Page No.
Date : / /20

Chapter-4

Python programming fundamentals

A python program, sometimes called a script is a sequence of definitions and commands. These definitions are evaluated and the commands are executed by the python interpreter, which is known as python shell.

- Python syntax refers to a set of rules that defines how users and the system should write and interpret a python program.

Python Character Set

- Character set is a set of valid characters recognized by python.
- A character represents any letter, digit or any other

- Letters: A-Z , a-z
- Digits: 0-9
- special symbols: + - / * \ ** ( ) [ ] { } // = ! == < > . " " () ; : % ! # ? $ & ^ < = > = @ _
- white spaces: Blank space, tabs ('\t'), carriage return(↵), newline, form feed.
- Other characters: All other 256 ASCII and Unicode characters

Token

- A token is the smallest element of a python script that is meaningful to the interpreter. Each component of a programming statement is referred to as a token.
- Categories of tokens exist: Identifiers, keywords, literals,

operators and delimiters.

Identifiers:- The name of any variable, constant, function or module is called an identifiers.

Rules for defining identifiers are:-

1. First character of an identifier should start with a character or an underscore (_) but not with a digit.
2. No special characters are allowed except underscore (_)
3. Reserve words can't be used as an identifier name.
4. Space is not allowed.
5. Upper and Lower case are treated differently.

Keywords:-

- Reserved words of python, which have special fixed meaning for the interpreter, are called keywords.
- No keywords can be used as an identifier.

⇒ Literals (Constants):

A fixed numeric or non-numeric value is called a literals. It can be defined as a number, texts or other data that represents values to be stored in variables.

⇒ Operators:-

A symbol or a word that performs some kind of operation on given values and returns the result.

⇒ Delimeters:-

- Delimeters are the symbols which can be used as seperators of values or to enclose some values.
- Example of Delimiters are: ( ) [ ], ; :

★ '#' symbol used to insert comments is not a token. Any comment itself is not a token.

Concept of an object

- Python treats every value or data item, whether numeric, string or other type, as an object in the sense that it can be assigned to some variable or can be passed to a function as an argument.
- Every object in python is assigned a unique identity (ID) which remains the same for the lifetime of that object.

Variables and data types

A variable is like a container that stores values that you can access or change.

- A variable/object has three main components:

A) Identity of the variable/object
B) Type of variable/object
C) Value of the variable/object

⇒ Identity of the Variable/object

- It refers to the objects address in the memory which does not change once it has been created.
- The address of an object can be checked using the method id(object).

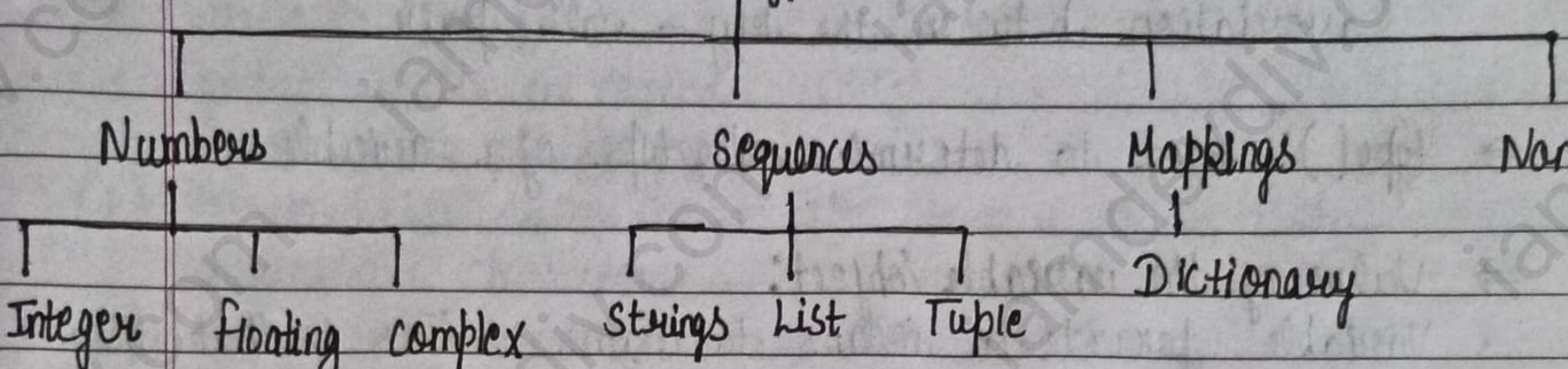
Syntax: >>> id(variable/object)

