

**Vijfhart,**

dat klopt voor jou!





# Angular



# Introduction round

Role / background within KPN and before 

Hobbies / interests / family / pets    

What do you hope to learn? 



# Before we start...

1. Please keep your camera on 😊
2. Questions? Just ask! Feel free to interrupt. 🗣️
3. Two 10 minutes breaks. ☕

# Overview



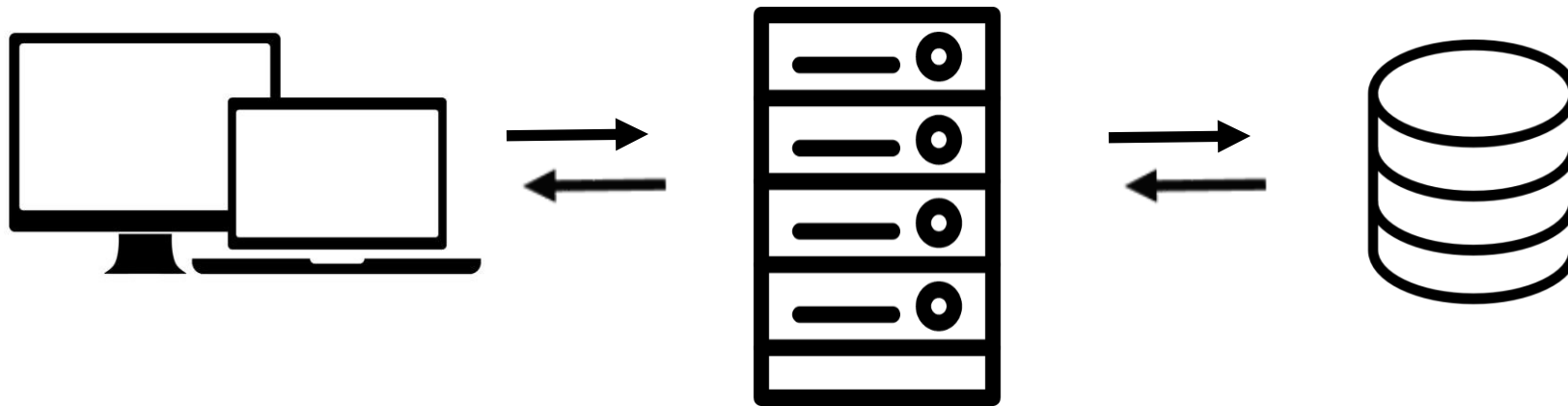
1. Overview web (development)
2. HTML, CSS, JavaScript and TypeScript
3. Angular Introduction + hello world demo
4. Angular Exercise
5. Angular demo + exercises

# Tooling



1. Preferably an IDE (Visual Studio Code, NotePad++, Atom, Sublime etc) + Node.js installed
2. Alternatively: online environment such as Stackblitz or CodePen
3. Must have: Browser (Chrome, Firefox or Edge)

# Overview web (development)



# HTML, CSS and JS

**HTML (Hypertext Markup Language):** The standard markup language used to structure content on the web.

**CSS (Cascading Style Sheets):** The style sheet language used to define the visual presentation of a web page.

**JS (JavaScript):** The programming language that allows for dynamic and interactive elements on a web page.





**HTML**





# What is HTML?

- Hyper Text Markup Language
- HTML determines what is on the web page (another word for website), for example text, buttons, forms and images.
- Using special words between < and >, the internet browser can display the content of the page.

# Why do we need HTML?

HTML is the content of web pages

This includes:

- Text
- Headers
- Forms
- Buttons
- Images
- Video
- And a lot more!



