Python Beginner Quiz- with answers

Introduction:

This contains quiz questions from Python Beginner Quiz along with the answer keys and some relevant explanation. Head over to the other document if you only want questions.

Quiz questions with answers

1. What is the output of the following?

```
greet = ['hello', 'hi']
for i in greet:
    i.upper()

print greet
```

a) [‘hello’, ‘hi’].
b) [‘HELLO’, ‘HI’].
c) [None, None].
d) none of the mentioned

Answer: a
Explanation:
upper() is a non mutating operation. Also, no where we are updating the original list greet.

2. What is the output of the following?

```
greet = ['hello', 'hi']
for i in x:
    x.append(i.upper())
print(x)
```

a) [‘HELLO’, ‘HI’]
b) [‘hello’, ‘hi’, ‘HELLO’, ‘HI’]
c) [‘hello’, ‘hi’]
d) none of the mentioned

Answer: d

Explanation:

The loop does not terminate as new elements are being added to the list in each iteration. It is a dangerous coding practice to modify the very sequence you are iterating on.

3. What is the output of the following?

```
countf = 0
while True:
    if count%2 == 1:
        break
    print count
    count += 2
```

a) 1
b) 1 2
c) 1 2 3 4 5 6 …
d) 1 3 5 7 9 11 …

Answer: d

Explanation:

count starts off with being zero and every loop will increment count by 2. Hence count will always remain to be an even number

4. What is the output of the following?

```
count = 4
while True:
    if count%3 == 0:
        break
    print count
    count += 1
```

a) 4 5
b) 4 5 6
c) error
d) none of the above

Answer: a

Explanation:

The numbers 4 and 5 are printed. The next value of count is 6 which is divisible by 3 and hence control exits the loop.

5. What is the output of the following?

```
count = 1
while False:
    if count%2 == 0:
        break
    print count
    count += 1
```

a) 1
b) 1 3 5 7 …
c) 1 2 3 4 …
d) none of the mentioned

Answer: d
Explanation:
while False will never allow the control to enter while loop

6. What is the output of the following?

```
greet = "hello"
for i in greet:
    print i
```

a) h e l l o
b) error
c) infinite loop
d) None of the above

Answer: a
Explanation:
String too is a sequence. So a for is capable of iterating through each of the items of string which happen to be individual characters.

7. What is the output of the following?

```
x = "abcdef"
i = "a"
while i in x:
    print i
```

a) no output
b) i i i i i i … (infinite loop)
c) a a a a a a ...(infinite loop)
d) a b c d e f

Answer: c

Explanation:

Note: this is a while loop as i is always a and the string x has i in it, the loop will run for ever.

8. What is the output of the following?

```
greet = 'hello'
for item in greet:
    print item.upper()
```

a) h e l l o
b) H E L L O
c) error
d) None of the above

Answer: b

Explanation:

We are printing the upper() of each character of the string.

9. What is the output of the following?

```
x = 'abcd'
for i in x:
   i.upper()
    print i
```

a) a b c d
b) A B C D
c) a B C D
d) error

Answer: a

Explanation:

upper() is not a mutating operation

10. What is the output of the following?

```
x = 'abcd'
for i in range(len(x)):
    print(i)
```

a) a b c d
b) 0 1 2 3
c) error

d) none of the mentioned

Answer: b

Explanation:

range starts from 0 and stops one before the parameter passed with a default increment of 1.

11. What is the output of the following?

```
x = 'abcd'
for i in range(len(x)):
    print i.upper()
```

a) a b c d
b) 0 1 2 3
c) error
d) 1 2 3 4

Answer: c

Explanation:

Objects of type int have no attribute upper().

12. What is the output of the following?

```
x = 'abcd'
for i in range(len(x)):
    x[i].upper()
print (x)
```

a) abcd
b) ABCD
c) error
d) none of the mentioned

Answer: a

Explanation:

Changes do not happen in-place, rather a new instance of the string is returned.

