***Python:***

Python was developed by Guido van Russum in the year 1991

It is used to created

* web developments (Server side)
* Data science
* Artificial intelligence
* Automation
* System scripting

Python can be used on a server to create web applications.

Python is a high level, interpreted programming language (No Need of compilation) known for its simplicity and readability

Python has vast libraries make it a greater choice for both beginners and experience professionals.

Python can be used alongside software to create workflows.

Python can connect to database systems; it can also read and modify the files.

Python can be used to handle big data and perform complex problems and mathematics.

Python can be used for rapid prototyping or for production ready software developments

Python can work on different platforms (Windows, Mac, Linux)

Python has a simple syntax similar to English language and easy to learn

Python is a Dynamic typed programming language means you can re assign the values even after assignment.

**Python Indentations:**

Python indention refers to the space at the beginning of the code line and it is very important

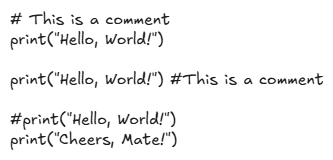
Python uses the indentation to indicate a block of code



The common use of indentions are 4 spaces but it has to be at least one

We have to use same number of spaces in the same block of code, otherwise python will throw an error.

**Python Comments:**

In Python comments can be used to make the code more readable and used to prevent executing when testing the code  
  
Examples:   
  
  
  
Python will ignore the entire line that starts with asterisk(#)  
This a to comment a single line in the code  
  
**In python we can comment multiplies also** for this we need to use a string without assigning a variable to it, so here python can ignore the string  
you can add a multiline string (ex: “””This is a Multiline string with three triple quotes”””) and we can also user three single quotes (ex: ‘’’ This is a multiline string comment with single quotes’’’)





**Python Variables:**In python a variable is created at the moment when you assign a value to it

Ex:

X = 5

Y = ‘john’

Z = 10.58

Variables do not need to be declare with a particular type, we can even change type of the variable after they have been set.

x = 4       # x is of type int  
x = "Sally" # x is now of type str  
print(x) # Sally

**Casting:**

If we want to specify the data type of a variable this can be done with casting.

For example, if you want to ***change an integer to string or integer to float***that can be done by using casting.

Examples:

x = str(3)    # x will be '3'  
y = int(3)    # y will be 3  
z = float(3)  # z will be 3.0

Here you can able to get the datatypes for the variables using **type () function**

Example:  
  
x = 5  
y = "John"  
print(type(x)) # int  
print(type(y)) # Str

**Note:** A string variable can be declared either by using single quotes or by using double quotes  
Example:  
  
x = "John"  
# is the same as  
x = 'John'

**Note:** Variable names are case sensitive.

Example:  
  
a = 4  
A = "Sally"  
#A will not overwrite a

**Python Variable Names:**

Python variable names can have a shote names like (Ex: x, y, a, b) or more descriptive words like (Ex: Age, age, Name, name, height etc...)

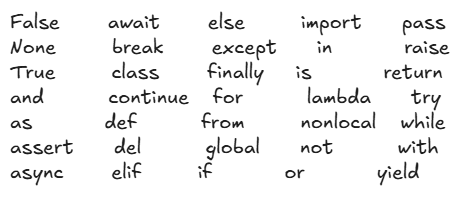
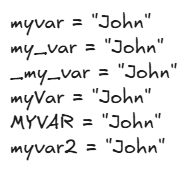
Python variable must start with a letter or with an Underscore character

**Note:** Python variable cannot start with a number

Python variable can only start with an alpha-numeric values (a-z or A-Z or with an \_)

Python variable names are case sensitive  
  
example:

Age, Age, big, Big, VALUE, value etc...

Python variable name cannot be any of the Keywords or soft reserved words  
  
  
Examples:  


Python multi word variable names with more than one word.

There are severable techniques you can make them more readable:

* Camel casing
* Pascal casing
* Snake casing

**Camel casing:** Each word, except the first word, starts with a capital letter.

Examples:

myVariableName = "John"

mySelfSiva = “Goodboy”

**Pascal casing:** Each word starts with a capital letter.

Examples:

MyVariableName = "John"

MySelfSiva = “Goodboy”

Snake casing: Each word is separated by an underscore.

