

Intro to Angular

HTTP Basics. Angular Overview. TypeScript Syntax.



SoftUni Team

Technical Trainers



SoftUni



Software University

<https://softuni.bg>

Table of Contents

1. HTTP Basics
2. Routing Overview
3. Angular Overview
4. Intro to TypeScript
5. Angular Installation & CLI
6. Demo



sli.do

#js-web

A background network diagram consisting of a central dark blue circle containing the text 'http://'. Surrounding this central circle are several smaller, light gray circles connected by thin gray lines, forming a web-like structure. The circles vary in size, and the lines connect them in a non-uniform, organic pattern.

http://

HTTP Basics

HTTP Server - Client

- **H**yper **T**ext **T**ransfer **P**rotocol (HTTP)
 - Client-server protocol for **transferring** Web **resources**
- Important properties of HTTP
 - Request-response model
 - Text-based format
 - Relies on a unique resource URLs
 - Provides resource metadata (e.g. encoding)
 - Stateless (cookies can overcome this)

HTTP: Request-Response Protocol

- Client program
 - Running on end host
 - E.g. Web browser
 - Requests a resource
- Server program
 - Running at the server
 - E.g. Web server
 - Provides resources

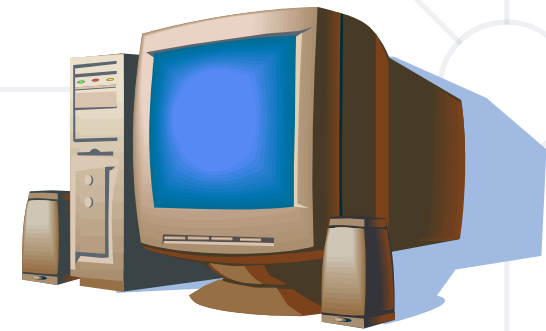


HTTP: Request-Response Protocol (2)



GET /index.html
HTTP/1.0

HTTP/1.0 200 OK
"Welcome to our
Web site!"



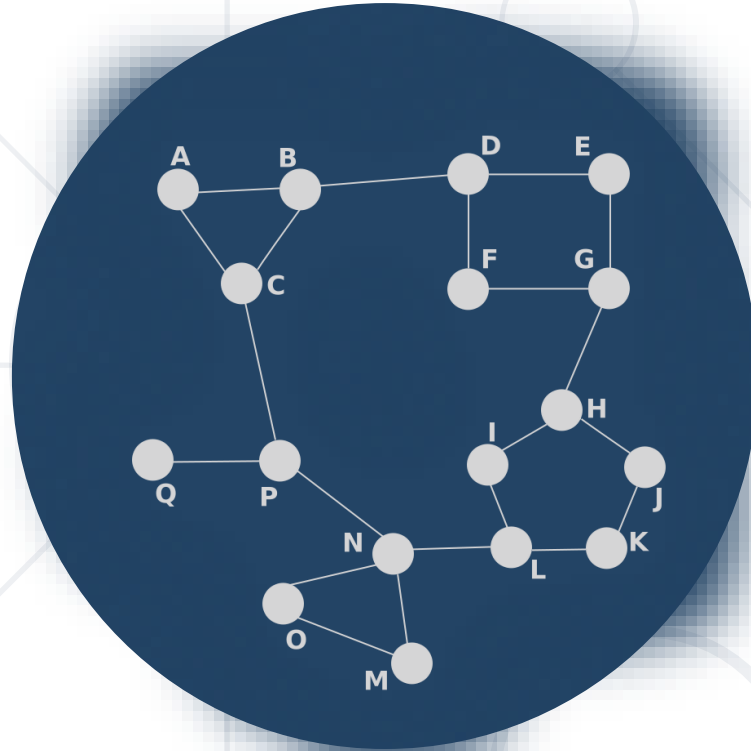
Example: Hyper Text Transfer Protocol

- HTTP request

```
GET /courses/about.aspx HTTP/1.1  
Host: www.softuni.com  
User-Agent: Mozilla/5.0  
<CRLF>
```

- HTTP response

```
HTTP/1.1 200 OK  
Date: Mon, 5 Jul 2010 13:09:03 GMT  
Server: Microsoft-HTTPAPI/2.0  
Last-Modified: Mon, 12 Jul 2010 15:33:23 GMT  
Content-Length: 54  
<CRLF>  
<html><title>Hello</title>Welcome to our site</html>
```

