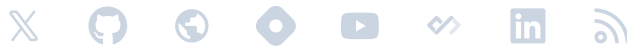


Ronnie Atuhaire's Blog 🤖

 Follow



KOTLIN FOR PYTHON DEVELOPERS



Syntax, Comments & Variables



Syntax, Comments & Variables



Ronnie Atuhaire

Aug 30, 2021 •  5 min read



PLAY THIS ARTICLE

0:00 / 4:17



16



Hello there 🙌, welcome back to my blog. Today, we shall learn how to write proper Kotlin code as we compare it with Python. We shall see how to make unreadable code with comments and how to declare variables.

Read my [previous article](#) about installation since this is just a continuation in the Kotlin for Python Developers Series .

🌟 Syntax 📋

Every language has its rules, syntax, grammar and the way how stuff is implemented.

Kotlin Syntax is quite different from python, and by syntax, I mean *the structure of statements in writing Kotlin code*. Since Kotlin is statically typed, let's refer to our previous example in our [previous tutorial](#).

COPY

```
fun main() {  
    println("Hello, World!")  
}
```

In Kotlin versions earlier than 1.3, the main function must have a parameter of type Array like the code below. But am assuming that you are using the latest Kotlin version in this tutorial.

COPY

```
fun main(args: Array<String>) {  
    println("Hello, World!")  
}
```



The same program in Python:

```
def fun():  
    print("Hello World!")  
  
fun() # calling the function
```

As you can see above, the ***main()*** function is something you will see in every Kotlin program, it's an entry point to the program and therefore we do not need to call the function like in the python example.

This ***main()*** keyword function is used to execute code. Any code inside the ***main()*** function's curly brackets { } will be executed. For example, the ***println()*** function is inside the ***main()*** function, meaning that this will be executed.

In Python, we use *indentation* instead of curly brackets to define a function scope!

Spacing

When writing Python code only the space at the beginning of a logical line matters and in Kotlin the space between the curly brackets and ***println()*** function does not matter and that same function can be achieved as:

```
fun main(){println("Hello, World!")}
```

Test this! So the spacing is for readability and it's conventional.

In Kotlin coding **conventions** it's recommended to use a single indent in cases where the l



en forced before.

Comments

In computer programming, a comment is a programmer-readable explanation or annotation in the source code of a computer program. They are added with the purpose of making the source code easier for humans to understand, and are generally ignored by compilers and interpreters.

Comments help explain what the real code is about.

Unlike Python where comments are initiated with '#' for inline and '"""' for multiline. In Kotlin we use // for inline and the block comments start with /* and end with */:

Kotlin Inline

COPY

```
// This is a comment!  
fun main() {  
    println("Hello, World!") // This is also a comment  
}
```

Python Inline

COPY

```
def fun():  
    print("Hello World!") # First comment  
  
fun() # Second comment
```

Kotlin Multiline



COPY

```
/* No-output
fun main() {
    println("Hello, World!")
}
*/
```

Python Multiline

COPY

```
"""
def fun():
    print("Hello World!")

fun()
"""
```

🌟 Variables 📁

In computer programming, a variable is a storage location paired with an associated symbolic name (an identifier) that contains a value. In other words, a variable is a storage location for data.

Python has no command for declaring a variable. A variable is created the moment you first assign a value to it but Kotlin has two different ways by using keywords *var* and *val*.

COPY

```
new_variable = "Python Variable"
```



You cannot declare a variable or value as constant in Python. Just don't change it!

var VS val

var is like a general variable and can be assigned multiple times and is known as the mutable variable in Kotlin. Whereas *val* is a constant variable and can not be assigned multiple times and can be initialized only a single time and is known as the immutable variable in Kotlin.

COPY

```
var new_variable_1 = "First Kotlin Mutable variable"
val new_variable_2 = "Second Kotlin Immutable variable"
```

Type Inference

In the above code, just like Python, Kotlin is smart enough to understand that the declared variables are of type strings. However, you can also specify the type like this:

COPY

```
var newVariable: String = "First Kotlin Mutable variable"
var myNumber: Int = 80
```

I changed the naming above to lowerCamelCase since it is preferred for writing variables in Kotlin. You can't write variables starting with numbers, punctuation marks.



COPY

```
var 2hashnode = "
```





COPY

```
var hasnode_2 = "Okay this is okay"
```



COPY

```
var 320 = "threetwenty"
```

✓ but not recommended

COPY

```
var _320 = "threetwenty"
```

✓ but for specific use cases (Receivers)

COPY

```
var $number = 89
```

Un-Assigned Variables

You may just want to name a variable that you might later use in time. In Python, because of its dynamic nature you don't need to declare things but you can simply assign it to *None*

COPY

```
my_var = None
```



However, Kotlin does not do that, because of its nullability mechanism. You must explicitly declare a variable nullable to allow null:

COPY

```
var myVar: String? = null # The ? allows a null type assignment
```

The other Kotlin option is to mark it *lateinit*.

COPY

```
lateinit var myVar: String // lateinit -> Late Initialisation
```

That's it for this article! Check out the official docs for more reading
>>>>>>

If you enjoyed this article, consider subscribing & following me for the next one in this series. We shall look at data types.

Ronnie Atuhaire 🕶️

Follow me on [Twitter](#)

Subscribe to my newsletter

Read articles from **Ronnie Atuhaire's Blog** 🧐 directly inside your inbox. Subscribe to the newsletter, and don't miss out.



Enter your email address

SUBSCRIBE

Kotlin

kotlin beginner

Python

Tutorial

Written by



Ronnie Atuhaire

My name is Ronnie & a fellow geek like you 🧐. I am passionately tech curious.

I blog, tweet, write, code, vlog, discuss and eat anything about tech.

🥳 Feel Free To Connect 🥳

Follow

ARTICLE SERIES

Kotlin for Python Developers

1

Introduction to Kotlin

📌 Learning a new language is quite easy especially if you already know another one, all you got to ...



2

Installation & Hello World!

Hello there 🙌, welcome back! We shall continue from our last post (Introduction to Kotlin.) I want t...

[Show all 12 posts](#)

15

Kotlin Enum Classes

Hey, welcome again to this series about Kotlin Enum Classes and if you have ...



16

Kotlin Data Classes

Hey, welcome again to my blog! Today, we shall learn more about Kotlin Data Classes and if you have ...



©2023 Ronnie Atuhaire's Blog 🤓

[Archive](#) · [Privacy_policy](#) · [Terms](#)

 **Publish with Hashnode**

Powered by [Hashnode](#) - Home for tech writers and readers

