

# Teaching guide

## IDENTIFICATION DETAILS

|                                      |                              |                       |       |
|--------------------------------------|------------------------------|-----------------------|-------|
| Degree:                              |                              |                       |       |
| Faculty/School:                      | Law, Business and Government |                       |       |
| Course:                              | PYTHON: FROM ZERO TO HERO    |                       |       |
| Type:                                | Compulsory Internal          | ECTS credits:         | 9     |
| Year:                                | 2                            | Code:                 | 45811 |
| Teaching period:                     |                              | Third-Fourth Semester |       |
| Teaching type:                       |                              | Classroom-based       |       |
| Language:                            |                              | Inglés                |       |
| Total number of student study hours: |                              | 225                   |       |

| Teaching staff                  | E-mail               |
|---------------------------------|----------------------|
| Pedro Francisco Anquela Lecuona | pedro.anquela@ufv.es |

## SUBJECT DESCRIPTION

The course provides a comprehensive Python programming education starting from fundamental syntax through advanced material to practical frameworks including Django and Flask. The course provides hands-on learning through exercises and projects and live coding for students who have no prior coding experience. The course teaches procedural and object-oriented programming as well as file handling and GUI development with Tkinter and PySide and database connectivity and full-stack web development using Python. Students will learn to create complete Python-based applications and understand backend web frameworks which businesses use after completing this course.

## GOAL

The primary goal of this course is to provide students with functional Python programming abilities that they can

use for data management and software creation and GUI development and web application deployment.  
Specific objectives:

Learn the core principles of Python programming.

Use object-oriented programming principles to solve real-world problems.

Work with files, databases, and external libraries. Design user interfaces using Tkinter and PySide.

Create and deploy web applications using Django and Flask.

## PRIOR KNOWLEDGE

No prior programming knowledge is required. However, familiarity with basic computer usage, file systems, and logic structures is recommended. Students must bring a laptop, first session will be dedicated to install the development environment:

Python (latest stable version)

IDE (e.g., PyCharm or VSCode)

Git

Conda

## COURSE SYLLABUS

### COURSE SYLLABUS

#### Part I — Python Basics

Introduction to Python and PyCharm

Variables and Data Types

Operators and Control Flow (if- else, loops)

Functions and Modularity

Collections: List, Tuple, Set, Dictionary

#### Part II — Object-Oriented Programming and Files

Classes and Objects

Encapsulation, Inheritance, Polymorphism

Exception Handling

File Handling

Mini-projects using OOP

#### Part III — Databases and GUI Applications

Connecting Python to PostgreSQL/ MySQL

CRUD operations and DAO design

Introduction to Tkinter and PySide

GUI components and layout managers  
GUI Projects: Calculator, Text Editor

#### Part IV — Web Development with Python

Introduction to Django Framework  
ORM, Views, URLs, Templates, Admin  
Full CRUD Web App with Django  
Flask Basics and Jinja Templates  
RESTful endpoints with Flask

## EDUCATION ACTIVITIES

The course theoretical combines knowledge with practical coding experience.  
Each session includes:

Live coding demonstrations.  
Collaborative coding exercises.  
Independent mini-projects

Students must:

Read material preparation before class Participate in code reviews and discussions  
Use GitLab to access and submit assignments  
The class will implement flipped learning to enhance student participation.

## DISTRIBUTION OF WORK TIME

| TEACHER-LED TRAINING ACTIVITIES | INDIVIDUAL WORK |
|---------------------------------|-----------------|
| 100 Horas                       | 125 Horas       |

## SKILLS

Students will learn Python programming fundamentals by mastering syntax rules and control structures and data types and functions.

Students will learn to use object-oriented programming principles for designing software application structures.

Students will create applications which connect to external systems through file and relational database interactions.

Students will learn to create graphical user interfaces (GUIs) through Python GUI frameworks.

Students will learn to construct complete web applications from front-end to back-end using Python web frameworks including Django and Flask.

## **SPECIFIC LEARNING RESULTS**

Write Python programs using structured and modular code practices.

Create and manipulate data collections using Python's built-in data structures.

Apply object-oriented principles to design reusable and maintainable code.

Handle exceptions and errors effectively in real-world scenarios.

Connect Python applications to relational databases (PostgreSQL/MySQL) and perform CRUD operations.

Develop and test GUI applications using Tkinter and PySide with event-driven programming logic.

Create RESTful endpoints and template-driven websites using Flask.

Build full web applications with Django including model definition, migrations, and user interfaces.

Use Git for version control and manage source code in collaborative environments.

## **LEARNING APPRAISAL SYSTEM**

Evaluation Items (Ordinary Call):

- [1] Written midterm exam — 20%
- [2] Individual assignments and mini-projects — 20%
- [3] Final Project — 50%
- [3] Attendance and in-class participation — 10%

## **ETHICAL AND RESPONSIBLE USE OF ARTIFICIAL INTELLIGENCE**

1.- The use of any Artificial Intelligence (AI) system or service shall be determined by the lecturer, and may only be used in the manner and under the conditions indicated by them. In all cases, its use must comply with the following principles:

- a) The use of AI systems or services must be accompanied by critical reflection on the part of the student regarding their impact and/or limitations in the development of the assigned task or project.
- b) The selection of AI systems or services must be justified, explaining their advantages over other tools or methods of obtaining information. The chosen model and the version of AI used must be described in as much detail as possible.
- c) The student must appropriately cite the use of AI systems or services, specifying the parts of the work where they were used and describing the creative process followed. The use of citation formats and usage examples may be

consulted on the Library website([https://www.ufv.es/gestion-de-la-informacion\\_biblioteca/](https://www.ufv.es/gestion-de-la-informacion_biblioteca/)).

d) The results obtained through AI systems or services must always be verified. As the author, the student is responsible for their work and for the legitimacy of the sources used.

2.- In all cases, the use of AI systems or services must always respect the principles of responsible and ethical use upheld by the university, as outlined in the [Guide for the Responsible Use of Artificial Intelligence in Studies at UFV](#). Additionally, the lecturer may request other types of individual commitments from the student when deemed necessary.

3.- Without prejudice to the above, in cases of doubt regarding the ethical and responsible use of any AI system or service, the lecturer may require an oral presentation of any assignment or partial submission. This oral evaluation shall take precedence over any other form of assessment outlined in the Teaching Guide. In this oral defense, the student must demonstrate knowledge of the subject, justify their decisions, and explain the development of their work.

## **BIBLIOGRAPHY AND OTHER RESOURCES**

### **Basic**

Eric Matthes Python Crash Course: A Hands-On, Project-Based Introduction to Programming 2nd Edition  
(Eric Matthes Python Crash Course: A Hands-On, Project-Based Introduction to Programming 2nd Edition ,  
||Luciano Ramalho Fluent Python: Clear, Concise, and Effective Programming 2nd Edition )

### **Additional**

Al Sweigart Automate the Boring Stuff with Python: Practical Programming for Total Beginners 2nd Edition