

TRAINING PROGRAM

PYTHON BEGINNER — ENI CERTIFICATION PREPARATION

TRAINING SUMMARY SHEET

Title	Python Beginner — ENI Certification Preparation
Duration	4 days (28 hours)
Reference	PY-DEB-4J-ENI
Price	€2,200 excl. tax / person
Format	In-person or remote
Audience	Beginners, career changers, developers, system administrators, data analysts
Prerequisites	No programming experience required. Basic computer literacy.
Certification	RS6701 — IT Python Language (Éditions ENI) — Score: 700/1000
Location	255 boulevard Pereire, 75017 Paris or remote (Zoom)

1. Target Audience

This training is designed for complete beginners, adults in career transition, developers wishing to add Python to their skill set, system and network administrators, as well as data processing professionals.

2. Prerequisites

- No prior programming knowledge is required
- Basic computer skills (navigating folders, using a web browser)

A placement test is offered before the start of training to verify the suitability of the course with the participant's profile.

3. Learning Objectives

Upon completion of this training, the participant will be able to:

- Write structured Python scripts with mastery of variables, types, and conversions (C1)
- Manipulate composite data structures: lists, tuples, dictionaries (C2)
- Implement conditional and iterative control structures (C3)
- Use standard input/output: print(), input(), f-string, format() (C4)
- Leverage built-in functions: len(), range(), random (C5)
- Define and call functions with parameters and return values (C6)

- Organize code into reusable modules and packages (C7, C8)
- Program using object-oriented principles: classes, inheritance, encapsulation, polymorphism (C9–C16)
- Implement exception handling: try/except, raise, custom exceptions (C17–C20)
- Interact with the file system and OS: os, pathlib, subprocess (C21–C23)

4. Target Certification

RS Code	RS6701
Title	Python Language
Certifier	Éditions ENI
Minimum Score	700 points out of 1000
Exam Format	Theoretical MCQ/SCQ + Practical assessment (Python coding)
Competencies	C1 to C23 distributed across 5 blocks

5. Detailed Program

DAY 1 — Python Language Fundamentals (7h)

ENI competencies targeted: C1, C4, C5

Morning (3h30) — Discovery and Getting Started

- Introduction to Python: history, philosophy, use cases in business
- Environment setup: Python, VS Code, useful extensions
- First script: print(), comments, running a program (C4-Cr1)
- Variables and data types: int, float, str, bool (C1-Cr1)
- Type conversions: str(), int(), float() (C1-Cr2)
- Operators: arithmetic, comparison, and logical

Afternoon (3h30) — Input/Output and Built-in Functions

- User input: input() and input loop with while (C4-Cr4, C4-Cr5)
- String formatting: format(), f-strings (C4-Cr2, C4-Cr3)
- len() function to count elements (C5-Cr1)
- range() function to generate sequences (C5-Cr3)
- random library: generating random numbers (C5-Cr4)
- Best practices: variable naming, PEP 8 conventions

Methods: Live demonstrations, progressive exercises, interactive demos

Practical Work:

- TP1: Interactive calculator with user input and result formatting
- TP2: Unit conversion program (km/miles, €/€, °C/°F) with f-strings
- TP3: Interactive quiz with final score and random question generation

Formative assessment: Group correction of practical work, end-of-day round table

DAY 2 — Data Structures and Control Flow (7h)

ENI competencies targeted: C2, C3, C5

Morning (3h30) — Control Structures

- Conditions: if, elif, else — syntax and indentation (C3-Cr1)
- Logical operators: and, or, not — combined conditions (C3-Cr1)
- for loop: iterating over sequences, range() (C3-Cr2)
- while loop: use cases, stopping conditions (C3-Cr2)
- Loop controls: break, continue (C3-Cr2)
- Handling common errors: missing values, unexpected types (C3-Cr4)

Afternoon (3h30) — Collections and Algorithms

- Lists: creation, indexing, slicing, main methods, append() (C2-Cr1)
- Extracting sublists with for, if, and the in operator (C2-Cr1)
- Tuples: differences from lists, immutability (C5-Cr2)

- Dictionaries: keys/values, items(), keys(), values() (C2-Cr2)
- Iterating over hierarchical structures with for (C2-Cr2)
- Designing efficient algorithms on data sets (C3-Cr3)

Methods: Guided exercises, collaborative problem solving

Practical Work:

- TP1: Mystery number game (while loop, break, random)
- TP2: Shopping list management with filtering and search
- TP3: Mini phone directory with nested dictionaries

Formative assessment: Mid-course MCQ (20 questions), group correction of practical work

DAY 3 — Functions, Modules, and File System (7h)

ENI competencies targeted: C6, C7, C8, C21, C22, C23

Morning (3h30) — Functions and Modularity

- Defining a function: def, parameters, return value (C6-Cr1 to Cr4)
- Variable scope: local vs global
- Default parameters, named arguments, *args, **kwargs
- Creating a module: package structure, naming (C7-Cr1 to Cr3)
- Importing a module: import, from, as (C8-Cr1, C8-Cr2)
- Resolving name conflicts with as (C8-Cr2)

