

Seminar 1

1. Installing Python

<https://www.python.org/downloads/>

2. Ausführen von Python-Code

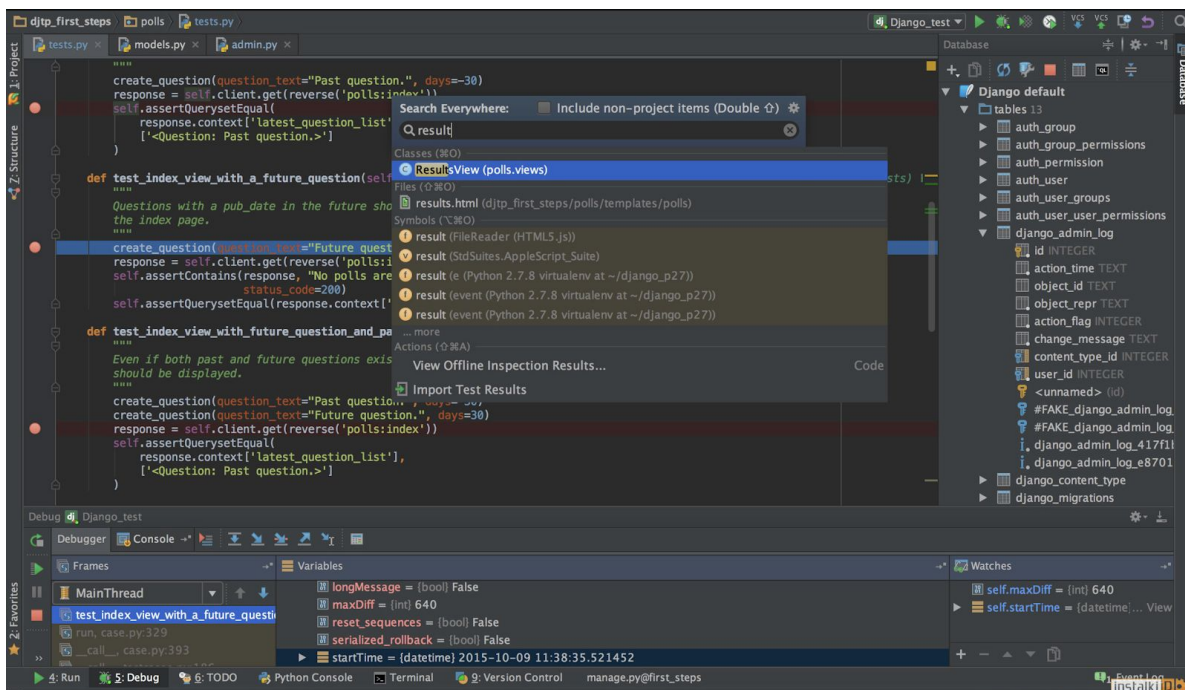
- man kann Python-Programme interaktiv eintippen (mit der interaktiven Shell)

```
Catalins-iMac:~ cat$ python3
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 26 2018, 23:26:24)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hallo Welt!")
Hallo Welt!
>>>
```

- oder in Dateien speichern und ausführen
- um Dateien zu editieren und abzuspeichern, benötigt man einen Editor
- man führt in Terminal (Linux, MacOS) bzw. Cmd/Eingabeaufforderung (Windows) den Code aus

```
Catalins-iMac:~ cat$ vim hallo.py
Catalins-iMac:~ cat$ ls
Applications      Documents          Library            Music              Public
Desktop           Downloads          Movies             Pictures           hallo.py
Catalins-iMac:~ cat$ cat hallo.py
print("Hallo Welt!")
Catalins-iMac:~ cat$ python3 hallo.py
Hallo Welt!
Catalins-iMac:~ cat$
```

- oder man verwendet einfach einen IDE, um Code zu schreiben und auszuführen



PyCharm IDE

<https://www.jetbrains.com/pycharm/>

3. Ein erstes Beispiel

```
x = 5
if x < 6:
    print "less"
else:
    print "more"

while x < 10:
    print x
    x += 1
```

```
a = 1
while a < 5:
    print a
    a = a + 1
print "gata"
```

```
a = 1
print a
a = "abc"
print a
a = [1, 2, 4, 3]
print a
a.sort()
print a
```

4. Ein- und Ausgabe

- für Eingabe über die Tastatur bietet Python die Funktion `input()`
- `input()` liefert immer einen String zurück
- Die Funktion `print()` druckt beliebig viele Werte aus, die durch Komma getrennt sind
- Mit dem Parameter `sep` kann man den Separator, der zwischen den Werten ausgegeben wird, auf einen beliebigen Wert setzen

```
>>> name = input('name=')
name=Cat
>>> print(name)
Cat
>>>
>>> num = int(input('num='))
num=10
>>> print(num)
10
>>> print(192,168,178,42,sep=".")
192.168.178.42
>>>
```

5. Listen und Strings

- Eine `list` kann Elemente unterschiedlichen Datentyps enthalten
- Die Funktion `len()` bestimmt die Anzahl der Elemente der Liste
- Listen können auch Listen enthalten, auch sich selbst
- Hinzugefügt werden Werte mit dem `+`-Operator und den Funktionen `append()` und `insert()`
- zugriff auf Elemente mit dem `[]`-Operator über den Index

```
>>> l = [1,2,3,"a", "b"]
>>> l[0]
1
>>> l[1]
2
>>> l.append("c")
>>> l
[1, 2, 3, 'a', 'b', 'c']
>>> len(l)
6
>>> l.pop()
'c'
>>> l
[1, 2, 3, 'a', 'b']
>>> len(l)
5
>>>
```

- string verwendet man für die Verarbeitung von Zeichenketten

```
>>> string = "abcd"
>>> len(string)
4
>>> string[1]
'b'
>>> string[2]="B"
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object does not support item assignment
>>>
```

6. Übungen

1. Schreiben Sie ein Programm, das die Summe zweier eingegebener ganzzahliger Zahlen ausgibt.
2. Schreiben Sie ein Programm, das Sie nach Ihrem Vor- und Nachnamen fragt. Die Anwendung gibt eine Begrüßung aus. Die Begrüßung enthält den gesamten Namen sowie die Anzahl der Zeichen, die der gesamte Name enthält.
3. Schreiben Sie ein Programm, das eine Folge von Ganzzahlen bis 0 einliest und die Summe aller gelesenen Zahlen ausgibt.
4. Schreiben Sie ein Programm, das alle Wiederholungen von Zeichen aus einer Zeichenkette entfernt.
5. Schreiben Sie ein Programm, das den kleinsten Primzahl größer als N bestimmt.