

## Python Chilla with Baba Ammar

Title= "Mr"\ Name= "Ali Nawaz"\ email = "nawazktk99@gmail.com"\ whatsapp  
= "03358043653"\ Artificial Intelligence Engineer at NUST\ Education : Master  
in Software Engineering

## First Line Of Code

```
In [ ]: print("Hello World!")
        print("Wellcome to Data Science Chilla")
        print(2+3)
```

```
Hello World!
Wellcome to Data Science Chilla
5
```

## Operator

```
In [ ]: # addition
        print(3+6)

        # sub
        print(6-3)

        # mul
        print(3*6)

        # div
        print(3/6)

        # reminder
        print(3//6)

        # modulus
        print(3%6)

        # power
        print(3**6)

        # precedence PEMDAS (for remembraing)
        print(3+6+(69*8)-8/2)
```

```
9
3
18
0.5
0
3
729
557.0
```

## Strings

```
In [ ]: # we are using string using like this
# single, double and three quotation marks.
print("Hello World!")
print("Wellcome to Data Science Chilla")
print('''Wellcome to Data Science Chilla''')
```

```
Hello World!
Wellcome to Data Science Chilla
Wellcome to Data Science Chilla
```

## Comments

```
In [ ]: # name = str(input("Enter Your Name: "))
# hash is used for commenting control + /
```

## Variables

```
In [ ]: a = 5
name= 'Ali'
pi = 3.14
print(a*name)

print(type(pi))
print(type(a))
print(type(name))
```

```
AliAliAliAliAli
<class 'float'>
<class 'int'>
<class 'str'>
```

## Input Variables

```
In [ ]: name = str(input("Enter Your Name: "))
height = float(input("Put your Height in Meters: "))
weight = float(input("Put your Weight in KG: "))
age = int(input("Enter Your Age: "))

print("This is your name:", name)
```

```
This is your name: 5
```

## Condition Logics

```
In [ ]: # print(4>3)
# print(4==4)
# print(4<3)
# print(4>=3)
# print(3>=3)
```

```
age = int(input("Age of students: "))
age_ate_school = int(input("Age of students: "))
print(age== age_ate_school)
```

False

## Conversion

In [ ]:

```
x = 10

print("x is of type:",type(x))

y = 10.6
print("y is of type:",type(y))

x = x + y

print(x)
print("x is of type:",type(x))

# Python code to demonstrate Type conversion
# using int(), float()

# initializing string
s = "10010"

# printing string converting to int base 2
c = int(s,2)
print ("After converting to integer base 2 : ", end="")
print (c)

# printing string converting to float
e = float(s)
print ("After converting to float : ", end="")
print (e)
```

```
x is of type: <class 'int'>
y is of type: <class 'float'>
20.6
x is of type: <class 'float'>
After converting to integer base 2 : 18
After converting to float : 10010.0
```

## Condition

In [ ]:

```
# python program to illustrate If statement

i = 10

if (i > 15):
    print("10 is less than 15")
print("I am Not in if")

# python program to illustrate If else statement
```

```
#!/usr/bin/python

i = 20
if (i < 15):
    print("i is smaller than 15")
    print("i'm in if Block")
else:
    print("i is greater than 15")
    print("i'm in else Block")
print("i'm not in if and not in else Block")

# Python program to illustrate if-elif-else ladder
#!/usr/bin/python

i = 20
if (i == 10):
    print("i is 10")
elif (i == 15):
    print("i is 15")
elif (i == 20):
    print("i is 20")
else:
    print("i is not present")
```

I am Not in if  
i is greater than 15  
i'm in else Block  
i'm not in if and not in else Block  
i is 20

## Functions

```
In [ ]: # A simple Python function

def fun():
    print("Welcome to Chilla")

# A simple Python function

def fun():
    print("Welcome to Chilla")

# Driver code to call a function
fun()

# A simple Python function to check
# whether x is even or odd

def evenOdd(x):
    if (x % 2 == 0):
        print("even")
    else:
        print("odd")
```

```

# Driver code to call the function
evenOdd(2)
evenOdd(3)

# Python program to demonstrate Keyword Arguments
def student(firstname, lastname):
    print(firstname, lastname)

# Keyword arguments
student(firstname='Python', lastname='Practice')
student(lastname='Practice', firstname='Python')

