```python
# Author: Arnab Nandi
# Date: 10.12.2022
# Topic: Pyhton Concept Revision

# To print something in the screen
print("Hello World")
# Now come to the point how we can easily use comments in our code
# To use comment we need to add '#' before the line
# Comment line simply means that the code execution process
# will just ignore the line
# Developers normally use the comment line to help other developer
# to understand his/her code or codeblocks in simple words.

# Multiline comments
"""
It
is a
multiline comment
"""
'''
It
is also
a multiline comment
'''
# Variables:-
# Variables are just like containers which can hold the value of
# some datatypes like integer, float, string, boolean etc.

# Examples:
'''
Note: '=' is an assignment operator used to store some value
in a variable.
'''
a = 1
b = 2.0
c = True  # Remember in python true is capital 'T'
d = "Arnab"
e = ["Arnab", 1, True, ["CSE", "AOT"]]
f = {"Arnab": 81, "Rohan": 82}
g = ("Arnab", "CSE")
# Verification of datatypes of the variables
# We use type function to check the type of the variable
print(type(a))  # <class 'int'>
print(type(b))  # <class 'float'>
print(type(c))  # <class 'bool'>
print(type(d))  # <class 'str'>
print(type(e))  # <class 'list'>
print(type(f))  # <class 'dict'>
print(type(g))  # <class 'tuple'>

# Now come to the point how we can take input from user
# to take input from the user we just need to use input() method
# input()  # It will wait for some input
# Now what if we want to prompt something during an input and store
# that value into a variable.

name = input("Enter your name: ")
print("You entered", name)

# Common Mistakes:
# --------------------

# Case 1: (mis-understanding)
a = "1"
b = "2"
print(a+b)
# Expected: 3
# Reality: 12
"""
Question: Now come to the point why?
Ans: Python just concatenate the two numbers as a string. It will
not recognize them as integer values.
"""
# Solution No. 1
a = 1
```

```
b = 2
print(a+b)  # It will perfectly work but not an efficient way

# Solution No. 2
'''
For this solution we need to understand what is type-casting?
Ans: Type-Casting is simply a process by which we can change the
datatype of the variable.
It can be of 2 types
1. Implicit Type-Casting (Done by interpreter itself)
2. Explicit Type-Casting (Done by programmer forcibly)
'''
# Implicit typecasting
a = 1
b = 2.2
print(a+b)  # Here a will autometically typecasted into float
# It is done by the interpreter from it's own

# Explicit typecasting (Main answer of Solution 2)
a = "1"
b = "2"
# We entered this as string and we want this act as an integer. How?
# We explicitly typecast the value of a & b into integer
print("Type-Casted answer: ", int(a)+int(b))

# Case 2:

# We generally thought that during input if we input numeric values
# then it will be as integer type but this concept is wrong
# Python by default take user input as a string

# Proof
a = input("Enter a number: ")
print(type(a))  # Output <class 'str'>
# so if we want to do some mathematical operation directly then it
# will not work. So for that we have to typecast what we are entering
a = int(input("Enter a number: "))
print(type(a))  # Output <class 'int'>

# Now come to the point String
a = "Arnab"  # Single line string
print(a)
b = "Arnab\"s"  # Use of escape sequence
print(b)
c = """In Python, a string is a sequence of characters enclosed
in quotes (either single or double quotes). Strings are used to
represent text-based data in a program, and they are one of the
most commonly used data types in Python."""
print(c)
# Basic methods of string in python
a = "Arnab"
# String is like array of characters. Indexing starts from 0
print(a[0])  # A
print(a[2])  # n
# To find out the length of a string
print(len(a))  # 5
# check presence of word/character in a string
print('r' in a)  # True
a = "I am a cool boy"
print("cool" in a)  # True
print("awesome" in a)  # False

# Same is for not in statement
print("awesome" not in a)  # True

# String slicing
a = "Arnab"
print(a[:4])  # By default start is from 0 and end is len(string)
print(a[1:4])  # First index is inclusive & second one is exclusive
# Negative slicing
print(a[-4:-1])
# Interpreter reads this statement like this
# print(a[len(a)-4:len(a)-1])
print(a.upper())
# Same concept goes for a.lower()
```

```python
a = "           Arnab"  # Having lots of white space begining
# To remove this we use strip()
# print(a) # Having whitespace in output
print(a.strip())  # Well-Formatted output
# Replace method
a = "Arnab"
print(a.replace('r', 'k'))  # Replaced r with k
# Split
a = "Hello,How,Are,You,Arnab?"
