MCQ

1. Which of the following is the correct syntax for an `if` statement in Python?

A) `if x:`

B) `if x = 5:`

C) `if x == 5`

D) `if x == 5:`

Answer: D) `if x == 5:`

2. Which of the following statements is true about `if-elif-else` structure?

- A) Only one `if` and one `else` statement are allowed.
- B) Multiple `if` statements are required.
- C) `elif` is always required.
- D) None of the above

Answer: A) Only one `if` and one `else` statement are allowed.

3. Choose the correct code to check if a number is divisible by both 2 and 5.

- A) `if number % 2 and number % 5 == 0:`
- B) `if number % 2 == 0 and number % 5 == 0:`
- C) `if number / 2 == 0 and number / 5 == 0:`
- D) `if number % 2 == 0 or number % 5 == 0:`

Answer: B) `if number % 2 == 0 and number % 5 == 0:`

4. What will be the output of the following code for `num = -4`?

```
python
Copy code
num = -4
        if num < 0:
                num = -num
                print(num)
```

A) -4

B) 0

C) 4

D) Error

Answer: C) 4

5. **Which of the following loops is most suitable for executing a block of code a specific number of times?**

A) `while`

B) `for`

C) `do-while`

D) `until`

Answer: B) `for`

6. **In a `while` loop, when does the loop condition get checked?**

- A) Only once, before the loop starts
- B) After each iteration
- C) Before each iteration
- D) Only after the loop ends

Answer: C) Before each iteration

7. **How many times will the following code print "Hello"?**

```
for i in range(1, 5):
        print("Hello")
```

A) 3

B) 4

C) 5

D) 6

Answer: B) 4

8. **Which of the following pseudocode represents a `while` loop?**

A) `REPEAT until x > 5`

B) `FOR i = 1 TO 10`

C) `WHILE x < 10 DO`

D) `IF x > 5 THEN`

Answer: C) `WHILE x < 10 DO`

9. Which control structure is best for checking if a number is prime?

- A) `for` loop
- B) `while` loop
- C) `if` statement
- D) Both A and B

Answer: D) Both A and B

10.. In a flowchart, which symbol is used to represent a decision-making step?

A) Rectangle

B) Diamond

C) Circle

D) Oval

Answer: B) Diamond

11. In a decision tree, what do leaf nodes represent?

- A) A decision-making step
- B) A loop or iterative step
- C) The outcome of a series of decisions
- D) A condition to evaluate

Answer: C) The outcome of a series of decisions

12. Which of the following data structures in Python is immutable?

A) List

B) Dictionary

C) Tuple

D) Set

Answer: C) Tuple

13. What will be the output of the following code?

```
numbers = [4, 8, 2, 10]
print(max(numbers))
```

A) 2

B) 4

C) 8

D) 10

Answer: D) 10

14. Which of the following options will return the list `[2, 4, 6, 8]` if `my_list = [2, 4, 6, 8, 10]`?

A) `my_list[0:3]`

B) `my_list[1:5]`

C) `my_list[0:4]`

D) `my_list[0:5]`

Answer: C) `my_list[0:4]`

15. What does `len(my_list)` return if `my_list = [3, 5, 1, 4, 7]`?

- A) 3
- B) 5
- C) 7
- D) 4

Answer: B) 5

16. Which of the following operators is used to compare two strings in Python?

A) `=`

B) `==`

C) `!=`

D) Both B and C

Answer: D) Both B and C

17. What will be the result of the following code?

```
str1 = "Hello"
str2 = "World"
result = str1 + " " + str2
print(result)
```

- A) `HelloWorld`
- B) `Hello World`
- C) `Hello, World`
- D) `World Hello`

Answer: B) `Hello World`

Given `str = "OpenAI"`, what will `str[1:4]` return?

A) `Ope`

B) `pen`

C) `OpeA`

D) `penA`

Answer: B) `pen`

18. Given `str = "OpenAI"`, what will `str[1:4]` return?

A) `Ope`

B) `pen`

C) `OpeA`

D) `penA`

Answer: B) `pen`

19.. Which function can be used to concatenate two strings `s1` and `s2` in Python?

A) `s1 + s2`

B) `concat(s1, s2)`

C) `s1.append(s2)`

D) `s1 & s2`

Answer: A) `s1 + s2`

20. If `s = "machine learning"`, what does `s.find("learn")` return?

A) `True`

B) `6`

C) `7`

D) `-1`

Answer: C) `7`

21. Given the string `str = "abcdef"`, what will `str[:3]` return?

A) `abc`

B) `bcd`

C) `def`

D) `cde`

Answer: A) `abc`

22. Which of the following functions will return the length of the string `str = "Python"`?

- A) `str.size()`
- B) `len(str)`
- C) `str.length`
- D) `length(str)`

Answer: B) `len(str)`

23. If `s = "abcdefg"`, what does `s[2:5]` return?

A) `cde`

B) `bcd`

C) `def`

D) `abc`

Answer: A) `cde`

24. If you have a list `numbers = [10, 20, 30, 40]`, how would you access the second element?

- A) `numbers[1]`
- B) `numbers[2]`
- C) `numbers[-2]`
- D) `numbers[0]`

Answer: A) `numbers[1]`

25. How many times will the following code print "Hello"?

```
for i in range(1, 5):
    print("Hello")
```

A) 3

B) 4

C) 5

D) 6

Answer: B) 4

1. **What is iterative computation?**

Iterative computation repeats a set of instructions multiple times, typically using loops, to perform repetitive tasks.

