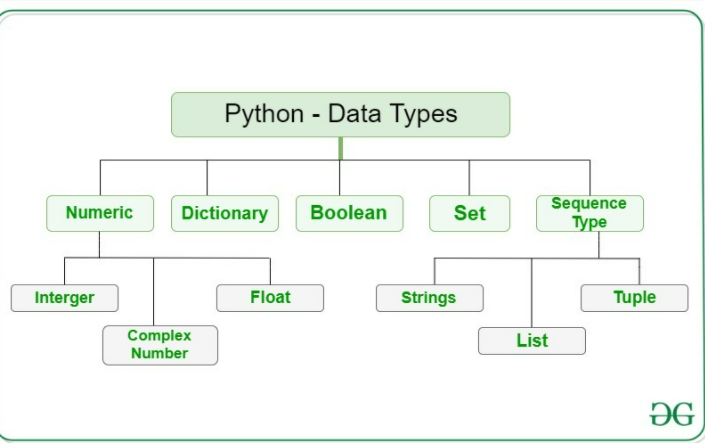
**Basic Concepts in Python**

Print() is the basic function used to print or output your answer

* Print(‘Hello World!’) 🡪Hello World!

In Python, comments begin with a #. This statement is ignored by the interpreter and serves as documentation for our code.

To print something in a new line either \n or “”” can be used

When there is () after a variable it is considered as function

We use input function to ask the user to insert or put a value

X= input ()

Print (X)

If we want the Inputs added to be in a specific data type, we can put int (input ()) or str (input ()) to make it to integer or string

**Variables: -**

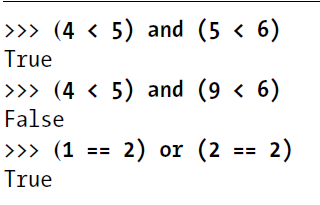
* You’ll store values in variables with an assignment statement.
* An assignment statement consists of a variable name, an equal sign (called the assignment operator), and the value to be stored.
* If you enter the assignment statement spam = 42, then a variable named spam will have the integer value 42 stored in it.
* Variables are case sensitive and they cannot start from a number

**Operators**

* **Arithmetic operators**

1. + is used for addition
2. – for subtraction
3. \* for multiplication
4. \*\* = power of e.g., print (2\*\*5) =32
5. ‘/’ is used for simple division
6. //= gives only integral part of quotient
7. % = gives only the remainder or mod of the number

* **Comparison operators: -** These are used to compare 2 values and give a single BOOLEAN value

1. = = 🡪equal to
2. < 🡪 greater than
3. >🡪 less then
4. != 🡪 not equal too
5. <= 🡪greater than or equal to
6. >= 🡪 less than or equal to

* **BOOLEAN Operators**: -

1. There are 3 BOOLEAN operators (and, or, not) these are used to compare between 2 BOOLEAN values
2. In (and) only if both the values are True the output is True and in all other cases its False
3. In (or) even if any one value is True the output is True, only when both values are False the output is False
4. Not can only be used for one value and it outputs the opposite of the given value

In python operators allows to write x=x+3 as x+=3 it can be used for strings and integers for all 4 operations

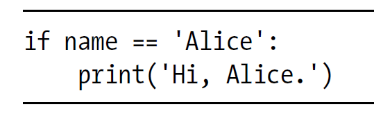
**Flow Control**

Flow control Statements are statements which lets the python to know whether to skip, repeat or stop the program and it also can decide which Python instructions to execute under which conditions.

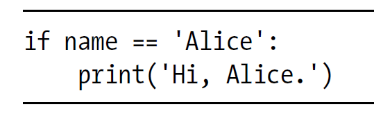
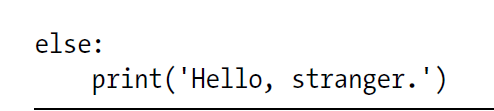
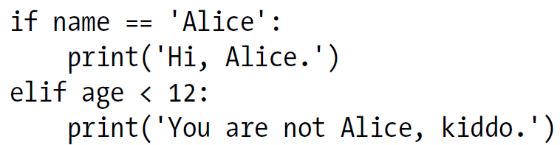
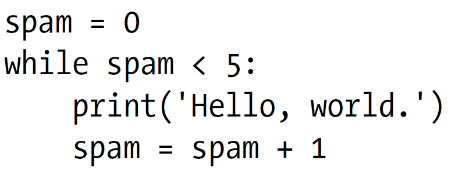
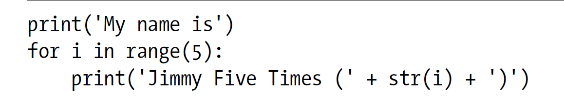
There are 3 elements in a flow control statement:

* **Condition:** The output of a given condition (mostly True, False) tells the statement whether or not to execute the statement
* **Blocks of code**: Lines of a python code can be grouped into blocks. we can tell if a block begins or closes based on its indentation. its main purpose is to make the code easy to understand
* **Program execution:** This term means that the current program is being executed. Not all programs execute from top to bottom if we put our finger and go though the code as its being executed, we may notice that sometimes the we skip an entire clause.

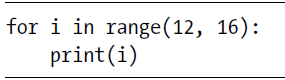
**Flow control statements:**

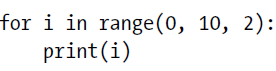
* **If statement:** This is the most common statement. An if statements clause (that is the block following the statement) executes a program if statements conditions is True. The clause is skipped if the condition is False

if statement consists of the following:

* + The if keyword.
  + A condition (that is, an expression that evaluates to True or False)
  + A colon
  + Starting on the next line, an indented block of code (called the if clause)
* **Else statement:** An if clause can optionally be followed by an else statement. The else clause is executed only when the if statement’s condition is False. In plain English, an else statement could be read as, “If this condition is true, execute this code. Or else, execute that code.” An else statement doesn’t have a condition, and in code, an else statement always consists of the following:
  + The else keyword
  + A colon
  + Starting on the next line, an indented block of code (called the else clause)
* **elif statement:** While only one of the if or else clauses will execute, you may have a case where you want one of many possible clauses to execute. The elif statement is an “else if” statement that always follows an if or another elif statement. It provides another condition that is checked only if any of the previous conditions were False. In code, an elif statement always consists of the following:
  + The elif keyword
  + A condition
  + A colon
  + Starting on the next line, an indented block of code
* **While statement:** You can make a block of code execute over and over again with a while statement. The code in a while clause will be executed as long as the while statement’s condition is True. In code, a while statement always consists of the following:
  + The while keyword
  + A condition
  + A colon
  + Starting on the next line, an indented block of code (called the while clause)
* **for Loops and the range() Function:** The while loop keeps looping while its condition is True (which is the reason for its name), but what if you want to execute a block of code only a certain number of times? You can do this with a for loop statement and the range () function. In code, a for statement looks something like ‘for i in range(5):’ and always includes the following:
  + The for keyword
  + A variable name
  + The in keyword
  + A call to the range() method with up to three integers passed to it
  + A colon
  + Starting on the next line, an indented block of code (called the for clause)

**Start stop and step in range of a for loop**

Here the for loop variable starts from 1st argument and end at the 2nd argument excluding the argument

Here the loop will start and stop at first and 2nd argument respectively, but the 3rd argument allows the variable to increase at the intervals of mentioned value. If the value is negative the loop will count backwards

If only one argument it specified it means that the variable will start from 0 and end at the given value

* **Break and continue statements:** these 2 statements can be us in both for and while loops.
* if the condition is True and Continue is used then it skips the current variable or testcase and goes to next one
* if the condition is true and break is used then it breaks the loop at that variable and comes out of the loop.

**Data Types**

1. Integer is a data type which stores values of whole numbers like 0,2,30,100
2. Floats or floating- point numbers is a datatype which is used to store the values of decimal numbers

e.g.; 0.5 ,5.6667

1. Strings or strs is a data type that stores text values. A string must always be surrounded by “” or’ ‘
2. List is a data type which
3. Sets
4. Tuples
5. Dictionary
6. Arrays

**Strings**

If you put a + operator in between 2 strings it joins them both it is called string concatenation.

If u multiply a string with a certain number, the string repeats itself for that many times. **Note: you can only multiply a string to an integer datatype**.