**Python MCQ 🡺 Python MCQ - sanfoundry.com**

<https://www.sanfoundry.com/python-questions-answers-variable-names/>

**Variable Names**

1. Is Python case sensitive when dealing with identifiers?

a) yes

b) no

c) machine dependent

d) none of the mentioned

Answer: a

Explanation: Case is always significant.

2. What is the maximum possible length of an identifier?

a) 31 characters

b) 63 characters

c) 79 characters

d) none of the mentioned

Answer: d

Explanation: Identifiers can be of any length.

3. Which of the following is invalid?

a) \_a = 1

b) \_\_a = 1

c) \_\_str\_\_ = 1

d) none of the mentioned

Answer: d

Explanation: All the statements will execute successfully but at the cost of reduced readability.

4. Which of the following is an invalid variable?

a) my\_string\_1

b) 1st\_string

c) foo

d) \_

Answer: b

Explanation: Variable names should not start with a number.

5. Why are local variable names beginning with an underscore discouraged?

a) they are used to indicate a private variables of a class

b) they confuse the interpreter

c) they are used to indicate global variables

d) they slow down execution

Answer: a

Explanation: As Python has no concept of private variables, leading underscores are used to indicate variables that must not be accessed from outside the class.

6. Which of the following is not a keyword?

a) eval

b) assert

c) nonlocal

d) pass

Answer: a

Explanation: eval can be used as a variable.

7. All keywords in Python are in \_\_\_\_\_\_\_\_\_

a) lower case

b) UPPER CASE

c) Capitalized

d) None of the mentioned

Answer: d

Explanation: True, False and None are capitalized while the others are in lower case.

8. Which of the following is true for variable names in Python?

a) unlimited length

b) all private members must have leading and trailing underscores

c) underscore and ampersand are the only two special characters allowed

d) none of the mentioned

Answer: a

Explanation: Variable names can be of any length.

9. Which of the following is an invalid statement?

a) abc = 1,000,000

b) a b c = 1000 2000 3000

c) a,b,c = 1000, 2000, 3000

d) a\_b\_c = 1,000,000

Answer: b

Explanation: Spaces are not allowed in variable names.

10. Which of the following cannot be a variable?

a) \_\_init\_\_

b) in

c) it

d) on

Answer: b

Explanation: in is a keyword.

# Basic Operators

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Basic Operators”.

1. Which is the correct operator for power(xy)?

a) X^y

b) X\*\*y

c) X^^y

d) None of the mentioned

Answer: b

Explanation: In python, power operator is x\*\*y i.e. 2\*\*3=8.

2. Which one of these is floor division?

a) /

b) //

c) %

d) None of the mentioned

Answer: b

Explanation: When both of the operands are integer then python chops out the fraction part and gives you the round off value, to get the accurate answer use floor division. This is floor division. For ex, 5/2 = 2.5 but both of the operands are integer so answer of this expression in python is 2. To get the 2.5 answer, use floor division.

3. What is the order of precedence in python?

i) Parentheses

ii) Exponential

iii) Multiplication

iv) Division

v) Addition

vi) Subtraction

a) i,ii,iii,iv,v,vi

b) ii,i,iii,iv,v,vi

c) ii,i,iv,iii,v,vi

d) i,ii,iii,iv,vi,v

Answer: a

Explanation: For order of precedence, just remember this PEMDAS (similar to BODMAS).

4. What is the answer to this expression, 22 % 3 is?

a) 7

b) 1

c) 0

d) 5

Answer: b

Explanation: Modulus operator gives the remainder. So, 22%3 gives the remainder, that is, 1.

5. Mathematical operations can be performed on a string.

a) True

b) False

Answer: b

Explanation: You can’t perform mathematical operation on string even if the string is in the form: ‘1234…’.

6. Operators with the same precedence are evaluated in which manner?

a) Left to Right

b) Right to Left

c) Can’t say

d) None of the mentioned

Answer: a

Explanation: None.

7. What is the output of this expression, 3\*1\*\*3?

a) 27

b) 9

c) 3

d) 1

Answer: c

Explanation: First this expression will solve 1\*\*3 because exponential has higher precedence than multiplication, so 1\*\*3 = 1 and 3\*1 = 3. Final answer is 3.

8. Which one of the following has the same precedence level?

a) Addition and Subtraction

b) Multiplication, Division and Addition

c) Multiplication, Division, Addition and Subtraction

d) Addition and Multiplication

Answer: a

Explanation: “Addition and Subtraction” are at the same precedence level. Similarly, “Multiplication and Division” are at the same precedence level. However, Multiplication and Division operators are at a higher precedence level than Addition and Subtraction operators.

9. The expression Int(x) implies that the variable x is converted to integer.

a) True

b) False

Answer: a

Explanation: None.

10. Which one of the following has the highest precedence in the expression?

a) Exponential

b) Addition

c) Multiplication

d) Parentheses

Answer: d

Explanation: Just remember: PEMDAS, that is, Parenthesis, Exponentiation, Division, Multiplication, Addition, Subtraction. Note that the precedence order of Division and Multiplication is the same. Likewise, the order of Addition and Subtraction is also the same.

# Core Data types

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Core Data Types”.

1. Which of these in not a core data type?

a) Lists

b) Dictionary

c) Tuples

d) Class

Answer: d

Explanation: Class is a user defined data type.

2. Given a function that does not return any value, What value is thrown by default when executed in shell.

a) int

b) bool

c) void

d) None

Answer: d

Explanation: Python shell throws a NoneType object back.

3. What will be the output of the following Python code?

>>>str="hello"

>>>str[:2]

>>>

a) he

b) lo

c) olleh

d) hello

Answer: a

Explanation: We are printing only the 1st two bytes of string and hence the answer is “he”.

4. Which of the following will run without errors?

a) round(45.8)

b) round(6352.898,2,5)

c) round()

d) round(7463.123,2,1)

Answer: a

Explanation: Execute help(round) in the shell to get details of the parameters that are passed into the round function.

5. What is the return type of function id?

a) int

b) float

c) bool

d) dict

Answer: a

Explanation: Execute help(id) to find out details in python shell.id returns a integer value that is unique.

6. In python we do not specify types, it is directly interpreted by the compiler, so consider the following operation to be performed.

>>>x = 13 ? 2

objective is to make sure x has a integer value, select all that apply (python 3.xx)

a) x = 13 // 2

b) x = int(13 / 2)

c) x = 13 % 2

d) All of the mentioned

Answer: d

Explanation: // is integer operation in python 3.0 and int(..) is a type cast operator.

7. What error occurs when you execute the following Python code snippet?

apple = mango

a) SyntaxError

b) NameError

c) ValueError

d) TypeError

Answer: b

Explanation: Mango is not defined hence name error.

8. What will be the output of the following Python code snippet?

def example(a):

a = a + '2'

a = a\*2

return a

>>>example("hello")

a) indentation Error

b) cannot perform mathematical operation on strings

c) hello2

d) hello2hello2

Answer: a

Explanation: Python codes have to be indented properly.

9. What data type is the object below?

L = [1, 23, 'hello', 1]

a) list

b) dictionary

c) array

d) tuple

Answer: a

Explanation: List data type can store any values within it.

10. In order to store values in terms of key and value we use what core data type.

a) list

b) tuple

c) class

d) dictionary

Answer: d

Explanation: Dictionary stores values in terms of keys and values.

11. Which of the following results in a SyntaxError?

a) ‘”Once upon a time…”, she said.’

b) “He said, ‘Yes!'”

c) ‘3\’

d) ”’That’s okay”’

Answer: c

Explanation: Carefully look at the colons.

