

What is Programming?

Just like we use Hindi or English to communicate with each other, we use a programming language like Python to communicate with the computer.

Programming is a way to instruct the computer to perform various tasks.

What is Python?

Python is a simple and easy to understand language which feels like reading simple English. This Pseudo Code nature of python makes it easy to learn and understandable by beginners.

Features of Python

Easy to understand = Less development time

Free and open source

High level language

Portable → Works on Linux / Windows / Mac

+ Fun to work with

Installation

Python can be easily installed from python.org. When you click on the download button, python can be installed right after you complete the setup by executing the file for your platform.

Just install
it like a game!



Chapter 1 : Modules, Comments & pip

Let's write our very first python program.

Create a file called `hello.py` and paste the below code in it

```
print ("Hello World") → print is a function (More later)
```

Execute this file (by file) by typing `python hello.py` and you will see Hello World printed on the screen.

Modules

A module is a file containing code written by somebody else (usually) which can be imported and used in our programs.

Pip

Pip is the package manager for python. You can use pip to install a module on your system.

→ `pip install flask` installs flask module.

Types of modules

There are two types of modules in Python

- 1> Built in modules → Pre installed in python
- 2> External modules → Need to install using pip.

Some examples of built in modules are `os`, `abc`, etc.

Some examples of external modules are `tensorflow`, `flask` etc.

Using Python as a Calculator

We can use python as a calculator by typing "python" + ↵ on the terminal

→ This opens REPL

or Read Evaluate Print Loop

Comments

Comments are used to write something which the programmer does not want to execute.

→ Can be used to mark author name, date etc.

Types of Comments

There are two types of comments in Python

1. Single line comments → Written using #
2. Multi line comments → Written using """ comment """

Chapter 1 - Practice Set

- 1 Write a program to print Twinkle twinkle little star poem in python.
- 2 Use REPL and print the table of 5 using it.
- 3 Install an external module and use it to perform an operation of your interest.
- 4 Write a python program to print the contents of a directory using os module. Search online for the function which does that.
- 5 Label the program written in Problem 4 with comments.

Chapter 2 - Variables and Datatypes

A variable is the name given to a memory location in a program. For example

a = 30 → Variables = Container to store a value.
b = "Harry" → Keywords = Reserved words in Python
c = 71.22 → Identifiers = Class/function/variable name

Data Types

Primarily there are following data types in Python

1. Integers
2. Floating point numbers
3. Strings
4. Booleans
5. None

Python is a fantastic language that automatically identifies the type of data for us.

a = 71 ⇒ Identifies a as class <int>
b = 88.44 ⇒ Identifies b as class <float>
name = "Harry" ⇒ Identifies name as class <str>

- Rules for defining a Variable name → Also applies to other Identifiers
- A variable name can contain alphabets, digits and underscores.
 - A variable name can only start with an alphabet and underscore.
 - A variable name can't start with a digit
 - No white space is allowed to be used inside a variable name.

Examples of a few variable names are :-
harry, one8, Seven, _Seven etc.

Operators in Python

Following are some common operators in Python :

- 1, Arithmetic operators $\Rightarrow +, -, *, /$ etc.
- 2, Assignment operators $\Rightarrow =, +=, -=$ etc.
- 3, Comparison operators $\Rightarrow ==, >, \geq, <, \leq, !=$ etc.
- 4, Logical operators \Rightarrow and, or, not

`type()` function and Typecasting

`type` function is used to find the data type of a given variable in Python.

