

Parldigi MasterClass

National- und Ständerat

Algorithmen in der Politik – Programmier-Crashkurs für Parlamentsmitglieder

Bern, 2. Dezember 2024

Prof. Dr. Marcel Gygli

Institut Public Sector Transformation, Berner Fachhochschule

Lena Georgescu

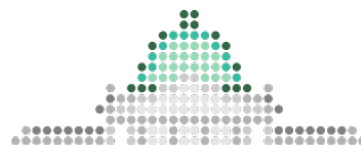
Institut Public Sector Transformation, Berner Fachhochschule

Eine Veranstaltung von:



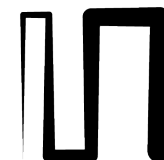
**Universität
Zürich** UZH

Digital Society Initiative



Parldigi

Unterstützt durch:



**Stiftung
Mercator
Schweiz**

Prof. Dr. Marcel Gygli

- ▶ Professor (Tenure Track) an der BFH
 - ▶ KI im Öffentlichen Sektor
- ▶ Innovation Fellow
 - ▶ Schweizerische Nationalbibliothek
- ▶ Wissenschaftlicher Mitarbeiter FHNW
- ▶ PhD in Computer Science



Lena Georgescu

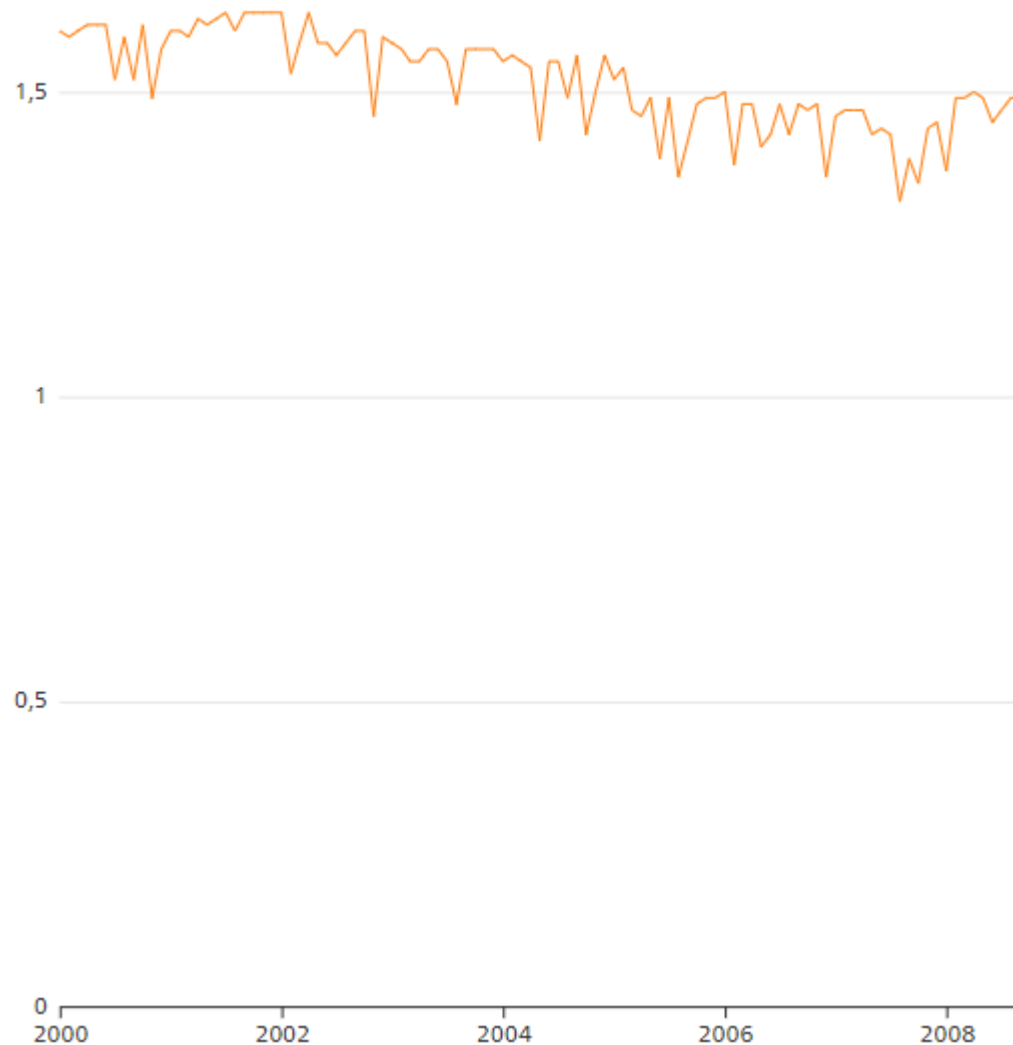


- ▶ Softwareentwicklerin an der BFH
 - ▶ IT im öffentlichen Beschaffungswesen

- ▶ Studentin Informatik Uni Bern

Technologie Heute

- ▶ Das Digitale Zeitalter