12. The following is displayed by a print function call. Select all of the function calls that result in this output.

tom

dick

harry

a)

print('''tom

\ndick

\nharry''')

b) print(”’tomdickharry”’)

c) print(‘tom\ndick\nharry’)

d)

print('tom

dick

harry')

Answer: c

Explanation: The \n adds a new line.

13. What is the average value of the following Python code snippet?

>>>grade1 = 80

>>>grade2 = 90

>>>average = (grade1 + grade2) / 2

a) 85.0

b) 85.1

c) 95.0

d) 95.1

Answer: a

Explanation: Cause a decimal value of 0 to appear as output.

14. Select all options that print.

hello-how-are-you

a) print(‘hello’, ‘how’, ‘are’, ‘you’)

b) print(‘hello’, ‘how’, ‘are’, ‘you’ + ‘-‘ \* 4)

c) print(‘hello-‘ + ‘how-are-you’)

d) print(‘hello’ + ‘-‘ + ‘how’ + ‘-‘ + ‘are’ + ‘you’)

Answer: c

Explanation: Execute in the shell.

15. What is the return value of trunc()?

a) int

b) bool

c) float

d) None

Answer: a

Explanation: Execute help(math.trunc) to get details.

# Numeric Types

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Numeric Types”.

1. What is the output of print 0.1 + 0.2 == 0.3?

a) True

b) False

c) Machine dependent

d) Error

Answer: b

Explanation: Neither of 0.1, 0.2 and 0.3 can be represented accurately in binary. The round off errors from 0.1 and 0.2 accumulate and hence there is a difference of 5.5511e-17 between (0.1 + 0.2) and 0.3.

2. Which of the following is not a complex number?

a) k = 2 + 3j

b) k = complex(2, 3)

c) k = 2 + 3l

d) k = 2 + 3J

Answer: c

Explanation: l (or L) stands for long.

3. What is the type of inf?

a) Boolean

b) Integer

c) Float

d) Complex

Answer: c

Explanation: Infinity is a special case of floating point numbers. It can be obtained by float(‘inf’).

4. What does ~4 evaluate to?

a) -5

b) -4

c) -3

d) +3

Answer: a

Explanation: ~x is equivalent to -(x+1).

5. What does ~~~~~~5 evaluate to?

a) +5

b) -11

c) +11

d) -5

Answer: a

Explanation: ~x is equivalent to -(x+1).

6. Which of the following is incorrect?

a) x = 0b101

b) x = 0x4f5

c) x = 19023

d) x = 03964

Answer: d

Explanation: Numbers starting with a 0 are octal numbers but 9 isn’t allowed in octal numbers.

7. What is the result of cmp(3, 1)?

a) 1

b) 0

c) True

d) False

Answer: a

Explanation: cmp(x, y) returns 1 if x > y, 0 if x == y and -1 if x < y.

8. Which of the following is incorrect?

a) float(‘inf’)

b) float(‘nan’)

c) float(’56’+’78’)

d) float(’12+34′)

Answer: d

Explanation: ‘+’ cannot be converted to a float.

9. What is the result of round(0.5) – round(-0.5)?

a) 1.0

b) 2.0

c) 0.0

d) Value depends on Python version

Answer: d

Explanation: The behavior of the round() function is different in Python 2 and Python 3. In Python 2, it rounds off numbers away from 0 when the number to be rounded off is exactly halfway through. round(0.5) is 1 and round(-0.5) is -1 whereas in Python 3, it rounds off numbers towards nearest even number when the number to be rounded off is exactly halfway through. See the below output.

Here’s the runtime output for Python version 2.7 interpreter.

$ python

Python 2.7.17 (default, Nov 7 2019, 10:07:09)

>>> round(0.5)

1.0

>>> round(-0.5)

-1.0

>>>

In the above output, you can see that the round() functions on 0.5 and -0.5 are moving away from 0 and hence “round(0.5) – (round(-0.5)) = 1 – (-1) = 2”

Here’s the runtime output for Python version 3.6 interpreter.

$ python3

Python 3.6.8 (default, Oct 7 2019, 12:59:55)

>>> round(0.5)

0

>>> round(-0.5)

0

>>> round(2.5)

2

>>> round(3.5)

4

>>>

In the above output, you can see that the round() functions on 0.5 and -0.5 are moving towards 0 and hence “round(0.5) – (round(-0.5)) = 0 – 0 = 0“. Also note that the round(2.5) is 2 (which is an even number) whereas round(3.5) is 4 (which is an even number).

10. What does 3 ^ 4 evaluate to?

a) 81

b) 12

c) 0.75

d) 7

Answer: d

Explanation: ^ is the Binary XOR operator.

# Precedence and Associativity – 1

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Precedence and Associativity – 1”.

1. The value of the expressions 4/(3\*(2-1)) and 4/3\*(2-1) is the same.

a) True

b) False

Answer: a

Explanation: Although the presence of parenthesis does affect the order of precedence, in the case shown above, it is not making a difference. The result of both of these expressions is 1.333333333. Hence the statement is true.

2. What will be the value of the following Python expression?

4 + 3 % 5

a) 4

b) 7

c) 2

d) 0

Answer: b

Explanation: The order of precedence is: %, +. Hence the expression above, on simplification results in 4 + 3 = 7. Hence the result is 7.

3. Evaluate the expression given below if A = 16 and B = 15.

A % B // A

a) 0.0

b) 0

c) 1.0

d) 1

Answer: b

Explanation: The above expression is evaluated as: 16%15//16, which is equal to 1//16, which results in 0.

4. Which of the following operators has its associativity from right to left?

a) +

b) //

c) %

d) \*\*

Answer: d

Explanation: All of the operators shown above have associativity from left to right, except exponentiation operator (\*\*) which has its associativity from right to left.

5. What will be the value of x in the following Python expression?

x = int(43.55+2/2)

a) 43

b) 44

c) 22

d) 23

Answer: b

Explanation: The expression shown above is an example of explicit conversion. It is evaluated as int(43.55+1) = int(44.55) = 44. Hence the result of this expression is 44.

6. What is the value of the following expression?

2+4.00, 2\*\*4.0

a) (6.0, 16.0)

b) (6.00, 16.00)

c) (6, 16)

d) (6.00, 16.0)

Answer: a

Explanation: The result of the expression shown above is (6.0, 16.0). This is because the result is automatically rounded off to one decimal place.

7. Which of the following is the truncation division operator?

a) /

b) %

c) //

d) |

Answer: c

Explanation: // is the operator for truncation division. It is called so because it returns only the integer part of the quotient, truncating the decimal part. For example: 20//3 = 6.

8. What are the values of the following Python expressions?

2\*\*(3\*\*2)

(2\*\*3)\*\*2

2\*\*3\*\*2

a) 64, 512, 64

b) 64, 64, 64

c) 512, 512, 512

d) 512, 64, 512

Answer: d

Explanation: Expression 1 is evaluated as: 2\*\*9, which is equal to 512. Expression 2 is evaluated as 8\*\*2, which is equal to 64. The last expression is evaluated as 2\*\*(3\*\*2). This is because the associativity of \*\* operator is from right to left. Hence the result of the third expression is 512.

9. What is the value of the following expression?

8/4/2, 8/(4/2)

a) (1.0, 4.0)

b) (1.0, 1.0)

c) (4.0. 1.0)

d) (4.0, 4.0)

Answer: a

Explanation: The above expressions are evaluated as: 2/2, 8/2, which is equal to (1.0, 4.0).

