



Sorsix Training

Release Version 0.2

Sorsix International

May 08, 2019

Table of contents

1	Front End Development	3
2	Back End Development	9
3	Kotlin	13

This is the home of Sorsix internal training.

1.1 HTML5 and CSS

1.1.1 Goals

- Refresh knowledge and practical experience with HTML5 and CSS.
- Create semantically and structure correct HTML layouts.

1.1.2 Exercise

<https://learn.freecodecamp.org/responsive-web-design/basic-html-and-html5>

1.1.3 Estimated time

2-4 hours

1.2 Basic CSS

1.2.1 Goals

- Introduction to basic CSS.
- color, fonts, positioning, spacing, sizing, decorations, transitions

1.2.2 Exercise

<https://learn.freecodecamp.org/responsive-web-design/basic-css>

1.2.3 Estimated time

2-4 hours

1.3 Applied Visual Design

1.3.1 Goals

- Learn typography, color theory, graphics, animation, and page layout to help deliver a site's message.
- Learn the basic concepts of solid UI/UX.

1.3.2 Exercise

<https://learn.freecodecamp.org/responsive-web-design/applied-visual-design>

1.3.3 Estimated time

4-6 hours

1.4 Responsive Web Design

1.4.1 Goals

- Learn designing web content that responds to the constraints of different devices.
- Learn the 'mobile-first' approach.

1.4.2 Exercise

<https://learn.freecodecamp.org/responsive-web-design/responsive-web-design-principles>

1.4.3 Estimated time

2-4 hours

1.5 CSS Flexbox

1.5.1 Goals

- Learn CSS3 Flexible Boxes, or flexbox, to create page layouts for a dynamic UI.

1.5.2 Exercise

<https://learn.freecodecamp.org/responsive-web-design/css-flexbox>

1.5.3 Estimated time

2-4 hours

1.6 CSS Grid

1.6.1 Goals

- Learn to build complex web designs by turning an HTML element into a grid container with rows and columns for you to place children elements where you want within the grid.

1.6.2 Exercise

<https://learn.freecodecamp.org/responsive-web-design/css-grid>

1.6.3 Estimated time

2-4 hours

1.7 Basic JavaScript

1.7.1 Goals

- Learn or refresh basic JavaScript programming concepts, which range from variables and arithmetic to objects and loops.

1.7.2 Exercise

<https://learn.freecodecamp.org/javascript-algorithms-and-data-structures/basic-javascript>

<https://learn.freecodecamp.org/javascript-algorithms-and-data-structures/basic-data-structures>

1.7.3 Estimated time

5-10 hours

1.8 Object Oriented and Functional Programming

1.8.1 Goals

- Learn OO JavaScript and learn some basic Functional Programming concepts.

1.8.2 Exercise

<https://learn.freecodecamp.org/javascript-algorithms-and-data-structures/object-oriented-programming>

<https://learn.freecodecamp.org/javascript-algorithms-and-data-structures/functional-programming>

1.8.3 Estimated time

1-2 hours

1.9 Basic Algorithms

1.9.1 Goals

- Solve at least 5 algorithmic problems to refresh your algorithmic skills.

1.9.2 Exercise

<https://learn.freecodecamp.org/javascript-algorithms-and-data-structures/basic-algorithm-scripting>

<https://learn.freecodecamp.org/javascript-algorithms-and-data-structures/intermediate-algorithm-scripting>

1.9.3 Estimated time

2-4 hours

1.10 Intermediate Front End Development Projects in Angular

1.10.1 Goals

- Gain experience and test your knowledge by developing several practical projects in Angular.

1.10.2 Setup Angular Development environment

<https://angular.io/guide/quickstart#devenv>

1.10.3 Estimated time

1-2 hours

1.11 Angular Hero Tutorial

1.11.1 Goals

- Learn Angular by using the official documentation by completing the project Angular Heroes Tour.

1.11.2 Exercise

<https://angular.io/tutorial>

1.11.3 Estimated time

12-16 hours

1.12 Random Quote

1.12.1 Goals

- Develop a simple front end application using a very simple API.

1.12.2 Exercise

<https://learn.freecodecamp.org/front-end-libraries/front-end-libraries-projects/build-a-random-quote-machine>

1.12.3 Estimated time

3-4 hours

1.12.4 GitLab

<https://gitlab.sorsix.com/tdelev/random-quote-spring-2018>

1.13 TV Shows

1.13.1 Goals

- Develop a medium sized multiple components Angular app.

1.13.2 Exercise

TV Shows

1.13.3 Estimated time

8-16 hours

2.1 Spring Framework Introduction

2.1.1 Spring Context and IOC/DI

<https://docs.spring.io/spring/docs/current/spring-framework-reference/index.html>

2.1.2 Spring MVC

<https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html#spring-web>

2.1.3 Spring Data JPA

<https://docs.spring.io/spring-data/jpa/docs/current/reference/html/>

2.1.4 Spring Boot

<https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>

2.1.5 Gradle setup

2.2 Timestamp Microservice

2.2.1 Goals

- Develop a simple timestamping microservice API.

2.2.2 Exercise

<https://learn.freecodecamp.org/apis-and-microservices/apis-and-microservices-projects/timestamp-microservice>

2.2.3 Estimated time

4-6 hours

2.2.4 GitLab

<https://gitlab.sorsix.com/tdelev/timestamp-interns>

2.3 URL Shortener Microservice

2.3.1 Goals

- Develop a simple URL shortener API.
- Learn persistence and DB access using JPA.

