

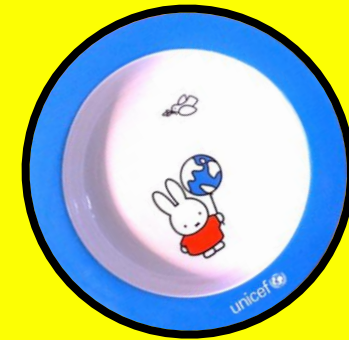
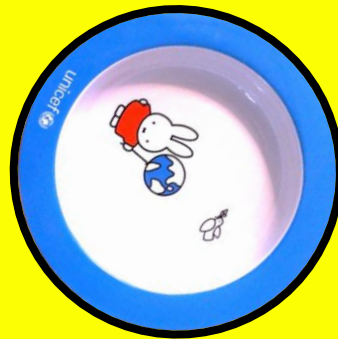
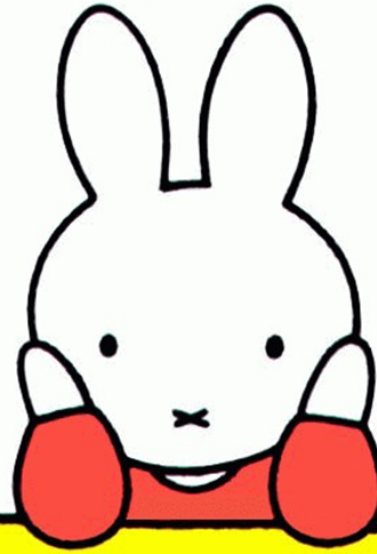
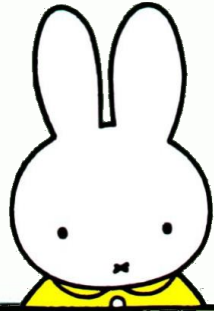
Bsidesoft co.





