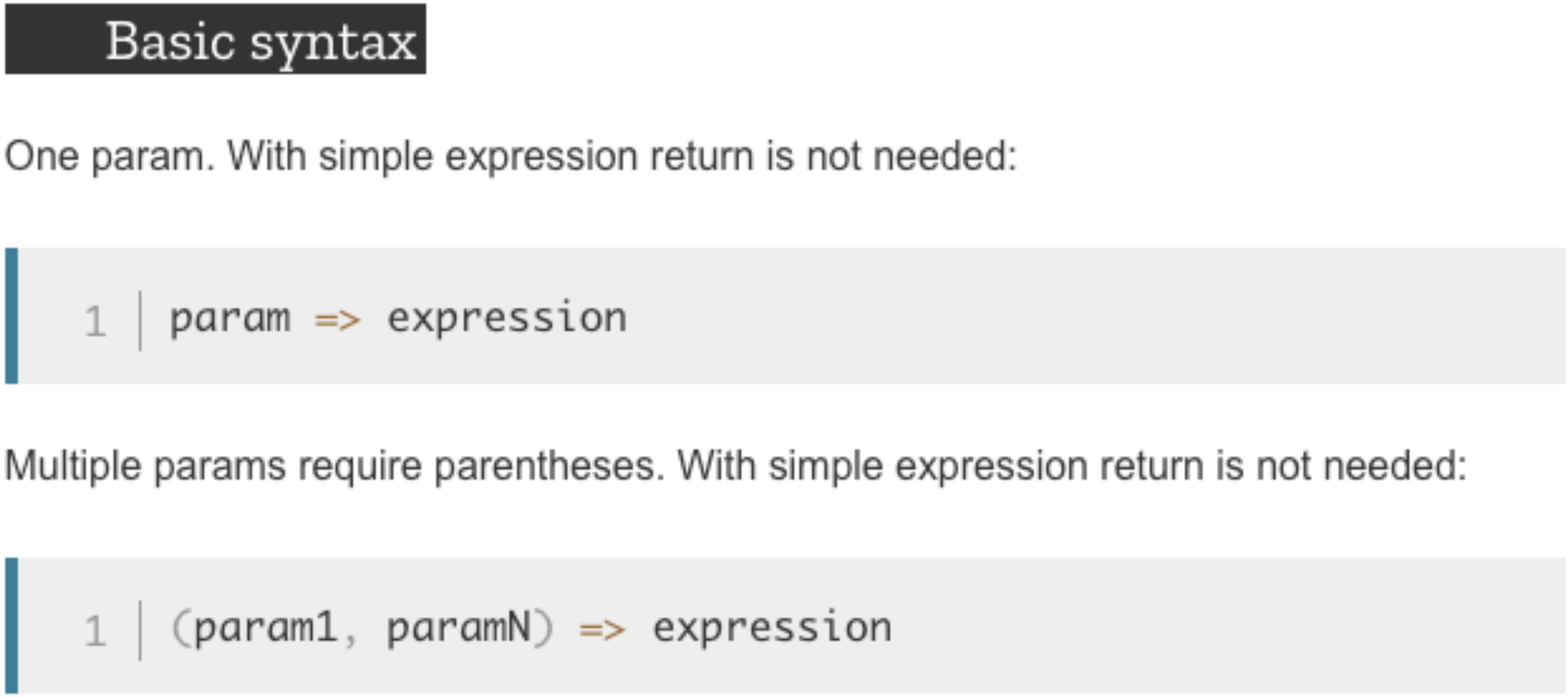
BOOTCAMP ’20

SESIONI #16

What will be covered?