10. What is the value of the following expression?

float(22//3+3/3)

a) 8

b) 8.0

c) 8.3

d) 8.33

Answer: b

Explanation: The expression shown above is evaluated as: float( 7+1) = float(8) = 8.0. Hence the result of this expression is 8.0.

# Precedence and Associativity – 2

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Precedence and Associativity – 2”.

1. What will be the output of the following Python expression?

print(4.00/(2.0+2.0))

a) Error

b) 1.0

c) 1.00

d) 1

Answer: b

Explanation: The result of the expression shown above is 1.0 because print rounds off digits.

2. What will be the value of X in the following Python expression?

X = 2+9\*((3\*12)-8)/10

a) 30.0

b) 30.8

c) 28.4

d) 27.2

Answer: d

Explanation: The expression shown above is evaluated as: 2+9\*(36-8)/10, which simplifies to give 2+9\*(2.8), which is equal to 2+25.2 = 27.2. Hence the result of this expression is 27.2.

3. Which of the following expressions involves coercion when evaluated in Python?

a) 4.7 – 1.5

b) 7.9 \* 6.3

c) 1.7 % 2

d) 3.4 + 4.6

Answer: c

Explanation: Coercion is the implicit (automatic) conversion of operands to a common type. Coercion is automatically performed on mixed-type expressions. The expression 1.7 % 2 is evaluated as 1.7 % 2.0 (that is, automatic conversion of int to float).

4. What will be the output of the following Python expression?

24//6%3, 24//4//2

a) (1,3)

b) (0,3)

c) (1,0)

d) (3,1)

Answer: a

Explanation: The expressions are evaluated as: 4%3 and 6//2 respectively. This results in the answer (1,3). This is because the associativity of both of the expressions shown above is left to right.

5. Which among the following list of operators has the highest precedence?

+, -, \*\*, %, /, <<, >>, |

a) <<, >>

b) \*\*

c) |

d) %

Answer: b

Explanation: The highest precedence is that of the exponentiation operator, that is of \*\*.

6. What will be the value of the following Python expression?

float(4+int(2.39)%2)

a) 5.0

b) 5

c) 4.0

d) 4

Answer: c

Explanation: The above expression is an example of explicit conversion. It is evaluated as: float(4+int(2.39)%2) = float(4+2%2) = float(4+0) = 4.0. Hence the result of this expression is 4.0.

7. Which of the following expressions is an example of type conversion?

a) 4.0 + float(3)

b) 5.3 + 6.3

c) 5.0 + 3

d) 3 + 7

Answer: a

Explanation: Type conversion is nothing but explicit conversion of operands to a specific type. Options 5.3 + 6.3 and 5.0 + 3 are examples of implicit conversion whereas option 4.0 + float(3) is an example of explicit conversion or type conversion.

8. Which of the following expressions results in an error?

a) float(‘10’)

b) int(‘10’)

c) float(’10.8’)

d) int(’10.8’)

Answer: d

Explanation: All of the above examples show explicit conversion. However the expression int(’10.8’) results in an error.

9. What will be the value of the following Python expression?

4+2\*\*5//10

a) 3

b) 7

c) 77

d) 0

Answer: b

Explanation: The order of precedence is: \*\*, //, +. The expression 4+2\*\*5//10 is evaluated as 4+32//10, which is equal to 4+3 = 7. Hence the result of the expression shown above is 7.

10. The expression 2\*\*2\*\*3 is evaluates as: (2\*\*2)\*\*3.

a) True

b) False

Answer: b

Explanation: The value of the expression (2\*\*2)\*\*3 = 4\*\*3 = 64. When the expression 2\*\*2\*\*3 is evaluated in python, we get the result as 256, because this expression is evaluated as 2\*\*(2\*\*3). This is because the associativity of exponentiation operator (\*\*) is from right to left and not from left to right.

# Bitwise – 1

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Bitwise – 1”.

1. What will be the output of the following Python code snippet if x=1?

x<<2

a) 8

b) 1

c) 2

d) 4

Answer: d

Explanation: The binary form of 1 is 0001. The expression x<<2 implies we are performing bitwise left shift on x. This shift yields the value: 0100, which is the binary form of the number 4.

2. What will be the output of the following Python expression?

bin(29)

a) ‘0b10111’

b) ‘0b11101’

c) ‘0b11111’

d) ‘0b11011’

Answer: b

Explanation: The binary form of the number 29 is 11101. Hence the output of this expression is ‘0b11101’.

3. What will be the value of x in the following Python expression, if the result of that expression is 2?

x>>2

a) 8

b) 4

c) 2

d) 1

Answer: a

Explanation: When the value of x is equal to 8 (1000), then x>>2 (bitwise right shift) yields the value 0010, which is equal to 2. Hence the value of x is 8.

4. What will be the output of the following Python expression?

int(1011)?

a) 1011

b) 11

c) 13

d) 1101

Answer: a

Explanation: The result of the expression shown will be 1011. This is because we have not specified the base in this expression. Hence it automatically takes the base as 10.

5. To find the decimal value of 1111, that is 15, we can use the function:

a) int(1111,10)

b) int(‘1111’,10)

c) int(1111,2)

d) int(‘1111’,2)

Answer: d

Explanation: The expression int(‘1111’,2) gives the result 15. The expression int(‘1111’, 10) will give the result 1111.

6. What will be the output of the following Python expression if x=15 and y=12?

x & y

a) b1101

b) 0b1101

c) 12

d) 1101

Answer: c

Explanation: The symbol ‘&’ represents bitwise AND. This gives 1 if both the bits are equal to 1, else it gives 0. The binary form of 15 is 1111 and that of 12 is 1100. Hence on performing the bitwise AND operation, we get 1100, which is equal to 12.

7. Which of the following expressions results in an error?

a) int(1011)

b) int(‘1011’,23)

c) int(1011,2)

d) int(‘1011’)

Answer: c

Explanation: The expression int(1011,2) results in an error. Had we written this expression as int(‘1011’,2), then there would not be an error.

8. Which of the following represents the bitwise XOR operator?

a) &

b) ^

c) |

d) !

Answer: b

Explanation: The ^ operator represent bitwise XOR operation. &: bitwise AND, | : bitwise OR and ! represents bitwise NOT.

9. What is the value of the following Python expression?

bin(0x8)

a) ‘0bx1000’

b) 8

c) 1000

d) ‘0b1000’

Answer: d

Explanation: The prefix 0x specifies that the value is hexadecimal in nature. When we convert this hexadecimal value to binary form, we get the result as: ‘0b1000’.

10. What will be the output of the following Python expression?

0x35 | 0x75

a) 115

b) 116

c) 117

d) 118

Answer: c

Explanation: The binary value of 0x35 is 110101 and that of 0x75 is 1110101. On OR-ing these two values we get the output as: 1110101, which is equal to 117. Hence the result of the above expression is 117.

# Bitwise – 2

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Bitwise – 2”.

1. It is not possible for the two’s complement value to be equal to the original value in any case.

a) True

b) False

Answer: b

Explanation: In most cases the value of two’s complement is different from the original value. However, there are cases in which the two’s complement value may be equal to the original value. For example, the two’s complement of 10000000 is also equal to 10000000. Hence the statement is false.

2. The one’s complement of 110010101 is:

a) 001101010

b) 110010101

c) 001101011

d) 110010100

Answer: a

Explanation: The one’s complement of a value is obtained by simply changing all the 1’s to 0’s and all the 0’s to 1’s. Hence the one’s complement of 110010101 is 001101010.

3. Bitwise \_\_\_\_\_\_\_\_\_ gives 1 if either of the bits is 1 and 0 when both of the bits are 1.

a) OR

b) AND

c) XOR

d) NOT

Answer: c

Explanation: Bitwise XOR gives 1 if either of the bits is 1 and 0 when both of the bits are 1.