1998

Dick bruna



CODE SPITZ

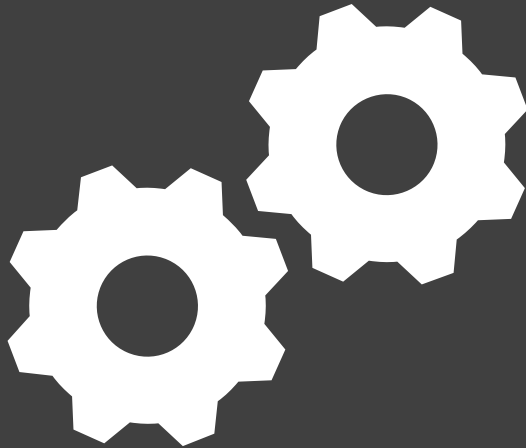


82

KOTLIN ELEMENTARY



setting



Gradle project

Gradle project

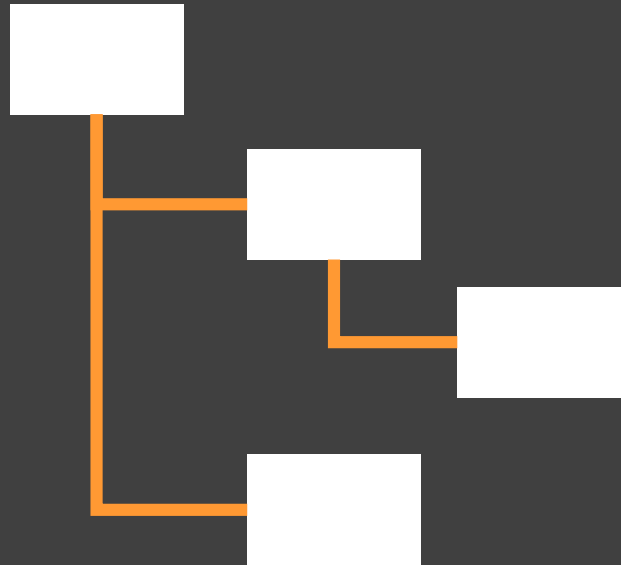
그라이들을 이용하여 빌드되는 프로젝트타입

Gradle project

그래이들을 이용하여 빌드되는 프로젝트타입
계층 구조로 구성 가능

Gradle project

그래이들을 이용하여 빌드되는 프로젝트타입
계층 구조로 구성 가능

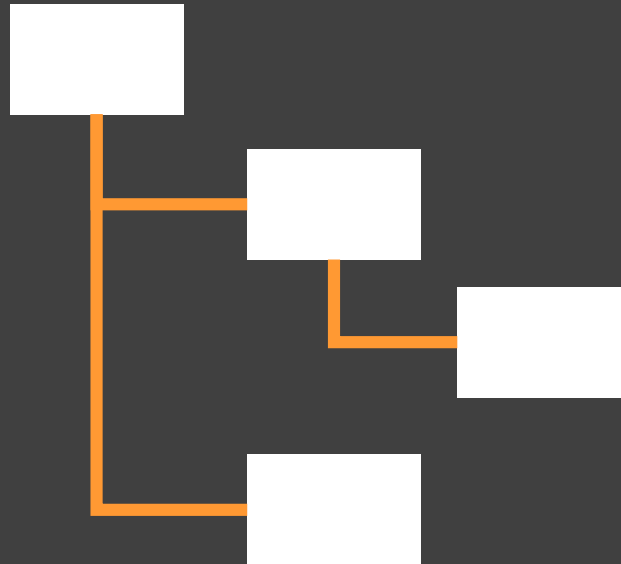


Gradle project

그래이들을 이용하여 빌드되는 프로젝트타입

계층 구조로 구성 가능

같은 계층 내의 자식만 모듈 간 의존성을 설정 가능

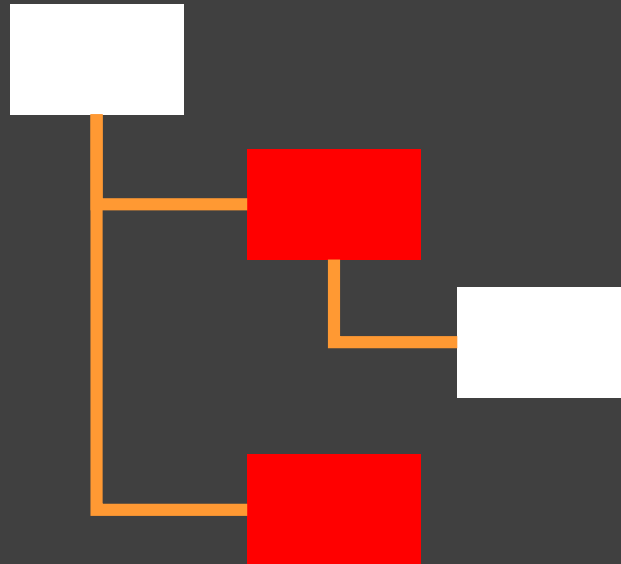


Gradle project

그래이들을 이용하여 빌드되는 프로젝트타입

계층 구조로 구성 가능

같은 계층 내의 자식만 모듈 간 의존성을 설정 가능



IntelliJ

IntelliJ

하나의 프로젝트에 다수 모듈을 갖음

IntelliJ

하나의 프로젝트에 다수 모듈을 갖음
목적파일 하나당 별도 모듈 필요

IntelliJ

하나의 프로젝트에 다수 모듈을 갖음

목적파일 하나당 별도 모듈 필요

안드로이드: apk1, apk2 - 모듈 2개 필요

IntelliJ

하나의 프로젝트에 다수 모듈을 갖음

목적파일 하나당 별도 모듈 필요

안드로이드: apk1, apk2 - 모듈 2개 필요

자바: jar1, jar2 - 모듈 2개 필요

IntelliJ

하나의 프로젝트에 다수 모듈을 갖음

목적파일 하나당 별도 모듈 필요

안드로이드: apk1, apk2 - 모듈 2개 필요

자바: jar1, jar2 - 모듈 2개 필요

app.js, lib.js - 모듈 2개 필요

IntelliJ

하나의 프로젝트에 다수 모듈을 갖음

목적파일 하나당 별도 모듈 필요

안드로이드: apk1, apk2 - 모듈 2개 필요

자바: jar1, jar2 - 모듈 2개 필요

app.js, lib.js - 모듈 2개 필요

app이 lib에 의존적

IntelliJ

하나의 프로젝트에 다수 모듈을 갖음

목적파일 하나당 별도 모듈 필요

안드로이드: apk1, apk2 - 모듈 2개 필요

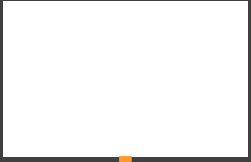
자바: jar1, jar2 - 모듈 2개 필요

app.js, lib.js - 모듈 2개 필요

app이 lib에 의존적

부모 그래이드들의 자식으로

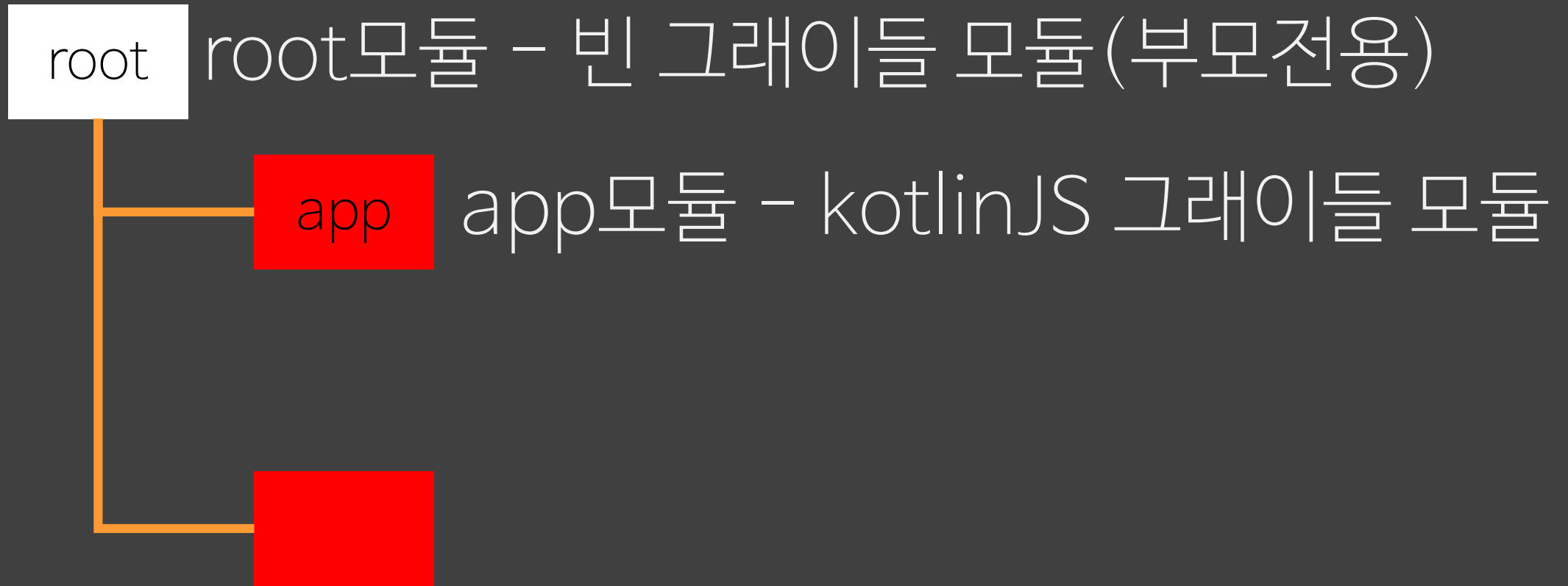
app그래이드들, lib그래이드들 모듈을 만듦

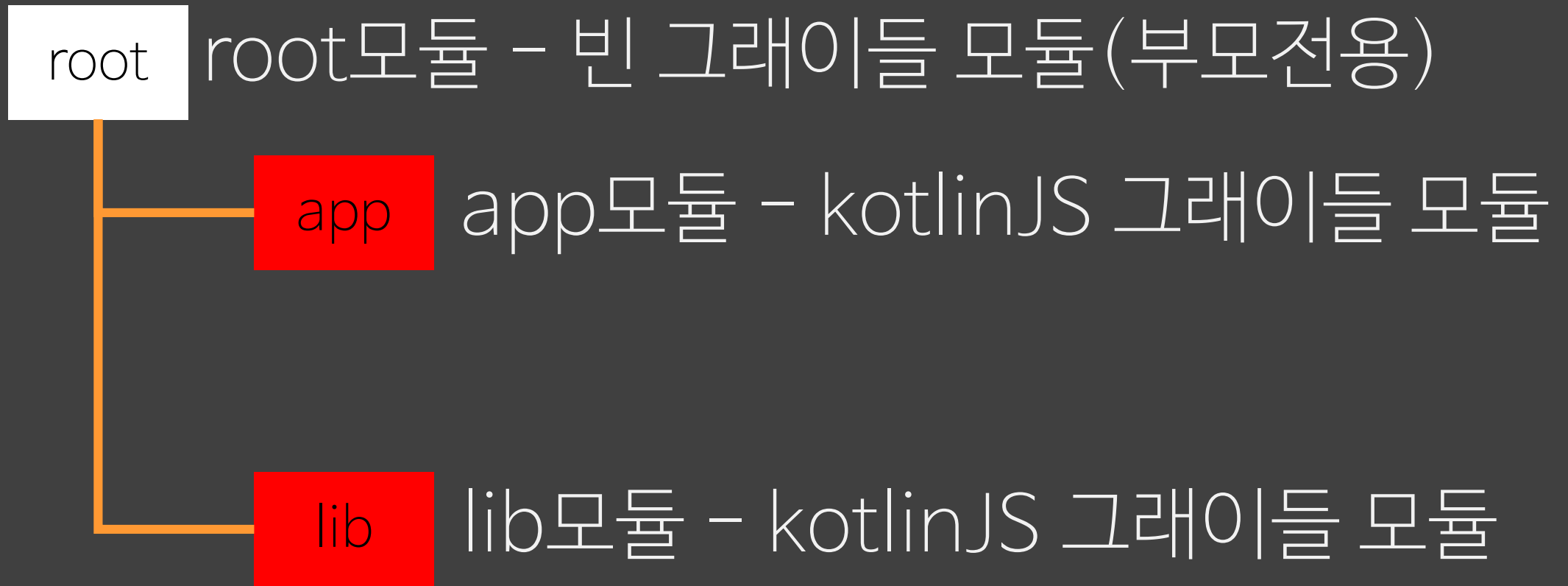


root

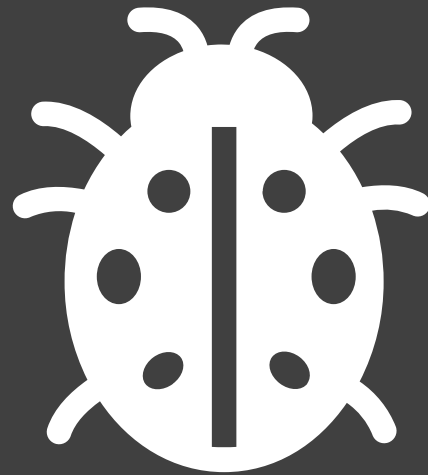
root모듈 - 빈 그래이들 모듈(부모전용)








project





new Project





new Project


 New Project ✕


 Java


 Java FX


 IntelliJ Platform Plugin


 Maven


 Gradle

 Groovy

 Scala


 Kotlin


 Empty Project


Project SDK:  11 (java version "11.0.1") ▼ New...


☐ Kotlin DSL build script


Additional Libraries and Frameworks:

☒  Java

☐  Groovy


☐  IntelliJ Platform Plugin

☐  Kotlin/JS

☐  Kotlin/JVM

Previous Next Cancel Help

new Project

 New Project ×

GroupId

ArtifactId

codespitz82

Version


Previous

Next

Cancel

Help

new Project

 New Project ✕

☒ Use auto-import

Group modules: ☐ using explicit module groups ☒ using qualified names

☐ Create separate module per source set

☒ Use default Gradle wrapper (recommended)
☐ Use local Gradle distribution

Gradle home:

Gradle JVM:


Previous

Next

Cancel

Help

new Project

 New Project ✕

Project name:

Project location: ...

▶ More Settings

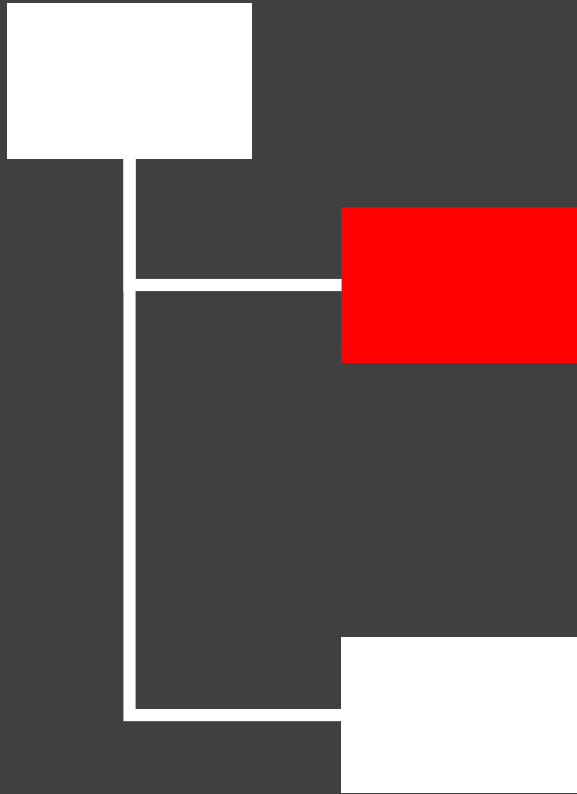
Previous

Finish

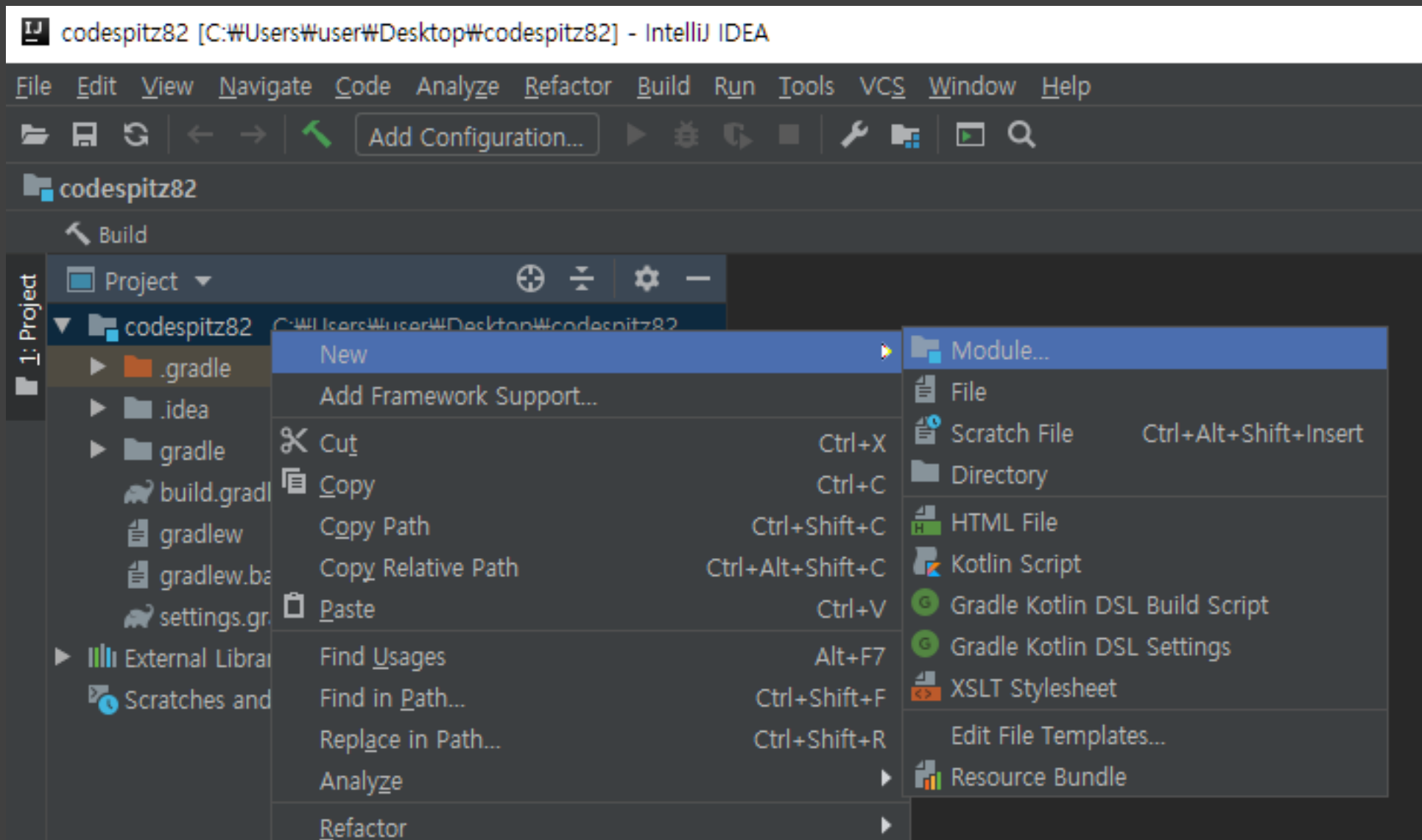
Cancel

Help

app module



app module



app module

New Module

Java

IntelliJ Platform Plugin

Maven

Gradle

Groovy

Scala

Kotlin

Empty Project

Module SDK:

Project SDK (11)

☐

Kotlin DSL build script

Additional Libraries and Frameworks:

☐

Java

☐

Groovy

☐

IntelliJ Platform Plugin

☒

Kotlin/JS

☐

Kotlin/JVM

A Kotlin library or application targeting JavaScript


Previous


Next

Cancel

Help

app module

 New Module ×

Add as module to codespitz82 

GroupId

ArtifactId

Version

☒ Inherit

☐ Inherit

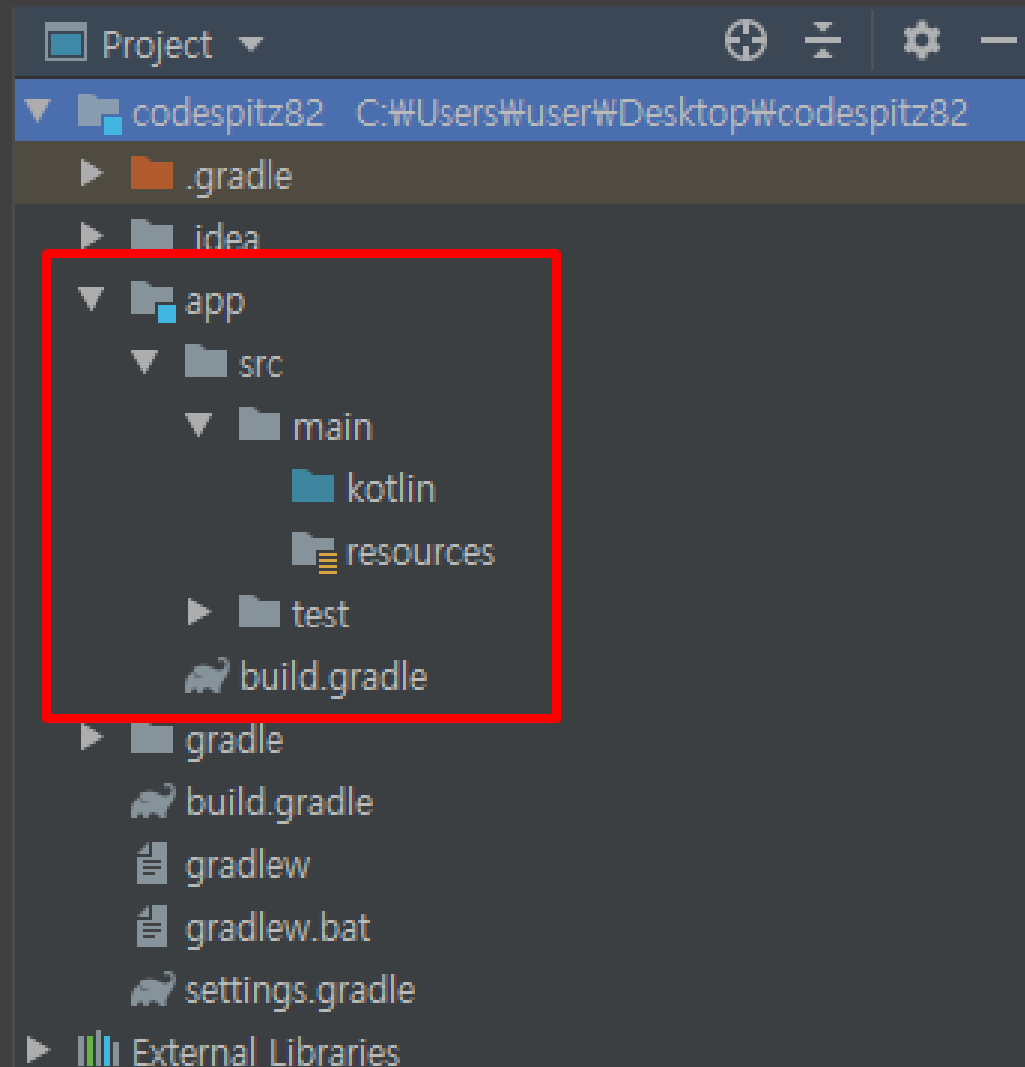
Previous

Next

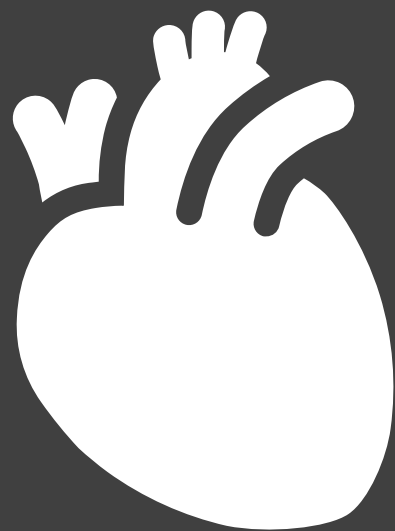
Cancel

Help

app module



hello world



hello world

Project ▾

codespitz82 C:\Users\user\Desktop\codespitz82

- .gradle
- .idea
- ▾ app
 - ▾ src
 - ▾ main
- kotlin**
- resources
- test
- build.gradle
- gradle
- ▾ lib
 - ▾ src
 - ▾ main

New

- Cut Ctrl+X
- Copy Ctrl+C
- Copy Path Ctrl+Shift+C
- Copy Reference Ctrl+Alt+Shift+C
- Paste Ctrl+V
- Find Usages Alt+F7

Kotlin File/Class

File

Scratch File Ctrl+Alt+Shift+Insert

Directory


HTML File

Kotlin Script

JavaFXApplication

```
1 plugins {  
2     id 'kotlin2js' version '1.  
3 }  
4  
5  
6 repositories {  
7     mavenCentral()
```



hello world

 New Kotlin File/Class ✕

Name:

↑↓

Kind:

 File

▼

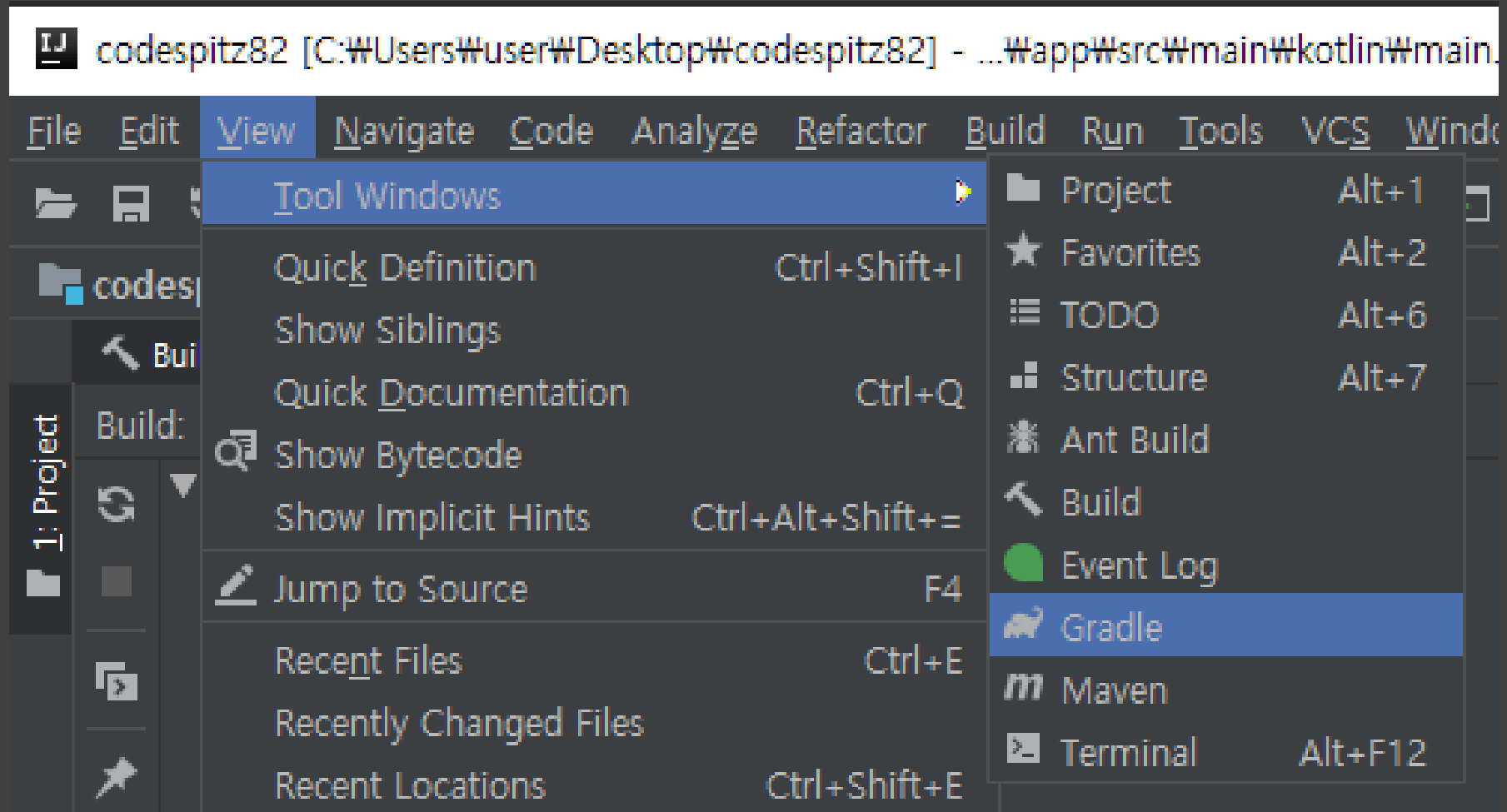
OK

Cancel

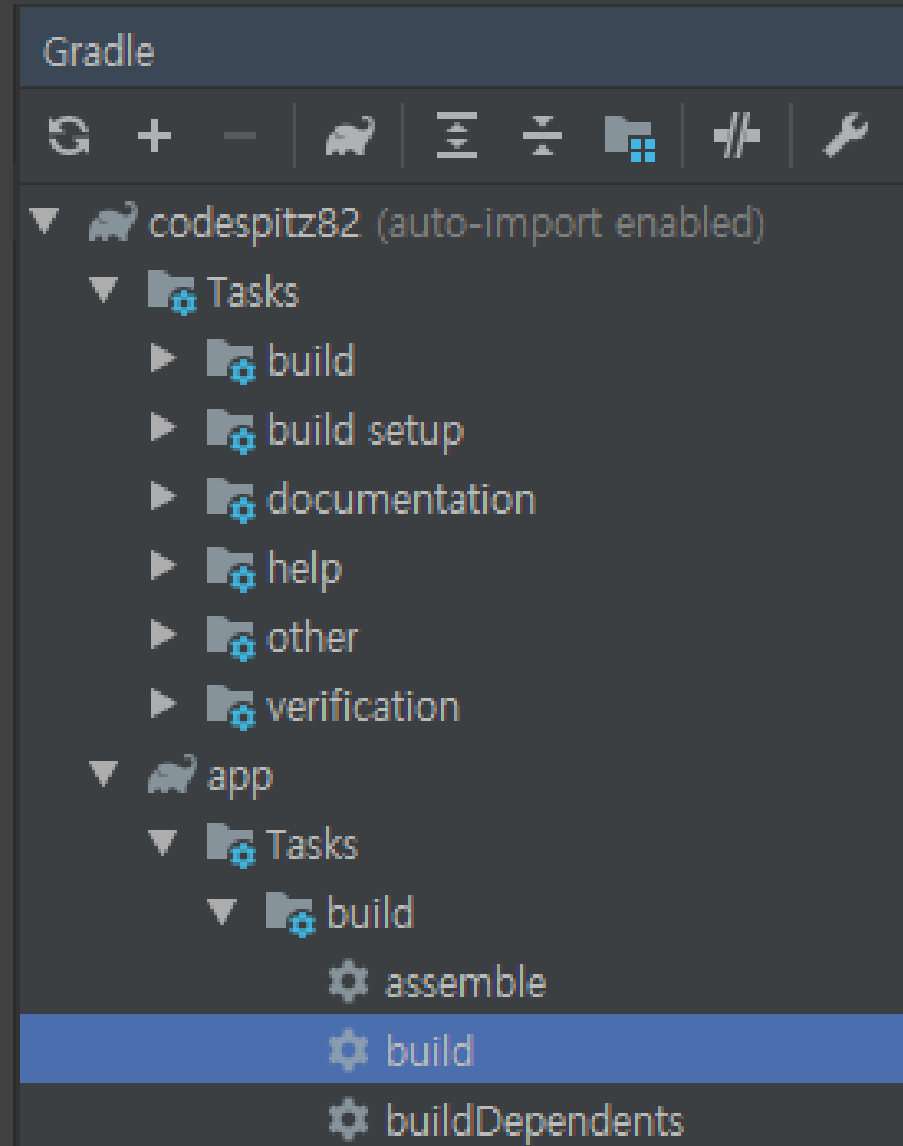
hello world

```
fun main(){  
    println("hello world")  
}
```

hello world



hello world



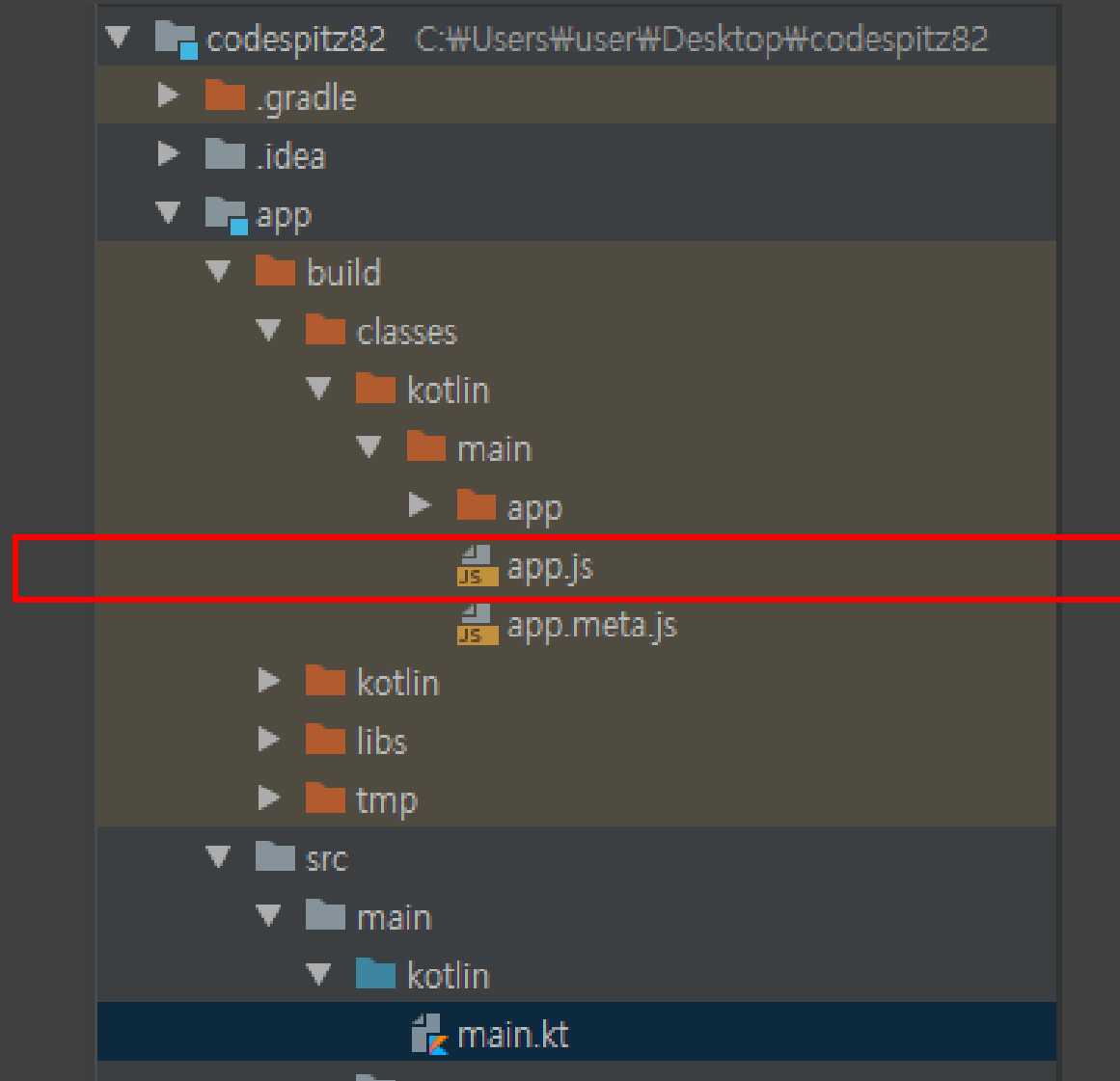
hello world

```
Run: codespitz82:app [build] x
오후 3:35:22: Executing task 'build'...

> Task :lib:compileJava NO-SOURCE
> Task :lib:compileKotlin2Js NO-SOURCE
> Task :lib:processResources NO-SOURCE
> Task :lib:classes UP-TO-DATE
> Task :lib:inspectClassesForKotlinIC
> Task :lib:jar
> Task :app:compileJava NO-SOURCE
> Task :app:compileKotlin2Js
> Task :app:processResources NO-SOURCE
> Task :app:classes UP-TO-DATE
> Task :app:compileTestJava NO-SOURCE
> Task :app:processTestResources NO-SOURCE
> Task :app:testClasses UP-TO-DATE
> Task :app:test NO-SOURCE
> Task :app:check UP-TO-DATE
> Task :app:build

BUILD SUCCESSFUL in 5s
5 actionable tasks: 5 executed
오후 3:35:28: Task execution finished 'build'.
```

hello world

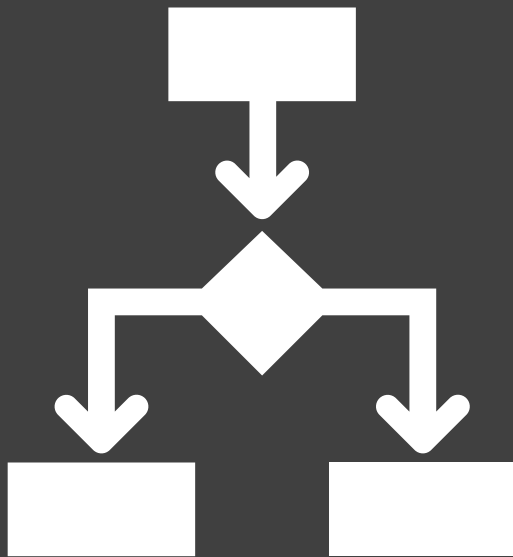


app.js

```
if (typeof kotlin === 'undefined') {  
    throw new Error("Error loading module 'app'. Its deper  
}
```

```
var app = function (_, Kotlin) {  
    'use strict';  
    var println = Kotlin.kotlin.io.println_s8jyv4$;  
    function main() {  
        println('hello world');  
    }  
    _.main = main;  
    main();  
    Kotlin.defineModule('app', _);  
    return _;  
}(typeof app === 'undefined' ? {} : app, kotlin);
```

build.gradle




app/build.gradle

```
plugins {  
    id 'kotlin2js' version '1.3.31'  
}  
  
repositories {  
    mavenCentral()  
}  
  
dependencies {  
    implementation "org.jetbrains.kotlin:kotlin-stdlib-js"  
    testImplementation "org.jetbrains.kotlin:kotlin-test-js"  
}
```

app/build.gradle

```
plugins {  
    id 'kotlin2js' version '1.3.31'  
}  
  
repositories {  
    mavenCentral()  
}  
  
dependencies {  
    implementation "org.jetbrains.kotlin:kotlin-stdlib-js"  
    testImplementation "org.jetbrains.kotlin:kotlin-test-js"  
}
```



<https://kotlinlang.org/docs/tutorials/javascript/getting-started-gradle/getting-started-with-gradle.html>

shorturl.at/mGHR3

app/build.gradle

```
buildscript {  
    ext.kotlin_version = '1.3.31'  
    repositories {  
        mavenCentral()  
    }  
    dependencies {  
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"  
    }  
}  
  
apply plugin: 'kotlin2js'  
  
repositories {  
    mavenCentral()  
}  
  
dependencies {  
    compile "org.jetbrains.kotlin:kotlin-stdlib-js:$kotlin_version"  
}
```