Afternoon (3h30) — System Interaction

- os and sys modules: getcwd(), getlogin(), environment variables (C21-Cr1, C21-Cr2)
- pathlib module: Path, absolute(), exists(), is_dir(), is_file() (C22-Cr1, C22-Cr2)
- File reading/writing with pathlib: open(), readlines(), writelines() (C22-Cr3)
- Handling text and CSV files
- subprocess module: run() and capturing standard output (C23-Cr1)
- Best practices: context management with 'with'

Methods: Demonstrations, self-paced exercises, guided work on real files and scripts

Practical Work:

- TP1: Utility function library organized as a package
- TP2: File system exploration script with pathlib
- TP3: Mini-project — System log analyzer (reading, filtering, writing)

Formative assessment: Group correction of practical work, end-of-day round table

DAY 4 — Object-Oriented Programming and Exceptions (7h)

ENI competencies targeted: C9 to C20

Morning (3h30) — OOP: Fundamentals and Advanced Concepts

- Declaring a class: class, __init__(), attributes (C9-Cr1 to Cr4)
- Declaring methods: def, self, accessing attributes (C10-Cr1 to Cr3)
- Instantiating a class: creating objects, accessing members (C11-Cr1, C11-Cr2)
- Encapsulation: double underscore, properties, @property (C12-Cr1 to Cr3)
- Class members: @classmethod, cls, class attributes (C13-Cr1 to Cr4)
- Inheritance: subclasses, super(), inherited methods (C14-Cr1 to Cr5)
- Method overriding and calling the superclass (C15-Cr1, C15-Cr2)
- Polymorphism: isinstance(), same method on different objects (C16-Cr1, C16-Cr2)

Afternoon (3h30) — Exception Handling and Final Project

- Raising an exception: raise, parameter testing (C17-Cr1 to Cr3)
- Exception handler: try, except, identifying types (C18-Cr1, C18-Cr2)
- Multiple handlers: except block ordering, hierarchy (C19-Cr1 to Cr4)

- Custom exceptions: inheriting from Exception (C20-Cr1 to Cr3)
- finally block and robustness best practices

Methods: Guided autonomous integrative mini-project

Practical Work:

- TP1: Library management system (inheritance: Document, Book, DVD)
- TP2: OOP contact manager with encapsulation and custom exceptions
- TP3 (Final mini-project): Employee management application — OOP + files + exceptions

Formative assessment: Mini-project presentation, final MCQ of 30 questions

6. Teaching Methods and Resources

METHODS USED

- Theoretical input: interactive presentations with slideshow, concrete examples, and live demonstrations
- Hands-on practice: individual practical work, progressive exercises, integrative mini-projects at end of day
- Active learning: collaborative problem solving
- Theory/practice balance: 40% theory / 60% practice ratio
- Certification link: each practical exercise is tied to ENI assessment criteria (C1–C23)

TECHNICAL RESOURCES (Indicator 17)

- Equipped training room: video projector, flipchart, high-speed Wi-Fi
- Individual workstation per participant with VS Code and Python extensions
- Remote: Zoom with screen sharing, collaborative whiteboard
- Course materials provided to participants

7. Assessment Methods

Before the start of training, each participant completes a placement test assessing their general computer literacy.

ENTRY PLACEMENT

- Online placement test before training

IN-COURSE ASSESSMENTS

- Formative assessment: mid-course MCQ of 20 questions (Day 2)
- Summative assessment: final MCQ of 30 questions (Day 4)
- Pass criteria: achieve 60% on the final MCQ
- Certificate of completion: issued at the end of training, stating objectives, duration, and results
- Certification: RS6701 — Python Language (Éditions ENI) examination at end of training

SATISFACTION

- End-of-training satisfaction survey (trainer, content, logistics, achievement of objectives)

8. Accessibility for People with Disabilities

Pythonia is committed to making its training accessible to people with disabilities. Our disability advisor is available to assess necessary accommodations:

- Adaptation of teaching materials (font size, contrast, audio format)
- Adjusted examination conditions (extra time, adapted breaks)

- Referral to specialized partners: Agefiph, Cap Emploi, MDPH

Disability advisor contact: contact@pythonia.fr — 06 66 06 38 59

9. Commitment and Follow-up

- Attendance tracking: attendance sheet signed per half-day
- Dropout prevention: individualized follow-up, pace adjustment, follow-up in case of absence
- Complaints: any complaint may be sent by email to contact@pythonia.fr. It will be processed within 48 business hours.

PythonIA — Python & Artificial Intelligence Training Center
255 boulevard Pereire, 75017 Paris | contact@pythonia.fr | 06 66 06 38 59
SIRET: 52349020900039 — NDA: 11757391975