

Bachelor Studiengang
Digital Construction

**Digital Twin
Programmieren**

Modul: Submodul 1: Grundlagen und Theorie

Thema: Einführung

Datum: 16.09.2022, 18:30 – 20:55 Uhr

Dozierende:

Michal Rontsinsky, beyondBIM AG

Nora Bukovits, PAZ ACADEMY

Andrés Velasco Muro, PAZ ACADEMY

Martin Loucka, ioLabs AG

Juri Jerg, HSLU

Hochschule Luzern

Departement Technik & Architektur

FH Zentralschweiz



Agenda

Agenda

Zeit	Thema
18:30 – 19:00	Vorstellung - Ausgangslage Einführung in das Modul, Semesterplan Semesterleistung und Modulendprüfung
19:00 – 19:30	Was ist Python, warum Python Kontext Scripting vs. Programming
19:30 – 20:00	Projektbeispiel Werkzeuge, ChatGPT Grundlagen und Weiterbildung
20:00 – 20:15	Aufgaben für die SW2 Ende

Vorstellung Team

Vorstellung Team und Gastreferenten



Michal Rontsinsky
beyondBIM.ch

Fokus:

- Python
- Entwicklungsumgebung
- Externe Module



Martin Loucka
ioLabs.ch

Fokus:

- Coaching
- MEP



Andrés Velasco Muro
pazacademy.ch

Fokus:

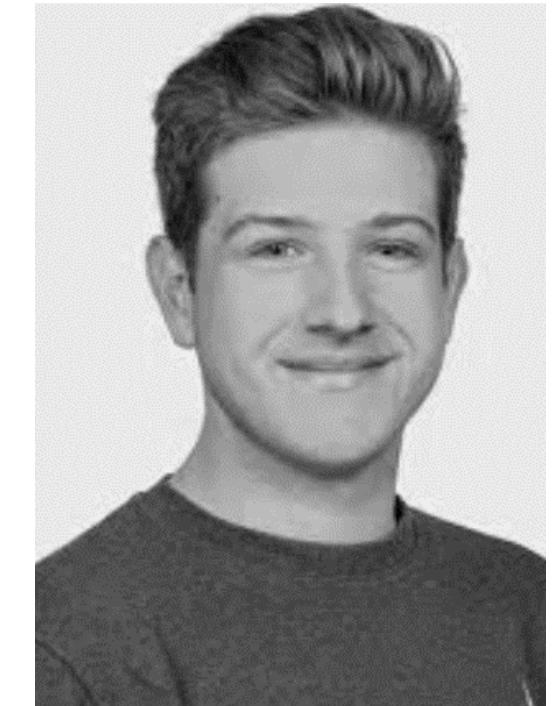
- Coaching
- Parametrisierung und Geometrie



Nora Bukovits
pazacademy.ch

Fokus:

- Coaching
- Parametrisierung und Geometrie



Juri Jerg
HSLU

Fokus:

- Tutoriat
- Python Grundlagen
- TkInter (UI Modul)
- Coaching

Michal Rontsinsky - Founder | Technology and Projects



Founder beyondBIM AG

Dozent Bachelor Studiengang Digital Construction HSLU

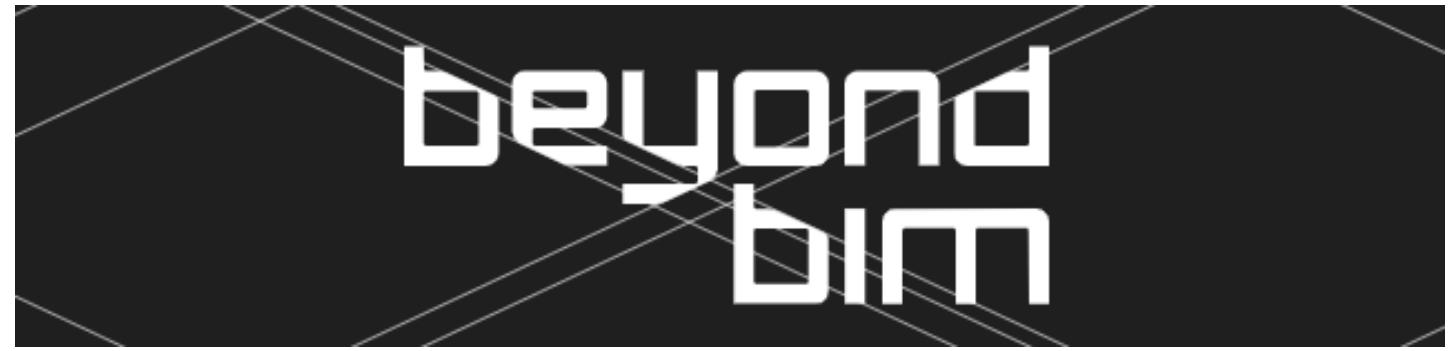
Dipl. Ing, TU Wien

Jahrgang 1985

Aufgabenschwerpunkte

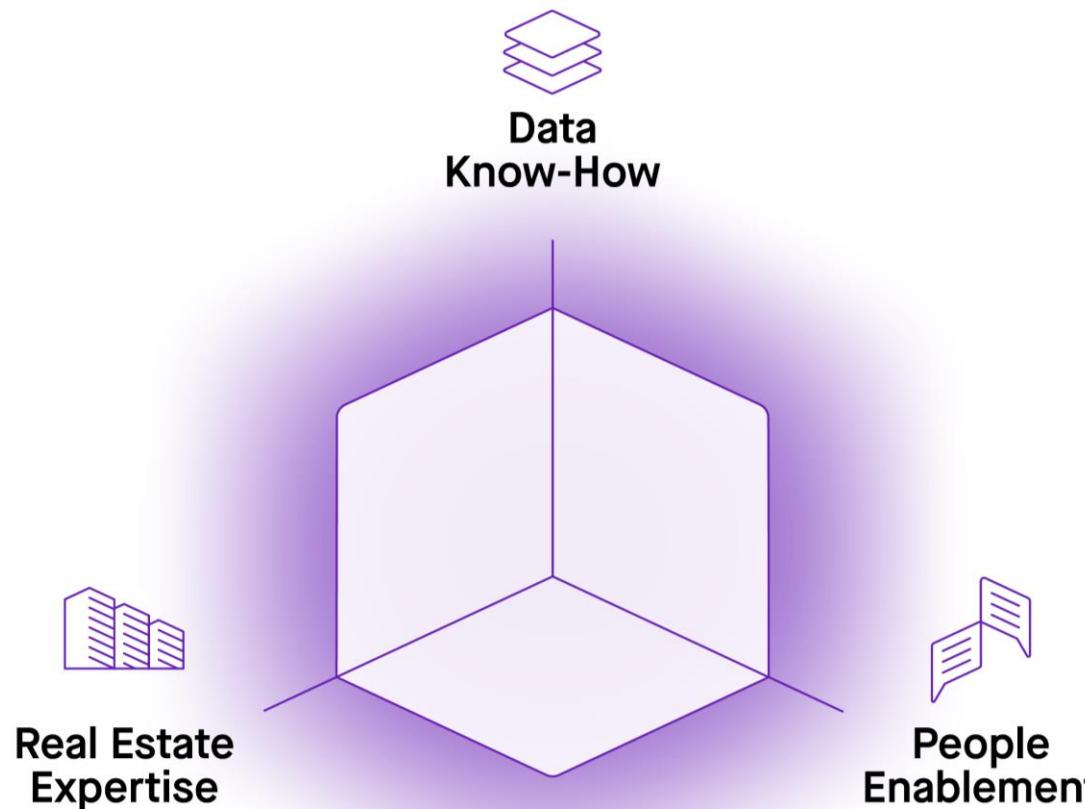
- Konzeption und Umsetzung von BIM Strategien und Implementationskonzepten
- BIM Projektentwicklung bauherrenseitig
- BIM Management bauherrenseitig
- BIM Gesamtkoordination und Aufbau der BIM-fähigen Organisation in Projekten, Qualitätsmanagement
- Schulung und Betreuung Projektbeteiligte
- BIM Wettbewerbsbetreuung
- Datenmanagement
- Programmierung und Entwicklung der Prototypen im Bereich BIM / LCDM (Life Cycle Data Management)





from data to value

beyondBIM befähigt Eigentümer:innen im digitalen Bestellen und Betreiben von komplexen Immobilien. Dabei stellen wir die Menschen und ihre Prozesse konsequent ins Zentrum und schaffen aus Daten messbare Mehrwerte.



BEYONDBIM

beyondBIM realisiert digitale Gesamtlösungen für komplexe Immobilien & Infrastrukturen und **vereint dabei Proptech und Consulting**. Der **Erfolgsfaktor** unseres Vorgehens liegt im **Verständnis für das Gleichgewicht zwischen Technologie, Innovation und den beteiligten Menschen**.

DATEN

Gründung: Juli 2022 (AG)

MA: 9

Kunden: CREM / REIM

Finanzierung: 100% unabhängig

Standort: Zürich

Wiederbeauftragung: 100%

LEISTUNGEN

Advisory

- ✓ Equity Stories
- ✓ Technical Due Diligence
- ✓ **Business Model Development**

Services

- ✓ BIM & LCDM Strategie und Organisationsentwicklung
- ✓ **Use Case Entwicklung**
- ✓ Datenqualitätsmanagement
- ✓ **BIM Umsetzung & Projektbegleitung**

Solutions

- ✓ Lösungs-Konzeption
- ✓ **Prototyping**
- ✓ Sandbox-as-a-Service
- ✓ Aufbau operativer DataPipelines

Academy

- ✓ Master Class BIM/LCDM
- ✓ Unternehmensspezifische Programme
- ✓ IFC-Deepdives