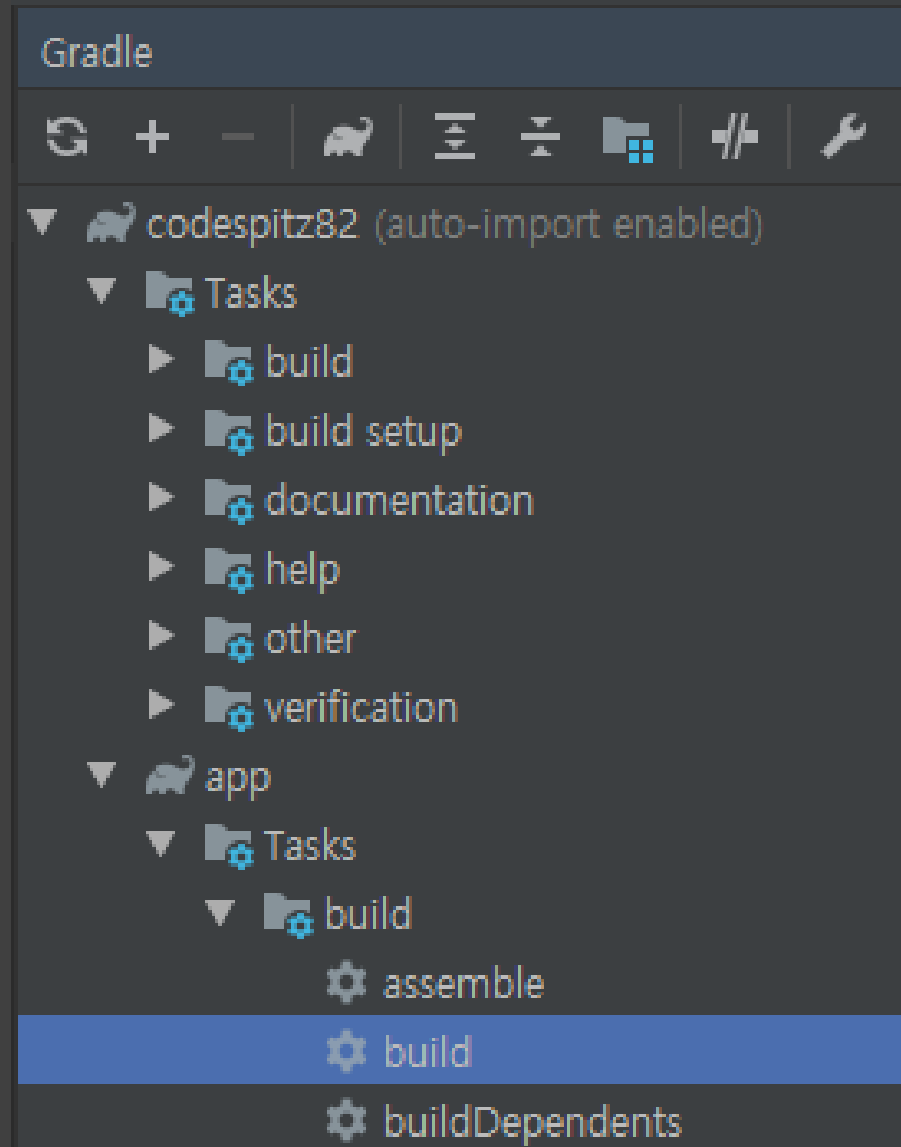
app/build.gradle

```
task assembleWeb(type: Sync) {
    configurations.compile.each { file ->
        from(zipTree(file.absolutePath), {
            includeEmptyDirs = false
            include { fileTreeElement ->
                def path = fileTreeElement.path
                path.endsWith(".js") && (path.startsWith("META-INF/resources/") ||
                    !path.startsWith("META-INF/"))
            }
        })
    }
    from compileKotlin2Js.destinationDir
    into "${projectDir}/web"

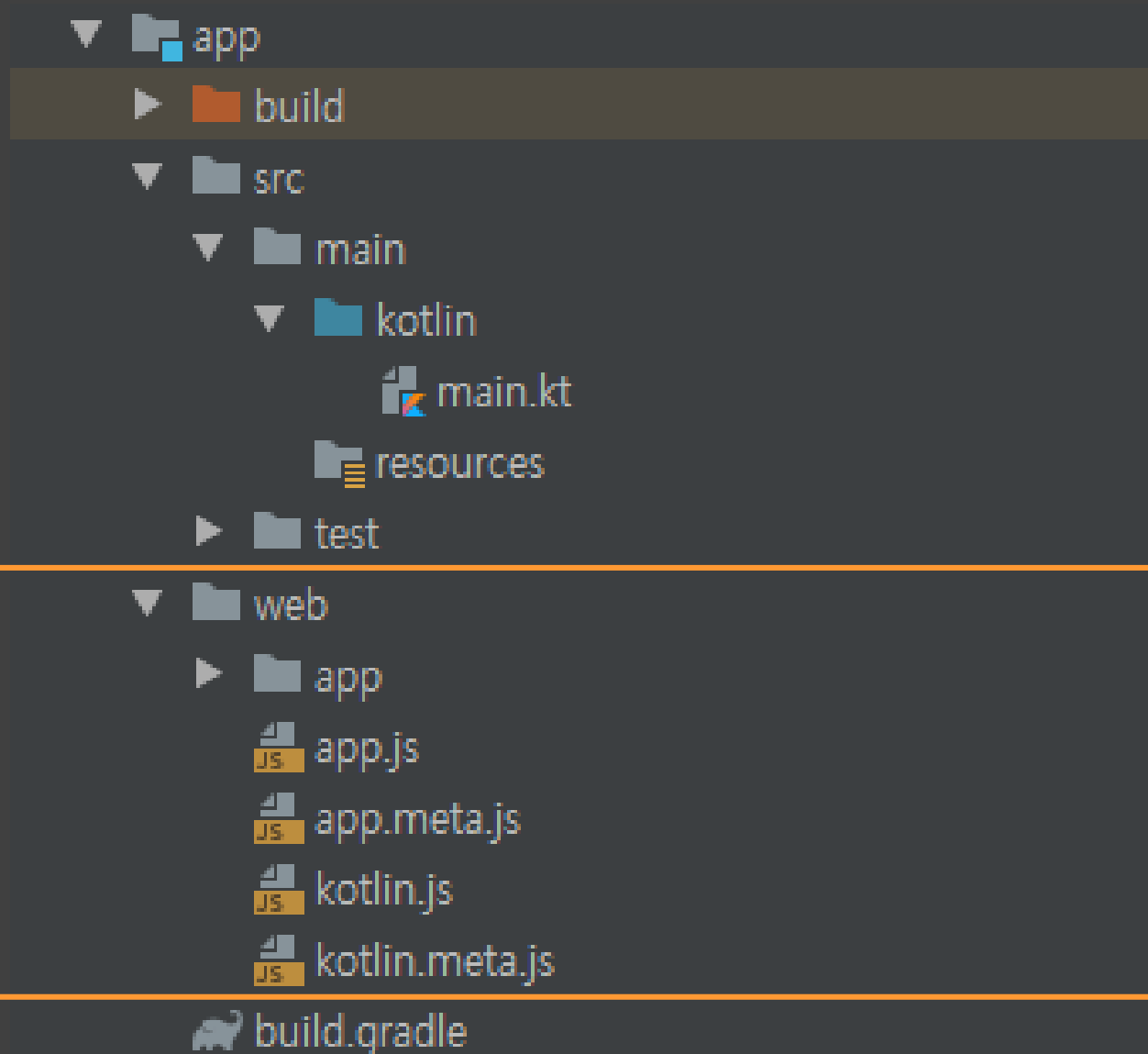
    dependsOn classes
}

assemble.dependsOn assembleWeb
```