13. What is the output of the following?

```
greet = 'hello'
for i in range(len(greet)):
     print greet[i].upper()
```

a) H E L L O
b) h e l l o
c) error
d) none of the mentioned

Answer: a

Explanation:

i will now act as the index of greet

14. What is the output of the following?

```
greet = 'hello'
for i in range(len(greet)):
    greet = 10
    print x
```

a) h
b) 10 10 10 10 10
c) error
d) none of the mentioned

Answer: c

Explanation:

range() is computed only at the time of entering the loop and hence the print statement will print 10 5 times

15. What is the output of the following?

```
greet = 'hello'
b = 'what'
for item in greet:
    b = item.upper()
print b
```

a) O
b) HELLO
c) what
d) None of the above

Answer: O

Explanation:

Note that print b is outside of the for loop

16. What is the output of the following?

```
dial = 100
for num in dial:
    print num
```

a) 1 0 0
b) 100
c) error
d) none of the mentioned

Answer: c
Explanation:
dial is an integer and we cannot iterate over an integer. The loop would have worked if dial was a string

17. What is the output of the following (ignore ordering)?

```
d = {'name': 'Aditya', 'email': 'sp.aditya@gmail.com', 'city':
'Bengaluru'}
for i in d:
    print i
```

a) Aditya sp.aditya@gmail.com Bengaluru
b) name email city
c) name Aditya email sp.aditya@gmail.com city Bengaluru
d) none of the mentioned

Answer: a
Explanation:
by default for iterates over the keys of a dictionary

18. What is the output of the following?

```
d = {'name': 'Aditya', 'email': 'sp.aditya@gmail.com', 'city':
'Bengaluru'}
for x, y in d:
    print(x, y)
```

a) Aditya sp.aditya@gmail.com Bengaluru
b) name email city
c) name Aditya email sp.aditya@gmail.com city Bengaluru
d) none of the mentioned

Answer: d

Explanation:

By default for iterates over the keys of a dictionary. In the above case, we are attempting to unpack a key into two variables x,y which will not work.

19. What is the output of the following?

```
d = {'name': 'Aditya', 'email': 'sp.aditya@gmail.com', 'city':
'Bengaluru'}
for x, y in d.items():
    print(x, y)
```

a) Aditya sp.aditya@gmail.com Bengaluru
b) name email city
c) name Aditya email sp.aditya@gmail.com city Bengaluru
d) none of the mentioned

Answer: c

Explanation:

Loops over key, value pairs.

20. What is the output of the following?

```
d = {'name': 'Aditya', 'email': 'sp.aditya@gmail.com', 'city':
'Bengaluru'}
for x in d.keys():
    print(d[x])
```

a) Aditya sp.aditya@gmail.com Bengaluru
b) name email city
c) name Aditya email sp.aditya@gmail.com city Bengaluru
d) none of the mentioned

Answer: a

Explanation:

Loops over the keys and prints the values.

21. What is the output of the following?

```
d = {'name': 'Aditya', 'email': 'sp.aditya@gmail.com', 'city':
'Bengaluru'}
for x in d.values():
```

```
    print(x)
```

a) Aditya sp.aditya@gmail.com Bengaluru
b) name email city
c) name Aditya email sp.aditya@gmail.com city Bengaluru
d) none of the mentioned

Answer: a
Explanation:
Loops over the values.

22. What is the output of the following(ignore the order)?

```
d = {0: 2, 1: 2, 2: 2}
for x in d.values():
    print(d[x])
```

a) 0 1 2
b) 2 2 2
c) error
d) none of the mentioned

Answer: b
Explanation:
2 also happens to be key which has a value of 2 hence each time the loop runs, it will print out 2 which is actually the value of item with key 2 :)

23. What is the output of the following?

```
d = {0, 1, 2}
for x in d.values():
    print(x)
```

a) 0 1 2
b) None None None
c) error
d) none of the mentioned

Answer: c
Explanation:
Without key, value pairs the presence of { } will create a "set"

24. What is the output of the following?

```
for i in range(0):
    print i
```

a) 0
b) no output
c) error
d) none of the mentioned

Answer: b
Explanation:
range(0) returns an empty list.

25. What is the output of the following?

```
for num in range(int(2.0)):
    print num
```

a) 0.0 1.0
b) 0 1
c) error
d) none of the mentioned

Answer: b
Explanation:
range(int(2.0)) is the same as range(2).

26. What is the output of the following?

```
x = float('inf')
print x
```

a) Error
b) inf
c) None of the above

Answer: b
Explanation:
inf is a valid float number

27. What is the output of the following?

```
for num in [1, 2, 3, 4][::-1]:
    print  num
```

a) 1 2 3 4
b) 4 3 2 1
c) error
d) none of the mentioned

Answer: b
Explanation:
[::-1] reverses the list.