print(a.split(","))  # Returns the splitted list

# Operators
print(4+2)
print(4-2)
print(4*2)
print(4/2)
print(4.4/2)
print(4.4//2)
print(5 % 2)
print(5**2)
print(True and True)
print(False and True)
print(False or True)
print(False or False)
print(5 >= 2)
print(5 <= 2)
print(5 == 5)
print(5 != 5)

# Details on list
fruits = ["Apple", "Guava", "Watermelon", "Orange"]
print(fruits[1])  # Guava
# To add something to list
fruits.append("Grapes")
print(fruits)
# To insert something at specific position
fruits.insert(1, "Inserted")
print(fruits)
# remove list item
fruits.remove("Guava")
print(fruits)
fruits.clear()  # Return a shallow copy of the list.
print(fruits)
numbers = [10, 8, 15, 100, 16, 1]
numbers.sort()  # To sort the whole list
print(numbers)
# Loops in python
# List iteration example with the help of for loop
# Case 1:
fruits = ["Apple", "Guava", "Watermelon", "Orange"]
for i in range(len(fruits)):
    print(fruits[i])
print("\nAnother approach\n")
# Case 2:
for i in fruits:
    print(i)

# Now come to the point -> while loop
a = 5
while (a > 0):
    print(a)
    a -= 1

# Tuples
"""
In Python, a tuple is a sequence of comma-separated values enclosed
in parentheses. Tuples are similar to lists, but they are immutable,
which means that the values in a tuple cannot be changed once they
are created.
"""
# A tuple with three elements
t = (1, 2, 3)
# A tuple with a string, an integer, and a float
t = ("hello", 42, 3.14)
# An empty tuple t = ()
```

```python
# Access the first element in the tuple
first_elem = t[0]
# Access the last element in the tuple
last_elem = t[len(t) - 1]

t1 = (1, 2, 3)
t2 = (4, 5, 6)
# Concatenate two tuples to create a new tuple
t3 = t1 + t2
# Combine multiple tuples into a single tuple
t4 = t1, t2, t3
print(t4)

# Create a tuple from a list
example = [1, 2, 3, 4, 5]
print(type(example))
tuple_example = tuple(example)
print(type(tuple_example))

# Do the opposite process for converting into list from tuple
# Important Note:
# To update a tuple we need to convert it into list then update
# the list then again make it tuple

# Set in python
set_example = {1, 2, 3, 1, 1}  # Duplicates are not allowed
print("Set: ", set_example)
set_example.add(6)
print("After addition: ", set_example)

set1 = {10.15, 41, 51}
list1 = ["Arnab", 51, 103.2, True]
# We can combine sets or lists by this
set_example.update(set1)
print(set_example)
set_example.update(list1)
print(set_example)

# Dictionaries in Python
thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
}
# Access specific value for the key
print(thisdict["brand"])
print(thisdict.keys())
# Update the dictionary
thisdict.update({"year": 2022})
# Add key-value pair in the dictionary
thisdict["color"] = "White"
print(thisdict)
# Remove something from dictionary
thisdict.pop("color")
print(thisdict)

# If-Else Statement
a = 4
if (a < 3):
    print("True")
elif (a == 4):
    print("Exactly 4")
else:
    print("False")

# Function calling in python


def my_function(choice):
    if (choice == 'Yes'):
        print("I am in the function")
    else:
        print("Plese call me")
```

```
# Function calling  demo
choice = input("Call that function? 'Yes' or 'No': ")
my_function(choice)
```