



# Python

2018, Fulda, Germany

Nationaler Akademietag in Fulda 2018

Maciek Wichary, Sławek Wernikowski, OpenEDG Python  
Institute

April 2018

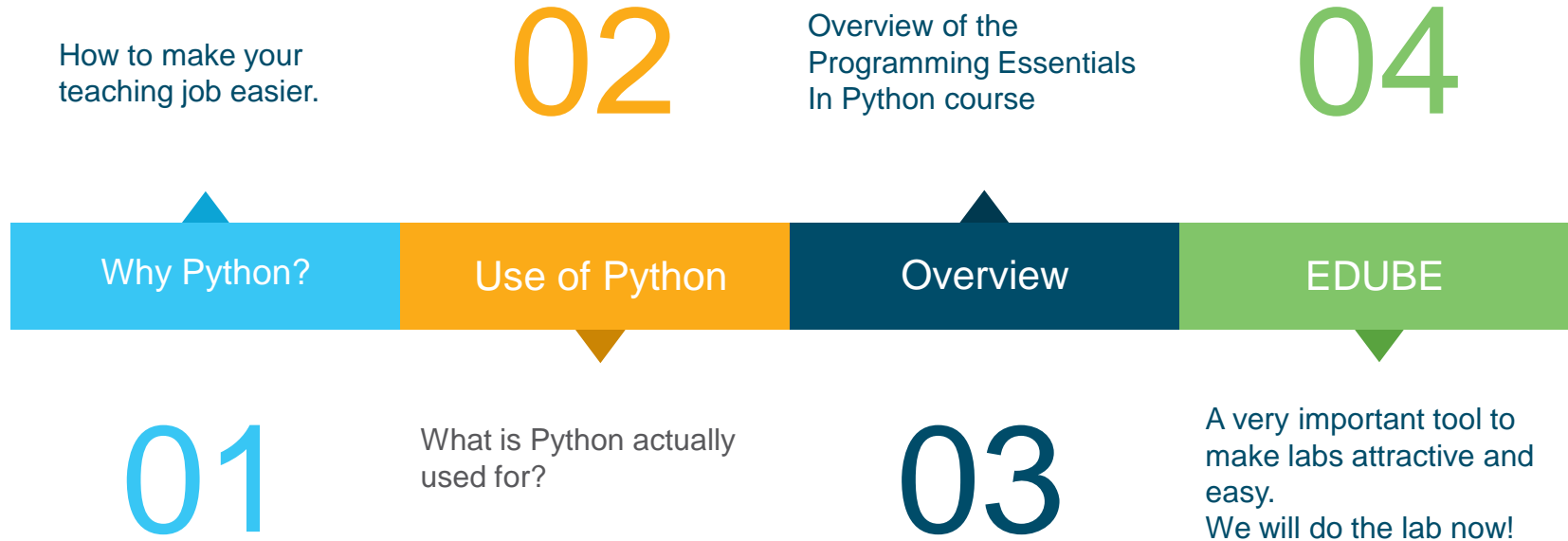




Yes  
You can find  
me in this  
picture!

My very first CCNA group back in 2002  
All of my students have great jobs now.

# Agenda





python Germany Search

jobs.thelocal.com

1-20 of 848 python jobs Sort by relevance

**Your Current Search**

Keywords: python Germany

**Refine Your Search**

Distance: Within 50 kms

**Position type**

Contract Full Time Part Time Temporary

**Category**

Administrative Agricultural Banking & Finance Biotech and R&D Business Strategy

**Company**

About You GmbH ALLEN GmbH

**Software Developers - Linguistics/NLP (Fluent German, Italian or French)**  
EU Recruit - Munich, Bavaria, DE  
Contract positions running until the end of the year focused on Natural Language Processing and Linguistics working on the development of a virtual assistant.

**Software Developer (m/f) Python**  
Yieldlive GmbH - Hamburg, Hamburg, DE  
Yieldlive GmbH is looking for a Software Developer (m/f) Python (ID NUMBER: 4844584) in Hamburg

**Java Developer Machine Learning - Java/ Python (m/f)**  
CHECK24 - Munich, Bavaria, DE  
CHECK24 is looking for a Java Developer Machine Learning - Java/ Python (m/f) (ID NUMBER: 4749098) in München

**Software Developer (C++/Python) for Autonomous Driving (f/m)**  
BMW Group - Munich, Bavaria, DE  
BMW Group is looking for a Software Developer (C++/Python) for Autonomous Driving (f/m) (ID NUMBER: 4648717) in München

**Web Full Stack Engineer (Python/React), Bayern**

the process

for assessments





# What is Python actually used for?



UBER



They were all written, to a greater or lesser extent, in Python.

