Hamburg Coding School

Henning Muszynski

Session 1

JavaScript for Web

Hi, I'm Henning 👋

- Head of Frontend at Doist
- Conference Speaker
- Beer Nerd & Brewer

Hamburg Coding School

Your Experience & Expectations

Being able to read, understand and write modern JavaScript code to work on web applications.

Course Goal

Additional coaching (30 or 60 minutes) is available on request

Important Info

Course App

Travel Blog

Travel Blog Features

🗺️ A map with markers of all places you've been

📌 Markers are blog posts with rich information

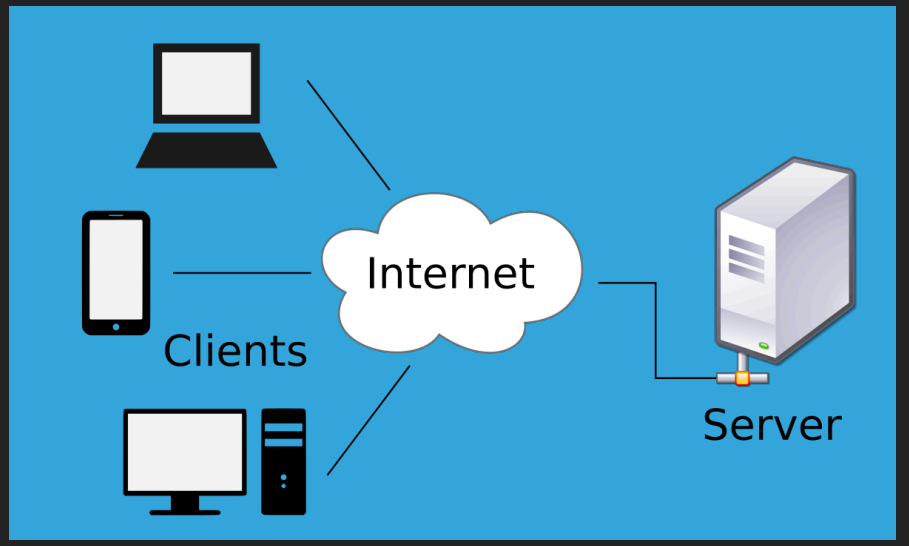
🔐 Only logged in users can see the blog

📝 Create new blog posts

🛑 View blog when offline

Frontend Architectures

- Server-Rendered Apps
- Single Page Apps
- Isomorpic / Universal Apps



- It always depends – for this course we decided to build a SPA

Modern JavaScript: Arrow Functions

```
function sumOfApples(bucket1, bucket2) {
    const sum = bucket1 + bucket2
    return sum
 }
```

```
const sumOfApples = (bucket1, bucket2) => {
    const sum = bucket1 + bucket2
    return sum
}
// or with implicit return:
const sumOfApples = (bucket1, bucket2) => bucket1 + bucket2
```

Modern JavaScript: Arrow Functions

Special case: Single parameter can omit brackets

```
const log = value => console.log('Value:', value)
```

Special case: No parameter have empty brackets

```
const sayHello = () => console.log('Hello')
```

Modern JavaScript: Template Literals

String concatenation is cumbersome:

```
const person = { name: 'Henning', age: 28, role: 'Engineer' }

console.log(person.name + ' is ' + person.age + ' years old.')

// logs: Henning is 28 years old
```

```
const person = { name: 'Henning', age: 28, role: 'Engineer' }

console.log(`${person.name} is ${person.age} years old.`)

// logs: Henning is 28 years old
```

Modern JavaScript: Array Destructuring

Select the values you're interested in

```
const values = ['Henning', 28, 'Engineer']
const [name, age] = values
```

Modern JavaScript: Object Destructuring

Select the properties you're interested in

```
const person = { name: 'Henning', age: 28, role: 'Engineer' }
const { name, age } = person
```

Supply default values for unknown properties

```
const person = { age: 28, role: 'Engineer' }
const { name = 'Henning', age = 20 } = person
```

Modern JavaScript: Object Spreading

Problem: Copying objects

```
const person = { name: 'Henning', age: 28 }
const person2 = person
person2.name = 'Nina'

console.log(`${person.name} is ${person.age} years old.`)
// prints: Nina is 28 years old
```

Modern JavaScript: Object Spreading

```
const person = { name: 'Henning', age: 28 }
const person2 = { ...person }
person2.name = 'Nina'

console.log(`${person.name} is ${person.age} years old.`)
// prints: Henning is 28 years old

console.log(`${person2.name} is ${person2.age} years old.`)
// prints: Nina is 28 years old
```