4. What will be the output of the following Python expression?

4^12

a) 2

b) 4

c) 8

d) 12

Answer: c

Explanation: ^ is the XOR operator. The binary form of 4 is 0100 and that of 12 is 1100. Therefore, 0100^1100 is 1000, which is equal to 8.

5. Any odd number on being AND-ed with \_\_\_\_\_\_\_\_ always gives 1. Hint: Any even number on being AND-ed with this value always gives 0.

a) 10

b) 2

c) 1

d) 0

Answer: c

Explanation: Any odd number on being AND-ed with 1 always gives 1. Any even number on being AND-ed with this value always gives 0.

6. What will be the value of the following Python expression?

bin(10-2)+bin(12^4)

a) 0b10000

b) 0b10001000

c) 0b1000b1000

d) 0b10000b1000

Answer: d

Explanation: The output of bin(10-2) = 0b1000 and that of bin(12^4) is ob1000. Hence the output of the above expression is: 0b10000b1000.

7. Which of the following expressions can be used to multiply a given number ‘a’ by 4?

a) a<<2

b) a<<4

c) a>>2

d) a>>4

Answer: a

Explanation: Let us consider an example wherein a=2. The binary form of 2 is 0010. When we left shift this value by 2, we get 1000, the value of which is 8. Hence if we want to multiply a given number ‘a’ by 4, we can use the expression: a<<2.

8. What will be the output of the following Python code if a=10 and b =20?

a=10

b=20

a=a^b

b=a^b

a=a^b

print(a,b)

a) 10 20

b) 10 10

c) 20 10

d) 20 20

Answer: c

Explanation: The code shown above is used to swap the contents of two memory locations using bitwise X0R operator. Hence the output of the code shown above is: 20 10.

9. What is the two’s complement of -44?

a) 1011011

b) 11010100

c) 11101011

d) 10110011

Answer: b

Explanation: The binary form of -44 is 00101100. The one’s complement of this value is 11010011. On adding one to this we get: 11010100 (two’s complement).

10. What will be the output of the following Python expression?

~100?

a) 101

b) -101

c) 100

d) -100

Answer: b

Explanation: Suppose we have an expression ~A. This is evaluated as: -A – 1. Therefore, the expression ~100 is evaluated as -100 – 1, which is equal to -101.

Boolean

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Boolean”.

1. What will be the output of the following Python code snippet?

bool(‘False’)

bool()

a)

True

True

b)

False

True

c)

False

False

d)

True

False

Answer: d

Explanation: The Boolean function returns true if the argument passed to the bool function does not amount to zero. In the first example, the string ‘False’ is passed to the function bool. This does not amount to zero and hence the output is true. In the second function, an empty list is passed to the function bool. Hence the output is false.

2. What will be the output of the following Python code snippet?

['hello', 'morning'][bool('')]

a) error

b) no output

c) hello

d) morning

Answer: c

Explanation: The line of code shown above can be simplified to state that ‘hello’ should be printed if the argument passed to the Boolean function amounts to zero, else ‘morning’ will be printed.

3. What will be the output of the following Python code snippet?

not(3>4)

not(1&1)

a)

True

True

b)

True

False

c)

False

True

d)

False

False

Answer: b

Explanation: The function not returns true if the argument amounts to false, and false if the argument amounts to true. Hence the first function returns false, and the second function returns false.

4. What will be the output of the following Python code?

['f', 't'][bool('spam')]

a) t

b) f

c) No output

d) Error

Answer: a

Explanation: The line of code can be translated to state that ‘f’ is printed if the argument passed to the Boolean function amount to zero. Else ‘t’ is printed. The argument given to the Boolean function in the above case is ‘spam’, which does not amount to zero. Hence the output is t.

5. What will be the output of the following Python code?

l=[1, 0, 2, 0, 'hello', '', []]

list(filter(bool, l))

a) Error

b) [1, 0, 2, 0, ‘hello’, ”, []]

c) [1, 0, 2, ‘hello’, ”, []]

d) [1, 2, ‘hello’]

Answer: d

Explanation: The code shown above returns a new list containing only those elements of the list l which do not amount to zero. Hence the output is: [1, 2, ‘hello’].

6. What will be the output of the following Python code if the system date is 21st June, 2017 (Wednesday)?

[] or {}

{} or []

a)

[]

{}

b)

[]

[]

c)

{}

[]

d)

{}

{}

Answer: c

Explanation: The code shown above shows two functions. In both the cases the right operand is returned. This is because each function is evaluated from left to right. Since the left operand is false, it is assumed that the right operand must be true and hence the right operand is returned in each of the above case.

7. What will be the output of the following Python code?

class Truth:

pass

x=Truth()

bool(x)

a) pass

b) true

c) false

d) error

Answer: b

Explanation: If the truth method is not defined, the object is considered true. Hence the output of the code shown above is true.

8. What will be the output of the following Python code?

if (9 < 0) and (0 < -9):

print("hello")

elif (9 > 0) or False:

print("good")

else:

print("bad")

a) error

b) hello

c) good

d) bad

Answer: c

Explanation: The code shown above prints the appropriate option depending on the conditions given. The condition which matches is (9>0), and hence the output is: good.

9. Which of the following Boolean expressions is not logically equivalent to the other three?

a) not(-6<0 or-6>10)

b) -6>=0 and -6<=10

c) not(-6<10 or-6==10)

d) not(-6>10 or-6==10)

Answer: d

Explanation: The expression not(-6<0 or -6>10) returns the output False.

The expression -6>=0 and -6<=10 returns the output False.

The expression not(-6<10 or -6==10) returns the output False.

The expression not(-6>10 or -6==10) returns the output True.

10. What will be the output of the following Python code snippet?

not(10<20) and not(10>30)

a) True

b) False

c) Error

d) No output

Answer: b

Explanation: The expression not(10<20) returns false. The expression not(10>30) returns true. The and operation between false and true returns false. Hence the output is false.

# Formatting – 1

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Formatting – 1”.

1. What will be the output of the following Python code snippet?

X=”hi”

print(“05d”%X)

a) 00000hi

b) 000hi

c) hi000

d) error

Answer: d

Explanation: The code snippet shown above results in an error because the above formatting option works only if ‘X’ is a number. Since in the above case ‘X’ is a string, an error is thrown.

2. What will be the output of the following Python code snippet?

X=”san-foundry”

print(“%56s”,X)

a) 56 blank spaces before san-foundry

b) 56 blank spaces before san and foundry

c) 56 blank spaces after san-foundry

d) no change

Answer: a

Explanation: The formatting option print(“%Ns”,X) helps us add ‘N’ number of spaces before a given string ‘X’. Hence the output for the code snippet shown above will be 56 blank spaces before the string “san-foundry”.

3. What will be the output of the following Python expression if x=456?

print("%-06d"%x)

a) 000456

b) 456000

c) 456

d) error

Answer: c

Explanation: The expression shown above results in the output 456.

4. What will be the output of the following Python expression if X=345?

print(“%06d”%X)

a) 345000

b) 000345

c) 000000345

d) 345000000

Answer: b

Explanation: The above expression returns the output 000345. It adds the required number of zeroes before the given number in order to make the number of digits 6 (as specified in this case).

5. Which of the following formatting options can be used in order to add ‘n’ blank spaces after a given string ‘S’?

a) print(“-ns”%S)

b) print(“-ns”%S)

c) print(“%ns”%S)

d) print(“%-ns”%S)

Answer: d

Explanation: In order to add ‘n’ blank spaces after a given string ‘S’, we use the formatting option:(“%-ns”%S).