Battlefield 2, Battlefield 2142 and  
All the games use Python



# Other examples

- Internet Applications (BitTorrent, Jogger Publishing Assistant, TheCircle, TwistedMatrix)
- 3D CAD/CAM (FreeCAD, Fandango, Blender, Vintech RCAM)
- Enterprise Applications (Odoo, Tryton, Picalo, LinOTP 2, RESTx)
- Image Applications (Gnofract 4D, Gogh, imgSeek, MayaVi, VPython)
- Mobile Applications (Aarlogic C05/3, AppBackup, Pyroute)
- Office Applications (calibre, faces, Notalon, pyspread)
- Personal Information Managers (BitPim, Narval, Prioritise, Task Coach, WikidPad)



# Generally, Python is a great choice for:

- Web and Internet development (e.g., Django and Pyramid frameworks, Flask and Bottle micro-frameworks)
- Scientific and numeric computing (e.g., SciPy – a collection of packages for the purposes of mathematics, science, and engineering; Ipython – an interactive shell that features editing and recording of work sessions)
- **Education (it's a brilliant language for teaching programming! And that's why we're offering this course to you!)**
- Desktop GUIs (e.g., wxWidgets, Kivy, Qt)
- Software Development (build control, management, and testing – Scons, Buildbot, Apache Gump, Roundup, Trac)
- Business applications (ERP and e-commerce systems – Odoo, Tryton)



Python's strength is not about its speed. It is all about simplicity and readability.



# Overview

## PCAP | Programming Essentials In Python

# 01

For beginners with little or no prior knowledge of programming.

# 02

Designed to be a full-semester course: 5 modules, 5 quizzes, 5 module assessments, 30+ lab exercises, 2 summary tests, and 1 final test.



# 03

Accessed online with no special equipment or system requirements.

# 04

Instructor-led training offered at no cost.

# The course and the lab online



# How to start your adventure with Python?

Please, enter our site:

<http://workshop.pythoninstitute.org>

# Agenda

Problem 1: The strange plant

Problem 2: Paying our dues

Problem 3: Indians' investments

Problem 4: How did Caesar write his mails?

Problem 5: How to read Caesar's mails?



# What is a **variable**?

A **variable** is a named container.

You can put a single value or even a couple of different values into such a container.

Some words could not be used to name variables and... doesn't matter.

**variable = value**

**variable = expression**

problem  
#1



# What is a **value**?

To make things clear, we can say that a value can be:

**a number:**

**1**

**3.1415**

**a string:**

**"This is a string"**

**'This is a string, too'**

**problem  
#1**



# How to output a value?

If you want to output a value to the screen/console, you use **print()**

```
variable = 123  
print(1)  
print("This is a string")  
print(variable)
```

problem  
#1





# How to input a value?

If you want to input a value from the keyboard, you use **input()**

Note: the input is always a string!

```
var1 = input()
```

```
var2 = input('Input a word')
```

problem  
#1



# How to input a number?

If you want to input a number, you need to convert it from a string – **int()** or **float()** will do it for you.

**howmany = int(input())**

**howmuch = float(input())**

problem  
#1



# How to check a variable?

If you want to check if a variable contains a specific value, you can ask about it:

```
if variable==1: print('equal')
```

problem  
#1



# How to check a variable?

You can also do something different if the check fails:

```
if variable==1: print('equal')  
else: print('sorry')
```

**problem  
#1**



OK,  
now we can deal with  
**problem #1**

**problem  
#1**





# How to evaluate an expression's value?

Python knows arithmetic – you can rely on it.

Python can **add** numbers:

```
var = a + b
```

problem  
#2



# How to evaluate expression's value?

Python knows arithmetic – you can rely on it.

Python can **multiply** numbers:

```
var = a * b
```

problem  
#2



# How to evaluate expression's value?

Python knows arithmetic – you can rely on it.

Python can **subtract** numbers:

**var = a - b**

problem  
#2





# How to evaluate expression's value?

Python knows arithmetic – you can rely on it.

Python can **divide** numbers:

**var = a / b**

...but, please, don't try do divide by zero  
(unless you want to cause a little disaster)

problem  
#2



# How to evaluate an expression's value?

Your expression can be as complex as you need – you can use parentheses, too.

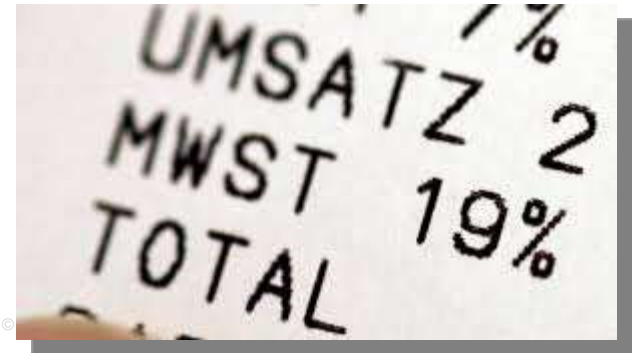
$$\text{var} = (a * (1 + b)) / (b - a)$$

problem  
#2



OK,  
now we can approach  
**problem #2**

problem  
#2





# How to repeat execution of any part of your code?

```
for v in range(1,5):  
    print(v)
```

```
1  
2  
3  
4
```

problem  
#3



# How to repeat execution of any part of your code?

```
for v in range(3):  
    print(v)
```

```
0  
1  
2
```

problem  
#3



OK,  
now we can launch  
**problem #3**

problem  
#3





# How letters are stored within the computer memory?

They are stored as numbers, based on ASCII standard.

