**MDSC-102 Assignment-1**

- Sumukha Sreenidhi (204219)

1. What is a variable in Python?

A. A variable is a name/container that stores a particular value.

2. How do you create a variable?

A. Syntax is:

<Variable name> = <value>

Type a valid variable name, put an ‘=’ sign and then the value.

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

A. By printing the variable.

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

A. By separating the multiple variables by commas.

For example:

x , y = 5 , 7

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

A. By using ‘=’ sign.

For example:

x = y = 10

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

A. By typing the same variable name, ‘=’ sign and the new value.

For example:

x = 10

x = 8

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

A. By using arithmetic operators.

For example:

x = 5

x += 1

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

A. It will increment the value of counter variable by 4.

9. What are the rules for naming a variable?

A. Rule 1: It must contain only letters (a-z, A-Z), numbers (0-9) and underscore.  
 Rule 2: It must not start with a digit

Rule 3: Variable names are case sensitive

Rule 4: A variable name can’t be one of the 35 reserved keywords of python.

10. Are variable names case-sensitive? Do **a\_variable**, **A\_Variable** and **A\_VARIABLE** represent the same variable or different ones?

A. Yes, they are case-sensitive. The given 3 variables are different from each other.

11. What is syntax? Why is it important?

A. Syntax is pre-defined structured way of typing code for any particular language. Syntax must be followed so that the structure of the code can be maintained and syntax errors can be avoided.

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

A. It may lead to syntax errors.

13. How do you check the data type of a variable?A. By using the built-in function: ‘type’.

For example:

x=10

print(type(x))

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

A. Numbers (integer, float, complex), strings, lists, tuples, dictionaries, sets are the built-in data types in Python.

15. What is a primitive data type?

A. Primitive data-type is a standalone data-type

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

A. Numbers

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

A. Structure in which multiple data types can be stored.

For example: Linked lists.

18. What are the container types available in Python?

A. List

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

A. It represents all the integers, both negative and non-negative.

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

A. No limit.

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

A. All the floating-point numbers, i.e., decimal numbers.

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

A. If a fraction ( / symbol) or a decimal point is there between digits, the variable is of float type. Otherwise it is int.

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

A. 1st way: Float declaration

x = 4.0

2nd way: Explicit typecasting

x = 4

x = float(x)

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

A. By typing 6.023e23.

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

A. It represents 23 x 10^-12.

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

A. No. It is dependent on system architecture.

27. What are the differences between integers and floats?

A. If a fraction ( / symbol) or a decimal point is there between digits, the variable is of float type. Otherwise it is int.

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

A. By Explicit typecasting:

x = 4

x = float(x)

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

A. By Explicit typecasting:

x = 4.0

x = int(x)

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

A. 1.

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

A. Result of / : float

Result of // : int

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

A. If the data can be classified into two types, either True or False, then Boolean data types can be used.

33. Which types of Python operators return booleans as a result?

A. bool operator

34. What happens if you try to use a boolean in arithmetic operation?

A. Returns True or False

35. How can any value in Python be covered to a boolean?

A. By using bool operator

36. What are truthy and falsy values?

A. Truthy is True and Falsy is False.

37. What are the values in Python that evaluate to False?

A. 0

38. Give some examples of values that evaluate to True.

A. 1, 2, 2.5, etc

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

A. NoneType is used for representing ‘no result’.

40. What is the purpose of None?

A. None is used for representing ‘no result’.

41. What kind of data does the String data type represent?

A. Any possible character inside single/double quotes.

42. What are the different ways of creating strings in Python?A. Using single/double/triple quotes.

43. What is the difference between strings creating using single quotes, i.e. `'` and `'` vs. those created using double quotes, i.e. `\"` and `\"`?A. Both are same.

44. How do you create multi-line strings in Python?A. Using triple quotes.

45. What is the newline character, `\\n`?A. For printing a new line.

46. What are escaped characters? How are they useful?A. \ is an escaped character. Helps to print quotes inside a string.

47. How do you check the length of a string?A. len(string\_name)

48. How do you convert a string into a list of characters?A. list=[]

for x in string:

list.append(x)

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

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

51. How do you check if a specific character occurs in a string?A. print(<character> in <string>)

52. How do you check if a smaller string occurs within a bigger string?A. print(<sub\_string> in <string>)

53. How do you join two or more strings?A. By using concatenation.

54. What are \"methods\" in Python? How are they different from functions?A. Methods are built-in functions in python. Each data-type have their own set of methods that make our work easier as they are already available and we didn’t code them.