Modern JavaScript: Array Spreading

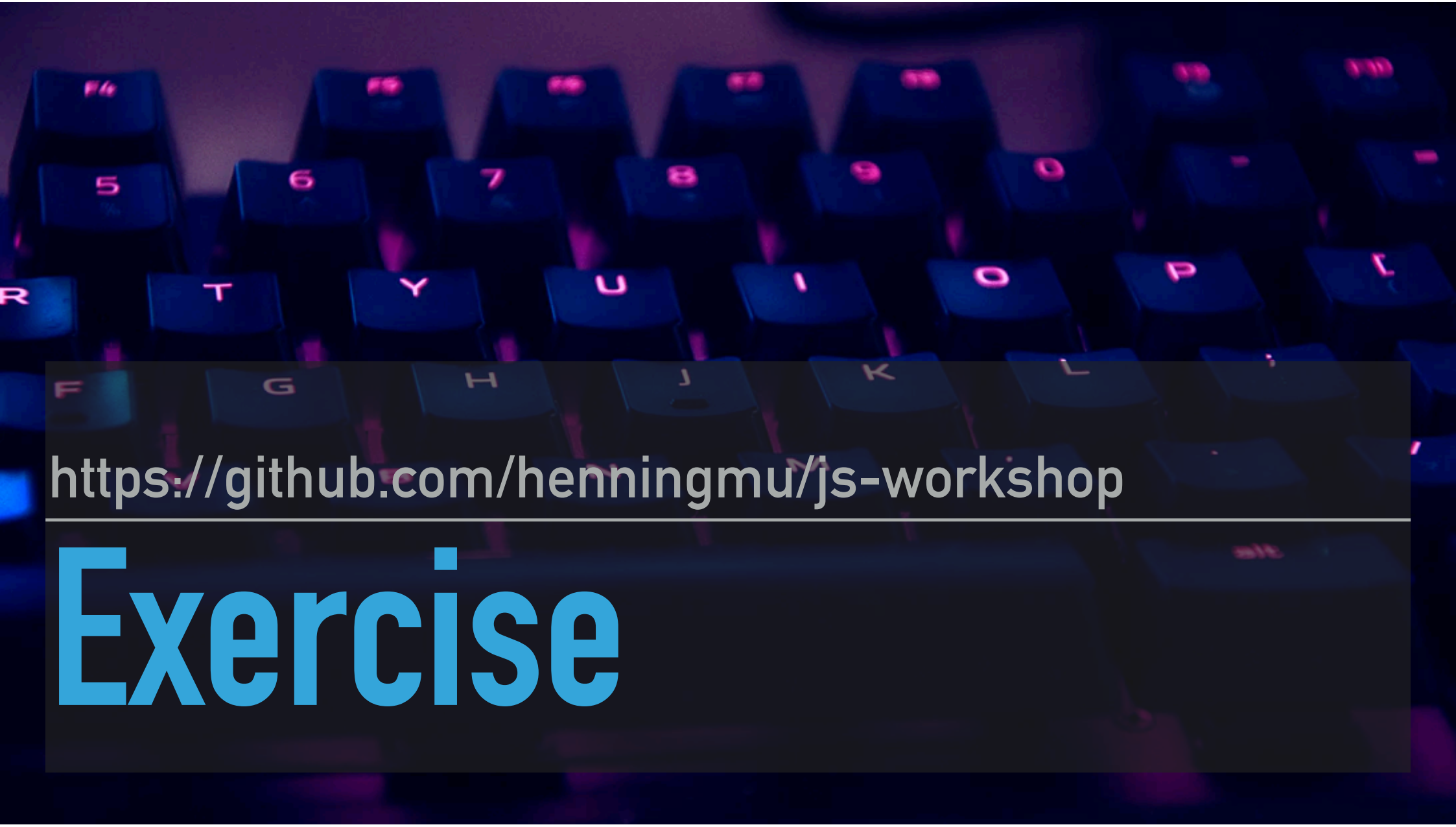
```
const values1 = [1, 2, 3]
const values2 = [4, 5, 6]
const allValues = [...values1, ...values2] // [1, 2, 3, 4, 5, 6]

// used less often as there's also:
const allValues = values1.concat(values2)
```

Modern JavaScript: Array.map()

Execute a function for each element in the array and return a new array

```
const numbers = [1, 2, 3, 4]
const doubles = numbers.map(number => number * 2)
// [2, 4, 6, 8]
```



https://github.com/henningmu/js-workshop

Exercise

Modern JavaScript: Array.filter()

Evaluate a condition for each element in an array and return array with elements that evaluate to true

```
const numbers = [1, 2, 3, 4, 5, 6, 7, 8]
const evens = numbers.filter(number => number % 2 === 0)
// [2, 4, 6, 8]
```

Modern JavaScript: Array.reduce()