**problem  
#4**



# How to get an ASCII code of a letter?

You need to use `ord()`

```
letter = 'A'  
code = ord(letter)  
print(code)
```

(Note: it will print **65** to the console – check it yourself)

problem  
#4





# How to get a letter stored as a given ASCII code?

You need to use `chr()`

```
code = 66  
letter = chr(code)  
print(letter)
```

(Note: it will print **B** to the console – check it yourself)

problem  
#4



# How to traverse through all the letters of a string?

You can use **for**.

```
string = 'abc'  
for c in string:  
    print(c)
```

(Note: it will print **a b c** to the console)

problem  
#4



# How to check if a character is actually an uppercase letter?

```
char1 = 'A'  
char2 = 'z'  
print(char1.isupper()) # True  
print(char2.isupper()) # False
```

problem  
#4



# How to control the output?

```
print('IULIUS')  
print('CAESAR')
```

**IULIUS  
CAESAR**

problem  
#4



# How to control the output?

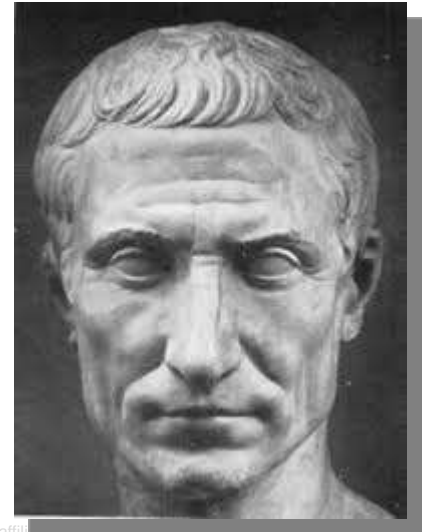
```
print('IULIUS', end="")  
print('CAESAR')
```

**IULIUSCAESAR**

problem  
#4



OK,  
we are ready to cope with  
problem #4



problem  
#4



# How to collect more than one value inside one variable?

You can use a **list**.

```
list = []      # list is empty  
list.append(1) # list has 1 elem  
list.append(2) # list has 2 elems  
print(list)   # [ 1, 2 ]
```

problem  
#5



# How to collect more than one value inside one variable?

You can use a **dictionary**.

```
dict = {}      # dict is empty
dict['a'] = 1  # 1st key/elem
dict['b'] = 2  # 2nd key/elem
print(dict)   # {'a': 1, 'b': 2}
```

problem  
#5





# How to read the content of a file into a variable?

```
with open('filename') as file:  
    text = file.read();
```

problem  
#5



# How to initialize letters' counters?

```
freqdict = {}  
for c in range(ord('A'),ord('Z')+1):  
    freqdict[chr(c)] = 0
```

problem  
#5



# How to count all letters?

```
for c in text:  
    if c.isalpha():  
        freqdict[c] += 1
```

problem  
#5



# How to sort letters by their frequencies?

```
transcodedict = {}  
for c in sorted(freqdict, key=lambda x : freqdict[x], reverse=True):  
    transcodedict[c] = freqstr[0]  
    freqstr = freqstr[1:]
```

problem  
#5



# How to sort letters by their frequencies?

```
transcodedict = {}  
for c in sorted(freqdict, key=lambda x : freqdict[x], reverse=True):  
    transcodedict[c] = freqstr[0]  
    freqstr = freqstr[1:]
```

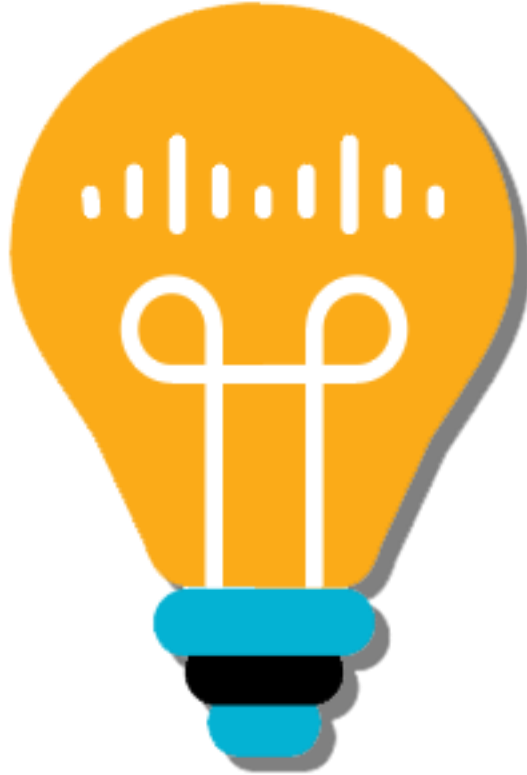
problem  
#5



OK,  
let's break the code!

problem  
#5





Q & A

