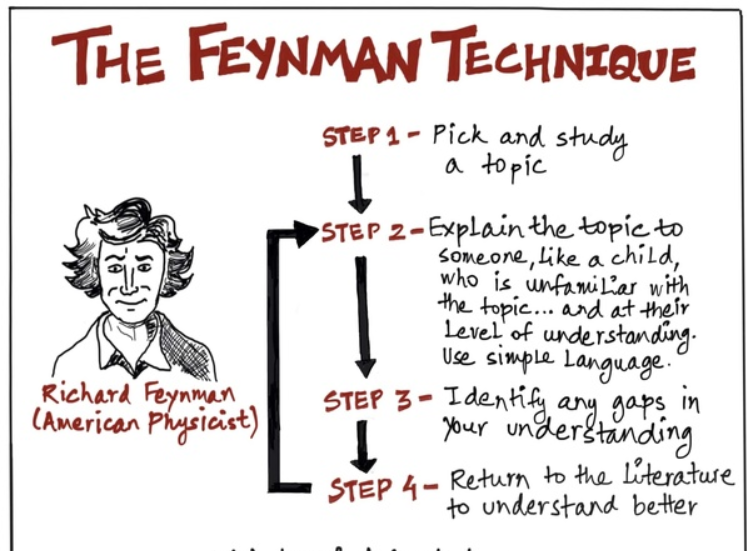
Notes



1. KeyWords

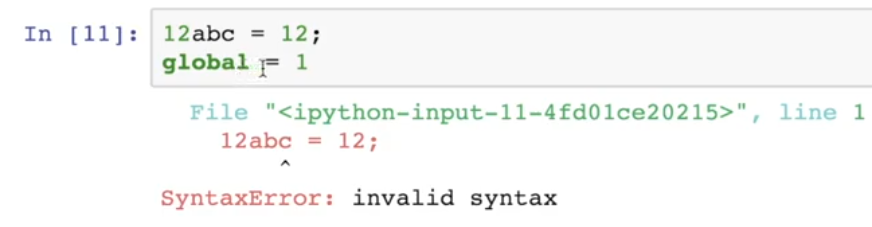
* Keywords are case sensitive

import keyword

print(keyword.kwlist)

1. Identifiers

* Identifier is the name given to entities like class, functions, variables etc. in python. It helps differentiating one entity from another.
* Keywords cannot be used as identifiers.
* Indentifiers can be a combination of letter or digit or an underscore.
* An identifier cannot not start with a digit. ***1variable*** is invalid but ***variable1*** is valid.

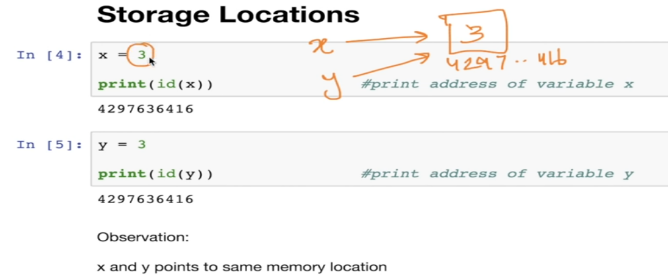


1. Variables

* A variable is a location in memory used to store some data(value).
* In python we don’t need to declare a variable before using it.
* Multiple Assignments can be done.

a,b,c = 10,5.5,"AI"

* **Storage locations**



* **Data Types**
  + *type(variable\_name)*
  + Numbers, Boolean, Strings(Index 0 based), List
  + Python list is an ordered sequence of items. It is one of the most used datatype in Python and is very Flexible.
  + All the items in a list do not need to be of the same type.