ETH zürich

Alfred Müller

INSELGRUPPE

Flughafen Zürich

BregallInvestments

EY Parthenon

Implenia®

Stiftung Gesundheitsförderung
Bad Zurzach + Baden

Kanton Zürich
Baudirektion

ZOOH!
ZÜRICH

IMMOFONDS
asset management

Kanton St.Gallen
Hochbauamt

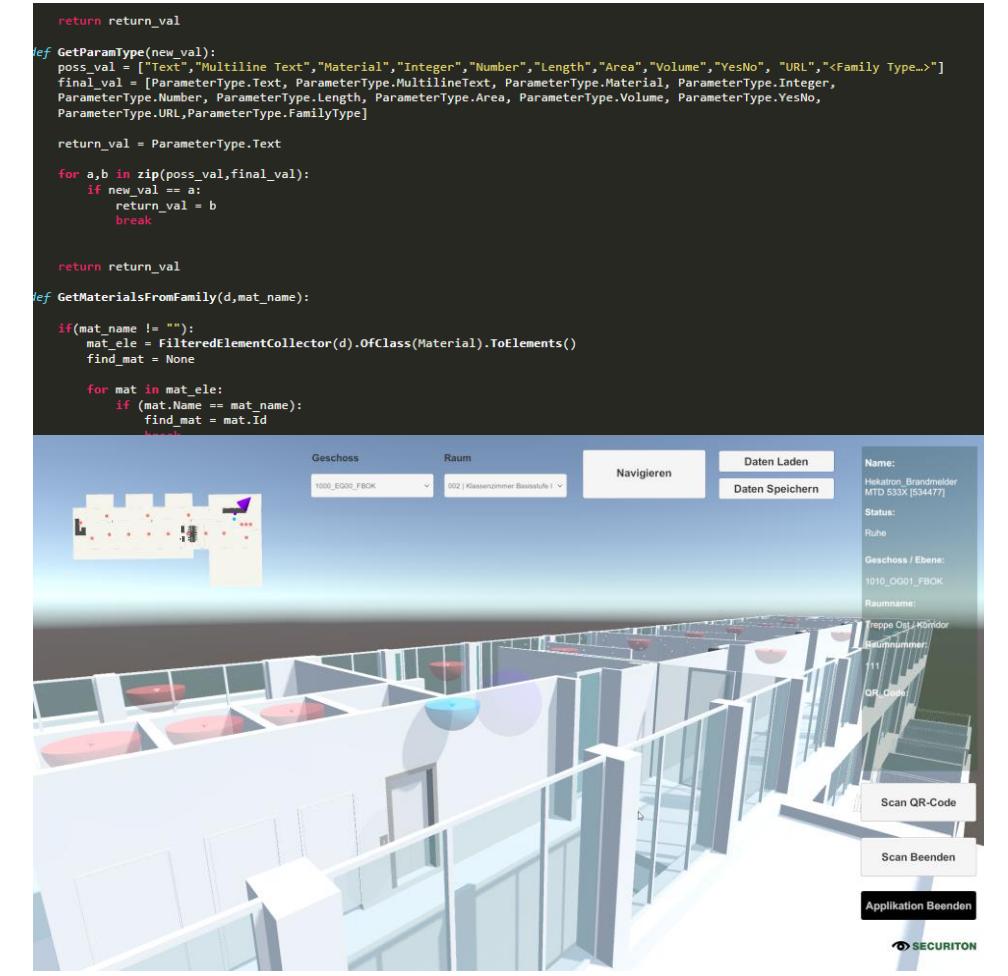
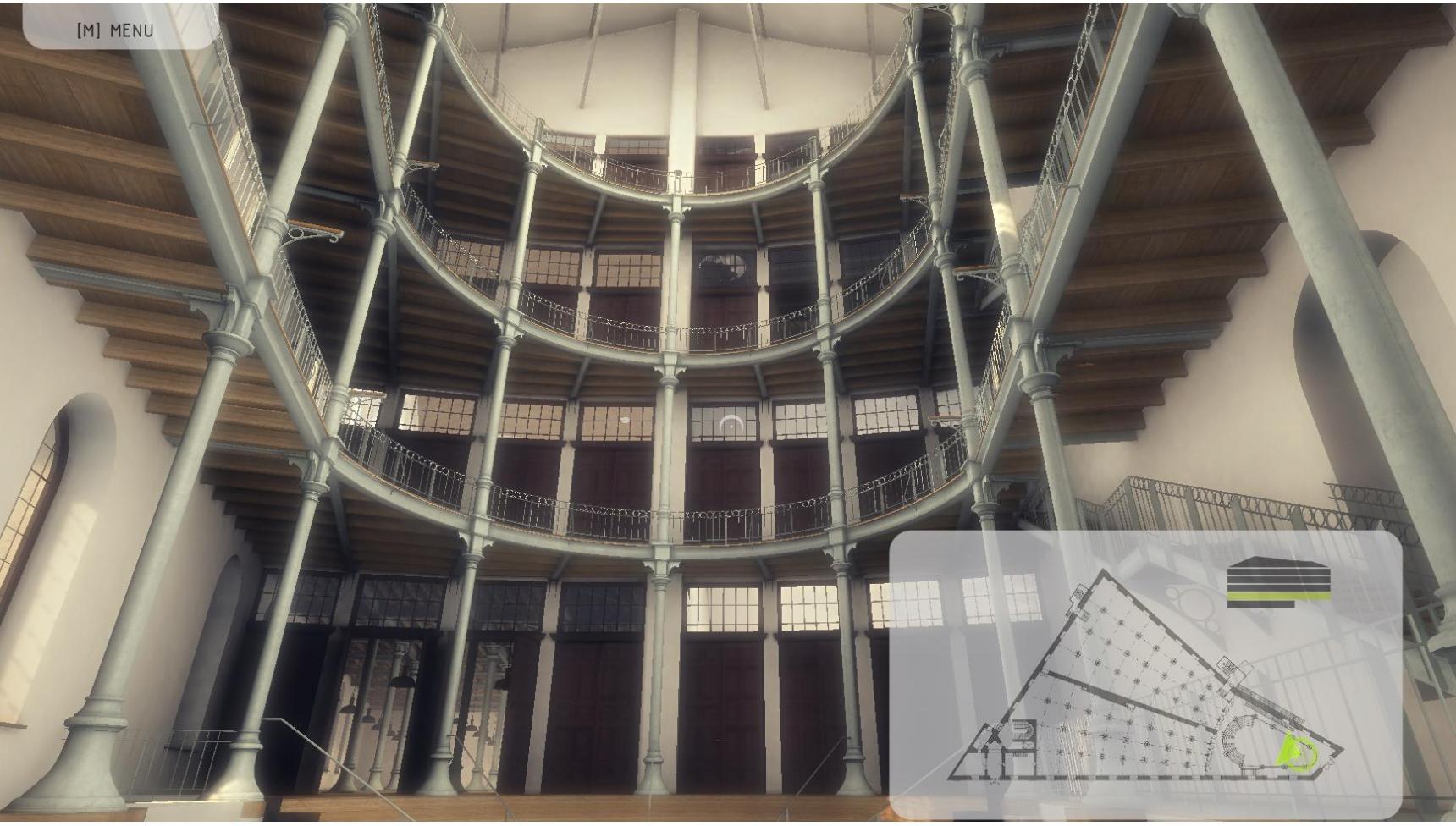
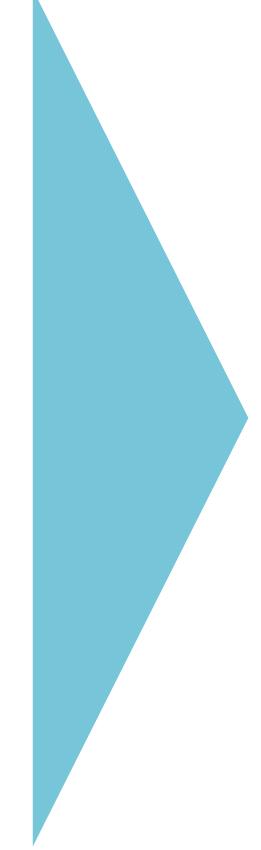
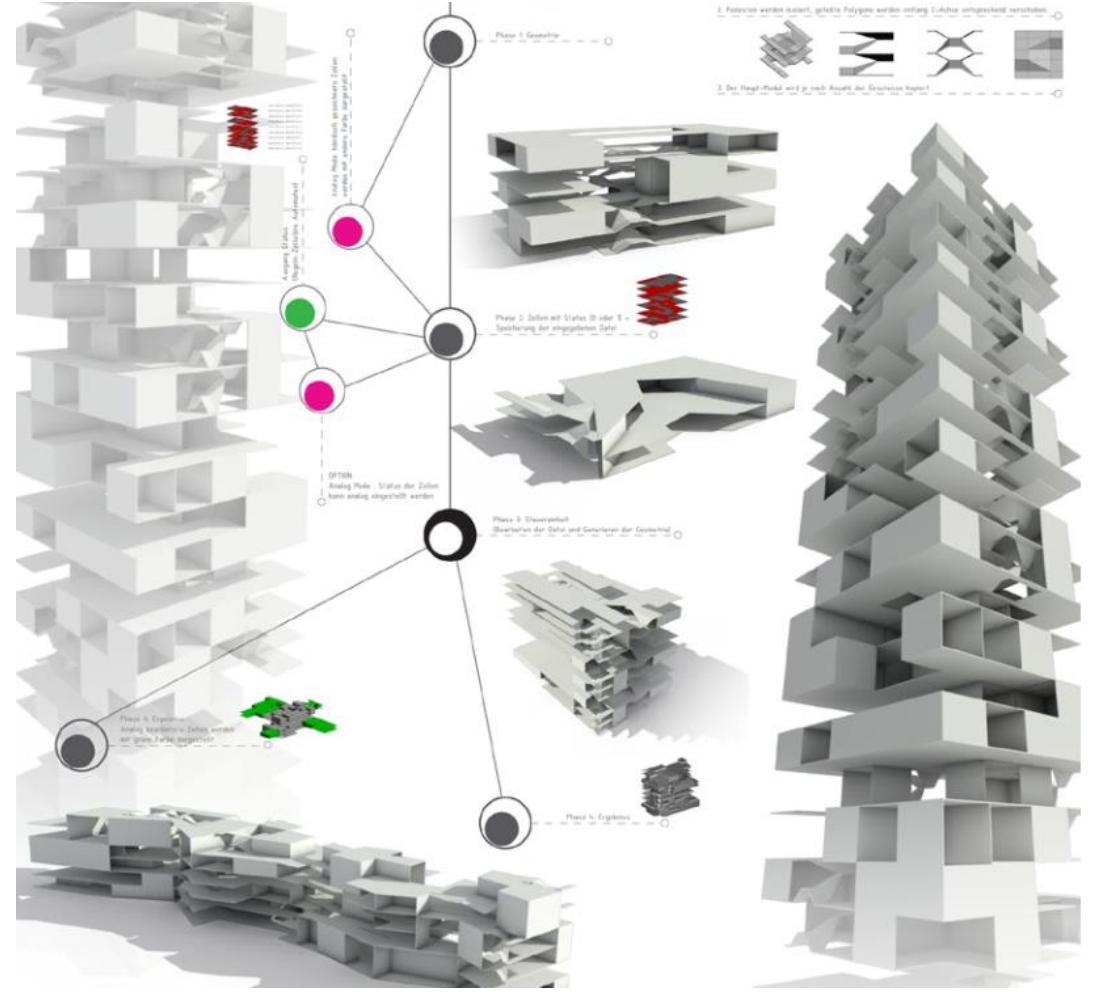
MIGROS

hps

HIRSLANDEN

vetschpartner

Scripting & Programming als Grundfähigkeit



Semesterarbeit TU Wien

**Mathematische Algorithmen für
Entwurf und Erstellung der
architektonischen räumlichen
Strukturen**

Diplomarbeit TU Wien Austrian Computer Graphic Award 2012

Best Interactive Solution

Externe Entwickler

**Python, Revit, Dynamo
Gaming, Unity**

Einführung in das Modul, Semesterplan

Programmübersicht

SW	Datum	Ort	Submodul	Bezeichnung Submodul	Thema	Inhalt	Anwesenheitspflicht	Dozent/in
SW1	16.09.2024	Online	Submodul 1	Grundlagen und Theorie	Einführung	Einführung ins Modul, Aufgabestellung Semesterprojekte Kontext Scripting vs. Programming	x	M. Rontsinsky
SW2	23.09.2024	Vor Ort	Submodul 1	Grundlagen und Theorie	Setup Werkzeuge, Wiederholung Python 1	Wiederholung Grundlagen Python 1 Werkzeuge, Entwicklungsumgebung		M. Rontsinsky
SW3	30.09.2024	Online	Submodul 1	Grundlagen und Theorie	Setup Werkzeuge, Wiederholung Python 2	Wiederholung Grundlagen Python 2 Werkzeuge, Entwicklungsumgebung, Live Coding		M. Rontsinsky
SW4	07.10.2024	Online	Submodul 1	Grundlagen und Theorie	Einführung Python Module und Bibliotheken	IfcOpenShell, Pandas, Tkinter, etc. Beispiele und Live Coding		M. Rontsinsky
SW5	14.10.2024	Vor Ort	Submodul 1	Grundlagen und Theorie	Vertiefung Parametrisierung und Geometrie	Erstellen und Manipulieren der Geometrie		N. Bukovits, A. Velasco Muro
SW6	21.10.2024	Online	Submodul 1	Grundlagen und Theorie	Vertiefung Parametrisierung und Geometrie	Erstellen und Manipulieren der Geometrie		N. Bukovits, A. Velasco Muro
SW7	28.10.2024	Online	Submodul 1	Grundlagen und Theorie	Spezialthemen und Algorithmen	Machine Learning		M. Rontsinsky, N. Bukovits, A. Velasco Muro
SW8	04.11.2024	Online	Submodul 2	Konzeption und Entwicklu	Zischenpräsentation 1		x	M. Rontsinsky, N. Bukovits, A. Velasco Muro, Martin Loucka
SW9	11.11.2024	Online	Submodul 2	Konzeption und Entwicklu	Coaching Semesterprojekte	"Get together", individuelles Coaching		M. Rontsinsky
SW10	18.11.2024	Vor Ort	Submodul 2	Konzeption und Entwicklu	Coaching Semesterprojekte	"Get together", individuelles Coaching		M. Rontsinsky, N. Bukovits, A. Velasco Muro
SW11	25.11.2024	Online	Submodul 2	Konzeption und Entwicklu	Coaching Semesterprojekte	"Get together", individuelles Coaching		M. Rontsinsky, N. Bukovits, A. Velasco Muro
SW12	02.12.2024	Online	Submodul 3	Testing und Anwendung	Coaching Semesterprojekte	"Get together", individuelles Coaching		M. Rontsinsky
SW13	09.12.2024	Online	Submodul 3	Testing und Anwendung	Zischenpräsentation 2		x	M. Rontsinsky, N. Bukovits, A. Velasco Muro, Martin Loucka
SW14	16.12.2024	Online	MEP Vorbereitung	MEP Vorbereitung	Abschluss und Vorbereitung MEP			M. Rontsinsky
MEP	tbd	Vor Ort	MEP Durchführung	MEP Durchführung	MEP			M. Rontsinsky, N. Bukovits, A. Velasco Muro, Martin Loucka

Programmübersicht

SW	Datum	Ort	Submodul	Bezeichnung Submodul	Thema	Inhalt	Anwesenheitspflicht	Dozent/in
SW1	16.09.2024	Online	Submodul 1	Grundlagen und Theorie	Einführung	Einführung ins Modul, Aufgabenstellung Semesterprojekte Kontext Scripting vs. Programming	x	M. Rontsinsky
SW2	23.09.2024	Vor Ort	Submodul 1	Grundlagen und Theorie	Setup Werkzeuge, Wiederholung Python 1	Wiederholung Grundlagen Python 1 Werkzeuge, Entwicklungsumgebung		M. Rontsinsky
SW3	30.09.2024	Online	Submodul 1	Grundlagen und Theorie	Setup Werkzeuge, Wiederholung Python 2	Wiederholung Grundlagen Python 2 Werkzeuge, Entwicklungsumgebung, Live Coding		M. Rontsinsky
SW4	07.10.2024	Online	Submodul 1	Grundlagen und Theorie	Einführung Python Module und Bibliotheken	IfcOpenShell, Pandas, Tkinter, etc. Beispiele und Live Coding		M. Rontsinsky
SW5	14.10.2024	Vor Ort	Submodul 1	Grundlagen und Theorie	Vertiefung Parametrisierung und Geometrie	Erstellen und Manipulieren der Geometrie		N. Bukovits, A. Velasco Muro
SW6	21.10.2024	Online	Submodul 1	Grundlagen und Theorie	Vertiefung Parametrisierung und Geometrie	Erstellen und Manipulieren der Geometrie		N. Bukovits, A. Velasco Muro
SW7	28.10.2024	Online	Submodul 1	Grundlagen und Theorie	Spezialthemen und Algorithmen	Machine Learning		M. Rontsinsky, N. Bukovits, A. Velasco Muro
SW8	04.11.2024	Online	Submodul 2	Konzeption und Entwicklu	Zischenpräsentation 1		x	M. Rontsinsky, N. Bukovits, A. Velasco Muro, Martin Loucka
SW9	11.11.2024	Online	Submodul 2	Konzeption und Entwicklu	Coaching Semesterprojekte	"Get together", individuelles Coaching		M. Rontsinsky
SW10	18.11.2024	Vor Ort	Submodul 2	Konzeption und Entwicklu	Coaching Semesterprojekte	"Get together", individuelles Coaching		M. Rontsinsky, N. Bukovits, A. Velasco Muro
SW11	25.11.2024	Online	Submodul 2	Konzeption und Entwicklu	Coaching Semesterprojekte	"Get together", individuelles Coaching		M. Rontsinsky, N. Bukovits, A. Velasco Muro
SW12	02.12.2024	Online	Submodul 3	Testing und Anwendung	Coaching Semesterprojekte	"Get together", individuelles Coaching		M. Rontsinsky
SW13	09.12.2024	Online	Submodul 3	Testing und Anwendung	Zischenpräsentation 2		x	M. Rontsinsky, N. Bukovits, A. Velasco Muro, Martin Loucka
SW14	16.12.2024	Online	MEP Vorbereitung	MEP Vorbereitung	Abschluss und Vorbereitung MEP			M. Rontsinsky
MEP	tbd	Vor Ort	MEP Durchführung	MEP Durchführung	MEP			M. Rontsinsky, N. Bukovits, A. Velasco Muro, Martin Loucka

SW 1 – SW 4

- Dozent/In: M. Rontsinsky, J.Jerg
- **Einrichten der Werkzeuge, Einführung in die Entwicklungsumgebung**
- **Wiederholung Python**
- Einführung in die externen Python Module
- **Auswahl Themen Semesterprojekte**

Programmübersicht

SW	Datum	Ort	Submodul	Bezeichnung Submodul	Thema	Inhalt	Anwesenheitspflicht	Dozent/in
SW1	16.09.2024	Online	Submodul 1	Grundlagen und Theorie	Einführung	Einführung ins Modul, Aufgabestellung Semesterprojekte Kontext Scripting vs. Programming	x	M. Rontsinsky
SW2	23.09.2024	Vor Ort	Submodul 1	Grundlagen und Theorie	Setup Werkzeuge, Wiederholung Python 1	Wiederholung Grundlagen Python 1 Werkzeuge, Entwicklungsumgebung		M. Rontsinsky
SW3	30.09.2024	Online	Submodul 1	Grundlagen und Theorie	Setup Werkzeuge, Wiederholung Python 2	Wiederholung Grundlagen Python 2 Werkzeuge, Entwicklungsumgebung, Live Coding		M. Rontsinsky
SW4	07.10.2024	Online	Submodul 1	Grundlagen und Theorie	Einführung Python Module und Bibliotheken	IfcOpenShell, Pandas, Tkinter, etc. Beispiele und Live Coding		M. Rontsinsky
SW5	14.10.2024	Vor Ort	Submodul 1	Grundlagen und Theorie	Vertiefung Parametrisierung und Geometrie	Erstellen und Manipulieren der Geometrie		N. Bukovits, A. Velasco Muro
SW6	21.10.2024	Online	Submodul 1	Grundlagen und Theorie	Vertiefung Parametrisierung und Geometrie	Erstellen und Manipulieren der Geometrie		N. Bukovits, A. Velasco Muro
SW7	28.10.2024	Online	Submodul 1	Grundlagen und Theorie	Spezialthemen und Algorithmen	Machine Learning		M. Rontsinsky, N. Bukovits, A. Velasco Muro
SW8	04.11.2024	Online	Submodul 2	Konzeption und Entwicklu	Zischenpräsentation 1		x	M. Rontsinsky, N. Bukovits, A. Velasco Muro, Martin Loucka
SW9	11.11.2024	Online	Submodul 2	Konzeption und Entwicklu	Coaching Semesterprojekte	"Get together", individuelles Coaching		M. Rontsinsky
SW10	18.11.2024	Vor Ort	Submodul 2	Konzeption und Entwicklu	Coaching Semesterprojekte	"Get together", individuelles Coaching		M. Rontsinsky, N. Bukovits, A. Velasco Muro
SW11	25.11.2024	Online	Submodul 2	Konzeption und Entwicklu	Coaching Semesterprojekte	"Get together", individuelles Coaching		M. Rontsinsky, N. Bukovits, A. Velasco Muro
SW12	02.12.2024	Online	Submodul 3	Testing und Anwendung	Coaching Semesterprojekte	"Get together", individuelles Coaching		M. Rontsinsky
SW13	09.12.2024	Online	Submodul 3	Testing und Anwendung	Zischenpräsentation 2		x	M. Rontsinsky, N. Bukovits, A. Velasco Muro, Martin Loucka
SW14	16.12.2024	Online	MEP Vorbereitung	MEP Vorbereitung	Abschluss und Vorbereitung MEP			M. Rontsinsky
MEP	tbd	Vor Ort	MEP Durchführung	MEP Durchführung	MEP			M. Rontsinsky, N. Bukovits, A. Velasco Muro, Martin Loucka

