Basics Questions of Python

Python Basics Questions:-

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

Ans.:- Python is a high-level, interpreted programming language known for its simple syntax and readability.It is popular due to its ease of use, wide range of libraries, versatility (web development, data science, AI, etc.),and large community support.

2. What is an interpreter in Python?

Ans.:- An interpreter in Python is a program that reads and executes code line by line.Python uses an interpreter to convert source code into machine-executable instructions during runtime.

3. What are pre-defined keywords in Python?

Ans.:- Pre-defined keywords are reserved words in Python that have special meanings,such as if, else, while, for, def, class, True, False, etc. They cannot be used as variable names.

4. Can keywords be used as variable names?

Ans.:- No, keywords cannot be used as variable names because they are reserved for specific functions in the language

5. What is mutability in Python?

Ans.:- Mutability refers to whether an object's content can be changed after it is created.If it can be changed, it's mutable (e.g., lists); if it cannot, it's immutable (e.g., tuples, strings).

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

Ans.:- Lists are mutable because they are designed to allow changes in their contents (adding/removing/modifying elements).Tuples are immutable to ensure data integrity and allow them to be used as dictionary keys.

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

Ans.:- == checks value equality (whether two variables have the same value). is checks identity equality (whether two variables point to the same object in memory).

8. What are logical operators in Python?

Ans.:- Logical operators are used to combine conditional statements: and: True if both conditions are true.

or: True if at least one condition is true.

not: Reverses the result (True becomes False and vice versa).

9. What is type casting in Python?

Ans.:- Type casting is the process of converting one data type into another, such as int(), float(), or str().

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

Ans.:- Implicit casting is done automatically by Python (e.g., converting an integer to a float).

Explicit casting is done manually by the programmer using functions like int(), float(), or str().

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

Ans.:- Conditional statements are used to make decisions in code.They execute different code blocks based on whether a condition is True or False.

12. How does the elif statement work?

Ans.:- The elif statement stands for "else if". It is used to check multiple conditions after an if statement.If the if condition is false, it checks each elif condition in order.

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

Ans.:- for loop is used when the number of iterations is known (e.g., iterating over a list). while loop is used when the number of iterations is unknown and depends on a condition.

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

Ans.:- A while loop is more suitable when you don’t know how many times the loop should run, such as:

password = '' while password != '1234': password = input("Enter password: ")

Practical Questions:-

1. Write a Python program to print ″Hello, World″

```
Ans.:- print(″Hello, World″)
```

2. Write a Python program that displays your name and age

```
Ans.:- name = ″Ravi Rai″
age = 25
print(″Name:″, name)
print(″Age:″, age)
```

3. Write code to print all the pre-defined keywords in Python using the keyword library

```
Ans.:- import keyword

print(″Python Keywords are:″)
print(keyword.kwlist)
```

4. Write a program that checks if a given word is a Python keyword

```
Ans.:- import keyword

word = input(″Enter a word: ″)
if keyword.iskeyword(word):
  print(f″'{word}′ is a Python keyword.″)
else:
  print(f″'{word}′ is NOT a Python keyword.″)
```

5. Create a list and tuple in Python, and demonstrate how attempting to change an element works differently for each.

```
Ans.:- ‴# Creating a list
my_list = [10, 20, 30]
print(″Original List:″, my_list)

# Changing an element in the list (mutable)
‴my_list[1] = 200‴
print(″Modified List:″, my_list)

# Creating a tuple
my_tuple = (10, 20, 30)
print(″Original Tuple:″, my_tuple)

# Attempting to change an element in the tuple (immutable)
try:
  my_tuple[1] = 200
except TypeError as e:
  print(″Error:″, e)‴
```

6. Write a function to demonstrate the behavior of mutable and immutable arguments

```
Ans.:- # Immutable example
def change_number(n):
  n = n + 10
  print(″Inside function (immutable):″, n)

# Mutable example
def change_list(lst):
   lst.append(4)
  print("Inside function (mutable):", lst)

# Calling with immutable type
num = 5
change_number(num)
print("Outside function (immutable):", num)

# Calling with mutable type
my_list =[1, 2, 3]
change_list(my_list)
print("Outside function (mutable):", my_list)
```

7. Write a program that performs basic arithmetic operations on two user-input numbers

Ans.:-

```
# Taking input from user
a = float(input("Enter first number: "))
b = float(input("Enter second number: "))

# Performing arithmetic operations
print("Addition:", a + b)
print("Subtraction:", a - b)
print("Multiplication:", a * b)
print("Division:", a / b)
print("Modulus:", a % b)
```

8. Write a program to demonstrate the use of logical operators

Ans.:-

```
a = True
b = False
# Logical AND
print("a and b:", a and b)
```

```
# Logical OR
print("a or b:", a or b)
# Logical NOT
print("not a:", not a)
print("not b:", not b)
```

9. Convert user input from string to integer, float, and boolean types

Ans.:-

```
user_input = input("Enter a number: ")

int_val = int(user_input)
float_val = float(user_input)
bool_val = bool(user_input)

print("As Integer:", int_val)
print("As Float:", float_val)
print("As Boolean:", bool_val)
```

