**Operation**

* ‘’ or “” = quotes (use in print command to print single line)
* “”” “”” = three quotes (use to print multiple line in a single print line)
* + = addition
* - = subtraction
* / = division (forward slash) (always the answer is in float)
* \* = multiplication (asterisk)
* () = circular brackets (parentheses)
* \*\* = exponentiation (power of the value e.g. 2^2 or 2\*\*2 = 4 like a square or cube)
* % = modulo operator (to get remainder of given division e.g. 5 % 2 = 1 where 1 is remainder)
* // = quotient (two forward slashes which is used to get quotient of given division e.g. 5//2 = 2 where 2 is the quotient)
* \ = backslash (use for escaping e.g. “He\’s good at games.” = He’s good at games)
* \n = new lines (to print new line in a single code)
  + e.g. first line\nsecond line
    - first line
    - second line
* \t = tab (this is a \ttab = this is a tab. The underscore is a tab)

**Data Types**

* String = ‘’ (quotes) or “” (quotation marks) e.g. ‘Python’ or “Python!”
* Float = decimal points (.) e.g. 34.54
* Int = integer value or whole number e.g. 345
* Concatenation = concatenation is used to add any 2 or more than 2 data either it may be string or float or int e.g. “hey my no. is “ + 53 = hey my no. is 53
* If you use multiplication in concatenation than it will print the statement as many time as the number, you have been multiplied e.g. “a” \* 3 = aaa

**Variables**

* Variables are used to get data to predefined data which is feed into the system.
  + E.g. user = Python
  + Print(user) = Python
* Input() = it is used to collect data from user input manually so that it can be used in the program to use as the data into it (it’s always take the data as a string no matter it’s a float or int)
* Input(“some text”) = in input you can use quote as a place holder so that you can tell the user that what kind of input are you taking from the user to process the data
* Int() or float() or str() = you can wrap float or int or str around input so that you can change the property of the variable according to your need
  + E.g. int(input(“Integer: “) = Integer: 3476
  + E.g. float(input(“Float: “) = Float: 4.345
  + E.g. str(input(“String: “) = String: ABC123
* Here input can be taken multiple of times as per user requirement used in the program
  + E.g.: age = input() = 43

name = input() = Python

print(name + “ is “ + age + “ years old”) = Python is 43 years old

**In-place operator**

* In-place operator is used to do mathematical operation so that it become easier to write the code
  + E.g. x = 2

print(x) = 2

x = x +3 or x += 3 # in this case both the answer will be the same

print(x) = 5

* Even you can use string also to add 2 string characters
  + E.g. x = “Python “
  + x += “is fun”
  + print(x) = Python is fun

**Booleans & Comparators**

* True or False = it is use to
* The following operators give you results in Booleans that means True or False
* < = less then
* > = greater then
* <= = less then equal to
* >= = greater then equal to
* == = equal to
* != = not equal to

**Control flow**

* if = if statements are being used for certain condition so that the actual statement can be achieved it uses logic operation
* say for e.g. 5 > 6 the answer in comparators will be in either True or False which programmer can understand but not non programmer so in this conditions we will print out that 5 is less than 6
* you can add multiple if statements into if statements
  + e.g. if conditions:

statement

if conditions:

statement

* in multiple if statement like above if any of one conditions is false then it will print the last statement and end the program there only
* else = if the conditions are false then it will print different kind of message to the user or do some other kind of operation
* you can add multiple of else statement in the program and else statement will go on continuing until it reaches to such conditions
* elif = elif is also knows as else/if statement it is use to replace multiple of else and if condition in a single if statements
* There is the best example if else is to find out the leap year where 1900 is not a leap year and 2000 is a leap year. Here is the question
* You need to make a program to take a year as input and output “Leap year” if it’s a leap year, and “Not a leap year”, if it’s not.
  + To check whether a year is a leap year or not, you need to check the following:

1. If the year is evenly divisible by 4, go to step 2. Otherwise, the year is NOT leap year
2. If the year is evenly divisible by 100, go to step 3. Otherwise the year is a leap year.
3. If the year is evenly divisible by 400, the year is a leap year. Otherwise, it is not a leap year.
4. Sample input: 2000
5. Sample output: Leap year

**Boolean Logic**

* and, or, not this are the three Boolean logical operators which is used for determining whether the condition is True or False.
* and = if both the condition is true then the output is true
* or = if any of the condition is true then the output is true
* not = if will give the opposite answer of the output e.g. True = False & False = True

**While loop & break & continue**

* while loop is use to perform a single statement multiple of time until the condition is been matched
* break is use to presently break the loop of while loop
* continue jumps back to the top line of the loop

**Lists (worst things for me to remember it)**