# Basic HTML document



```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <title>Basic HTML Document</title>
```

```
</head>
```

```
<body>
```

```
    <h1>Welcome to My Basic HTML Page</h1>
```

```
</body>
```

```
</html>
```





# HTML elements



HTML element is a component on a web page

```
<startTag>Content</endTag>
```

There are many different types of HTML elements, for example: p, form, div, h1

HTML elements can contain other HTML elements

More information and more elements:  
[https://www.w3schools.com/html/html\\_elements.asp](https://www.w3schools.com/html/html_elements.asp)



# HTML attributes

- HTML attributes provide additional information about an element and help define its properties or behavior.
- They are always specified in the start tag (or the opening tag) of an HTML element.
- They are usually presented in name/value pairs like name="value".

```
<a  
href="https://www.example.com">Visit  
Example</a>
```

```

```



# Attributes we'll need for CSS and JavaScript

- Class: HTML elements can have a class attribute. This can be used to group certain elements and give it a certain layout or behavior.

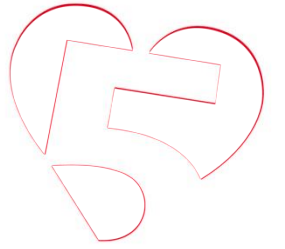
```
<p class="special">Some text</p>
```

- Id: HTML element can have an id. This must be a unique id for the page.

```
<p id="test">Some text</p>
```



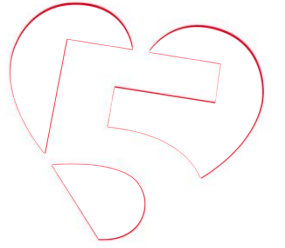
# HTML element



```
<tag attr="value">  
  <inner>  
    Some text  
  </inner>  
  <inner>  
    Some more text  
  </inner>  
</tag>
```



# HTML element



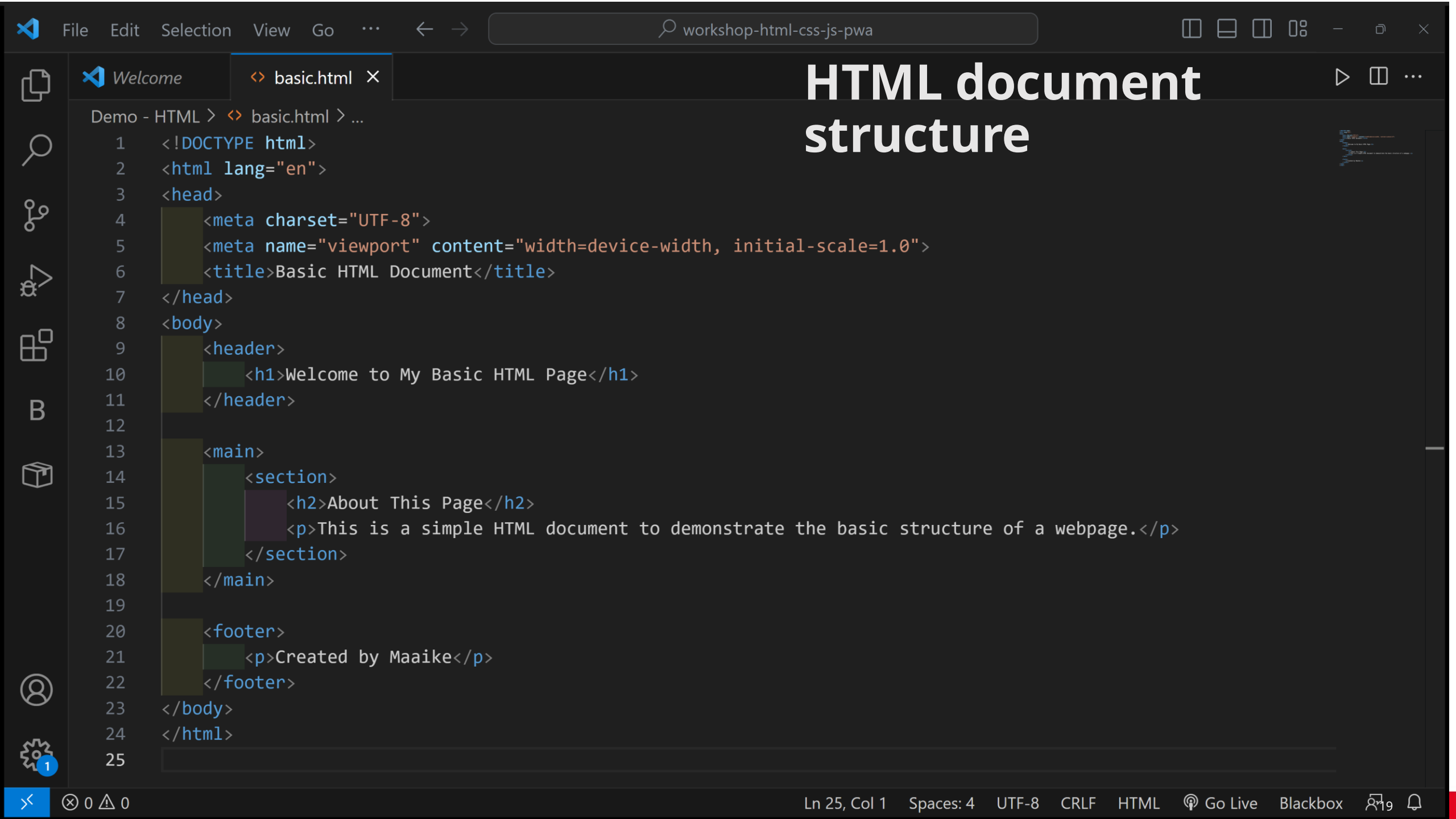
```
<tag attr="value">inner HTML</tag>
```



