

The Python programming language



1. Origins
2. Who uses Python?
3. Characteristics
4. Syntax
5. Comparison to other languages
6. Code examples

Origins of Python

- Interpreted script language
- Developed in the early 1990s by Guido van Rossum
- Named after British comedy group “Monty Python”
- Current version number: 2.5.2
- Implementations: *CPython*, *Jython*, *IronPython*, *PyPy*

Who uses Python?

- Zope Application Server
- Google
- YouTube
- BitTorrent (prior to version 6.0)
- NASA
- Industrial Light & Magic

Characteristics

- Syntax inspired by C, Lisp, Perl and Java
- Python code is designed to be easy to read (“pseudo code that works”)
- Multi-paradigm
- Very large standard library
- Can be easily extended using C / C++ modules

Syntax

- Uses only the most important keywords and language constructs
- Uses English words instead of punctuation
- No semicolons are needed at the end of an instruction
- No curly braces used to delimit an instruction block (indentations are used)

Syntax (II)

- Default types (bool, int, float, strings...)
- Supports arrays and dictionaries
- Every data type is an object
- You can create your own data types using classes
- Type of a variable set by value, not by name

- *if ... elif ... else:*
A conditional block. Python doesn't use *switch-case* or things like *goto*, you have to implement these things on your own
- *def:*
Defines a function or a class method
- *for:*
Iterates over an array
- *while:*
Creates a loop which is executed until a certain condition is fulfilled
- *class:*
Defines a class

Comparison to other languages

- **C / Java:**
 - Python programs are executed slower
 - Python programs can be programmed faster and are also shorter
 - C / Java more suitable for low-level programs
- **Perl:**
 - Similar background (Unix scripting)
 - Can both be extended using C / C++
 - Python is more readable
- **PHP:**
 - Can both be extended using C / C++
 - Python is more object-oriented
 - Python core is much smaller, better naming conventions

Code example

```
# This is a comment
```

```
def add(a, b):  
    a + b = c  
    return c
```

```
number1 = 2
```

```
number2 = 3
```

```
print add(number, number2)
```

Code example (II)

```
mark = 3
```

```
if mark < 5:
```

```
    print "You've passed the exam"
```

```
else:
```

```
    print "You have not passed the exam"
```

Code example (III)

```
# Print numbers from 0 to 20

x = 0

while x < 21:
    print x
```



Ende

Thanks for listening!