1. This is the first index in a string.

a. −1 b. 1 c. 0 d. The size of the string minus one

2. This is the last index in a string.

a. 1 b. 99 c. 0 d. The size of the string minus one

3. This will happen if you try to use an index that is out of range for a string.

a. A ValueError exception will occur. b. An IndexError exception will occur. c. The string will be erased and the program will continue to run. d. Nothing—the invalid index will be ignored.

4. This function returns the length of a string.

a. length b. size c. len d. lengthof

5. This string method returns a copy of the string with all leading whitespace characters

removed. a. lstrip b. rstrip c. remove d. strip_leading

6. This string method returns the lowest index in the string where a specified substring is found.

a. first_index_of b. locate c. find d. index_of

7. This operator determines whether one string is contained inside another string.

a. contains b. is_in c. == d. in

8. This string method returns true if a string contains only alphabetic characters and is at least one character in length.

a. the isalpha method b. the alpha method c. the alphabetic method d. the isletters method

9. This string method returns true if a string contains only numeric digits and is at least one character in length.

a. the digit method b. the isdigit method c. the numeric method d. the isnumber method

10. This string method returns a copy of the string with all leading and trailing whitespace characters removed.

a. clean b. strip c. remove_whitespace d. rstrip

True or False:

1. Once a string is created, it cannot be changed.

True

2. You can use the for loop to iterate over the individual characters in a string.

True

3. The isupper method converts a string to all uppercase characters.

False

4. The repetition operator (*) works with strings as well as with lists.

True

5. When you call a string’s split method, the method divides the string into two substrings.

False

Short Answer:

1. What does the following code display?

mystr = ‘abc’ mystr2 = ‘123’ mystr += mystr2 print(mystr)

Answer: The code will display: abc123

The += operator concatenates mystr2 to mystr, resulting in the string ‘abc123’.

2. What does the following code display?

mystr = ‘abc’ * 3 print(mystr)

Answer: The code will display: abcabcabc

The * operator repeats the string ‘abc’ three times.

3. What will the following code display?

mystr = ‘abracadabra’ print(mystr[6:9])

Answer: The code will display: ada

This slices the string from index 6 (inclusive) to index 9 (exclusive).

4. What does the following code display?

numbers = [1, 2, 3, 4, 5, 6, 7] print(numbers[4:6])

Answer: The code will display: [5, 6]

This slices the list from index 4 (inclusive) to index 6 (exclusive).

5. What does the following code display?

name = ‘joe’ print(name.lower()) print(name.upper()) print(name)

Answer: The code will display: joe JOE joe

The lower() method returns the string in lowercase, upper() returns it in uppercase, but these methods don’t modify the original string, so the final print shows the original string.