# Nesting HTML elements



- Nested elements are HTML elements inside other elements.
- Outer element is the "parent"; inner element is the "child".
- Enables complex structures, like lists within lists.
- Ensure proper opening and closing to avoid display errors.



# HTML document structure

Demo - HTML > basic.html > ...

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Basic HTML Document</title>
7  </head>
8  <body>
9      <header>
10         <h1>Welcome to My Basic HTML Page</h1>
11     </header>
12
13     <main>
14         <section>
15             <h2>About This Page</h2>
16             <p>This is a simple HTML document to demonstrate the basic structure of a webpage.</p>
17         </section>
18     </main>
19
20     <footer>
21         <p>Created by Maaike</p>
22     </footer>
23 </body>
24 </html>
25
```

CSS



# What is CSS?

Cascading Style Sheets

Used for creating the layout of the web page

With CSS we define rules for certain HTML elements

We can do a lot of things with CSS! Change the font, color, position, size, shadows, borders, shape... We can even create complete animations!

# Adding CSS to your page

Different ways to add CSS to your page:

- Style attribute on an element
- Style HTML element
- Link a separate page (preferred way)

To link a separate page, in the head tag add:

```
<link rel="stylesheet" href="name-  
css-file.css">
```







# CSS Syntax



Selector

**h1**

Declaration

**{ color:blue; font-size:12px; }**

Declaration

Property

Value

Property

Value



# CSS Selectors



We'll keep it simple here:

- By tag name:

```
p {  
  color: red;  
}
```

- By class:

```
.special {  
  color: green;  
}
```

- By id:

```
#element2 {  
  color: yellow;  
}
```



# CSS properties

Used for adjusting a certain part of the layout

More info:

[https://www.w3schools.com/css/css\\_colors.asp](https://www.w3schools.com/css/css_colors.asp)





# Specificity



When CSS declarations are conflicting, the one with the most points win:

- Id: 100
- Class: 10
- Tagname: 1

The most specific declaration determines the layout that shows. That's called specificity.



# CSS Demo



# JavaScript





# What is JavaScript?

Scripting language

That can be used server side (node.js) and client side

Client side: we'll use it to make our pages interactive





# We could do a 7 week course on JavaScript...



But here's the 10 minutes version to get you ready for Angular:

- Declare variables
- Writing a function
- Conditionals and loops



# Declare a variable



- We often need to have a placeholder for a value that will be set during program execution.
- `let nr = 5;`
- `let name = "Maaike";`
- `console.log(name, nr);`

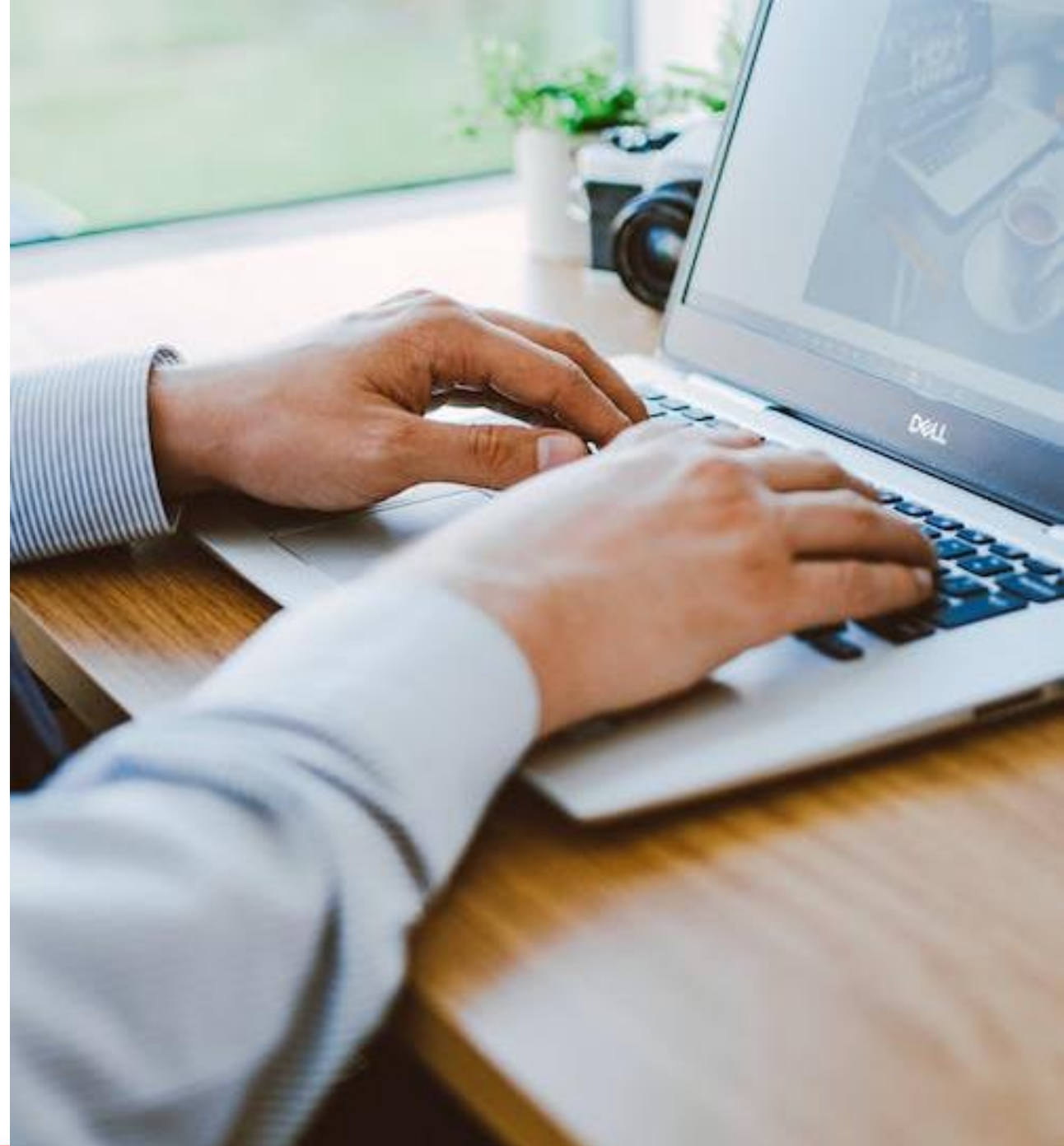
# Writing a function

- Functions encapsulate reusable code.
- Define using function keyword.

