MDSC-102-Practical Lab  
  
Assignment - I

Roll No: 204221

Name: Vidyasager GR

1. What is a variable in Python?

A: A variable is a name given to the value that is stored in memory.

2. How do you create a variable?

A: A Variable name should be given. So, we write it as variable\_name(it can be any name) = (some value) For ex. var = 10

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

A: We can check the values within a variable two ways. One is to just type the variable name of the variable you created and press enter, it will display the value within. The other way is to use the inbuilt print function. print(variable\_name) will display the value within.

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

A: We specify all the variable names separated by commas and then have the common ‘=’ assignment operator and then followed by the values of the respective variables separated by commas. For ex. x,y=10,20

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

A: It is the same as creating a single variable but we just add the another variable name after the first ‘=’ assignment operator of the first variable. For ex. a=b=10

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

A: Changing the value of a variable is the same as what we had to do to create a variable. For ex.

a=10

a

10

a=20

a

20

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

A: Reassigning a variable is same as above.

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

A: It increments the value of counter by 4 or in other words it adds 4 to the value that is within the variable ‘counter’.

9. What are the rules for naming a variable?

A: Variable names are case-sensitive, so a and A are not the same, Variable names can contain alphabets, underscore and numbers but should not start with a number, Variable names cannot be any of the 35 reserved keywords or the built-in constant names such as True, False and None.

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 and No, ‘a\_variable’, ‘A\_Variable’ and ‘A\_VARIABLE’ don’t represent the same variable.

11. What is Syntax? Why is it important?

A: Syntax are the instructions or the predefined ways of writing code in such a way that the compiler or interpretor recognizes the code written. So it is important to follow syntax of each programming language for the code to function properly.

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

A: Usually a syntax error will be thrown when syntax is not followed.

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

A: The built-in ‘type’ function is used to check the data type of a variable.

14. What are the built-in data types in Python?  
A: Number, Floating Point Number, String, Dictionary, List, Set, Tuple

15. What is a primitive data type?  
A: A datatype that does not contain any datatype as a component.

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

A: Number

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

A: A Data Structure is a container that lets us store a collection of objects of any datatype in specific ways.

18. What are the container types available in Python?

A: List, Tuple, Dictionary, Set

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

A: The Integer data type represents numbers such as 1, 4, -10, etc.

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

A: There are no numerical limits of the integer data type.

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

A: The float data type represents real numbers.

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

A: Python considers any number containing a period (“.”) as a floating-point number.

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

A: x = float(4)

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

A: x = 6.023e23, y = 1.23e-07

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

A: 0.00000000000023

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

A: No. Because the representation of float do not have the exact value but the approximated value.

27. What are the differences between integers and floats?

A: Integer arithmetic is exact while it is not the case for floats.

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

A: You typecast the integer into a float. For ex. float(4) will give you 4.0.

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

A: You typecast the float into an integer. For ex. int(4.0) will give you 4.

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

A: We get 1.

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

A: Float and Integer.

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

A: The bool data such as True and False.

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

A: The Logical Operators such as ==, !=, >, <, >=, <= returns booleans as a result.

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

A: True represents 1 and False represents 0, so arithmetic operation using a boolean will still work. For ex. 1 + True gives you 2 and 1 + False gives you 1.

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

A: The bool() function can be used to typecast any value in python to a boolean. Any non-zero value is converted to True and 0 is converted to False.

36. What are truthy and falsy values?

A: Non-Boolean values when converted to Boolean and if they are True, then they are Truthy values. If not they are Falsy values.

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

A: 0 and None evaluates to False.

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

A: Any Non-Zero value can be evaluated to True. Examples: 10,999,-20.

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

A: NoneType.

40. What is the purpose of None?

A: It is the equivalent of NULL in other programming languages. It means there is no data.

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

A: String data type represents text data.

42. What are the different ways of creating strings in Python?

A: One way is to use escape sequences such as \n to indicate the string to continue in the next line to get the desired foramtting. Another way is to use triple-quoted strings.

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

A: There is no difference.

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

A: Encapsulating the strings in triple quotes to indicate it is a multi-line string.

45. What is the newline character, `\\n`?

A: It continues the string mentioned after \n in the next line.

46. What are escaped characters? How are they useful?

A: Escaped characters will not be printed in the string. It is useful in specifying the format of the string like \n for newline, \t for horizontal tab, etc.

47. How do you check the length of a string?

A: The built-in “len” method returns the length of a string.

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

A: Typecasting.

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

A: Indexing.

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

A: Slicing.

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

A: If we take a string “Knight”, we can check whether the character “K” is present in the string by typing “K” in “Knight” and it will return True if it exists and False if not.