SW 5 – SW 8

- Dozent/In: N. Bukovits, A. Velasco Muro, M. Rontsinsky, J.Jerg
- Vertiefung im Bereich **Geometrie** und **Parametrisierung**
- **Machine Learning**
- **Zwischenpräsentation 1 + Test**

Programmübersicht

SW	Datum	Ort	Submodul	Bezeichnung Submodul	Thema	Inhalt	Anwesenheitspflicht	Dozent/in
SW1	16.09.2024	Online	Submodul 1	Grundlagen und Theorie	Einführung	Einführung ins Modul, Aufgabestellung Semesterprojekte Kontext Scripting vs. Programming	x	M. Rontsinsky
SW2	23.09.2024	Vor Ort	Submodul 1	Grundlagen und Theorie	Setup Werkzeuge, Wiederholung Python 1	Wiederholung Grundlagen Python 1 Werkzeuge, Entwicklungsumgebung		M. Rontsinsky
SW3	30.09.2024	Online	Submodul 1	Grundlagen und Theorie	Setup Werkzeuge, Wiederholung Python 2	Wiederholung Grundlagen Python 2 Werkzeuge, Entwicklungsumgebung, Live Coding		M. Rontsinsky
SW4	07.10.2024	Online	Submodul 1	Grundlagen und Theorie	Einführung Python Module und Bibliotheken	IfcOpenShell, Pandas, Tkinter, etc. Beispiele und Live Coding		M. Rontsinsky
SW5	14.10.2024	Vor Ort	Submodul 1	Grundlagen und Theorie	Vertiefung Parametrisierung und Geometrie	Erstellen und Manipulieren der Geometrie		N. Bukovits, A. Velasco Muro
SW6	21.10.2024	Online	Submodul 1	Grundlagen und Theorie	Vertiefung Parametrisierung und Geometrie	Erstellen und Manipulieren der Geometrie		N. Bukovits, A. Velasco Muro
SW7	28.10.2024	Online	Submodul 1	Grundlagen und Theorie	Spezialthemen und Algorithmen	Machine Learning		M. Rontsinsky, N. Bukovits, A. Velasco Muro
SW8	04.11.2024	Online	Submodul 2	Konzeption und Entwicklu	Zischenpräsentation 1		x	M. Rontsinsky, N. Bukovits, A. Velasco Muro, Martin Loucka
SW9	11.11.2024	Online	Submodul 2	Konzeption und Entwicklu	Coaching Semesterprojekte	"Get together", individuelles Coaching		M. Rontsinsky
SW10	18.11.2024	Vor Ort	Submodul 2	Konzeption und Entwicklu	Coaching Semesterprojekte	"Get together", individuelles Coaching		M. Rontsinsky, N. Bukovits, A. Velasco Muro
SW11	25.11.2024	Online	Submodul 2	Konzeption und Entwicklu	Coaching Semesterprojekte	"Get together", individuelles Coaching		M. Rontsinsky, N. Bukovits, A. Velasco Muro
SW12	02.12.2024	Online	Submodul 3	Testing und Anwendung	Coaching Semesterprojekte	"Get together", individuelles Coaching		M. Rontsinsky
SW13	09.12.2024	Online	Submodul 3	Testing und Anwendung	Zischenpräsentation 2		x	M. Rontsinsky, N. Bukovits, A. Velasco Muro, Martin Loucka
SW14	16.12.2024	Online	MEP Vorbereitung	MEP Vorbereitung	Abschluss und Vorbereitung MEP			M. Rontsinsky
MEP	tbd	Vor Ort	MEP Durchführung	MEP Durchführung	MEP			M. Rontsinsky, N. Bukovits, A. Velasco Muro, Martin Loucka

SW 9 – SW 14

- Dozent/In: N. Bukovits, A. Velasco Muro, M. Rontsinsky, J.Jerg
- Arbeit an den **Semesterprojekten** inkl. Coaching
- **Zwischenpräsentation 2**

Semesterleistung und Modulendprüfung

Modulleistung

3 ECTS = 90 Arbeitsstunden

Davon ca. **34h** Kontaktunterricht (ca. 2.4 h x 14 Wochen)

ca. **12h** MEP und Prüfungsvorbereitung

ca. **44h** Selbststudium (bzw. Geführtes Selbststudium)

Unterricht bzw. Selbststudium

	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8	SW9	SW10	SW11	SW12	SW13	SW14	MEP
Kontaktunterricht	2.4 h												
Selbststudium															12h

Semesterleistung bzw. Notenverteilung

	Grundlagen und Theorie					Konzeption und Entwicklung					Testing und Anwendung					MEP
Leistungsnachweis																
Notenaufteilung						15%								15%		70%

MEP

Ausgangslage:

Das Ziel des Moduls ist, dass die Studierende im Laufe des Semesters **eigene, voll funktionierende Programme** für **konkrete Zielsetzungen** selbstständig konzipieren und entwickeln können. Für die Entwicklung wird entsprechende **Entwicklungsumgebung** eingerichtet, dabei wird mit den Systemen für die **Versionskontrolle** gearbeitet (**GitHub**) und als Entwicklungssprache wird **ausschliesslich Python** und seine Module verwendet. Die **Zwischenprüfungen** werden als **Begleitung und Vorbereitung** für die Entwicklung der Programme gesehen. **Finale Programme** werden zum Schluss bei der **MEP präsentiert** und vorgestellt.

Aufgabenstellung – Themen/Kontext:

Die Studierende arbeiten an den Projekten **individuell** oder in max. **Zweiergruppen**. Die Themen und Inhalte für die Projekte sollten durch die Studierenden selbstständig konzipiert werden, sie sollten aber in ein der folgenden **Kontexte eingeordnet** werden:

- **Datenauswertung**
- **«IFC Editor»**
- **Einfaches «Model Checker/Qualifier»**
- **Daten-Visualisierung (Dashboards)**
- **Calculator (Wettbewerb, Kosten, Nachhaltigkeit, Kreislaufwirtschaft, etc.)**
- **Individuell (nach Absprache mit MV)**

Aufgabenstellung – Python Module:

Bei der Entwicklung der Programme sollten folgende Python-Module eingesetzt werden (mindestens ein externes Python Modul muss verwendet werden):

- ***IfcOpenShell***
- ***Pandas***
- ***Matplotlib***
- ***SQLite Datenbank***
- ***TkInter (Modul für Benutzeroberfläche - UI)***

Aufgabenstellung – Entwicklungsumgebung:

Die Entwicklung der Programme erfolgt in sog. Entwicklungsumgebung, welche Studierende für eigene Projekte selbstständig einrichten. Dabei müssen folgende Bestandteile berücksichtigt werden:

- ***Git und GitHub (Versionskontrolle)***
- ***Conda, Python Venv (Verwalten der lokalen Umgebungen und installierten Module)***
- ***Python (Als Programmiersprache)***
- ***Visual Studio Code oder PyCharm (Editor nach Wahl)***

Abgabe und Ablauf

5.3 Schwerpunkte und Bewertungskriterien

- **Innovation und Selbständigkeit:** Neue Ansätze, selbständige Konzeption, Lösungsfindung und Umsetzung
- **Funktionalität:** Reifegrad und Vollständigkeit der Programme.
- **Werkzeuge und Entwicklungsumgebung:** Eingesetzte Werkzeuge und Bibliotheken, Einrichtung der Entwicklungsumgebung.

5.4 Prüfungsform und Ablauf

Die MEP wird wie folgt durchgeführt:

- **mündliche Prüfung**, individuell oder in den Gruppen mit max. 2 Personen
- ca. **10-15 Minuten Präsentation**, ca. **5-10 Minuten Fragenbeantwortung**
- alle Gruppenmitglieder müssen in etwa gleiche Anteile präsentieren
- Die Frage werden an einzelne Gruppenmitglieder adressiert

5.5 Abgabe

- es kann eine **separate Dokumentation** und eine **Präsentation** oder nur eine (umfassende) Präsentation als PDF abgegeben werden. Optional kann auch ein **Video zur Dokumentation** der Funktionalität und Anwendung des fertigen Programmes abgegeben werden.
- Für das A3-Semesterblatt wird eine Vorlage auf Ilias zur Verfügung gestellt.
- **Vollständiges Programm auf Plattform GitHub.** Das Repository mit dem Programm muss öffentlich sein. Bereitstellen des Zugangs erfolgt durch Angabe des Links zum Repository in der Dokumentation.

Modulleistung - Bewertungskriterien

3. Bewertungskriterien

Notenanteil

3.1 Bewertungskriterien Semesterleitung

	Notenanteil
Zwischenpräsentation 1: Vorstellung Konzept des Semesterprojektes Python Grundlagen: Test zur Überprüfung der Grundkenntnisse von Python und Theorie aus dem ersten Submodul durchgeführt als Multiple-Choice-Test über ILIAS.	15%
Zwischenpräsentation 2: Vorstellung des Umsetzungsstandes des Semesterprojektes	15%
Total-Semesterleistung	30%

Notenanteil

3.2 Bewertungskriterien MEP

Innovation und Selbständigkeit: Innovationsgrad des Semesterprojektes (Thema, Umsetzungsansatz), Anwendungsbereich des Programmes.	10%
Funktionalität: Vollständigkeit, Durchgängigkeit und Implementierung der finalen Lösung. Struktur des Programmes insbesondere der modulare Aufbau des Programmes: Hauptkomponente, Hilfsfunktionen und Klassen, Benutzeroberfläche (falls vorhanden) etc. Erläuterung der verwendeten Entwicklungsumgebung (IDE): Versionskontrolle (Git), Nutzung der Python Module, Editor (z.B. Visual Studio Code) etc.	30%
Dokumentation: Aufbau, Struktur, Inhalt der Dokumentation. Finale Anwendungen/Programme sind über Plattform Github abzugeben.	20%
Präsentation: Aufbau, Struktur und visuelle Aufbereitung der Präsentation als auch mündliche Vorstellung und Präsentationstechnik.	10%
Total-MEP	70%

Zwischenprüfung - 1

Zwischenprüfung 1: Grundlagen und Konzeption Semesterprojekte

Datum:	Mo 04.11.2024, 18:30-20:55
Verantwortlicher:	Michal Rontsinsky
Form:	Mündlich, Präsentationen in Gruppen. Schriftlich, Multiple Choice Test
Aufgabe:	Vorstellung der Ideen und Konzepte der Semesterprojekte als kurze Präsentation mit dem Berücksichtigen der folgenden Inhaltspunkte: <ul style="list-style-type: none">• Vorstellen des Themas und Zielsetzung für das Semesterprojekt (muss nicht abschliessend sein)• Entwicklungsumgebung, Module und Werkzeuge, die eingesetzt werden• Herausforderungen und nächste Schritte Test zur Überprüfung des Wissens aus dem ersten Submodul (Python, theoretische Grundlagen Programmierung) als Multiple Choice Test über ILIAS.

Zwischenprüfung - 2

Zwischenprüfung 2: Entwicklung und Anwendung

Datum:	Mo 09.12.2024, 18:30-20:55
Verantwortlicher:	Michal Rontsinsky
Form:	Mündlich, Präsentationen in Gruppen
Aufgabe:	<p>Vorstellung der fast finalen Prototypen der Semesterprojekte. Folgende Themen sollte in dieser Phase geklärt und vorgestellt werden:</p> <ul style="list-style-type: none">• Vorstellen des Themas und Zielsetzung für das Semesterprojekt als finale Version• Benutzbare und funktionierende Variante des Programmes• Gewisse Funktionen/Features können noch in der Arbeitsversion vorliegen

Was ist Python, warum Python

Warum Python?

I use Python to arrange, analyze, and visualize data. There are so many different software libraries to work with in Python, such as matplotlib to create charts, pandas to sort and filter data and search for trends, and BeautifulSoup for web scraping, I feel that knowing Python has given me a versatile skill set.

With Python, Amanda is able to answer questions like: “[Which Supreme Court Justice is the funniest?](#)” For this, she wrote a Python script to count the number of times the word “laughter” appeared in the Supreme Court oral arguments.

Even though the Justice responsible for the most “laughter” (Justice Breyer) didn’t necessarily have the funniest jokes, her article at Law360 did have the nerdiest take on the story compared to other news outlets.

Final Thoughts

Python is no longer just for computer programmers. At [One Month](#), we’ve noticed an increased demand for our Python course from financial analysts, entrepreneurs, sales, and marketing professionals. We’ve even had a few lawyers! Looking forward it’s clear: **Python is here to stay.**

<https://learn.onemonth.com/6-jobs-you-can-land-if-you-learn-python/>

Top Reasons to Learn Python

Python programming is powering the global job market because the benefits of Python are clear. Python is one of the top three programming languages in the world, and is poised to become the most popular, according to [ZDNet](#). In fact, according to the [PYPL Index](#), Python is the most popular programming language world-wide, so if you want to work in a different country, you have a good chance of landing a job in, say, Switzerland or Australia. Where would you like to work? Adding [Python](#) to your skill set could be your ticket to anywhere.

Also, Python programming is a general-purpose skill used in almost all fields, including:

- Data science
- Scientific and mathematical computing
- Web development
- Finance and trading
- System automation and administration
- Computer graphics
- Basic game development
- Security and penetration testing
- General and application-specific scripting
- Mapping and geography (GIS software)

Thus, if you learn Python, you’ll have more job opportunities than you can imagine – at home or abroad. What field do you prefer? [Data science](#) is all the rage, but options like game development, finance and trading, security and penetration testing, and network mapping and geography are other potentially exciting possibilities. It’s your choice how you leverage the benefits of Python.

<https://www.simplilearn.com/why-learn-python-a-guide-to-unlock-your-python-career-article>

Geschichte von Python

2.1.1 Geschichte und Entstehung ▲▼

Die Programmiersprache Python wurde Anfang der 1990er-Jahre von dem Niederländer Guido van Rossum am Centrum voor Wiskunde en Informatica (CWI) in Amsterdam entwickelt. Ursprünglich war sie als Skriptsprache für das verteilte Betriebssystem Amoeba gedacht. Der Name Python lehnt sich nicht etwa an die Schlangenfamilie an, sondern ist eine Hommage an die britische Komikertruppe Monty Python.

