YOUR SCIENCE

CREATING OPPORTUNITY

Applied Analytics
AI & ML for BI & Automation

1. Getting Started with Python

Your Science
Mathematical Consulting
Prof. Norbert Poncin

2025

Your Science Center: 31, Boulevard Prince Henri, L-1724 Luxembourg
Phone: +352 621 674 917, **Email**: info@yourscience.eu, **Website**: yourscience.eu
Business Permit: 10154927, **LBR**: A44409, **VAT**: LU35024328

YOUR SCIENCE

CREATING OPPORTUNITY

Applied Analytics
AI & ML for BI & Automation

1. Getting Started with Python

Your Science
Mathematical Consulting
Prof. Norbert Poncin

2025

Your Science Center: 31, Boulevard Prince Henri, L-1724 Luxembourg
Phone: +352 621 674 917, **Email**: info@yourscience.eu, **Website**: yourscience.eu
Business Permit: 10154927, **LBR**: A44409, **VAT**: LU35024328

Contents

Python Environments, Hierarchy, and Code Structure

1 Training Presentation

This training module provides insights into Artificial Intelligence (AI) and Machine Learning (ML) with applications in Data Analysis, Automation, Customer Relationship Management (CRM), and Business Intelligence (BI). The goal is to *equip participants with the foundational knowledge necessary for independent skill development*.

Python

The module leverages Python, a versatile programming language renowned for its extensive applications in industrial data ecosystems, which will be explored progressively throughout the course.

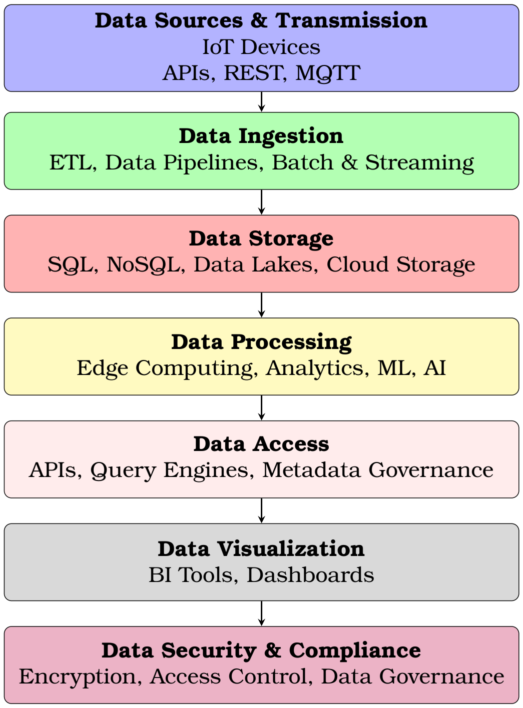
Data Sources & Transmission
IoT Devices
APIs, REST, MQTT
Data Ingestion
ETL, Data Pipelines, Batch & Streaming
Data Storage
SQL, NoSQL, Data Lakes, Cloud Storage
Data Processing
Edge Computing, Analytics, ML, AI
Data Access
APIs, Query Engines, Metadata Governance
Data Visualization
BI Tools, Dashboards
Data Security & Compliance
Encryption, Access Control, Data Governance

Figure 1: Python in Industrial Data Ecosystems

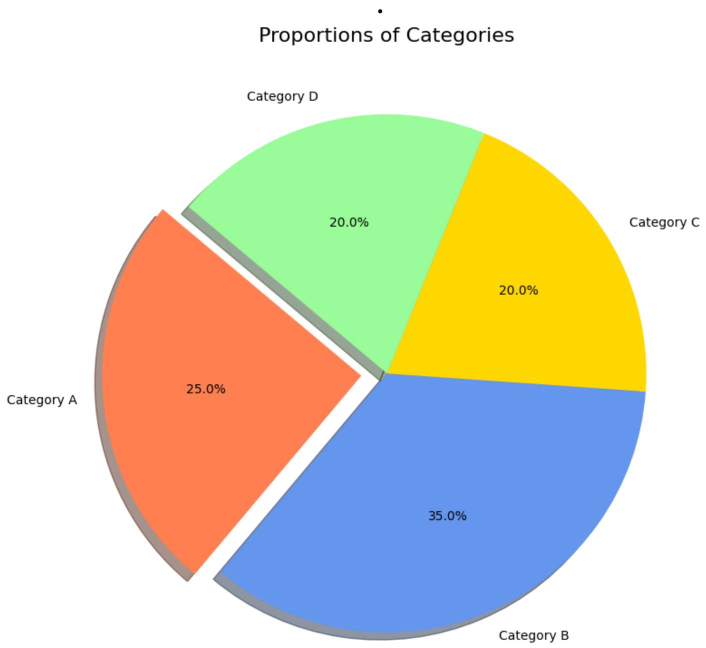

Figure 5: Output Code SWML1Ex1.py

A Command Prompt window is a **basic text-based interface**, which may feel less user-friendly compared to modern environments...

