

Variables

In programming, a variable is a container (storage area) to hold data. To indicate the storage area, each variable should be given a unique name (identifier). Variable names are just the symbolic representation of a memory location.

In this chapter, you will learn about Python variables and data types and the operation that can be performed on them. In a program you can put any value into it, and then retrieve the value later for use in calculation. Each variable will contain different values at different times, depending on the requirements.

Rules of variable name declarations:

The variable value can be change during the execution of program. The name you assign to a variable is called identifier. Some rules of variable declaration are following.

1. Identifier/Variable may contain uppercase and lowercase Letters, Numbers, or underscore. The first character after dollar sign must be a letter.
2. Identifier/Variable cannot contain special character and spaces.
3. Identifier/Variable are case sensitive.
4. Identifier/Variable cannot contain reserve words or keywords of Python language.

Convention Of variable:

Syntax:

VariableName = Content ;

Example:

Father_Name = Content;

ConvanceAllownace = Content;

Variables can store data of different types, and different data types can do different things. A variable's specific data type is very important in programming because the data type helps determine the manner in which the value is stored and how much memory the computer allocates for the data stored in the variable. The data type also governs the kinds of operations that can be performed on a variable.

Python Data Types

In Python, a data type tells us what kind of value a variable store. The values, or data, contained in variables are classified into categories known as data types. Python has several built-in data types that are commonly used in programming.

Data Type	Keyword	Example
Integer	int	25
Float	float	25.5
String	str	"Hello"
Boolean	bool	True

Numeric Data Types

(Integer — int)

Stores whole numbers, without a decimal point. It could be positive or negative number an integer must have at least one digit and must not have a decimal point can be either.

```
age = 25
marks = 90
temperature = -5
population_Of_India=1480137241
population_Of_Pakistan=259299791
```

```
print("Age is ", age)
```

```
print(type(age))
```

Output:

```
25
<class 'int'>
```

(Decimal Number — float)

Stores numbers containing decimal points.

```
price = 99.99
height = 5.8
pi = 3.14159
```

```
print(price)
```

```
print(type(price))
```

Output:

```
99.99
<class 'float'>
```

String Data Type

(String — str)

A string is a sequence of characters enclosed in quotes.

```
name = "Ali"
city = 'Karachi'
```

```
print(name)
```

```
print(type(name))
```

Output:

```
Ali
```

```
<class 'str'>
```

Noted: You can use either single or double quotes:

```
name1 = "Ahmed"
name2 = 'Ahmed'
```

String operations

```
first_name = "Ali"
last_name = "Khan"
```

```
full_name = first_name + " " + last_name
```

```
print(full_name)
```

Output:

```
Ali Khan
```

You can also access individual characters:

```
word = "Python"
```

```
print(word[0])
```

```
print(word[2])
```

Output:

```
P
```

```
t
```

Noted : Python starts indexing from 0.

Boolean Data Type

(Boolean— bool)

A Boolean represents one of two values:

- True
- False

Example:

```
is_student = True
```

```
is_married = False
```

```
print(is_student)
```

```
print(type(is_student))
```

Output:

```
True
```

```
<class 'bool'>
```

Booleans are commonly used with conditions:

```
age = 20
```

```
print(age >= 18)
```

Output:

```
True
```

Output:

```
True
```

```
<class 'bool'>
```

Booleans are commonly used with conditions:

```
age = 20
```

```
print(age >= 18)
```

Output:

```
True
```

Other Data Type

Types	Keyword	Example	
List	list	[1, 2, 3]	
Tuple	tuple	(1, 2, 3)	
Set	set	{1, 2, 3}	
Dictionary	dict	{"name": "Ali"}	
Complex	complex	3 + 4j	
None	NoneType	None	

List Data Type

List/Array

A list stores multiple values in a single variable. Lists are ordered and changeable (mutable) and allow duplicate and contain different data types.

```
fruits = ["apple", "banana", "mango"]
```

```
print(fruits)
print(type(fruits))
```

Output:

```
['apple', 'banana', 'mango']
```

```
<class 'list'>
```

Lists can contain different data types:

```
student = ["Ali", 20, 85.5, True]
```

```
print(student)
```

Accessing list elements

```
print (" Student Name : ",student[0])
```

```
print(" Percentage : ",student[2] )
```

Output:

```
Student Name: Ali
```

```
Percentage : 85.5
```

Changing a list

```
fruits = ["apple", "banana", "mango"]
```

```
fruits[1] = "orange"
```

```
print(fruits)
```

Output:

```
['apple', 'orange', 'mango']
```

Tuple Data Type

tuple

A tuple is similar to a list, but it is immutable, meaning its elements cannot be changed after creation.

```
colors = ("red", "green", "blue")
```

```
print(colors)
```

```
print(type(colors))
```

Output:

```
('red', 'green', 'blue')
```

```
<class 'tuple'>
```

Accessing elements:

```
print(colors[0])
```

Output:

```
red
```

This will cause an error:

```
colors[0] = "yellow"
```

because tuples cannot be modified.

