Variables in Python

Variable: A variable is simply a name that stores some value in a program. A variable like a container or a labeled box that holds data — like numbers, text, etc. — so you can use or change it later in your code.

Why We Use Variables in Python:

- To store data for later use
- To reuse values without repeating them
- To make code easier to read and understand
- To change values dynamically in the program
- To handle input from users or other sources
- To avoid repetition in calculations or operations
- To organize data logically and cleanly
- To perform operations using stored values (e.g., math)

Assignment Statement

Variables allow programs to refer to values using names rather than memory locations. Ex: age refers to a person's age, and birth refers to a person's date of birth.

A statement can set a variable to a value using the assignment operator (=). Note that this is different from the equal sign of mathematics. Ex: age = 6 or birth = "May 15". The left side of the assignment statement is a variable, and the right side is the value the variable is assigned.

Variable Naming Rules

- A variable name can consist of letters, digits, and underscores and be of any length.
- The name cannot start with a digit. Ex: 101class is invalid. Also, letter case matters. Ex: Total is different from total.
- Python's style guide recommends writing variable names in snake case, which is all lowercase with underscores in between each word, such as first_name or total_price.
- A name should be short and descriptive, so words are preferred over single characters in programs for readability. Ex: A variable named count indicates the variable's purpose better than a variable named c.

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
asynch	elif	if	or	yield

Variables in Python

How to Create a Variable?

You just assign a value using the = sign.

```
a = 5
b = 3.2
c = "Hello"
```

Here:

- a is an integer variable
- b is a float variable
- c is a string variable

Ways to Declare a Variable in Python

Method	Description	Example
1. Single Assignment	Assign a single value to one variable	x = 10
2. Multiple Assignment	Assign multiple values to multiple variables	a, b = 5, 6
3. Same Value to Multiple Vars	Assign one value to multiple variables	x = y = z = 100
4. Unpacking Collection	Assign multiple values from a list/tuple	a, b, c = [1, 2, 3]
5. Type Casting	Convert and assign data types	age = int("25")
6. User Input Assignment	Assign value from user input	name = input("Enter name: ")
7. Type Hinting (Optional)	Declare type using hints (Python 3.6+)	score: float = 99.5

A. Single Assignment

Assigning **one value** to **one variable** at a time.

Example:

```
# Single Assignment
name = "Alice"
age = 20
height = 5.6

print(name)
print(age)
```

```
print(height)
```

Output:
Alice
20
5.6

B. Multiple Assignment

```
# Multiple Assignment
a, b, c = 10, 20, 30

print(a)
print(b)
print(c)
```

Output:
10
20
30

Here:
- a gets 10
- b gets 20
- c gets 30

C. Same Value to Multiple Variable

You can assign **the same value** to **many variables** at once.

Example:

```
# Same value to multiple variables
x = y = z = 100
rint(x)
print(y)
print(z)
```

Output:
100
100
100

NOTE: All three variables hold the same value 100.

D. Unpacking Collection

Variables in Python

If you have a **list, tuple, or collection**, you can "unpack" (split) its values into separate variables.

Example:

```
# Unpacking a list
fruits = ["apple", "banana", "cherry"]
a, b, c = fruits

print(a)
print(b)
print(c)
```

Output:

```
apple
banana
cherry
```

NOTE: Works with tuples and other collections too.

E. Type Casting

Changing (or converting) the data type of a variable manually.

Example:

```
# Type Casting
a = "10"      # string
b = int(a)    # convert to integer
c = float(a)  # convert to float

print(a, type(a))
print(b, type(b))
print(c, type(c))
```

Output:

```
10 <class 'str'>
10 <class 'int'>
10.0 <class 'float'>
```

Common type casting functions:

- int() → convert to integer
- float() → convert to float
- str() → convert to string

F. User Input Assignment

Take input from the user and assign it to a variable.
By default, input() takes data as a string, so you may need type casting.

Variables in Python

Example:

```
# User Input Assignment
name = input("Enter your name: ")
age = int(input("Enter your age: "))

print("Hello", name)
print("You are", age, "years old.")
```

Output (example):

```
Enter your name: Rahul
Enter your age: 18
Hello Rahul
You are 18 years old.
```