Examples:

my\_variable\_name = "John"

my\_Self\_Siva = “Goodboy”

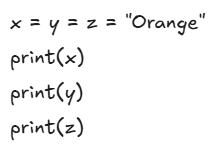
**Assigning Multiple values:**

Python allows you to assign values to multiple variables in one line

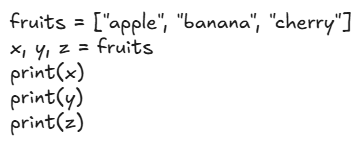
Example:  
  


In python we can assign one value to multiple variables in one line

Example:



Python we can unpack a collection: If we have collection of values in a list or in a tuple, python allows us to extract the values into variables, this is called unpacking.

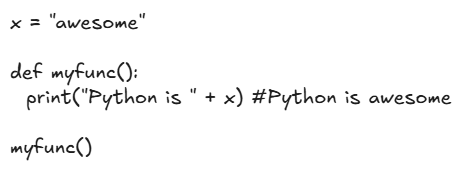


**Python Global variables:**

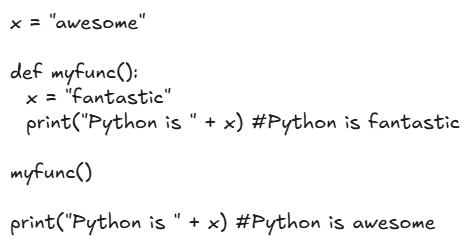
Variables that are created outside the function as known as global variables.

Global variables can be used both inside and outside of the function.

Example for Global variable:



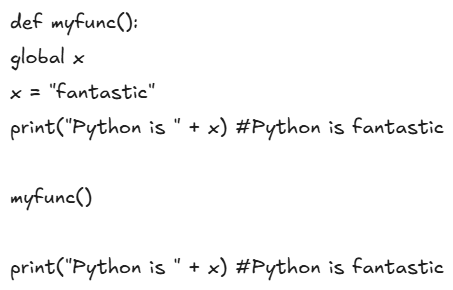
If you create a variable with the same name inside a function, this variable will be local and can only be used inside the function, the global variable with the same name will remain as it was global and with the original value.



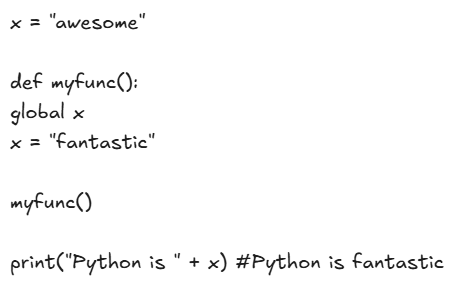
**Python Global keyword:**

When we create a variable inside a function, the variable is local and can only be used inside the function.  
  
To create a global variable inside a function, you can use the **global** keyword.

If we use the **global** keyword, the variable belongs the global scope



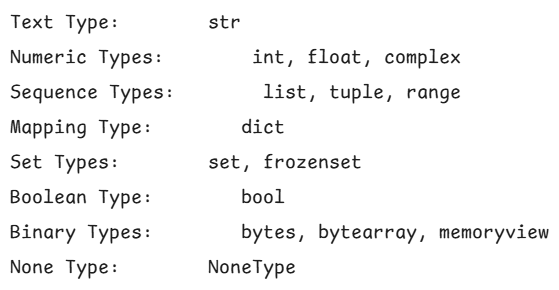
We can also user global keyword if you want to change a global variable inside a function

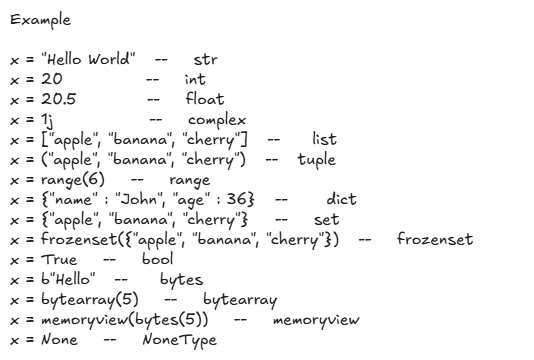


**Python Data types:**

variables can store data of different types and different types can do different things.

