

Learn **Python** in Ten Minutes

The tutorial: The book



Stavros Korokithakis

Learn Python in Ten Minutes

The tutorial: The book

Stavros Korokithakis

This book is for sale at <http://leanpub.com/learn-python>

This version was published on 2018-05-02



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2011 - 2018 Stavros Korokithakis

Tweet This Book!

Please help Stavros Korokithakis by spreading the word about this book on [Twitter](#)!

The suggested hashtag for this book is [#learn-python](#).

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

[#learn-python](#)

Contents

1.	Learn Python in Ten Minutes	1
1.1	Preliminary fluff	1
1.2	Properties	1
1.3	Getting help	1
1.4	Syntax	2

1. Learn Python in Ten Minutes

1.1 Preliminary fluff

So, you want to learn the Python programming language but can't find a concise and yet full-featured tutorial. This tutorial will attempt to teach you Python in 10 minutes. It's probably not so much a tutorial as it is a cross between a tutorial and a cheatsheet, so it will just show you some basic concepts to start you off. Obviously, if you want to really learn a language you need to program in it for a while. I will assume that you are already familiar with programming and will, therefore, skip most of the non-language-specific stuff. Also, pay attention because, due to the terseness of this tutorial, some things will be introduced directly in code and only briefly commented on.

1.2 Properties

Python is strongly typed (i.e. types are enforced), dynamically, implicitly typed (i.e. you don't have to declare variables), case sensitive (i.e. var and VAR are two different variables) and object-oriented (i.e. everything is an object).

1.3 Getting help

Help in Python is always available right in the interpreter. If you want to know how an object works, all you have to do is call `help(<object>)`! Also useful are `dir()`, which shows you all the object's methods, and `<object>.__doc__`, which shows you its documentation string:

```
1 >>> help(5)
2 Help on int object:
3 (etc etc)
4
5 >>> dir(5)
6 ['__abs__', '__add__', ...]
7
8 >>> abs.__doc__
9 'abs(number) -> number'
10
11 Return the absolute value of the argument.'
```

1.4 Syntax

Python has no mandatory statement termination characters and blocks are specified by indentation. Indent to begin a block, dedent to end one. Statements that expect an indentation level end in a colon (:). Comments start with the pound (#) sign and are single-line, multi-line strings are used for multi-line comments. Values are assigned (in fact, objects are bound to names) with the `_equals_` sign (`=`), and equality testing is done using two `_equals_` signs (`==`). You can increment/decrement values using the `+=` and `-=` operators respectively by the right-hand amount. This works on many datatypes, strings included. You can also use multiple variables on one line. For example:

```
1  myvar = 3
2  >>> myvar += 2
3  >>> myvar
4  5
5  >>> myvar -= 1
6  >>> myvar
7  4
8  """This is a multiline comment.
9  The following lines concatenate the two strings."""
10 >>> mystring = "Hello"
11 >>> mystring += " world."
12 >>> print mystring
13 Hello world.
14 # This swaps the variables in one line(!).
15 # It doesn't violate strong typing because values aren't
16 # actually being assigned, but new objects are bound to
17 # the old names.
18 >>> myvar, mystring = mystring, myvar
```