



CODE BASED
LEARNING

PYTHON-DOCS

- BACKGROUND -

PROF. DR. RER. NAT.
ALEXANDER Voß

INFORM-PROFESSUR
FH AACHEN



CODE BASED
LEARNING

BDFL
FACTS
ZEN OF PYTHON
SOURCES

BACKGROUND

- A FEW WORDS ON PYTHON -



Benevolent Dictator for Life

The one who first defined the language, i.e. invented it, is Guido van Rossum, also known as **Benevolent Dictator for Life (BDFL)**. You should have heard it once.

According to van Rossum Python

- is a simple, intuitive language, not inferior to competitors in power;
- is open source, so anyone can help with development;
- has source code that is as easy to read as plain English;
- is suitable for everyday tasks and allow short development times.

Whether and to what extent this is true, you can decide for yourself at the end of the module 😊

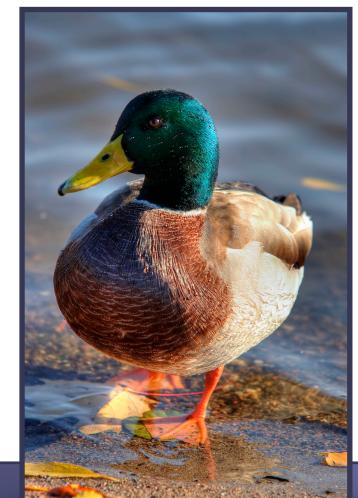


Guido van Rossum 2006, Wikipedia



Some Facts on Python

- Name inspired by the British comedy group Monty Python (spam and eggs)
- First released in 1991 as Python 0.9.0, as 2.0 in 2000 and 3.0 in 2008
- High-level, general-purpose programming language
- Supports multiple programming paradigms, including object-oriented, aspect-oriented, meta- and functional programming
- Duck typed, typed objects but untyped variable names
- Garbage-collected
- Batteries included
- Whitespace indentation rather than curly brackets
- Further developed by Python Enhancement Proposals ([PEPs](#))



If it walks like a duck and it quacks like a duck, then it must be a duck.

!





The Zen of Python (PEP 20)

Background

- 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.

by Tim Peters in 2004, [Zen of Python](#)

Some ideas can be seen directly in the language.

- 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!



Sources to start with

Python

- <https://www.python.org>
- <https://docs.python.org> and <https://peps.python.org>
- <https://pypi.org>
- <https://www.w3schools.com/python>
- <https://realpython.com>
- <https://www.tutorialspoint.com/python>



Ecosystem

- <https://github.com/pyenv/pyenv> and <https://github.com/pyenv-win/pyenv-win>
- <https://stackoverflow.com/questions/41573587/what-is-the-difference-between-venv-pyvenv-pyenv-virtualenv-virtualenvwrappe>
- <https://www.jetbrains.com/help/pycharm>





CODE BASED LEARNING

If you are missing something or have suggestions for improvement, please send me an [E-Mail](#).

!