Vor Python hatte van Rossum an der Entwicklung der Programmiersprache ABC mitgewirkt, die mit dem Ziel entwickelt wurde, so einfach zu sein, dass sie problemlos einem interessierten Laien ohne Programmiererfahrung beigebracht werden kann. Die Erfahrung aus positiver und negativer Kritik an ABC nutzte van Rossum für die Entwicklung von Python. Er schuf damit eine Programmiersprache, die mächtig und zugleich leicht zu erlernen ist.

Mit der Version 3.0, die im Dezember 2008 erschien, wurde die Sprache von Grund auf überarbeitet. Dabei sind viele kleine Unschönheiten und Designfehler beseitigt worden, die man in bisherigen Versionen aufgrund der Abwärtskompatibilität stets in der Sprache behalten musste.

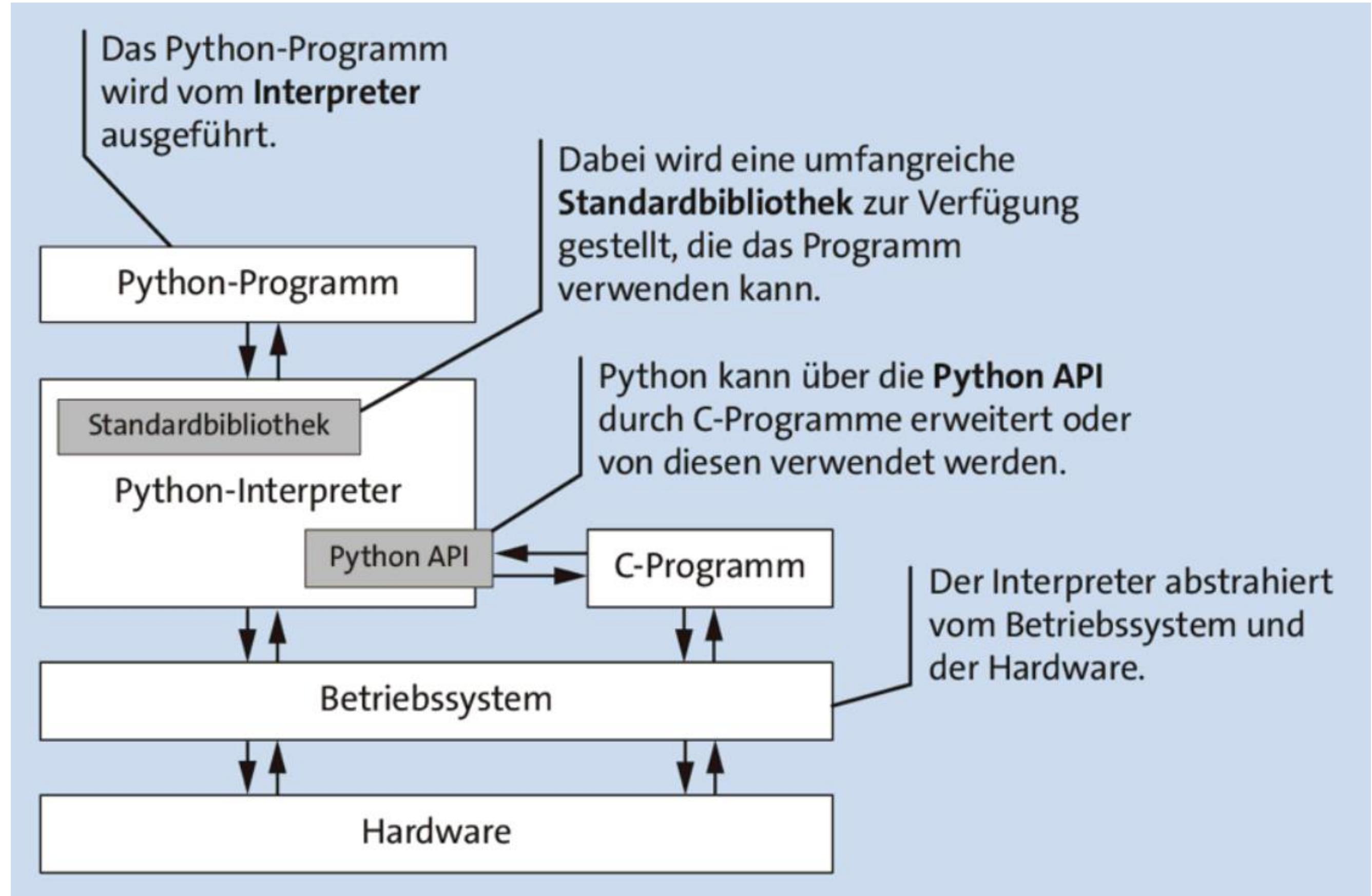
Mittlerweile hat sich Python zu einer der beliebtesten Programmiersprachen ihres Typs entwickelt und nimmt bei Popularitätsindizes von Programmiersprachen[1](zum Beispiel TIOBE, RedMonk oder PYPL) regelmäßig Spitzenpositionen ein.

Seit 2001 existiert die gemeinnützige Python Software Foundation, die die Rechte am Python-Code besitzt und Lobbyarbeit für Python betreibt. So organisiert die Python Software Foundation beispielsweise die PyCon-Konferenz, die jährlich in den USA stattfindet. Auch in Europa finden regelmäßig größere und kleinere Python-Konferenzen statt.



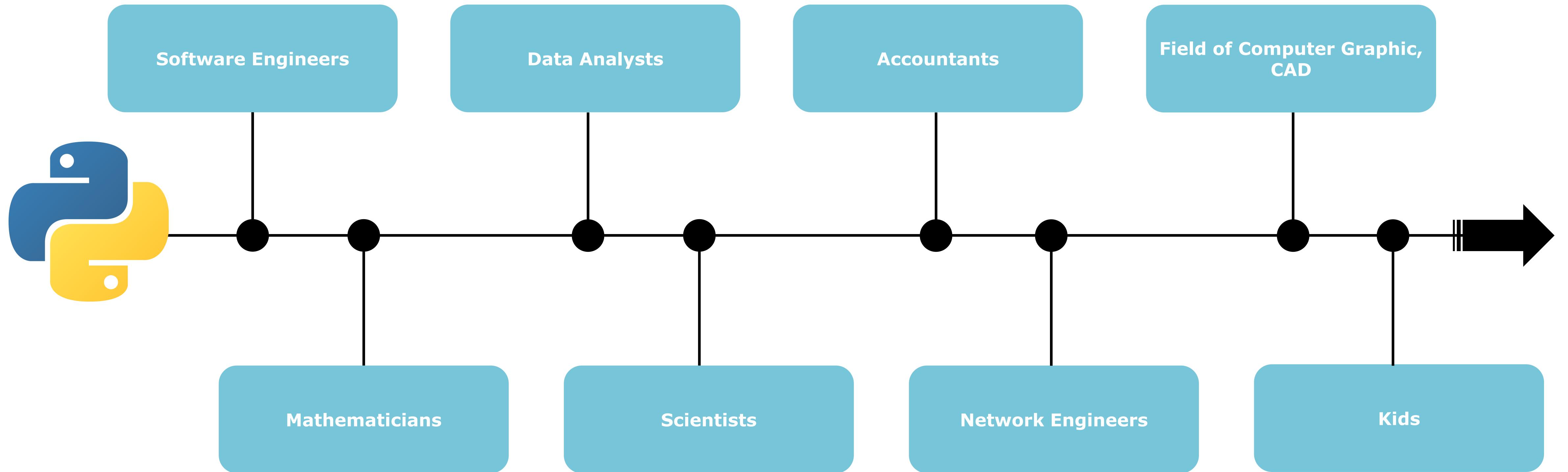
https://openbook.rheinwerk-verlag.de/python/02_001.html

Grundlagen Konzept von Python

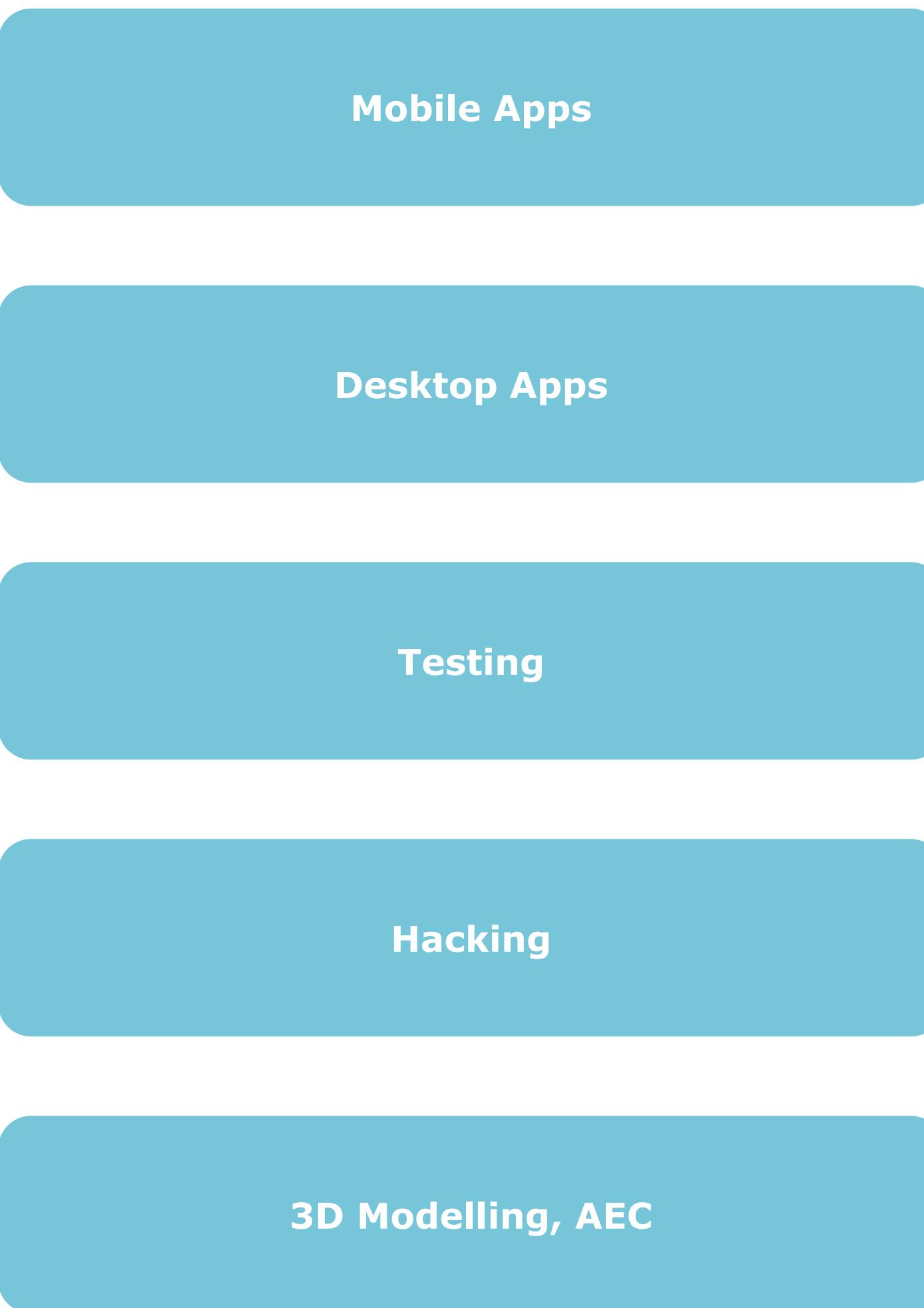
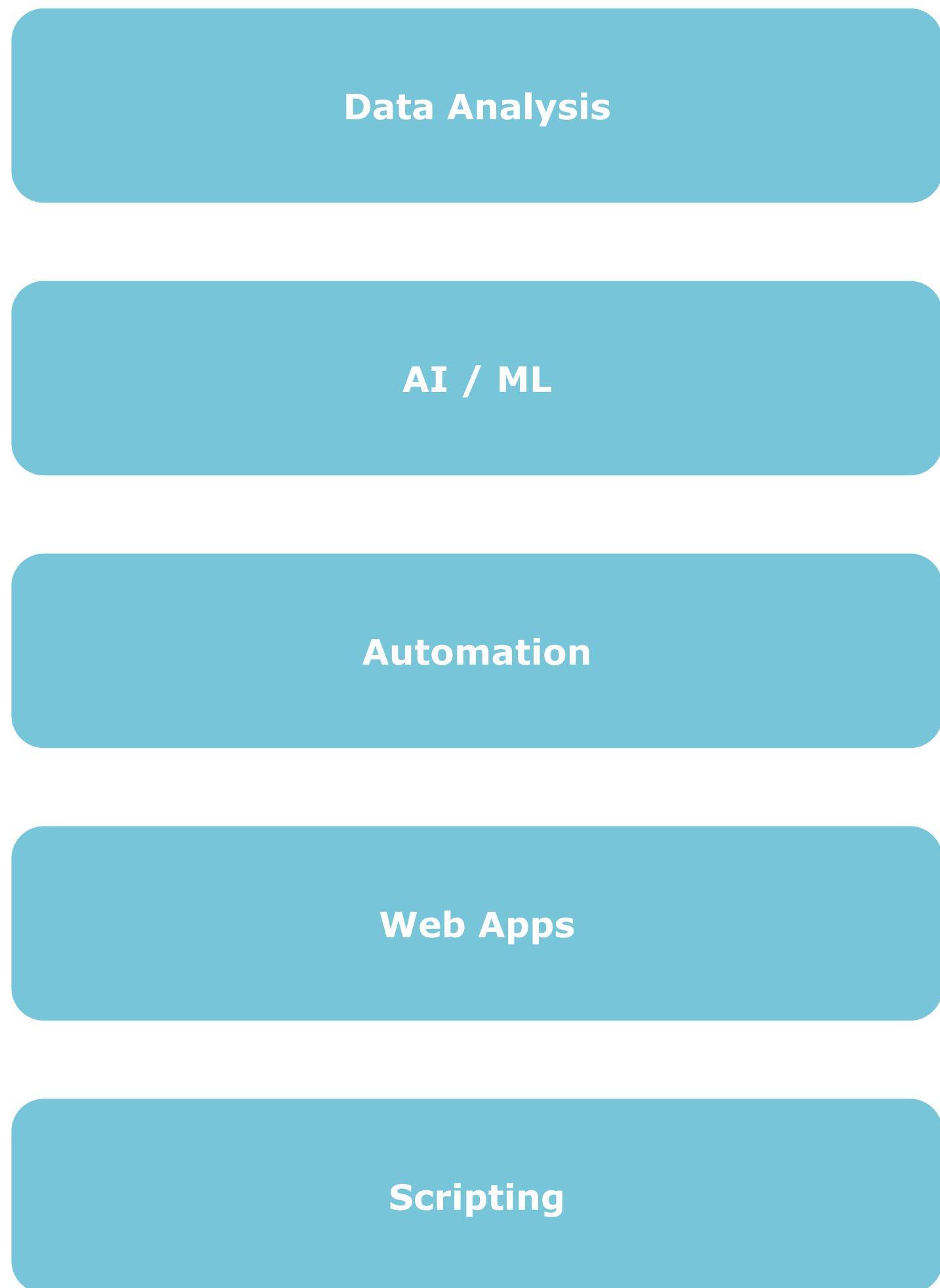


https://openbook.rheinwerk-verlag.de/python/02_001.html

Warum Python – Einsatzgebiete Branchen



Warum Python – Einsatzgebiete Tätigkeiten



Warum Python – Einfachheit, schneller Einstieg

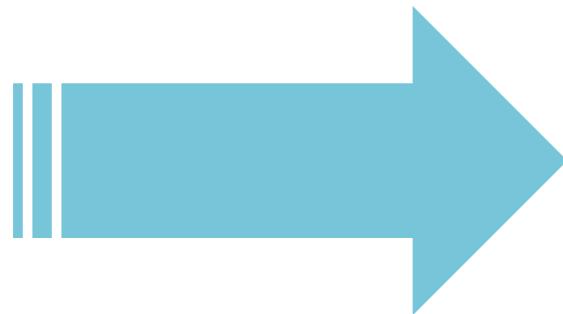
Solve complex problems in **less time** with **fewer lines of code**

PYTHON >>

Moderne Scriptsprache, einfache Syntax, weniger Aufwand bei dem Coding

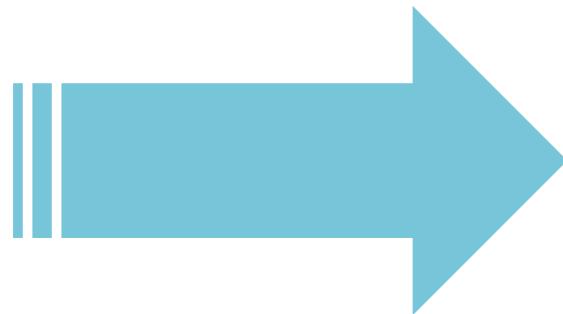
Warum Python - Einfachheit

C#



str.Substring(0, 3)

JAVASCRIPT



str.substr(0, 3)

PYTHON



str[0:3]

Warum Python – Andere Vorteile

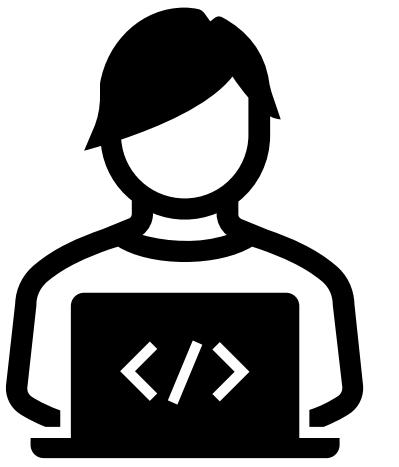


Einführung Programmierung

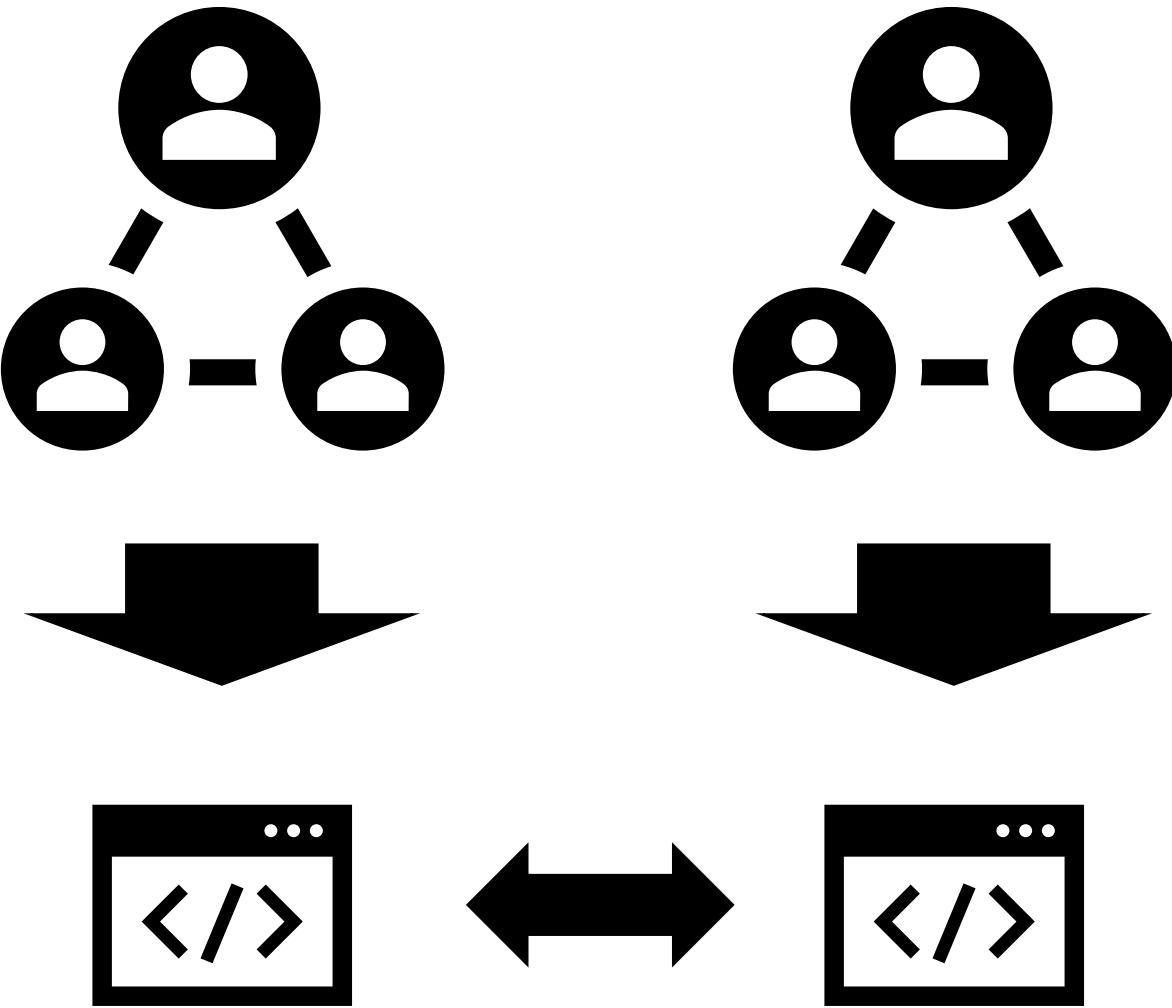
Scripting vs. Programming

Scripting vs. Programming

Scripting



Programming



Scripting vs. Programming

Scripting



Programming

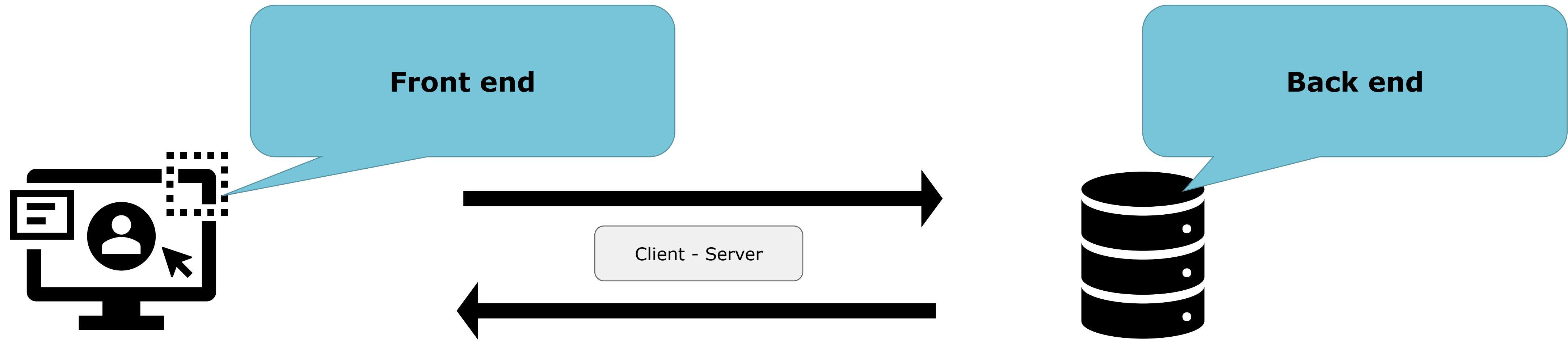
- ▶ Kleine Anwendungen («Makros»)
- ▶ Anwendbar für diverse Aufgaben:
 - Automatisierung, Datenverarbeitung, Design
 - und Erstellend er Geometrie, Datenaustausch
 - und Datenverarbeitung, Schnittstellen

- ▶ Entwicklung der selbständigen Anwendungen
 - innerhalb von anderen Plattformen/Umgebungen
 - oder komplett selbständig
- ▶ Entwicklung der Daten- und UI-Komponente :
 - Front und Back End
- ▶ Low Level Code Programming (z.B. Power Apps,
 - Azure Data Lake, Databricks etc.)

Einführung Programmierung

Full stack Development

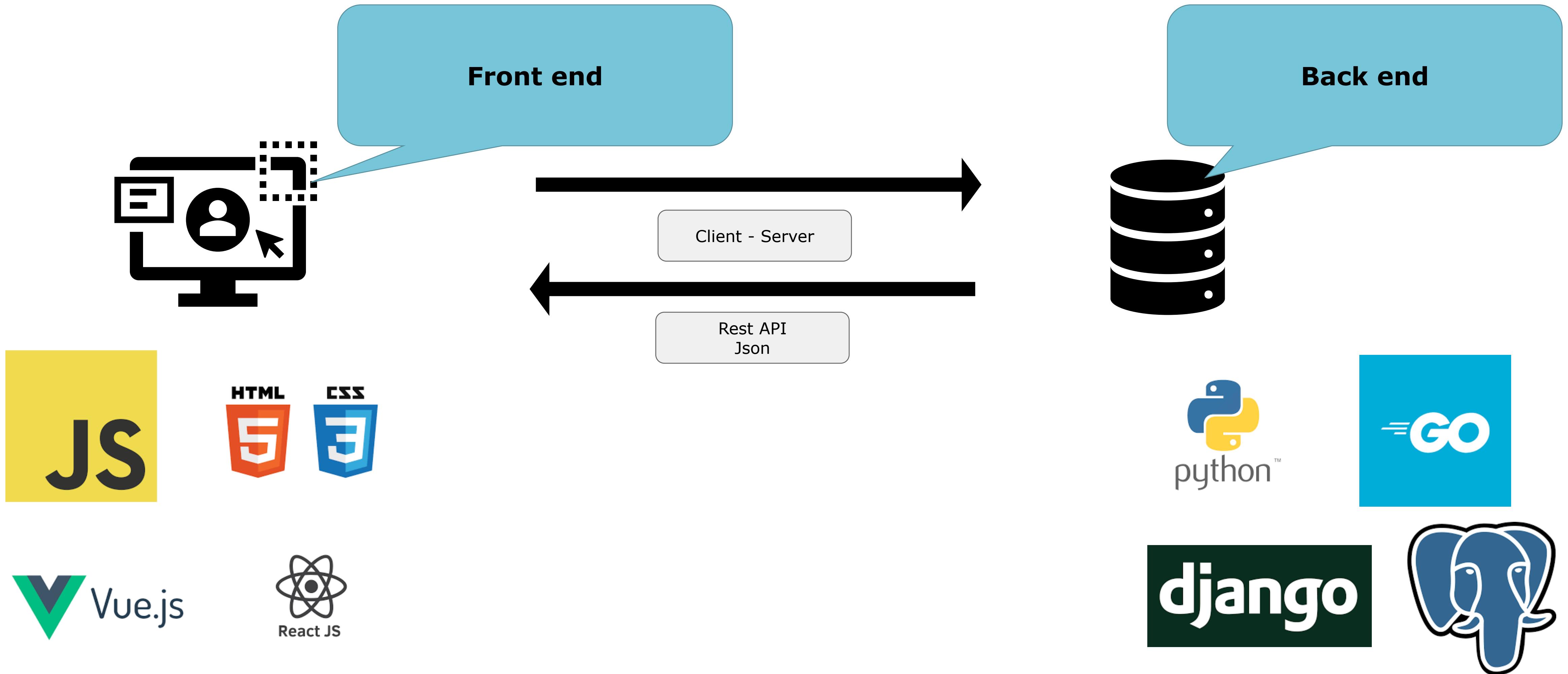
Full stack Development - Erläuterung



- ▶ Visuelle Oberfläche der Anwendung
- ▶ Eingabe und Interaktion mit dem Anwender
- ▶ Single-Page-Anwendungen werden als Gesamtpaket in Browser geladen und dort in der Laufzeitumgebung berechnet (z.B. React, Vue, Angular, etc.)

- ▶ Grundlegende Daten der Anwendung und Hosting
- ▶ Admin-Daten und Daten der Anwendung
- ▶ Bereitstellen der Daten (API Framework)

Full stack Development – Sprachen und Frameworks



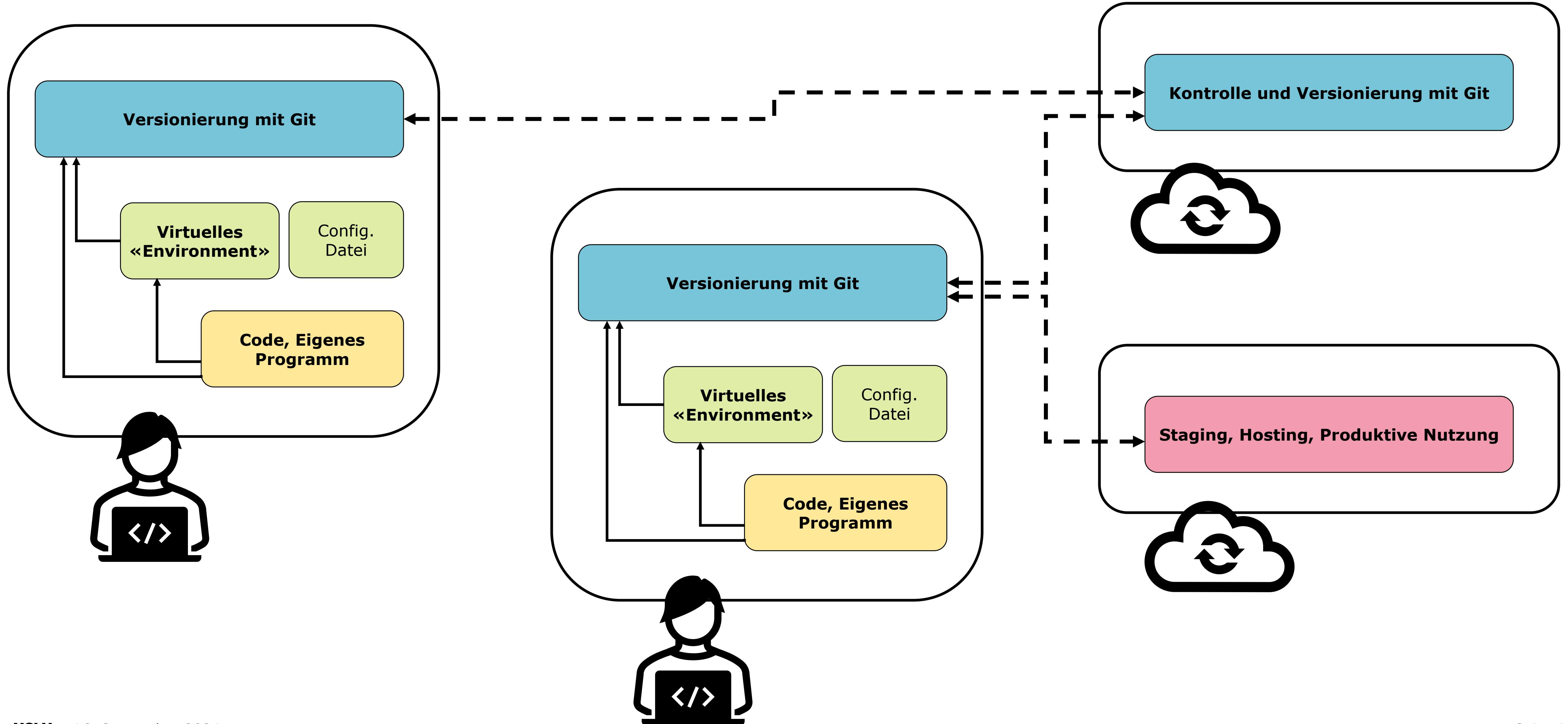
Einführung Programmierung

Werkzeuge im Kontext

Werkzeuge im Kontext – Setup des Projektes und der Entwicklungs-Umgebung

- ▶ **Code Editor**
- ▶ **Git** – Kontrollsystem für die Versionierung und Teilen des Codes und Projektstruktur
- ▶ **Entwicklungsumgebung für Projekt** (Environment - Package Management)
- ▶ **Modularer Aufbau des Programmes** (Ordnerstruktur, Module und Referenzieren der Komponente)
- ▶ **Distribution, Hosting** (Heroku, Docker, Digital Ocean)