28. What is the output of the following?

```
s = ''.join(reversed(list('abcd')))
print x
```

a) abcd
b) dcba
c) error
d) none of the mentioned

Answer: b
Explanation:
‘’.join(reversed(list(‘abcd’))) reverses a string.

29. What is the output of the following?

```
for c in 'hello'[::-1]:
    print c
```

a) h e l l o
b) o l l e h
c) error
d) none of the mentioned

Answer: b
Explanation:
[::-1] reverses the string.

30. What is the output of the following?

```
s = ''
for c in s:
    print c
```

a) None
b) nothing is printed
c) error
d) none of the mentioned

Answer: b
Explanation:
The string does not have any character to loop over.

31. What is the output of the following?

```
count = 2
for i in range(count):
    count += 1
    print count
```

a) 0 1 2 3 4 …
b) 0 1
c) 3 4
d) 0 1 2 3

Answer: c
Explanation:
Range(count) gets calculated the first time and then only count is incremented

32. What is the output of the following?

```
title = "the discovery of India"
for i in title.split():
    print i
```

a) The loop works for 4 times
b) The number of times loop works is same as the length of title
c) error
d) none of the above

Answer: a
Explanation:
split by default splits the string by whitespaces

33. What is the output of the following?

```
title = "the discovery of India"
for i in title:
    print i
```

a) The loop works for 4 times
b) The number of times loop works is same as the length of title
c) error
d) none of the above

Answer: b
Explanation:
string title is a sequence too

34. What is the output of the following?

```
a = [0, 1, 2, 3]
for a[-1] in a:
    print a[-1]
```

a) 0 1 2 3
b) 0 1 2 2
c) 3 3 3 3
d) error

Answer: b
Explanation:
The value of a[-1] changes in each iteration.

35. What is the output of the following?

```
a = [0, 1, 2, 3]
for a[0] in a:
    print(a[0])
```

a) 0 1 2 3
b) 0 1 2 2

c) 3 3 3 3
d) error

Answer: a
Explanation:
The value of a[0] changes in each iteration. Since the first value that it takes is itself, there is no visible error in the current example.

36. What is the output of the following?

```
print "heLLO".capitalize()
```

a) Hello
b) HeLLO
c) HELLO
d) None of the above

Answer: a
Explanation:
The first letter of the string is converted to uppercase and the others are converted to lowercase.

37. What is the output of the following?

```
print 'Hello there'.replace('ello', 'ELLO')
```

a) Hello there
b) HELLO there
c) No output
d) none of the mentioned

Answer: b
Explanation:
All occurrences ello are replace by ELLO.

38. What is the output of the following?

```
print 'Hello hello there'.replace('ello', 'ELLO')
```

a) Hello there
b) HELLO hELLO there
c) No output
d) none of the mentioned

Answer: b

Explanation:

All occurrences of ello are replaced by ELLO.

39. What is the output of the following?

```
print 'Hello hello HEllo there'.replace('ello', 'ELLO')
```

a) Hello hELLO HELLO there
b) HELLO hELLO hEllo there
c) No output
d) none of the mentioned

Answer: b

Explanation:

All occurrences of ello are replaced by ELLO. its case sensitive

40. What is the output of the following?

```
print 'abcdefcdghcd'.split('cd')
```

a) [‘ab’, ‘ef’, ‘gh’].
b) [‘ab’, ‘ef’, ‘gh’, ”].
c) (‘ab’, ‘ef’, ‘gh’)
d) (‘ab’, ‘ef’, ‘gh’, ”)

Answer: b

Explanation:

The given string is split and a list of substrings is returned. Note the presence of last empty ''

41. What will be the output?

```
values = [[3, 4, 5, 1 ], [33, 6, 1, 2]]

for row in values:
   row.sort()
    for ele in row:
      print ele
```

a) The program prints two rows 3 4 5 1 followed by 33 6 1 2
b) The program prints on row 3 4 5 1 33 6 1 2
c) The program prints two rows 3 4 5 1 followed by 33 6 1 2
d) The program prints two rows 1 3 4 5 followed by 1 2 6 33

Answer: d

Explanation:

row.sort is a mutating operation.