List vs Tuple

List

```
numbers = [1, 2, 3]  
numbers[0] = 10    # Allowed
```

Tuple

```
numbers = (1, 2, 3)  
numbers[0] = 10    # Not allowed
```

Set Data Type

set

- A set is an unordered collection of unique values. Duplicates are automatically removed.

```
numbers = {1, 2, 3, 4}
```

```
print(numbers)
```

```
print(type(numbers))
```

Example of duplicate removal:

```
numbers = {1, 2, 2, 3, 3, 4}
```

```
print(numbers)
```

Output:

```
{1, 2, 3, 4}
```

Sets are useful when you want to store unique values.

You can add values:

```
numbers = {1, 2, 3}
```

```
numbers.add(4)
```

```
print(numbers)
```

Output

```
{1, 2, 3, 4}
```

Dictionary Data Type

dict

A dictionary stores data as key-value pairs.

```
student = {  
    "name": "Ali",  
    "age": 20,  
    "marks": 85  
}  
  
print(student)  
print(type(student))
```

Output:

```
{'name': 'Ali', 'age': 20, 'marks': 85}  
  
<class 'dict'>
```

You can access a value using its key:

```
print(student["name"])  
print(student["marks"])
```

Output:

```
Ali  
85
```

You can change values:

```
student["marks"] = 90
```

```
print(student)
```

You can add a new key:

```
student["city"] = "Karachi"  
print(student)
```

None Data Type

NoneType None represents no value or the absence of a value. <pre>result = None</pre> <pre>print(result)</pre> <pre>print(type(result))</pre> Output: <pre>None</pre> <pre><class 'NoneType'></pre> It is commonly used when a variable doesn't currently have a value: <pre>middle_name = None</pre>	Complex— Number complex Stores numbers containing a real and imaginary part. <pre>z = 3 + 4j</pre> <pre>print(z)</pre> <pre>print(type(z))</pre> Output: <pre>(3+4j)</pre> <pre><class 'complex'></pre> Here: • 3 → real part • 4j → imaginary part You can access them: <pre>print(z.real)</pre> <pre>print(z.imag)</pre>
---	---

Concatenation

The Python concatenation operator (+) is used to combine two string or other type's values to create one string.

```
name = "Muhammad Ali";
age = 21;
newstring = name + " and " + age + " Years olds.";
print( newstring )
```

Note: Here dot (+) use for concatenation between the two String values

Strongly/Static typed Programming

Strongly or Static typed programming language is in every variable must be declared with a data type and data type cannot be change after declaration. Java and C++ are strongly or Static typed programming languages.

Loosely/Dynamic typed Programming

A loosely typed Programming is a programming language that does not require a variable to be defined. You can declare a variable, but it doesn't require you to classify the type of variable. Loose typing is also known as dynamic typing because the data type for a variable can be change after it has been declared. Python and Perl is a loosely typed programming language.

The Python language interpreter automatically determines what type of data is stored in a variable and assigns the variable's data type accordingly.

Exercise

Theory Questions:

1. What difference between variables and constant?
2. Name and describe the four common data type in Python.
3. Explain the purpose of the NULL data type.
4. What do you mean by strongly and loosely typed programming?
5. What is Concatenation?

Practical Questions:

1. Write a simple program of the following output using Constants.
Student Roll No. is 001
Student Name is ABC
Student Marks is 25000
Student Percentage are 78.53
2. Repeat the above program using Variables.
Hint: Save the given data at 4 memory locations respectively and **print** on the screen.
3. How do you declare a constant in Python with example?
4. Write make a simple program to set your G.R No. and Name in the variables and display G.R No. and name by echo and **print** functions.

Objective & MCQ's

1. Positive and negative number and 0 with no decimal places belong to which data type.
 - a) Double
 - b) Float
 - c) String
 - d) Integer
2. Which of the following is a valid variable name?
 - a) TotalSalary
 - b) Total Salary
 - c) TotalSalary;
 - d) Total-Salary
3. Which is the correct syntax for declaring a variable and assigning it a string?
 - a) Name="Muhammad"
 - b) Name=Muhammad
 - c) "Muhmmad"= Name;
 - d) Name = (Muahammad)

4. How many decimal places does an integer store
 - a) One decimal
 - b) Two decimal
 - c) Three decimal
 - d) Integer does not store decimal places.
5. Variable name could be starting which symbol.
 - a) _ (under score)
 - b) @
 - c) space
 - d) %
6. Combine the two or more string or other value by using concatenation_____ symbol
 - a) &
 - b) *
 - c) #
 - d) + (dot)
7. String constant value
 - a) Must be enclosed in double quotes
 - b) Must be enclosed in commas
 - c) Must be enclosed in round parentheses
 - d) Must be square brackets
8. A constant is case-sensitive by default. By convention, constant identifiers are always_____.
 - a) Lower case
 - b) Camel case
 - c) Upper case
 - d) Normal case
9. A loosely typed programming language _____.
 - a) Does not required data typed of a variables to be declared.
 - b) Requires data types of variables to be declared
 - c) Does not have variable
 - d) Does not have different data types.