6. What will be the output of the following Python expression if X = -122?

print("-%06d"%x)

a) -000122

b) 000122

c) –00122

d) -00122

Answer: c

Explanation: The given number is -122. Here the total number of digits (including the negative sign) should be 6 according to the expression. In addition to this, there is a negative sign in the given expression. Hence the output will be – -00122.

7. What will be the output of the following Python expression if the value of x is 34?

print(“%f”%x)

a) 34.00

b) 34.0000

c) 34.000000

d) 34.00000000

Answer: c

Explanation: The expression shown above normally returns the value with 6 decimal points if it is not specified with any number. Hence the output of this expression will be: 34.000000 (6 decimal points).

8. What will be the output of the following Python expression if x=56.236?

print("%.2f"%x)

a) 56.00

b) 56.24

c) 56.23

d) 0056.236

Answer: b

Explanation: The expression shown above rounds off the given number to the number of decimal places specified. Since the expression given specifies rounding off to two decimal places, the output of this expression will be 56.24. Had the value been x=56.234 (last digit being any number less than 5), the output would have been 56.23.

9. What will be the output of the following Python expression if x=22.19?

print("%5.2f"%x)

a) 22.1900 b) 22.00000

c) 22.19 d) 22.20

Answer: c

Explanation: The output of the expression above will be 22.19. This expression specifies that the total number of digits (including the decimal point) should be 5, rounded off to two decimal places.

10. The expression shown below results in an error.

print("-%5d0",989)

a) True

b) False

Answer: b

Explanation: The expression shown above does not result in an error. The output of this expression is -%5d0 989. Hence this statement is incorrect.

# Formatting – 2

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Formatting – 2”.

1. What will be the output of the following Python code snippet?

'%d %s %g you' %(1, 'hello', 4.0)

a) Error

b) 1 hello you 4.0

c) 1 hello 4 you

d) 1 4 hello you

Answer: c

Explanation: In the snippet of code shown above, three values are inserted into the target string. When we insert more than one value, we should group the values on the right in a tuple. The % formatting expression operator expects either a single item or a tuple of one or more items on its right side.

2. The output of which of the codes shown below will be: “There are 4 blue birds.”?

a) ‘There are %g %d birds.’ %4 %blue

b) ‘There are %d %s birds.’ %(4, blue)

c) ‘There are %s %d birds.’ %[4, blue]

d) ‘There are %d %s birds.’ 4, blue

Answer: b

Explanation: The code ‘There are %d %s birds.’ %(4, blue) results in the output: There are 4 blue birds. When we insert more than one value, we should group the values on the right in a tuple.

3. What will be the output of the python code shown below for various styles of format specifiers?

x=1234

res='integers:...%d...%-6d...%06d' %(x, x, x)

res

a) ‘integers:…1234…1234 …001234’

b) ‘integers…1234…1234…123400’

c) ‘integers:… 1234…1234…001234’

d) ‘integers:…1234…1234…001234’

Answer: a

Explanation: The code shown above prints 1234 for the format specified %d, ‘1234 ’ for the format specifier %-6d (minus ‘-‘ sign signifies left justification), and 001234 for the format specifier %06d. Hence the output of this code is: ‘integers:…1234…1234 …001234’

4. What will be the output of the following Python code snippet?

x=3.3456789

'%f | %e | %g' %(x, x, x)

a) Error

b) ‘3.3456789 | 3.3456789+00 | 3.345678’

c) ‘3.345678 | 3.345678e+0 | 3.345678’

d) ‘3.345679 | 3.345679e+00 | 3.34568’

Answer: d

Explanation: The %f %e and %g format specifiers represent floating point numbers in different ways. %e and %E are the same, except that the exponent is in lowercase. %g chooses the format by number content. Hence the output of this code is: ‘3.345679 | 3.345679e+00 | 3.34568’.

5. What will be the output of the following Python code snippet?

x=3.3456789

'%-6.2f | %05.2f | %+06.1f' %(x, x, x)

a) ‘3.35 | 03.35 | +003.3’

b) ‘3.3456789 | 03.3456789 | +03.3456789’

c) Error

d) ‘3.34 | 03.34 | 03.34+’

Answer: a

Explanation: The code shown above rounds the floating point value to two decimal places. In this code, a variety of addition formatting features such as zero padding, total field width etc. Hence the output of this code is: ‘3.35 | 03.35 | +003.3’.

6. What will be the output of the following Python code snippet?

x=3.3456789

'%s' %x, str(x)

a) Error

b) (‘3.3456789’, ‘3.3456789’)

c) (3.3456789, 3.3456789)

d) (‘3.3456789’, 3.3456789)

Answer: b

Explanation: We can simply convert strings with a %s format expression or the str built-in function. Both of these methods have been shown in this code. Hence the output is: ) (‘3.3456789’, ‘3.3456789’)

7. What will be the output of the following Python code snippet?

'%(qty)d more %(food)s' %{'qty':1, 'food': 'spam'}

a) Error

b) No output

c) ‘1 more foods’

d) ‘1 more spam’

Answer: d

Explanation: String formatting also allows conversion targets on the left to refer to the keys in a dictionary coded on the right and fetch the corresponding values. In the code shown above, (qty) and (food) in the format string on the left refers to keys in the dictionary literal on the right and fetch their assorted values. Hence the output of the code shown above is: 1 more spam.

8. What will be the output of the following Python code snippet?

a='hello'

q=10

vars()

a) {‘a’ : ‘hello’, ‘q’ : 10, ……..plus built-in names set by Python….}

b) {……Built in names set by Python……}

c) {‘a’ : ‘hello’, ‘q’ : 10}

d) Error

Answer: a

Explanation: The built in function vars() returns a dictionary containing all the variables that exist in the place. Hence the output of the code shown above is: {‘a’ : ‘hello’, ‘q’ : 10, ……..plus built-in names set by Python….}

9. What will be the output of the following Python code?

s='{0}, {1}, and {2}'

s.format('hello', 'good', 'morning')

a) ‘hello good and morning’

b) ‘hello, good, morning’

c) ‘hello, good, and morning’

d) Error

Answer: c

Explanation: Within the subject string, curly braces designate substitution targets and arguments to be inserted either by position or keyword. Hence the output of the code shown above:’hello, good,and morning’.

10. What will be the output of the following Python code?

s='%s, %s & %s'

s%('mumbai', 'kolkata', 'delhi')

a) mumbai kolkata & delhi

b) Error

c) No output

d) ‘mumbai, kolkata & delhi’

Answer: d

Explanation: In the code shown above, the format specifier %s is replaced by the designated substitution. Hence the output of the code shown above is: ‘mumbai, kolkata & delhi’.

11. What will be the output of the following Python code?

t = '%(a)s, %(b)s, %(c)s'

t % dict(a='hello', b='world', c='universe')

a) ‘hello, world, universe’

b) ‘hellos, worlds, universes’

c) Error

d) hellos, world, universe

Answer: a

Explanation: Within the subject string, curly braces represent substitution targets and arguments to be inserted. Hence the output of the code shown above:

‘hello, world, universe’.

12. What will be the output of the following Python code?

'{a}, {0}, {abc}'.format(10, a=2.5, abc=[1, 2])

a) Error

b) ‘2.5, 10, [1, 2]’

c) 2.5, 10, 1, 2

d) ’10, 2.5, [1, 2]’

Answer: b

Explanation: Since we have specified that the order of the output be: {a}, {0}, {abc}, hence the value of associated with {a} is printed first followed by that of {0} and {abc}. Hence the output of the code shown above is: ‘2.5, 10, [1, 2]’.

