🐍 **Python** is a high-level, versatile programming language known for its simplicity and readability. It was created by **Guido van Rossum** and first released in **1991**. Here's what makes Python stand out:

**💡 Key Features**

* **Easy to Learn & Read**: Python’s syntax is clean and resembles natural language, making it beginner-friendly.
* **Multi-Paradigm**: Supports procedural, object-oriented, and functional programming styles.
* **Interpreted Language**: You can run code directly without compiling, which speeds up development.
* **Cross-Platform**: Works on Windows, macOS, Linux, and even Raspberry Pi.

**🛠️ What Can You Do With Python?**

* **Web Development**: Frameworks like Django and Flask make building websites a breeze.
* **Data Science & Machine Learning**: Libraries like NumPy, Pandas, and TensorFlow are industry staples.
* **Automation & Scripting**: Great for writing scripts to automate repetitive tasks.
* **Game Development**: Tools like Pygame let you build simple games.
* **App Development**: You can even build desktop and mobile apps.

**🧠 Why Developers Love It**

* Huge community and tons of resources.
* Massive library ecosystem

Absolutely! Python has a rich set of built-in data types that help you store and manipulate different kinds of information. Here's a breakdown of the main categories:

**🔢 Numeric Types**

Used for numbers:

* int: Integer values (e.g., 5, -42)
* float: Floating-point numbers (e.g., 3.14, -0.001)
* complex: Complex numbers (e.g., 2 + 3j)

**📚 Sequence Types**

Ordered collections of items:

* str: Strings of text (e.g., "hello")
* list: Mutable sequences (e.g., [1, 2, 3])
* tuple: Immutable sequences (e.g., (1, 2, 3))
* range: Sequence of numbers, often used in loops

**🗺️ Mapping Type**

* dict: Key-value pairs (e.g., {"name": "Alice", "age": 30})

**✅ Boolean Type**

* bool: Represents True or False

**🧺 Set Types**

Collections of unique items:

* set: Mutable set (e.g., {1, 2, 3})
* frozenset: Immutable set

**💾 Binary Types**

Used for binary data:

* bytes: Immutable byte sequences
* bytearray: Mutable byte sequences
* memoryview: Memory view object for binary data

**🚫 Special Type**

* NoneType: Represents the absence of a value (None)

If you want to check the type of any variable in Python, just use the type() function:

x = 42

print(type(x)) # Output: <class 'int'>

# Taking two inputs from the user

num1 = float(input("Enter the first number: "))

num2 = float(input("Enter the second number: "))

# Performing arithmetic operations

print("\n--- Arithmetic Operations ---")

print(f"Addition: {num1} + {num2} = {num1 + num2}")

print(f"Subtraction: {num1} - {num2} = {num1 - num2}")

print(f"Multiplication: {num1} \* {num2} = {num1 \* num2}")

# Handling division by zero

if num2 != 0:

print(f"Division: {num1} / {num2} = {num1 / num2}")

print(f"Modulus: {num1} % {num2} = {num1 % num2}")

print(f"Exponentiation: {num1} \*\* {num2} = {num1 \*\* num2}")

print(f"Floor Division: {num1} // {num2} = {num1 // num2}")

else:

print("Division, modulus, and floor division by zero are not allowed.")

Enter the first number: 2

Enter the second number: 4

--- Arithmetic Operations ---

Addition: 2.0 + 4.0 = 6.0

Subtraction: 2.0 - 4.0 = -2.0

Multiplication: 2.0 \* 4.0 = 8.0

Division: 2.0 / 4.0 = 0.5

Modulus: 2.0 % 4.0 = 2.0

Exponentiation: 2.0 \*\* 4.0 = 16.0

Floor Division: 2.0 // 4.0 = 0.0

=== Code Execution Successful ===