Python Training: Beginners

by Florian Dahlitz

16.07.19

About Me

- name: Florian Dahlitz

- corporate student@IBM

- applied computer-science

- author@Real Python

First of All

- 2 hours session

- ask questions at any time

- feedback welcomed

- resources on GitHub

Outline

1. Installation

2. What is Python?

3. REPL Session

4. IDLE Session

5. Wrap-Up

1. Installation

- generally: python.org

- Windows 10: Windows Store

2. What is Python?

Excursion: Programming Languages

Compiled Languages

- compiled into machine-readable language

- faster on linear execution

- overall better performance

Interpreted Languages

- directly executed

- faster on branched execution

2. What is Python?

Python consists of …

… the language (interpreter)
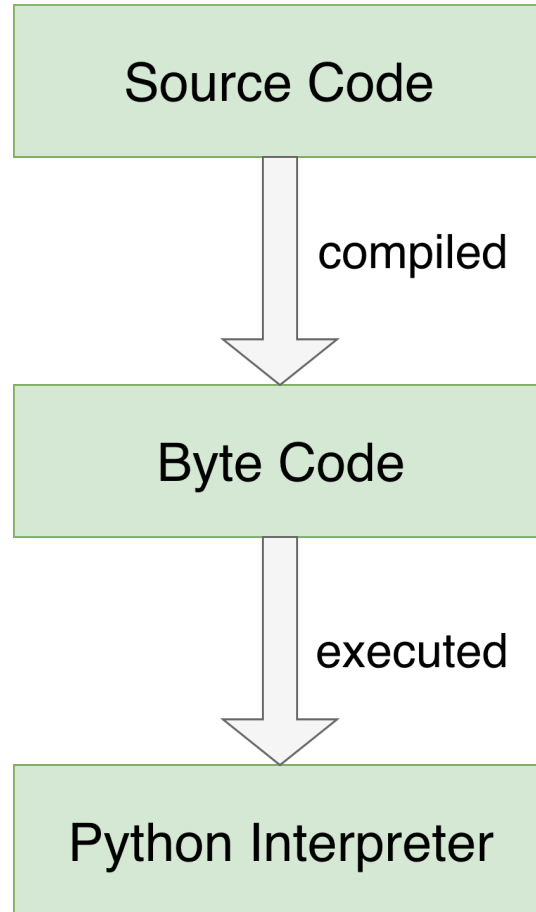

… the compiler

… the standard library

Language

- interpreted

- object-oriented

- high-level

- dynamic semantics

Compiler

```
┌─────────────────────┐
│     Source Code     │
└─────────────────────┘
          │
          │ compiled
          ▼
┌─────────────────────┐
│     Byte Code       │
└─────────────────────┘
          │
          │ executed
          ▼
┌─────────────────────┐
│  Python Interpreter │
└─────────────────────┘
```

Standard Library

- very extensive

- *Swiss Army Knife*

    - *email* module→ everything for managing emails

    - *json* module → everything for managing json

3. REPL Session

REPL

- <u>R</u>ead-<u>E</u>val-<u>P</u>rint <u>L</u>oop

- immediate feedback/results

- enter the REPL using "python" command

Data Types and Structures

- numbers (int, float, complex)

- strings (str)

- booleans (bool)

Data Types and Structures

- lists (list)

- tuples (tuple)

- sets (set)

- dictionaries (dict)