- Local Python Installation in JupyterLab:

  JupyterLab is a **user-friendly graphical interface** (it allows users to interact with the system through visual elements like menus, buttons, and windows, rather than relying solely on text-based commands), similar to Google Colab, but it runs *locally* on your machine.

  To launch JupyterLab:

  - Open a command prompt.
  - Navigate to your Downloads folder: `cd C:/Users/YourName/Downloads`.
  - Start JupyterLab: `python -m jupyterlab` (`m` stands for `module`).

```
Microsoft Windows [Version 10.0.19045.5131]
(c) Microsoft Corporation. All rights reserved.
```

```
C:\WINDOWS\System32>cd C:/Users/norbert.poncin/Downloads

C:\Users\norbert.poncin\Downloads>python -m jupyterlab
```

Pressing `Enter` will display the following in the command prompt:

```
Microsoft Windows [Version 10.0.19045.5131]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:/Users/norbert.poncin/Downloads

C:\Users\norbert.poncin\Downloads>python -m jupyterlab
[I 2025-01-12 11:53:04.418 ServerApp]  jupyter_lsp extension was
successfully linked.
[I 2025-01-12 11:53:04.418 ServerApp]  jupyter_server_terminals
extension was successfully linked.
...
[C 2025-01-12 11:53:10.666 ServerApp]

    To access the server, open this file in a browser:
...
```

After a few seconds, a new web browser window opens with the address `http://localhost:8888`, which resembles a web URL (Uniform Resource Locator) but refers to a local web server running on your own machine. The *browser-based design* of JupyterLab is particularly advantageous when Python and JupyterLab are installed on a *remote server*, or when your computer acts as a server. Such a configuration allows you to access JupyterLab from any device with a web browser, eliminating the need for local installations. It facilitates collaboration, centralized resource management, and enhanced security.

To create a new notebook in JupyterLab, navigate to **File** → **New** → **New Notebook**. Alternatively, click the **+** button in the file browser,

- For new individuals' heights, we can estimate their weights using the red least squares line as an approximation. This line is known as a **linear regression model** because it represents a *linear* relationship between 'squared height' and 'weight', is derived from *regression* analysis of previously observed data, and serves as a *model* to predict weights based on (squared) heights.

3.2 Python Building Blocks and Code Structure

Next, we present the code that computes the least squares line and generates the plot shown above. We introduce it step by step, providing explanations for each part before moving to the next. This code will allow us to understand Python's main building blocks and the standard structure of a Python code.

3.2.1 Python's Components

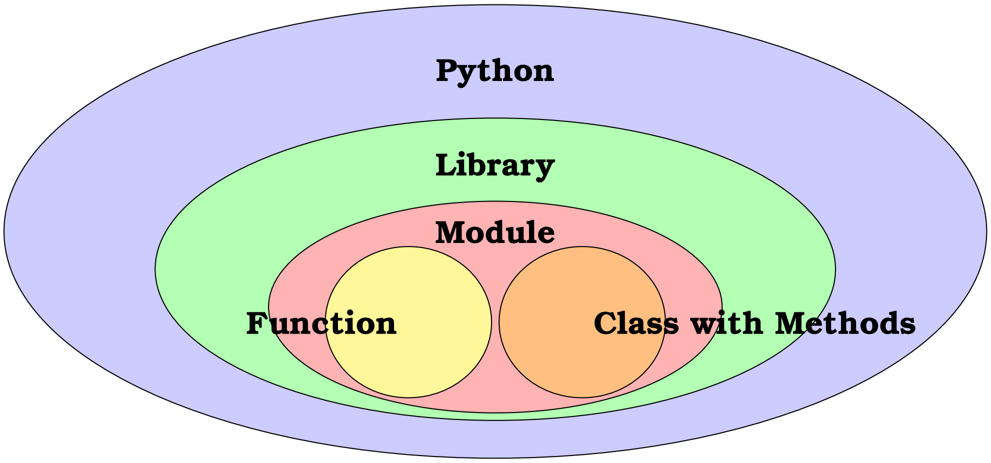

Figure 9: Python's Hierarchy

1. Python's ecosystem contains specialized **libraries** that can be imported into program scripts.

```python
import numpy as np
import os
```

Pages 18–31 are not part of this preview.

3.3 The Good News: Learning Python with ChatGPT

How do you best learn the Python programming language? Think of how you learned your mother tongue – by listening to those around you and actively practicing. With Python, your closest relative is ChatGPT, a true Python poet. Start by reading the code it generates based on your instructions. Identify the key components of the code – such as libraries, modules, functions, classes, and methods – and gain insight into its core logic in the main script. Analyze which parts align with your expectations and which do not. Debug, improve, and validate the code. Then, challenge yourself by integrating it into larger projects, gradually expanding your skills and ambitions.