`a = 31`

`type(a) \Rightarrow class <int>`

`b = "31"`

`type(b) \Rightarrow class <str>`

A number can be converted into a String and vice versa (if possible)

There are many functions to convert one data type into another

`str(31) \Rightarrow "31"` \Rightarrow Integer to String conversion

`int("32") \Rightarrow 32` \Rightarrow String to Integer conversion

`float(32) \Rightarrow 32.0` \Rightarrow Integer to float conversion

... and so on

Here "31" is a string literal and 31 a numeric literal.

input() function

This function allows the user to take input from the keyboard as a string

`a = input("Enter name")` \Rightarrow If a is "harry", the user entered harry

It is important to note that \Rightarrow If a is "34" user the output of input is always entered 34 a string (even if the number is entered)

Chapter 2 - Practice Set

- 1 Write a Python program to add two numbers
- 2 Write a Python program to find remainder when a number is divided by 2.
- 3 Check the type of the variable assigned using input() function.
- 4 Use comparison operators to find out whether a given variable *a* is greater than '*b*' or not.
Take *a* = 34 and *b* = 80
- 5 Write a python program to find average of two numbers entered by the user.
- 6 Write a python program to calculate square of a number entered by the user.

Chapter 3 - Strings

String is a data type in Python.

String is a sequence of characters enclosed in quotes.

We can primarily, write a string in these three ways

1. Single quoted strings → `a = 'Harry'`
2. Double quoted strings → `b = "Harry"`
3. Triple quoted strings → `c = """Harry"""`

String Slicing

A String in Python can be sliced for getting a part of the string.

Consider the following string:

`name = "H a r r y" ⇒ length = 5`

0 1 2 3 4
↓ ↓ ↓ ↓ ↓
(-5) (-4) (-3) (-2) (-1)

The index in a string starts from 0 to $(\text{length}-1)$ in Python. In order to slice a string, we use the following syntax:

`sl = name [ind_start : ind_end]`
first index included \Rightarrow last index is not included

`sl[0 : 3]` returns "Hai" → characters from 0 to 3

`sl[1 : 3]` returns "ai" → characters from 1 to 3

Negative Indices : Negative Indices can also be used as shown in the figure above. -1 corresponds to the $(\text{length}-1)$ index, -2 to $(\text{length}-2)$

Slicing with skip value

We can provide a skip value as a part of our slice like this :

word = "amazing"

word[1:6:2] → 'mzn'

Other advanced slicing techniques

word = "amazing"

word[:7] → word[0:7] → 'amazing'

word[0:] → word[0:7] → 'amazing'

String functions

Some of the mostly used functions to perform operations on or manipulate strings are :

- 1> len() function → This function returns the length of the string

len("Harry") → returns 5

- 2> string.endswith("rry") → This function tells whether the variable string ends with the string "rry" or not. if string is "Harry", it returns true for "rry" since Harry ends with rry

- 3> string.count("c") → Counts the total number of occurrence of any character

- 4> string.capitalize() → This function capitalizes the first character of a given string.

5. `String::find(word)` - This function finds a word and returns the index of first occurrence of that word in the string.
6. `String::replace(oldword, newword)` - This function replaces the oldword with newword in the entire string.

Escape Sequence Characters

Sequence of characters after backslash '\' → Escape Seq. characters

Escape sequence character comprises of more than one characters but represents one character when used within the strings.

Examples \n, \t, \', \" etc.
newline Tab Single quote → backslash.

Chapter 3 - Practice Set

1 Write a Python program to display a user entered name followed by Good Afternoon using input() function

2 Write a program to fill in a letter template given below with name and date.

letter = "Dear <1 NAME1>,
you are Selected!
<1 DATE1>""

3 Write a program to detect double spaces in a string

4 Replace the double spaces from Problem 3 with single spaces.

5 Write a program to format the following letter using escape sequence characters.

letter = "Dear Harry, This Python course is nice. Thanks!"

Chapter 4 - Lists and Tuples

Python Lists are containers to store a set of values of any data type

```
friends = ["Apple", "Akash", "Rohan", 7, False]
           ↓       ↓       ↴      ↴
           str()    str()   int()  bool()
Can store value of any datatype
```

List Indexing

A List can be indexed just like a String

$$L1 = [7, 9, "Harry"]$$

$$L1[0] \Rightarrow 7$$

$$L1[1] \Rightarrow 9$$

$$L1[70] \Rightarrow \text{Error}$$

$$L1[0:2] \Rightarrow [7, 9] \Rightarrow \text{List Slicing}$$

List Methods

Consider the following list :

$$L1 = [1, 8, 7, 2, 21, 15]$$

1. `L1.sort()`: updates the list to `[1, 2, 7, 8, 15, 21]`

2. `L1.reverse()`: updates the list to `[15, 21, 2, 7, 8, 1]`

3. `L1.append(8)`: adds 8 at the end of the list

4. `L1.insert(3, 8)`: This will add 8 at 3 index

5. `L1.pop(2)`: Will delete element at index 2 and return its value

6. `L1.remove(2)`: Will remove 2 from the list.