```

```

Welcome to Chilla
even
odd
Python Practice
Python Practice

```

## Loops in Python

In [ ]:

```

# Python program to illustrate
# while loop
count = 0
while (count < 3):
    count = count + 1
    print("Hello Students")

for letter in 'geeksforgeeks':

    # break the loop as soon it sees 'e'
    # or 's'
    if letter == 'e' or letter == 's':
        break

print('Current Letter :', letter)

# Python program to illustrate
# Iterating over range 0 to n-1

n = 4
for i in range(0, n):
    print(i)

list = ["geeks", "for", "geeks"]
for index in range(len(list)):
    print(list[index])

```

```

Hello Students
Hello Students
Hello Students
Current Letter : e
0

```

```
1
2
3
geeks
for
geeks
```

## Libraries Importing

```
In [ ]: # importing module calc.py
import math
import statistics
print('value of pi is:' ,math.pi)

# importing sqrt() and factorial from the
# module math
from math import sqrt, factorial

# if we simply do "import math", then
# math.sqrt(16) and math.factorial()
# are required.
print(sqrt(16))
print(factorial(6))

# importing sys module
import sys

# importing sys.path
print(sys.path)

x = [10,20,80,6,100, 20]
print("means of x is :", statistics.mean(x))
```

value of pi is: 3.141592653589793

4.0

720

[ 'd:\\Python ka Chilla\\python\_chilla\\Python\_hands\_on', 'c:\\Users\\Ali\\.vscode\\extensions\\ms-toolsai.jupyter-2021.11.1001550889\\pythonFiles', 'c:\\Users\\Ali\\.vscode\\extensions\\ms-toolsai.jupyter-2021.11.1001550889\\pythonFiles\\lib\\python', 'C:\\Users\\Ali\\anaconda3\\envs\\python-chilla\\python36.zip', 'C:\\Users\\Ali\\anaconda3\\envs\\python-chilla\\DLLs', 'C:\\Users\\Ali\\anaconda3\\envs\\python-chilla\\lib', 'C:\\Users\\Ali\\anaconda3\\envs\\python-chilla', '', 'C:\\Users\\Ali\\anaconda3\\envs\\python-chilla\\lib\\site-packages', 'C:\\Users\\Ali\\anaconda3\\envs\\python-chilla\\lib\\site-packages\\win32', 'C:\\Users\\Ali\\anaconda3\\envs\\python-chilla\\lib\\site-packages\\win32\\lib', 'C:\\Users\\Ali\\anaconda3\\envs\\python-chilla\\lib\\site-packages\\Pythonwin', 'C:\\Users\\Ali\\anaconda3\\envs\\python-chilla\\lib\\site-packages\\IPython\\extensions', 'C:\\Users\\Ali\\.ipython']

means of x is : 39.333333333333336

## Error Trouble Shooting

```
In [ ]: # initialize the amount variable
amount = 10000

# check that You are eligible to
# purchase Dsa Self Paced or not
```

```
# if(amount > 2999):
# print("You are eligible to purchase Dsa Self Paced")

# initialize the amount variable
marks = 10000

# perform division with 0
a = marks / 0
print(a)
```

```
-----
ZeroDivisionError                                Traceback (most recent call last)
<ipython-input-14-9fb6f8d4d176> in <module>
    12
    13 # perform division with 0
----> 14 a = marks / 0
    15 print(a)
```