13. What will be the output of the following Python code?

'{0:.2f}'.format(1.234)

a) ‘1’

b) ‘1.234’

c) ‘1.23’

d) ‘1.2’

Answer: c

Explanation: The code shown above displays the string method to round off a given decimal number to two decimal places. Hence the output of the code is: ‘1.23’.

14. What will be the output of the following Python code?

'%x %d' %(255, 255)

a) ‘ff, 255’

b) ‘255, 255’

c) ‘15f, 15f’

d) Error

Answer: a

Explanation: The code shown above converts the given arguments to hexadecimal and decimal values and prints the result. This is done using the format specifiers %x and %d respectively. Hence the output of the code shown above is: ‘ff, 255’.

15. The output of the two codes shown below is the same.

i. '{0:.2f}'.format(1/3.0)

ii. '%.2f'%(1/3.0)

a) True

b) False

Answer: a

Explanation: The two codes shown above represent the same operation but in different formats. The output of both of these functions is: ‘0.33’. Hence the statement is true.

# Advanced Formatting Tools

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Advanced Formatting Tools”.

1. What will be the output of the following Python code?

l=list('HELLO')

'first={0[0]}, third={0[2]}'.format(l)

a) ‘first=H, third=L’

b) ‘first=0, third=2’

c) Error

d) ‘first=0, third=L’

Answer: a

Explanation: In the code shown above, the value for first is substituted by l[0], that is H and the value for third is substituted by l[2], that is L. Hence the output of the code shown above is: ‘first=H, third=L’. The list l= [‘H’, ‘E’, ‘L’, ‘L’, ‘O’].

2. What will be the output of the following Python code?

l=list('HELLO')

p=l[0], l[-1], l[1:3]

'a={0}, b={1}, c={2}'.format(\*p)

a) Error

b) “a=’H’, b=’O’, c=(E, L)”

c) “a=H, b=O, c=[‘E’, ‘L’]”

d) Junk value

Answer: c

Explanation: In the code shown above, the value for a is substituted by l[0], that is ‘H’, the value of b is substituted by l[-1], that is ‘O’ and the value for c is substituted by l[1:3]. Here the use of \*p is to unpack a tuple items into individual function arguments.

3. The formatting method {1:<10} represents the \_\_\_\_\_\_\_\_\_\_\_ positional argument, \_\_\_\_\_\_\_\_\_ justified in a 10 character wide field.

a) first, right

b) second, left

c) first, left

d) second, right

Answer: b

Explanation: The formatting method {1:<10} represents the second positional argument, left justified in a 10 character wide field.

4. What will be the output of the following Python code?

hex(255), int('FF', 16), 0xFF

a) [0xFF, 255, 16, 255]

b) (‘0xff’, 155, 16, 255)

c) Error

d) (‘0xff’, 255, 255)

Answer: d

Explanation: The code shown above converts the value 255 into hexadecimal, that is, 0xff. The value ‘FF’ into integer. Hence the output of the code shown is: (‘0xff’, 255, 255).

5. The output of the two codes shown below is the same.

i. bin((2\*\*16)-1)

ii. '{}'.format(bin((2\*\*16)-1))

a) True

b) False

Answer: a

Explanation: The output of both of the codes shown above is ‘0b1111111111111111’. Hence the statement is true.

6. What will be the output of the following Python code?

'{a}{b}{a}'.format(a='hello', b='world')

a) ‘hello world’

b) ‘hello’ ‘world’ ‘hello’

c) ‘helloworldhello’

d) ‘hello’ ‘hello’ ‘world’

Answer: c

Explanation: The code shown above prints the values substituted for a, b, a, in the same order. This operation is performed using the format function. Hence the output of the code is: ‘helloworldhello’.

7. What will be the output of the following Python code?

D=dict(p='san', q='foundry')

'{p}{q}'.format(\*\*D)

a) Error

b) sanfoundry

c) san foundry

d) {‘san’, ‘foundry’}

Answer: b

Explanation: The code shown above prints the values substituted for p and q in the same order. Note that there is no blank space between p and q. Hence the output is: sanfoundry.

8. What will be the output of the following Python code?

'The {} side {1} {2}'.format('bright', 'of', 'life')

a) Error

b) ‘The bright side of life’

c) ‘The {bright} side {of} {life}’

d) No output

Answer: a

Explanation: The code shown above results in an error. This is because we have switched from automatic field numbering to manual field numbering, that is, from {} to {1}. Hence this code results in an error.

9. What will be the output of the following Python code?

'{0:f}, {1:2f}, {2:05.2f}'.format(1.23456, 1.23456, 1.23456)

a) Error

b) ‘1.234560, 1.22345, 1.23’

c) No output

d) ‘1.234560, 1.234560, 01.23’

Answer: d

Explanation: In the code shown above, various formatting options are displayed using the format option. Hence the output of this code is: ‘1.234560, 1.234560, 01.23’

10. What will be the output of the following Python code?

'%.2f%s' % (1.2345, 99)

a) ‘1.2345’, ‘99’

b) ‘1.2399’

c) ‘1.234599’

d) 1.23, 99

Answer: b

Explanation: In this code, we must notice that since multiple values haven been given, they should be enclosed in a tuple. Since the formatting format is %.2f, the value 1.2345 is reduced to two decimal places. Hence the output of the code shown above: ‘1.2399’.

11. What will be the output of the following Python code?

'%s' %((1.23,),)

a) ‘(1.23,)’

b) 1.23,

c) (,1.23)

d) ‘1.23’

Answer: a

Explanation: The formatting expression accepts either a single substitution value, or a tuple of one or more items. Since single item can be given either by itself or within the tuple, a tuple to be formatted must be provided as a tested tuple. Hence the output of the code is: >>> ‘%s’ %((1.23,),).

12. What will be the output of the following two codes?

i. '{0}'.format(4.56)

ii. '{0}'.format([4.56,])

a) ‘4.56’, ‘4.56,’

b) ‘4.56’, ‘[4.56]’

c) 4.56, [4.56,]

d) 4.56, [4.56,]

Answer: b

Explanation: The code shown above shows the formatting option on the same value, that is 4.56, where in the second case, the value is enclosed in a list. Hence the output of the code shown above is:

‘4.56’, ‘[4.56]’

# Decorators

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Decorators”.

1. What will be the output of the following Python code?

def mk(x):

def mk1():

print("Decorated")

x()

return mk1

def mk2():

print("Ordinary")

p = mk(mk2)

p()

a)

Decorated

Decorated

b)

Ordinary

Ordinary

c)

Ordinary

Decorated

d)

Decorated

Ordinary

Answer: d

Explanation: The code shown above first prints the word “Decorated” and then “ordinary”. Hence the output of this code is:

Decorated

Ordinary.

2. In the following Python code, which function is the decorator?

def mk(x):

def mk1():

print("Decorated")

x()

return mk1

def mk2():

print("Ordinary")

p = mk(mk2)

p()

a) p()

b) mk()

c) mk1()

d) mk2()

Answer: b

Explanation: In the code shown above, the function mk() is the decorator. The function which is getting decorated is mk2(). The return function is given the name p().

3. The \_\_\_\_\_\_ symbol along with the name of the decorator function can be placed above the definition of the function to be decorated works as an alternate way for decorating a function.

a) #

b) $

c) @

d) &

Answer: c

Explanation: The @ symbol along with the name of the decorator function can be placed above the definition of the function to be decorated works as an alternate way for decorating a function.