for ex:- >>> x=10
>>> id(x)

⇒ Type (Data type) of the variable/object:

- Data Type of a value refers to the value it has and the

allowable operations on those values.



⇒ Escape sequence

- Non-graphic characters are those characters which cannot be directly typed from the keyboard, such as, backspace, tabspace,
- These non-graphic characters are represented by using escape sequences.
- It is represented by a backslash (\) followed by one or more characters.

Escape Sequences in Python

Token	Category
\\	Backslash (\)
\'	Single quote (')
\"	Double quote (")
\a	ASCII Bell (BEL)
\b	ASCII Backspace (BS)
\f	ASCII formfeed (FF)
\n	ASCII Linefeed (LF)
\r	ASCII Carriage Return(CR)
\t	ASCII Horizontal tab(TAB)
\v	ASCII Vertical Tab (VT)

Page No.
Date : / /20

⇒ Usage of python Data Types

- If our data is being constantly modified or we need a fast lookup based on a custom key or we need a logical association between the key - value pair.

⇒ type( ): used to determine the type of a variable.

⇒ Value of the variable /object:

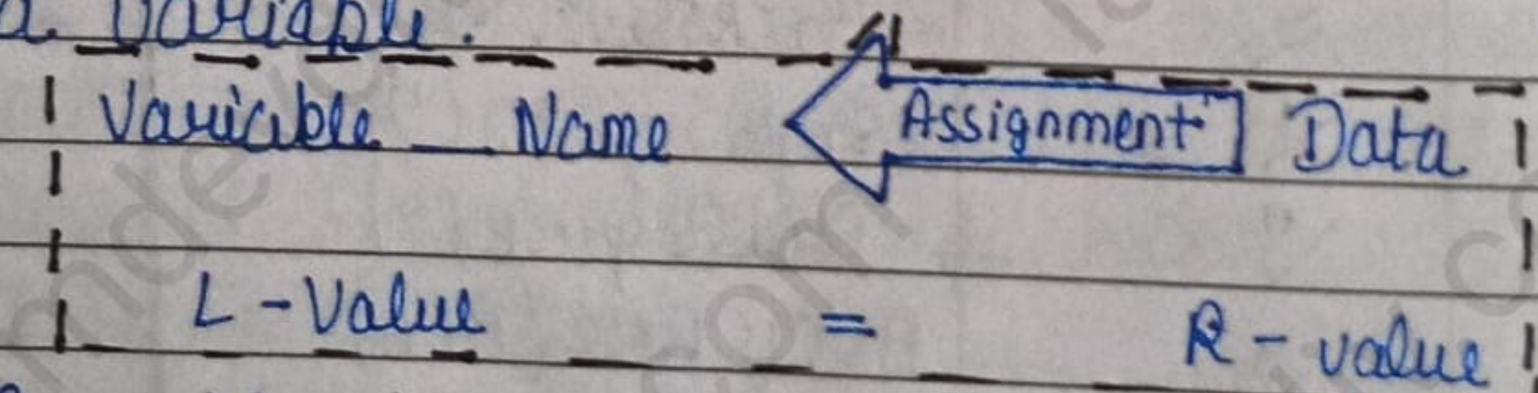
- Variable provide a means to name values so that they can be used and manipulated later.
- To bind value to a variable, we use assignment operator (=). This process is also termed as building of a variable.