- Arrow functions
- Callbacks
- Closures
- High-order functions
- Objects
- Document Object
- Window Object
- Math, Number, and Date Objects

Arrow functions



Basic syntax

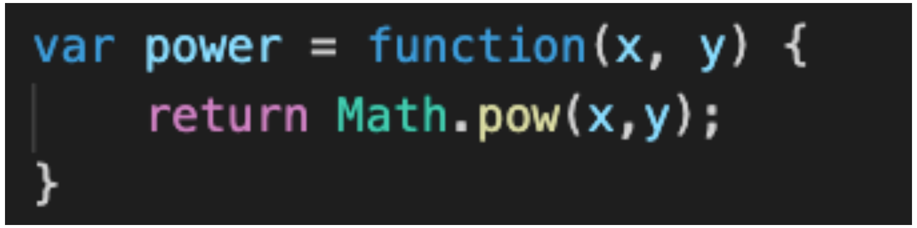
One param. With simple expression return is not needed:

```
param => expression
```

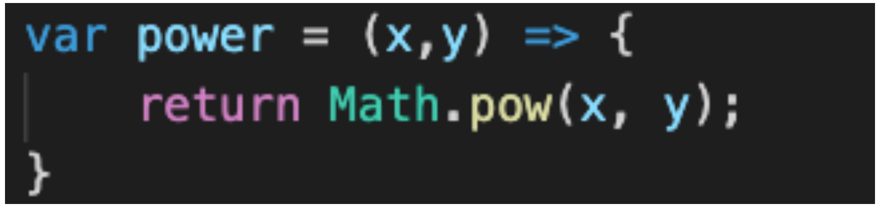
Multiple params require parentheses. With simple expression return is not needed:

```
(param1, paramN) => expression
```

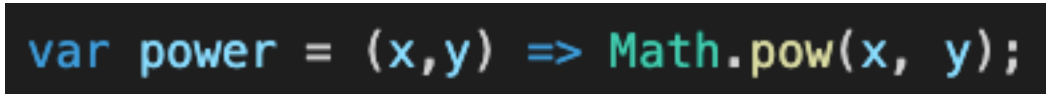
Arrow functions



```
var power = function(x, y) {
    return Math.pow(x,y);
}
```

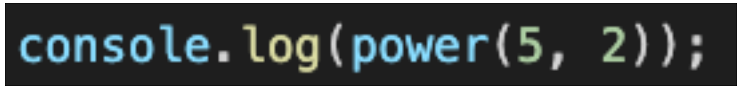


```
var power = (x,y) => {
    return Math.pow(x, y);
}
```



