Question Bank

* MCQ Questions

1. Is Python case sensitive when dealing with identifiers?
a) yes
b) no
c) machine dependent
d) none of the mentioned
Answer: a
Explanation: Case is always significant.

2. What is the maximum possible length of an identifier?
a) 31 characters
b) 63 characters
c) 79 characters
d) none of the mentioned
Answer: d
Explanation: Identifiers can be of any length.

3. Which of the following is invalid?
a) _a = 1
b) __a = 1
c) __str__ = 1
d) none of the mentioned
Answer: d
Explanation: All the statements will execute successfully but at the cost of reduced readability.

4. Which of the following is an invalid variable?
a) my_string_1
b) 1st_string
c) foo
d) _
Answer: b
Explanation: Variable names should not start with a number.

5. Why are local variable names beginning with an underscore discouraged?
a) they are used to indicate a private variables of a class
b) they confuse the interpreter
c) they are used to indicate global variables
d) they slow down execution
Answer: a
Explanation: As Python has no concept of private variables, leading underscores are used to indicate variables that must not be accessed from outside the class.

6. Which of the following is not a keyword?
a) eval

b) assert
c) nonlocal
d) pass
Answer: a
Explanation: eval can be used as a variable.

7. All keywords in Python are in _________
a) lower case
b) UPPER CASE
c) Capitalized
d) None of the mentioned
Answer: d
Explanation: True, False and None are capitalized while the others are in lower case.

8. Which of the following is true for variable names in Python?
a) unlimited length
b) all private members must have leading and trailing underscores
c) underscore and ampersand are the only two special characters allowed
d) none of the mentioned
Answer: a
Explanation: Variable names can be of any length.

9. Which of the following is an invalid statement?
a) abc = 1,000,000
b) a b c = 1000 2000 3000
c) a,b,c = 1000, 2000, 3000
d) a_b_c = 1,000,000
Answer: b
Explanation: Spaces are not allowed in variable names.

10. Which of the following cannot be a variable?
a) __init__
b) in
c) it
d) on
Answer: b
Explanation: in is a keyword.

11. What is the output of print 0.1 + 0.2 == 0.3?
a) True
b) False
c) Machine dependent
d) Error
View Answer
Answer: b
Explanation: Neither of 0.1, 0.2 and 0.3 can be represented accurately in binary. The

round off errors from 0.1 and 0.2 accumulate and hence there is a difference of 5.5511e-17 between (0.1 + 0.2) and 0.3.

12. Which of the following is not a complex number?
a) k = 2 + 3j
b) k = complex(2, 3)
c) k = 2 + 3l
d) k = 2 + 3J
Answer: c
Explanation: l (or L) stands for long.

13. What is the type of inf?
a) Boolean
b) Integer
c) Float
d) Complex
Answer: c
Explanation: Infinity is a special case of floating point numbers. It can be obtained by float('inf').

14. What does ~4 evaluate to?
a) -5
b) -4
c) -3
d) +3
Answer: a
Explanation: ~x is equivalent to -(x+1).

15.What does ~~~~~~5 evaluate to?
a) +5
b) -11
c) +11
d) -5
Answer: a
Explanation: ~x is equivalent to -(x+1).

16.Which of the following is incorrect?
a) x = 0b101
b) x = 0x4f5
c) x = 19023
d) x = 03964
Answer: d
Explanation: Numbers starting with a 0 are octal numbers but 9 isn't allowed in octal numbers.

17. What is the result of cmp(3, 1)?
a) 1
b) 0
c) True
d) False
Answer: a
Explanation: cmp(x, y) returns 1 if x > y, 0 if x == y and -1 if x < y.

18. Which of the following is incorrect?

a) float(‘inf’)

b) float(‘nan’)

c) float(’56’+’78’)

d) float(’12+34′)

Answer: d

Explanation: ‘+’ cannot be converted to a float.

19. What is the result of round(0.5) – round(-0.5)?
a) 1.0
b) 2.0
c) 0.0
d) None of the mentioned
Answer: b
Explanation: Python rounds off numbers away from 0 when the number to be rounded off is exactly halfway through. round(0.5) is 1 and round(-0.5) is -1.

20. What does 3 ^ 4 evaluate to?
a) 81
b) 12
c) 0.75
d) 7
Answer: d
Explanation: ^ is the Binary XOR operator.

21. Which of the following statements create a dictionary?
a) d = {}

b) d = {‘‘john’’:40, ‘‘peter’’:45}

c) d = {40:’’john’’, 45:’’peter’’}

d) All of the mentioned
Answer: d
Explanation: Dictionaries are created by specifying keys and values.