55. What do the `.lower`, `.upper` and `.capitalize` methods on strings do?A. .lower converts all letters to lowercase and prints the string.

.upper converts all letters to uppercase and prints the string.

.capitalize capitalizes the first character of string and prints it.

56. How do you replace a specific part of a string with something else?A. print(string.replace(<old sub\_string>,<new\_sub\_string>)

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

58. How do you remove whitespace from the beginning and end of a string?A. By using .strip() method

59. What is the string `.format` method used for? Can you give an example?A. The method helps us in inserting a new substring readily into already made string and print it.

For example:

string="Welcome back {}!"

print(string.format("Sumukha"))

60. What are the benefits of using the `.format` method instead of string concatenation?A. It saves time when we’re inserting a substring in middle of the main string.

61. How do you convert a value of another type to a string?A. Using explicit typecasting.

62. How do you check if two strings have the same value?A. string1==string2

63. Where can you find the list of all the methods supported by strings?A. help(str)

64. What is a list in Python?A. List is a mutable data-type in python that are represented by []. It can surprisingly hold elements of different data-types.

65. How do you create a list?A. list1=[]

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

67. Can a list contain another list as an element within it?A. Yes. Its called nested list.

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

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

70. How do you retrieve a value from a list?A. Using indexing.

71. What is the smallest and largest index you can use to access elements from a list containing five elements?A. Smallest: 0 or -5. Largest: 4 or -1.

72. What happens if you try to access an index equal to or larger than the size of a list?A. IndexError: list index out of range

73. What happens if you try to access a negative index within a list?A. In a list with n elements, -ve indexing starts from -1 and goes on upto -n.

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

75. How many elements does the list returned by the expression `a\_list[2:5]` contain?A. 3 elements.

76. What do the ranges `a\_list[:2]` and `a\_list[2:]` represent?A. Index 0 to 1. Index 2 to (len-1)

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

For example:

x=[1,2,3]

x[1]=100

print(x)

78. How do you insert a new item at the beginning, middle, or end of a list?A. By mentioning the index value at which we want to insert inside .insert method.

For example:

x=[1,2,3]

x.insert(1,100)

print(x)

79. How do you remove an item from a list?A. By using .remove method and mentioning the element that we want to remove.

For example:

x=[12,2,3]

x.remove(12)

80. How do you remove the item at a given index from a list?A. By using .pop method and mentioning the index that we want to remove.

For example:

x=[12,2,3]

x.pop(0)

81. How do you check if a list contains a value?A. <value> in <list\_name>

82. How do you combine two or most lists to create a larger list?A. Add the variables of the lists, i.e., Concatenate.

83. How do you create a copy of a list?A. By using .pop method and mentioning the index that we want to remove.

For example:

x=[12,2,3]

y=x.copy()

84. Does the expression `a\_new\_list = a\_list` create a copy of the list `a\_list`?A. Yes.

85. Where can you find the list of all the methods supported by lists?A. help(list)

86. What is a Tuple in Python?A. An immutable data-type that are represented by ().

87. How is a tuple different from a list?A. A tuple is immutable whereas a list mutable.

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

89. How do you create a tuple with just one element?A. x=(1)

90. How do you convert a tuple to a list and vice versa?A. By using explicit typecasting.

91. What are the `count` and `index` method of a Tuple used for?A. To return the no. of times an element is repeating and its index.

92. What is a dictionary in Python?A. A mutable data-type that is represented by {}. It will store key-value pairs.

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

94. What are keys and values?A. Keys are user defined indeces for each value. Values are the values associated to each user-defined key.

95. How do you access the value associated with a specific key in a dictionary?A. By using indexing where index is the specific key.

96. What happens if you try to access the value for a key that doesn't exist in a dictionary?A. It will raise a KeyError.

97. What is the `.get` method of a dictionary used for?A. For returning the value of a particular index, i.e., specific key.

98. How do you change the value associated with a key in a dictionary?A. By using indexing.

For example:

x={1:2,2:3}

x[2]=200

99. How do you add or remove a key-value pair in a dictionary?A. Adding:

x={2:100}

x[3]=200

Removing:

x={2:100,3:200}

x.pop(3)

100. How do you access the keys, values, and key-value pairs within a dictionary?"A. print(x)

for y in x.keys():

print(y)

for y in x.values():

print(y)