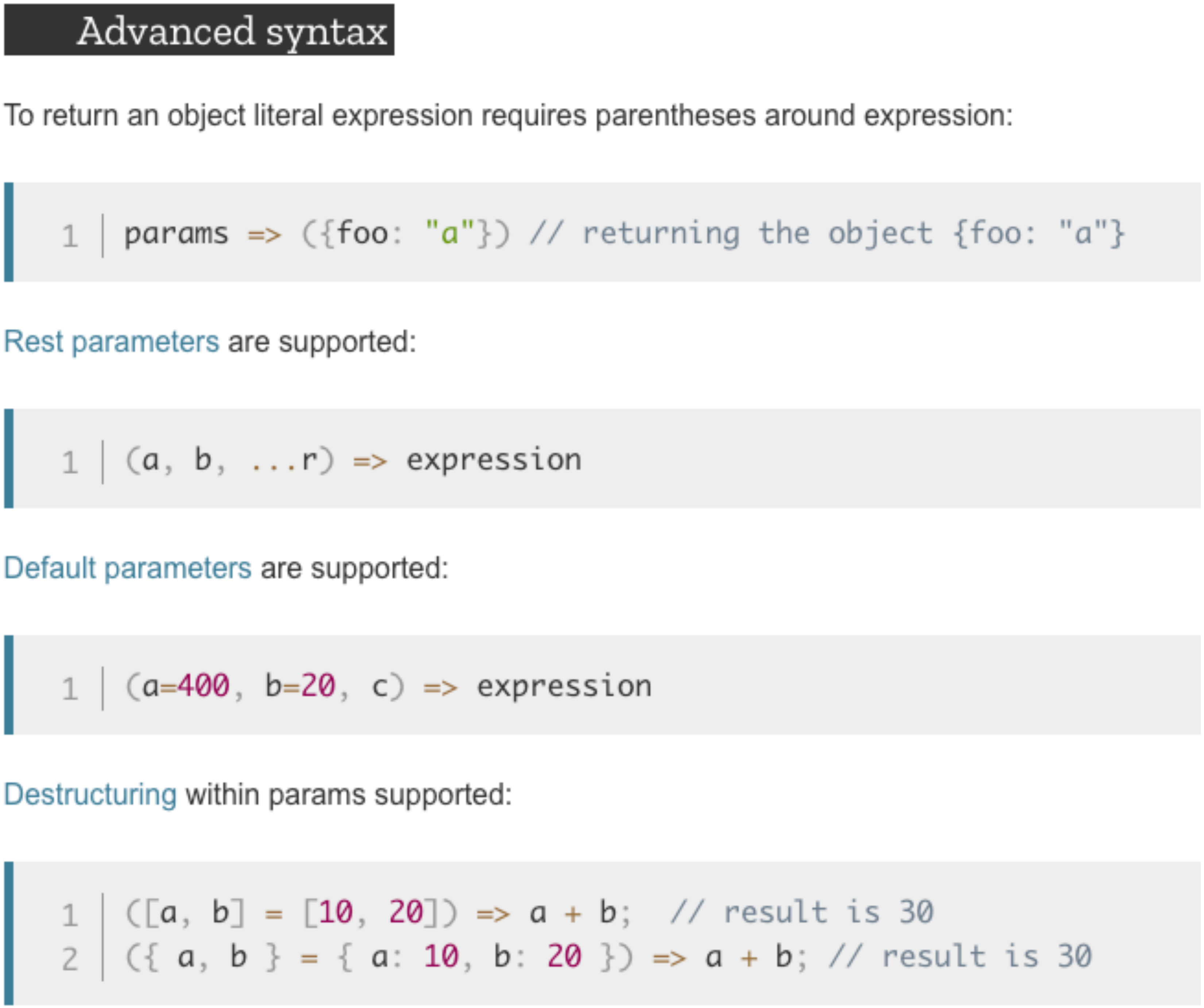
```
var power = (x,y) => Math.pow(x, y);
```

Thirrja e funksionit:



```
console.log(power(5, 2));
```

Arrow functions



Advanced syntax

To return an object literal expression requires parentheses around expression:

```
params => ({foo: "a"}) // returning the object {foo: "a"}
```