 - ▶ Technologie beeinflusst «alles»
- ▶ Software-Entwicklung als Basis
 - ▶ Datenanalysen
 - ▶ Automatisiert repetitive Aufgaben



Was ist Software

- ▶ Enthält klare Anweisungen an den Computer
 - ▶ «Kochrezept»
- ▶ Programmiersprache
 - ▶ Tool für die «Kommunikation» mit Computer
- ▶ Python
 - ▶ Eine von vielen Sprachen

Python Grundlagen

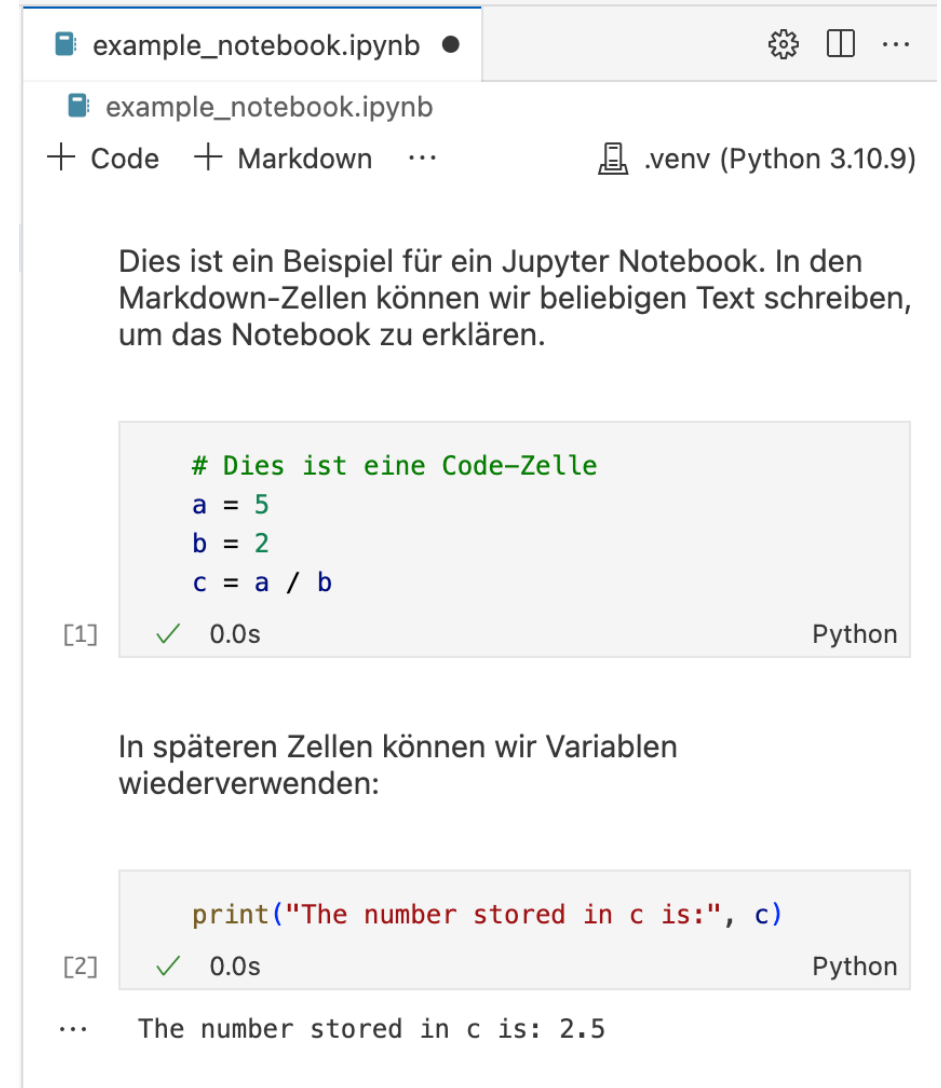
- ▶ Enthält Anweisungen, welche als Ganzes ausgeführt werden
- ▶ Grundbausteine
 - ▶ Variablen und Datentypen
 - ▶ Grundoperationen
 - ▶ Kontrollstrukturen
 - ▶ Funktionen

```
example.py •
hs23_code > example.py > ...
1  """An example python program."""
2
3  def say_hello_to(name):
4      print(f"Hello {name}")
5
6  class_list = ["Paula", "Michael", "Sandra"]
7
8  for student in class_list:
9      say_hello_to(student)
10
```

```
hs23_code — -zsh — 80x24
[schmaett@schmaetts-Air hs23_code % python3 example.py
Hello Paula
Hello Michael
Hello Sandra
schmaett@schmaetts-Air hs23_code %
```