2. **What is control flow?**

Control flow determines the order in which individual statements, instructions, or function calls are executed in a program. This includes conditional statements and loops.

3. **What is a flowchart and pseudocode?**

- A flowchart is a diagram that represents an algorithm or process. It uses different shapes to denote steps and arrows to indicate the flow of operations.

- Pseudocode is a high-level description of an algorithm written in a style resembling code but meant for humans to understand. It omits specific syntax and is language-independent.

4. **What is a `while` loop in Python?**

A `while` loop continues executing a block of code as long as a specified condition is true.

Example:

```
x = 0
while x < 5:
    print(x)
    x += 1  # Outputs 0 to 4
```

5. **Write Python Code for Interest Calculation (Simple Interest)**

```
principal = 1000
rate = 5  # Interest rate in percentage
time = 2  # Time in years
interest = (principal * rate * time) / 100
print("Simple Interest:", interest)
```

6. **What is debugging? What are errors in Python?**

Debugging is the process of identifying, analyzing, and fixing errors or bugs in a program to make it work correctly.

Errors are problems in a program that stop it from running successfully. Python errors include syntax errors (problems with code structure) and runtime errors (issues during execution).

7. **What are exceptions in Python?**

Exceptions are runtime errors that occur when an operation is impossible to execute (e.g., dividing by zero). Common exceptions include `ZeroDivisionError`, `TypeError`, and `ValueError`.

8. **What is a tuple in Python?**

A tuple is an ordered, immutable collection of items, defined with parentheses `()`. Like lists, elements in tuples are accessed by index.

Example: `my_tuple = (1, 2, 3, 4)`

9. **How do you find the maximum and minimum in a list or tuple?**

Use the `max()` and `min()` functions.

Example:

```
my_list = [10, 20, 30]
print(max(my_list))  # Output: 30
print(min(my_list))  # Output: 10
```

10. **How do you find the mean of a list of numbers?**

Sum all elements with `sum()` and divide by the length with `len()`.

Example:

```
my_list = [10, 20, 30]
mean = sum(my_list) / len(my_list)
print(mean)  # Output: 20
```

11. **What is a dictionary in Python?**

A dictionary is a collection of key-value pairs, defined with curly braces `{}`. Keys are used to access values, making dictionaries fast for lookups.

Example: `my_dict = {'a': 1, 'b': 2}`

12. **How do you compare two strings in Python?**

Use comparison operators like ==, `!=`, <, >. They compare strings lexicographically.

Example:

```
print("apple" == "apple")  # Output: True
print("apple" < "banana")  # Output: True
```

13. **How do you concatenate (join) two strings?**

Use the + operator to concatenate strings.

Example:

```
str1 = "Hello"
str2 = "World"
result = str1 + " " + str2
print(result)  # Output: "Hello World"
```

14. **How do you get a substring in Python?**

Use slicing to get a substring from a string.

Example:

```
my_string = "Hello World"
substring = my_string[0:5]
print(substring)  # Output: "Hello"
```

15. **How do you perform a linear search on a list or tuple?**

Use a loop to check each element. If found, return the index; if not, return -1.

Example:

```
def linear_search(data, target):
        for index, value in enumerate(data):
                if value == target:
                        return index
                        return -1
print(linear_search([1, 2, 3], 2))  # Output: 1
```

16. **What is a string in Python? How do you find the length of a string?**

A string is a sequence of characters enclosed in quotes, either single (`' '`) or double (`" "`).

Use the `len()` function to get the number of characters in a string.

- Example:

```
my_string = "Hello"
print(len(my_string))  # Output: 5
```

17. **How do you convert a string to uppercase or lowercase?**

Use the `upper()` and `lower()` methods.

Example:

```
my_string = "Hello"
print(my_string.upper())  # Output: "HELLO"
print(my_string.lower())  # Output: "hello"
```

18. **How do you check if a substring exists in a string?**

Use the `in` keyword to check for a substring.

Example:

```
my_string = "Hello World"
print("World" in my_string)  # Output: True
```

19. **How do you replace part of a string?**

Use the `replace()` method to replace a substring with another string.

Example:

```
my_string = "Hello World"
new_string = my_string.replace("World", "Python")
print(new_string)  # Output: "Hello Python"
```

20. **How do you join a list of strings into a single string?**

Use the `join()` method with a separator.

Example:

```
python
Copy code
words = ["Hello", "World"]
sentence = " ".join(words)
print(sentence)  # Output: "Hello World"
```

21. **What is bubble sort? How does bubble sort work?**

Bubble sort is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. This process repeats until the list is sorted.

Bubble sort compares each pair of adjacent elements in a list and swaps them if needed, "bubbling" the largest unsorted element to its correct position with each pass.

22. **What is insertion sort? How does insertion sort work?**

Insertion sort is a sorting algorithm that builds a sorted portion of the list one element at a time by inserting each element into its correct position in the sorted part.

It takes each element from an unsorted list and inserts it into its correct position in a growing sorted portion of the list.