Rest parameters are supported:

```
(a, b, ...r) => expression
```

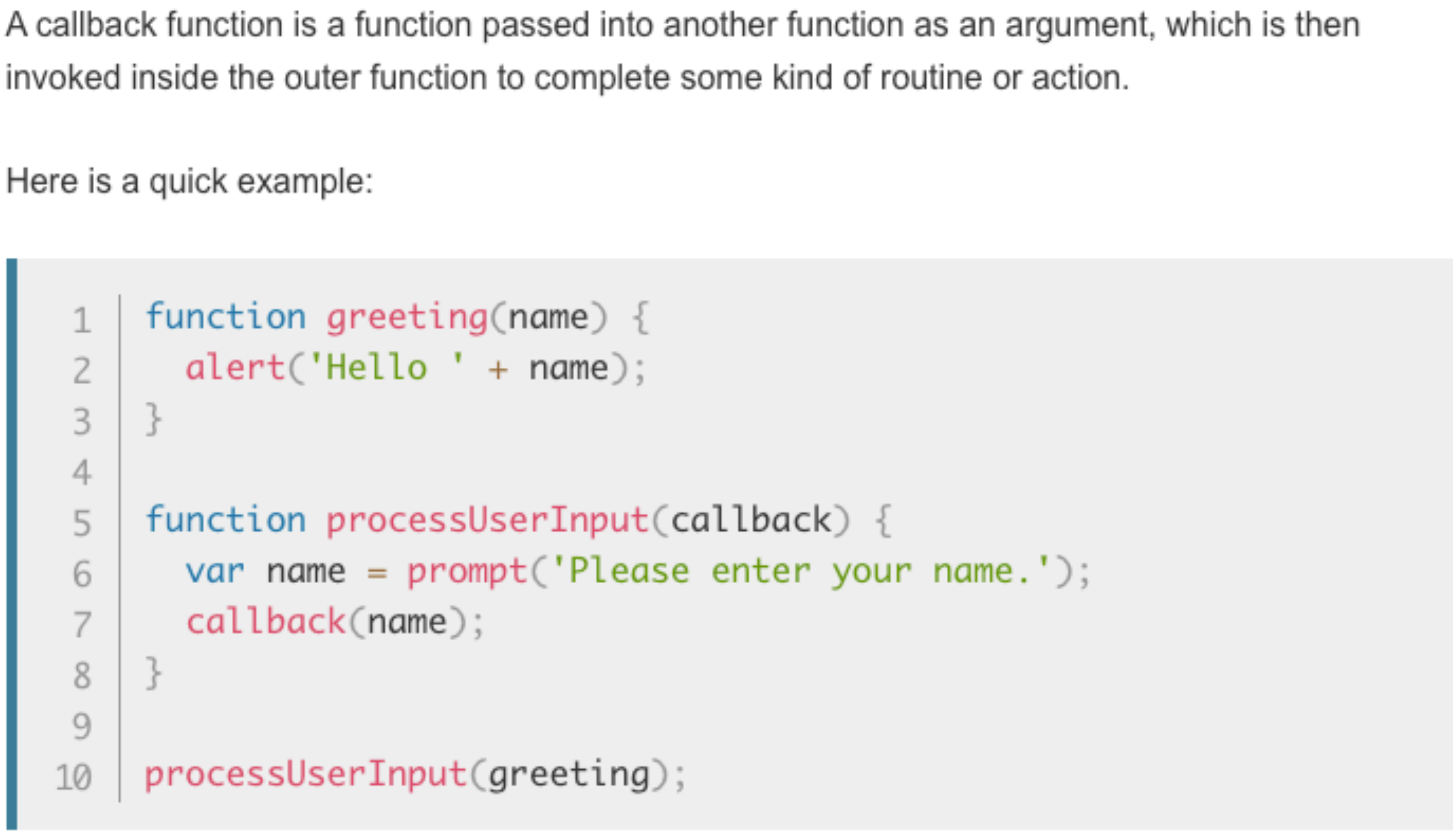
Default parameters are supported:

```
(a=400, b=20, c) => expression
```

Destructuring within params supported:

```
([a, b] = [10, 20]) => a + b;  // result is 30
({ a, b } = { a: 10, b: 20 }) => a + b; // result is 30
```