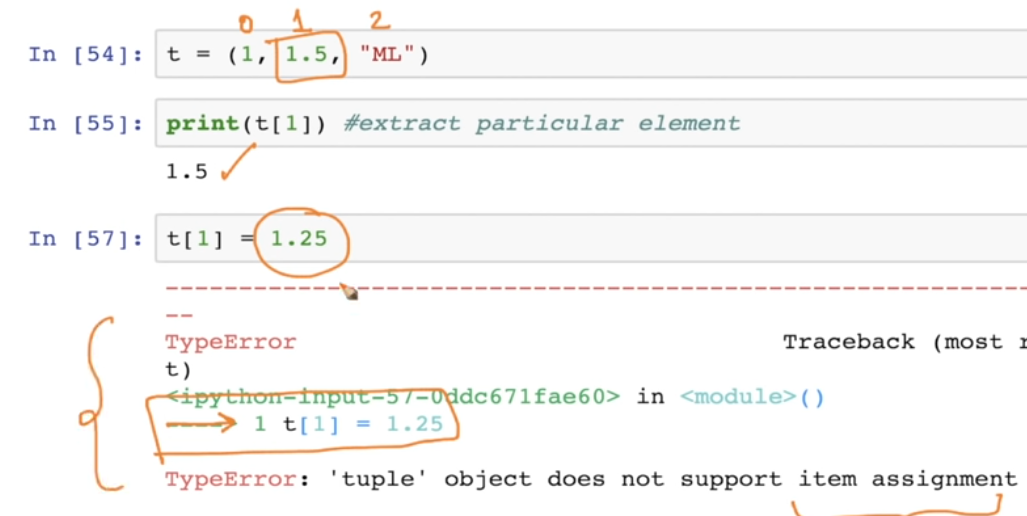
a = [19,20,20.5,”Hello”]

* + You can modify the items in list

a[1] = 30

* **Python Tuple**
  + Tuple is an ordered sequence of items same as list. ***The only difference is that tuples are immutable. Tuples once created cannot be modified.***

tuple1 = (1,1.5,”ML”)



* **Python Set**
  + Set is an unordered collection of unique items. Set is defined by values by comma inside braces { }. Items in a set are not ordered.
  + a = {10,20,30,3}
  + We can perform set operations like union ,intersection on two sets. Set have unique values.
  + Set is an unordered collection so we can not go to a particular element s[1]. ***Set object does not support indexing.***
* **Python Dictionary**
  + Dictionary is an unordered collection of key-values pairs.
  + Dictionaries are defined within braces{ } like sets. But each item being a pair in the form key:value.
  + Key and value can be of any type
  + Dictionary1 = {‘a’:”apple”,’b’:”ball”,’c’:20}



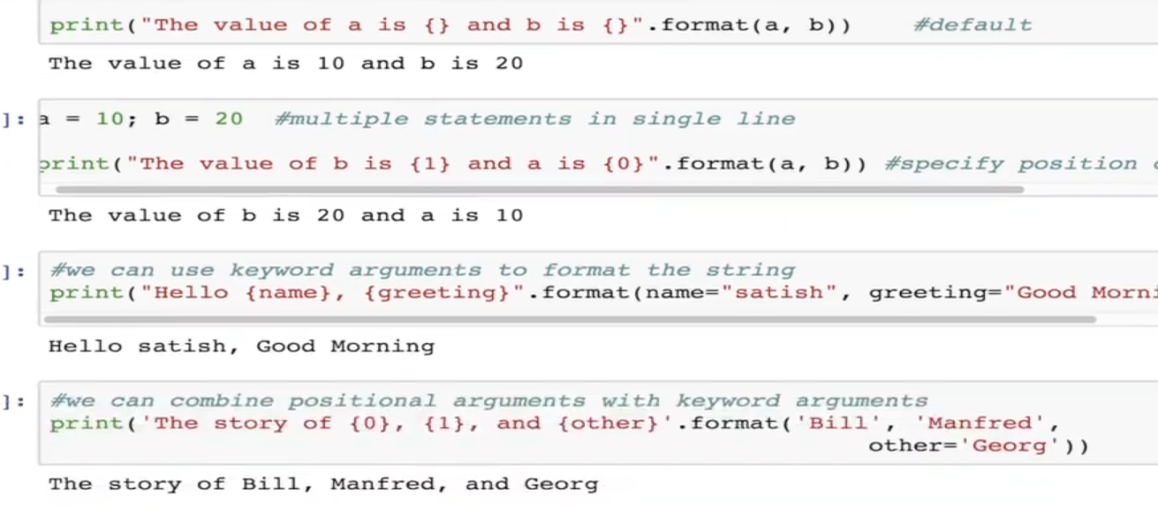
* **Conversion between DataTypes**
  + We can convert between different data types by using different type conversion functions like int(),float(),str().
  + Conversion to and from string must contain compatible values
  + Int(‘10p’) # gives error invalid literal int(‘10’) works fine.
  + We can convert one sequence to other

