Python

Introduction to Python

Python is dynamically typed langage. Python is an object-oriented, interpreted, high-level programming language. It is general purpose, and we use it to develop GUI and web applications.

* Variables
* Data types in python
* Conditional statements
* Looping statements
* Functions

1. Variables :whenever we build any application we need to store some data in our memory.we will do that by using variables.

Simply, variables are used store and retrieve data from our systems memory.

* We can store value in a variable by either assigning a value to a variable or getting the value as input from user.
* A variable should start with a letter or an underscore and cannot start with numbers.

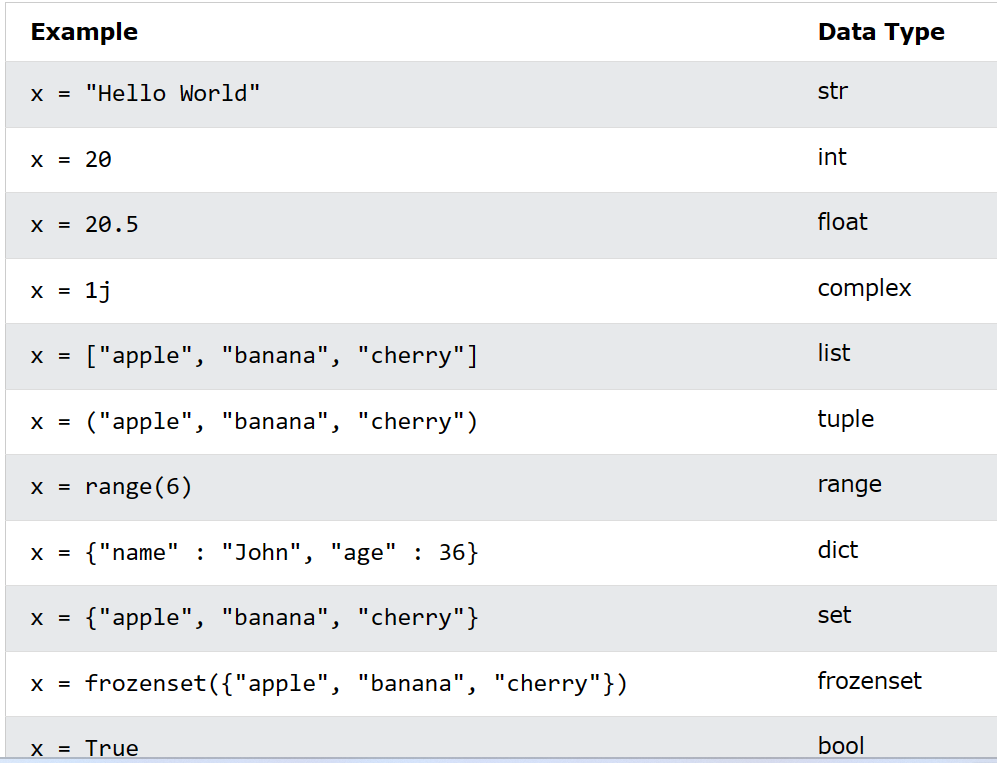
There are two ways of assigning values to a variable:

* Assigning a Single Value. Like a=10
* Assigning Multiple Values like a=b=c= 10,20,30

1. Data Types: A Data Type is simply a piece of information associated with a variable to indicate to the interpreter what type of data is stored in a variable, e.g.: Number, Text etc.

This information can then be used to determine what kind of operations are valid on a variable or multiple variables





1. Conditional Statements : Sometimes in an application we have to perform certain tasks if a given condition is true e.g. Load profile if user is logged in etc. To accomplish this in code we use conditional statements.

Conditional Statements are used to change the flow of execution when a particular condition evaluates to True or False There are three kinds of conditional statements.

1. If - (An if statement is used execute some code if certain condition evaluates to be True)

## If statements

Python supports the usual logical conditions from mathematics:

* Equals: a == b
* Not Equals: a != b
* Less than: a < b
* Less than or equal to: a <= b
* Greater than: a > b
* Greater than or equal to: a >= b

Ex:

a = 33  
b = 200  
if b > a:  
  print("b is greater than a"

1. If – elif- (The elif keyword used when we have multiple conditions and want to check them one by one)

## Elif

The elif keyword is Python's way of saying "if the previous conditions were not true, then try this condition".

