**1. Variables**

Variables are containers for storing data, and they follow these rules:

* Variable names must start with a letter, underscore (\_), or a letter.
* They are case-sensitive.
* You cannot use reserved words (e.g., if, for, while).

Variable Declaration Example:

# name=value

x = 5 # Integer

username = “Bob” # String

pi = 3.14 # Float

**2. Data Types**

In Python, the basic data types are as follows:

* Integer: Represents whole numbers.

a = 10 #integer

* Float: Represents numbers with decimal points.

b = 3.14 # Float

* String: Represents text data, enclosed in single (') or double (") quotes.

name = “Alice” # String

* Boolean: Represents True or False values.

is\_active = True # Boolean

* None: Represents Empty or No values.

is\_none = None # None

**3. Casting**

Type casting refers to converting a variable from one data type to another. This is useful when you need to perform operations on different data types or when you want to ensure that your data is in the correct format.

Common Type Casting Functions

int(): Converts a value to an integer.

x = 5.7

y = int(x)

print(y) # 5

float(): Converts a value to a float.

x = 5

y = int(x)

print(y) # 5.0

str(): Converts a value to a string.

x = 10

y = str(x)

print(y) # “10”

**4. Comparator**

Common Comparison Operators

Equal to (==): Checks if two values are equal.

a = 5

b = 5

print(a==b) # True

Not equal to (!=): Checks if two values are not equal.

a = 5

b = 3

print(a!=b) # True

Greater than (>): Checks if the left value is greater than the right value.

a = 5

b = 5

print(a>b) # True

Less than (<): Checks if the left value is less than the right value.

a = 5

b = 8

print(a<b) # True

Greater than or equal to (>=): Checks if the left value is greater than or equal to the right value.

a = 5

b = 5

print(a>=b) # True

Less than or equal to (<=): Checks if the left value is less than or equal to the right value.

a = 5

b = 10

print(a<=b) # True

**5. Logical Operators in Python**

Logical operators are used to combine conditional statements or evaluate boolean expressions.

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | A and B | A or B |
| T | T | T | T |
| T | F | F | T |
| F | T | F | T |
| F | F | F | F |

**and**

The and operator returns True if both conditions are True. If either condition is False, it returns False.

a = 5

b = 10

print(a>0 and b>5) # True

**is**

The is operator checks if two variables point to the same object in memory (i.e., it checks for identity). It returns True if both variables refer to the same object, and False otherwise.

a = [1,2,3]

b = a

print(a is b) # True

**Note:** Use == to check for value equality, which compares the values rather than the identities of the objects.

a = [1,2,3]

b = a

c = [1,2,3]

print(a is b) # True, because a and b are the same object

print(a is c) # False, because a and c are different objects

print(a == b) # True, because both lists have the same values

print(a == c) # True, because both lists have the same values, even though they are different objects

**not**

The not operator negates a boolean value. It returns True if the value is False, and False if the value is True.

a = 5

b = 10

print(a is not b) # True

**Combining Logical Operators**

You can combine these operators in complex conditions:

a = 5

b = 10

c = None

print(a > 0 and (b > 5 or c is None)) # True

**\*input()**

The input() function in Python is used to get user input. When called, it displays a prompt to the user and waits for the user to enter something from the keyboard. Once the user presses enter, it returns the input as a string.

user\_input = input(“Please enter your name: ”)

print(“Hello, ” + user\_input + “!”) # True