1. What is a variable in Python?

  Variable is a name which is used to store a value.

2. How do you create a variable?

  variable_name = value

3. How do you check the value within a variable?

  print(variable_name)

4. How do you create multiple variables in a single statement?

  variable1 , variable2 , ... , variable_n = value1,value2 , ... , value_n

5. How do you create multiple variables with the same value?

  variable1 = variable2 = ... = variable_n = value

6. How do you change the value of a variable?

  variable_name = old_value

  variable_name = new_value

7. How do you reassign a variable by modifying the previous value?

  variable_name = new_value

8. What does the statement `counter += 4` do?

  counter = counter + 4

9. What are the rules for naming a variable?

  can contain any letter , underscore(_) , any digit(0-9).

  Should not start with a digit and must not be a reserved keyword.

10. Are variable names case-sensitive? Do `a_variable`, `A_Variable`, and `A_VARIABLE` represent the same variable or different ones?

  Yes , variable names are case-sensitive. The given variable names represent different variables.

11. What is Syntax? Why is it important?

  Syntax of a language determines the rules that are defined to write code in that language.

  It helps the compiler/interpreter to compile the code easily.

12. What happens if you execute a statement with invalid syntax?

  The compiler/interpreter throws a syntax error by terminating the compilation of the file.

13. How do you check the data type of a variable?

  print(type(variable_name))

14. What are the built-in data types in Python?

  Integers , float , string , character , arrays , bool , list , tuples , sets , dictionary.

15. What is a primitive data type?

  Primitive data types are the pre-defined data types that have fixed memory size and are used to create and store different types of values in variables.

16. What are the primitive data types available in Python?

  Integers , float , string , character , arrays , list , tuples , sets , dictionary.

17. What is a data structure or container data type?

    A data structure is a composite data type in python in which different types of values can be stored. It also has a pre-defined methods/functions associated with it to perform certain basic tasks.

18. What are the container types available in Python?

    lists , tuples , sets , dictionary.

19. What kind of data does the Integer data type represent?

    Any real-valued integer number.

20. What are the numerical limits of the integer data type?

    Unlimited

21. What kind of data does the float data type represent?

    Any decimal number.

22. How does Python decide if a given number is a float or an integer?

    If given number is a decimal number , python stores it as a float value , as an integer value otherwise.

23. How can you create a variable which stores a whole number, e.g., 4 but has the float data type?

    X = 4.0

24. How do you create floats representing very large (e.g., 6.023 x 10^23) or very small numbers (0.000000123)?

    Exponential notation. Eg: 6.023E23 , 1.23E-10

25. What does the expression `23e-12` represent?

    23 x 10^-12

26. Can floats be used to store numbers with unlimited precision?

    No

27. What are the differences between integers and floats?\

    Integers only store real-valued integers while floats can store any decimal number as well. Integers only divison return on floor value while floats return exact quotient on division.

28. How do you convert an integer to a float?

    x = 4
    x = float(x)

29. How do you convert a float to an integer?

    x = 4.3
    x = int(x)

30. What is the result obtained when you convert 1.99 to an integer?

    1

31. What are the data types of the results of the division operators `/` and `//`?

    / - returns float value and // - returns integer value.

32. What kind of data does the Boolean data type represent?
    True , False

33. Which types of Python operators return booleans as a result?
    logical operators(is,or,and) and comparison operators(>=,==,<=)

34. What happens if you try to use a boolean in arithmetic operation?
    The boolean value gets converted into corresponding integer value and gets evaluated.

35. How can any value in Python be covered to a boolean?
    variable_name = bool(variable_name)

36. What are truthy and falsy values?
    Non boolean values used in a boolean context.

37. What are the values in Python that evaluate to False?
    0 , None , ''

38. Give some examples of values that evaluate to True.
    "a random string" , -25 , 4.3.

39. What kind of data does the None data type represent?
    NoneType

40. What is the purpose of None?
    To check if a variable has a value or not.

41. What kind of data does the String data type represent?
    Numbers , symbols and any character.

42. What are the different ways of creating strings in Python?
    z = ' some string'
    z = """ for multi line strings"""
    z = str(10)

43. What is the difference between strings creating using single quotes, i.e. `'` and `'` vs. those created using double quotes, i.e. `\"` and `\"`?   **No difference**

44. How do you create multi-line strings in Python?
     using """ """

45. What is the newline character, `\\n`?
    All the characters next to it in the string will appear in the new line.

46. What are escaped characters? How are they useful?
    Characters indicated with a backslash are known as escape characters. They are used to include quotes and more than one line in a string.

47. How do you check the length of a string?
    z = 'some random string'
    print(len(z))

48. How do you convert a string into a list of characters?

```
z = 'some string'
z = list(z)
```

49. How do you access a specific character from a string?

Using index of the character starting from 0

50. How do you access a range of characters from a string?

```
using slicing
z = 'this is a string' , i = 3 , j = 9
print(z[i:j])
```

51. How do you check if a specific character occurs in a string?

```
z = 'check for char'
print('f' in z)
```

52. How do you check if a smaller string occurs within a bigger string?

```
z = 'small' , x = 'small and big char'
print(z in x)
```

53. How do you join two or more strings?

```
z = '1st string' , y = '2nd string'
z = z + ' ' + y
```

54. What are \"methods\" in Python? How are they different from functions?

Methods are the functions which belong to a class whereas functions need not be part of a class.

