Python Basics Questions

```
1.What is Python, and why is it popular ?

   ->  Python is a high-level, interpreted, object-oriented programming
   language known for its readability and ease of use. It was created by Guido
   van Rossum and first released in 1991.

 >Python is popular for:
    >>1. Easy to read and understand :Python has clean, English-like syntax
    that makes it easy to write and read code.
    >>2. Interpreted Language :Python code runs line-by-line, which simplifies
    debugging and testing.
    >>3. Object-Oriented Programming Language: Supports OOP principles like
    classes and inheritance for modular, reusable code.
    >>4. Free and open Source : Python is freely available for use, and its
    source code can be modified and shared.
    >>5. Versatile and Extensible :Can be used in web, AI, data science, games,
    and extended with C/C++ code.
    >>6. Multi-platform : Works across all major OS like Windows, macOS, and
    Linux without code changes.

2.What is an interpreter in Python?

  >Interpreters are the computer program that will convert the source code or
  an high level language into intermediate code (machine level language). It is
  also called translator in programming terminology. Interpreters executes each
  line of statements slowly. This process is called Interpretation.

  >An interpreter in Python is a program that reads and executes code
  line-by-line rather than compiling the entire code at once.

  >*Working of Interpreter*

  >>source code ---> interpreter--->Executable Code----> next line ----->
  sourcecode.....

   >>You write Python code (.py file).
```

1.What is Python, and why is it popular ?

-> Python is a high-level, interpreted, object-oriented programming language known for its readability and ease of use. It was created by Guido van Rossum and first released in 1991.

> Python is popular for:
>
> 1. Easy to read and understand :Python has clean, English-like syntax that makes it easy to write and read code.
> 2. Interpreted Language :Python code runs line-by-line, which simplifies debugging and testing.
> 3. Object-Oriented Programming Language: Supports OOP principles like classes and inheritance for modular, reusable code.
> 4. Free and open Source : Python is freely available for use, and its source code can be modified and shared.
> 5. Versatile and Extensible :Can be used in web, AI, data science, games, and extended with C/C++ code.
> 6. Multi-platform : Works across all major OS like Windows, macOS, and Linux without code changes.

2.What is an interpreter in Python?

> Interpreters are the computer program that will convert the source code or an high level language into intermediate code (machine level

```
    >>The Python interpreter reads one line at a time.


3.What are pre-defined keywords in Python?
   > Pre-defined keywords in Python are reserved words that have special
   meaning in the language. You cannot use them as variable names, function
   names, or identifiers.

  > Key Points:
    >1.   Fixed meaning :   Keywords perform specific roles in the syntax e.g.,
    if, while, def.
    >2.   Case-sensitive :True is valid, but true is not.
    >3.   Cannot be redefined: You can't use them for custom variable or
    function names.

  >Examples : if, else, elif, while, for, break, continue, pass,  def, return,
  lambda, class

4.Can keywords be used as variable names?

  >No, you cannot use keywords as variable names in Python.

  >Reason:
    >Python keywords are reserved words that have special meanings in the
    language syntax. Using them as variable names would confuse the interpreter

5.What is mutability in Python?

 ->Mutability refers to whether or not an object's value can be changed after
 it is created.

Mutable: Can be changed (e.g. lists, dictionaries).

Immutable: Cannot be changed (e.g. strings, tuples).

6.Why are lists mutable, but tuples are immutable?

  ->Lists are designed for dynamic data – you can add, remove, or change
  elements.
Tuples are used for fixed data – once created, their contents cannot be altered
This immutability allows tuples to be used as dictionary keys and enhances
performance and safety.

7.What is the difference between "==" and "is" operators in Python?
```

language). It is also called translator in programming terminology. Interpreters executes each line of statements slowly. This process is called Interpretation.

An interpreter in Python is a program that reads and executes code line-by-line rather than compiling the entire code at once.

Working of Interpreter

source code ---> interpreter--->Executable Code----> next line -----> sourcecode.....

You write Python code (.py file).

The Python interpreter reads one line at a time.

3.What are pre-defined keywords in Python?