```
function greet() {  
    alert("Hello, World!");  
}
```

- You can call this function with:

```
greet()
```







# If statements



If statements in JavaScript are used to perform different actions based on different conditions.

This control structure allows the program to make decisions, and execute a certain section of code only if a particular condition evaluates to true.

```
if (score > 50) {  
    console.log('You passed!');  
}
```

# Loops



- Loops in JavaScript are used to execute a block of code repeatedly until a specified condition returns false.
- There are different kinds of loops, we'll only talk about the for loop here.

```
for (let i = 0; i < 5; i++) {  
  console.log(i);  
}
```



# JavaScript Demo



# TypeScript



# JavaScript and TypeScript



- TypeScript is a superset of JavaScript that adds static types.
- This means you can assign types to variables, functions, and properties.
- It transpiles to plain JavaScript.

# Types for variables

- Variables can be defined with specific types.
- This ensures they hold that specific type of value such as number, string, boolean, etc.
- This type-checking can prevent many common errors in JavaScript.

```
let isCompleted: boolean = false;
```







# Types for methods



Methods can be given a specific type for their return value and for their parameters.

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



# TypeScript Demo





# Angular

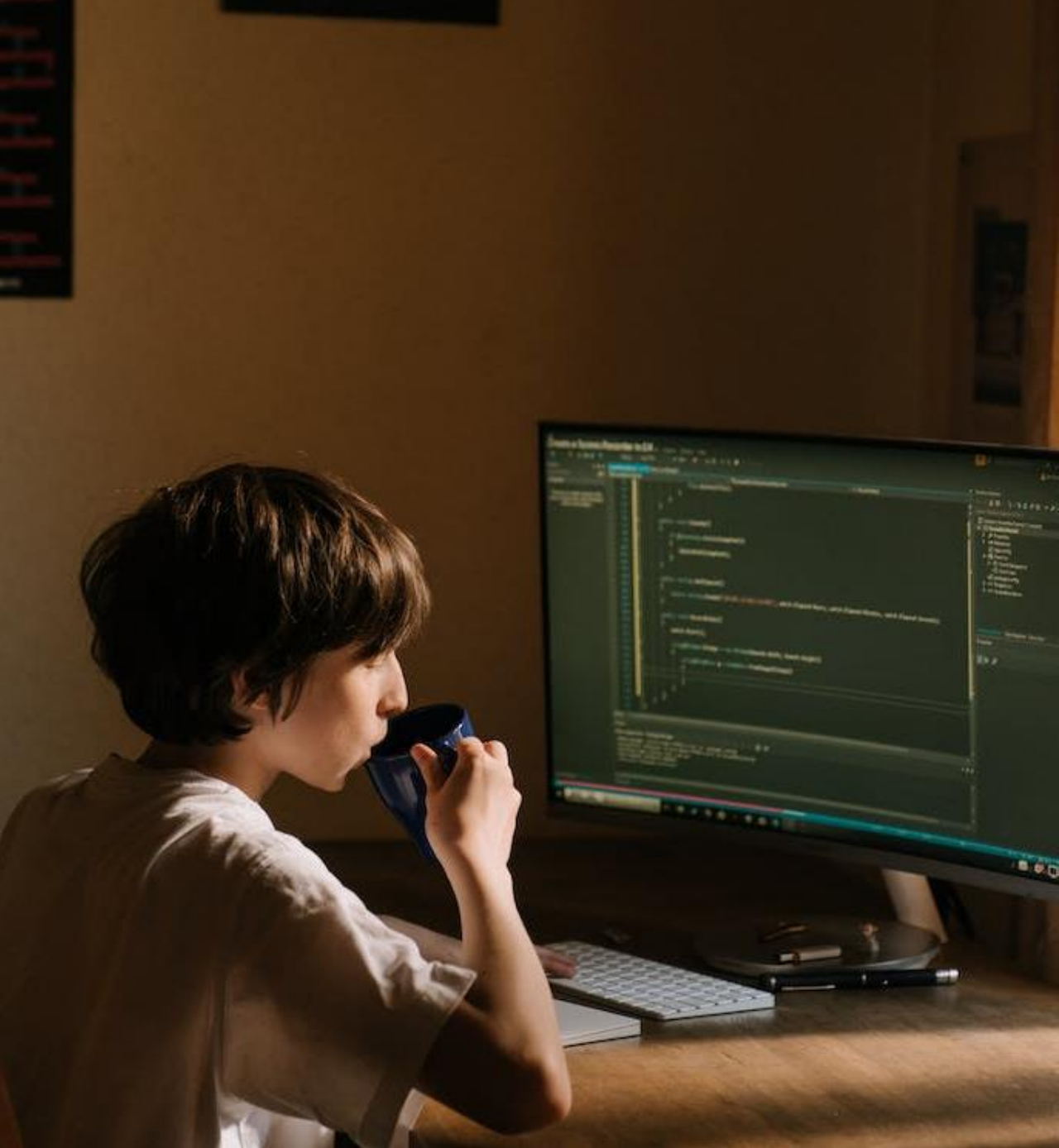


# What is Angular?



- Angular is an open-source web application framework maintained by Google
- Great for developing single-page applications
- Reusable code and abilities for Progressive Web Apps
- Works with Dependency Injection





# Angular's Building Blocks



- Modules: Organize the application into cohesive blocks of functionality.