4. What will be the output of the following Python code?

def ordi():

print("Ordinary")

ordi

ordi()

a)

Address

Ordinary

b)

Error

Address

c)

Ordinary

Ordinary

d)

Ordinary

Address

Answer: a

Explanation: The code shown above returns the address on the function ordi first, after which the word “Ordinary” is printed. Hence the output of this code is:

Address

Ordinary.

5. The two snippets of the following Python codes are equivalent.

CODE 1

@f

def f1():

print(“Hello”)

CODE 2

def f1():

print(“Hello”)

f1 = f(f1)

a) True

b) False

Answer: a

Explanation: The @ symbol can be used as an alternate way to specify a function that needs to be decorated. The output of the codes shown above is the same. Hence they are equivalent. Therefore this statement is true.

6. What will be the output of the following Python function?

def f(p, q):

return p%q

f(0, 2)

f(2, 0)

a)

0

0

b)

Zero Division Error

Zero Division Error

c)

0

Zero Division Error

d)

Zero Division Error

0

Answer: c

Explanation: The output of f(0, 2) is 0, since o%2 is equal to 0. The output of the f(2, 0) is a Zero Division Error. We can make use of decorators in order to avoid this error.

7. What will be the output of the following Python code?

def f(x):

def f1(a, b):

print("hello")

if b==0:

print("NO")

return

return f(a, b)

return f1

@f

def f(a, b):

return a%b

f(4,0)

a)

hello

NO

b)

hello

Zero Division Error

c) NO

d) hello

Answer: a

Explanation: In the code shown above, we have used a decorator in order to avoid the Zero Division Error. Hence the output of this code is:

hello

NO

8. What will be the output of the following Python code?

def f(x):

def f1(\*args, \*\*kwargs):

print("\*"\* 5)

x(\*args, \*\*kwargs)

print("\*"\* 5)

return f1

def a(x):

def f1(\*args, \*\*kwargs):

print("%"\* 5)

x(\*args, \*\*kwargs)

print("%"\* 5)

return f1

@f

@a

def p(m):

print(m)

p("hello")

a)

\*\*\*\*\*

%%%%%

hello

%%%%%

\*\*\*\*\*

b) Error

c) \*\*\*\*\*%%%%%hello%%%%%\*\*\*\*\*

d) hello

Answer: a

Explanation: The code shown above uses multiple decorators. The output of this code is:

\*\*\*\*\*

%%%%%

hello

%%%%%

\*\*\*\*\*

9. The following python code can work with \_\_\_\_ parameters.

def f(x):

def f1(\*args, \*\*kwargs):

print("Sanfoundry")

return x(\*args, \*\*kwargs)

return f1

a) 2

b) 1

c) any number of

d) 0

Answer: c

Explanation: The code shown above shows a general decorator which can work with any number of arguments.

10. What will be the output of the following Python code?

def f(x):

def f1(\*args, \*\*kwargs):

print("\*", 5)

x(\*args, \*\*kwargs)

print("\*", 5)

return f1

@f

def p(m):

p(m)

print("hello")

a)

\*\*\*\*\*

hello

b)

\*\*\*\*\*

\*\*\*\*\*

hello

c) \*\*\*\*\*

d) hello

Answer: d

Explanation: In the code shown above, we have not passed any parameter to the function p. Hence the output of this code is: hello.

11. A function with parameters cannot be decorated.

a) True

b) False

Answer: a

Explanation: Any function, irrespective of whether or not it has parameters can be decorated. Hence the statement is false.

12. Identify the decorator in the snippet of code shown below.

def sf():

pass

sf = mk(sf)

@f

def sf():

return

a) @f

b) f

c) sf()

d) mk

Answer: d

Explanation: In the code shown above, @sf is not a decorator but only a decorator line. The ‘@’ symbol represents the application of a decorator. The decorator here is the function mk.

13. What will be the output of the following Python code?

class A:

@staticmethod

def a(x):

print(x)

A.a(100)

a) Error

b) Warning

c) 100

d) No output

Answer: c

Explanation: The code shown above demonstrates rebinding using a static method. This can be done with or without a decorator. The output of this code will be 100.

14. What will be the output of the following Python code?

def d(f):

def n(\*args):

return '$' + str(f(\*args))

return n

@d

def p(a, t):

return a + a\*t

print(p(100,0))

a) 100

b) $100

c) $0

d) 0

Answer: b

Explanation: In the code shown above, the decorator helps us to prefix the dollar sign along with the value. Since the second argument is zero, the output of the code is: $100.

15. What will be the output of the following Python code?

def c(f):

def inner(\*args, \*\*kargs):

inner.co += 1

return f(\*args, \*\*kargs)

inner.co = 0

return inner

@c

def fnc():

pass

if \_\_name\_\_ == '\_\_main\_\_':

fnc()

fnc()

fnc()

print(fnc.co)

a) 4

b) 3

c) 0

d) 1

Answer: b

Explanation: The code shown above returns the number of times a given function has been called. Hence the output of this code is: 3

# While and For Loops – 1

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “While and For Loops”.

1. What will be the output of the following Python code?

x = ['ab', 'cd']

for i in x:

i.upper()

print(x)

a) [‘ab’, ‘cd’]

b) [‘AB’, ‘CD’]

c) [None, None]

d) none of the mentioned

Answer: a

Explanation: The function upper() does not modify a string in place, it returns a new string which isn’t being stored anywhere.

2. What will be the output of the following Python code?

x = ['ab', 'cd']

for i in x:

x.append(i.upper())

print(x)

a) [‘AB’, ‘CD’]

b) [‘ab’, ‘cd’, ‘AB’, ‘CD’]

c) [‘ab’, ‘cd’]

d) none of the mentioned

Answer: d

Explanation: The loop does not terminate as new elements are being added to the list in each iteration.

3. What will be the output of the following Python code?

i = 1

while True:

if i%3 == 0:

break

print(i)

i + = 1

a) 1 2

b) 1 2 3

c) error

d) none of the mentioned

Answer: c

Explanation: SyntaxError, there shouldn’t be a space between + and = in +=.

4. What will be the output of the following Python code?

i = 1

while True:

if i%0O7 == 0:

break

print(i)

i += 1

a) 1 2 3 4 5 6

b) 1 2 3 4 5 6 7

c) error

d) none of the mentioned

Answer: a

Explanation: Control exits the loop when i becomes 7.

5. What will be the output of the following Python code?

i = 5

while True:

if i%0O11 == 0:

break

print(i)

i += 1

a) 5 6 7 8 9 10

b) 5 6 7 8

c) 5 6

d) error

Answer: b

Explanation: 0O11 is an octal number.

6. What will be the output of the following Python code?

i = 5

while True:

if i%0O9 == 0:

break

print(i)

i += 1

a) 5 6 7 8

b) 5 6 7 8 9

c) 5 6 7 8 9 10 11 12 13 14 15 ….

d) error

Answer: d

Explanation: 9 isn’t allowed in an octal number.

7. What will be the output of the following Python code?

i = 1

while True:

if i%2 == 0:

break

print(i)

i += 2

a) 1

b) 1 2

c) 1 2 3 4 5 6 …

d) 1 3 5 7 9 11 …

Answer: d

Explanation: The loop does not terminate since i is never an even number.

8. What will be the output of the following Python code?

i = 2

while True:

if i%3 == 0:

break

print(i)

i += 2

a) 2 4 6 8 10 …

b) 2 4

c) 2 3

d) error

Answer: b

Explanation: The numbers 2 and 4 are printed. The next value of i is 6 which is divisible by 3 and hence control exits the loop.

9. What will be the output of the following Python code?

i = 1

while False:

if i%2 == 0:

break

print(i)

i += 2

a) 1

b) 1 3 5 7 …

c) 1 2 3 4 …

d) none of the mentioned

Answer: d

Explanation: Control does not enter the loop because of False.

