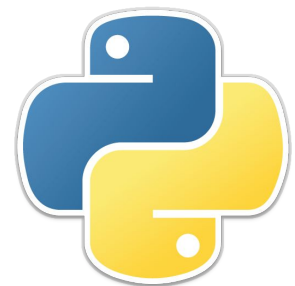
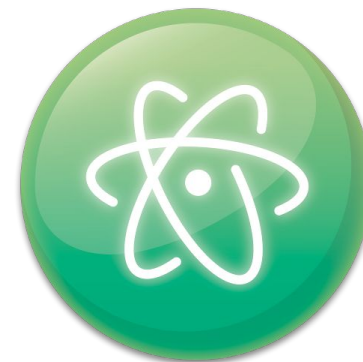
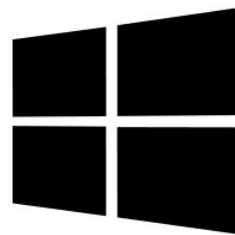
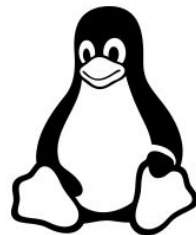
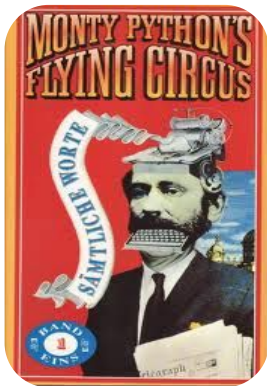


# Grundlagen der Programmierung



# wichtige Information

# Struktur

Dr. Iulian Benta | Bogdan Mic | Dr. Cătălin Rusu

## Workload (in Stunden):

**Vorlesung: 2**

**Seminar/Labor: 2 + 2**

**MS TEAMS:** ron7zmb (Grundlagen der Programmierung)

**Email:** [vasile.rusu@ubbcluj.ro](mailto:vasile.rusu@ubbcluj.ro)

Fragen und Feedback sind immer erwünscht

# Prüfungsform

- **K**lausur (20%)
- **Z**wischenprüfung (20%) → **Woche 7**
- **L**ab (30%)
- **P**raktische Prüfung (30%)

## Minimale Leistungsstandards

**K, P, L  $\geq$  5**



# Anwesenheit

---

- Seminar: **10/14**
  - Labor: **12/14**
  - Kurs: ...
- 
- ohne diese Anwesenheit kann man die Klausur nicht bestehen
  - man sollte in der Regel mit seiner Gruppe das Seminar besuchen
  - Abwesenheiten muss man immer begründen (vom Arzt)

# Ziele

- Wie schreibt man Python-Programme
- Grundlegende Konzepte der objektorientierten Programmierung verstehen und selbstständig nutzen
- Herausforderungen und Lösungsansätze des Softwareentwicklungsprozesses verstehen
- Verstehen der Vorgehens- und Denkweisen von Informatikern
- Erster Einblick in Fähigkeiten und Möglichkeiten der Informatik in IT-Projekten

# Kursinhalt

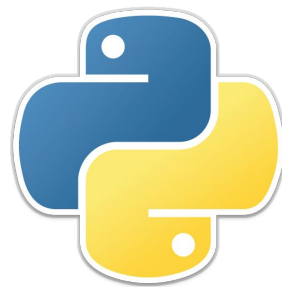
- Hello Python
- Software Entwicklung Intro
- Prozedurale Programmierung
- Modulare Programmierung
- Objektorientierte Programmierung
- Softwarearchitektur
- Vereinheitlichte Modellierungssprache (UML)
- Software Testing
- Rekursive Programmierung
- Komplexitätstheorie
- Suchalgorithmen
- Sortieralgorithmen
- Divide-et-Impera
- Backtracking

# Dieses Semester





# 1. Hello Python





# Dieses Semester

---

Ziel ist

so viel wie möglich

Code zu schreiben

- learn by doing
- wenig Theorie



Python...ist nicht cool

# Python?

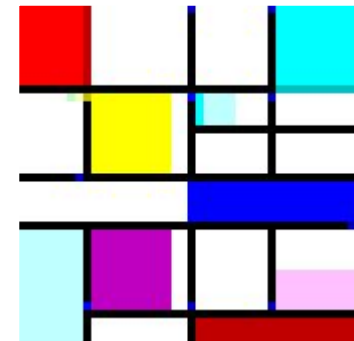
```

🐡 🍷 🍇
🐷 📞 → 🍷 🍇
🍌 firstLetter 🪄 🐶 0 1
🍌 rest 🪄 🐶 1 🐣 🐶
🍏 🍌 rest firstLetter 🍷 🍷 🍌 🍌
🍉
🍉

🏁 🍷
😊 📞 🍷 cat 🍷
😊 📞 🍷 development 🍷
😊 📞 🍷 computer 🍷
🍉

```

IT'S SHOWTIME  
TALK TO THE HAND "Hello, World!"  
YOU HAVE BEEN TERMINATED



```

+++++++ [ >+++++++>+++++++>+++<<<- ] >++.>+.+++++++
..+++.>+>.<<+++++++>.>..+++.- - - - - . - - - - - .>+.