Additionally, focus on critical thinking when evaluating code for logic, scalability (e.g., handling larger datasets), and real-world applicability. Moreover, understanding how Python integrates with industrial data ecosystems enhances your ability to optimize your code.

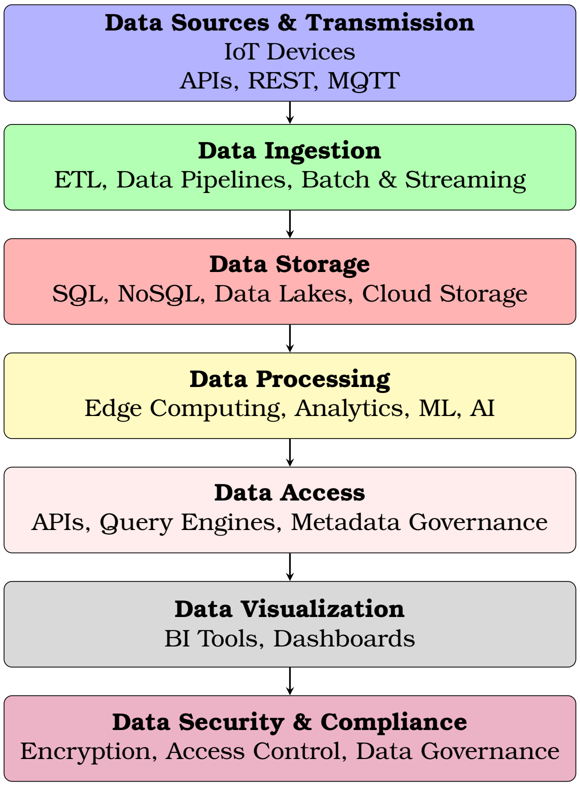
Data Sources & Transmission
IoT Devices
APIs, REST, MQTT
Data Ingestion
ETL, Data Pipelines, Batch & Streaming
Data Storage
SQL, NoSQL, Data Lakes, Cloud Storage
Data Processing
Edge Computing, Analytics, ML, AI
Data Access
APIs, Query Engines, Metadata Governance
Data Visualization
BI Tools, Dashboards
Data Security & Compliance
Encryption, Access Control, Data Governance

Figure 13: Python in Industrial Data Ecosystems

4 Learning Outcomes

After thoroughly working through this chapter, the readers should be able to:

- Understand the differences between a local Python installation used in a text-based command prompt and a graphical interface like JupyterLab. Explore and test these interfaces, including Google Colab, to grasp their features and differences in functionality and user experience.

- Explain the concepts of mean squared error and both linear and parabolic regression models. Clarify the meaning of the term 'regression' in this context.

- Describe Python's main building blocks, providing examples for each. Study an example of code to understand Python's code architecture.

- Grasp the concepts of class, object, and method in Python. Review a simple example generated by their AI assistant, walking through the code line by line, explaining what each part does in a beginner-friendly manner.

- Work through the self-paced activity at the end of the chapter to apply and review all fundamental concepts in a practical example.

YOUR SCIENCE

CREATING OPPORTUNITY

About the Author

Norbert Poncin is a Luxembourgish mathematician, who was originally educated as a mathematical analyst and has worked extensively in partial differential equations (PDEs) at the University of Liège. His Master's thesis focused on the propagation of singularities in boundary value problems (BVPs) for dynamic hyperbolic systems. Applying the finite element method (FEM), his subsequent dissertation addressed BVPs for complex elliptic systems of PDEs. For his doctoral thesis, he explored mathematical quantization, while his post-doctoral education at the Polish Academy of Sciences strongly emphasized theoretical physics and its models.

Norbert has served as a Full Professor of Mathematics at the University of Luxembourg for more than 25 years and collaborated with more than 25 foreign professors and post-doctoral scholars. He has organized numerous academic events, notably approximately 10 international research meetings and over 20 research seminars focusing on theories, frameworks, concepts and models in Physics and Engineering. Beyond a substantial publication record in Differential Geometry, Algebraic Topology, and related disciplines, he has contributed roughly 25 papers to the fields of Mathematical Physics and Quantum Theory.

He was the leading instructor for over 20 university courses. Spanning a diverse spectrum of subjects, including mathematical analysis, probability theory, inferential statistics, point and solid dynamics, Lagrangian and Hamiltonian mechanics, mechanics of deformable solids, fluid dynamics, special relativity, quantum physics, geometric methods in mathematical physics, and supersymmetric models, his teaching portfolio underscores his extensive experience in applied aspects of mathematics.

In 2023, Norbert Poncin founded the mathematical consulting agency Your Science, where he currently serves as director. His primary interests include data science and artificial intelligence, along with mathematical modeling and computational science.