



Web development in Kotlin

Kotlin workshop by Schwarz IT

IntelliJ IDEA cheat sheet

<https://github.com/SIT-Kotlin-Workshop>

Function	Windows / Linux	MacOS
Search everywhere	Double Shift	Double Shift
Show documentation	F1	F1
Auto-completion	Control + Space	Command + Space
Intention actions	Alt + Enter	Option + Enter
Comment out/in	Control + /	Command + /

Kotlin cheat sheet

Function	Code examples
Variable declarations	<pre>var mutableString: String = "Mutable" val immutableString: String = "Immutable" val inferredString = "Type inferred" var nullableString : String? = "Nullable" // Can be null</pre>
Nullability	<pre>val safeNavigation = object?.property val elvis = nullableValue ?: "Alternative"</pre>
Lists	<pre>val immutableList = listOf("Immutable", "list") val mutableList = mutableListOf("Mutable", "list") val firstEntry = list[0] / list.first() / list.firstOrNull()</pre>
Data classes	<pre>data class MyClass(val primary: String = "Hello",) { val secondary: String? = "World" }</pre>
Functions	<pre>fun functionExpression(name : String) = "Hello, \$name" fun functionBlock(name : String) : String { return "Hello, \${name.uppercase()}" } fun <T, V> higherOrder(x : T, f : (T) -> V) : V = f(x)</pre>

Immutability & basic syntax

Exercise: Immutability

- Open the file `ExerciseImmutableValues.kt` and complete the tasks therein.
- Open the file `ExerciseImmutableLists.kt` and complete the tasks therein.

Hint: In IntelliJ, press Shift twice and type the name of the file to quickly jump to it.

Exercise: Basic syntax

Open the file `ExerciseGradePrinter.kt` and complete the tasks therein.

You can play around with various ways of completing the task.

Exercise: FizzBuzz

Open the file `ExerciseFizzBuzz.kt` and complete the tasks therein.

Advanced exercise: Fractions

Open the file `ExerciseFractions.kt` and complete the tasks therein.

Testing

Exercise: Testable FizzBuzz

- Why is your implementation of FizzBuzz in file `ExerciseFizzBuzz.kt` hard to test?
- In file `ExerciseTestableFizzBuzz.kt`, implement the function.
- In file `ExerciseTestableFizzBuzzTest.kt`, write a few tests for this function.

Hint: It is common to separate the production code (`/main/...`) from testing code (`/test/...`). Use Control-Leftclick (Windows) or Command-Leftclick (MacOS) on function names to quickly navigate through the codebase.

Exercise: Test-driven development

Test driven development (TDD) is the practice of implementing the tests *before* writing the actual production code. This way, the test acts as a sort of specification for the code.

`ExerciseTddTest.kt` contains tests for a `mysteriousFunction`.
In `ExerciseTdd.kt`, implement that function so that all tests are successful.

Advanced exercise: Testing a complicated function

`ExerciseTestMe.kt` contains a function that does ... something.

Try to find out what the function does by writing tests for in `ExerciseTestMeTest.kt`.

Advanced exercise: Advanced TDD

Time for another round of test-driven development!
`ExerciseTddAdvancedTest.kt` contains the tests, implement the function in `ExerciseTddAdvanced.kt`.

Immutable data class and nullability

Exercise: Data classes

Open the file `ExerciseDataClasses.kt` and complete the tasks therein.

Exercise: Nullability

Open the file `ExerciseNullability.kt` and complete the tasks therein.

Higher order functions

Exercise: Using higher-order functions

Open the file `ExerciseUsingHOF.kt` and complete the tasks therein.

Exercise: Writing higher-order functions

Open the file `ExerciseWritingHOF.kt` and complete the tasks therein.

Advanced exercise: Higher-order functions everywhere!

Go through the code for the earlier exercises and identify places where higher-order functions can be used

Web development

Exercise: Web-development basics

- Open the file `ExerciseRoutes.kt` and complete the tasks therein.
- Open the file `ExerciseRoutesTest.kt` and complete the tasks therein.

Advanced exercise: Kodee's little online shop!

The file `AdvancedRoutes.kt` contains endpoints implementing a REST API that is used by a rudimentary frontend available at <http://localhost:8080/frontend/advanced.html>.

- Take a look at the existing backend codebase.
- Oh no! The “Delete” button next to each order in the frontend doesn’t work!
Complete the first task in `AdvancedRoutes.kt`, implementing a DELETE endpoint for `"/advanced/order/{identifier}"`.
Respond with an appropriate error when the order does not exist.
(Can this error actually be caused by using the frontend?)
- The “Reschedule” functionality doesn’t seem to work either.
Complete the second task in `AdvancedRoutes.kt`, implementing a POST endpoint for `"/advanced/order/{identifier}"`.
- To `AdvancedRoutesTest.kt`, add tests for your endpoints!
- Once you have the basic functionality working, you can implement some improvements, for example:
 - Orders can only be rescheduled if they are not in delivery yet.
 - Kodee only delivers on Monday to Tuesday 9:00 to 17:00 or Friday and Saturday 9:00 to 12:00