Introcution to JavaScript

Nils Twelker

March 2023

What learned we last Week?

- Comparisons `1 > 2` , `'a' !== 'b'`
- Branching `if (condition) { ... } else { ... }`
- Loops `while (condition) { ... }` ,
  `for (let i = 0; i < 10; i++) { ... }`
- Logical Operators `&&` , `||` , `!`
- Prompt `let answer = prompt('What is your name?')`

Tipp: Dont forget to commit and push your changes.

Goals of this week

- Functions
  - Arguments / Default Arguments
  - Return Values
  - Scope
  - Hoisting
- Function Expressions
- Arrow Functions
- Basic Arrays

Functions

```
function functionName(parameter1, parameter2) {
    // code to be executed
}

functionName(argument1, argument2);
```

```
function add(a, b) {
    return a + b;
}

let result = add(1, 2); // result = 3
```

Function Arguments

```
let message = "Hello World";

function formatMessage(msg) {
    msg = "!" + msg + "!";
    return msg;
}

console.log(formatMessage(message)); // !Hello World!
console.log(message); // Hello World
```

Return Values

```
function isAdult(age) {
    if (age >= 18) {
        return true;
    } else {
        return false;
    }
}

isAdult(17); // false
```

```
function add(a, b) {
   if(typeof a !== 'number') {
       return;
   }
   if(typeof b !== 'number') {
       return;
   }
    return a + b;
}
```

Default Arguments

```
function add(a, b = 0) {
    return a + b;
}

add(1); // 1
add(1, 2); // 3
```

Tipp: Default arguments are only used if the argument is `undefined` .

Function Scope

```
let message = "Hello World";

function formatMessage() {
    console.log(message); // Hello World

    let message = "Hello Universe";

    console.log(message); // Hello Universe
}
```

Function Expressions

```
let add = function(a, b) {
    return a + b;
}

add(1, 2); // 3
```

```
function callTwice(fn) {
    fn(); fn();
}

callTwice(function() {
    console.log("Hello World");
});

// Hello World
// Hello World
```

Arrow Functions

```
let add = (a, b) => a + b

let sub = (a, b) => {
    return a - b;
}

add(1, 2); // 3
sub(1, 2); // -1
```

Function Hoisting

```
console.log(appendA("Hello ")); // Hello A
console.log(appendB("Hello ")); // ReferenceError: appendB is not
defined

function appendA(a ) {
    return a + "A";
}

const appendB = (a) => a + "B";
```

Basic Arrays

```
let numbers = [1, 2, 3, 4, 5];

console.log(numbers[0]); // 1
console.log(numbers[1]); // 2
console.log(numbers[2]); // 3
console.log(numbers[3]); // 4
console.log(numbers[4]); // 5
console.log(numbers[5]); // undefined
console.log(numbers.length); // 5
```

Strings are Arrays

```
let message = "Hello World";
// same as
// ["H", "e", "l", "l", "o", " ", "W", "o", "r", "l", "d"];
console.log(message[0]); // H
console.log(message[1]); // e
console.log(message[9]); // l
console.log(message[10]); // d
console.log(message[11]); // undefined

console.log(message.length); // 11
```

Modifying Arrays

```
let numbers = [1, 2, 3, 4, 5];

numbers[0] = 10;

console.log(numbers); // [10, 2, 3, 4, 5]
```

Tasks and Points

Goal is to get 100 Points.

- `basic-arrays` (25 Points)
- `basic-functions` (25 Points)
- `default-arguments` (25 Points)
- `expressions-arrows` (25 Points)
- `return-values` (25 Points)
- `shop` (50 Points)
- `tic-tac-toe` (50 Points)