42. What will be the output?

```
l = [[1,4,2],[5,2,6],[3,6,1],[2,6,3]]
l.sort()
print l
```

a) [[1,4,2],[5,2,6],[3,6,1],[2,6,3]]
b) [[1,4,2] ,[2,6,3],[3,6,1] ,[5,2,6]]
c) error
d) none of the above

Answer: b

Explanation:

sort will automatically sort a sequence of sequences by the first item of the inner sequence

43. What will be the output?

```
points = [[1, 2], [3, 1.5], [0.5, 0.5]]
points.sort()
print points
```

a) [[1, 2], [3, 1.5], [0.5, 0.5]].
b) [[3, 1.5], [1, 2], [0.5, 0.5]].
c) [[0.5, 0.5], [1, 2], [3, 1.5]].
d) [[0.5, 0.5], [3, 1.5], [1, 2]].

Answer: c

Explanation:

sequence of sequence sorting sorts on the first item by default

44. What is the data type of (1)?

a) Tuple
b) Integer
c) List
d) Both tuple and integer

Answer: b

Explanation:

A tuple of one element must be created as (1,).

45. If a=(1,2,3,4), a[1:-1] is

a) Error, tuple slicing doesn't exist
b) [2,3].
c) (2,3,4)
d) (2,3)

Answer: d

Explanation:

a[1:-1] returns (2,3).

46. What is the output of the following piece of code?

```
a=("Hello") * 3
print  a
```

a) ('Hello','Hello','Hello')
b) * Operator not valid for tuples
c) 'HelloHelloHello'
d) Syntax error

Answer: c

Explanation:

Here ("Hello") is a string not a tuple because there is no comma after the element.

47. What is the output of the following code?

```
a=("Hello","Hi")
del a[0]
print a
```

a) ("Hi",)
b) Error
c) None of the above
d) depends on the data type of the item

Answer: b

Explanation:

'tuple' objects are immutable

48. What is the output of the following code?

```
a_t=([1,2,3],[2,3,4])
del a_t[0][0]
print a_t
```

a) Error as tuples are immutable
b) ([2,3],[2,3,4])
c) ([2,3,4],)
d) None of the above

Answer: b

Explanation:

mutation of tuples is different from the mutation of the items of the tuple

49. How to get all the values of a dictionary

a) value()
b) values()
c) getvalues()
d) None of the above

Answer: b

50. a.extend(b) method of list helps to extend "a" list by adding the items of the "b" to the end of "a"

a)True
b)False

Answer: a

51. a = Car() . Consider "a" to be an object of Car(). How do we implement custom equals operation?

a) __eq__
b) equals
c) _eq_
d) None of the above

Answer: a

52. a = “5” * 5 results in value of a being

a) 55555
b) Need to convert int to string
c) Error
d) None of the above

Answer: a

Explanation:

Multiplication on strings is a supported operation

53. Is the below code pythonic?

```
a = []
for itr in range(10):
      a.append(itr)
```

a) No
b) Yes

Answer: a

54. If a = “hello” , then a[-1] would give a result of

a) o
b) negative index is not allowed
c) l
d) None of the above

Answer a

55. if a = {‘a’:’b’, ‘c’:’d’} then a[1] would result in

a) (‘a’,’b’)
b) Error. A dictionary cannot be accessed through index
c) None of the above
d) Would return back the first index

Answer: b

56. a =[1,2,3,4] b =5 then a.insert(1, b) would result in

a) [1,5,2,3,4]
b) [1,5,3,4]
c) Error as there is an item at 1 already

d) None of the above

Answer: a

57. a = ‘hello’ then a[::-1] would

a) reverse the string
b) start and end are mandatory
c) atleast the start is mandatory
d) –ve is not allowed as the stepper

Answer: a

58. The following helps in opening a file inpython

a) open(<filepath>,<mode>)
b) openfile(<filepath>,<mode>)
c) filehandle(<filepath>,<mode>)
d) None of the above

Answer: a

59. In a list, items can repeat

a) True
b) False

Answer: a
Explanation:

A list can have duplicate values

60. Writing a generator will help

a) In providing an iterator which can be used in a for loop
b) There is no such thing as a generator
c) Generators will return a list
d) None of the above

Answer: a

61. If a =[1,2,3] and b=a then

a) id(a) == id(b)
b) a and b are different objects
c) assignment cannot be done, we need to use list() function
d) None of the above

Answer a

62. By setting " " we can provide a list of directories for python to search for module imports

a) sys.args
b) sys.path
c) os.path
d) None of the above

Answer: b

63. If a={'a':'b','c':'d'} is dictionary then a.items() will give me

a) [(a,b),(c,d)]
b) [[a,b],[c,d]]
c) ((a,b),(c,d))
d) None of the above

Answer: a

64. How to get all the keys of a dictionary

a) key()
b) keys()
c) getkeys()
d) None of the above

Answer: b

65. join() is a method of list

a) True
b) False

Answer: b

66. a = Car() . Consider a to be an object of Car(). "print a" calls with method of "a"?

a) __repr__
b) __str__
c) print
d) None of the above

Answer: b

67. a = “5” + 5 results in value of a being

a) 55
b) 10
c) Error
d) None of the above

Answer : a

68. Python is a loosely typed language

a) False
b) True

Answer: a

69. If a = “hello” , then a[10] would

a) Give an empty string
b) Raise an error
c) Strings are treated as sparse arrays. It creates an empty location
d) None of the above

Answer: b

70. if a = {‘a’:’b’, ‘c’:’d’} then a[‘e’] = ‘f’ would result in

a) “a” being {‘a’:’b’, ‘c’:’d’, ‘e’:’f’}
b) IndexError as “e” is not a part of the dictionary
c) None of the above
d) Cannot assign a value to dictionary like this

Answer: a

71. a =[1,2,3,4] b =[3,4,5] then a.append(b)

a) will result in “a” whose -1 index is the list [3,4,5]
b) 3,4,5 will get appended as three items in indices 4,5,6
c) as 3,5 already exists in “a” only 5 will be added
d) append is not a function of the list

Answer: a

72. a = ‘hello’ then a[-1:1:-1] will give

a) oll
b)‘’

c)o
d)ello

Answer: a

73. The following helps in opening a file inpython

a) open(<filepath>,<mode>)
b) openfile(<filepath>,<mode>)
c) filehandle(<filepath>,<mode>)
d) None of the above

Answer: a

74. In a set the items can repeat

a) True
b) False

Answer: b

75. When to use “yield”

a) While writing a generator
b) There is no such thing as yield
c) yield is a custom function
d) None of the above

Answer: a

76. If a =[1,2,3] then a.insert(-1,5) will result in a being

a) [1,2,5,3]
b) [1,2,3,5]
c) Error
d) None of the above

Answer: a

77. Python uses the values in “ “ environment variable to figure out the directories to search for module imports

a) PYTHON_PATH
b) PYTHONPATH
c) PATH
d) MODULEPATH

Answer: b

78. If a is dictionary then a.items() will give me

a) A set of keys as keys are unique
b) A list of tuples where each tuple has two items , one key and the other value
c) A list of values
d) None of the above

Answer: b

79. What type of data is: a=[(1,1),(2,4),(3,9)]?

a) Array of tuples
b) List of tuples
c) Tuples of lists
d) Invalid type

Answer: b
Explanation:
The variable a has tuples enclosed in a list making it a list of tuples.

80. Which of the statements about dictionary values if false?

a) More than one key can have the same value
b) The values of the dictionary can be accessed as dict[key].
c) Values of a dictionary must be unique
d) Values of a dictionary can be a mixture of letters and numbers

Answer: c
Explanation:
More than one key can have the same value.

81. What is the output of the following snippet of code?

```
a = {}
a[1] = 1
a['1'] = 2
print len(a)
```

a) An exception is thrown
b) 3
c) 6
d) 2

Answer: d

Explanation:

A key of integer 1 is different from a key of string '1'

82. _____ is a string literal denoted by triple quotes for providing the documentation of certain program elements.

a) Interface
b) Modularity
c) Client
d) Docstring

Answer: d

Explanation:

Docstring used for providing the specifications of program elements.

83. Which of the following statement is true

a) Python searches in all the subfolders of directories present in the sys.path
b) Python seraches in all the subfolders of directories present in the sys.path only if they contain __init__.py file
c) Python only searches in the directories present in sys.path
d) None of the above

Answer: b

84. The function of re.search is:

a) Matches a pattern at the start of the string
b) Matches a pattern at the end of the string
c) Matches a pattern from any part of a string
d) Such a function does not exist

Answer: c

Explanation:

The re module of Python consists of a function re.search. It's function is to match a pattern from anywhere in a string.