```

...aber einfach funktioniert

# Python?

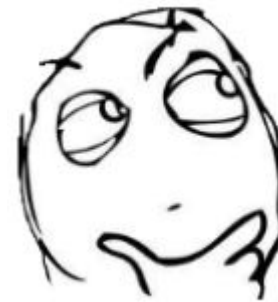


Swap to integers without temp

**CHALLENGE ACCEPTED**



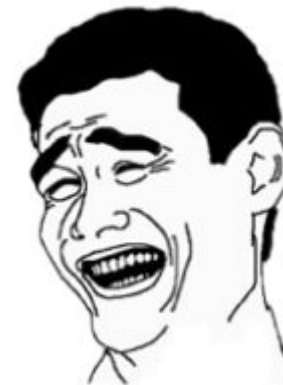
```
a = a + b;  
b = a - b;  
a = a - b;
```



```
a^=b;  
b^=a;  
a^=b;
```



```
a, b = b, a
```





# wieso finde ich Python attraktiv

## Python

- ist **einsteigerfreundlich** und **leicht** zu lernen
- bietet **Viele Möglichkeiten**, ohne unübersichtlich zu werden
- unterstützt mehr als ein **Programmierparadigma** unterstützt
- kommt mit wenigen **Keywords** aus





# wieso finde ich Python attraktiv

## Most popular technologies

This year, we're comparing the popular technologies across three different groups: All respondents, Professional Developers, and those that are learning to code.

### Programming, scripting, and markup languages

2022 marks JavaScript's tenth year in a row as the most commonly used programming language.

But, it's a different picture for those learning to code. HTML/CSS, Javascript and Python are almost tied as the most popular languages for people learning to code. People learning to code are more likely than Professional Developers to report using Python (58% vs 44%), C++ (35% vs 20%), and C (32% vs 17%).

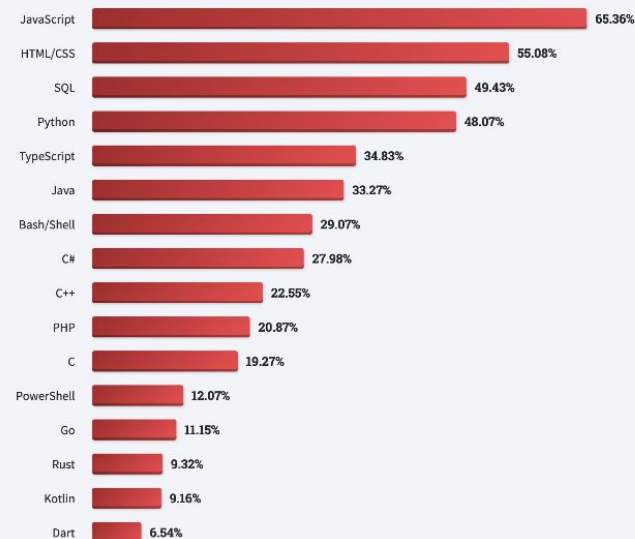
Compared to Professional Developers, those learning to code are less likely to report using SQL (38% vs 53%), TypeScript (15% vs 40%), and Bash/Shell (19% vs 29%).

All Respondents

Professional Developers

Learning to Code

71,547 responses



## Overview

### Developer Profile

#### Technology

Most popular technologies

Most loved, dreaded, and wanted

Worked with vs. want to work with

Top paying technologies

Version Control

Web3

### Work

### Community

### Professional Developers

### Methodology

#### Knowledge, Collaboration, Community

Stack Overflow for Teams enables enterprises to capture, share, & collaborate on knowledge internally.

#### Advertise with us

Promote your product or service to developers and technologists on Stack Overflow.

#### Build your employer brand

Engage the world's technology talent with your employer brand.

#### Explore Collectives™ on Stack Overflow

Find trusted answers faster, engage with product experts.

# Python ist wichtig weil...



Business Technology Standard  
[www.managebt.org](http://www.managebt.org)