Callbacks



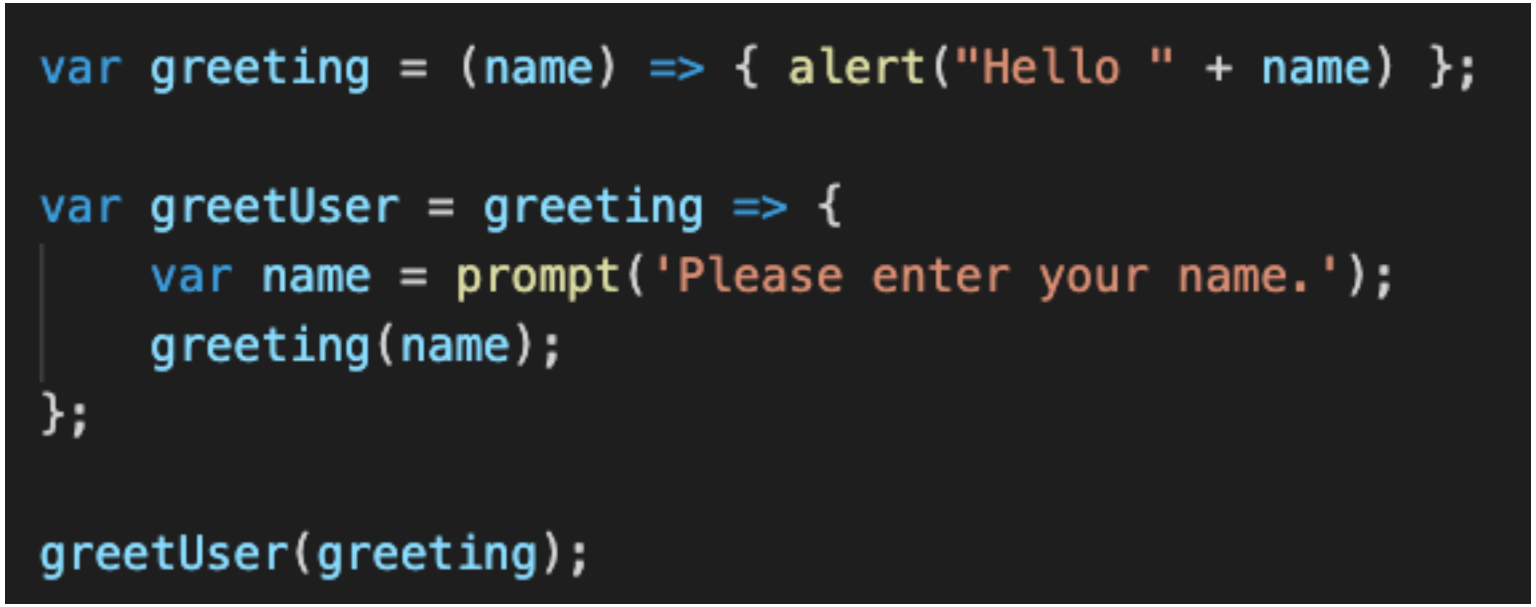
A callback function is a function passed into another function as an argument, which is then invoked inside the outer function to complete some kind of routine or action.

Here is a quick example:

```
function greeting(name) {
  alert('Hello ' + name);
}

function processUserInput(callback) {
  var name = prompt('Please enter your name.');
  callback(name);
}

processUserInput(greeting);
```

Callbacks

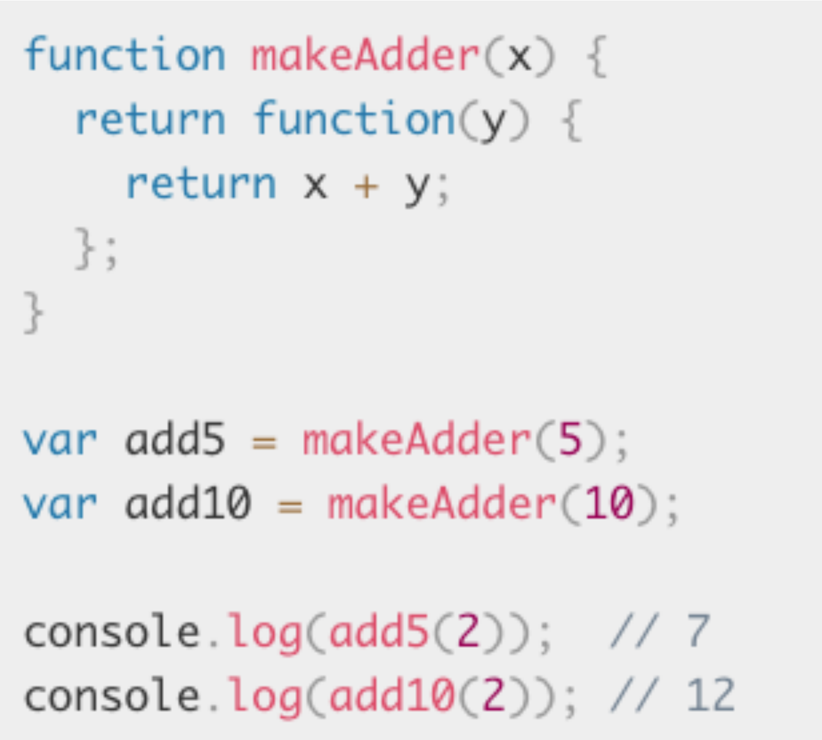
Me arrow functions:



```
var greeting = (name) => { alert("Hello " + name) };

var greetUser = greeting => {
    var name = prompt('Please enter your name.');
    greeting(name);
};

greetUser(greeting);
```

