

Alexander Jenke, Theodor Straube

# Übung9: Generators, Requests, Übungsaufgaben

Programmierkurs Python // Mittwoch, 8. Januar 2020

# ToC

- Generators
- Requests-Package
- Aufgaben
- Lösungen

# Generators

- Ermöglichen das Iterieren über eine Menge genau wie ein Iterator
- Aber die Menge muss vorher nicht geladen sein, sondern die Elemente können bei Anforderung generiert werden
- Funktion mit *yield* statt *return*
- Nächstes Element mit *next()*

```
def fibonacci(n):  
    a, b = 0, 1  
    for i in range(n):  
        if i < 2:  
            yield i  
        a, b = b, a + b  
        yield b
```

```
fib = fibonacci(1000)  
print(fib)  
for i in range(11):  
    print(next(fib))
```

```
class Fibonacci:  
    def __init__(self):  
        self.a = 0  
        self.b = 1  
        self.i = 0  
  
    def __iter__(self):  
        return self  
  
    def __next__(self):  
        self.i += 1  
        if self.i < 3:  
            return self.i - 1  
  
        self.a, self.b = self.b, self.a + self.b  
  
        return self.b  
  
fib = Fibonacci()  
c = 0  
for i in fib:  
    print(i)  
    c += 1  
    if c > 10: break
```

# Requests-Package

```
import requests
r = requests.get("http://tu-dresden.com")
print(r.text)
```

- `requests.get(url):` ruft wie angegebene URL auf und gibt ein `requests.api`-Objekt zurück
- `r.content:` gibt Inhalt der Webseite in Byte zurück
- `r.text:` gibt Inhalt der Webseite als String zurück
- `r.json:` gibt Inhalt der Webseite als JSON-Objekt zurück

```
def get_xkcd(id):
    r = requests.get(f"https://xkcd.com/{id}")
    img_link = re.findall(r'<div id="comic">[ \n]+<img src="(.*jpg)"', r.text)
    r = requests.get(f"https://{img_link[0]}")
    img = Image.open(io.BytesIO(r.content))
    return img

if __name__ == '__main__':
    for i in range(1,10):
        img = get_xkcd(i)
        plt.imshow(img)
        plt.show()
```

# Aufgaben

- Aufgaben der letzten Stunde lösen
- Gerne in Gruppen arbeiten
- Bei Fragen helfen wir

# Lösungen

- Werden zusammen erarbeitet
- Werden anschließend in GitHub hochgeladen

# Nächste Woche

- PyTorch