

application or language. It has been adopted by all modern software providers and now allows data to be transported through many different platforms, devices and applications without corruption.

Unicode can be implemented by different character encodings. The Unicode Standard defines Unicode Transformation Formats like UTF-8, UTF-16, and UTF-32, and several other encodings are in use. The most commonly used encodings are UTF-8, UTF-16 and UCS-2 (Universal Coded Character Set), a precursor of UTF-16. UTF-8 is dominantly used by websites (over 90%), uses one byte for the first 128 code points and up to 4 bytes for other characters.

Regular Python strings are *not* Unicode: they are just plain bytes. To create a Unicode string, use the 'u' prefix on the string literal. For example,

```
1. >>> unicode_string = u'A unicode \u018e string \xf1'  
2. >>> unicode_string  
'A unicode \u018e string \xf1'
```

①-② A Unicode string is a different type of object from regular "str" string type.

5.7 Summary

- A string is a sequence of characters.
- To access values through slicing, square brackets are used along with the index.
- Various string operations include conversion, comparing strings, padding, finding a substring in an existing string and replace a string in Python.
- Python strings are immutable which means that once created they cannot be changed.

Multiple Choice Questions

1. The arithmetic operator that cannot be used with strings is

- a. +
- b. *
- c. -
- d. All of these

2. Judge the output of the following code,

```
print(r"\nWelcome")
```

- a. New line and welcome
- b. \nWelcome
- c. The letter r and then welcome
- d. Error

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

```
print("Sunday".find("day"))
```

- a. 6
- b. 5
- c. 3
- d. 1

4. The output of the following code is,

```
print("apple is a fruit".split("is"))
```

- a. ['is a fruit']
- b. [fruit]
- c. ['apple', 'a fruit']
- d. ['apple']

5. For the given string s = "nostradamus", which of the following statement is used to retrieve the character t?

- a. s[3]
- b. sgetitem(3)
- c. s.__getitem__(3)
- d. s.getItem(3)

6. The output of the following:

```
print("\tapple".lstrip())
```

- a. \tapple
- b. apple"
- c. apple
- d. "\tapple

7. Deduce the output of the following code:

```
print('hello' 'newline')
```

- a. Hello
- b. hellonewline
- c. Error
- d. Newline

8. What is the output of the following code?

```
"tweet"[2:]
```

- a. We
- b. wee
- c. eet
- d. Twee

9. What is the output of the following code?
"apple is a fruit"[7:10]

- a. Apple
- b. s a
- c. Fruit
- d. None of the above

10. Identify the output of the following code:

```
print("My name is %s" % ('Charles Darwin'))
```

- a. My name is Charles Darwin
- b. Charles
- c. %Charles
- d. %

11. The prefix that is used to create a Unicode string is

- a. u
- b. h
- c. o
- d. c

12. The function that is used to find the length of the string is

- a. len(string)
- b. length(string)
- c. len[string]
- d. length[string]

13. What is the output of the following code?

```
string = "Lion is the king of jungle"
```

```
print("%s" %string[4:7])
```

- a. of
- b. king
- c. The
- d. is

14. For the statement given below

```
example = "\t\ntweet\n"
```

The output for the expression example.strip() is

- a. \t\nweet\n
- b. \t\nweet
- c. tweet\n
- d. 'weet'

15. Deduce the output of the following code:

- ```
print('Data Science'.istitle())
```
- a. True
  - b. False
  - c. Error
  - d. None

16. Predict the output of the following code:

- ```
print('200.123'.isnumeric())
```
- a. True
 - b. False
 - c. Error
 - d. None

Review Questions

1. What is the use of the *len()* function? Give one example.
2. With the help of an example, explain how we can create string variables in Python.
3. What is slice operation? Explain with an example.
4. List all the escape characters in Python with examples.
5. Explain *in* operator with an example.
6. Write a short note on the *format* operator.
7. Differentiate between the following.
 - a. *isidentifier()* and *isnumeric()*
 - b. *find()* and *casefold()*
 - c. *split()* and *splitlines()*
8. What would happen if a mischievous user typed in a word when you ask for a number?
9. Write a function called *rotate_word* that takes a string and an integer as parameters, and that function should return a new string containing the letters from the original string "rotated" by the given amount. For example, "cheer" rotated by 7 is "jolly" and "melon" rotated by -10 is "cubed".
10. Given that message is a string, what does *message[:]* indicate?
11. Write a function that takes a string as an argument and displays the letters backward, one per line.
12. Write a Python program to access the last character of the string with the help of *len()* function.
13. Ask the user for a string, and then for a number. Print out that string, that many times. (For example, if the string is Python and the number is 3 you should print out PythonPythonPython.)