Introduction to Python

Learning Objectives

By the end of this section, you will be able to:

- Understand Python's origins and core philosophy
- Install Python on your computer
- Set up a development environment with an IDE or text editor
- Write and run your first Python program
- Recognize basic Python syntax elements

Python's History and Philosophy

Origins of Python

Python was created by Guido van Rossum and first released in 1991. The language was named after the British comedy group Monty Python, not the snake! Guido designed Python as a successor to the ABC language, aiming to create a language that emphasized readability and simplicity.

Python Philosophy

Python's design philosophy is captured in "The Zen of Python," a collection of 19 guiding principles. You can view these principles by typing this command in your Python interpreter:

```python
import this
```

Some key principles include:

- **Readability counts** - Clean, readable code is a priority
- **Simple is better than complex** - Solutions should be straightforward
- **Explicit is better than implicit** - Code should be clear about what it's doing
- **There should be one obvious way to do it** - Python favors a single, clear approach to solving problems

Python Versions

There are two major Python versions:

- **Python 2.x**: Officially retired on January 1, 2020
- **Python 3.x**: The current and future version of Python

We'll be using Python 3 throughout this course, as Python 2 is no longer supported.

Installing Python and Setting Up Your Environment

Installing Python

Windows Installation

1. Visit the official Python website
2. Download the latest Python 3 installer for Windows
3. Run the installer
4. **Important**: Check the box that says "Add Python to PATH" before clicking Install
5. Click "Install Now"

macOS Installation

1. Visit the official Python website
2. Download the latest Python 3 installer for macOS
3. Run the installer package and follow the instructions

Alternatively, if you have Homebrew installed:

```
brew install python
```

Linux Installation

Most Linux distributions come with Python pre-installed. To check if Python is installed:

```
python3 --version
```

If Python is not installed:

```
# For Debian/Ubuntu
sudo apt update
sudo apt install python3 python3-pip

# For Fedora
sudo dnf install python3 python3-pip

# For Arch Linux
sudo pacman -S python python-pip
```

Verifying Your Installation

Open a terminal or command prompt and type:

```
python3 --version
# or on some Windows systems
python --version
```

You should see the version number displayed. If you see an error, Python might not be installed correctly or may not be in your system's PATH.

Python IDEs and Text Editors

What is an IDE?

An Integrated Development Environment (IDE) is a software application that provides comprehensive facilities for software development, including code editing, debugging, and execution tools.

Popular Python IDEs and Editors

1. Visual Studio Code (VS Code)

Setup Instructions:

1. Download VS Code from code.visualstudio.com
2. Install the Python extension:
    - Open VS Code
    - Go to Extensions (or press Ctrl+Shift+X)
    - Search for "Python"
    - Install the Python extension by Microsoft

Key Features:

- Free and open-source
- Lightweight but powerful

2. Other Good Options

- **PyCharm**: Specifically designed for Python with powerful debugging tools
- **IDLE**: Comes bundled with Python, simple but useful for beginners
- **Jupyter Notebook**: Excellent for data science and learning
- **Sublime Text**: Fast, lightweight text editor (requires configuration for Python)
- **Atom**: Customizable editor with Python packages

Recommendation for Beginners

If you're just starting out, I recommend:

1. **VS Code** for a general-purpose editor with great Python support
2. **PyCharm Community Edition** for a dedicated Python IDE
3. **IDLE** for a simple, no-setup-required option

Running Your First Python Program

Using the Interactive Shell

The Python interpreter can be used in interactive mode for quick experiments:

1. Open a terminal or command prompt
2. Type `python` or `python3` and press Enter
3. You'll see the Python prompt `>>>`
4. Type a Python command and press Enter to execute it:

```
>>> print("Hello, Python!")
Hello, Python!
```

To exit the interactive shell, type `exit()` or press Ctrl+Z (on Windows) or Ctrl+D (on Unix-based systems).

Creating and Running a Python Script

1. Open your chosen editor or IDE
2. Create a new file with a `.py` extension (e.g., `first_program.py`)
3. Write the following code:

```python
# This is my first Python program
print("Hello, Python World!")
```

4. Save the file
5. Run the program:

From terminal/command prompt:

```
# Navigate to the folder containing your file
cd path/to/your/folder

# Run the script
python3 first_program.py
```

From VS Code:

- Open the file
- Click the "Run" button (triangle) in the top right, or
- Right-click in the editor and select "Run Python File in Terminal"

You should see the output in the terminal or console:

```
Hello, Python World!
```

Python Syntax Basics

Comments

Comments in Python start with the `#` character:

```python
# This is a single-line comment

"""
This is a multi-line comment or docstring
It can span multiple lines
"""
```

Variables and Assignment

Variables in Python don't need explicit type declarations:

```python
# Variable assignment
name = "John"
age = 25
height = 1.75
is_student = True

# Print variables
print(name)
print(age)
```

Basic Data Types

```python
# Numeric types
integer_number = 42
float_number = 3.14159

# Strings
text = "Hello, Python!"
another_text = 'Single quotes work too'

# Boolean
is_true = True
is_false = False

# None type (similar to null in other languages)
empty_value = None
```

Basic Operations

```
# Arithmetic operations
sum_result = 10 + 5  # Addition
difference = 10 - 5  # Subtraction
product = 10 * 5     # Multiplication
quotient = 10 / 5    # Division
remainder = 10 % 3   # Modulo (remainder)
power = 10 ** 2      # Exponentiation

# String operations
greeting = "Hello"
name = "Python"
message = greeting + ", " + name + "!"  # String concatenation
repeated = "Python" * 3  # String repetition: "PythonPythonPython"

# Print with multiple arguments
print("The sum is:", sum_result)
```

Indentation

Python uses indentation (whitespace at the beginning of a line) to define code blocks:

```python
# Example of indentation
if age >= 18:
    print("You are an adult.")  # This code is part of the if block
    print("You can vote.")     # This code is also part of the if block
print("This will print regardless of age.")  # Outside the if block
```

Important: Python is strict about indentation! Use consistent spacing (typically 4 spaces per indentation level).

Practice Exercises

1. **Setup Exercise**: Install Python and an IDE of your choice, then verify your installation by running Python in interactive mode.
2. **Hello World Plus**: Create a Python file that:
   - Prints a greeting with your name
   - Calculates and prints your age in days (your_age * 365)
   - Prints the result of 7 raised to the power of 3
3. **Syntax Explorer**: Create a Python file that demonstrates:
   - At least 3 different data types
   - String concatenation
   - A basic mathematical calculation
   - A multi-line comment explaining your code

Key Takeaways

- Python is designed to be readable and straightforward
- Python 3 is the current version you should be using
- Python is an interpreted language - code is executed line by line
- Indentation in Python is not just for style - it's required syntax
- Python's interactive shell is great for quick experiments
- Python syntax is generally clean and uses fewer special characters than many other languages

Next Steps

Now that you understand the basics of Python syntax and environment setup, you're ready to explore Python's fundamental concepts such as control flow, functions, and data structures!