- Components: Define views, which are sets of screen elements.
- Services: Manage data and logic independent of views.



# Angular CLI



Command Line tool for creating and managing Angular applications

(CLI = Command line interface)

You'll use it for:

- Quickstart: Sets up a new project
- Generators: Scaffold components, services, models and modules



# Angular Hello World Demo



Code along





# Exercise

# Data Binding and Directives

One-Way Data Binding: Display variables in your HTML

Two-Way Data Binding: Keeps model and view in sync. (E.g. forms)

Structural Directives: Change the DOM layout by adding and removing elements. For example: `ngIf` and `ngFor`

Note: There's a lot more to say about these... But that's not for now.





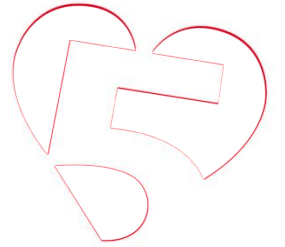
# Angular Data Binding Demo





# Exercise

# Angular Components



A component controls a patch of screen called a view.

Each component has its own user interface and logic encapsulated.

Component consists of:

- class to handle logic and data
- HTML template
- Stylesheet

```
import { Component } from  
'@angular/core';
```

```
@Component({  
  selector: 'app-hello-world',  
  template: `<h1>Hello, {{name}}!</h1>`,  
  styles: ['h1 { font-weight: normal; }']  
})  
export class HelloWorldComponent {  
  name: string = 'Angular';  
}
```





# Angular Component demo



Let's do these steps together!



# Exercise

# Wrap up



1. HTML
2. CSS
3. JavaScript & TypeScript
4. Angular



Contact Rokus for more information: [r.janssen@vijfhart.nl](mailto:r.janssen@vijfhart.nl)

T<sup>1</sup> H<sup>4</sup> A<sup>1</sup> N<sup>1</sup> K<sup>1</sup>  
Y<sup>4</sup> O<sup>1</sup> U<sup>2</sup>



# Tot ziens!



Volg ons op LinkedIn



Ontvang onze nieuwsbrief



Bekijk onze last minutes

**vijfhart.nl**



**vijfhart**  
IT-OPLEIDINGEN