Jupyter Notebooks

- ▶ Besteht aus mehreren Zellen. Der ganze Code in einer Zelle wird gemeinsam ausgeführt
- ▶ Die berechneten Variablen stehen den nächsten Zellen zur Verfügung
- ▶ Man kann auch «Markdown»-Zellen erstellen, um den Code zu beschreiben und das Notebook zu strukturieren



The screenshot shows a Jupyter Notebook window titled "example_notebook.ipynb". The interface includes a top bar with a settings gear, a window icon, and a menu icon. Below the title bar, there are tabs for "example_notebook.ipynb" and a "+ Code" button. The notebook content is displayed in a light gray background. The first cell is a markdown cell containing the text: "Dies ist ein Beispiel für ein Jupyter Notebook. In den Markdown-Zellen können wir beliebigen Text schreiben, um das Notebook zu erklären." The second cell is a code cell containing Python code: "# Dies ist eine Code-Zelle", "a = 5", "b = 2", and "c = a / b". The code cell has a status bar at the bottom left showing "[1] ✓ 0.0s" and "Python" at the bottom right. The third cell is a code cell containing the code: "print('The number stored in c is:', c)". This code cell has a status bar at the bottom left showing "[2] ✓ 0.0s" and "Python" at the bottom right. Below the code cell, the output is displayed: "... The number stored in c is: 2.5".

Variablen und Datentypen

- ▶ Variablen

- ▶ Speichert Information innerhalb einer Applikation

- ▶ Datentypen

- ▶ Integer (Ganze Zahlen) `einwohner = 8600000`
- ▶ Floats (Fließkommazahlen) `zins_rate = 1.5`
- ▶ Strings (Zeichenketten) `land = «Schweiz»`
- ▶ Boolean (Wahrheitswerte) `is_eu_member = false`

Grundoperationen

- ▶ Arithmetische Operationen (+, -, *, /)

```
previous_population = 8500000
current_population = 8600000
growth = current_population - previous_population
growth_rate = (growth / previous_population) * 100

print(f"Growth rate: {growth_rate}%")
```

Kontrollstrukturen - Bedingungen

- ▶ Erlaubt Entscheidungen im Code
 - ▶ Schlüsselwort `if`

```
budget = 750000
if budget > 500000:
    print("Budget exceeds half a million francs.")
else:
    print("Budget is within acceptable limits.")
```

Kontrollstrukturen - Schleifen

- ▶ Aktion für mehrere Elemente ausführen

```
cantons = ["Zurich", "Bern", "Geneva"]  
  
for canton in cantons:  
    print(f"Canton: {canton}")
```