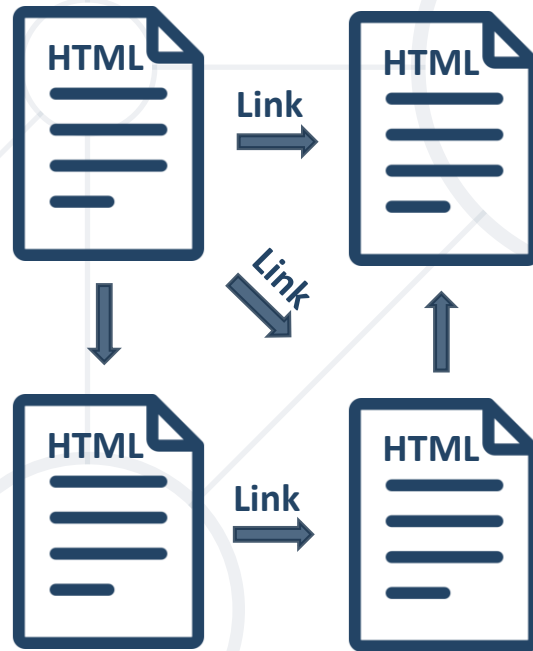



Routing Overview

Navigation for Single Page Apps

What is Routing?

- Allows navigation, **without reloading** the page
- Pivotal element of writing **Single Page Applications**



Standard Navigation



Navigation using Routing

- A **Router** loads the appropriate content when the **location changes**
 - E.g. when the user manually **enters an address**
- Conversely, a change in content is reflected in the address bar
 - E.g. when the user **clicks on a link**
- Benefits
 - Load all scripts **only once**
 - **Maintain state** across multiple pages
 - Browser **history** can be used
 - Build User Interfaces that **react quickly**