Tuples in Python

A tuple is an immutable data type in python.
↳ Cannot change

`a = ()` ⇒ Empty tuple

`a = (1,)` ⇒ Tuple with only one element needs a comma

`a = (1, 7, 2)` ⇒ Tuple with more than one element

Once defined, a tuple's elements can't be altered or manipulated.

tuple methods

Consider the following tuple.

`a = (1, 7, 2)`

1. `a.count(1)`: `a.count(1)` will return number of times 1 occurs in a.

2. `a.index(1)`: `a.index(1)` will return the index of first occurrence of 1 in a.

Chapter 4 - Practice Set

- 1 Write a program to store seven fruits in a list entered by the user.
- 2 Write a program to accept marks of 6 students and display them in a sorted manner
- 3 Check that a tuple cannot be changed in python
- 4 Write a program to sum a list with 4 numbers.
- 5 Write a program to count the number of zeros in the following tuple:

$$a = (7, 0, 8, 0, 0, 9)$$

Chapter 5 : Dictionary & Sets

Dictionary is a collection of key-value pairs

Syntax :

```
a = {  
    "key": "Value",  
    "Harry": "Code",  
    "Marks": "100",  
    "List": [1, 2, 9]  
}
```

`a["key"]` \Rightarrow Prints "Value"

`a["list"]` \Rightarrow Prints [1, 2, 9]

Properties of a Python Dictionaries

- 1. It is unordered
- 2. It is mutable
- 3. It is indexed
- 4. Cannot contain duplicate keys

Dictionary Methods

(Consider the following dictionary)

```
a = {"name": "Harry",  
     "from": "India",  
     "marks": [92, 98, 96]}
```

1. `a.items()` : Returns a list of (key, value) tuples

2. `a.keys()` : returns a list containing dictionary's keys

3. `a.update({"friend": "Sam"})` : updates the dictionary with supplied key-value pairs

4: `s.get("name")`: returns the value of the specified keys (And Value is returned e.g. "Harry" is returned here)

More methods are available on docs.python.org

Sets in Python

Set is a collection of non repetitive elements

`s = Set()`

\Rightarrow No repetition allowed!

`s.add(1)`

`s.add(2)`

\Rightarrow or `s = \{1, 2\}`

If you are a programming beginner without much knowledge of mathematical operations on sets, you can simply look at sets in python as data types containing unique values.

Properties of Sets

1. Sets are unordered \Rightarrow Element's order doesn't matter
2. Sets are unindexed \Rightarrow Can't access elements by index
3. There is no way to change items in sets.
4. Sets cannot contain duplicate values

Operations on Sets

Consider the following set :

$$s = \{1, 8, 2, 3\}$$

1. `len(s)`: Returns 4, the length of the set

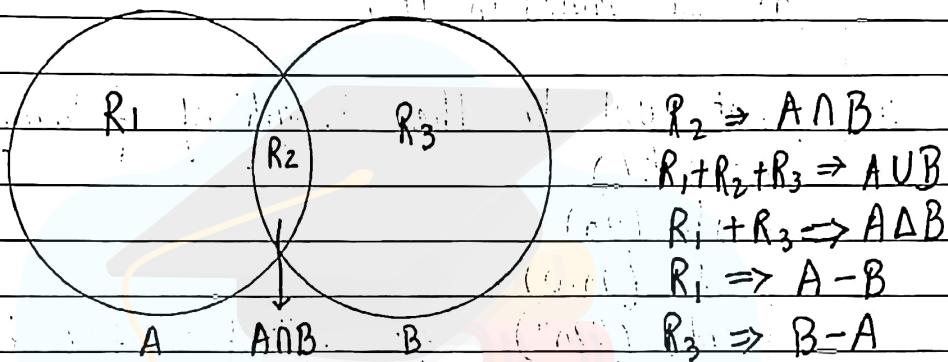
2. `s.remove(8)`: Updates the set `s` and removes 8 from `s`.