**ZeroDivisionError:** division by zero

## Data Structure and Indexing

In [ ]:

```
# Python3 program for demonstration
# of list index() method

list1 = [1, 2, 3, 4, 1, 1, 1, 4, 5]

# Will print the index of '4' in list1
print(list1.index(4))

list2 = ['cat', 'bat', 'mat', 'cat', 'pet']

# Will print the index of 'cat' in list2
print(list2.index('cat'))

# Python3 program for demonstration
# of index() method

list1 = [1, 2, 3, 4, 1, 1, 1, 4, 5]

# Will print index of '4' in sublist
# having index from 4 to 8.
print(list1.index(4, 4, 8))

# Will print index of '1' in sublist
# having index from 1 to 7.
print(list1.index(1, 1, 7))

list2 = ['cat', 'bat', 'mat', 'cat',
        'get', 'cat', 'sat', 'pet']

# Will print index of 'cat' in sublist
# having index from 2 to 6
print(list2.index('cat', 2, 6))
```

```
# of index() method

list1 = [6 , 8 , 5 , 6 , 1 , 2]

# Will print index of '3' in sublist
# having index from 1 to end of the list.
print(list1.index(6 , 1))
```

```
3
0
7
4
3
3
```

## Data Structures

### Tuples

```
In [ ]: # Tuples
# Creating an empty Tuple
Tuple1 = ()
print("Initial empty Tuple: ")
print(Tuple1)

# Creating a Tuple
# with the use of string
Tuple1 = ('Chilla', 'For')
print("\nTuple with the use of String: ")
print(Tuple1)

# Creating a Tuple with
# the use of list
list1 = [1, 2, 4, 5, 6]
print("\nTuple using List: ")
print(tuple(list1))

# Creating a Tuple
# with the use of built-in function
Tuple1 = tuple('Chilla')
print("\nTuple with the use of function: ")
print(Tuple1)
```

```
Initial empty Tuple:
()
```

```
Tuple with the use of String:
('Chilla', 'For')
```

```
Tuple using List:
(1, 2, 4, 5, 6)
```

```
Tuple with the use of function:
('C', 'h', 'i', 'l', 'l', 'a')
```

```
In [ ]: # Accessing Tuple
# with Indexing
```



```

Tuple1 = tuple("Chilla")
print("\nFirst element of Tuple: ")
print(Tuple1[0])

# Tuple unpacking
Tuple1 = ("Chilla", "For", "Python")

# This line unpack
# values of Tuple1
a, b, c = Tuple1
print("\nValues after unpacking: ")
print(a)
print(b)
print(c)

```

First element of Tuple:  
C

Values after unpacking:  
Chilla  
For  
Geeks

```

In [ ]: # Concatenation of tuples
Tuple1 = (0, 1, 2, 3)
Tuple2 = ('Chilla', 'in', 'Data science')

Tuple3 = Tuple1 + Tuple2

# Printing first Tuple
print("Tuple 1: ")
print(Tuple1)

# Printing Second Tuple
print("\nTuple2: ")
print(Tuple2)

# Printing Final Tuple
print("\nTuples after Concatenation: ")
print(Tuple3)

```

Tuple 1:  
(0, 1, 2, 3)

Tuple2:  
( 'Chilla', 'in', 'Data science' )

Tuples after Concatenation:  
(0, 1, 2, 3, 'Chilla', 'in', 'Data science')

```

In [ ]: # Slicing of a Tuple

# Slicing of a Tuple
# with Numbers
Tuple1 = tuple('Chila in Data Science')

# Removing First element
print("Removal of First Element: ")
print(Tuple1[1:])

```

```
# Reversing the Tuple
print("\nTuple after sequence of Element is reversed: ")
print(Tuple1[::-1])

# Printing elements of a Range
print("\nPrinting elements between Range 4-9: ")
print(Tuple1[4:9])
```

Removal of First Element:

```
('h', 'i', 'l', 'a', ' ', 'i', 'n', ' ', 'D', 'a', 't', 'a', ' ', 'S', 'c', 'i', 'e',
'n', 'c', 'e')
```

Tuple after sequence of Element is reversed:

```
('e', 'c', 'n', 'e', 'i', 'c', 'S', ' ', 'a', 't', 'a', 'D', ' ', 'n', 'i', ' ', 'a',
'l', 'i', 'h', 'C')
```

Printing elements between Range 4-9:

```
('a', ' ', 'i', 'n', ' ')
```

In [ ]:

```
# Deleting a Tuple

Tuple1 = (0, 1, 2, 3, 4)
del Tuple1

print(Tuple1)
```

**NameError**

Traceback (most recent call last)

~\AppData\Local\Temp\ipykernel\_8012\3786783553.py in <module>

```
4 del Tuple1
5
----> 6 print(Tuple1)
```