Pre-defined keywords in Python are reserved words that have special meaning in the language. You cannot use them as variable names, function names, or identifiers.

Key Points:

1. Fixed meaning : Keywords perform specific roles in the syntax e.g., if, while, def.
2. Case-sensitive :True is valid, but true is not.
3. Cannot be redefined: You can't use them for custom variable or function names.

Examples : if, else, elif, while, for, break, continue, pass, def, return, lambda, class

4.Can keywords be used as variable names?

No, you cannot use keywords as variable names in Python.

Reason: Python keywords are reserved words that have special meanings in the language syntax. Using them as variable names

```
   ->== checks if values are equal (content comparison).
    is checks if two variables point to the same object in memory (identity
    comparison).

8.What are logical operators in Python ?

 ->Logical operators are used to combine boolean expressions:

and – True if both conditions are true

or – True if at least one condition is true

not – Inverts the boolean value.

9.What is type casting in Python?

->Type casting is converting one data type into another.

10.What is the difference between implicit and explicit type casting?

  -> Implicit casting: Done automatically by Python.
e.g.
x = 5 + 3.0  # x becomes 8.0 (int to float)

Explicit casting: Manually done by the programmer.
e.g.
x = int("7")  # Converts string to integer.

11.What is the purpose of conditional statements in Python?

 ->Conditional statements are used to make decisions in code based on
 conditions.
They allow the program to execute different blocks depending on whether a
condition is True or False.

12.How does the elif statement work ?

->elif (short for "else if") allows checking multiple conditions in an if block

13.What is the difference between for and while loops?

 ->for loop: Used when number of iterations is known or finite (e.g., iterating
 through a list).
while loop: Used when the condition is checked continuously and may run
indefinitely until the condition is false.
```

> would confuse the interpreter.

5.What is mutability in Python?

->Mutability refers to whether or not an object’s value can be changed after it is created.

Mutable: Can be changed (e.g. lists, dictionaries).

Immutable: Cannot be changed (e.g. strings, tuples).

6.Why are lists mutable, but tuples are immutable?

->Lists are designed for dynamic data — you can add, remove, or change elements. Tuples are used for fixed data — once created, their contents cannot be altered. This immutability allows tuples to be used as dictionary keys and enhances performance and safety.

7.What is the difference between “==” and “is” operators in Python?

->== checks if values are equal (content comparison). is checks if two variables point to the same object in memory (identity comparison).

8.What are logical operators in Python ?

->Logical operators are used to combine boolean expressions:

and – True if both conditions are true

or – True if at least one condition is true

not – Inverts the boolean value.

9.What is type casting in Python?

->Type casting is converting one data type into another.

10.What is the difference between implicit and explicit type casting?

-> Implicit casting: Done automatically by Python. e.g. x = 5 + 3.0 # x becomes 8.0 (int to float)

Explicit casting: Manually done by the programmer. e.g. x = int("7") # Converts string to integer.

11.What is the purpose of conditional statements in Python?

```
14.Describe a scenario where a while loop is more suitable than a for loop.

 -> Scenario: Reading user input until they type "exit".
e.g.:
user_input = ""
while user_input != "exit":
    user_input = input("Type 'exit' to quit: ")

Here, you don’t know how many times the loop will run — perfect for a while
loop.
```

->Conditional statements are used to make decisions in code based on conditions. They allow the program to execute different blocks depending on whether a condition is True or False.

12.How does the elif statement work ?

->elif (short for "else if") allows checking multiple conditions in an if block.

13.What is the difference between for and while loops?

->for loop: Used when number of iterations is known or finite (e.g., iterating through a list). while loop: Used when the condition is checked continuously and may run indefinitely until the condition is false.

14.Describe a scenario where a while loop is more suitable than a for loop.

-> Scenario: Reading user input until they type "exit". e.g.: user_input = "" while user_input != "exit": user_input = input("Type 'exit' to quit: ")

Here, you don’t know how many times the loop will run — perfect for a while loop.