Angular Overview

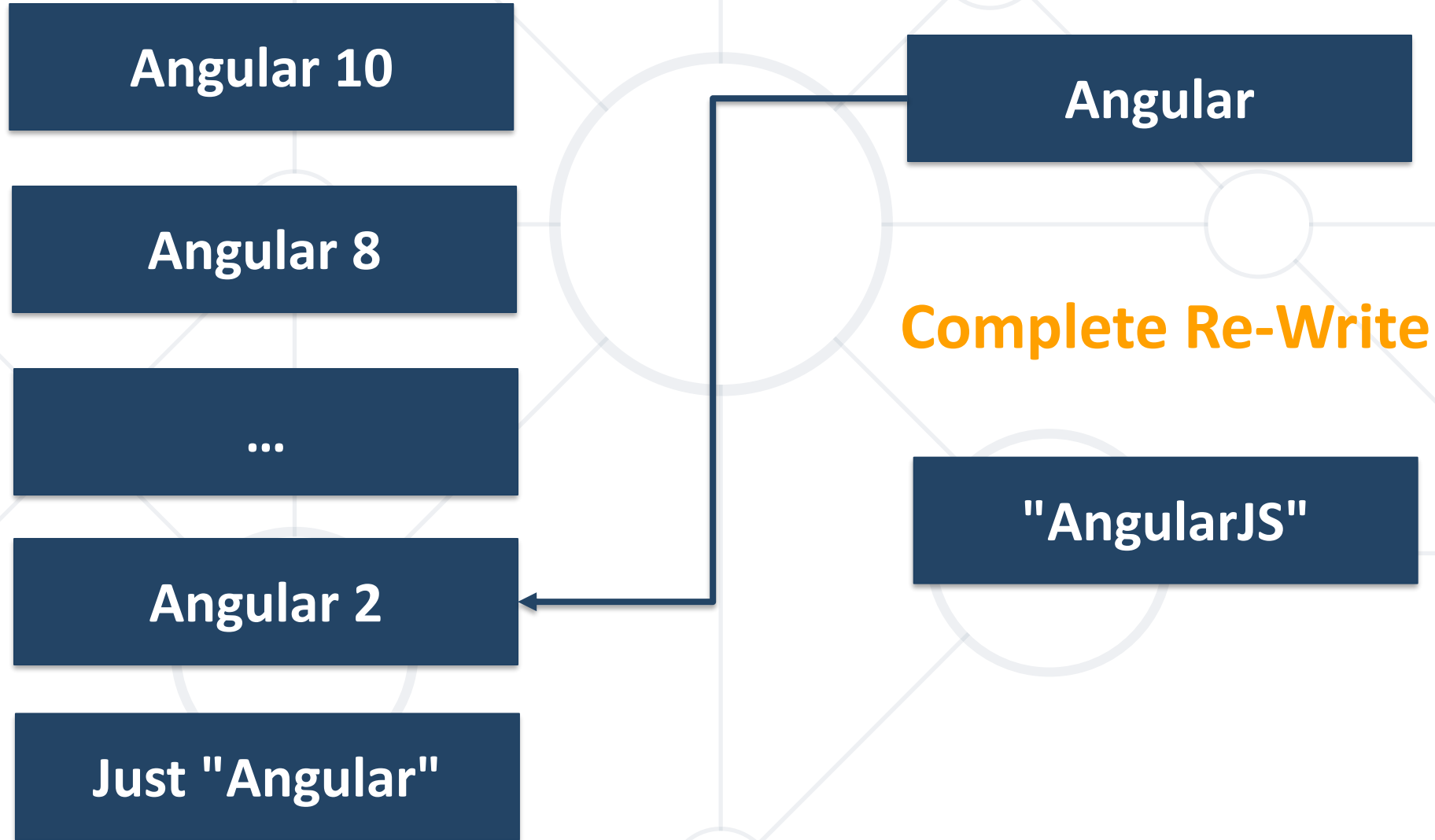
Web Application Platform

What is Angular?

- **Angular is a platform** for building complex front-end apps
- Focused on end-to-end **tooling** and **best practices**
- **Developed** by the Angular team at **Google**

```
import { Component } from '@angular/core';  
  
@Component({  
  selector: 'my-app',  
  template: `<h1>Hello {{name}}</h1>`  
})  
  
export class AppComponent { name = 'Angular'; }
```

Angular Versions





Introduction to TypeScript

A JavaScript Superset

Introduction to TypeScript

- Install globally via npm

```
npm install -g typescript
```

- TypeScript uses the .ts file extension (supported by VS Code)

```
tsc myfile.ts
```

- To compile your code
- Compilation output is plain JavaScript

Variable Types



```
let isDone: boolean = false;
```

```
let decimal: number = 6;  
let hex: number = 0xf00d;  
let binary: number = 0b1010;  
let octal: number = 0o744;
```

```
let color: string = "blue";  
color = 'red';
```

```
let list: number[] = [1, 2, 3];  
let list: Array<number> = [1, 2, 3];
```

More at <https://www.typescriptlang.org/docs/handbook/intro.html>

Classes

Access modifier could be
public/private/protected

```
class Greeter {  
  public greeting : string;  
  constructor(message : string) {  
    this.greeting = message;  
  }  
  greet() : string {  
    return `Hello, ${this.greeting}`;  
  }  
}
```

Functions could also
have a return type

```
let greeter : Greeter = new Greeter("world!");  
console.log(greeter.greet());
```



Inheritance

```
class Animal {  
  move(distanceInMeters: number = 0) : void {  
    console.log(`Animal moved ${distanceInMeters}m.`);  
  }  
}  
class Dog extends Animal {  
  bark() : void {  
    console.log('Woof! Woof!');  
  }  
}  
const dog = new Dog();  
dog.bark();  
dog.move(10);  
dog.bark();
```



Interfaces

```
function printLabel(labelledObj: { label: string }) {  
    console.log(labelledObj.label);  
}
```

Property assertion

```
let myObj = {size: 10, label: "Size 10 Object"};  
printLabel(myObj);
```

```
interface LabelledValue {  
    label: string;  
}  
  
function printLabel(labelledObj: LabelledValue) { ... }
```



Generics and Enumerations

```
function identity<T>(arg: T): T {  
    return arg;  
}  
  
let output = identity<string>("myString");  
// type of output will be 'string'  
  
let output = identity(5);  
// type of output will be 'number'
```

Type inference

```
enum Direction {  
    Up = 1,  
    Down,  
    Left,  
    Right,  
}
```



Modules

```
export default interface StringValidator {  
    isAcceptable(s: string): boolean;  
}
```

```
export { ZipCodeValidator };  
export { ZipCodeValidator as mainValidator };
```

```
import { ZipCodeValidator } from "../ZipCodeValidator";
```

```
import * as validator from "../ZipCodeValidator";
```

```
import num from "../OneTwoThree";
```