⇒ Concept of L - value and R - value:-

- L-value represents an object that occupies some identifiable address in memory.
- R-value can be any valid expression.

L-value = R-value.

- A variable is a place in the computer's memory that holds a temporary value
- We can place data into a variable by Assigning it to a variable.



Concept of Assigning a value to a variable

⇒ Multiple Assignments

- It can be done in the following two ways:

→ Assigning multiple value to multiple variables:

var1, var2, var3..., var n = value 1, value 2, value 3, ... value

Evergreen
Page No.
Date : / /20

→ Assigning same value to multiple variables:

var.1 = var.2 = var.3 = , - - - - - = var.n = value

⇒ Variable Names

- Certain rules in Python which have to be followed to form valid variable names are:-

(i) A variable name must be start with an alphabet or an underscore character (_).

(ii) A variable name cannot contain spaces.

(iii) A variable name can contain any number of letters, digits and underscore (_). No other characters apart from these are allowed.

(iv) A Python Keyword can't be used as a variable name.

(v) Variables names are case-sensitive, i.e. name and NAME are ~~case-sensit~~ two different variables.

☆ eval() and int() functions are used to evaluate the value of a string.

Mutable and Immutable data types

- Variables whose values can be changed after they are created and assigned values are called mutable variables.
- Variables whose values ~~are~~ cannot be changed after they are created and assigned values ~~are~~ are called immutable variables.

☆ When an attempt is made to update the value of an immutable variable, the old variable is destroyed and a new variable is created by the same name in memory.

Evergreen
Page No.
Date : / /20

★ Mutable means the new values can be stored in the same memory location, whereas immutable types never allow to change their values in the same memory location.
Examples of mutable objects: list, dictionary, set, etc.
Examples of immutable objects: int, float, complex, bool, string, tuple etc.

* Python handles mutable and immutable objects differently
* Immutable objects are quicker to access than mutable objects

Keywords in Python

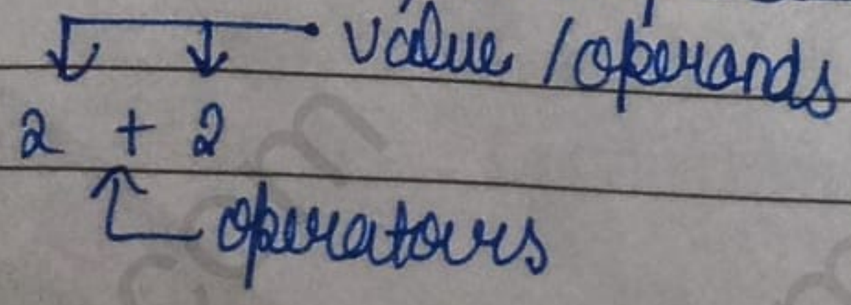
- Keywords are different from variable names, keywords are reserved words used by python interpreter to recognize the structure of a program.
- To check / display the list of keywords available in pythons

```
import Keyword
print (Keyword.kwlist)
```

* All these keywords are in small alphabets except four False, None and True, which start with capital alphabets.

Expressions

- An expression is a combination of values, i.e. constant, variable and operators.
- It generates a single value, which by itself is an expressions.
- operands contains the values an operator uses and operators are the special symbols which represent simple calculations like addition, subtraction, multiplication, etc.



Evergreen
Page No.
Date : / /20

⇒ Type Conversion

- An arithmetic expression that uses numeric data of various types is known as mixed mode expression or impure expression.
- While executing an impure expression, the process of converting the value from one data type to another is known as Type conversion.

Explicit Conversion

- Explicit conversion also called type casting, happens when data type conversion takes place deliberately i.e. the programmer forces it in the program.
- In an arithmetic expression (pure or impure) when the data types of the result get converted into a specific type based on user's request, it is known as Explicit Type conversion or Type casting.

- General form of an Explicit data type conversion is:

(Explicit new_data_type) (Expression)

OR Variable = (Explicit new_data_type) (Expression)

90 100

Evergreen
Page No.
Date : / /20

Function	Description	Output
int(x)	Converts x into an integer	>>> int(10.00) 10

float(x)	Converts x into a floating-point number. This x can be an integer or even a string value	>>> float(10)
		10.00
		>>> float("98.65")
		98.65

Str(x)	Converts x into a string representations	>>> Str("1000") '1000'
Chr(x)	Converts x into a character	>>> Chr(65) 'A'

Implicit Conversion

- Implicit conversion, also known as coercion, happens when data type conversion is done automatically by Python and is instructed by the programmer.
- The hierarchical order of the data types is int, float and complex.

Operators

- Operators are special symbols or tokens which represent computation or perform various mathematical or logical operations.
- They are applied on ~~opts~~ operands which can be values or variables.

Evergreen
Page No.
Date : / /20

⇒ Mathematical / Arithmetic operators

A number of operators are available in python to form and solve arithmetic or algebraic expressions.

Arithmetic operators in python

Symbol	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division

Symbol	Description
%	Remainder/Modulus
**	Exponentiation
//	Integer division

⇒ Precedence of Arithmetic operators

decreasing order
- () Parantheses
- ** (exponentiation)
- - (negation)
- / (division) // (integer division) * (multiplication) % (modulus)
- + (addition) - (subtraction)

⇒ Relational operators

- Relational operators are used for comparing two expressions and result in either True or False
- In an expression comprising both arithmetic and relational operators, the arithmetic operators have high precedence than relational operators.

Syntax:

<expression 1><comparison operator><expression 2>

Symbol	Description	Symbol	Description
<	less than	!= , <>	not equal to
>	greater than	==	equal to
<=	less than equal to		
>=	greater than equal to		

* Python compares strings lexicographically, using ASCII value of the characters. If the first characters of both the strings are same, the second characters are compared and so on.

⇒ Logical operators

- The three logical operators supported by python are and, or and not.

Symbol	Description
Or	If any one of the operands is true, then the condition becomes true
and	If both the operands are true, then the condition becomes true
not	Reverses the state of operand / condition

- Precedence of logical operators

not
and
or

decreasing order

* Relational operators are also known as comparision operators and yield values. True or False, as the output.

Evergreen
Page No.
Date : / /20

⇒ Shorthand / Argumented Assignment operators

- A shorthand Assignment operator (or compound assignment operator or an argumented operator) is a combination of a binary operation and assignment. It is so called because it reduces the size of an arithmetic statement.

* Same operator may perform a different function depending on the data type of the value to which it is applied.

⇒ Membership operators

- Python offers two membership operators for checking whether a particular character exists in the given string or not.

→ 'in' operator:- It returns true if a character / substring exists in the given string.

→ 'not in' operator:- It returns true if a character / substring does not exist in the given string.

Syntax:

<substring> in <string>
<substring> not in <string>

⇒ Identity operators

Identity operators are used to compare the objects if both the operands are actually the same and share the same memory location. These operators are 'is' and 'is not'.

→ 'is' operator:- It returns true if both variables are pointing to the same objects.

Ex:- var1 is var2 results to True if id(var1) = id(var2)

→ 'is not' operator!- It returns true if both variables are not the same objects.
Ex:- var1 is not var2 results to True if id(var1) is not equal to id(var2)

User Input

- The values inputted by the user are fetched and stored in the variables.
- Python provides three important functions for getting user's input.

1. input( ): input() function is used to get data from the user while working with the script mode. It enables us to accept an input in the form of string from the user without evaluating its value.
- The function input continues to read input text from the user until it encounters a new line.
- ☆ The input() function takes string as an argument.
- ☆ int() function converts the inputted string value into numeric value into numeric value and shall store the value as a ~~number~~ number and not as a string.

2. eval(): This function is used to evaluate the value of a string. It takes a string as an argument evaluates this string as a number and returns the numeric result.

☆ If the given argument is not a string, or if it cannot be evaluated as a number, then eval() result in an error

Evergreen
Page No.
Date : / /20

User-defined functions

- A function is a group of statements that exists within a program for the purpose of performing a specific task.
- Instead of writing a large program as one long sequence of instructions, it can be written as several small functions, each one performing a specific part of the task.

⇒ How to Define a call a function in python

A user defined python function is created or defined by the def statement followed by the function name and parantheses( ) as shown in the syntax:

Syntax:

function definition
```
def function_name (comma_separated_list_of_parameters):
    """docstring"""
    statements
```

Indentation in Python

★ Statements below def begin with four spaces. This is called identation. It is a requirement of python that the code following a colon must be idented.

- Indentation is shifting of a statement to the right of another statement by adding white space before a statement.
- In python, identation is used to form suites (block of statement) that indicates the group of statements belongs to a particular block of code.

* A suite is a group of statements which is treated as a

unit.
- This concept is used extensively in functions, conditional statements and loops.

Rules and & conventions for writing Python Programs.

a) Statement Termination: ~~The~~ There is no demarcation or symbol in Python to indicate the termination of statement. When you end typing a statement in python by pressing Enter key, the statement is considered terminated by default.

b) Clarity and Simplification of Expression:-
Writing ~~the~~ unambiguous expression is a skill that must be developed by every Programmer.

c) Simplicity of Instructions:- The Instructions given to a computer must be clear and simple so that any user reading those instructions should be able to understand them.

d) Maximum Line length:- Line length should be of maximum 79 characters.

e) Lines and Indentation:- Blocks of code in python are denoted by Line Indentation, which is implemented using 4 spaces per indentation level.

f.) Avoid multiple statements in one line:- Although you can type multiple statements using semicolon(;) between two statements in a single line, it is not recommended.

Evergreen
Page No.
Date: / /20

Comments

- Comments are statements in a script that are used to explain python code and are ignored by the python interpreter.

- A comment in python starts with a hash (#) symbol anywhere in a line and extends till the end of the line.
* Triple quotes for specifying a multiline comment.
for ex:- """ This is a multiline comment """

Debugging

- Most application codes written for developing applications have errors in them.
- The process of finding errors in a program is termed as Debugging.

- Errors in python may classified mainly into three types:-
(a) Syntax Error
(b) Run-time Error
(c) Logical Error

⇒ Syntax Error

- A syntax Error is an error in the syntax of a sequence of characters or tokens that is intended to be written in a particular programming Language.
- These types of errors are generated when we violate the syntax or, in other words, the gramamatical rules of a programming language.
Examples of a Syntax Error:-
1. Incorrect indentation
2. Misspelling a keyword

Evergreen
Page No.
Date : / /20

3. Leaving out a symbol such as colon (:) comma(,) or parantheses(( ))

⇒ Run-time Error

- A run-time error occurs after python interpreter interprets the code you write and the computer begins to execute it.
- Some examples of python run-time errors are:-

1. division by zero
2. performing an operation on incompatible types.
3. Using an identifier variable which has not been defined.
4. Accessing a list element, dictionary value or object attribute which doesn't exist.
5. trying to access a file which doesn't exist.

⇒ Logical Errors

- A logical Error / bug does not stop execution but the program behaves incorrectly and produces undesired/wrong output.
- Since the program interprets successfully even when logical errors are present in it, it is sometimes difficult to identify these errors.

→ Some examples of logical errors are:-

1. Using the wrong variable name for calculations.
2. Using integer division or modulus operator in place of division operator.
3. Giving operator precedence wrong.

Evergreen
Page No.
Date : / /20

Ques - 25%3**2/15+6**2

25%9/15+36

7/15+36

1+36 = 37 (Ans)

Ques - 12+3*(4-6)/2

12+3*(-2)/2

12-6/2

12-3=9 (Ans)