Funktionen

- ▶ Wiederverwendbare Funktionalität
 - ▶ Führen klare Aktion aus

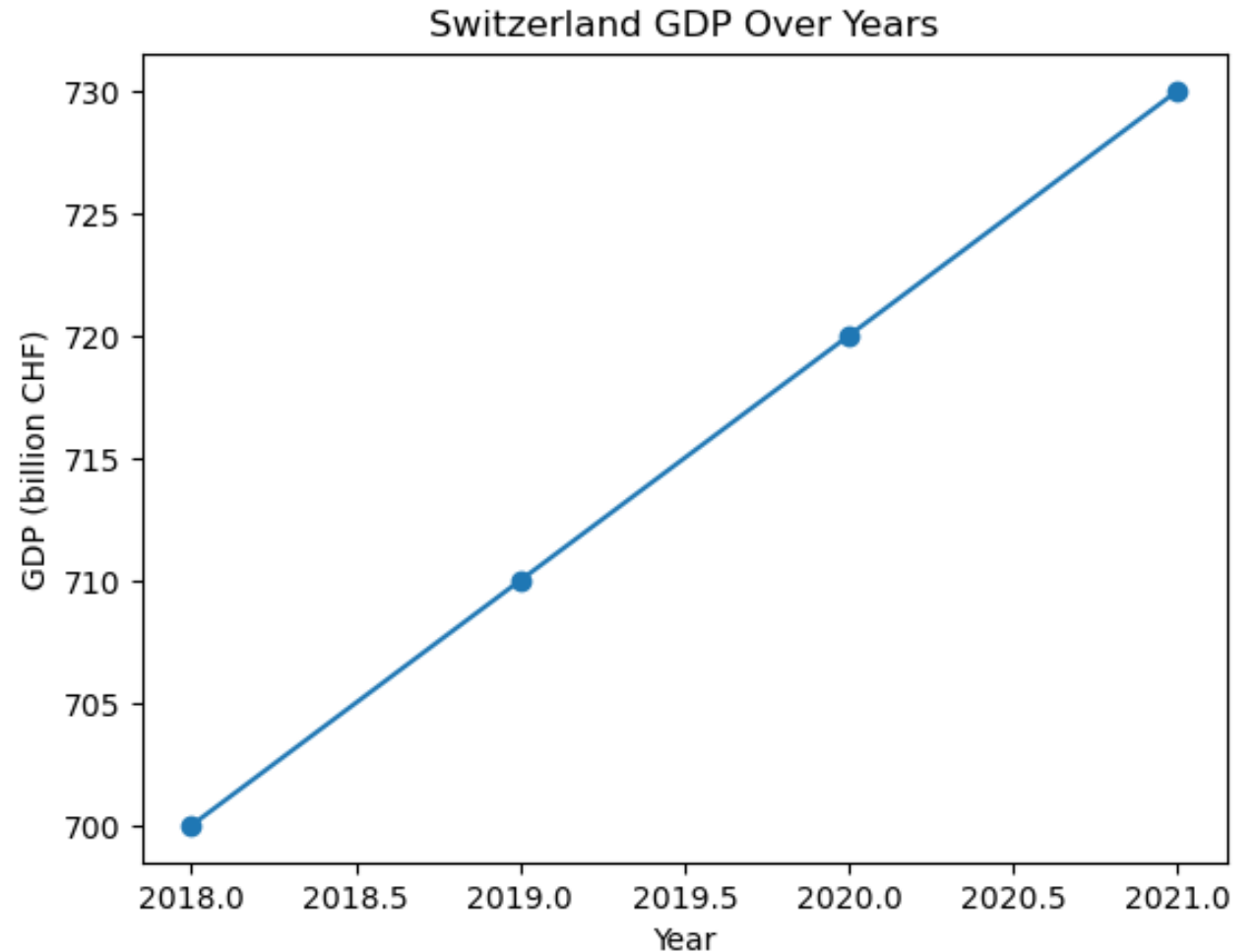
```
def greet(name):  
    return f"Hello, {name}!"  
  
print(greet("Member of Parliament"))
```

Alles zusammen - Datenvisualisierung

```
import matplotlib.pyplot as plt

years = [2018, 2019, 2020, 2021]
gdp = [700, 710, 720, 730]

plt.plot(years, gdp, marker='o')
plt.title("Switzerland GDP")
plt.xlabel("Year")
plt.ylabel("GDP (billion CHF)")
plt.show()
```



Jetzt

- ▶ Auswerten von Daten aus Smartvote
- ▶ Verwendung von Jupyter-Notebooks
- ▶ Link: <https://bit.ly/parldigi-masterclass>

Parldigi MasterClass

National- und Ständerat

nächste Veranstaltung **am 3. März 2025**

Digitale Souveränität

Prof. Dr. Yaniv Benhamou
Université de Genève

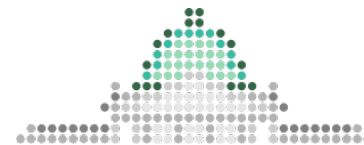
Prof. Dr. Matthias Stürmer
Bernere Fachhochschule / Universität Bern

Eine Veranstaltung von:



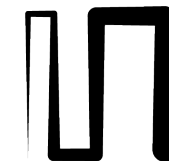
**Universität
Zürich** ^{UZH}

Digital Society Initiative



Parldigi

Unterstützt durch:



**Stiftung
Mercator
Schweiz**