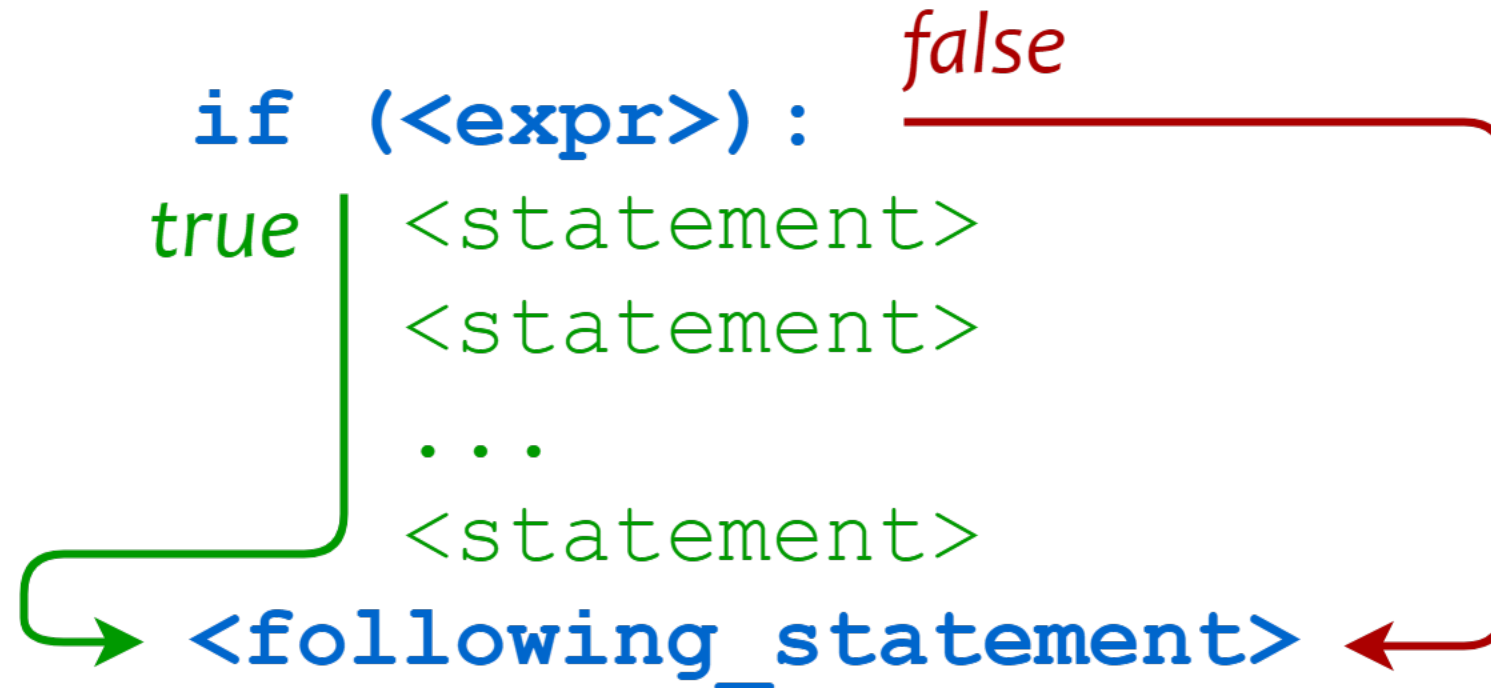# Variables

- without type specification

- overwritable type

```
>>> x = 5
>>> x
5
>>> type(x)
<class 'int'>
>>> x = "Hello World"
>>> x
'Hello World'
>>> type(x)
<class 'str'>
```

Operators

- Arithmetic operators

- Assignment operators

- Comparison operators

- Logical operators

- Identity operators

- Membership operators

- (Bitwise operators)
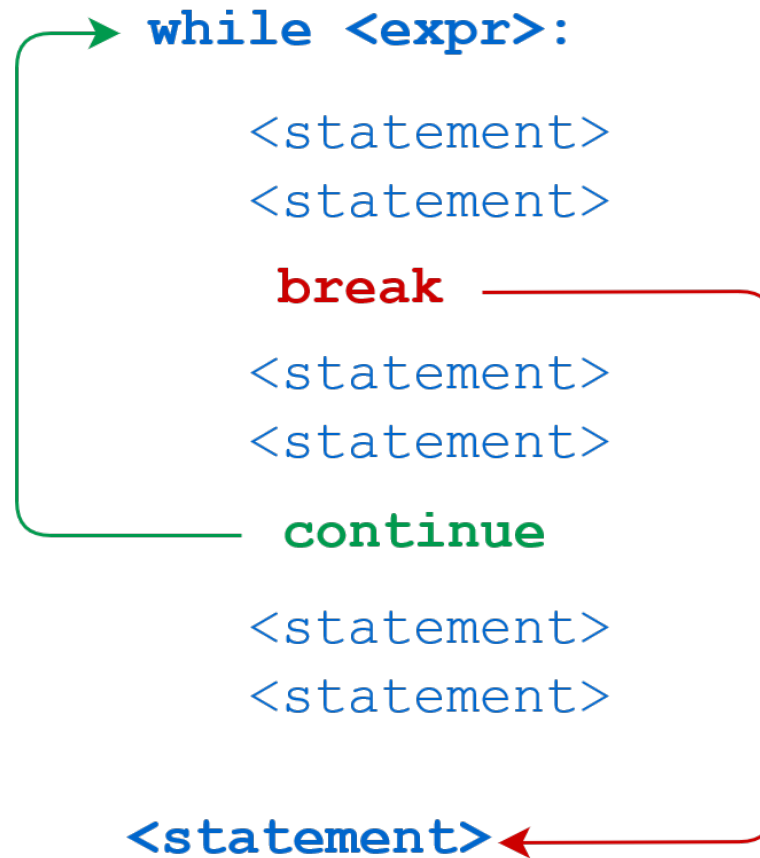
Conditional Statements

