Toegepaste Informatica

1TX

2022-2023

Front-end development

Part 2: Dynamic web applications

2. Javascript Functions

J. Van Hee, J. Pieck, G. Jongen, A. Vranken

Javascript: Functions & Error handling

- Recap functions
- First-class functions
- Higher-order functions

Recap: Functions

- Classic

```
function calculateSum(a, b) {
  return a + b;
}
…
let s = calculateSum(2,4);
```

- Modern: arrow functions

```
const calculateSum = (a, b) => {
  return a + b;
}
…
let s = calculateSum(2,4);
```

Recap: Functions

- Shorthand version, if function contains 1 line:

```
const calculateSum = (a, b) => a + b;
…
let s = calculateSum(2,4);
```

- Try to use JS as a “functional” language and express your functionality with functions (even if they are only 1 line).

First-class functions

- In JS functions are first-class functions: they can be treated like **variables**

```
const calculateSum = (a, b) => a + b;
…
let other = calculateSum; //assign function to other variable

console.log(other(2,4))                 // “6”
console.log(calculateSum(2,1))          // “3”
```

Higher-order functions

- Functions that leverage other functions by either **receiving** or **returning** them
- Pass functions as **argument**

```
const sum = (a, b) => a + b;
const multiply = (a, b) => a * b;

const calculate = (a, b, operation) => “Result: “ + operation(a,b);

let res = calculate(2, 3, sum); // “Result: 5”
res = calculate(2, 3, multiply); // “Result: 6”
```

Higher-order functions

- Looping over arrays with the **forEach** higher-order function

```
const persons = [“John”, “Annie”];

const greet = (person) => console.log(“Hello “ + person);

// execute greet function for every element in array
persons.forEach(greet);      // “Hello John”, “Hello Annie”

// Inline as an anonymous function
persons.forEach((person) => console.log(“Hello “ + person))
```

Higher-order functions

- Filtering arrays with the **filter** higher-order function

```
const persons = [“John”, “Annie”, “Martha”];

// Only retain persons whose length is more than 4
// characters
const filtered = persons.filter((person) => person.length > 4);

console.log(filtered) // “Annie”, “Martha”
```