



THE COMPLETE JAVASCRIPT COURSE

FROM ZERO TO EXPERT!



@JONASSCHMEDTMAN

SECTION

A CLOSER LOOK AT FUNCTIONS

LECTURE

FIRST-CLASS AND HIGHER-ORDER
FUNCTIONS

JS

FIRST-CLASS VS. HIGHER-ORDER FUNCTIONS

FIRST-CLASS FUNCTIONS

- 👉 JavaScript treats functions as **first-class citizens**
- 👉 This means that functions are **simply values**
- 👉 Functions are just another "**type**" of object

- 👉 Store functions in variables or properties:

```
const add = (a, b) => a + b;  
  
const counter = {  
  value: 23,  
  inc: function() { this.value++; }  
}
```

- 👉 Pass functions as arguments to OTHER functions:

```
const greet = () => console.log('Hey Jonas');  
btnClose.addEventListener('click', greet)
```

- 👉 Return functions FROM functions

- 👉 Call methods on functions:

```
counter.inc.bind(someOtherObject);
```

HIGHER-ORDER FUNCTIONS

- 👉 A function that **receives** another function as an argument, that **returns** a new function, or **both**
- 👉 This is only possible because of first-class functions

- 1 Function that receives another function

```
const greet = () => console.log('Hey Jonas');  
btnClose.addEventListener('click', greet)
```

Higher-order
function

Callback
function



- 2 Function that returns new function

```
function count() {  
  let counter = 0;  
  return function() {  
    counter++;  
  };  
}
```

Higher-order
function

Returned
function