Execute a function for each element in the array, return a single reduced result

```
const numbers = [1, 2, 3, 4]
const initialValue = 0
const sum = numbers.reduce((result, number) => {
    return result + number
}, initialValue)

// sum is 10
```

Recap Array Methods

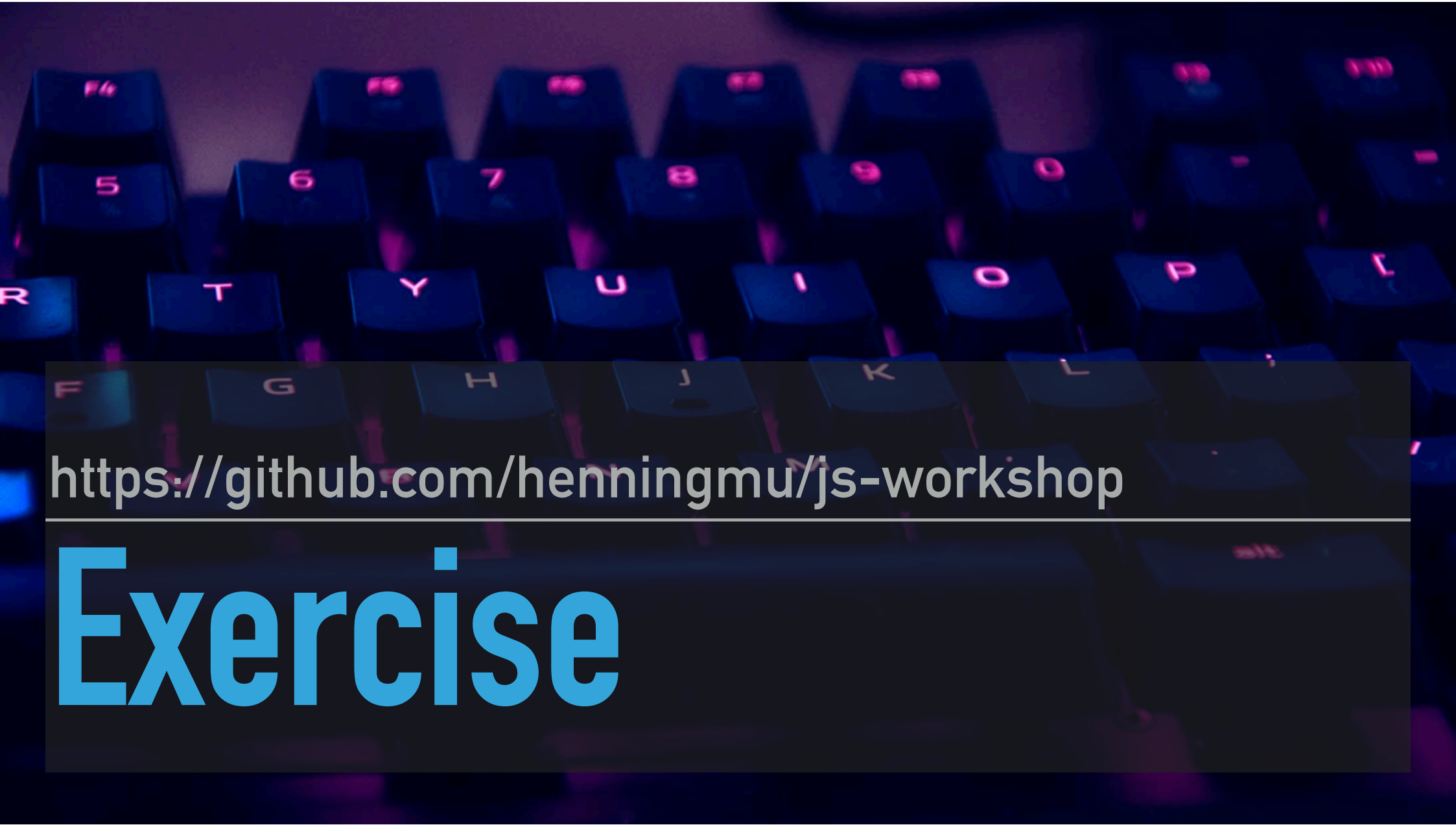
- **Array.map** Change each element, return array of same size
- **Array.filter** Filter only relevant values from array, return new array
- **Array.reduce** Create a single value from array

Modern JavaScript: Ternary Operator

Not really new, but experiences a revival

```
const number = 3
if (number > 5) {
    console.log('Larger than 5')
} else {
    console.log('Lesser or equal 5')
}
// Lesser or equal 5

number > 5
    ? console.log('Larger than 5')
    : console.log('Lesser or equal 5')
// prints: Lesser or equal 5
```



https://github.com/henningmu/js-workshop

Exercise

Quick Feedback