Closures

A *closure* is the combination of a function and the lexical environment within which that function was declared.



```
function makeAdder(x) {
  return function(y) {
    return x + y;
  };
}

var add5 = makeAdder(5);
var add10 = makeAdder(10);

console.log(add5(2));  // 7
console.log(add10(2)); // 12
```

Në këtë shembull add5 dhe add10 janë closures
Secila prej tyre ruan mjedisin leksikor të ndryshëm

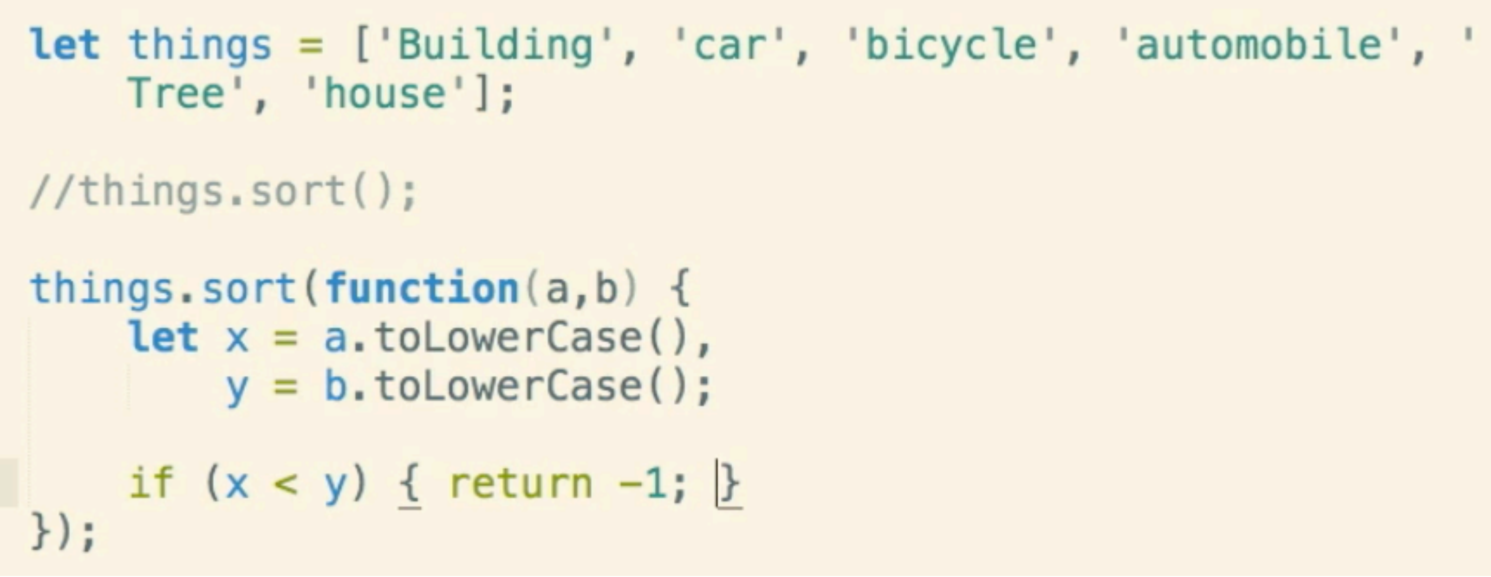
add5 – x e ka me vlerë 5
add10 – x e ka me vlerë 10

Le të shohim një shembull më atraktiv:

https://jsfiddle.net/vnkuZ/7726/

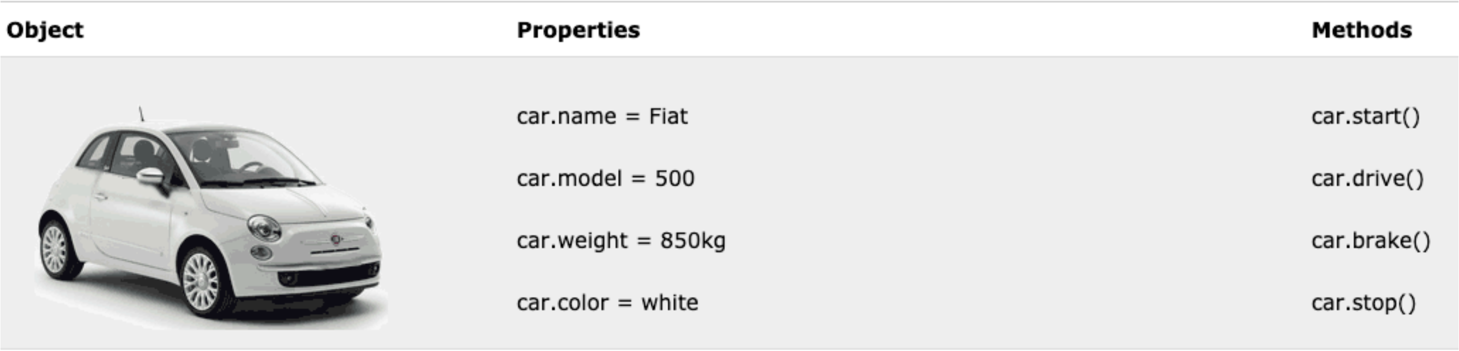
High-order functions

- Funksioni i cili pranon dhe/ose kthen ndonjë funksion quhet funksion i rendit të lartë

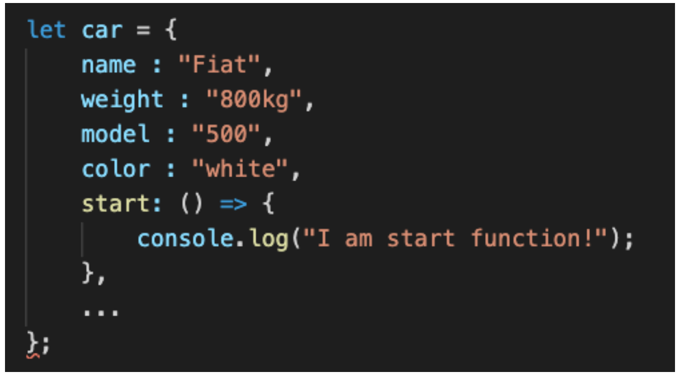


```
let things = ['Building', 'car', 'bicycle', 'automobile', '
    Tree', 'house'];

//things.sort();

things.sort(function(a,b) {
    let x = a.toLowerCase(),
        y = b.toLowerCase();

    if (x < y) { return -1; }
});
```

Objects

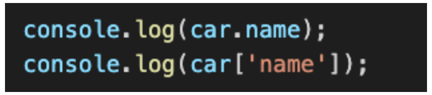


Object	Properties	Methods
	car.name = Fiat	car.start()
	car.model = 500	car.drive()
	car.weight = 850kg	car.brake()
	car.color = white	car.stop()

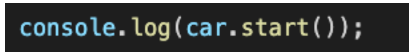


```
let car = {
    name : "Fiat",
    weight : "800kg",
    model : "500",
    color : "white",
    start: () => {
        console.log("I am start function!");
    },
    ...
};
```

Thirrja e atributeve dhe funksioneve të objektit:



```
console.log(car.name);
console.log(car['name']);
```



```
console.log(car.start());
```

Document Object

The HTML DOM document object is the owner of all other objects in your web page.

https://www.w3schools.com/js/js_htmldom_document.asp

https://developer.mozilla.org/en-US/docs/Web/API/Document

Window Object

The Browser Object Model (BOM) allows JavaScript to "talk to" the browser.

https://www.w3schools.com/js/js_window.asp

https://developer.mozilla.org/en-US/docs/Web/API/Window

Math, Number, and Date Objects

Math:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math

Number:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Number

Date:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date

QUESTIONS