

Python Beginners

About this course

Python is rapidly becoming a language of choice for many programmers. Python is a versatile programming language which can be used to power websites, build games, and program hardware. This course teaches participants how to programme using Python. No prior experience in programming is needed.

Duration: 5 days

Participants: Min 5, Max 15, Optimal 12

Certification: Option to take the Microsoft Technical Associate 98-381 Introduction to Programming using Python

Learning outcomes

Upon completion of this course participants will be able to:

- Perform Operations using data types & Operators
- Control Flow and Decision Loops
- Perform input and output operations
- Document and structure code
- Perform troubleshooting & Error Handling
- Perform Operations using modules and Tools

Who should attend

This course is suitable to those that are new to programming and wish to learn programming through Python. It is also open to experienced programmers who wish to learn Python as a new programming language.

Course syllabus

Module 1: Introduction to Python

- What is Python?
- Installing your Python IDE.

Module 2: Operations using data types & Operators

- Evaluate an expression to identify the data type Python will assign to each variable. Data types include str, int, float, and bool
- Convert between and work with data types. Type casting; constructing data structures; indexing and slicing operations

- Determine the sequence of execution based on operator precedence. Assignment; Comparison; Logical; Arithmetic; Identity (is); Containment (in)
- Select the appropriate operator to achieve the intended result. Assignment; Comparison; Logical; Arithmetic; Identity (is); Containment (in)

Module 3: Control Flow with Decisions & Loops

- Construct and analyse code segments that use branching statements. if; elif; else; nested and compound conditionals
- Construct and analyse code segments that perform iteration. while; for; break; continue; pass; nested loops and loops that include compound conditionals

Module 4: Perform Input & Output Operations

- Construct and analyse code segments that perform file input and output operations. open; close; read; write; append; check existence; delete; with statement
- Construct and analyse code segments that perform console input and output operations. Read input from console; print formatted text; use of command line arguments

Module 5: Document & Structure Code

- Document code segments using comments and documentation strings. Use of indentation and white space; comments and documentation strings; pydoc
- Construct and analyse code segments that include function definitions. Call signatures; default values; return; def; pass

Module 6: Troubleshooting & Error Handling

- Analyse, detect, and fix code segments that have errors. Syntax errors; logic errors; runtime errors
- Analyse and construct code segments that handle exceptions. Try; except; else; finally; raise

Module 7: Operations using Modules and Tools

- Perform basic operations using built-in modules. math; datetime; io; sys; os; os.path; random
- Solve complex computing problems by using built-in modules. math; datetime; random.