Ex:

a = 33  
b = 33  
if b > a:  
  print("b is greater than a")  
elif a == b:  
  print("a and b are equal")

1. If – else - (An if else statement is used execute some code if certain conditione valuates to be True and some other code if statement evaluates to false)

Else

The else keyword catches anything which isn't caught by the preceding conditions.

Ex:

a = 200  
b = 33  
if b > a:  
  print("b is greater than a")  
elif a == b:  
  print("a and b are equal")  
else:  
  print("a is greater than b")

1. **Nested If** You can have if statements inside if statements, this is called nested if statements.

Ex: x = 41  
  
if x > 10:  
  print("Above ten,")  
  if x > 20:  
    print("and also above 20!")  
  else:  
    print("but not above 20.")

1. Looping Statements (If and While Loop statements )

When programming there are times when you have to perform certain tasks for a number of times, e.g. printing a name 100 times

You can copy and paste some code multiple time to do that or you can instruct you application to do it a number of times using loops

Looping is the process in which we have a some code that gets executed repeatedly until a particular condition is satisfied

There are two kinds of loops used in python

For – (The for statement is used to loop over a group or collection of data)

While-(The while statement simply loops until a condition is evaluates to False)

**Python For Loops**

A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).

This is less like the for keyword in other programming languages, and works more like an iterator method as found in other object-orientated programming languages.

With the for loop we can execute a set of statements, once for each item in a list, tuple, set etc.

Print each fruit in a fruit list:

fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
  print(x)

## The while Loop

With the while loop we can execute a set of statements as long as a condition is true.

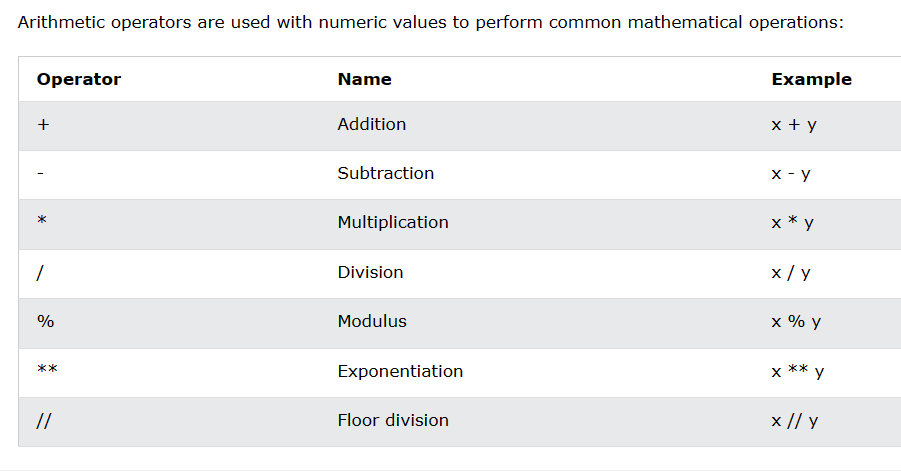
Print i as long as i is less than 6:

i = 1  
while i < 6:  
  print(i)  
  i += 1

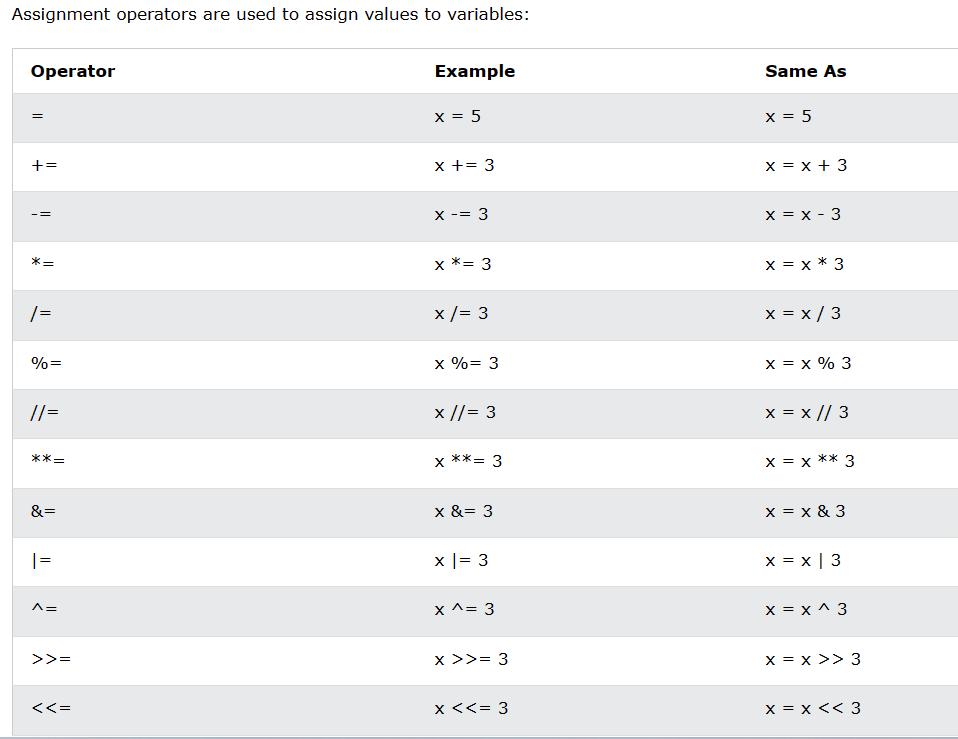
Python divides the operators in the following groups:

* Arithmetic operators
* Assignment operators
* Comparison operators
* Logical operators
* Identity operators
* Membership operators
* Bitwise operators

