**PYTHON BASICS NOTES**

**print() FUNCTION — Output in Python**

**What it does:**  
The print() function displays output (text, numbers, variables, etc.) to the screen.  
It’s your main way of talking *to the user*.

**Syntax:**

print(object, sep=' ', end='\n')

**Parameters:**

* object: What you want to display (can be strings, numbers, variables, etc.)
* sep: Separator between multiple objects (default = space ' ')
* end: What to print at the end (default = new line \n)

**Examples:**

print("Hello World!") # simple text

print("Python", "is", "cool") # default separator is space

print("Python", "is", "cool", sep='-') # changes separator

print("Hello", end=' ') # stays on same line

print("World") # continues from same line

**Output:**

Hello World!

Python is cool

Python-is-cool

Hello World

**input() FUNCTION — Taking User Input**

**What it does:**  
input() lets you take data from the user via the keyboard.  
It **always** returns data as a string (str type), even if you enter a number.

**Syntax:**

variable = input("Prompt message: ")

**Example:**

name = input("Enter your name: ")

print("Hey", name, "welcome to Python!")

**Type conversion example:**

age = int(input("Enter your age: ")) # convert input to integer

print("Next year, you’ll be", age + 1)

**VARIABLES — Data Containers**

**What they are:**  
Variables store data values in memory. You don’t need to declare their type explicitly (Python is dynamically typed).

**Rules for naming variables:**

* Must start with a letter or underscore \_
* Can’t start with a number
* Can only contain letters, digits, or underscores
* Are **case-sensitive** (age ≠ Age)

**Examples:**

name = "Mellow"

age = 20

is\_coder = True

height = 5.8

**Valid vs Invalid Names:**

# Valid

user\_name = "Alex"

\_age = 21

A1 = 45

# Invalid

1name = "Jake" # can’t start with a number

user-name = "Leo" # no hyphens allowed

**DATA TYPES**

Everything in Python is an **object**, even data types.

**Common built-in types:**

| **Type** | **Example** | **Description** |
| --- | --- | --- |
| int | 10, -5, 0 | Whole numbers |
| float | 3.14, -2.7 | Decimal numbers |
| str | "Hello", 'Python' | Text or characters |
| bool | True, False | Logical values |
| list | [1, 2, 3], ['a', 'b'] | Ordered, changeable collection |
| tuple | (1, 2, 3) | Ordered, unchangeable collection |
| set | {1, 2, 3} | Unordered, unique items |
| dict | {"name": "Mellow", "age": 20} | Key-value pairs |

**Examples:**

a = 10 # int

b = 3.14 # float

c = "Python" # string

d = True # bool

e = [1, 2, 3] # list

f = (4, 5, 6) # tuple

g = {"x", "y", "z"} # set

h = {"name": "Mellow", "age": 20} # dict

You can check types using:

print(type(a))

**OPERATORS**

Operators are special symbols that perform operations on variables and values.

**1. Arithmetic Operators**

x = 10

y = 3

print(x + y) # 13 addition

print(x - y) # 7 subtraction

print(x \* y) # 30 multiplication

print(x / y) # 3.333 division (float)

print(x // y) # 3 floor division (integer result)

print(x % y) # 1 modulus (remainder)

print(x \*\* y) # 1000 exponent (power)

**2. Comparison Operators**

Used to compare values → return True or False

a, b = 5, 10

print(a == b) # False

print(a != b) # True

print(a > b) # False

print(a < b) # True

print(a >= b) # False

print(a <= b) # True

**3. Logical Operators**

Used for combining conditions

x, y = True, False

print(x and y) # False

print(x or y) # True

print(not x) # False

**4. Assignment Operators**

x = 10

x += 5 # x = x + 5 → 15

x -= 3 # x = x - 3 → 12

x \*= 2 # x = x \* 2 → 24

x /= 4 # x = x / 4 → 6.0

**5. Membership Operators**

fruits = ["apple", "banana", "cherry"]

print("apple" in fruits) # True

print("grape" not in fruits) # True

**6. Identity Operators**

They check **memory location**, not just value.

a = [1, 2, 3]

b = [1, 2, 3]

c = a

print(a is b) # False (different memory)

print(a is c) # True (same memory)

**Small Practice Code**

**Code:**

name = input("Enter your name: ")

age = int(input("Enter your age: "))

print("Hey", name + "!")

print("You are", age, "years old.")

next\_age = age + 1

print("Next year, you’ll be", next\_age)

# Operators demo

num1 = 15

num2 = 4

print("Addition:", num1 + num2)

print("Power:", num1 \*\* num2)

print("Division:", num1 / num2)

print("Floor Division:", num1 // num2)

**Sample Output:**

Enter your name: mellow

Enter your age: 20

Hey mellow!

You are 20 years old.

Next year, you’ll be 21

Addition: 19

Power: 50625

Division: 3.75

Floor Division: 3