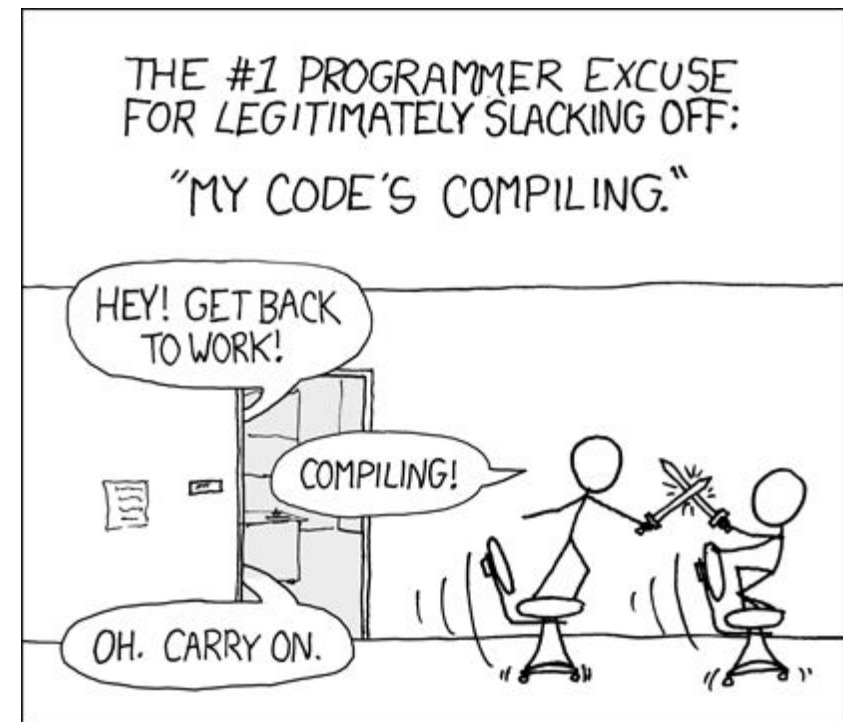
# Python ist leicht zu lernen

- ist meist wohl strukturiert
- intuitiv
- gut lesbar

```
127 found_pub=crossref_lookup.lookup(xml)
128 if len (found_pub):
129     most_probable = max(found_pub,key=lambda item:item[2])[1]
130
131     return most_probable
132
133 def main ():
134     random.seed(64)
135     prepare_data ()
136
137     pop = toolbox.population(n=50)
138     CXPB, MUTPB, NGEN = 0.5, 0.2, 20
139
140     print("Start of evolution")
141
142     # Evaluate the entire population
143     fitnesses = list(map(toolbox.evaluate, pop))
144     for ind, fit in zip(pop, fitnesses):
145         ind.fitness.values = fit
146
```

# Python ist eine Interpretersprache

- mit interaktiver Shell
- erzeugt Python-Bytecode
- nutzt Stackbasierte VM
- gut dokumentiert!



<http://xkcd.com/303/>

# Python ist eine moderne Sprache

- Objektorientiert
- Skalierbar
- OS unabhängig
- Reich an Libraries
- Erweiterbar



<http://xkcd.com/138/>



# Einige Regeln für dieses Semester

- **Beautiful** is better than ugly
- **Explicit** is better than implicit
- **Simple** is better than complex
- **Flat** is better than nested
- **Sparse** is better than dense
- **Readability** counts
- ...
- Code muss getestet werden
- Ohne Kommentare geht nicht
- Source Code Versioning ist wichtig

Zen of Python - Ein Mantra



# Python Grundlagen

man kann Python-Programme interaktiv eintippen (mit der interaktiven Shell)

```
>>> print ("Hallo Welt!")  
Hallo Welt!  
>>>
```

```
>>> a = int(input ("a: "))  
a: 4  
>>> b = int(input ("b: "))  
b: 6  
>>> c = a + b  
>>> print (c)  
10  
>>>
```



# Python Grundlagen

lass uns üben



**Programmers then:**



**I just coded for  
Apollo mission with  
50KB storage.**

**Programmers now:**



**My code wont  
compile because of  
indention error :(**





## Beispiel

```
numar = int(input("Introduceți un număr: "))
prim = True
if numar > 1:
    # Verificăm divizibilitatea numărului cu
    # numerele de la 2 la rădăcina pătrată a numărului
    for i in range(2, int(numar**0.5) + 1):
        if (numar % i) == 0:
            prim = False
            break
if prim:
    print("numărul nu este prim.")
else:
    print("numărul este prim.")
```



## Beispiel

```
suma = 0

for num in range(100, 201):
    if num > 1:
        is_prim = True
        for i in range(2, num):
            if (num % i) == 0:
                is_prim = False
                break
        if is_prim:
            suma += num

print("Suma numerelor prime între 100 și 200 este", suma)
```



## Beispiel

```
lista = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
sum2 = 0
```

```
produs3 = 1
```

```
for i in range(len(lista)):
```

```
    if i % 2 == 0:
```

```
        sum2 += lista[i]
```

```
    if i % 3 == 0:
```

```
        produs3 *= lista[i]
```

```
print("Suma fiecărui al doilea element:", sum2)
```

```
print("Produsul fiecărui al treilea element:", produs3)
```



# Python Grundlagen

lass uns üben

1. Schreibe ein Programm, das die Anzahl von Erscheinungen eines Elementes in einem String ausgibt

Beispiel: a und abcdeada → 3

2. Schreibe ein Programm, das alle Vokale aus einem String mit ? ersetzt

Beispiel: abcdeada → ?bcd??d?

3. Schreibe ein Programm, das die Summe aller Zahlen einer Liste ausgibt

Beispiel: ['a', 12, [1], 23, 'b'] → 35