



Python 3 Tutorial

- The Origins of Python
- Starting with Python: The Interactive Shell
- Executing a Script
- Indentation
- Data Types and Variables
- Operators
- Sequential Data Types: Lists and Strings
- List Manipulations
- Shallow and Deep Copy
- Dictionaries
- Sets and Frozen Sets
- An Extensive Example Using Sets
- input via the keyboard
- Conditional Statements
- Loops, while Loop
- For Loops
- Difference between iterators und Iterables
- Output with Print
- Formatted output with string modulo and the format method
- Functions
- Recursion and Recursive Functions
- Parameter Passing in Functions
- Namespaces
- Global and Local Variables
- Decorators
- Memoization with Decorators
- Read and Write Files
- Modular Programming and Modules
- Packages in Python
- Regular Expressions
- Regular Expressions, Advanced
- Lambda Operator, Filter, Reduce and Map
- List Comprehension
- Iterators and Generators
- Exception Handling
- Tests, DocTests, UnitTests
- Object Oriented Programming
- Class and Instance Attributes
- Properties vs. getters and setters
- Inheritance
- Multiple Inheritance
- Magic Methods and Operator Overloading

Next Chapter: Starting with Python: The Interactive Shell

History of Python

Easy as ABC

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde & Informatica). The greatest achievement of ABC was to influence the design of Python.

Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners<sup>1</sup>, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."

Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."



Comedy, Snake or Programming Language

So, what about the name "Python": Most people think about snakes, and even the logo depicts two snakes, but the origin of the name has its root in British humour. Guido van Rossum, the creator of Python, wrote in 1996 about the origin of the name of his programming language<sup>1</sup>: "Over six years ago, in December 1989, I was looking for a 'hobby' programming project that would keep me occupied during the week around Christmas. My office ... would be closed, but I had a home computer, and not much else on my hands. I decided to write an interpreter for the new scripting language I had been thinking about lately: a descendant of ABC that would appeal to Unix/C hackers. I chose Python as a working title for the project, being in a slightly irreverent mood (and a big fan of Monty Python's Flying Circus)."

The Zen of Python

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Flat is better than nested.
- Sparse is better than dense.
- Readability counts.
- Special cases aren't special enough to break the rules.
- Although practicality beats purity.
- Errors should never pass silently.
- Unless explicitly silenced.
- In the face of ambiguity, refuse the temptation to guess.
- There should be one -- and preferably only one -- obvious way to do it.
- Although that way may not be obvious at first unless you're Dutch.
- Now is better than never.
- Although never is often better than \*right\* now.
- If the implementation is hard to explain, it's a bad idea.
- If the implementation is easy to explain, it may be a good idea.
- Namespaces are one honking great idea -- let's do more of those!

Development Steps of Python

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system.

Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.

Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode.

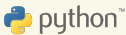
Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."

Some changes in Python 3.0:

- Print is now a function
- Views and iterators instead of lists
- The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
- There is only one integer type left, i.e. int. long is int as well.
- The division of two integers returns a float instead of an integer. "/" can be used to have the "old" behaviour.
- Text Vs. Data Instead Of Unicode Vs. 8-bit

<sup>1</sup> January 13, 2003, <http://www.artima.com/intv/pythonP.html> <sup>2</sup> Foreword for "Programming Python" (1st ed.) by Mark Lutz, O'Reilly

Next Chapter: Starting with Python: The Interactive Shell



This website is free of annoying ads. We want to keep it like this. You can help with your donation:

Donate

The need for donations

Bernd Klein on Facebook

Search this website:

- OOP, Inheritance Example
- Slots
- Classes and Class Creation
- Road to Metaclasses
- Metaclasses
- Metaclass Use Case: Count Function Calls
- Abstract Classes

Learning From History

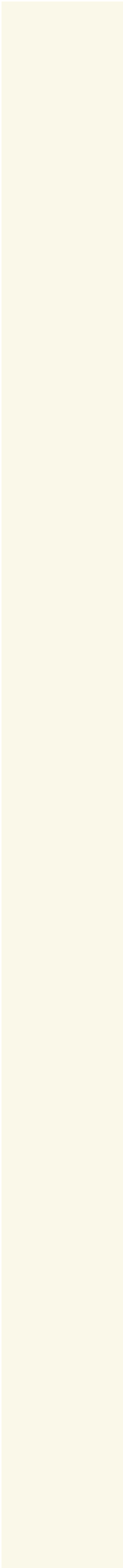


"Hegel was right when he said that we learn from history that man can never learn anything from history."  
(George Bernard Shaw)

This website is supported by:  
**bodenseo**  
Python Training Courses

Freedom in Programming

*"Python is an experiment in how much freedom programmers need. Too much freedom and nobody can read another's code; too little and expressiveness is endangered."*  
(Guido van Rossum, 13 Aug 1996)











































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































This topic in German  
/ Deutsche  
Übersetzung:  
[Geschichte und  
Philosophie von  
Python](#)

#### Python 3

This is a tutorial in  
Python3, but this  
chapter of our course  
is available in a  
version for Python  
2.x as well: [History  
and Philosophy of  
Python in Python 2.x](#)

#### Book a Dedicated Course

The goal of this  
website is to provide  
educational material,  
allowing you to learn  
Python on your own.  
Nevertheless, it is  
faster and more  
efficient to attend a  
"real" Python course  
in a classroom, with  
an experienced

trainer. So why not attend one of the live [Python courses](#)



in Paris, London, Berlin, Munich or Lake Constance by Bernd Klein, the author of this tutorial?

#### Onsite Training Courses

Let us come to your company or institute and train your employees, as we've done it many times in Amsterdam (The Netherlands), Berlin (Germany), Bern (Switzerland), Basel (Switzerland), Zurich (Switzerland), Locarno (Switzerland), Den Haag (The Hague), Hamburg (Germany), Frankfurt (Germany), Toronto (Canada), Edmonton (Canada), Munich (Germany) and many other cities. We do training courses in England, Switzerland, Liechtenstein, Austria, Germany, France, Belgium, the Netherlands, Luxembourg, Poland, UK, Italy and other locations in Europe and in Canada.

This way you will get a perfect training up to your needs and it will be extremely cost efficient as well. [Contact us](#) so we can define and find the best course curriculum to meet your needs, and schedule course sessions to be held at your location.

#### Skilled Python Programmers

You can hire skilled Python programmers or even a skilled team of Python developers to work exclusively on your project. [Contact us](#), if you want more information.

#### Quote of the Day:

*"If you use the original World Wide Web program, you never see a URL or have to deal with HTML. That was a surprise to me that people were prepared to painstakingly write HTML."* (Tim Berners Lee)

#### Data Protection Declaration

[Data Protection Declaration](#)