55. What do the `.lower`, `.upper` and `.capitalize` methods on strings do?

.lower converts all characters of a string into lowercase.
.upper converts all characters of a string into uppercase.
.capitalize converts the starting character of each word of a string into uppercase.

56. How do you replace a specific part of a string with something else?

```
z = 'change this string'
z.replace('change this','changed')
```

57. How do you split the string \"Sun,Mon,Tue,Wed,Thu,Fri,Sat\" into a list of days?

```
s = 'Sun,Mon,Tue,Wed,Thu,Fri,Sat'
s.split(',')
```

58. How do you remove whitespace from the beginning and end of a string?

```
s ='this is a string'
s.strip()
```

59. What is the string `.format` method used for? Can you give an example?

To add desired characters in particular positions into a string format function is used.

```
Z = '{} is my age'
z.format(20)
```

60. What are the benefits of using the `.format` method instead of string concatenation?
   string concatenation adds new string to the end of the current string whereas format function can add dynamic new string into a specified index of a string.

61. How do you convert a value of another type to a string?
   z = 4.3
   z = str(z)

62. How do you check if two strings have the same value?
   s1 = 'string1' , s2 = 'string2'
   print(s1==s2)

63. Where can you find the list of all the methods supported by strings?
   In String class

64. What is a list in Python?
   A list is a composite data type which can store collection of different types of data like interger , string , boolean and float. It also has different functions to manipulate the data.

65. How do you create a list?
   new_list = []

66. Can a Python list contain values of different data types?   **Yes.**

67. Can a list contain another list as an element within it?   **Yes**

68. Can you create a list without any values?   **Yes.**

69. How do you check the length of a list in Python?
   print(len(list_name))

70. How do you retrieve a value from a list?
   Using index number of the value in the list

71. What is the smallest and largest index you can use to access elements from a list containing five elements?   **0,4**

72. What happens if you try to access an index equal to or larger than the size of a list?
   list index out of range error is thrown.

73. What happens if you try to access a negative index within a list?
   corresponding value is returned starting from the end of the list.

74. How do you access a range of elements from a list?
   using slicing

75. How many elements does the list returned by the expression `a_list[2:5]` contain?   **5**

76. What do the ranges `a_list[:2]` and `a_list[2:]` represent?
   0 to 2 , 2 to len(list)-1

77. How do you change the item stored at a specific index within a list?

    list_name[index] = value

78. How do you insert a new item at the beginning, middle, or end of a list?

    beginning: list_name.insert(0,new_value)

    middle: list_name.insert(len(list_name)/2,value)

    end: list_name.append(value)

79. How do you remove an item from al list?

    list_name.remove(value)

80. How do you remove the item at a given index from a list?

    list_name.pop(index)

81. How do you check if a list contains a value?

    print(value in list_name)

82. How do you combine two or most lists to create a larger list?

    list1 = ['1st' ,'list'] , list2 = ['2nd' ,'list']

    list1.extend(list2)

83. How do you create a copy of a list?

    copy function

84. Does the expression `a_new_list = a_list` create a copy of the list `a_list`?   **No**

85. Where can you find the list of all the methods supported by lists?   **In list class.**

86. What is a Tuple in Python?

    Tuple is an immutable list in python

87. How is a tuple different from a list?

    In tuple new items cannot be added , but in list new elements can be added.

88. Can you add or remove elements in a tuple?   **No**

89. How do you create a tuple with just one element?

    tuple_name = (value,)

90. How do you convert a tuple to a list and vice versa?

    z = (1,'sample',4.3)

    z = list(z)

    z = tuple(z)

91. What are the `count` and `index` method of a Tuple used for?

    count returns the number of times a value is present in the tuple.

    Index returns the index of first occurrence of a value in the tuple.

92. What is a dictionary in Python?

    Dictionary is a collection of key-value pairs in which key are unqiuly used to access the values associated with the keys.

93.How do you create a dictionary?
    dict_name = {}

94.What are keys and values?
    keys are the names which uniquely indentify each value in a dictionary.
    Values are the data that are associated with different keys and can be accessed by their keys.

95.How do you access the value associated with a specific key in a dictionary?
    dict_name[key]

96.What happens if you try to access the value for a key that doesn't exist in a dictionary?
    A key error occurs

97. What is the `.get` method of a dictionary used for?
    .get method returns the value at the specified key.

98.How do you change the value associated with a key in a dictionary?
    dict_name[key] = value

99.How do you add or remove a key-value pair in a dictionary?
    dict_name[key_name] = value
    del dict_name[key_name]

100. How do you access the keys, values, and key-value pairs within a dictionary?
    dict_name.keys() , dict_name.values() , dict_name.items()