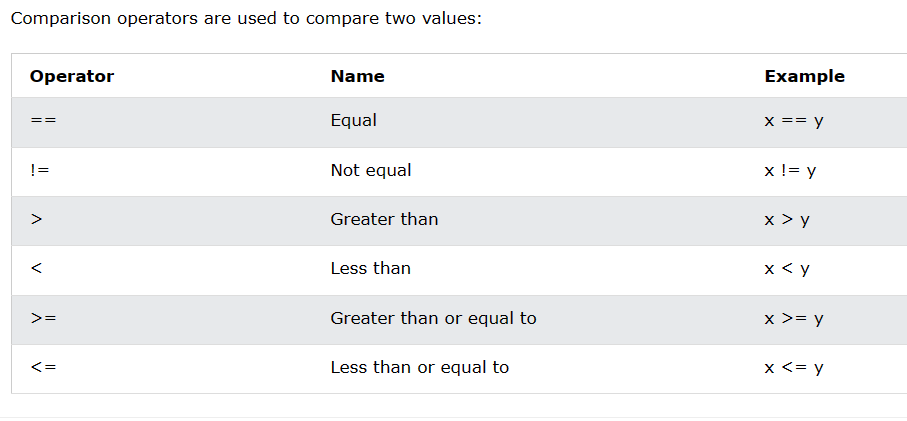
1. Arithmetic operators



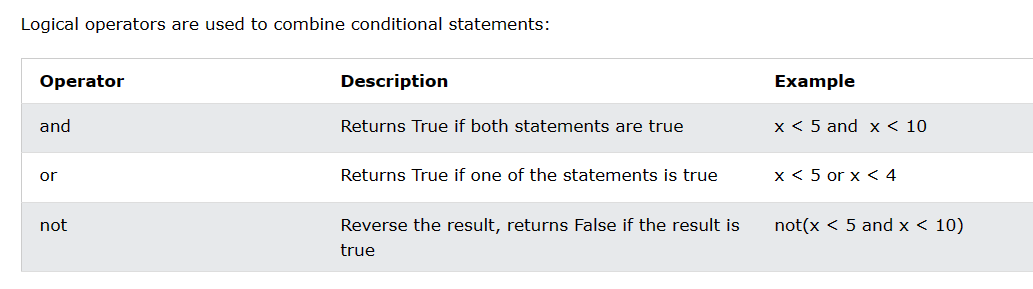
1. Assignment operators



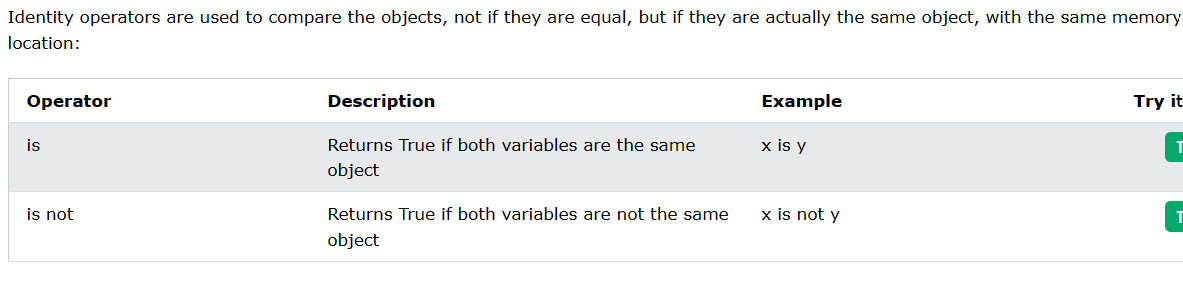
1. Comparison operators



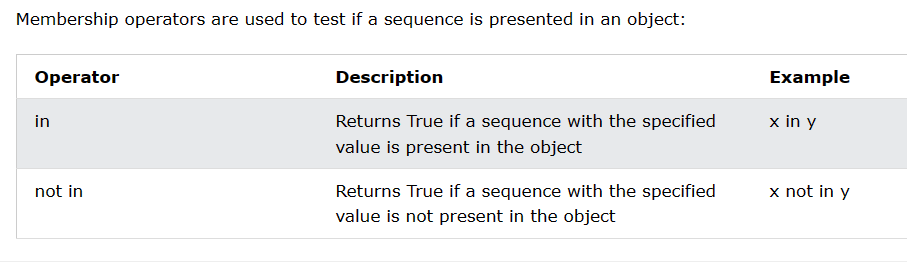
1. Logical operators



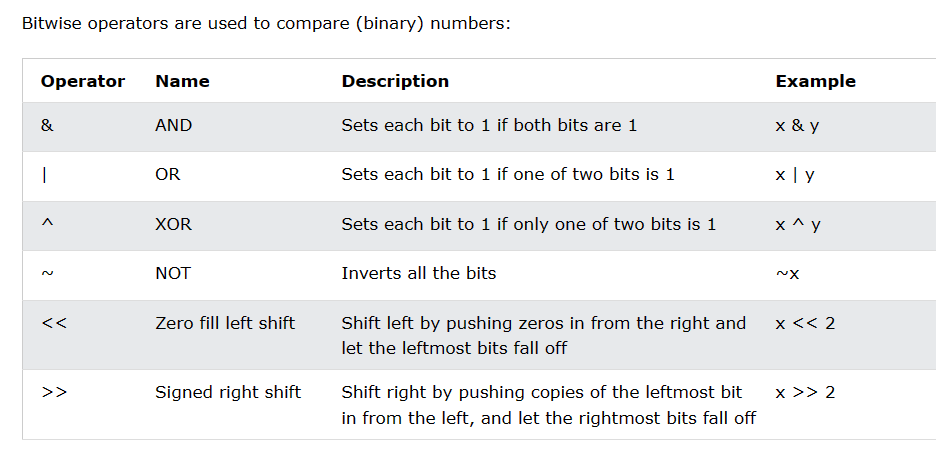
1. Identity operators



1. Membership operators



1. Bitwise operators



Python Collections (Arrays)

There are four collection data types in the Python programming language:

* [List](https://www.w3schools.com/python/python_lists.asp) is a collection which is ordered and changeable. Allows duplicate members.
* [Tuple](https://www.w3schools.com/python/python_tuples.asp) is a collection which is ordered and unchangeable. Allows duplicate members.
* [Set](https://www.w3schools.com/python/python_sets.asp) is a collection which is unordered, unchangeable\*, and unindexed. No duplicate members.
* Dictionary is a collection which is ordered\*\* and changeable. No duplicate members.