Source: https://realpython.com/python-conditional-statements/

Conditional Expression

```
<expr1> if <conditional_expr> else <expr2>
```

Loops

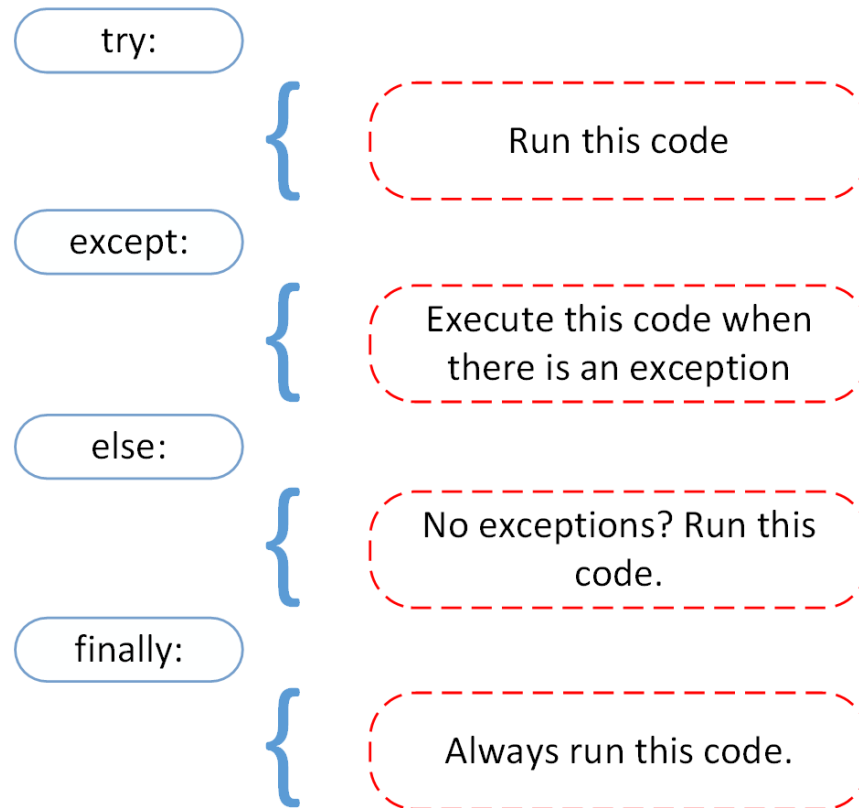- definite iteration (for-loop)

- indefinite iteration (while-loop)

Loops: continue and break

```
while <expr>:

        <statement>
        <statement>
      break

        <statement>
        <statement>
      continue

        <statement>
        <statement>

<statement>
```

Source: https://realpython.com/python-while-loop/

Exception Handling

try:

{ Run this code

except:

{ Execute this code when there is an exception

else:

{ No exceptions? Run this code.

finally:

{ Always run this code.

Source: https://realpython.com/python-exceptions/

Functions

Object-Oriented Programming

4. IDLE Session

5. Wrap-Up

5. Wrap-Up

- What is Python?

- data types and structures

- variables

- operators

5. Wrap-Up

- conditional statements (expressions)

- for-loop and while-loop

- exception handling

- functions and object-oriented programming

Any questions left?