Werkzeuge im Kontext – Gesamtübersicht



Werkzeuge im Kontext – Gitlab

The screenshot shows a GitLab project interface. On the left, a sidebar lists project settings like 'Project information', 'Repository', 'Issues', 'Merge requests', 'CI/CD', 'Security and Compliance', 'Deployments', 'Packages and registries', 'Infrastructure', 'Monitor', 'Analytics', 'Wiki', 'Snippets', and 'Settings'. The main area displays a 'Projectübersicht' (Project Overview) with statistics: 51 Commits, 5 Branches, 0 Tags, and 369 KB Project Storage. Below this is a 'refactoring' commit by Sven Zbinden. A central commit history table shows various commits from different users. A large blue callout bubble labeled 'Projektübersicht in Cloud' points to the top right of the project stats. Another blue callout bubble labeled 'Versionsverlauf, Branches' points to the commit history table.

Projektübersicht in Cloud

Versionsverlauf, Branches

Backend_Group3

51 Commits 5 Branches 0 Tags 369 KB Project Storage

refactoring Sven Zbinden authored 24 minutes ago

master backend_group3 / +

Add README Add LICENSE Add CHANGELOG Add CONTRIBUTING Add Auto DevOps Add Kubernetes cluster Add CI/CD Add Wiki

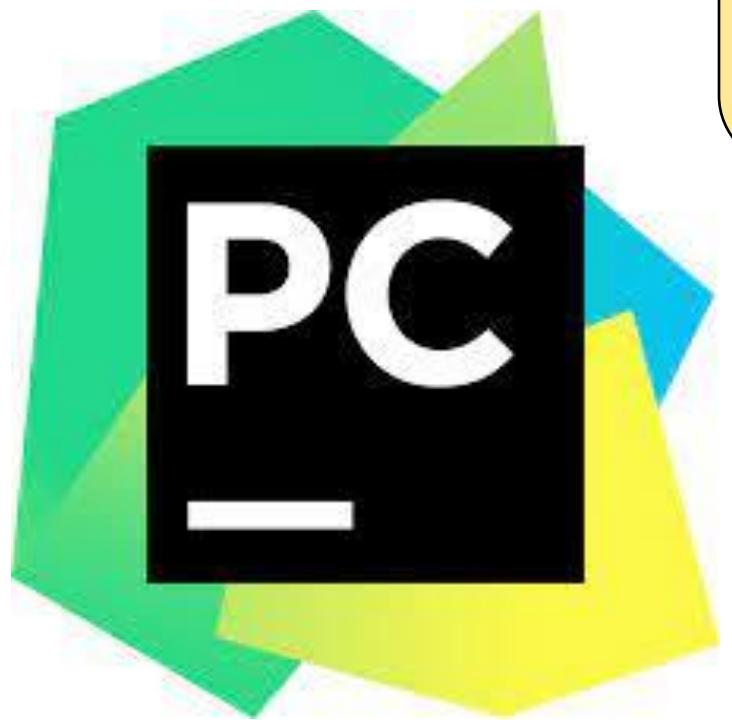
Configure Integrations

Name	Last commit	Last update
comment	fixed the comment connection to post	2 days ago
djangoUser	refactoring	24 minutes ago
friend	merged with master	3 days ago
image	SharedPost and Image	2 days ago
post	Url update	1 day ago
project	fix of deployment	2 hours ago
registrationProfile	fixed validation	1 day ago
sharedPost	SharedPost and Image	2 days ago
thingsUserLike	merged Swagger	10 hours ago
uml	Initial commit	1 week ago
userProfile	merged Swagger	10 hours ago
.gitignore	Initial commit	1 week ago
Procfile	fix deployment	8 hours ago
manage.py	start deployment	10 hours ago
requirements.txt	added heroku-url to allowed hosts	8 hours ago
requirements.yml	setup heroku	9 hours ago

Url update
SharedPost and Image
SharedPost and Image
SharedPost and Image
fixed the comment connection to post
added your of registration profile
view for post
post with views
things User Likes Retrieve
added represtation of DjangoUser to UserProfile
Delete User --> DjangoUser + RegistProfile
fixed missing migration in post
merged with master
view for UserProfile
Merge branch 'master' of https://gitlab.propulsion-home.ch/full-stack/batch3
fixed cirucular import on post
Merge branch 'master' of https://gitlab.propulsion-home.ch/full-stack/batch3
post view start
fixed cirucular import on post
Merge branch 'master' into 'master'
post, shareage models
origin/test_signals
origin/post

MrBeyondBIM 27.06.2023 19
MrBeyondBIM 27.06.2023 19
MrBeyondBIM 27.06.2023 19
MrBeyondBIM 27.06.2023 19
SvenZ 27.06.2023 18:5
SvenZ 27.06.2023 18:1
MrBeyondBIM 27.06.2023 18
MrBeyondBIM 27.06.2023 18
SvenZ 27.06.2023 09:2
SvenZ 26.06.2023 12:1
SvenZ 26.06.2023 10:5
SvenZ 26.06.2023 10:0
SvenZ 26.06.2023 09:3
SvenZ 24.06.2023 22:4
MrBeyondBIM 24.06.2023 20
SvenZ 24.06.2023 19:5
MrBeyondBIM 24.06.2023 19
SvenZ 24.06.2023 19:3
Sven Zbinden 24.06.2023 18:4
MrBeyondBIM 24.06.2023 14

Werkzeuge im Kontext – Coder Editor



Projektstruktur



Code

The screenshot shows a Coder Editor interface with several tabs open, including `post\views.py`, `image\views.py`, `image\models.py`, `post\models.py`, `image\urls.py`, `post\urls.py`, `apps.py`, and `project"urls.py`. The `post\views.py` tab is active, displaying code for a `ListCreatePost` view. The code includes methods for `get` and `post`, which handle returning all posts or creating a new post instance respectively. The `get` method uses a queryset of all posts and a serializer class of `PostSerializer`. The `post` method gets the serializer, checks if it's valid, and prints the request data. Below the code editor is a terminal window showing a git pull command from a GitLab repository. To the right of the code editor is a database browser window titled "Database" showing the structure of a PostgreSQL database named "postgres" on "localhost". The database contains a public schema with various objects like access methods, casts, extensions, languages, and virtual views.

Versionskontrolle

Werkzeuge im Kontext – Modularer Aufbau des Programmes

The screenshot shows a Python Django project structure and a code editor displaying a file named `views.py`.

Project Structure:

- backend_group3_m
- Project
- backend_group3_m
- commer
- djangoUs
- friend
- image
- post
 - migrations
 - _init_.py
 - admin.py
 - apps.py
 - models.py
 - serializers.py
 - tests.py
 - urls.py
 - views.py
- project
- registrationProfile
- sharedPost
 - static
- thingsUserLike
- uml
- userProfile
 - .env
 - .gitignore
- manage.py
- Procfile
- requirements.txt
- requirements.yml
- runtime.txt

Code Editor (views.py):

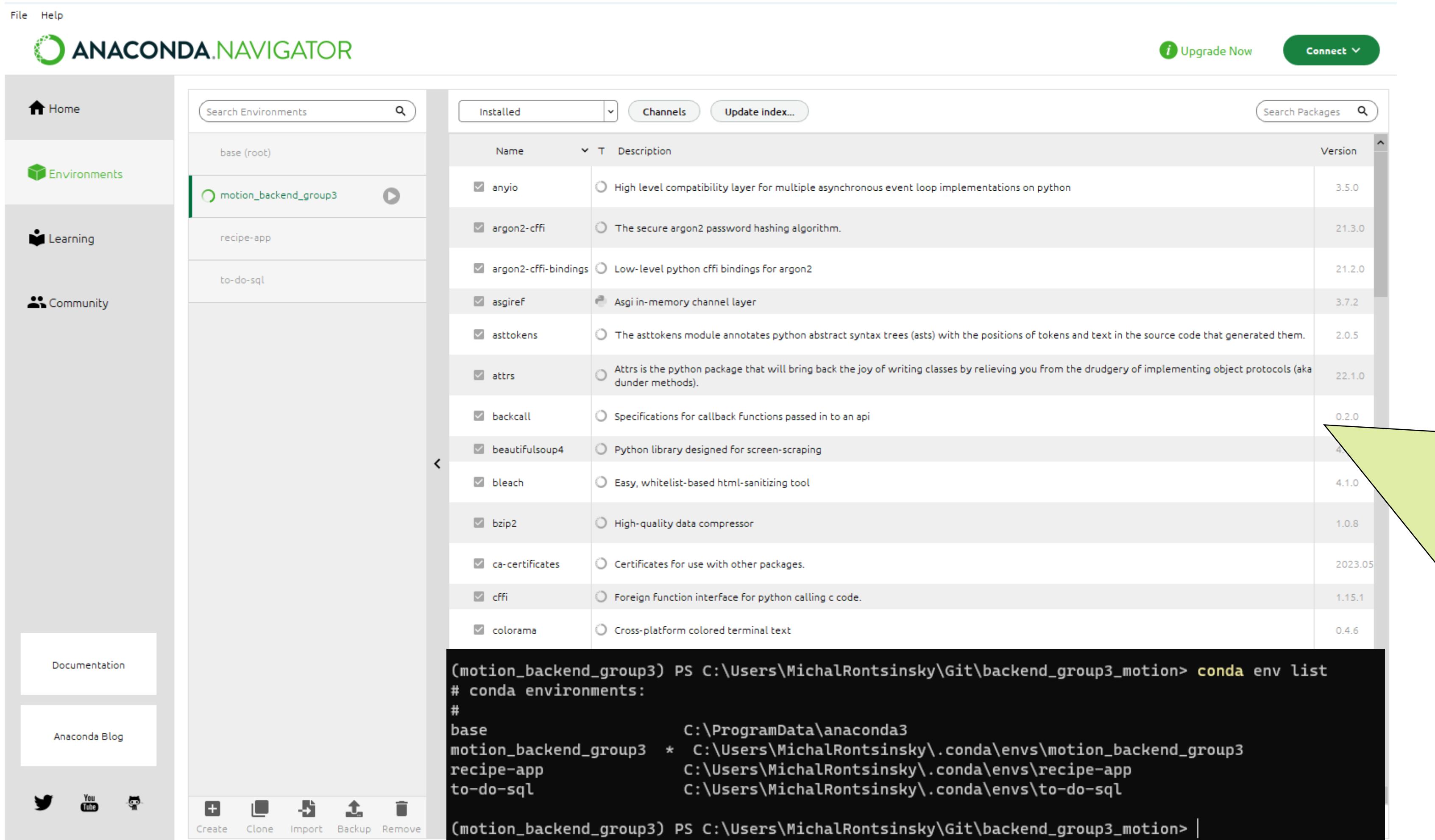
```
up3_motion - post\views.py
image\views.py × image\models.py × post\models.py × image\urls.py ×
shortcuts import render
views here.

4
5     from rest_framework import filters, status
6     from rest_framework.generics import GenericAPIView
7
8     from rest_framework.response import Response
9     from rest_framework.permissions import IsAuthenticated
10
11    from post.models import Post
12    from post.serializers import PostSerializer
13
14
15 2 usages ▲ MrBeyondBIM
parameters is given
all return all post that contain it
ns it
queryset = Post.objects.all()
serializer_class = PostSerializer
```

Annotations:

- Projektstruktur, Scripte mit unterschiedlichen Funktionen und Aufgaben.** Points to the project structure and the `views.py` file.
- Hauptscript (Ausführen der Anwendung)** Points to the `manage.py` file.
- Referenzieren der Module** (Notwendig für die Anwendung im konkreten File) Points to the `from ... import ...` statements in the code.

Werkzeuge im Kontext – Entwicklungsumgebung für Projekt



«Virtuelle Maschine oder Umgebung»

Alle Module und Pakete sind nicht auf dem Computer installiert, sondern in einem virtuellen geschlossenen «Container». Dieser Container kann flexibel zwischen unterschiedlichen Systemen transferiert werden.