3. $S.pop()$: Removes an arbitrary element from the set and returns the element removed

4. $S.clear()$: Empties the set S

5. $S.union(\{8, 11\})$: Returns a new set with all items from both sets. $\Rightarrow \{1, 8, 2, 3, 11\}$

6. $S.intersection(\{8, 11\})$: Returns a set which contains only items in both sets. $\Rightarrow \{8\}$



Chapter 5 : Practice Set

1 Write a program to create a dictionary of Hindi words with values as their english translation. Provide user with an option to look it up!

2 Write a program to input eight numbers from the user and display all the unique numbers (once).

3 Can we have a set with 18(int) and "18"(str) as values in it?

4 What will be the length of following set S :

S = set()

S.add(20)

S.add(20.0)

S.add("20") \Rightarrow length of S after these operations?

5 S = {}

what is the type of S?

6 Create an empty dictionary. Allow 4 friends to enter their favourite language as values and use keys as their names. Assume that the names are unique

7 If names of 2 friends are same; what will happen to the program in Problem 6?

8 If languages of two friends are same; what will happen to the program in Problem 6?

9 Can you change the values inside a list which is contained in Set S

S = { 8, 7, 12, "Harry", [1, 2] }

Chapter 6 - Conditional Expressions

Sometimes we want to play pubG on our phone if the day is Sunday.

Sometimes we order Icecream online if the day is sunny.

Sometimes we go hiking if our parents allow.

All these are decisions which depends on a condition being met.

In Python programming too, we must be able to execute instructions on a condition(s) being met. This is what conditionals are for!

If else and elif in Python

If else and elif statements are a multiway decision taken by our program due to certain conditions in our code.

Syntax :

```
if (condition1):           => if condition 1 is true
    Indentation ←         print ("yes")
    ↓
    elif (condition2):     => if condition 2 is true
        print ("No")
    else:                  => otherwise.
        print ("Maybe")
```

Code example :

a = 22

```
if (a > 9):
    print ("Greater")
else:
    print ("Lesser")
```

Quick Quiz : Write a program to print yes when the age entered by the user is greater than or equal to 18.

Relational Operators

Relational operators are used to evaluate conditions inside the if statements. Some examples of relational operators are :

$= =$ → equals

$> =$ → greater than/equal to

$< =$, etc.

Logical operators

In python logical operators operate on Conditional Statements. Example :

and → true if both operands are true else false

or → true if at least one operand is true else false

not → inverts true to false & false to true

elif clause

elif in python means [else if]. An if statement can be chained together with a lot of these elif statements followed by an else statement

if (Condition1):

Code

elif (condition 2):

Code

elif (condition 3):

Code

else:

Code

⇒ This ladder will stop once a condition in an if or elif is met.



Important notes:

1. There can be any number of elif statements.
2. last else is executed only if all the conditions inside elifs fail.

Chapter 6 - Practice Set

1 Write a program to find greatest of four numbers entered by the user.

2 Write a program to find out whether a student is pass or fail, if it requires total 40% and at least 33% in each subject to pass. Assume 3 Subjects and take marks as an input from the user.

3 A spam comment is defined as a text containing following keywords:

"make a lot of money", "buy now", "subscribe this", "click this". Write a program to detect these spams.

4 Write a program to find whether a given username contains less than 10 characters or not.

5 Write a program which finds out whether a given name is present in a list or not.

6 Write a program to calculate the grade of a student from his marks from the following scheme:

90 - 100 → Ex

80 - 90 → A

70 - 80 → B

60 - 70 → C

50 - 60 → D

<50 → F