Practical Questions

```
#Write a Python program to print "Hello, World!"

print("Hello, Worrld!")
```

```
Hello, Worrld!
```

```
#write a python program that displays your name and age.

Name = "Kamre Alam"
Age = "27 years old"
print("Neme:", Name)
print("Age:", Age)
```

```
Neme: Kamre Alam
Age: 27 years old
```

```
#write code to print all the pre-define keywords in python using the keyword library.

import keyword
print("python keywords:")
print(keyword.kwlist)
```

```
python keywords:
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for'
```

```
#write a program that cheaks if a given words is a python keywords.

ord = input("Enter a word:")
if keyword.iskeyword(ord):
    print(ord,"is a python keyword.")
else:
    print(ord,"is not a python keyword.")
```

```
Enter a word: Kamre Alam
 Kamre Alam is not a python keyword.
```

```
#creat a list and tuple in python, and demonstrate how attempting to change an element works differently for each.

my_list = [1, 2, 3, 4, 5]
my_tuple = (1, 2, 3, 4, 5)

my_list[0] = 100
print("Modified List:", my_list)
```

```
Modified List: [100, 2, 3, 4, 5]
```

```
#write a function to demonstrate the behavior of mutable and immutable arguments.

def modify_list(lst):
    lst.append(10)

def modify_number(num):
    num += 10
    return num
```

```
my_list = [1, 2, 3]
my_number = 5

modify_list(my_list)
new_num = modify_number(my_number)

print("Modified List:", my_list)
print("Original Number (immutable):", my_number)
print("Modified Number:", new_num)
```

```
Modified List: [1, 2, 3, 10]
Original Number (immutable): 5
Modified Number: 15
```

```
from re import A
#write a program that performs basic arithmetic oprations on two user-input numbers.

a = float(input("Enter first number:"))
b = float(input("Enter second number:"))

print("sum:", a + b)
print("diffrence:", a - b)
print("product:", a * b)
print("quotient:", a / b)
```

```
Enter first number:35
Enter second number:32
sum: 67.0
diffrence: 3.0
product: 1120.0
quotient: 1.09375
```

```
#write a program to demonstrate the use of logical oprators.

a = 20
b = 30

print(a > 10 and b < 50)
print(a > 10 or b < 50)
```

```
print(not a > 10)
```

```
True
True
False
```

```
#write a python program to convert user input from sring to integer, flot, and boolean types.

user_input = input("enter something: ")

integer_value = int(user_input)
float_value = float(user_input)
boolean_value = bool(user_input)

print("Integer:", integer_value)
print("Float:", float_value)
print("Boolean:", boolean_value)
```

```
enter something: 7524
Integer: 7524
Float: 7524.0
Boolean: True
```

```
#write code to demonstrate type casting with list elements.

str_list = ["1", "2", "3", "4", "5"]
int_list = [int(x) for x in str_list]

print("Integer List:", int_list)
```

```
Integer List: [1, 2, 3, 4, 5]
```

```
#write a program that cheaks if a number is positive, negative, or zero.

num = float(input("Enter a number:"))

if num > 0:
    print("positive number")
elif num < 0:
    print("negative number")
```

```
else:
    print("zero")
```

```
Enter a number:4275
positive number
```

```
#write a for loop to print numbers from 1 to 10.

for i in range(1,11):
    print(i)
```

```
1
2
3
4
5
6
7
8
9
10
```

```
#write a python program to find the sum of all even numbers between 1 and 50.

total = 0
for i in range(2, 51, 2):
    total += i

print("Sum of even numbers between 1 and 50:", total)
```

```
Sum of even numbers between 1 and 50: 650
```

```
#write a program to reverse a string using a while loop.

next = input("Enter a string:")
reverse = ""
length = len(next)

while length > 0:
    reverse += next[length - 1]
    length -= 1
```

```
print("Reversed string:", reverse)
```

```
Enter a string:KAMRE
Reversed string: ERMAK
```

```
# Write a Python program to calculate the factorial of a number provided by the user using a while loop

num = int(input("Enter a number: "))
fact = 1
i = 1

while i <= num:
    fact *= i
    i += 1

print("Factorial:", fact)
```

```
Enter a number: 5
Factorial: 120
```