22. What will be the output of the following Python code snippet?
d = {"john":40, "peter":45}

a) ‘‘john’’, 40, 45, and ‘‘peter’’

b) ‘‘john’’ and ‘‘peter’’

c) 40 and 45

d) d = (40:’’john’’, 45:’’peter’’)

Answer: b
Explanation: Dictionaries appear in the form of keys and values.

23. What will be the output of the following Python code snippet?
d = {"john":40, "peter":45}
"john" in d

a) True
b) False
c) None
d) Error

Answer: a
Explanation: In can be used to check if the key is int dictionary.

24. What will be the output of the following Python code snippet?
d1 = {"john":40, "peter":45}
d2 = {"john":466, "peter":45}
d1 == d2

a) True
b) False
c) None
d) Error

Answer: b
Explanation: If d2 was initialized as d2 = d1 the answer would be true.

25. What will be the output of the following Python code snippet?
d1 = {"john":40, "peter":45}
d2 = {"john":466, "peter":45}
d1 > d2

a) True
b) False
c) Error
d) None
Answer: c
Explanation: Arithmetic > operator cannot be used with dictionaries.

26. Which of the following is a Python tuple?
a) [1, 2, 3]
b) (1, 2, 3)
c) {1, 2, 3}
d) {}
Answer: b
Explanation: Tuples are represented with round brackets.

27. Suppose t = (1, 2, 4, 3), which of the following is incorrect?
a) print(t[3])
b) t[3] = 45
c) print(max(t))
d) print(len(t))
Answer: b
Explanation: Values cannot be modified in the case of tuple, that is, tuple is immutable.

28. What will be the output of the following Python code?

```
>>>t=(1,2,4,3)
>>>t[1:3]
```

a) (1, 2)
b) (1, 2, 4)
c) (2, 4)
d) (2, 4, 3)
Answer: c
Explanation: Slicing in tuples takes place just as it does in strings.

29. What will be the output of the following Python code?

```
>>>t=(1,2,4,3)
>>>t[1:-1]
```

a) (1, 2)
b) (1, 2, 4)
c) (2, 4)
d) (2, 4, 3)
Answer: c
Explanation: Slicing in tuples takes place just as it does in strings.

30. What will be the output of the following Python code?

```
>>>t = (1, 2, 4, 3, 8, 9)
>>>[t[i] for i in range(0, len(t), 2)]
```

a) [2, 3, 9]
b) [1, 2, 4, 3, 8, 9]
c) [1, 4, 8]
d) (1, 4, 8)

Answer: c
Explanation: Execute in the shell to verify.

31. What will be the output of the following Python code?

```
d = {"john":40, "peter":45}
d["john"]
```

a) 40
b) 45
c) “john”
d) “peter”
Answer: a
Explanation: Execute in the shell to verify.

32. What will be the output of the following Python code?

```
>>>t = (1, 2)
>>>2 * t
```

a) (1, 2, 1, 2)
b) [1, 2, 1, 2]
c) (1, 1, 2, 2)
d) [1, 1, 2, 2]
Answer: a
Explanation: * operator concatenates tuple.

33. Which of these about a set is not true?
a) Mutable data type
b) Allows duplicate values
c) Data type with unordered values
d) Immutable data type
Answer: d
Explanation: A set is a mutable data type with non-duplicate, unordered values, providing the usual mathematical set operations.

34. Which of the following is not the correct syntax for creating a set?
a) set([[1,2],[3,4]])
b) set([1,2,2,3,4])
c) set((1,2,3,4))
d) {1,2,3,4}
Answer: a
Explanation: The argument given for the set must be an iterable.

35. What will be the output of the following Python code?

```
nums = set([1,1,2,3,3,3,4,4])print(len(nums))
```

a) 7
b) Error, invalid syntax for formation of set

c) 4
d) 8
Answer: c

Explanation: A set doesn’t have duplicate items.

36. What will be the output of the following Python code?

```
a = [5,5,6,7,7,7]
b = set(a)def test(lst):
    if lst in b:
        return 1
    else:
        return 0for i in    filter(test, a):
    print(i,end=" ")
```

a) 5 5 6
b) 5 6 7
c) 5 5 6 7 7 7
d) 5 6 7 7 7

37. Which of the following statements is used to create an empty set?
a) { }
b) set()
c) [ ]
d) ( )
Answer: b
Explanation: { } creates a dictionary not a set. Only set() creates an empty set.

38. What will be the output of the following Python code?

```
>>> a={5,4}>>> b={1,2,4,5}>>> a<b
```

a) {1,2}
b) True
c) False
d) Invalid operation
Answer: b
Explanation: a<b returns True if a is a proper subset of b.

39. If a={5,6,7,8}, which of the following statements is false?
a) print(len(a))
b) print(min(a))
c) a.remove(5)
d) a[2]=45
Answer: d
Explanation: The members of a set can be accessed by their index values since the elements of the set are unordered.

40. If a={5,6,7}, what happens when a.add(5) is executed?
a) a={5,5,6,7}
b) a={5,6,7}

c) Error as there is no add function for set data type
d) Error as 5 already exists in the set
Answer: b

Explanation: There exists add method for set data type. However 5 isn’t added again as set consists of only non-duplicate elements and 5 already exists in the set. Execute in python shell to verify.

41. What will be the output of the following Python code?
>>> a={4,5,6}>>> b={2,8,6}>>> a+b
a) {4,5,6,2,8}
b) {4,5,6,2,8,6}
c) Error as unsupported operand type for sets
d) Error as the duplicate item 6 is present in both sets
Answer: c
Explanation: Execute in python shell to verify.

*Answer In One Sentence :

1. What is name space in Python?
2. What are local variables and global variables in Python?
3. Is python case sensitive?
4. What is type conversion in Python?
5. Is indentation required in python?
6. What is the difference between Python Arrays and lists?
7. What is a lambda function?
8. What is self in Python?
9. How does break, continue and pass work?
10. How can you randomize the items of a list in place in Python?
11. What are python iterators?
12. How do you write comments in python?
13. What is pickling and unpickling?
14. How will you convert a string to all lowercase?
15. How will you capitalize the first letter of string?
16. How to comment multiple lines in python?
17. What are docstrings in Python?
18. What is the purpose of is, not and in operators?
19. What is a dictionary in Python?
20. What does len() do?
21. Explain split(), sub(), subn() methods of “re” module in Python.
22. What are Python packages?
23. Does Python have OOps concepts?
24. How to import modules in python?

25. How are classes created in Python?
26. What is Polymorphism in Python?
27. Define encapsulation in Python?
28. How do you do data abstraction in Python?
29. Does python make use of access specifiers?
30. How to create an empty class in Python?

* Short Notes Questions :

1. Mention five benefits of using Python?
2. Explain what is Flask & its benefits?
3. Explain the key features of Python?
4. What are the global and local variables in Python?
5. Define modules in Python?
6. Define pickling and unpickling in Python?
7. Write a program to display Fibonacci sequence in Python?
8. Write a program to check whether the given number is prime or not?
9. What is the difference between range and xrange?
10. Does multiple inheritance is supported in Python?
11. Does Python make use of access specifiers?
12. Define Constructor in Python?
13. How can we create a constructor in Python programming?
14. How Does Python Handle Memory Management?
15. What Are The Optional Statements Possible Inside A Try-Except Block In Python?
16. What Is Slicing In Python?
17. What Is %S In Python?
18. What Is The Index In Python?
19. Is A String Immutable Or Mutable In Python?
20. How Many Basic Types Of Functions Are Available In Python?

* Short Answer Questions :

1. How Do We Write A Function In Python?
2. What Is “Call By Value” In Python?
3. What Is “Call By Reference” In Python?
4. Is It Mandatory For A Python Function To Return A Value?Comment?
5. What Does The *Args Do In Python?
6. What Does The **Kwargs Do In Python?
7. Does Python Have A Main() Method?
8. What Is The Purpose Of “End” In Python?

9. When Should You Use The “Break” In Python?
10. What Is The Difference Between Pass And Continue In Python?
11. What Does The Ord() Function Do In Python?
12. What Is Rstrip() In Python?
13. How Is Python Thread Safe?
14. How Does Python Manage The Memory?
15. What Is A Tuple In Python?
16. Is Python List A Linked List?
17. What Is Class In Python?
18. What Are Attributes And Methods In A Python Class?
19. How To Assign Values For The Class Attributes At Runtime?
20. What Is Inheritance In Python Programming?

* Long Answer Questions :

1. What is the difference between lists and tuples?
2. Explain Inheritance in Python with an example.
3. What is a dictionary in Python?
4. What are negative indexes and why are they used?
5. What are split(), sub(), and subn() methods in Python?
6. How are range and xrange different from one another?
7. Explain all file processing modes supported in Python.
8. How are Python arrays and Python lists different from each other?
9. Explain List and Queues with example?
10. Explain All OOPS Concept With example?