Populärste Tools für Management der virtuellen Umgebungen:
VENV und **CONDA**

Projektbeispiel

Werkzeuge und ChatGPT

Grundlagen und Weiterbildung

Projektbeispiel – Live Beispiel

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The title bar reads "LakeHub_Quantity". The left sidebar has an "EXPLORER" view showing a folder "LAKEHUB_QUANTITY" containing "Helpers", "Processors", and "main.py". The main editor area displays the "main.py" file:

```
File Edit Selection View Go Run Terminal Help ← → ⌂ LakeHub_Quantity
EXPLORER ... LAKEHUB_QUANTITY
Helpers Processors main.py
Welcome main.py > start_func
main.py quantity_processor.py config_processor.py Data_Helpers.py
1 import Processors.config_processor as config_p
2 import Processors.quantity_processor as quant_p
3
4
5 #Globale Variablen - Listen
6 glb_conf_lst = []
7
8 def start_func():
9     #####
10    #####
11    ####>>>MAIN PROGRAM START HERE<<#####
12    #####
13    #####
14    #####
15
16    glb_conf_set = config_p.read_source_data()
17    quant_p.compute_quantities(glb_conf_set)
18
19
20
21
22 if __name__ == '__main__':
23     start_func()
```

The bottom right corner of the editor shows a small preview of the code's execution results. The bottom navigation bar includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, showing command-line output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Compute Quantities >>>
Type: WAN_WIW_D_150 150
Compute Quantities >>>
PS C:\Users\MichalRontsinsky\OneDrive - beyondBIM\Dokumente\VS_Projects\LakeHub_Quantity>
* History restored
```

The status bar at the bottom indicates the terminal is running Python 3.11.3 ('cas-hslu': conda).

Werkzeuge - ChatGPT

Erleben Sie eine frühe Vorschau von ChatGPT in Ihrer Sprache. Alpha beitreten X

Default (GPT-3.5)

