

## Django und Python

-ein mögliches Framework wäre Django welches auf Python basiert

### **Python:**

-Die Sprache ist Multiparadigmisch, man kann mit ihr imperativ, objektorientiert und funktional programmieren

-ist übersichtlicher als andere Sprachen, es gibt weniger Schlüsselwörter und eine reduzierte Syntax (Einrückungen statt Scopes)

-Unterschied zu Java in Python ist alles ein Objekt(Klassen, Typen, Methoden, Module etc...), d.h. der Datentyp ist an den Wert gebunden und nicht an eine Variable

-Ausserdem wird Python meist als Skriptsprache benutzt (bei Skriptsprachen wird z.B. meist auf den Deklarationszwang verzichtet)

-ausserdem sehr beliebte Programmiersprache → viel Hilfestellung im Netz (Tutorial z.B. bei Codecademy)

-Nachteil Zeitaufwand zum lernen

kleines Codebeispiel zum Ausgeben aller Primzahlen Zahlen:

```
# python program to check if x is a perfect square
import math

# A utility function that returns true if x is perfect square
def isPerfectSquare(x):
    s = int(math.sqrt(x))
    return s*s == x

# Returns true if n is a Fibonacci Number, else false
def isFibonacci(n):

    # n is Fibonacci if one of 5*n*n + 4 or 5*n*n - 4 or both
    # is a perfect square
    return isPerfectSquare(5*n*n + 4) or isPerfectSquare(5*n*n - 4)

# A utility function to test above functions
for i in range(1,11):
    if (isFibonacci(i) == True):
        print i, "is a Fibonacci Number"
    else:
        print i, "is not a Fibonacci Number "
```

Output:

1 is a Fibonacci Number

2 is a Fibonacci Number

3 is a Fibonacci Number

4 is not a Fibonacci Number

5 is a Fibonacci Number

6 is not a Fibonacci Number

7 is not a Fibonacci Number

8 is a Fibonacci Number

9 is not a Fibonacci Number

10 is not a Fibonacci Number

## **Django:**

-ist ein eher Backend lastiges Framework, dadurch vielleicht unnötig kompliziert und viel extra Code

-es besteht die Möglichkeiten Datenbanken zu integrieren

-Stackoverflow: angularjs ca 191.000 Fragen, Django 212.000

-Pygraphviz als Modul für Python um Graphen zu erstellen:

```
+++++
from __future__ import absolute_import
from __future__ import unicode_literals
from __future__ import print_function
from __future__ import division

__author__ = """Aric Hagberg (hagberg@lanl.gov)"""

import pygraphviz as pgv

A=pgv.AGraph()

A.add_edge(1,2)
A.add_edge(2,3)
A.add_edge(1,3)

print(A.string()) # print to screen
print("Wrote simple.dot")
A.write('simple.dot') # write to simple.dot
+++++
```

-aber sehr simple Funktionen, kaum komplexere Berechnungen möglich

-geschätzte Dauer der Einarbeitung: 10-15h (Codeacademy angabe)