2.3.2 Exercise

<https://learn.freecodecamp.org/apis-and-microservices/apis-and-microservices-projects/url-shortener-microservice/>

2.3.3 Estimated time

10-12 hours

2.3.4 GitLab

<https://gitlab.sorsix.com/tdelev/url-shortener-interns>

2.4 Final project

2.4.1 Goals

- Final project for developing full stack application with back end and front end.
- Design the DB from the requirements.
- Implement full API with user authentication and security.
- Implement authentic UI and front end application in Angular.

2.4.2 Exercise

- <https://learn.freecodecamp.org/coding-interview-prep/take-home-projects/build-a-voting-app/>
- <https://learn.freecodecamp.org/coding-interview-prep/take-home-projects/build-a-nightlife-coordination-app/>
- <https://learn.freecodecamp.org/coding-interview-prep/take-home-projects/chart-the-stock-market/>
- <https://learn.freecodecamp.org/coding-interview-prep/take-home-projects/manage-a-book-trading-club/>
- <https://learn.freecodecamp.org/coding-interview-prep/take-home-projects/build-a-pinterest-clone/>

2.4.3 Estimated time

40-60 hours

2.4.4 GitLab

<https://gitlab.sorsix.com/tdelev/interns-final-project>

2.5 External resources

2.5.1 Foundations of RESTful Architecture

<https://dzone.com/refcardz/rest-foundations-restful>

2.5.2 Getting Started with Git

<https://dzone.com/refcardz/getting-started-git>

2.5.3 Spring Annotations

<https://dzone.com/refcardz/spring-annotations>

2.5.4 Core Java

<https://dzone.com/refcardz/core-java>

2.5.5 Getting started with JPA

<https://dzone.com/refcardz/getting-started-with-jpa>

2.5.6 What's New in JPA 2.0

<https://dzone.com/refcardz/whats-new-jpa-20>

2.5.7 Java Performance Optimization

<https://dzone.com/refcardz/java-performance-optimization>

2.5.8 Getting Started With Docker

<https://dzone.com/refcardz/getting-started-with-docker-1>

3.1 Learn Kotlin

Having a solid Java background any developer should be able at least to read Kotlin code and understand good part of it. Best way to get on track and learn the basics of the language is the official documentation and reference on the language home page <https://kotlinlang.org>.

3.1.1 Kotlin by Example

Learning Kotlin by example should be different kind of fun.

<https://play.kotlinlang.org/byExample/overview>

3.1.2 Kotlin Koans

Once you have covered the basics of the language the best way to learn it, would be to spend some time hands on with the Kotlin Koans tutorial.

<https://kotlinlang.org/docs/tutorials/koans.html>

3.1.3 Kotlin Presentations

<https://www.youtube.com/watch?v=7EVXypZDOos&t=1s>

3.1.4 Kotlin Courses

<https://www.coursera.org/learn/kotlin-for-java-developers>

3.2 Kotlin in Spring

Kotlin can be used in any project that compiles for JVM, and Spring Framework projects are no exception.

3.2.1 Building web applications with Spring Boot and Kotlin

A good tutorial with a sample blog application by combining the power of Spring Boot and Kotlin.

<https://spring.io/guides/tutorials/spring-boot-kotlin/>

3.2.2 Integrating Kotlin in existing Spring Boot project

In your project `build.gradle` file add:

Listing 3.1: Add Kotlin dependencies

```
buildscript {
    ext {
        kotlinVersion = '1.3.10'
    }
    repositories {
        mavenCentral()
    }
    dependencies {
        ...
        classpath("org.jetbrains.kotlin:kotlin-gradle-plugin:${kotlinVersion}")
        ...
    }
}

apply plugin: 'kotlin'
```

Compiler plugins

For better convenience you should use the **compiler plugins**.

Listing 3.2: Add compiler plugins

```
apply plugin: 'kotlin-spring'
apply plugin: 'kotlin-jpa'
```

Compile Kotlin to Java 1.8

Listing 3.3: Compile configuration

```
compileKotlin {
    kotlinOptions.jvmTarget = JavaVersion.VERSION_1_8
}

compileTestKotlin {
    kotlinOptions.jvmTarget = JavaVersion.VERSION_1_8
}
```

```
tasks.withType(org.jetbrains.kotlin.gradle.tasks.KotlinCompile).all {  
    kotlinOptions {  
        jvmTarget = JavaVersion.VERSION_1_8  
    }  
}
```

Standard Library dependencies

Listing 3.4: Standard library dependency for JDK 1.8

```
dependencies {  
    ...  
    compile "org.jetbrains.kotlin:kotlin-stdlib-jdk8"  
    compile "org.jetbrains.kotlin:kotlin-reflect"  
    ...  
}
```

3.3 Sorsix projects using Kotlin

Here is a list of some of our projects and libraries using Kotlin:

- <https://gitlab.sorsix.com/sorsix/ild>
- <https://gitlab.sorsix.com/sorsix/matrix-education>
- <https://gitlab.sorsix.com/sorsix/roa>
- <https://gitlab.sorsix.com/sorsix/kemarto>
- <https://gitlab.sorsix.com/sorsix/html2pdf>
- <https://gitlab.sorsix.com/sorsix/pdf2html>
- <https://gitlab.sorsix.com/sorsix/docs2pdf>
- <https://gitlab.sorsix.com/sorsix/job-queue>
- <https://gitlab.sorsix.com/sorsix/libs/dbdoc>
- <https://gitlab.sorsix.com/training/voting>