

14 June

Chapter 7

Python Fundamentals

Notes

- 1) Character Set : Character Set is a set of valid characters that a language can recognize. Python has the following character set:
- a) Letters : A-Z, a-z
 - b) Digits : 0-9
 - c) Special symbols : + - * ** \ [] () { } = != == etc.
 - d) Whitespaces : → ← , newline, form feed.
 - e) Other characters : Python can process all ASCII and Unicode characters as part of data or literals.

2) Tokens : Token (Lexical units) are smallest individual units in program. Tokens are of 5 types.

- Keywords
- Identifiers
- Literals
- Punctuators
- Operator

a) Keywords :- It is a word having special meaning reserved by programming language. It is reserved for special purpose and must not be used for normal identifier names.

examples:- while, if, for, print, elif

b) Identifiers :- Identifiers are fundamental building blocks of a program and are used as the general terminology for given to different part of the program.
(variables, functions)

Identifiers starts with letters or a underscore.

Special Character not allowed.

It should not be a keyword.

Python is case-sensitive language.

examples of valid identifiers: file_123, rollno,

_date, Date, my_name, file123

examples of invalid identifiers: my book, 123_files, print, no@2

c) Literals :- Data items that have a fixed and a constant values:

There are 4 types of ~~strings~~ Literals:

String

Numeric

Boolean

Special literal None.

String: It's a sequence of letters surrounded by quotes (single, double, triple).

For example: "a23", "name"
print("Hello World")

Non graphic character are those character that can't be typed directly from ~~the~~ Keyboard.

For example, newline, tab etc.

They can be represented using escape sequence, which is single character and consumes one byte in ASCII representation.

An escape sequence is represented by

backslash (\) followed by one or more character.

Types of Strings

a) Single line String : Enclosed in single or double quotes. For example,
a = 'Python'
b = "Hello World"

b) Multi line String : By adding a ~~to~~ back slash at the end of the normal single or double quoted string.

a = 'Python \n
Program'

c) Triple quotation mark : By ~~addin~~ Enclosed in triple single or double quotes. For example :

a = '''Seema's Pen'''
b = """Hello World"""

d) Boolean : It's used to represent one of the two boolean values, True and False.

e) Special Literal None : It is used to indicate absence of value.

f) Operator : Operators are tokens that trigger some computation when applied to variables and other objects in an expression.

g) Punctuation : Punctuators are symbols that are used in programming languages to organize programming sentences structure and more. For example: {}, (), []

h) Comments : Comments are the additional readable information to clarify the source code.

Single line comment : It begins with # sign
Multi line comment : they are enclosed in triple quotes.

For example,

i) # This program prints Hello World

```
a = 'Hello'
b = "World"
print(a + b)
```

ii) # This program prints addition of two numbers.

```
print('Loading...')
a = int(input('Enter first number: '))
# Take second number from user
b = int(input('Enter second number: '))
# print output value
print(a + b)
```

Comments enclosed in triple quotes are called Docstrings.

Variables: variables named labels, whose values can be used and processed during program are called variables. For example,

```
number_1 = '3'
number_2 = '5'
print(number_1 + number_2)
output = 35
```

- i) A variable is defined only when you assign some value to it.
- ii) The variables in python don't have fixed locations
- iii) The location they refer to changes every time when the value changes.

Some examples for variables:

i) `x = 5`
`print(x)`

ii) `name1 = "Jyad"`
`name2 = "Ziyad"`
`print(name1 + " and " + name2 + " are "`
`"brothers")`

iii) `a, b = 10`
`print(b)`

iv) `a, b, c = 5, 9, 15`
`print("7")`
`print("b")`
`print(a)`

Dynamic Typing: A variable pointing to a value of a certain type can be made to point a value of different type. This is called dynamic typing. For example,

i) `x = 5`
`print(x)`
`x = "Anugrah"`
`print(x)`

ii) `x = 10`
`y = 0`
`y = x/2`
`x = "day"`
`y = "to"`
`print(x, y)`

`Type()` → It is used to determine the type of the object. For example,

`a = 2`
`type(a)`
`a = "cat"`
`type(a)`

output
int
str

Classify the following variables as valid or invalid

- | | |
|------------------|---------|
| i) Length | valid |
| ii) %age | invalid |
| iii) 2nd innings | invalid |
| iv) half_yearly | valid |
| v) Innings2 | valid |