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

A: If we take a string “Knight”, we can check whether the substring “Kni” is present in the string by typing “Kni” in “Knight” and it will return True if it exists and False if not.

53. How do you join two or more strings?

A: String concatenation can be done using the + operator.

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

A: Methods are built-in functions that belong to a class. They can only be called using a reference of an object of the class.

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

A: The .lower method returns the string as lowercase and .upper returns as uppercase. The .capitalize method returns the string with the first letter as uppercase.

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

A: Using the .replace method specifying the index and the replacement letter.

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

A: We can achieve this using the .split method for strings.

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

A: By using the .strip() method we can remove whitespaces from the beginning and end of a string.

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

A: The .format method is used for string formatting where a few things which can be changed in the string and not fixed can be mentioned and the values in those blanks can be sent through the format method as parameters. For ex. ‘My name is {}’.format(‘Vidyasager’).

60. What are the benefits of using the `.format` method instead of string concatenation?

A: String concatenation requires the process of adding different strings together in a particular order difficult while the usage of blanks in the form of {} in the string lets us skip the process of adding strings together.

61. How do you convert a value of another type to a string?

A: The str() function can be used to convert a value of another type to a string.

62. How do you check if two strings have the same value?

A: The logical operator ‘==’ can be used to check if two strings have the same value. If they have the same value a boolean True is returned and False is returned if not.

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

A: The dir() function will list all the methods of the data type of the variable passed into the function. So if you pass str or any string variable inside, you can find the list of all the methods supported by strings.

64. What is a list in Python?

A: A list is a data structure in python which is an ordered, mutable array of objects.

65. How do you create a list?

A: List notation is ‘[]’, so alist = [1,2,3,4] will make a list containing 4 numbers.

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, a list can contain another list as an element within it.

68. Can you create a list without any values?

A: Yes we can create an empty list.

69. How do you check the length of a list in Python?

A: The len() function will work for lists too which returns the length of a list.

70. How do you retrieve a value from a list?

A: List is just an array of objects so we can retrieve a value from it by indexing.

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

A: 0 and 4

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

A: We get an index error thrown at us that says the ‘list index is out of range’.

73. What happens if you try to access a negative index within a list?

A: Negative indexing will get you the elements of a list from the last element. For ex. list1 = [1,2,3] contains 3 elements. We can access 3 by using list1[2] or list1[-1].

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

A: The indexing can be done within a range by mentioning two indices separated by a colon. For ex. list1[0:2] will return 1 and 2 where it will take the values till index 2 but not the value in index 2.

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: a\_list[:2] represents the range of elements from the index 0 till 2 but not including 2. a\_list[2:] represents the range of elements from the index 2 till the last index of the list.

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

A: Using indexing and assignment operator we can change the item stored at a specific index.

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

A: We can use the insert method for lists to specify the beginning index and middle index to insert items at those locations. To insert an item at the end of the list we can either use the insert method specifying the last index or use the append method which adds an item at the end of the list.

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

A: Using the remove method or the pop method.

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

A: Using the pop method where you specify the index as a parameter.

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

A: Using the membership operator “in”.

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

A: By using the extend method.

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

A: By using the copy method or the assignment operator.

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: dir(list)

86. What is a Tuple in Python?

A: Tuple is also a data structure in python like lists but unlike lists, tuples are immutable.

87. How is a tuple different from a list?

A: Tuples are immutable

88. Can you add or remove elements in a tuple?

A: No.

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

A: Add a comma at the end of that single element.

90. How do you convert a tuple to a list and vice versa?  
A: Using the list() constructor and tuple() constructor.

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

A: Same as it is used in Lists. Count method will tell you how many times a particular element is occuring in the tuple and index method will return the position of a particular element in the tuple.

92. What is a dictionary in Python?

A: A dictionary is an associative array where each value is indexed by an unique key. It is a collection of key-value pairs.

93. How do you create a dictionary?

A: The key and values are separated by colons and each pair is separated by commas and all of these are enclosed inside braces.

94. What are keys and values?

A: Keys are the uniquely identifying elements which are associated with each value in the dictionary. Values are objects that are associated with a unique key.

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

A: It is similar to indexing but instead of the index, the unique key is mentioned to get the value assiociated with it.

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

A: A Key Error will be thrown.

97. What is the `.get` method of a dictionary used for?

A: It is used to retrieve a value in the dictionary given a key if it exists.

98. How do you change the value associated with a key in a dictionary?

A: Using assignment operator. Index the dictionary using the key and using assignment operator, reassign the value associated to that key.

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

A: Using the .pop() method or the .popitem() method.

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

A: Using the .keys() method, .values() method and .items() method respectively.

---------------------------------