10. Type casting with list elements

Ans.:-

```
my_list = ['10', '20', '30']
int_list = [int(x) for x in my_list]  # Convert all to integers
print("Original list:", my_list)
print("After type casting:", int_list)
```

11. Check if a number is positive, negative, or zero

Ans.:-

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

if num > 0:
  print("The number is positive.")
elif num < 0:
  print("The number is negative.")
else:
  print("The number is zero.")
```

12. For loop to print numbers from 1 to 10

Ans.:-

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

13. Find the sum of all even numbers between 1 and 50

```
Ans:- total = 0
for i in range(2, 51, 2):
  total += i
print("Sum of even numbers from 1 to 50:", total)
```

14. Reverse a string using a while loop

Ans.:-

```
text = input("Enter a string: ")
reversed_str = ""
index = len(text) - 1

while index >= 0:
  reversed_str += text[index]
  index -= 1

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

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

Ans.:-

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

if num < 0:
  print("Factorial does not exist for negative numbers.")
else:
  while i <= num:
      factorial *= i
      i += 1
  print(f"Factorial of {num} is {factorial}")
```

1. Write a Python program to print "Hello, World"

Ans.:- print("Hello, World")

2. Write a Python program that displays your name and age

Ans.:- name = "Ravi Rai"
age = 25
print("Name:", name)
print("Age:", age)

3. Write code to print all the pre-defined keywords in Python using the keyword library

Ans.:- import keyword

print("Python Keywords are:")
print(keyword.kwlist)

4. Write a program that checks if a given word is a Python keyword

```
Ans.:- import keyword

word = input("Enter a word: ")
if keyword.iskeyword(word):
  print(f"'{word}' is a Python keyword.")
else:
  print(f"'{word}' is NOT a Python keyword.")
```

5. Create a list and tuple in Python, and demonstrate how attempting to change an element works differently for each.

```
Ans.:- '''# Creating a list
my_list = [10, 20, 30]
print("Original List:", my_list)

# Changing an element in the list (mutable)
'''my_list[1] = 200'''
print("Modified List:", my_list)

# Creating a tuple
my_tuple = (10, 20, 30)
print("Original Tuple:", my_tuple)

# Attempting to change an element in the tuple (immutable)
try:
  my_tuple[1] = 200
except TypeError as e:
  print("Error:", e)'''
```

6. Write a function to demonstrate the behavior of mutable and immutable arguments

```
Ans.:- # Immutable example
def change_number(n):
  n = n + 10
  print("Inside function (immutable):", n)
```

```
# Mutable example
def change_list(lst):
  lst.append(4)
  print("Inside function (mutable):", lst)

# Calling with immutable type
num = 5
change_number(num)
print("Outside function (immutable):", num)

# Calling with mutable type
my_list =[1, 2, 3]
change_list(my_list)
print("Outside function (mutable):", my_list)
```

7. Write a program that performs basic arithmetic operations on two user-input numbers

Ans.:-

```
# Taking input from user
a = float(input("Enter first number: "))
b = float(input("Enter second number: "))

# Performing arithmetic operations
print("Addition:", a + b)
print("Subtraction:", a - b)
print("Multiplication:", a * b)
print("Division:", a / b)
print("Modulus:", a % b)
```

8. Write a program to demonstrate the use of logical operators

Ans.:-

```
a = True
b = False

# Logical AND
```

```
print("a and b:", a and b)

# Logical OR
print("a or b:", a or b)

# Logical NOT
print("not a:", not a)
print("not b:", not b)
```

9. Convert user input from string to integer, float, and boolean types

Ans.:-

```
user_input = input("Enter a number: ")

int_val = int(user_input)
float_val = float(user_input)
bool_val = bool(user_input)

print("As Integer:", int_val)
print("As Float:", float_val)
print("As Boolean:", bool_val)
```

10. Type casting with list elements

Ans.:-

```
my_list = ['10', '20', '30']
int_list = [int(x) for x in my_list]  # Convert all to integers
print("Original list:", my_list)
print("After type casting:", int_list)
```

11. Check if a number is positive, negative, or zero

Ans.:-

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

if num > 0:
  print("The number is positive.")
elif num < 0:
  print("The number is negative.")
else:
  print("The number is zero.")
```

12. For loop to print numbers from 1 to 10

Ans.:-

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

13. Find the sum of all even numbers between 1 and 50

Ans:-

```
total = 0
for i in range(2, 51, 2):
  total += i
print("Sum of even numbers from 1 to 50:", total)
```

14. Reverse a string using a while loop

Ans.:-

```
text = input("Enter a string: ")
reversed_str = ""
index = len(text) - 1

while index >= 0:
  reversed_str += text[index]
  index -= 1

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

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

Ans.:-

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

```
if num < 0:
  print(″Factorial does not exist for negative numbers.″)
else:
  while i <= num:
      factorial *= i
      i += 1
  print(f″Factorial of {num} is {factorial}″)
```