Technische Hochschule Rosenheim

Modul - Introduction to AI (AI1)

Bachelor Programme AAI

02 - Python (PART I)

Prof. Dr. Marcel Tilly

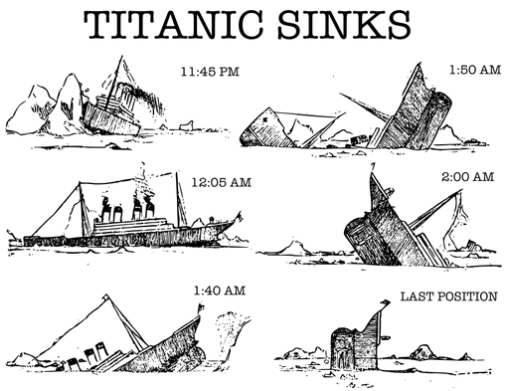
Faculty of Computer Science, Cloud Computing

Agenda

Technische Hochschule Rosenheim

On the menu for today:

- Python
  - Setup
  - IDE and editors
  - Built-in types
  - Control structures



What is Python?

Technische Hochschule Rosenheim

Python ([ˈpʰyːtɔn]) is a general-purpose, commonly interpreted higher programming language.

- **Python** is an interpreted, object-oriented, functional, dynamically-typed, higher programming language with automatic memory management.
- **Python** was developed in February 1991 by Guido van Rossum at the Centrum Wiskunde and Informatica in Amsterdam.
- **Python** is distributed in versions 2.x and 3.x (not backward compatible!).

TIP: Use Python 3 (end of support for Python2 since 2020!).

Note: The name *Python* has nothing to do with the snake, but refers to *Monty Python's Flying Circus*.

Special features of Python

Technische Hochschule Rosenheim

- **Object Orientation**: Everything is an object.
- **Strongly typed**: Each object has a unique immutable type.
- **Dynamically typed**: The type of a variable is decided only at runtime.
- **Automatic memory management**: No manual (error-prone) creation and release of memory.
- **Whitespace sensitive**: Indentation decides about grouping of statements in logical blocks.

Python-Programm

Technische Hochschule Rosenheim

myprogramm.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

from argparse import ArgumentParser

calc = {"+" : lambda a, b: a + b,
        "-" : lambda a, b: a - b,
        "*" : lambda a, b: a * b,
        "/" : lambda a, b: a / b,
        "^" : lambda a, b: a ** b}

parser = ArgumentParser()

def do(op):
    if op in calc:
        print("Result: ", calc[op](args.op1, args.op2))
    else:
        parser.error("{} no valid op".format(op))


if __name__ == '__main__':

    parser.add_argument("-o", "--operation", dest="operation", default="plus")
    parser.add_argument("op1", type=float)
    parser.add_argument("op2", type=float)
    args = parser.parse_args()
    op = args.operation

    do(op)
```

REPL

Technische Hochschule Rosenheim

What happens when you call Python without parameters?

```
> python3
```

REPL

Technische Hochschule Rosenheim

What happens when you call Python without parameters?

```
> python3
```

```
Python 3.5.2 (default, Nov 12 2018, 13:43:14)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print(3+2)
5
>>>
```

- Starts Python in REPL (=Read Eval Print Loop) mode
  - *Interactive* or *Language Shell*

Python: Where can I get it?

Technische Hochschule Rosenheim

Python is available on most platforms:

- Windows, Linux/Unix, Mac, ...
- Also for microcontrollers (https://micropython.org/)

Download here:

- Python.org: https://www.python.org/downloads/
- Anaconda: https://www.anaconda.com/
- Miniconda: https://docs.conda.io/en/latest/miniconda.html

Edit Python code

Technische Hochschule Rosenheim

There are several ways to work with *Python*:

- **Shell**: Use the shell in REPL mode
- **Editor**: Visual Studio Code, VIM, Atom, Notepad++, ...
- **IDEs**: PyCharm (Community Edition), Pyzo, ...
- **Online**:
  - https://repl.it/languages/python3
  - http://micropython.org/unicorn/
- **Jupyter Notebooks**: ...later...

Let's get the party started

Technische Hochschule Rosenheim

Variables, assignments and types

The following program example shows definition and usage of simple variables:

```
x = 1    # x references an object 1 of type int
y = 1.0  # y references an object 1.0 of type float

print("x =", x, "has  the  Type",type(x))
print("y =", y, "has  the  Type",type(y))

x = x + 3
y = y * 2
print(x)
print(y)
```

https://repl.it/@marceltilly/TH-Rosenheim#lecture/01-variables.py

Built-in types

Technische Hochschule Rosenheim

The main *built-in* types in Python are:

- **Numeric types**: Integers (int), Floats (float), and Complex numbers (complex).
- **Boolean type**: (boolean) with values *True* and *False*.
- Text string type: **string** (str)
- **Sequence types**: List, tuple and range
- **set types**: set and frozenset
- **mapping type**: Dictionary (dict)

Operators

Technische Hochschule Rosenheim

How to calculate in Python?

If x, y and z are variables, then the following binary and unary arithmetic operations are possible in Python:

addition	z=x+y	x=x+y	x+=y
subtraction	z=x-y	x=x-y	x-=y
multiplication	z=x*y	x=x*y	x*=y
(real) division	z=x/y	x=x/y	x/=y
rounded division	z=x//y	x=x//y	x//=y
modulo	z=x%y	x=x%y	x%=y
power	z=x**2	x=x**2	
negation	z=-x	x=-x	

Strings

Technische Hochschule Rosenheim

Strings can be created and concatenated in several ways:

```
# Strings can be defined in different
# ways (string literals)
name = 'Bond'
prename = "James"
salutation = """My name is"""

#  output
agent = salutation + " " + name + ", " + prename + " " + name + "."
print(agent)

s = "x" * 5
print(s)
'xxxxx'
```

https://repl.it/@marceltilly/TH-Rosenheim#lecture/02-string.py

Control structures 101

Technische Hochschule Rosenheim

IF

```
#  condition  if
condition = "" or (3 - 3)
if condition:
    print("Condition  is true!")
elif 1 == 2:
    print("1 is  equal to 2!")
else:
    print("Nothing  true  here :(")
```

https://repl.it/@marceltilly/TH-Rosenheim#lecture/06-if.py

Exercise 1

Technische Hochschule Rosenheim

Write your first Python programm

Given two numbers, write a Python code to find the Maximum of these two numbers.

Examples:

```
a = 2
b = 4

> b is greater! b is 4

a = -1
b = -4

> a is greater! a is -1
```

Sequence types

Technische Hochschule Rosenheim

```
# list  ist  mutable
l = list()
print(l)
l = [1, "2", list()]
print(l)
l[0] = 9
l.append(10)
print(l)

#  tuple  ist  immutable
t = tuple()
t = ("value", 1)
print(t)
t[0] = t[1]
# error

#  range  ist  immutable
r = range(0,4,1)
# das  selbe  wie  range (4)
print(r)
print(list(r))
```

https://repl.it/@marceltilly/TH-Rosenheim#lecture/03-sequences.py

Zuweisungen

Technische Hochschule Rosenheim

Bei Zuweisungen werden keine Kopien erstellt!

```
l = [0, 4, 42]
k = l
k[1] = 7
print(k)
print(l)

# int , float , str  sind immutable!
i = 42
j = i
j += 1
#  erzeugt  '43  Objekt'  und  weist  es j zu
print(j)
print(i)
```

https://repl.it/@marceltilly/TH-Rosenheim#lecture/04-zuweisungen.py

Set type und dictionary type

Technische Hochschule Rosenheim

- **set**: Sets with unique elements
- **dict**: Key-Value Pair Store

```
# set
s = set()
print(s)
s = set([1 ,2 ,3])
print(s)
print(s == set([1,2,2,1,3]))
#  dictionary
d = dict()
d = {'key': 'value'}
d = {'Paris': 2.5, 'Berlin': 3.4}
print ('Inhabitants  Paris:', d['Paris'], 'Mio')
print ('INhabitants  Berlin:', d['Berlin'], 'Mio')
```

https://repl.it/@marceltilly/TH-Rosenheim#lecture/05-set.py

Control strutures 101

Technische Hochschule Rosenheim

IF

```
#  if
condition = "" or (3 - 3)
if condition:
    print("Condition  is true!")
elif 1 == 2:
    print("1 is  equal to 2!")
else:
    print("Nothing  true  here :(")
```

https://repl.it/@marceltilly/TH-Rosenheim#lecture/06-if.py

Functions

Technische Hochschule Rosenheim

- Functions are used to swap and systematize statements, which can then be conveniently executed any number of times by calling the function. The function can receive input arguments and return objects itself.
- Functions in Python are called with the keyword `def`, the function name and the passed parameter list as follows:

```
def function_name(a,b,...):
    <operations>

#  Define a function to sum numbers
def sum(a, b):
    return a+b

result = sum (1,2)
print(result)
```

https://repl.it/@marceltilly/TH-Rosenheim#lecture/10-functions.py

Exercise 2

Technische Hochschule Rosenheim

Write your second Python programm

Given two numbers, write a Python function whcih gets 2 numbers and outputs the resutl:

Examples:

```
Input: a = 2,b = 4

Output: b is greater

Input: a = -1,b = -4

Output: a is greater
```

Summary

Technische Hochschule Rosenheim

Lessons learned today:

- Python Basics
  - built-in types
  - control structures
  - etc.
- Python setup and editors

Final remark

Technische Hochschule Rosenheim

STOP making fun of different programming languages

C is FAST

Java is POPULAR

Ruby is COOL

Python is BEAUTIFUL

Javascript

Haskell is INTRIGUING