that it is worth remembering the first two also because it might be handy in some instances. For example, when the property starts with a number or spaces in the names or when having to lookup using a variable. When having to update is also quite easy:

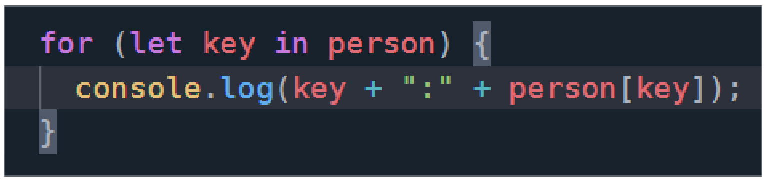


```
person["age"]+=1;
person.city="Rome";

console.log(person.age);
console.log(person.city);
```

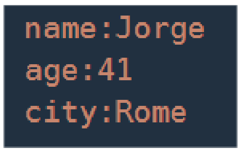


```
Jorge
Jorge
41
Rome
```

But to me the most interesting part was being able to have access to all the items in the object using a for-loop:



```
for (let key in person) {
  console.log(key + ":" + person[key]);
}
```



```
name:Jorge
age:41
city:Rome
```

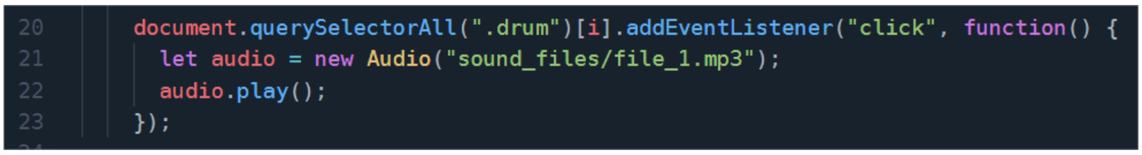
The rest of the materials (calling methods, math functions, JSON) were more familiar to me.

Ch6: Document Object Model (DOM)

I like the DOM because it is were finally JS meets HTML & CSS and you can start seeing the great advantages of using JS. I didn't know that the DOM could be accessed using other languages, I guess that I have being just introduced to all this concepts using JS and I just made a one way connection between them all. However, I didn't know about the tree model that is created in behind or that the HTML was turned into a bunch of objects that all combined make the object model part. The rest of the chapter was more familiar, especially the part of document.getElementbyId that was how I got to know about the DOM in the first place. (For example, see above in the this part where I used a div, an id and document.getElementbyID clause).

Ch7: Events

Along with the text I came across a very interesting website https://patatap.com that basically triggers an event anytime you press any key between A & Z. I was also reviewing their code, and although it has a degree of complexity above my current knowledge, I liked how it is full of the clause eventListener (See link). On how to incorporate sound into a keyboard stroke I was reading on MDN (see link) and I think I have more clear the idea of how to make it work if having mp3 files as well. Something in this line:



```
document.querySelectorAll(".drum")[i].addEventListener("click", function() {
  let audio = new Audio("sound_files/file_1.mp3");
  audio.play();
});
```

On the rest I liked the idea of triggering events depending on how the user reacts. Now I think I know how those noisy pop-ups that change their angle on the website to prevent you from clicking them away are done. Until now it was a mistery for me but now I think I start to understand.