build



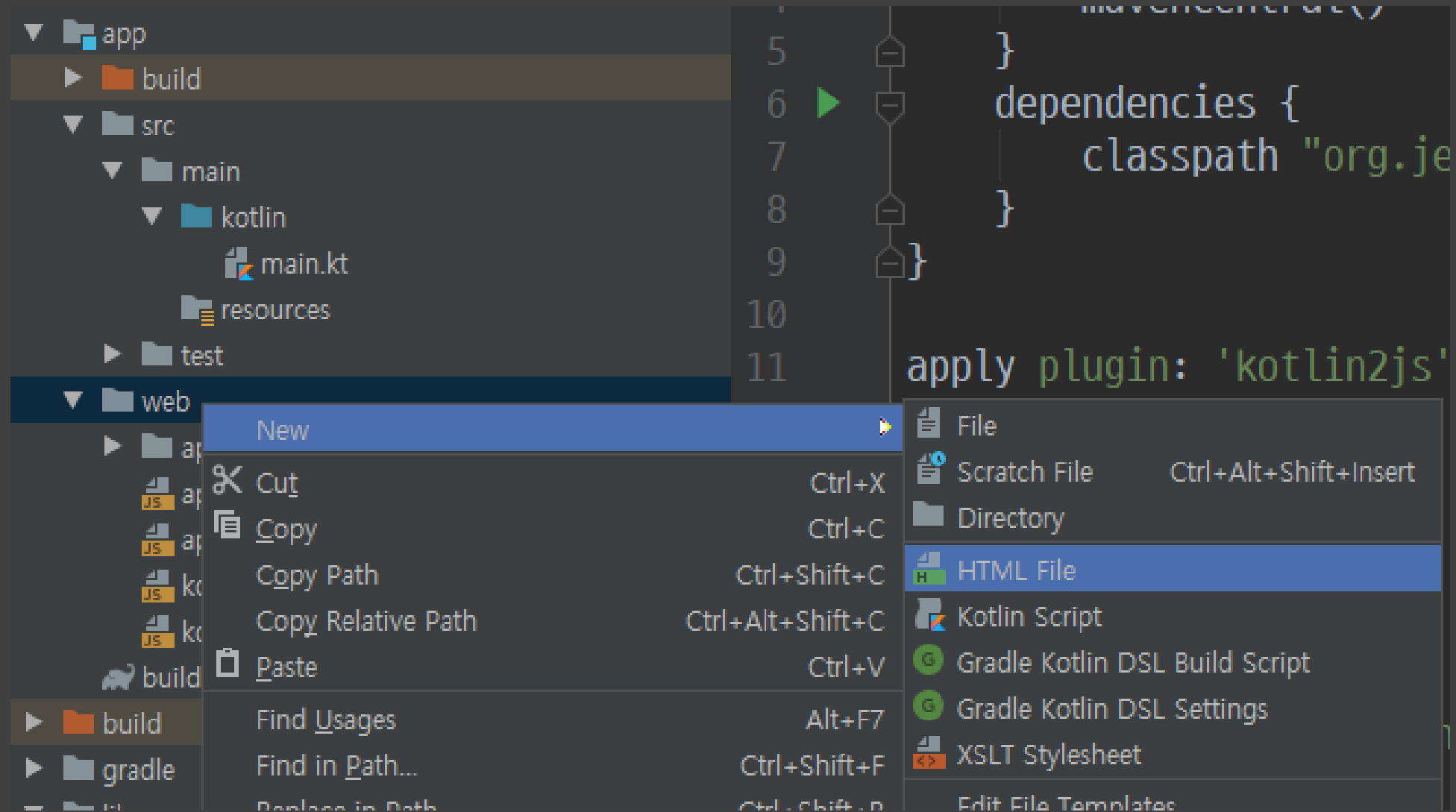
build



browser test



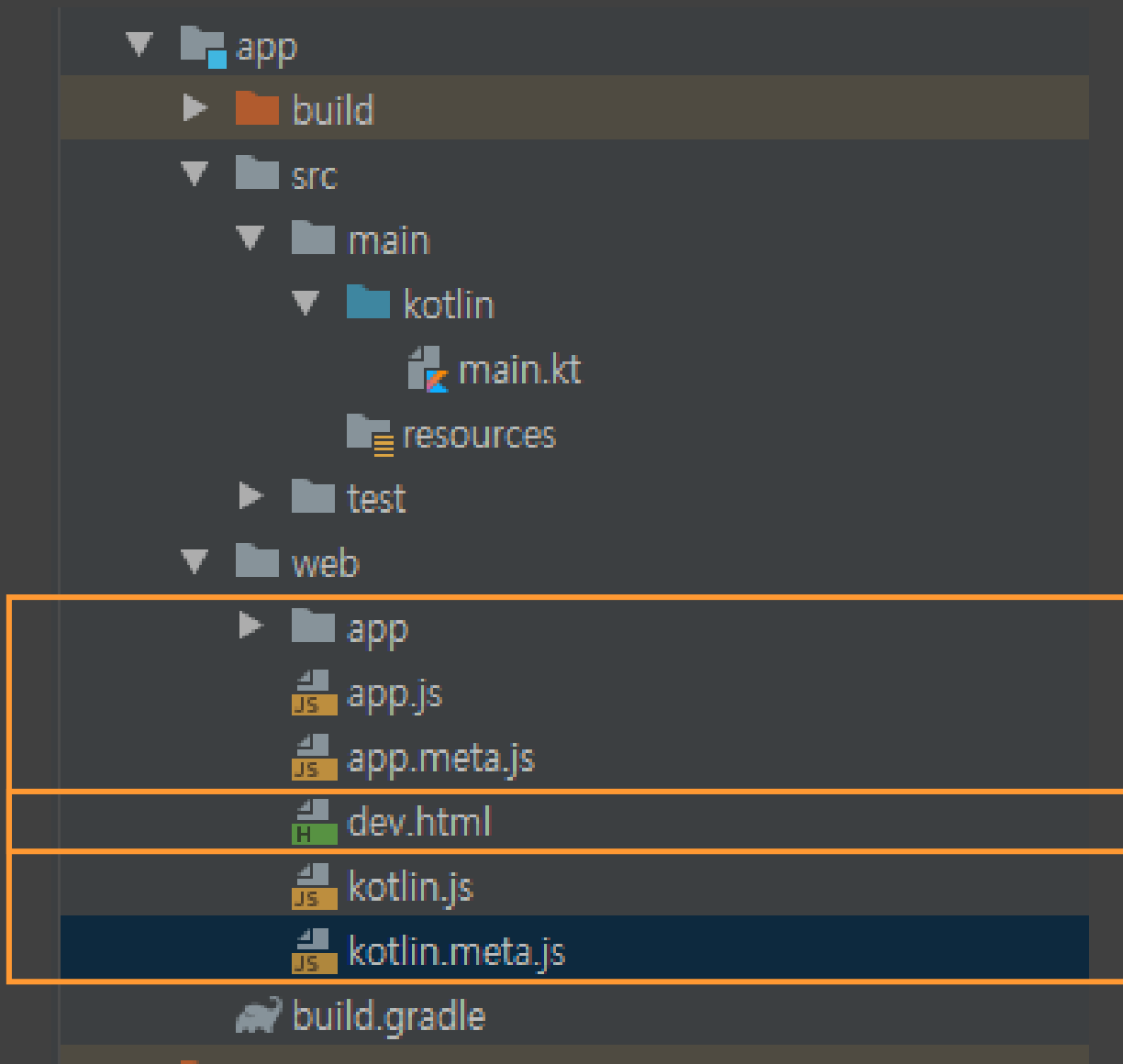
web/dev.html



web/dev.html

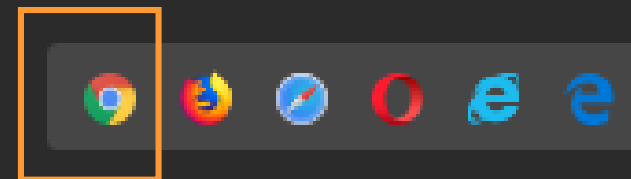
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <script src="kotlin.js"></script>
  <script src="app.js"></script>
</body>
</html>
```

web/dev.html

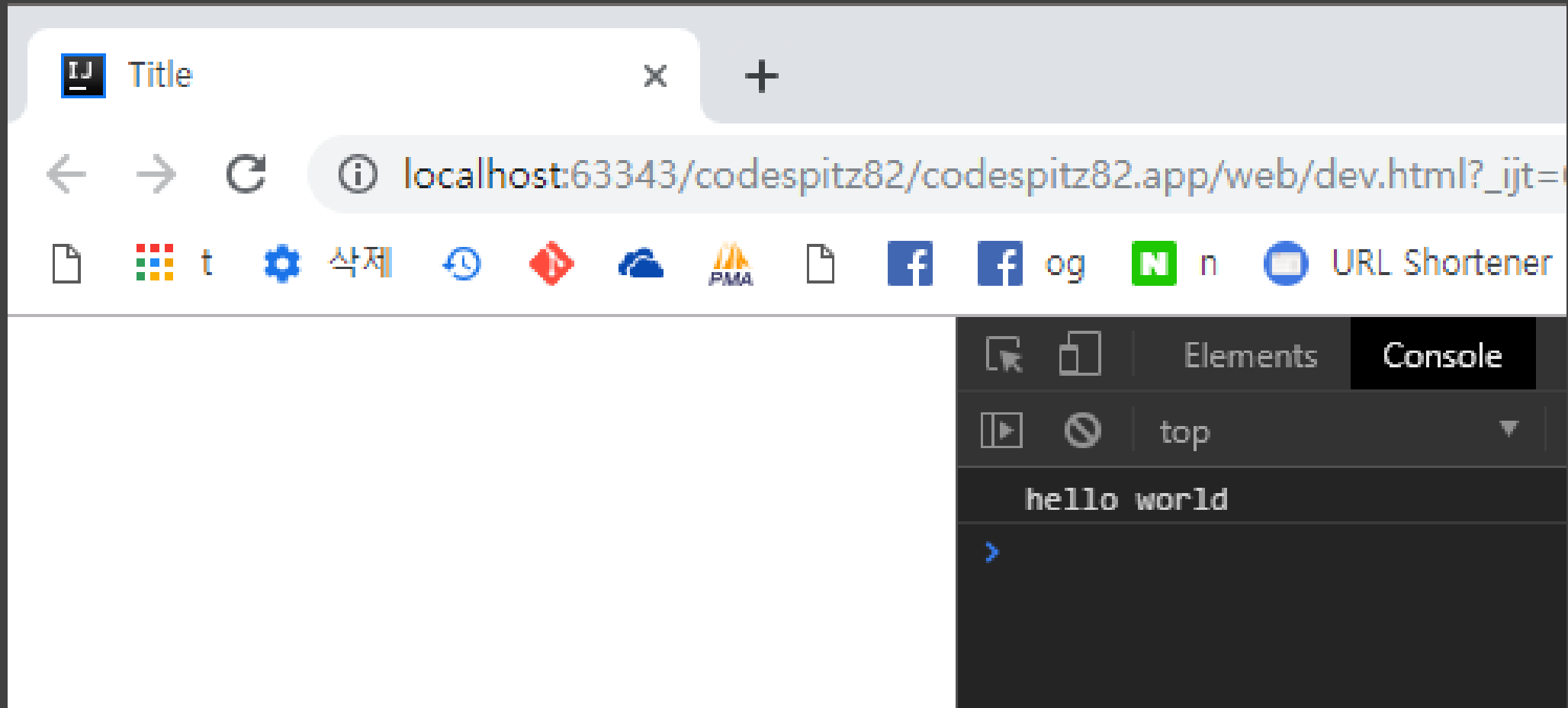


web/dev.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <script src="kotlin.js"></script>
  <script src="app.js"></script>
</body>
</html>
```



web/dev.html



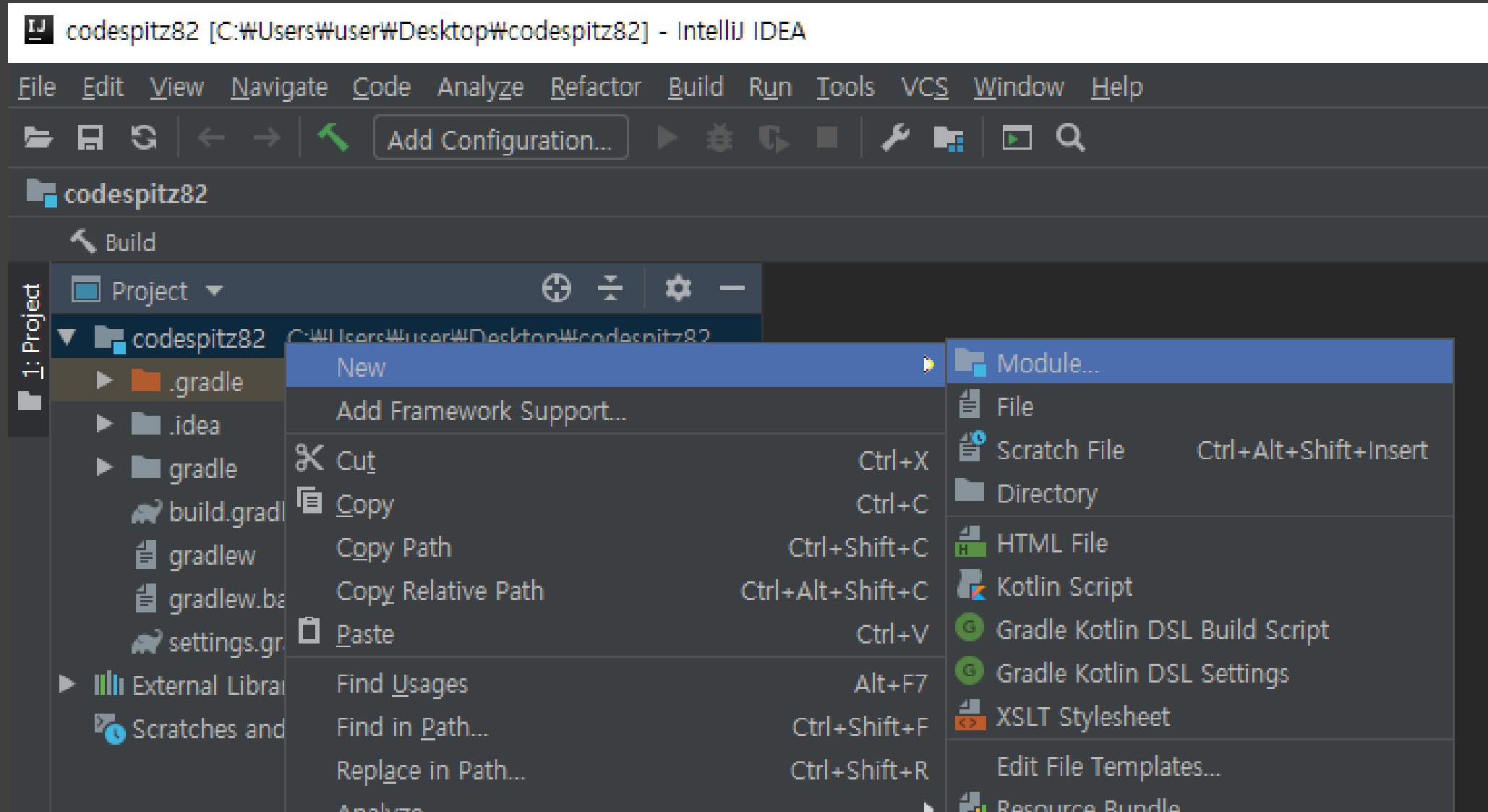
lib module



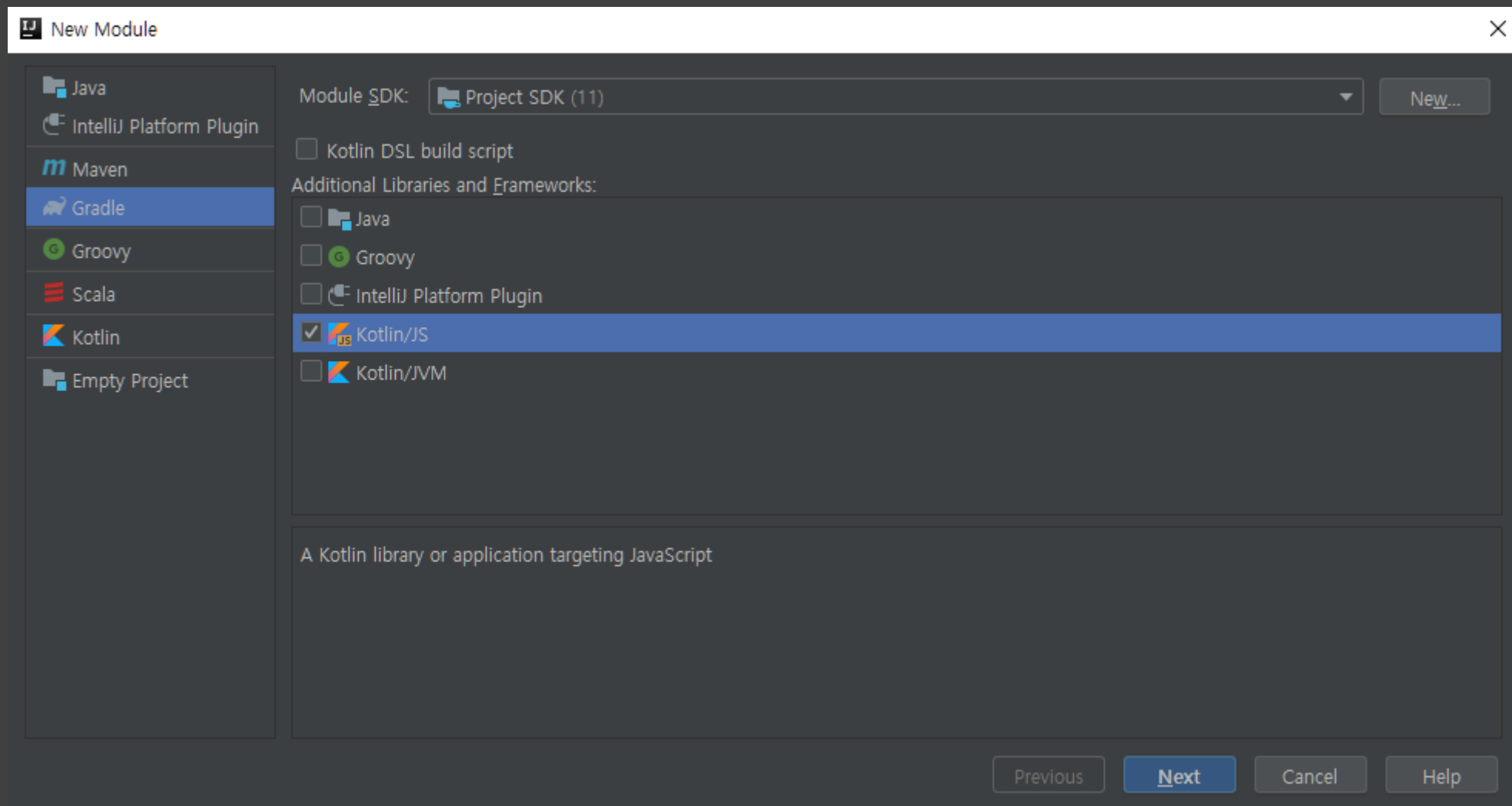
root/build.gradle

```
pluginManagement {  
    resolutionStrategy {  
        eachPlugin {  
            if (requested.id.id == "kotlin2js") {  
                useModule("org.jetbrains.kotlin:kotlin-gradle-plugin:${requested.version}")  
            }  
  
            if (requested.id.id == "kotlin2js") {  
                useModule("org.jetbrains.kotlin:kotlin-gradle-plugin:${requested.version}")  
            }  
        }  
    }  
}  
  
rootProject.name = 'codespitz82'  
include ':app'  
include ':lib'
```

root/build.gradle



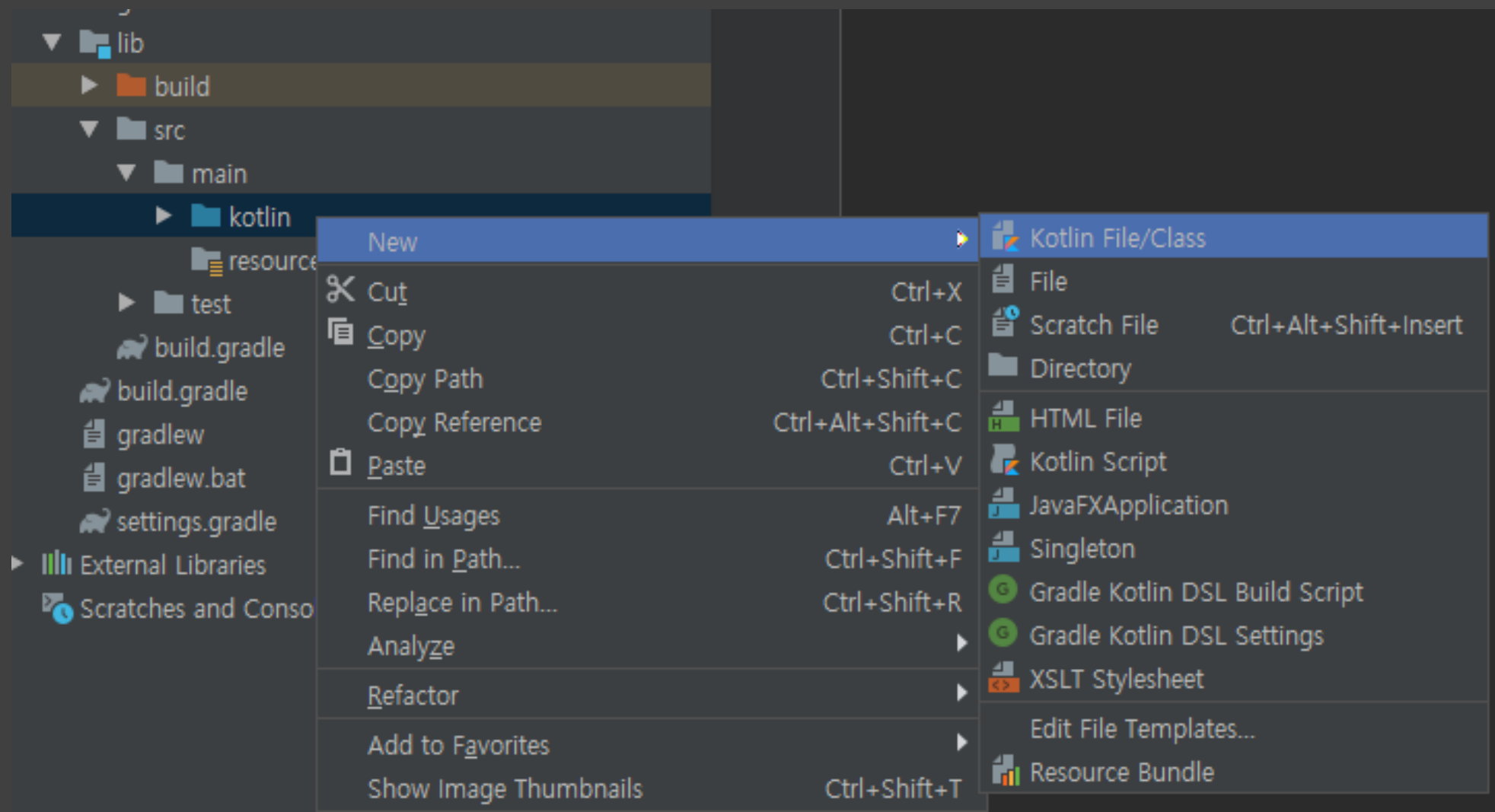
root/build.gradle




app/build.gradle

```
dependencies {  
    compile "org.jetbrains.kotlin:kotlin-stdlib-js:$kotlin_version"  
    compile project(":lib")  
}
```

lib/hello.kt



lib/hello.kt


 New Kotlin File/Class ✕

Name:

hello

↑↓

Kind:

 File ▼

OK

Cancel

lib/hello.kt

```
fun hello() {  
    println("hello world")  
}
```

app/main.kt

```
fun main(){  
    println("hello world")  
}
```



```
fun main(){  
    hello()  
}
```

app/web/app.js

```
if (typeof kotlin === 'undefined') {  
    throw new Error("Error loading module 'app'. Its depend  
}  
if (typeof lib === 'undefined') {  
    throw new Error("Error loading module 'app'. Its depend  
}  
var app = function (_, Kotlin, $module$lib) {  
    'use strict';  
    var hello = $module$lib.hello;  
    function main() {  
        hello();  
    }  
    _.main = main;  
    main();  
    Kotlin.defineModule('app', _);  
    return _;  
}(typeof app === 'undefined' ? {} : app, kotlin, lib);
```

app/web/app.js

```
if (typeof kotlin === 'undefined') {  
    throw new Error("Error loading module 'app'. Its depend  
}  
if (typeof lib === 'undefined') {  
    throw new Error("Error loading module 'app'. Its depend  
}  
var app = function (_, Kotlin, $module$lib) {  
    'use strict';  
    var hello = $module$lib.hello;  
    function main() {  
        hello();  
    }  
    _.main = main;  
    main();  
    Kotlin.defineModule('app', _);  
    return _;  
}(typeof app === 'undefined' ? {} : app, kotlin, lib);
```


app/web/app.js

```
if (typeof kotlin === 'undefined') {  
    throw new Error("Error loading module 'app'. Its depend  
}  
if (typeof lib === 'undefined') {  
    throw new Error("Error loading module 'app'. Its depend  
}  
var app = function (_, Kotlin, $module$lib) {  
    'use strict';  
    var hello = $module$lib.hello;  
    function main() {  
        hello();  
    }  
    _.main = main;  
    main();  
    Kotlin.defineModule('app', _);  
    return _;  
}(typeof app === 'undefined' ? {} : app, kotlin, lib);
```

app/web/app.js

```
if (typeof kotlin === 'undefined') {  
    throw new Error("Error loading module 'app'. Its depend  
}  
if (typeof lib === 'undefined') {  
    throw new Error("Error loading module 'app'. Its depend  
}  
var app = function (_, Kotlin, $module$lib) {  
    'use strict';  
    var hello = $module$lib.hello;  
    function main() {  
        hello();  
    }  
    _.main = main;  
    main();  
    Kotlin.defineModule('app', _);  
    return _;  
}(typeof app === 'undefined' ? {} : app, kotlin, lib);
```

app/web/dev.html









```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <script src="kotlin.js"></script>
  <script src="lib.js"></script>
  <script src="app.js"></script>
</body>
</html>
```

kotlin DCE

Dead Code Elimination



2m!!

이름	수정한 날짜	유형	크기
 app	2019-05-14 오후...	파일 폴더	
 app.js	2019-05-14 오후...	JS 파일	1KB
 app.meta.js	2019-05-14 오후...	JS 파일	1KB
 dev.html	2019-05-14 오후...	Chrome HTML D...	1KB
 kotlin.js	2019-05-14 오후...	JS 파일	1,955KB
 kotlin.meta.js	2019-05-14 오후...	JS 파일	153KB
 lib.js	2019-05-14 오후...	JS 파일	1KB
 lib.meta.js	2019-05-14 오후...	JS 파일	1KB

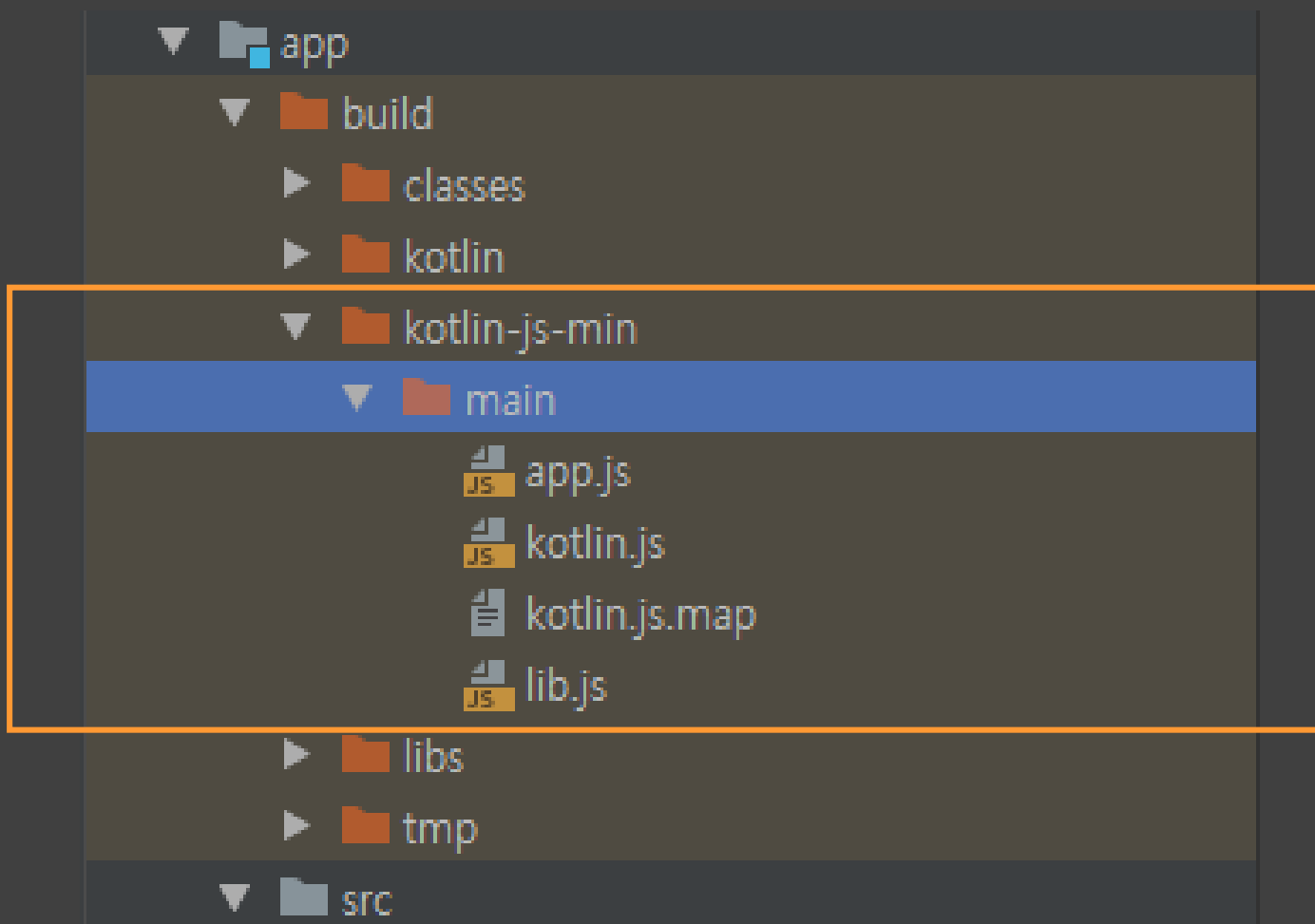
<https://kotlinlang.org/docs/reference/javascript-dce.html>

shorturl.at/fhmqL





app/build.gradle

```
buildscript {  
    ext.kotlin_version = '1.3.31'  
    repositories {  
        mavenCentral()  
    }  
    dependencies {  
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"  
    }  
}  
  
apply plugin: 'kotlin2js'  
apply plugin: 'kotlin-dce-js'  
  
repositories {  
    mavenCentral()  
}
```

build!!



0.1m!!

이름	수정한 날짜	유형	크기
 app.js	2019-05-14 오후...	JS 파일	1KB
 kotlin.js	2019-05-14 오후...	JS 파일	113KB
 kotlin.js.map	2019-05-14 오후...	링커 주소 맵	965KB
 lib.js	2019-05-14 오후...	JS 파일	1KB

kotlin-js-min/dev.html

The screenshot shows an IDE interface with a project structure on the left and a build script on the right. The project structure is as follows:

- Project (C:\Users\User\Desktop\codespitz82)
 - .gradle
 - .idea
 - app
 - build
 - classes
 - kotlin
 - kotlin-js-min
 - main
 - app.js
 - kotlin.js
 - kotlin.js
 - lib.js
 - libs
 - tmp
 - src
 - main

The build script on the right is:

```
1 buildscript {  
2     ext.kotlin_version = '1.3.30'  
3     repositories {  
4         mavenCentral()  
5     }  
6     dependencies {  
7         classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"  
8     }  
}
```

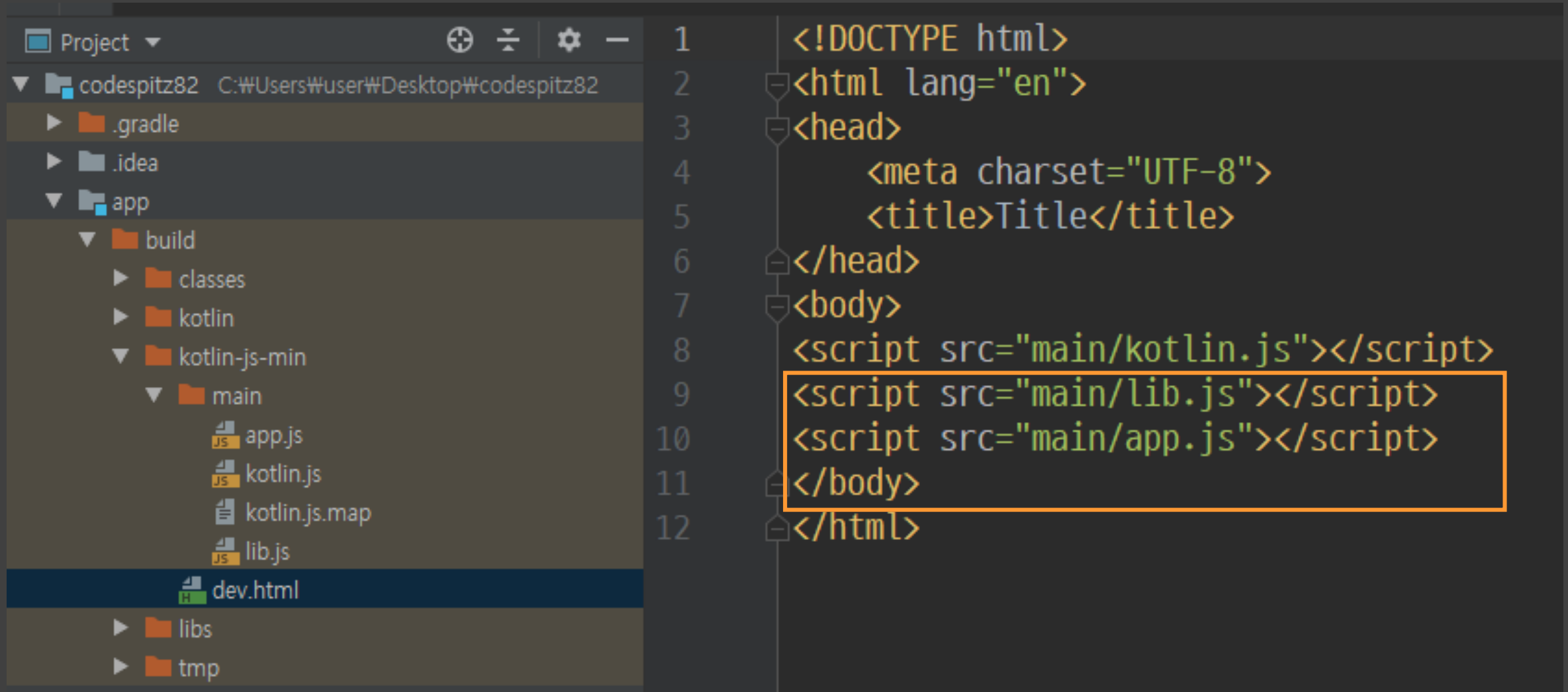
A context menu is open over the 'kotlin-js-min' folder, showing the following options:

- New
- Cut (Ctrl+X)
- Copy (Ctrl+C)
- Copy Path (Ctrl+Shift+C)
- Copy Relative Path (Ctrl+Alt+Shift+C)
- Paste (Ctrl+V)
- Find Usages (Alt+F7)
- Find in Path... (Ctrl+Shift+F)
- Replace in Path... (Ctrl+Shift+R)

The 'New' submenu is open, showing the following options:

- File
- Scratch File (Ctrl+Alt+Shift+Insert)
- Directory
- HTML File
- Kotlin Script
- Gradle Kotlin DSL Build Script
- Gradle Kotlin DSL Settings
- XSLT Stylesheet
- Edit File Templates...

kotlin-js-min/dev.html



The image shows a screenshot of an IDE with a project structure on the left and an HTML file on the right. The project structure is as follows:

- Project
 - codespitz82 C:\Users\User\Desktop\codespitz82
 - .gradle
 - .idea
 - app
 - build
 - classes
 - kotlin
 - kotlin-js-min
 - main
 - app.js
 - kotlin.js
 - kotlin.js.map
 - lib.js
 - dev.html
 - libs
 - tmp

The HTML file, `dev.html`, contains the following code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6 </head>
7 <body>
8     <script src="main/kotlin.js"></script>
9     <script src="main/lib.js"></script>
10    <script src="main/app.js"></script>
11 </body>
12 </html>
```

The code is displayed with line numbers 1 through 12 on the left. The script tags on lines 9, 10, and 11 are highlighted with an orange box.

and Future



type

type
function

type
function
collection

type

function

collection

class & delegate

type

function

collection

class & delegate

kotlinJS & async