Angular Installation

Packages, Setup, Structure

Creating A New App

- Install globally via **npm**

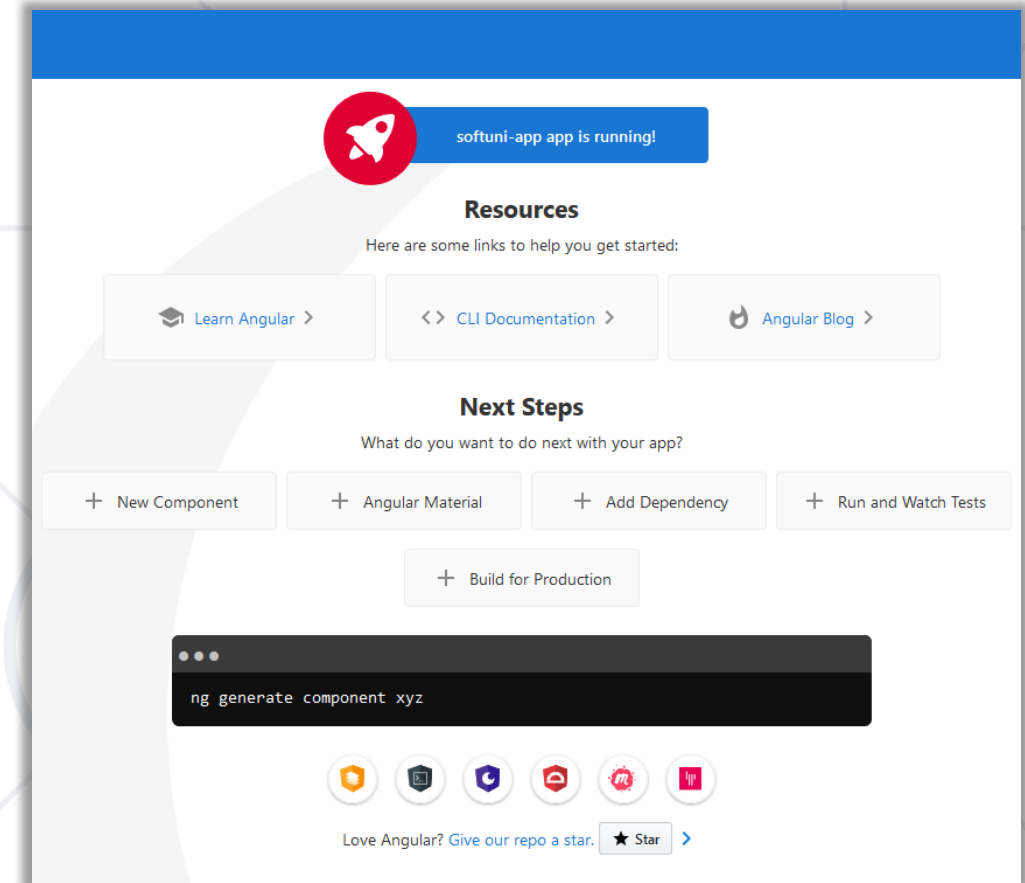
```
npm install -g @angular/cli
```

- **Create** new project

```
ng new some-app  
cd some-app
```

- **Start** a dev server on port 4200

```
ng serve
```



Finding Information

- Visit the **official website**

<https://angular.io/>

- Documentation

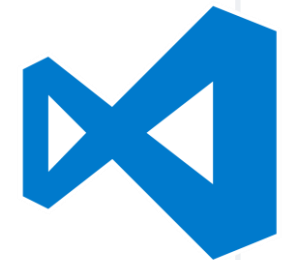
<https://angular.io/docs>

- Online sandbox

<https://plnkr.co/>



- **Visual Studio Code** fully supports TypeScript
 - You may use your favorite IDE (most have **plugins**)
- By using the **Angular CLI**
 - You do not need to use a **linter**
 - You do not need install any specific **plugin**
 - **Everything** is included





Live Demo

A Simple Angular App!