**NameError:** name 'Tuple1' is not defined

## List

In [ ]:

```
# Python program to demonstrate
# Creation of List

# Creating a List
List = []
print("Blank List: ")
print(List)

# Creating a List of numbers
List = [10, 20, 14]
print("\nList of numbers: ")
print(List)

# Creating a List of strings and accessing
```

```
# using index
List = ["Chilla", "For", "Data Science"]
print("\nList Items: ")
print(List[0])
print(List[2])

# Creating a Multi-Dimensional List
# (By Nesting a List inside a List)
List = [['Chilla', 'For'], ['Data Science']]
print("\nMulti-Dimensional List: ")
print(List)
```

Blank List:

```
[]
```

List of numbers:

```
[10, 20, 14]
```

List Items:

Chilla

Data Science

Multi-Dimensional List:

```
[['Chilla', 'For'], ['Data Science']]
```

In [ ]:

```
# Creating a List with
# the use of Numbers
# (Having duplicate values)
List = [1, 2, 4, 4, 3, 3, 3, 6, 5]
print("\nList with the use of Numbers: ")
print(List)

# Creating a List with
# mixed type of values
# (Having numbers and strings)
List = [1, 2, 'Python', 4, 'For', 6, 'Begginner']
print("\nList with the use of Mixed Values: ")
print(List)
```

List with the use of Numbers:

```
[1, 2, 4, 4, 3, 3, 3, 6, 5]
```

List with the use of Mixed Values:

```
[1, 2, 'Python', 4, 'For', 6, 'Begginner']
```

In [ ]:

```
# Creating a List
List1 = []
print(len(List1))

# Creating a List of numbers
List2 = [10, 20, 14]
print(len(List2))
```

```
0
```

```
3
```

In [ ]:

```
# Python program to demonstrate
# Addition of elements in a List
```

```

# Creating a List
List = []
print("Initial blank List: ")
print(List)

# Addition of Elements
# in the List
List.append(1)
List.append(2)
List.append(4)
print("\nList after Addition of Three elements: ")
print(List)

# Adding elements to the List
# using Iterator
for i in range(1, 4):
    List.append(i)
print("\nList after Addition of elements from 1-3: ")
print(List)

# Adding Tuples to the List
List.append((5, 6))
print("\nList after Addition of a Tuple: ")
print(List)

# Addition of List to a List
List2 = ['For', 'Python']
List.append(List2)
print("\nList after Addition of a List: ")
print(List)

```

Initial blank List:  
[]

List after Addition of Three elements:  
[1, 2, 4]

List after Addition of elements from 1-3:  
[1, 2, 4, 1, 2, 3]

List after Addition of a Tuple:  
[1, 2, 4, 1, 2, 3, (5, 6)]

List after Addition of a List:  
[1, 2, 4, 1, 2, 3, (5, 6), ['For', 'Python']]

In [ ]:

```

# Python program to demonstrate
# accessing of element from List

# Creating a List with
# the use of multiple values
List = ["Python", "For", "World"]

# accessing a element from the
# list using index number
print("Accessing a element from the list")
print(List[0])
print(List[2])

# Creating a Multi-Dimensional List

```

```
# (By Nesting a List inside a List)
List = [['Python', 'For'], ['DS']]

# accessing an element from the
# Multi-Dimensional List using
# index number
print("Accessing a element from a Multi-Dimensional list")
print(List[0][1])
print(List[1][0])
```

Accessing a element from the list

Geeks

Geeks

Accessing a element from a Multi-Dimensional list

For

Geeks

## Dictionary

```
In [ ]: # Creating a Dictionary
# with Integer Keys
Dict = {1: 'Python', 2: 'For', 3: 'All'}
print("\nDictionary with the use of Integer Keys: ")
print(Dict)

# Creating a Dictionary
# with Mixed keys
Dict = {'Name': 'Python', 1: [1, 2, 3, 4]}
print("\nDictionary with the use of Mixed Keys: ")
print(Dict)
```

Dictionary with the use of Integer Keys:

{1: 'Python', 2: 'For', 3: 'All'}

Dictionary with the use of Mixed Keys:

{'Name': 'Python', 1: [1, 2, 3, 4]}

```
In [ ]: # Creating an empty Dictionary
Dict = {}
print("Empty Dictionary: ")
print(Dict)

# Creating a Dictionary
# with dict() method
Dict = dict({1: 'Python', 2: 'For', 3: 'all'})
print("\nDictionary with the use of dict(): ")
print(Dict)

# Creating a Dictionary
# with each item as a Pair
Dict = dict([(1, 'Python'), (2, 'For')])
print("\nDictionary with each item as a pair: ")
print(Dict)
```

Empty Dictionary:

{}

Dictionary with the use of dict():

```
{1: 'Python', 2: 'For', 3: 'all'}
```

Dictionary with each item as a pair:

```
{1: 'Python', 2: 'For'}
```

In [ ]:

```
# Creating a Nested Dictionary
# as shown in the below image
Dict = {1: 'Python', 2: 'For',
        3:{'A' : 'Welcome', 'B' : 'To', 'C' : 'Python'}}

print(Dict)
```

```
{1: 'Python', 2: 'For', 3: {'A': 'Welcome', 'B': 'To', 'C': 'Python'}}
```

In [ ]:

```
# Creating an empty Dictionary
Dict = {}
print("Empty Dictionary: ")
print(Dict)

# Adding elements one at a time
Dict[0] = 'Python'
Dict[2] = 'For'
Dict[3] = 1
print("\nDictionary after adding 3 elements: ")
print(Dict)

# Adding set of values
# to a single Key
Dict['Value_set'] = 2, 3, 4
print("\nDictionary after adding 3 elements: ")
print(Dict)

# Updating existing Key's Value
Dict[2] = 'Welcome'
print("\nUpdated key value: ")
print(Dict)

# Adding Nested Key value to Dictionary
Dict[5] = {'Nested' :{'1' : 'Begginner', '2' : 'Python'}}
print("\nAdding a Nested Key: ")
print(Dict)
```

```
Empty Dictionary:
{}

```

```
Dictionary after adding 3 elements:
{0: 'Python', 2: 'For', 3: 1}

```

```
Dictionary after adding 3 elements:
{0: 'Python', 2: 'For', 3: 1, 'Value_set': (2, 3, 4)}
```

```
Updated key value:
{0: 'Python', 2: 'Welcome', 3: 1, 'Value_set': (2, 3, 4)}
```

```
Adding a Nested Key:
{0: 'Python', 2: 'Welcome', 3: 1, 'Value_set': (2, 3, 4), 5: {'Nested': {'1': 'Begginner', '2': 'Python'}}
```

In [ ]:

```
# Python program to demonstrate
# accessing a element from a Dictionary
```

```
# Creating a Dictionary
Dict = {"Course": 'Python', 'name': 'For', 3: 'Students'}

# accessing a element using key
print("Accessing a element using key:")
print(Dict['name'])

# accessing a element using key
print("Accessing a element using key:")
print(Dict['Course'])
```

Accessing a element using key:  
For  
Accessing a element using key:  
Python

In [ ]:

```
# Creating a Dictionary
Dict = {1: 'Python', 'name': 'For', 3: 'Name'}

# Deleting entire Dictionary
Dict.clear()
print("\nDeleting Entire Dictionary: ")
print(Dict)
```

Deleting Entire Dictionary:  
{}

## Sets

In [ ]:

```
# Python program to demonstrate
# Creation of Set in Python

# Creating a Set
set1 = set()
print("Initial blank Set: ")
print(set1)

# Creating a Set with
# the use of a String
set1 = set("codetolive")
print("\nSet with the use of String: ")
print(set1)

# Creating a Set with
# the use of Constructor
# (Using object to Store String)
String = 'pythontobreath'
set1 = set(String)
print("\nSet with the use of an Object: " )
print(set1)

# Creating a Set with
# the use of a List
set1 = set(["python", "For", "students"])
```

```
print("\nSet with the use of List: ")
print(set1)
```

Initial blank Set:  
set()