85. Which of the following mode will refer to binary data?

a) r
b) w
c) +
d) b

Answer: d

Explanation:

Mode Meaning is as explained below:

86. What is the correct syntax of open() function?

a) file = open(file_name [, access_mode][, buffering])
b) file object = open(file_name [, access_mode][, buffering])
c) file object = open(file_name)
d) none of the mentioned

Answer: b

Explanation:

Open() function correct syntax with the parameter details as shown below:

87. Correct syntax of readlines() provided fh is the file handle?

a) fh.readlines(<filepath>);
b) fh.readlines();
c) fh.readlines(sequence)
d) none of the mentioned

Answer: b

Explanation:

fh.readlines() will return a list of strings where each string is a line in the file

88. Which function overloads the + operator?

a) __add__()
b) __plus__()
c) __sum__()
d) none of the mentioned

Answer: a

89. Which function overloads the == operator?

a) __eq__()
b) __equ__()
c) __isequal__()
d) none of the mentioned

Answer: a
Explanation:
The other two do not exist.

90. Which piece of code creates an empty class?

a)
```
class A(object):
   return
```
b)
```
class A(object):
   pass
```
c)
```
class A(object):
```
d) It is not possible to create an empty class.

Answer: b

91. Which of the following is not an exception handling keyword in Python?

a) try
b) except
c) accept
d) finally

Answer: c

Explanation:

The keywords ‘try’, ‘except’ and ‘finally’ are exception handling keywords in python whereas the word ‘accept’ is not a keyword at all.

92. Choose the right answer after looking at the below piece of code

```
def add(x,y):
      return x + y
```

a) add is a function which accepts integers
b) add can take any arguments which implement __add__
c) add only works with integers
d) add prints the sum of two numbers

Answer: b

Explanation:

everything is classes and objects in Python. ONly if they implement __add__ a + operator can be applied. Else it would raise an error

93. Choose the right answer

```
a = 10
def f():
      print a
      a = 20
f()
```

a) Works fine
b) raises an error
c) will change the value of a to 20
d) None of the above

Answer: b
Explanation:
The function f() is trying to access a as both global and local. Hence the error

94. Choose the right answer

```
def f(a, b = 3, c=1):
      pass
```

a) f(1) works
b) f(b=3) works

c) wrong function implementation
d) None of the above

Answer: a
Explanation:

b fails because you would not be passing a non default argument

95. Choose the right answer

```
def f(*args):

      return sum(args)


print f(1,2,3)
```

a) 6
b) error
c) No output
d) None of the above

Answer: a

96. Choose the right answer

```
l = [1,2,3,4]
a = l.pop()
print a
```

a) 4
b) error
c) no output
d) None of the above

Answer: a

97. Choose the right output

```
print list(zip([1,2,3],[4,5,6]))
```

a) [(1,2,3), (4,5,6)]
b) [(1,4),(2,5),(3,6)]
c) error
d) None of the above

Answer: b

Explanation:

Zip will take in two or more sequences of same length and put them into a list of tuples which contains one item from each sequence

98. Choose the right output

```
l=[1,2,3]
print list(enumerate(l))
```

a) [(0,1),(1,2),(2,3)]
b) [(0,1,2),(1,2,3)]
c) Error
d) None of the above

Answer: a

Explanation:

Enumerate will also provide the index of the item along with the item

99. How can we access the command line arguments while executing a python script

a) Create an explicit function to receive the arguments
b) are available to us through sys.argv
c) We cannot access command line arguments
d) None of the above

Answer: b

100. Choose the right answer

a) the directories mentioned in PYTHONPATH are added before python's standard libraries in sys.path
b) the directories mentioned in PYTHONPATH are added after python's standard libraries in sys.path
c) Python processes PYTHONPATH and sys.path separately
d) None of the above

Answer: a

101. Choose the right answer

a) .pyc files are always created
b) .pyc files are created only when a python file is imported as a module
c) .pyc files are recreated if .py file's timestamp is more recent as compared to .pyc file
d) both b and c

Answer: d