Chpater3 --Python Fundamentals

- **Python Character Set :-** Character set is a set of valid characters that a language can recognize.
  A character represents any letter, digit or any other symbol. Python supports Unicode encoding standard.
  Eg-

Letters	A-Z, a-z
Digits	0-9
Special symbols	Space + -* / ** \()[ ] { }//=! = == <, > . ‘ “” , ; :% ! $ # <= >= @ _ (underscore)
Whitespaces	Blank space, tabs , new line
Other characters	Python can process all ASCII and Unicode Character as part of data or literals.

- **Token –** The smallest individual unit in a program is known as a Token.

- **Kewords –** Are the words that convey a special meaning to the language compiler
  - These are reserved for special purpose and must not be used as normal identifier.
  - Eg- Fals, del, in, or, while , for etc

- **Identifiers-** Are fundamental building blocks of a program and are used as the general terminology for the names given to different parts of the program eg variables, objects , classes, functions, arrays etc.
  - ✓ Identifiers can have alphabets, digits and underscore and doller sign characters.
  - ✓ They must not be a keyword or Boolean literal or null literal
  - ✓ They must not begin with digit.
  - ✓ They can be of any length.
  - ✓ Java is case sensitive i.e upper-case letters and lower-case letters are treated differently.

  Conventions

  - ✓ The names of public methods and instance variables should brgin with a lower case letter.
    Eg maximum sum

- ✓ For names having multiple words, second and subsequent words beginning character is made capital so as to enhance readability eg avg**S**alary**O**f**E**mployees
- ✓ Private and local variables should use lower case letters eg width, results, final_score
- ✓ Class names and interface names begin with an upper case letter eg Employee
- ✓ The constants should be named using all capital letters and underscores eg MAX_VALUE, MAX_MARKS, SPECIAL_SALARY, TOTAL

- **Literals(CONSTANT)**

  - ✓ Literals (often referred to as constants) are data items that are fixed data values.
  - ✓ Python allow several kinds of literals (i) String literals (ii)Numeric literals (iii) Boolean literals
    (iv)Special Literal None (v)Literal Collections

- **Punctuators are** symbols that are used in programming languages to organize programming –sentence, structures and indicate the rhythm and emphasis of expressions, statements and program structure. eg ( ) {} [ ] . ,

- **OPERATORS** The operations (specific tasks)are represented by operators and the objects
  of the operations(s) are referred to as operands eg **arithmetic Operators, Relational Operators, Logical Operators, Assignment Operators, Conditional Operator ?**
  - o "Operators are token that trigger some computation / action when applied to variables and other objects in an expression.

Unary Operators	
+	Unary plus
-	Unary minus
~	Bitwise complement
not	Logical negation
Binary Operators	
+,-,*,/,%,**,//	Addition,Subtraction,Multiplication,Division,Remainder/Modulus,** exponent, floor division
Bitwise Operators	
& , ^, |	Bitwise AND, Bitwise exclusive OR(XOR) Bitwise OR
Shift operators	
<<	Shift left
>>	Shift right

Identity operators	
Is , is not	Is the identity same ? , is the identity not same?
Relational operators	
<, >, <=,>=,==,!=	Less than, greater than, less than or equal to, Greater than or equal to, equal to, not equal to
Logical operator	
AND, OR	Logical AND , Logical OR
Assignment operators	
=,/=,+=,*=,%=,-=,**=,//=	Assignment , Assign quotient , Assign sum, Assign product, Assign remainder, Assign difference, Assign Exponent , Assign Floor division
Membership operators	
In , not in	Whether variable in sequence , whether variable not in sequence

(i) **Expressions**---An expression is any legal combination of symbols that represents a value.

Eg **15 ,2.9 } expressions that are values only**

A+5 } **complex expressions that produce a value when evaluated**

(3+5)/4

(ii) **Statement –** A statement is a programming instruction that does something i.e some action takes place. Eg print**(“Hello”)**

(iii) **Comments-** Comments are the additional readable information to clarify the source code. Comments in Python begin with symbol # and generally end with end of the physical line.

(iv) **Functions –** A function is a code that has a name and it can be reused(executed again) by specifying its name in the program, where needed.

(v) **Blocks and Indentation**- “A group of statements which are part of another statement or a function are called block or code –block or suite in Python.”

- Variables - Named labels , whose values can be used and processed during program run, are called variables.
- Variable creation :- Python variables are created by assigning value of desired type to them, eg to create a numeric variable, assign a numeric value to variable_name ; to create a string variable, assign a string value to variable_name and so on.

  Eg Student =’Jacob’

  Age =16

  Lvalues and Rvalues---Lvalues are the objects to which we can assign a value or expression. Lvalues can come on lhs or rhs of an assignments statement. Rvalues are the literals and expressions that are assigned to lvalues. Rvalues can come on rhs of an assignment statement.

- Multiple Assignments – **a=b=c=10 , x,y,z=10,20,30**
- Variable definition-**--a variable is defined only when we assign some value to it. Using an underdefined variable in an expression /statement causes an error called NameError.**

- **Dynamic Typing –"A variable pointing to a value of a certain type , can be made to point to a value/object of different type. This is called Dynamic Typing.**

Eg

```
X=10

Print(x)

X=" Sainik"
```

Here variable X is fist pointing to /referring to an integer value 10 and then to a string value "Sainik" . Here variable X does not have a type but the value it points to does have a type. So we can make a variable point to a value of different type by reassigning a value of that type ; Python will not [illegible]

- **Simple Input and output—To get input from user interactively, we can use built –in function input().**

```
>>>  name=input("Enter your Name"

   Enter your name= Sainik
```

```
>>>  marks=input("Enter marks:")

Enter marks :89
```

Program to obtain three numbers and print their sum

```
# to input 3 numbers and print their sum

Num1=int(input("Enter number 1:"))

Num2=int(input("Enter number2:"))

Num3=int(input("Enter number3:"))
```

```
Output

Enter number1: 7

Enter number2:3

Enter number3:13
```

Program to obtain length and breadth of a rectangle and calculate its area

```
# to input length and breadth of a rectangle and calculate its area

Length=float(input("Enter length of the rectangle:"))

Breadth=Float(input("Enter breadth of the rectangle:"))

Area=length * breadth
```

```
Output

Enter length of the rectangle:8.75

Enter breadth of the rectangle:35.0

Rectangle Specifications
```