Set with the use of String:  
{ 't', 'd', 'o', 'c', 'e', 'v', 'l', 'i' }

Set with the use of an Object:  
{ 't', 'o', 'e', 'h', 'a', 'r', 'y', 'n', 'p', 'b' }

Set with the use of List:  
{ 'python', 'For', 'students' }

In [ ]:

```
# Creating a Set with
# a List of Numbers
# (Having duplicate values)
set1 = set([1, 2, 4, 4, 3, 3, 3, 6, 5])
print("\nSet with the use of Numbers: ")
print(set1)

# Creating a Set with
# a mixed type of values
# (Having numbers and strings)
set1 = set([1, 2, 'python', 4, 'For', 6, 'Students'])
print("\nSet with the use of Mixed Values")
print(set1)
```

Set with the use of Numbers:  
{1, 2, 3, 4, 5, 6}

Set with the use of Mixed Values  
{ 'python', 1, 2, 4, 'For', 6, 'Students' }

In [ ]:

```
# Python program to demonstrate
# Addition of elements in a Set

# Creating a Set
set1 = set()
print("Initial blank Set: ")
print(set1)

# Adding element and tuple to the Set
set1.add(8)
set1.add(9)
set1.add((6,7))
print("\nSet after Addition of Three elements: ")
print(set1)

# Adding elements to the Set
# using Iterator
for i in range(1, 6):
    set1.add(i)
print("\nSet after Addition of elements from 1-5: ")
print(set1)
```

Initial blank Set:  
set()



Set after Addition of Three elements:  
{8, 9, (6, 7)}

Set after Addition of elements from 1-5:  
{1, 2, 3, 4, 5, 8, 9, (6, 7)}

```
In [ ]: # Addition of elements in a Set

# Addition of elements to the Set
# using Update function
set1 = set([ 4, 5, (6, 7)])
set1.update([10, 11])
print("\nSet after Addition of elements using Update: ")
print(set1)
```

Set after Addition of elements using Update:  
{4, 5, 10, 11, (6, 7)}

```
In [ ]: # Accessing of elements in a set

# Creating a set
set1 = set(["Python", "For", "Students"])
print("\nInitial set")
print(set1)

# Accessing element using
# for loop
print("\nElements of set: ")
for i in set1:
    print(i, end=" ")

# Checking the element
# using in keyword
print("Python" in set1)
```

Initial set  
{'Students', 'Python', 'For'}

Elements of set:  
Students Python For True

```
In [ ]: # Deletion of elements in a Set

# Creating a Set
set1 = set([1, 2, 3, 4, 5, 6,
            7, 8, 9, 10, 11, 12])
print("Initial Set: ")
print(set1)

# Removing element from the
# Set using the pop() method
set1.pop()
print("\nSet after popping an element: ")
print(set1)
```

Initial Set:  
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}

Set after popping an element:  
{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}

```
In [ ]: # Deletion of elements in a Set

# Creating a Set
set1 = set([1, 2, 3, 4, 5, 6,
            7, 8, 9, 10, 11, 12])
print("Initial Set: ")
print(set1)

# Removing elements from Set
# using Remove() method
set1.remove(5)
set1.remove(6)
print("\nSet after Removal of two elements: ")
print(set1)

# Removing elements from Set
# using Discard() method
set1.discard(8)
set1.discard(9)
print("\nSet after Discarding two elements: ")
print(set1)

# Removing elements from Set
# using iterator method
for i in range(1, 5):
    set1.remove(i)
print("\nSet after Removing a range of elements: ")
print(set1)
```

Initial Set:  
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}

Set after Removal of two elements:  
{1, 2, 3, 4, 7, 8, 9, 10, 11, 12}

Set after Discarding two elements:  
{1, 2, 3, 4, 7, 10, 11, 12}

Set after Removing a range of elements:  
{7, 10, 11, 12}

```
In [ ]: #Creating a set
set1 = set([1,2,3,4,5])
print("\n Initial set: ")
print(set1)

# Removing all the elements from
# Set using clear() method
set1.clear()
print("\nSet after clearing all the elements: ")
print(set1)
```

Initial set:  
{1, 2, 3, 4, 5}

Set after clearing all the elements:  
set()

In [ ]: