Session 17

Mohamed Emary

May 8, 2024

1 Objects in JS

Objects are a way to store data in key-value pairs. Objects are non-primitive data types in JavaScript. Inside the object you can store any type of data, including other objects.

example:

```
var student = {
    name: "Mohamed",
    age: 30,
    city: "Cairo"
}
```

And to access values inside the object, we can use the form `objectName.keyName`:

```
console.log(student.name); // Mohamed
```

You can add values to the object after it's created:

```
student.gpa = 3.5;
```

Then you can access the new value:

```
console.log(student.gpa); // 3.5
```

To access a value of an object inside another object, you can use the form `objectName.keyName.keyName`:

```
student.subjects = {
    math: 90,
    science: 85,
    history: 70,
    english: 80
}
console.log(student.subjects.math); // 90
```

You can even store functions inside objects:

```
student.sayHello = function() {
    console.log("Hello");
```

```
}
student.sayHello(); // Hello
console.log(student.sayHello); // Function Definition
console.log(student.sayHello()); // undefined
```

The first `console.log` will print the function itself, while the second one will print the return value of the function, which is `undefined` in this case because the function doesn't return anything but it will also print `Hello` because the function prints `Hello` to the console.

If our function have a return value, it will be printed:

```
student.getAge = function() {
    return student.age;
}
console.log(student.getAge()); // 30
```

Function inside an object is called a method.

We can use backticks to print variables inside strings:

```
console.log(`My name is ${student.name}`);
```

Backticks also allow us to write multi-line strings:

```
console.log(`My name is ${student.name}
          I'm ${student.age} years old
          I live in ${student.city}`);
```

Any word that have a dot `.` after it, is an object. For example, `console.log` the `console` here is an object that have a method called `log`. Also `document` is an object that has many methods such as `getElementById`, `querySelector`, `querySelectorAll`, etc. Those are JS built-in objects.

`console.dir` is a method that prints the object itself:

One of the important built-in objects is the `Math` object. It has many methods such as:

- `Math.min()`
- `Math.max()`
- `Math.abs()`
- `Math.pow()`
- `Math.random()`
- `Math.floor()`
- `Math.ceil()`
- `Math.round()`

2 Array

An array is a list of items. It can store multiple values in a single variable. Arrays are also non-primitive data types.

Example:

```
var fruits = ["apple", "banana", "orange", "grape"];
```

To access an element in an array, we use the index of the element. The index is a number that represents the position of the element in the array. The index starts from 0.

```
console.log(fruits[0]); // apple
console.log(fruits[3]); // grape
```

To print all elements in the array, we can use a loop:

```
for (var i = 0; i < fruits.length; i++) {
    console.log(fruits[i]);
}
```

`fruits.length` will return the number of elements in the array.

You can store any type of data in an array, including other arrays:

```
var mixedArray = [
  "apple",
  5,
  true,
  ["banana", "orange"],
  { name: "Mohamed", age: 30 },
  function () {
    console.log("Hello");
  },
];
console.log(typeof mixedArray); // object
```

Later on when we learn about APIs, we will see that the data we get from APIs is usually in the form of an array of objects and it's called JSON.

3 Functional Programming in JS

In JS we can apply the concepts of functional programming such as:

- Assigning functions to variables
- Making a function as a property of an object (method)
- Function can be returned from another function
- Function can be passed as an argument to another function