academeez

Functions

Function behaviour in Javascript

Functions

- With Functions we can define multiple statements that can be called many times with a single command
- Functions can be configured how to work by passing arguments

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

// call a function
console.log(add(1,2)); // 3
```

Default value

academeez

- A function can get a default value, that will be assigned if undefined is passed

```
function add(a = 10, b = 20) {
    return a + b;
}

console.log(add(undefined, 10)); // 20
```

Spread operator

academeez

- With the spread operator, your function can get part (or all) of the arguments in an array

```
function multiplyBy(factor, ...numToMultiply) {
    return numToMultiply.map(function(num) {
        return num * factor;
    })
}

console.log(multiplyBy(2, 1,2,3,4,5)); // [2,4,6,8,10]
```

Arrow functions

- Another syntax for creating function is arrow functions
- The function is anonymous
- There is no **this**

```
// arrow function: syntax1
const add1 = (a, b) => {
    return a + b;
}

// arrow function: syntax2
const add2 = (a, b) => a + b;

// arrow function: syntax3
const multiply = a => a * a;

// call by save function to variable
add1(10, 20);
```

Summary

academeez

- Functions is one of the way we can prevent copy paste of code
- It allows us to reuse the code in a more generic way
- It is one of the building blocks of every Javascript application

academeez

Thank You

Next Lesson: Promise