**Introduction to Python**

1. **Python Basics**

Python is a high-level, interpreted programming language known for its readability and simplicity. Key features that make Python popular include:

* **Readable and Concise Syntax**: Python's syntax is designed to be easy to read and write.
* **Interpreted Language**: Python code is executed line-by-line, which makes debugging easier.
* **Dynamically Typed**: Variables in Python do not need an explicit declaration to reserve memory space.
* **Extensive Standard Library**: Python has a rich set of libraries and frameworks for various tasks.
* **Cross-Platform**: Python can run on various operating systems like Windows, macOS, and Linux.

**Use Cases**:

* **Web Development**: Frameworks like Django and Flask.
* **Data Science and Machine Learning**: Libraries like pandas, NumPy, and scikit-learn.
* **Automation and Scripting**: Automating repetitive tasks.
* **Software Development**: Prototyping and building applications.

1. **Installing Python**

**Windows:**

1. Download the Python installer from [python.org](https://www.python.org/downloads/).
2. Run the installer.
3. Complete the installation.

**macOS**:

1. Install Homebrew if not already installed:

>>> /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

1. Install Python:

>>> brew install python

**Linux**:

1. Install Python using your package manager. For example, on Ubuntu:

>>> sudo apt update

>>> sudo apt install python3

**Verify Installation**:

>>> python --version

**or**

>>> python3 –version

**Set Up a Virtual Environment**:

1. Install virtualenv:

>>> pip install virtualenv

1. Create a virtual environment:

>>> virtualenv venv

1. Activate the virtual environment:
   * **Windows**:

>>> venv\Scripts\activate

* + **macOS/Linux**:

>>> source venv/bin/activate

**3. Python Syntax and Semantics**

**Write a simple Python program that prints "Hello, World!" to the console. Explain the basic syntax elements used in the program.**

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

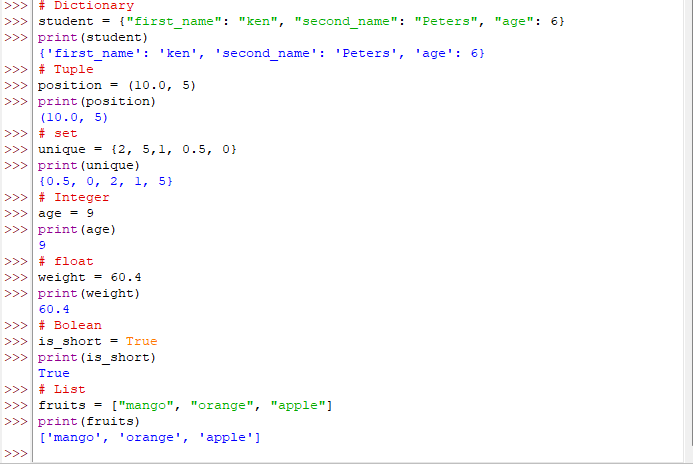
**Explanation**:

* print is a built-in function that outputs text to the console.
* "Hello, World!" is a string, a sequence of characters enclosed in double-quotes.

**4. Data Types and Variables**

**Basic Data Types**:

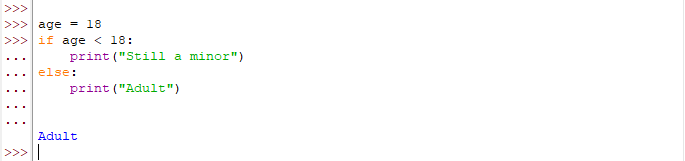
* **int**: Integer numbers.
* **float**: Floating-point numbers.
* **str**: Strings, a sequence of characters.
* **bool**: Boolean values, True or False.
* **list**: Ordered, mutable collection of items.
* **dict**: Unordered collection of key-value pairs.
* **tuple**: Ordered, immutable collection of items.
* **set**: Unordered collection of unique items.



**5. Control Structures**

**Explain the use of conditional statements and loops in Python. Provide examples of an if-else statement and a for loop.**

**Conditional Statements**: Conditional statements allow you to execute different code blocks based on conditions.



**Loops**: Loops allow you to execute a block of code multiple times.

**Example of a for loop**:

A white background with black dots

Description automatically generated

**6. Functions in Python**

Functions are reusable blocks of code that perform a specific task. They help in organizing code, making it modular and reusable.

**Example**:



**7. Lists and Dictionaries**

**Describe the differences between lists and dictionaries in Python. Write a script that creates a list of numbers and a dictionary with some key-value pairs, then demonstrates basic operations on both.**

**Lists**:

* Ordered collection of items.
* Items can be accessed by index.
* Mutable.

**Dictionaries**:

* Unordered collection of key-value pairs.
* Items can be accessed by keys.
* Mutable.

**Example**:

A computer screen shot of a computer code

Description automatically generated

**8. Exception Handling**

**What is exception handling in Python? Provide an example of how to use try, except, and finally blocks to handle errors in a Python script.**

Exception handling is a way to handle errors gracefully without terminating the program abruptly.

**Example**:

A computer screen shot of text

Description automatically generated

**9. Modules and Packages**

**Explain the concepts of modules and packages in Python. How can you import and use a module in your script? Provide an example using the math module.**

**Modules**: A module is a file containing Python code that can be imported and used in other scripts.

**Packages**: A package is a collection of modules organized in directories.

**Example using math module**:

A white background with black text

Description automatically generated

**10. File I/O**

**How do you read from and write to files in Python? Write a script that reads the content of a file and prints it to the console, and another script that writes a list of strings to a file.**

**Reading from a file**:

A close-up of a white background

Description automatically generated

A white background with black and green text

Description automatically generated