Identify the type of literals

- | | |
|----------|---------|
| i) True | Boolean |
| ii) "3" | Str |
| iii) 3 | int |
| iv) 30.7 | float |

What is the size of following characters.

- | | |
|-----------------|---------|
| i) "it's" | 4 bytes |
| ii) "XY \n YZ" | 4 bytes |
| iii) "XY \n YZ" | 5 bytes |

i) input(): It is used to get input from the user. It always returns a value of string type. For example,

Classify the following variables as valid or invalid

- | | |
|------------------|---------|
| i) Length | valid |
| ii) %age | invalid |
| iii) 2nd innings | invalid |
| iv) half_yearly | valid |
| v) Innings2 | valid |

Identify the type of literals

- | | |
|----------|---------|
| i) True | Boolean |
| ii) "3" | Str |
| iii) 3 | int |
| iv) 30.7 | float |

What is the size of following characters.

- | | |
|-----------------|---------|
| i) "it's" | 4 bytes |
| ii) "XY \n YZ" | 4 bytes |
| iii) "XY \n YZ" | 5 bytes |

i) input(): It is used to get input from the user. It always returns a value of string type. For example,

i) `date = input("Write today's date: ")`
`print("Today's date is " + date)`

k) Features of print statement: There are many features of print().

i) `print("Hi" + "All")`

ii) `print("Hi", "All", "Welcome", sep = "...")`

iii) `print("Python")`

`print("Program")`

iv) `print("Python", end = "")` ←
`print("Program")`

outputs

i) Hi All

ii) Hi... All... Welcome

iii) Python

Program

iv) Python Program

l) Write a program to obtain 2 numbers and print their sum.

`a = int(input("Enter first number: "))`

`b = int(input("Enter second number: "))`

`c = a + b`

`print(c)`

output

Enter first number: 40
Enter second number: 25
65

m) Program to obtain length and breadth of the rectangle

```
length = float(input("Enter length: "))  
breath = float(input("Enter breath: "))  
area = length * breath  
print("The area is " + area)
```

n) Program to obtain 3 subject marks and get the output of it.

```
sub1-mark = float(inputint("Enter first subject mark: "))  
sub2-mark = float(input("Enter second subject mark: "))  
sub3-mark = float(input("Enter third subject mark: "))  
total-mark = sub1-mark + sub2-mark + sub3-mark  
avgmark = total-mark / 3  
print(avgmark)
```

Q) Find the errors in the following code.

i) A = 2
temp = 10
print(A; temp)

output
error at line 3, print(A; temp)
SyntaxError: invalid syntax

ii) x = 24
4 = x
y = x + 24
y = x + 25
print(x, y)

output
error at line 2
4 = x
SyntaxError: can't assign
to literal

iii) a = input("value:")
b = a / 2
print(b)

output
Traceback: error at line 2
b = a / 2
TypeError: unsupported operand
type(s) for / : 'str' and
'int'

P) Find the output of

```
first = 2  
second = 3  
third = first * second  
print(first, second, third)
```

```
first = first + second + third  
first third = second * first  
print (first, second, third)
```

Ans) output

2 3 6
11 3 33

9) Lvalues and Rvalues:

Lvalues - expressions that can come on the LHS of an assignment.

Rvalues - expressions that can come on the RHS of an assignment. ~~for~~

For example,

A = 2

B = 3

20 = C

a * 5 = π

error at line 3, can't assign to literal

The literals or the expressions that evaluate a value can't come on LHS, but variable names can come on LHS.