A = [1,2,3]

S = set(A)

List(“hello”) # Converted to [‘h’,’e’,’l’,’l’,’o’]

1. Output formatting



1. Python Input

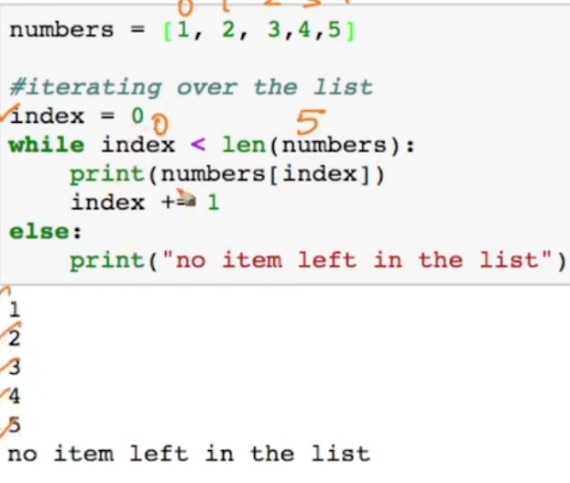
Num = input(“Enter you number”)

1. Python Operators

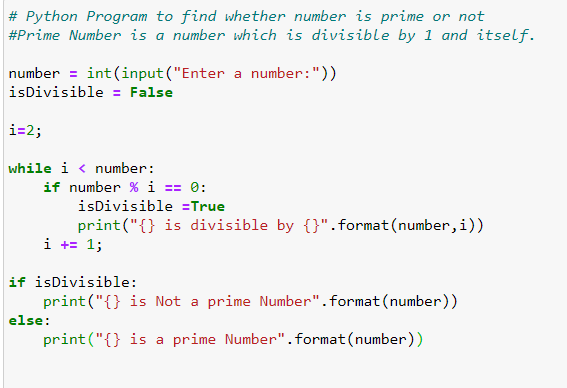
* Operators are special symbols in Python that carry out arithmetic or logical computations.
* Arithmetic, Comparison(Relational) , Logical (Boolean) , Bitwise , Assignment ,Special Operators
* Special Operators
  + Identity Operator
  + is and is not are the identity operators in python.
  + They are used to check if two values(variables) are located on the same part of the memory.
  + a,b=5
  + print(a is b) #True
  + a,b=5,6
  + print(a is b) #False
  + But for complex data type they don’t share same memory loction.
* Membership Operators
  + in and not in are the membership operators in Python.
  + They are used to test whether a values or variable is found in sequence (string,list,tuple,set and dictionary).

1. Flow Control in Python

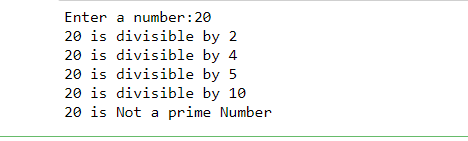
* If\else , elif, while
* While loop can have else block.



* Program to check that given number is prime or not



* Output 1



* Output 2



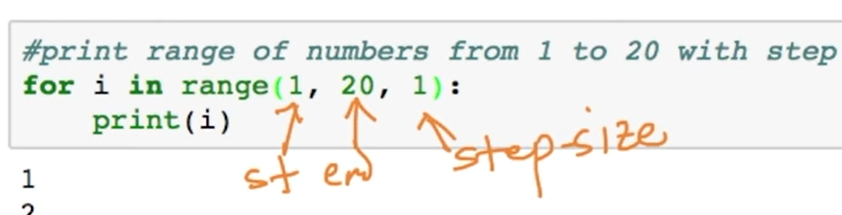
* For loop in Python

for element in sequence:

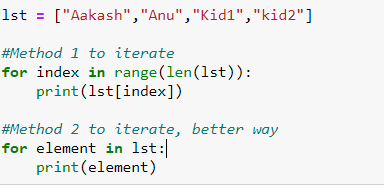
Body

* Range() function -: We can generate a sequence of numbers using range() function . range(10 will generate numbers from 0 to 9).
* Range() can be used with steps also ,

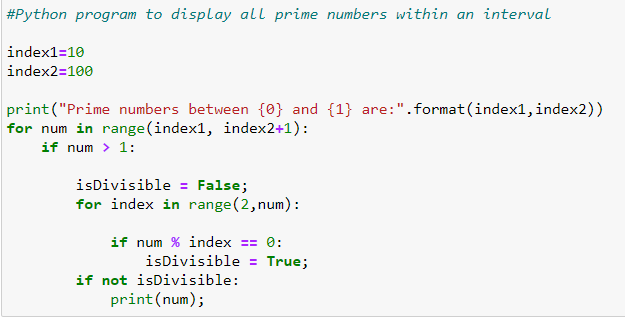
range(1,20,1) first param is start, then end(not included) and last step size



* More example



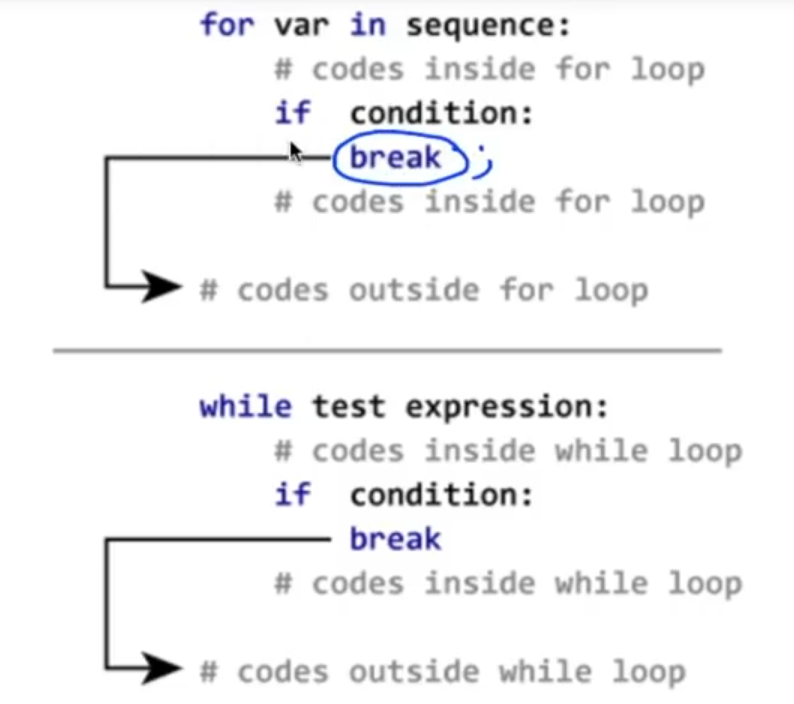
* Python Program to display all prime number within an interval



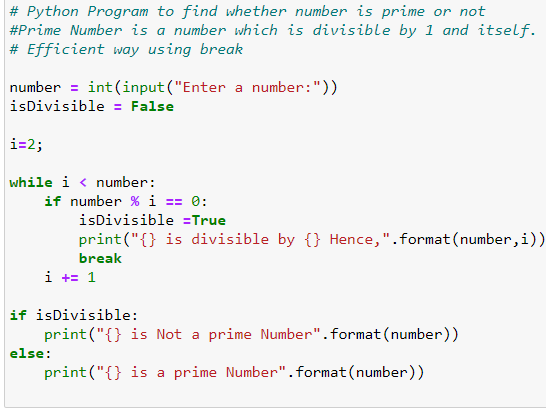
* **Python Break and continue Statements**

In python break and continue statements can alter the flow of a normal loop.

* **Python Break Statement**



* Break example



* Continue example
* Continue takes back to the loop , just does not runs anything below it. It does goes to else block at the end or if condition is not satisfied. But break does not goes to else block.