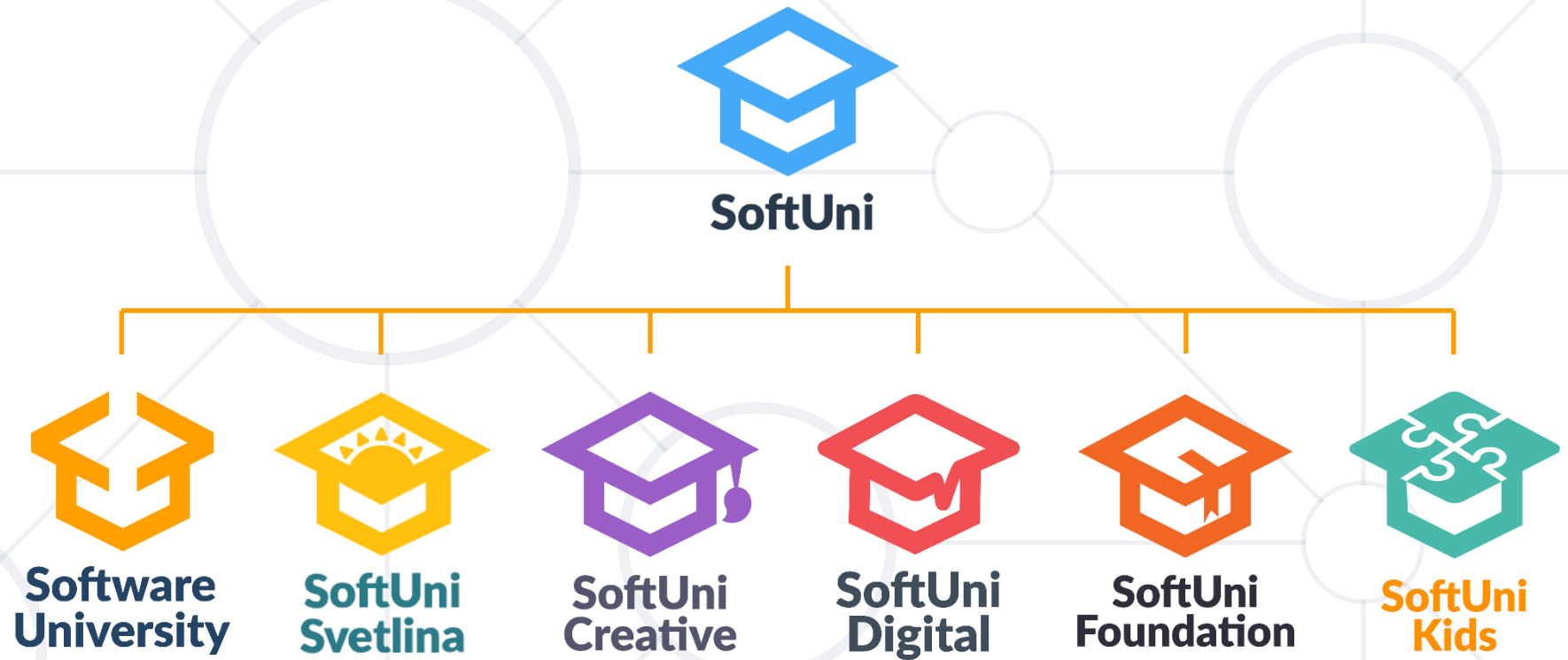
- **Angular** is a **framework** for front-end apps
- **TypeScript** is JavaScript **superset** language

```
interface LabelledValue {  
    label: string;  
}  
  
function print(labelledObj: LabelledValue) { ... }
```

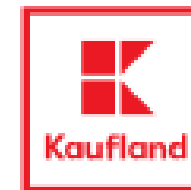
- The **Angular CLI** is a complete **toolkit** for working with Angular

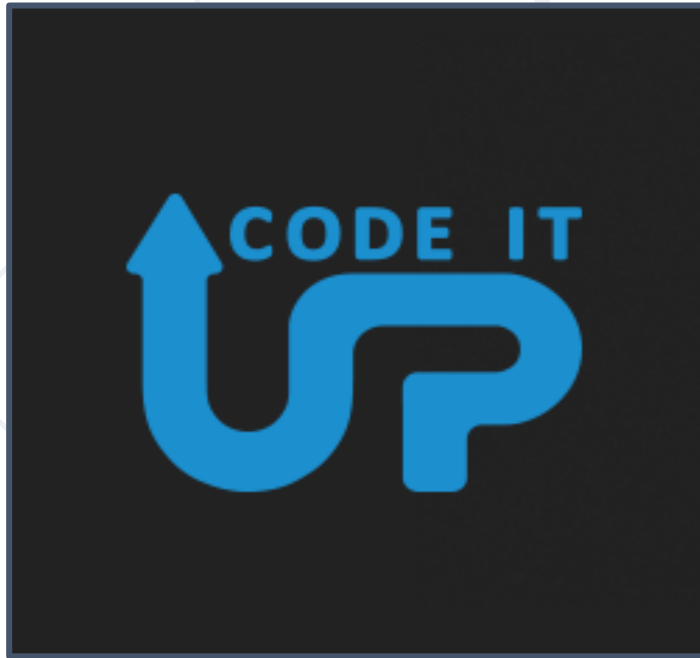


Questions?



SoftUni Diamond Partners





VIRTUAL RACING SCHOOL



- Software University – High-Quality Education, Profession and Job for Software Developers

- softuni.bg, softuni.org

- Software University Foundation

- softuni.foundation

- Software University @ Facebook

- facebook.com/SoftwareUniversity

- Software University Forums

- forum.softuni.bg



Software University



- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – <https://softuni.org>
- © Software University – <https://softuni.bg>

