URBAN EDGE

Chapter 1 : Modules, Comments & pip

Lets 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 (.py 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 '''

URBAN EDGE

Chapter 2 - Variables and Datatypes

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

```
a = 30
b = "Harry"
c = 71.22
```

→ Variables = Containers to store a value.
→ Keywords = Reserved words in Python
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 cant start with a digit
→ No white space is allowed to be used inside a variable name.

URBAN EDGE

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 ⇒ +, -, *, / etc.
2. Assignment operators ⇒ =, +=, -= etc.
3. Comparison operators ⇒ ==, >, >=, <, != etc.
4. Logical operators ⇒ 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) => class <int>

b = "31"
type(b) => 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) => "31"      => Integer to String Conversion
int("32") => 32      => String to Integer Conversion
float(32) => 32.0    => Integer to Float Conversion

... and so on
```

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

URBAN EDGE

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

a = input("Enter name") ⇒ If a is "harry", the user entered harry

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

URBAN EDGE

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

	H	a	r	r	y
	0	1	2	3	4
	(-5)	(-4)	(-3)	(-2)	(-1)

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

sl = name [ind_start : ind_end]
first index included — last index is not included

sl [0:3] returns "Har" → characters from 0 to 3
sl [1:3] returns "ar" → characters from 1 to 3

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

URBAN EDGE

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

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

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

URBAN EDGE

5. String.find(word) - This function finds a word and returns the index of first occurence 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.
(\n → newline, \t → Tab, \' → Single quote, \\ → backslash.)