

Introduction to Python Programming

Introduction to Python Programming

Quick?reference guide for beginners

1. What Is Python?

Feature	Description
High?level	Abstracts away low?level details (memory management, pointers).
Interpreted	Code is executed line?by?line by the Python interpreter; no separate compile step.
Dynamic typing	Variable types are inferred at runtime.
Multi?paradigm	Supports procedural, object?oriented, and functional styles.
Extensive standard library	“Batteries?included” modules for file I/O, networking, web, data, etc.
Cross?platform	Runs on Windows, macOS, Linux, and many embedded systems.

2. Setting Up the Environment

1. Install Python

- Download from <https://python.org> (latest stable release, e.g., 3.12).
- Verify: `python --version` (or `python3 --version` on Unix).

2. Choose an editor / IDE

- Light: VS?Code, Sublime Text, Atom.
- Full?featured: PyCharm, Thonny, Spyder, Jupyter Notebook/Lab.

3. Virtual environments (recommended)

```
# Create
python -m venv myenv
# Activate
# Windows
```

Introduction to Python Programming

```
myenv\Scripts\activate
# macOS / Linux
source myenv/bin/activate
# Install packages
pip install numpy pandas
```

3. Basic Syntax & Concepts

3.1. Hello, World!

```
print("Hello, World!")
```

3.2. Variables & Data Types

Type	Literal Example	Common Operations
int	42	+ - * // % **
float	3.14	+ - * / **
str	"text"	concatenation +, formatting
bool	True/False	logical and, or, not
list	[1, 2, 3]	indexing, slicing, append, pop
tuple	(1, 2)	immutable sequence
dict	{'key': 'value'}	key?value lookup
set	{1, 2, 3}	membership, set ops

3.3. Control Flow

```
# Conditional
if score >= 90:
    grade = 'A'
elif score >= 80:
    grade = 'B'
else:
    grade = 'C'

# Loop - for (iterates over any iterable)
```

Introduction to Python Programming

```
for i in range(5):           # 0,1,2,3,4
    print(i)
```

```
# Loop - while
count = 0
while count < 5:
    print(count)
    count += 1
```

3.4. Functions

```
def greet(name: str, times: int = 1) -> None:
    """Print a greeting `times` times."""
    for _ in range(times):
        print(f"Hello, {name}!")
```

```
greet("Alice", 3)
```

- **Parameters** can have default values and type hints.
- Use `return` to send a value back.