Python has built-in-datatypes as default:



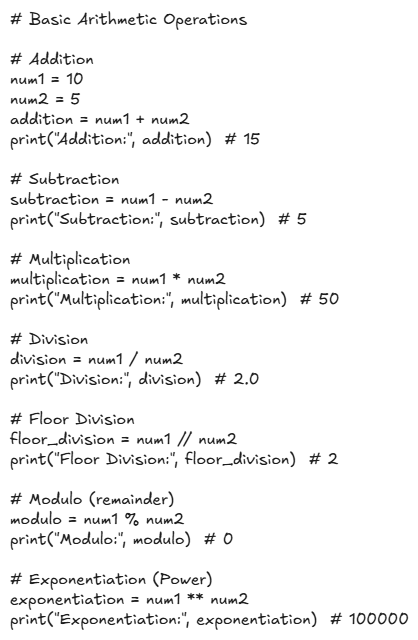


**Numbers:**

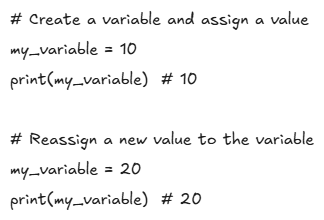
Python has various type of numbers (Numeric literals), we are mainly focus on integers and floating-point numbers.

Integers are just whole numbers, positive or negative.

**Basic Arithmetic operations:**

Example:  


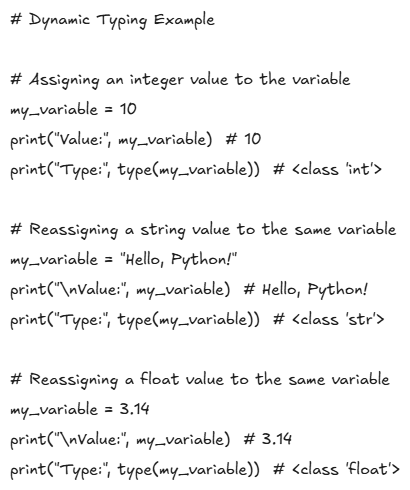
**Python Variable Assignment:**



**Dynamic Typing:**

Python uses dynamic typing; means we can re-assign the variables to different data types.

This makes python very flexible in assigning datatypes.



**Python Strings:**

Strings are used in python to record text information, such as names. Strings in python are actually a sequence, which basically means python keeps track of every element in the string as a sequence.

For Example:

Python understand the string “hello” to be a sequence of letters in a specific order.

This Means we will be able to use indexing to grab particular letters. (ex: string slicing)

Creating a String:

“This is a string”

‘This is a string’

Printing a string:

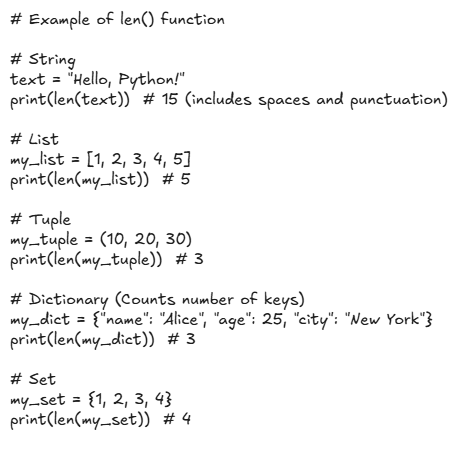
print (“This is a string”)

print (‘This is a string’)

print (‘This is a string in \n new line’) # Here \n is used to print the remining thing in a new line.

**Len () Function:**

Python build-in Len () function counts all of the characters in the string, including spaces and punctuations.



**String indexing:**

We know that strings are sequence which means python can use indexing to call of the sequence.

Here user is square brackets [] after an object to call its index.

Indexing starts with 0 for python.

Examples:

S = “Hello world”

Print (s) # Hello world

# Let’s try indexing here

print(S[0]) # ‘H’

print(S[1]) # ‘e’

print(S[2]) # ‘l’

# Now we will use colon: to perform string slicing which grabs everything up to destination point.

# syntax: - variable\_name[start : stop :step]

print(S[1:]) # ’ell0 world’

print(S) #’Hello world’

print(S[:3]) # ‘Hel’

**Note:** In the above slicing here, we will be telling python to grab everything for index 0 to index 3, here it will not include the 3rd index, where statement and are usually in the context of ‘’ up to, but not including”.

print(S[:]) #’Hello world’

print(S[-1]) # ‘d’ - # Last letter (one index behind 0 so it loops back around)

print(S[:-1]) # 'Hello Worl' - # Grab everything but the last letter

**Step size:**