

Category: Programming

## Digi2Cloud Intro to Python: From Zero to Hero #080202

---

Das Training findet sich täglich vom 08:30 bis 12:30 statt.

Trainings-Dauer: 28.04.2025 bis zum 25.06.2025

Feiertage: 01.05, 29.05, 09.06

Die ersten 3 Tage finden sich am Ort in Circle 6 am Zürich Flughafen statt.

Das Training in den Resttagen (bis zum 25.06.2025) findet sich nur Online statt.

The minimum hardware requirements for Training Students:

- 1 GHz single-core processor,
- 2 GB of RAM,
- 10 GB of free disk space,

Software requirements on Students-Laptops:

- Basic operating system like Windows 7/8/10, macOS, or Linux
- Office, Browser and Internet Verbindung
- PyChar als integrierte Entwicklungsumgebung (open source full featured IDE )
- PostgreSQL DB (Installer: certified by EnterpriseDB for all supported PostgreDB Versions)
- SQLAlchemy/Alembic
- Pydantic (Python Package)
- Redis (NoSQL database)
- Java Script

Ziel Publikum: Anfänger und Quereinsteiger

## Anforderungen

No previous knowledge required

## Key Learnings

- Introduction to Python & Development Environment
- Variables and Data Types
- User Input and Basic Operations
- Control Flow and basic logic
  - Loops
  - If /elif / else
- Functions
- Error Handling and Debugging
  - Syntax Errors
  - Try / Except / Finally
- Basic Data Structures
  - Lists, Dictionaries, Tuples
  - JSON
- Writing Modular Code & Importing Modules
- Basic Algorithms
- File Handling – Working with Files and Directories
- Introduction to OOP – Classes and Objects
  - Class Methods and Attributes
  - Inheritance
- Introduction to RESTful APIs
  - Python's `requests` library
  - Handling API responses (JSON parsing)
  - Error Handling in API Calls
- Working with External Libraries
  - Installing packages with `pip`
  - Using libraries like `datetime`, `random`
- Build your Own Python Project

## Inhalt

Wk. 1	Python Fundamentals
Mon	<p>(08:30) Introduction to Python, setting up development environments (Replit).</p> <p>(09:30) 30 min practice with basic Python syntax &amp; 30 min solutions.</p> <p>(10:30) 1-hour (coding together) tutorial: Writing your first Python script.</p> <p>(11:30) Independent work on Homework, reading course materials and "warm-up" exercises.</p>
Tue	<p>(08:30) 1 hour lecture: Variables and Data Types in Python.</p> <p>(09:30) 30 min practice &amp; 30 min solutions on variables and data types.</p> <p>(10:30) 1-hour (coding together) tutorial: Working with int, float, str, and type conversions.</p> <p>(11:30) Independent work on Homework and exercises.</p>
Wed	<p>(08:30) 1 hour lecture: User Input, String Manipulation, and Basic Operators.</p> <p>(09:30) 30 min exercises &amp; 30 min solutions on string formatting and user input handling.</p> <p>(10:30) 1 hour (coding together) tutorial: Creating an interactive Python script with user input.</p> <p>(11:30) 1 hour Home exercises and coding practice.</p>
Thur	<p>(08:30) 1 hour lecture: Control Flow – if, elif, else conditions.</p> <p>(09:30) 30 min exercises &amp; 30 min solutions on control flow.</p> <p>(10:30) 1-hour (coding together) tutorial: Writing decision-making programs.</p> <p>(11:30) 1 hour Home exercises and coding practice.</p>
Fri	<p>(08:30) 1 hour lecture: Loops – for and while.</p> <p>(09:30) 30 min exercises &amp; 30 min solutions on loops.</p> <p>(10:30) 1-hour (coding together) tutorial: Writing iterative programs.</p> <p>(11:30) 1 hour of Home exercises and coding practice.</p>

Wk. 2	Functions, Debugging, and File Handling
Mon	(08:30) 1-hour lecture: Introduction to Functions – Defining and Calling Functions. (09:30) 30 min practice with Functions & 30 min solutions. (10:30) 1-hour (coding together) tutorial: Writing reusable Python Functions. (11:30) 1 hour Independent work on Homework and reading course materials.
Tue	(08:30) 1-hour lecture: Error Handling & Debugging – Syntax Errors and Logical Errors. (09:30) 30 min debugging exercises & 30 min solutions. (10:30) 1-hour (coding together) tutorial: Using print debugging. (11:30) Independent work on Homework and exercises.
Wed	(08:30) 1-hour lecture: Try / Except / Finally (09:30) 30 min exercises & 30 min solutions on error handling. (10:30) 1-hour (coding together) tutorial: Building a robust input validation system. (11:30) 1 hour of Home exercises and coding practice.
Thur	(08:30) 1-hour lecture: Introduction to Base Python Libraries. (09:30) 30 min exercises & 30 min solutions on importing and using modules. (10:30) 1-hour (coding together) tutorial: Using random, and datetime modules. (11:30) 1 hour of Home exercises and coding practice.
Fri	(08:30) 1 hour lecture: Recap of all the previous topics (09:30) 30 min exercises & 30 min solutions to problems with previous topics.. (10:30) 1-hour (coding together) tutorial: Simple Expense Calculator (11:30) 1 hour Project Work & Questions

<b>Wk. 3</b>	Data Structures – Lists, Dictionaries, Tuples
<b>Mon</b>	Data Structures – Lists, Dictionaries, Tuples
<b>Tue</b>	(08:30) 1-hour lecture: Introduction to Lists and Tuples. (09:30) 30 min exercises & 30 min solutions on list operations. (10:30) 1-hour (coding together) tutorial: Implementing list-based applications. (12:00) Independent work on Homework and exercises.
<b>Wed</b>	(08:30) 1.5-hour lecture: Working with Dictionaries. (09:30) 30 min exercises & 30 min solutions on dictionary manipulation. (10:30) 1-hour (coding together) tutorial: Building a dictionary-based application. (11:30) Independent work on Homework and exercises.
<b>Thur</b>	(08:30) 1-hour lecture: Advanced Data Structures – Nested Lists and Dictionaries. (09:30) 30 min exercises & 30 min solutions on nested data structures. (10:30) 1-hour (coding together) tutorial: Implementing a basic student record system. (11:30) 1 hour of Home exercises and coding practice.
<b>Fri</b>	(08:30) 1 hour lecture: Object-Oriented Programming (OOP) Basics. (09:30) 30 min exercises & 30 min solutions on classes and objects. (10:30) 1-hour (coding together) tutorial: Writing a class-based application. (11:30) 1 hour of Home exercises and coding practice.

Wk. 4	Working with Data & Advanced Python Concepts
Mon	<p>(08:30) 1-hour lecture: Understanding Files, Paths, and Directories in Python.</p> <p>(09:30) 30 min exercises &amp; 30 min solutions on reading/writing files (txt, CSV).</p> <p>(10:30) 1-hour (coding together) tutorial: Implementing a simple file-based storage system.</p> <p>(11:30) 1 hour of Home exercises and coding practice.</p>
Tue	<p>(08:30) 1-hour lecture Introduction to JSON – Working with structured data.</p> <p>(09:30) 30 min exercises &amp; 30 min solutions on reading/writing JSON.</p> <p>(10:30) 1 hour (coding together) tutorial: Parsing and storing JSON data in Python.</p> <p>(11:30) 1 hour of Home exercises and coding practice.</p>
Wed	<p>(08:30) 1-hour lecture Introduction to Python Modules – Creating and Using Modules.</p> <p>(09:30) 30 min exercises &amp; 30 min solutions on importing and using modules.</p> <p>(10:30) 1 hour (coding together) tutorial - Writing and packaging your own Python module</p> <p>(11:30) 1 hour Project Work &amp; Questions</p>
Thur	<p>(08:30) 1-hour lecture on basic algorithms</p> <p>(09:30) 30 min exercises and 30 min solutions to basic algorithm questions</p> <p>(10:30) 1 hour (coding together) tutorial -- optimizing an algorithm for speed</p> <p>(11:30) 1 hour Project Work and Questions</p>
Fri	<p>(08:30) 1-hour lecture Review of key topics from the week</p> <p>(14:00) 30 min exercises &amp; 30 min solutions Solving real-world problems using file handling, JSON, and modules</p> <p>(10:30) 1 hour (coding together) tutorial on loading and editing local data</p> <p>(11:30) 1 hour Project Work &amp; Questions</p>