```
import ifcopenshell

# Load the IFC file
ifc_file = ifcopenshell.open("your_ifc_file.ifc")

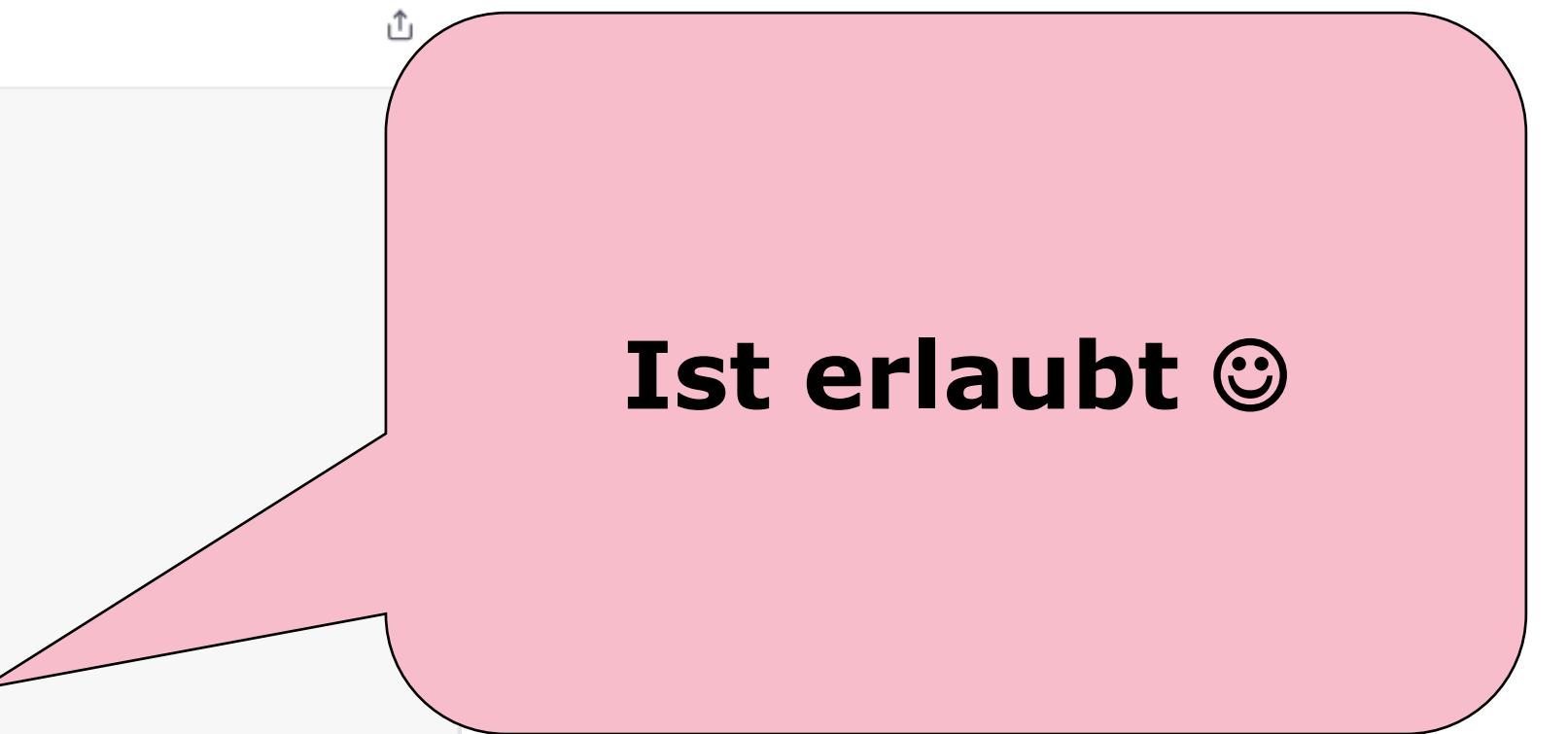
# Iterate through IfcBuildingElementPart elements
for element in ifc_file.by_type("IfcBuildingElementPart"):
    # Check the IsDecomposedBy relationship
    if element.is_a("IfcBuildingElementPart"):
        decomposition = element.IsDecomposedBy[0] if element.IsDecomposedBy
        if decomposition:
            # Traverse the decomposition relationship to find the IfcWall
            wall = decomposition.RelatedObjects[0] if decomposition.RelatedO
            if wall and wall.is_a("IfcWall"):
                # You've found the IfcWall to which the IfcBuildingElementPa
                print(f"The IfcBuildingElementPart '{element.Name}' belongs
```

Please note that the code above assumes that each IfcBuildingElementPart is associated with only one IfcWall through the decomposition relationship. In reality, an element can be part of multiple elements, so you might need to adjust the code if that's the case in your IFC file.

Regenerate

Send a message ?

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. [ChatGPT August 3 Version](#)



Installation Python

The screenshot shows the Python.org homepage. At the top, there is a navigation bar with links for Python, PSF, Docs, PyPI, Jobs, and Community. Below the navigation bar is the Python logo and a search bar. The main content area features a blue header with tabs for About, Downloads, Documentation, Community, Success Stories, News, and Events. The Downloads tab is active, showing a sidebar with code snippets and a list of download options: All releases, Source code, Windows, macOS, Other Platforms, License, and Alternative Implementations. A large callout box highlights the 'Download for Windows' section, which includes a link to 'Python 3.11.5'. A note states that Python 3.9+ cannot be used on Windows 7 or earlier. Below this, it says 'Not the OS you are looking for? Python can be used on many operating systems and environments.' and 'View the full list of downloads.' At the bottom of the page, there are four sections: 'Get Started', 'Download', 'Docs', and 'Jobs', each with a brief description and a link.

Python is a programming language that lets you work quickly
and integrate systems more effectively. [»» Learn More](#)

Get Started
Whether you're new to
programming or an experienced
developer, it's easy to learn and use
Python.
[Start with our Beginner's Guide](#)

Download
Python source code and installers
are available for download for all
versions!
Latest: [Python 3.11.5](#)

Docs
Documentation for Python's
standard library, along with tutorials
and guides, are available online.
[docs.python.org](#)

Jobs
Looking for work or have a Python
related position that you're trying to
hire for? Our [relaunched](#)
[community-run job board](#) is the
place to go.
[jobs.python.org](#)

Python

- **Link:** www.python.org

Werkzeuge / Editoren

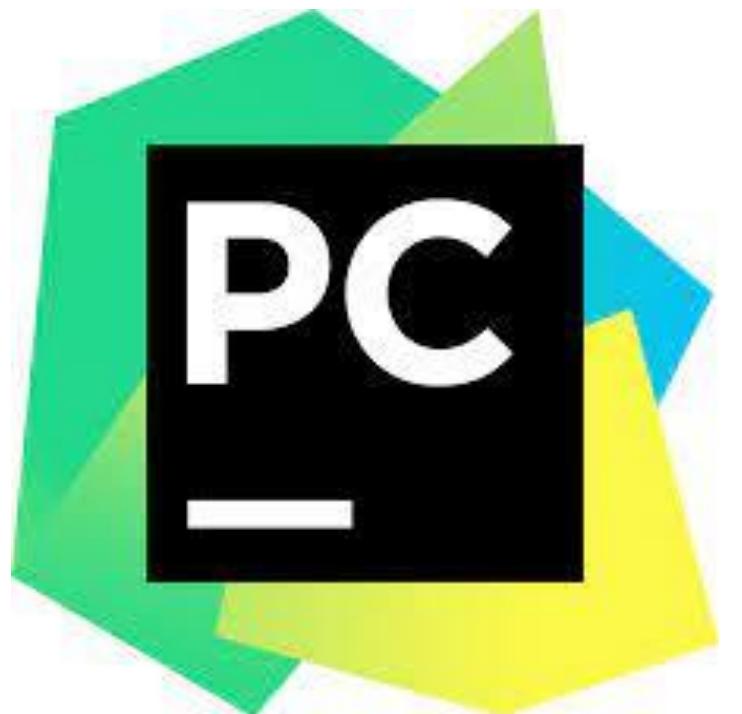
Visual Studio Code

- Link: <https://code.visualstudio.com/>

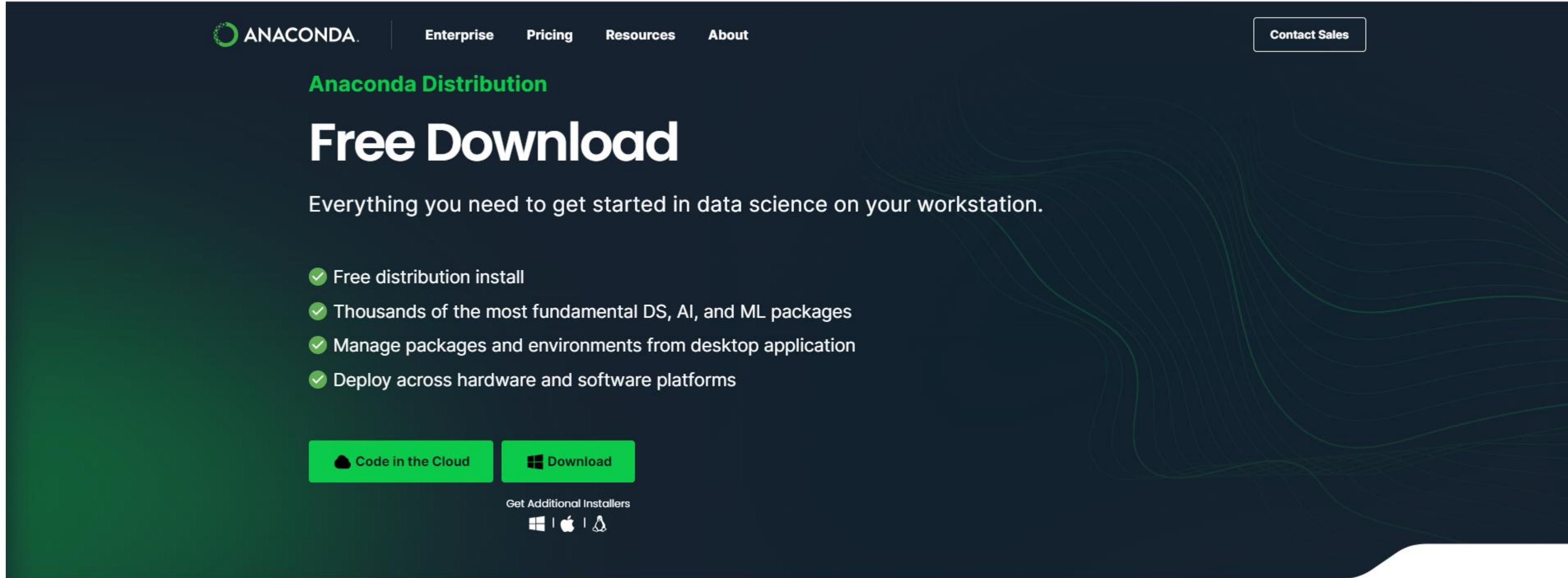


PyCharm

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



Werkzeuge / Management der Umgebungen



Anaconda Navigator

- **Link: <https://www.anaconda.com/download>**

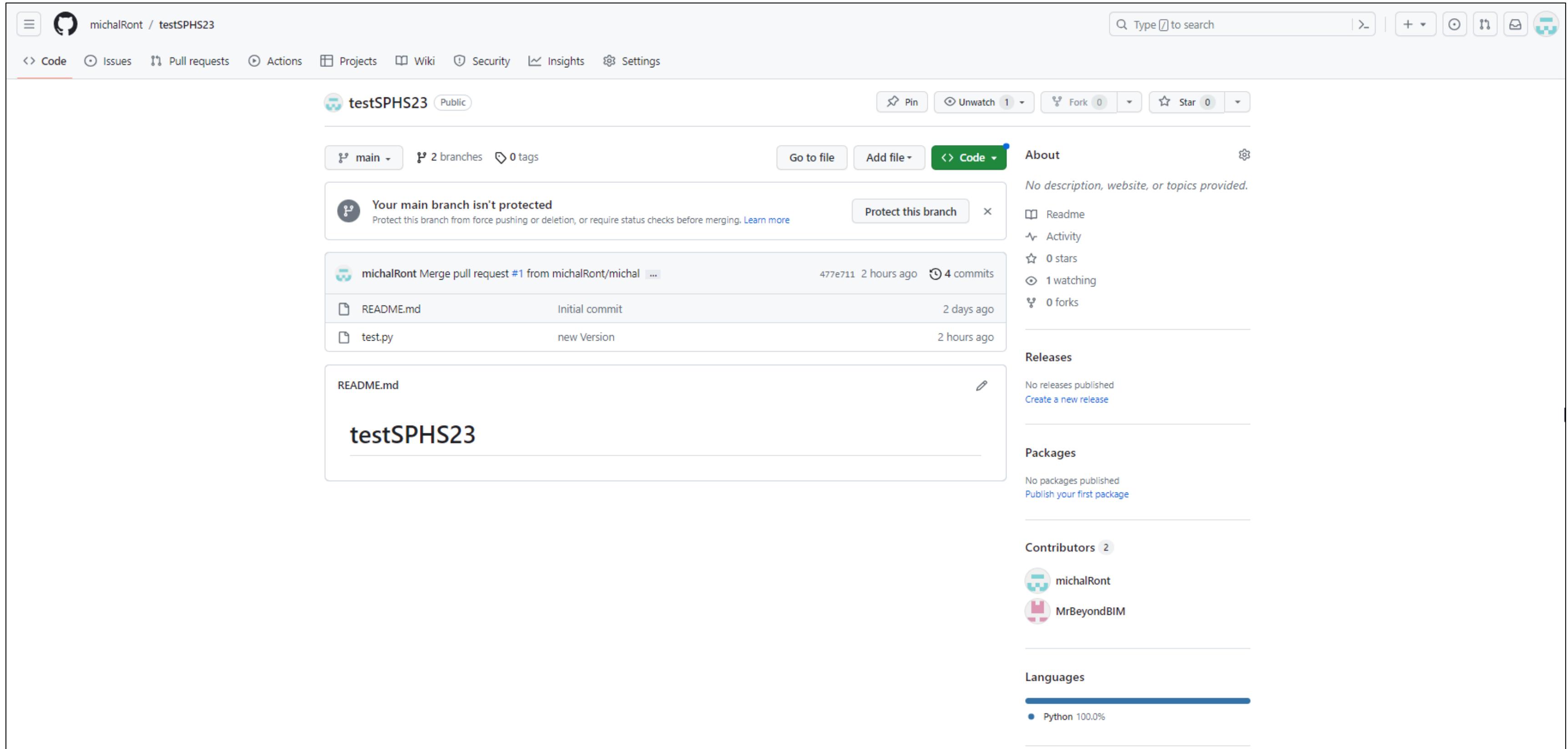
Werkzeuge / Git für die Versionskontrolle (Technologie)

The screenshot shows the official Git website (<https://git-scm.com/>). At the top, the Git logo and the tagline "distributed-is-the-new-centralized" are displayed. A search bar is located at the top right. Below the header, there are two main sections of text: one about Git's distributed nature and another about its performance and features. To the right of the text is a diagram illustrating a distributed Git repository with multiple nodes connected by red lines. Below this, there are several navigation links: "About" (with a gear icon), "Documentation" (with a book icon), "Downloads" (with a download arrow icon), and "Community" (with a speech bubble icon). On the left side, there is a small image of the "Pro Git" book. In the center, there is a large monitor icon displaying the text "Latest source Release 2.42.0" and a "Download for Windows" button. Below the monitor are links for "Windows GUIs", "Tarballs", "Mac Build", and "Source Code". At the bottom, there is a section titled "Companies & Projects Using Git" featuring logos for Google, Microsoft, Twitter, LinkedIn, Netflix, Camel, PostgreSQL, and Android.

Git

- Link: <https://git-scm.com/>

Werkzeuge / GitHub Account (Plattform)



Git

- **Link: <https://github.com/>**

ILIAS : Grundlagen, Informationen + Kommunikation

TA.BA_DT_PROGR_MM.H2401
TA Digital Twin Programmieren

Zur Favoriten Inhalt Info Einstellungen Mitglieder Lernfortschritt Metadaten Export Rechte Papierkorb Voransicht als Mitglied aktivieren >

Zeilen Verwalten Sortieren

Neues Objekt hinzufügen Seite gesetzen



Digital Twin Programmieren
Skriptung und Programmierung bieten die Möglichkeit für mehr Effizienz und Innovation. Von der einfachen Automatisierung sich wiederholender Aufgaben über die Möglichkeit, die Funktionalität der Software an die eigenen Bedürfnisse und Arbeitsmethoden anzupassen und zu erweitern, bis hin zur Erstellung massgeschneiderter Applikationen - Skriptung und Programmierung eröffnen eine neue Welt der Möglichkeiten im Rahmen der Digital Construction. In diesem Modul werden die Grundlagen der Programmierung und des Skripts mit dem Schwerpunkt Python vermittelt und an eigenen Projekten praktisch eingesetzt.

Zoom Link
Meeting-ID: 695 8079 5469
Kenncode: S60523

Miroboard Link

DC-Software Kiosk

Unterricht
Kursunterlagen
Abgaben / Prüfungen

Allgemeine Informationen
Semesterplan
Leistungsnachweis
Allgemeine Dokumente

Ansprechpartner
Modulverantwortliche
[Michal Rontsinsky](#)
Dozierende
[Nora Bukovits](#)
[Andrés Velasco Muro](#)

Unterrichtsdaten
Kontaktunterrichtszeiten:
Montag: 18.30 Uhr - 20.55 Uhr
Datum: 16.09.2024 - 16.12.2024
Unterrichtsraum physisch:
Raumnummer: E209 oder online

Python Hauptseite

The screenshot shows the official Python website homepage. At the top, there is a dark navigation bar with tabs for "Python", "PSF", "Docs", "PyPI", "Jobs", and "Community". Below the navigation bar is the Python logo and a search bar with a "Donate" button. The main content area features a large image of the Python logo with a glowing effect, surrounded by three cardboard boxes. To the left of the image, there is a block of text listing reasons why Python is popular: "Python is powerful... and fast; plays well with others; runs everywhere; is friendly & easy to learn; is Open." Below this text, a smaller note says: "These are some of the reasons people who use Python would rather not use anything else." On the left side, there is a section titled "Getting Started" with a sub-section "Friendly & Easy to Learn". Both sections contain descriptive text and links to further resources.

**Python is powerful... and fast;
plays well with others;
runs everywhere;
is friendly & easy to learn;
is Open.**

These are some of the reasons people who use Python would rather not use anything else.

Getting Started

Python can be easy to pick up whether you're a first time programmer or you're experienced with other languages. The following pages are a useful first step to get on your way writing programs with Python!

- Beginner's Guide, Programmers
- Beginner's Guide, Non-Programmers
- Beginner's Guide, Download & Installation
- Code sample and snippets for Beginners

Friendly & Easy to Learn

The community hosts conferences and meetups, collaborates on code, and much more. Python's documentation will help you along the way, and the mailing lists will keep you in touch.

- Conferences and Workshops
- Python Documentation
- Mailing Lists and IRC channels

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

Python Code Academy

codecademy Catalog Resources ▾ Community ▾ Pro Pricing ▾ For Business

Log In Sign Up

Languages ^

HTML & CSS

Python

JavaScript

Java

SQL

Bash/Shell

Ruby

C++

R

C#

PHP

Go

Swift

Kotlin

Subjects ^

Web Development

Data Science

Computer Science

Python

Python is a general-purpose, versatile, and powerful programming language. It's a great first language because it's concise and easy to read. Whatever you want to do, Python can do it. From web development to machine learning to data science, Python is the language for you.

Why we love it:

- Great first language
- Large programming community
- Excellent online documentation
- Endless libraries and packages
- World-wide popularity
- Powerful and flexible

Featured resources

FORUM Python 3 Codecademy Forums

CHEATSHEET Python 3: Syntax Cheatsheet

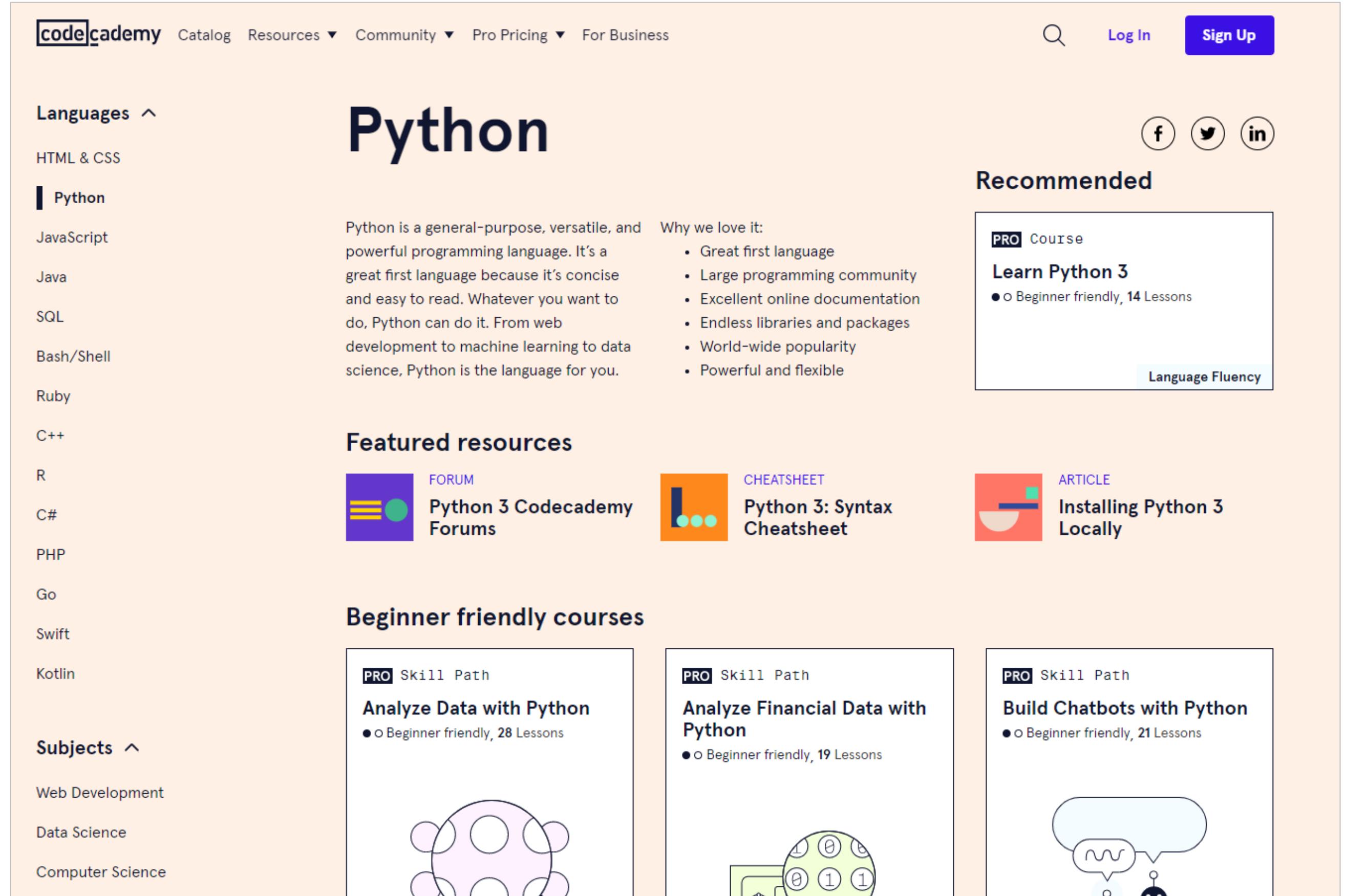
ARTICLE Installing Python 3 Locally

Recommended

PRO Course Learn Python 3

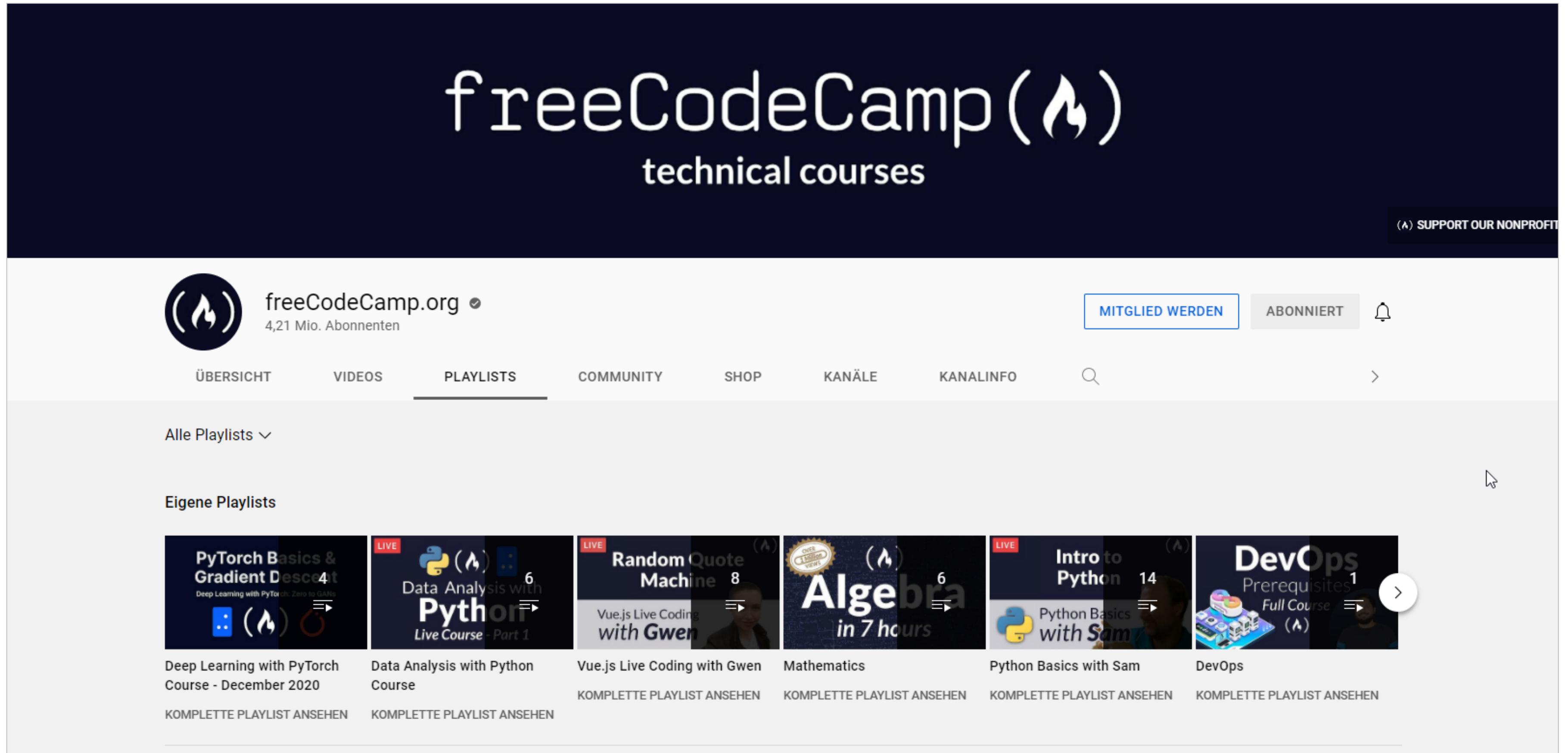
• Beginner friendly, 14 Lessons

Language Fluency



<https://www.codecademy.com/catalog/language/python>

YouTube Quellen - FreeCodeCamp



<https://www.youtube.com/c/Freecodecamp/playlists>

YouTube Quellen – Programming with Mosh

The screenshot shows the YouTube channel page for 'Programming with Mosh'. The channel has 1,99 Mio. Abonnenten and is currently subscribed. The main heading is 'Become the **BEST CODER** You can be'. Below it, there's a search bar with the hashtags '#python #javascript #csharp'. The navigation bar includes links for ÜBERSICHT, VIDEOS, PLAYLISTS (which is underlined), COMMUNITY, SHOP, KANÄLE, KANALINFO, and a search icon. A 'SORTIEREN NACH' button is also present. The page displays several playlists:

- Mobile Development: React Native IN 2 HOURS (6 videos)
- Job Interview Preparation Videos: DATA STRUCTURES For Beginners (6 videos)
- Programming Languages: TOP 3 PROGRAMMING LANGUAGES (8 videos)
- Front-end Development: #1 JAVASCRIPT FRAMEWORK? (11 videos)
- Back-end Development: What is Node.js? (16 videos)
- Beginner's Questions: HOW TO CODE & EARN \$60,000 (14 videos)
- Python Tutorials: PYTHON FULL COURSE (21 videos)
- JavaScript Tutorials: What is JavaScript? (21 videos)
- Node.js Tutorials: Chrome Node.js (14 videos)
- TypeScript Tutorials for Angular/React Developers: TYPESCRIPT IN 1 HOUR (6 videos)
- Angular Tutorials: ANGULAR IN 2 HOURS (14 videos)
- C#.NET Tutorials: Events & Delegates (14 videos)

Each item includes a thumbnail, title, number of videos, and a 'KOMPLETTE PLAYLIST ANSEHEN' link.

<https://www.youtube.com/c/programmingwithmosh/playlists>

Aufgaben für nächste Woche

Aufgaben für nächste Woche

1. Python auf eigenem Gerät installieren (falls noch nicht vorhanden).

2. Installation Editor nach Wahl > Visual Studio Code oder PyCharm.

3. Installation Git.

4. Erstellen eines Accounts für GitHub.

5. Überlegen Inhalte, Ziele und Kontext für eigene Projekte.

6. Einteilen in die Gruppen überlegen.

7. Wiederholung Python Grundlagen und Syntax.

Hochschule Luzern

Michal Rontsinsky
Dozent

Michal.Rontsinsky@hslu.ch