

# Python Programming — Day 2: Variables, Data Types & Operators

**Course:** Python Programming (Basic to Advanced)

**Presented by:** Mahak & MahakTech Team

---

## ◆ 1. Variables in Python

**Definition:**

A variable in Python is a **named storage location** that can hold a value, which you can refer to later by name.

**Key Points:**

- You **do not** explicitly declare the data type of a variable in Python: the interpreter infers it when you assign a value. [python-course.eu+1](#)
- You assign a value using the `=` operator:

```
age = 18  
name = "Mahak"
```

- Variables are **references** to objects/data in memory rather than fixed-typed slots. [python-course.eu](#)
- Valid variable names: start with a letter or underscore, can include letters/numbers/underscore; case-sensitive. [python-course.eu](#)
- Avoid using reserved words (keywords) like `if`, `for`, `while`, `True`, `None`, etc. [python-course.eu](#)

**Example:**

```

x = 10
y = 20
sum_value = x + y
print(sum_value) # Outputs: 30

```



## 2. Data Types

In Python, every value has a data type, which tells the interpreter what kind of value it is, and what operations are valid. [GeeksforGeeks+1](#)

### Common Built-in Data Types:

Data Type	Description	Example
<b>int</b>	Whole numbers (positive, negative, zero)	age = 18
<b>float</b>	Numbers with a decimal point / fractional values	pi = 3.14
<b>complex</b>	Complex numbers with real and imaginary parts	z = 4 + 2j
<b>str</b>	Sequence of characters (text)	name = "Mahak"
<b>bool</b>	Boolean values: <code>True</code> or <code>False</code>	is_valid = True
<b>list</b>	Ordered, mutable collection of items	fruits = ["apple", "banana"]
<b>tuple</b>	Ordered, immutable sequence of items	coords = (10, 20)
<b>set</b>	Unordered collection of unique items	ids = {101, 102, 103}
<b>dict</b>	Key-value mapping (dictionary)	student = {"name": "Mahak", "age": 18}

### Checking Type:

```

print(type(name)) # <class 'str'>
print(type(pi)) # <class 'float'>

```

## Type Conversion (Casting):

```
num = 10
flt = float(num) # Converts to 10.0
txt = str(num) # Converts to "10"
```

## 3. Operators

Operators allow you to perform operations on variables and values.

### Arithmetic Operators:

- `+` : Addition
- `-` : Subtraction
- `*` : Multiplication
- `/` : Division (always returns float)
- `//` : Floor division (integer result)
- `*` : Exponentiation (power)
- `%` : Modulus (remainder)

### Example:

```
a = 10
b = 3
print(a + b) # 13
print(a - b) # 7
print(a * b) # 30
print(a / b) # 3.333333333333335
print(a // b) # 3
print(a ** b) # 1000
print(a % b) # 1
```

## Comparison Operators:

- `==`, `!=`, `>`, `<`, `>=`, `<=`

## Assignment Operators:

- `=`, `+=`, `-=`, `*=`, `/=`, etc.

## Logical Operators:

- `and`, `or`, `not`

## Important Note:

When mixing types (e.g., int + str) you may get an error unless converted explicitly.

## 📌 4. Putting It All Together — Mini Program Example

```
# Student information
student_name = "Mahak"
student_age = 18
student_grade = 9.5

print("Name:", student_name)
print("Age:", student_age)
print("Grade:", student_grade)

# Some calculations
next_year_age = student_age + 1
print("Next year age will be:", next_year_age)

height_m = 1.65
height_cm = height_m * 100
print("Height in cm:", height_cm)
```



## 5. Recap — What You Learned Today

- ✓ Variables: naming, assignment, storage of data
  - ✓ Data types: int, float, str, bool, list, tuple, set, dict
  - ✓ Operators: arithmetic, comparison, logical, assignment
  - ✓ Type conversions & checking types
- 



## Next Class Preview

- Conditional statements (`if`, `elif`, `else`)
  - Loops: `for`, `while`
  - More on input/output (getting user input)
  - Simple problem-solving practice
- 



## Assignment

✍ Write a Python program that:

1. Asks the user for their **name**, **age**, and **favorite number**.
2. Prints a greeting with their name.
3. Calculates and prints:
  - Their age after 5 years.
  - Their favorite number squared.
4. Uses at least one comment explaining what a piece of code does.

Example:

```
# This program asks for user details and computes values.  
name = input("Enter your name: ")  
age = int(input("Enter your age: "))  
fav_num = int(input("Enter your favorite number: "))  
  
print("Hello, " + name + "!"")
```

```
age_after_5 = age + 5
print("In 5 years you will be", age_after_5, "years old.")

fav_num_squared = fav_num ** 2
print("Your favorite number squared is", fav_num_squared)
```