Wk. 5	Working with APIs & External Libraries
Mon	<p>(08:30) 1-hour lecture: Introduction to RESTful APIs and how they work.</p> <p>(09:30) 30 min exercises &amp; 30 min solutions on API requests and responses.</p> <p>(10:30) 1-hour (coding together) tutorial: Making API calls using Python's <code>requests</code> library.</p> <p>(11:30) 1 hour Independent work on Homework and reading course materials.</p>
Tue	<p>(08:30) 1-hour lecture: Handling API responses (JSON parsing).</p> <p>(09:30) 30 min exercises &amp; 30 min solutions on parsing JSON data from APIs.</p> <p>(10:30) 1-hour (coding together) tutorial: Implementing an API-powered application.</p> <p>(11:30) 1 hour Independent work on Homework and exercises.</p>
Wed	<p>(08:30) 1-hour lecture: Error handling in API calls.</p> <p>(09:30) 30 min debugging exercises &amp; 30 min solutions.</p> <p>(10:30) 1-hour (coding together) tutorial: Handling API errors (timeouts, status codes).</p> <p>(11:30) 1 hour of Home exercises and coding practice.</p>
Thur	<p>(08:30) 1-hour lecture: Data Serialization and File Formats (JSON, CSV, XML).</p> <p>(09:30) 30 min exercises &amp; 30 min solutions on working with different formats.</p> <p>(10:30) 1-hour (coding together) tutorial: Reading and writing structured data files.</p> <p>(11:30) 1 hour of Home exercises and coding practice.</p>
Fri	<p>(08:30) 1-hour lecture: Pandas introduction</p> <p>(09:30) 30 min exercises &amp; 30 min solutions: Reading in CSV data.</p> <p>(10:30) 1-hour (coding together) tutorial: Reading and writing structured data files.</p> <p>(11:30) 1 hour of Home exercises and coding practice.</p>

Wk. 6	Object-Oriented Programming (OOP)
Mon	(08:30) 1-hour lecture Introduction to Object-Oriented Programming (OOP). (09:30) 30 min exercises & 30 min solutions on defining classes and objects. (10:30) 1-hour (coding together) tutorial: Writing your first class in Python. (11:30) 1 hour project work and questions
Tue	(08:30) 1-hour lecture: Class Methods and Attributes. (09:30) 30 min exercises & 30 min solutions on instance vs. class attributes. (10:30) 1-hour (coding together) tutorial: Implementing class methods in an application. (11:30) 1 hour project work and questions
Wed	(08:30) 1-hour lecture: Understanding Inheritance and Polymorphism. (09:30) 30 min exercises & 30 min solutions on extending classes. (10:30) 1-hour (coding together) tutorial: Creating a class hierarchy. (11:30) 1 hour Project Work & Questions
Thur	(08:30) 1-hour lecture: Combining OOP and APIs in Python Applications. (09:30) 30 min exercises & 30 min solutions on API-based OOP applications. (10:30) 1-hour (coding together) tutorial: Developing a small class-based API tool. (11:30) 1 hour Project Work & Questions
Fri	(08:30) 1-hour lecture: Recap (09:30) 30 min exercises & 30 min solutions on recap topics (10:30) 1-hour (coding together) tutorial: ____ (11:30) 1 hour Project Work & Questions



Wk. 7	Working on a Python Project
Mon	(08:30) Introduction to the Final Project – Planning and Structuring the Code. (09:30)Brainstorming project ideas and outlining features. (10:30)Group discussions and feedback on project structure. (11:30)Independent project development time.
Tue	(08:30) -(11:30)Full workdays on individual/group Python projects with instructor check-ins.
Wed	(08:30) -(11:30)Full workdays on individual/group Python projects with instructor check-ins.
Thur	(08:30) -(11:30)Full workdays on individual/group Python projects with instructor check-ins.
Fri	(08:30) 1-hour review of project progress, debugging session. (09:30)Code reviews and feedback sessions. (10:30)Finalizing project structure (11:30)Continue working on individual/group projects.

Wk. 8	Project Presentations & Course Wrap-Up
Mon	(08:30) -(11:30)Continued work on projects, implementing final touches.
Tue	(08:30) -(11:30)Continued work on projects, implementing final touches.
Wed	(08:30) Final debugging and code reviews. (09:30)Final testing and deployment (if applicable). (10:30)Students preparing presentation slides. (11:30)Independent work.
Thur	(08:30) 2-hours Final Project Presentations. (10:30)2-hours Feedback and discussion.
Fri	(08:30) 2-hours Course recap and next steps in Python learning. (10:30)2-hours Closing Q&A and Farewell session.