10. What will be the output of the following Python code?

True = False

while True:

print(True)

break

a) True

b) False

c) None

d) none of the mentioned

Answer: d

Explanation: SyntaxError, True is a keyword and it’s value cannot be changed.

# While and For Loops – 2

This set of Advanced Python Questions & Answers focuses on “While and For Loops – 2”.

1. What will be the output of the following Python code?

i = 0

while i < 5:

print(i)

i += 1

if i == 3:

break

else:

print(0)

a) 0 1 2 0

b) 0 1 2

c) error

d) none of the mentioned

Answer: b

Explanation: The else part is not executed if control breaks out of the loop.

2. What will be the output of the following Python code?

i = 0

while i < 3:

print(i)

i += 1

else:

print(0)

a) 0 1 2 3 0

b) 0 1 2 0

c) 0 1 2

d) error

Answer: b

Explanation: The else part is executed when the condition in the while statement is false.

3. What will be the output of the following Python code?

x = "abcdef"

while i in x:

print(i, end=" ")

a) a b c d e f

b) abcdef

c) i i i i i i …

d) error

Answer: d

Explanation: NameError, i is not defined.

4. What will be the output of the following Python code?

x = "abcdef"

i = "i"

while i in x:

print(i, end=" ")

a) no output

b) i i i i i i …

c) a b c d e f

d) abcdef

Answer: a

Explanation: “i” is not in “abcdef”.

5. What will be the output of the following Python code?

x = "abcdef"

i = "a"

while i in x:

print(i, end = " ")

a) no output

b) i i i i i i …

c) a a a a a a …

d) a b c d e f

Answer: c

Explanation: As the value of i or x isn’t changing, the condition will always evaluate to True.

6. What will be the output of the following Python code?

x = "abcdef"

i = "a"

while i in x:

print('i', end = " ")

a) no output

b) i i i i i i …

c) a a a a a a …

d) a b c d e f

Answer: b

Explanation: As the value of i or x isn’t changing, the condition will always evaluate to True.

7. What will be the output of the following Python code?

x = "abcdef"

i = "a"

while i in x:

x = x[:-1]

print(i, end = " ")

a) i i i i i i

b) a a a a a a

c) a a a a a

d) none of the mentioned

Answer: b

Explanation: The string x is being shortened by one character in each iteration.

8. What will be the output of the following Python code?

x = "abcdef"

i = "a"

while i in x[:-1]:

print(i, end = " ")

a) a a a a a

b) a a a a a a

c) a a a a a a …

d) a

Answer: c

Explanation: String x is not being altered and i is in x[:-1].

9. What will be the output of the following Python code?

x = "abcdef"

i = "a"

while i in x:

x = x[1:]

print(i, end = " ")

a) a a a a a a

b) a

c) no output

d) error

Answer: b

Explanation: The string x is being shortened by one character in each iteration.

10. What will be the output of the following Python code?

x = "abcdef"

i = "a"

while i in x[1:]:

print(i, end = " ")

a) a a a a a a

b) a

c) no output

d) error

Answer: c

Explanation: i is not in x[1:].

# While and For Loops – 3

This set of Tough Python Questions & Answers focuses on “While and For Loops”.

1. What will be the output of the following Python code?

x = 'abcd'

for i in x:

print(i)

x.upper()

a) a B C D

b) a b c d

c) A B C D

d) error

Answer: b

Explanation: Changes do not happen in-place, rather a new instance of the string is returned.

2. What will be the output of the following Python code?

x = 'abcd'

for i in x:

print(i.upper())

a) a b c d

b) A B C D

c) a B C D

d) error

Answer: b

Explanation: The instance of the string returned by upper() is being printed.

3. What will be the output of the following Python code?

x = 'abcd'

for i in range(x):

print(i)

a) a b c d

b) 0 1 2 3

c) error

d) none of the mentioned

Answer: c

Explanation: range(str) is not allowed.

4. What will be the output of the following Python code?

x = 'abcd'

for i in range(len(x)):

print(i)

a) a b c d

b) 0 1 2 3

c) error

d) 1 2 3 4

Answer: b

Explanation: i takes values 0, 1, 2 and 3.

5. What will be the output of the following Python code?

x = 'abcd'

for i in range(len(x)):

print(i.upper())

a) a b c d

b) 0 1 2 3

c) error

d) 1 2 3 4

Answer: c

Explanation: Objects of type int have no attribute upper().

6. What will be the output of the following Python code snippet?

x = 'abcd'

for i in range(len(x)):

i.upper()

print (x)

a) a b c d

b) 0 1 2 3

c) error

d) none of the mentioned

Answer: c

Explanation: Objects of type int have no attribute upper().

7. What will be the output of the following Python code snippet?

x = 'abcd'

for i in range(len(x)):

x[i].upper()

print (x)

a) abcd

b) ABCD

c) error

d) none of the mentioned

Answer: a

Explanation: Changes do not happen in-place, rather a new instance of the string is returned.

8. What will be the output of the following Python code snippet?

x = 'abcd'

for i in range(len(x)):

i[x].upper()

print (x)

a) abcd

b) ABCD

c) error

d) none of the mentioned

Answer: c

Explanation: Objects of type int aren’t subscriptable. However, if the statement was x[i], an error would not have been thrown.

9. What will be the output of the following Python code snippet?

x = 'abcd'

for i in range(len(x)):

x = 'a'

print(x)

a) a

b) abcd abcd abcd

c) a a a a

d) none of the mentioned

Answer: c

Explanation: range() is computed only at the time of entering the loop.

10. What will be the output of the following Python code snippet?

x = 'abcd'

for i in range(len(x)):

print(x)

x = 'a'

a) a

b) abcd abcd abcd abcd

c) a a a a

d) none of the mentioned

Answer: d

Explanation: abcd a a a is the output as x is modified only after ‘abcd’ has been printed once.

# While and For Loops – 4

This set of Python Questions and Answers for Freshers focuses on “While and For Loops”.

1. What will be the output of the following Python code?

x = 123

for i in x:

print(i)

a) 1 2 3

b) 123

c) error

d) none of the mentioned

Answer: c

Explanation: Objects of type int are not iterable.

2. What will be the output of the following Python code?

d = {0: 'a', 1: 'b', 2: 'c'}

for i in d:

print(i)

a) 0 1 2

b) a b c

c) 0 a 1 b 2 c

d) none of the mentioned

Answer: a

Explanation: Loops over the keys of the dictionary.

3. What will be the output of the following Python code?

d = {0: 'a', 1: 'b', 2: 'c'}

for x, y in d:

print(x, y)

a) 0 1 2

b) a b c

c) 0 a 1 b 2 c

d) none of the mentioned

Answer: d

Explanation: Error, objects of type int aren’t iterable.

4. What will be the output of the following Python code?

d = {0: 'a', 1: 'b', 2: 'c'}

for x, y in d.items():

print(x, y)

a) 0 1 2

b) a b c

c) 0 a 1 b 2 c

d) none of the mentioned

Answer: c

Explanation: Loops over key, value pairs.

5. What will be the output of the following Python code?

d = {0: 'a', 1: 'b', 2: 'c'}

for x in d.keys():

print(d[x])

a) 0 1 2

b) a b c

c) 0 a 1 b 2 c

d) none of the mentioned

Answer: b

Explanation: Loops over the keys and prints the values.

6. What will be the output of the following Python code?

d = {0: 'a', 1: 'b', 2: 'c'}

for x in d.values():

print(x)

a) 0 1 2

b) a b c

c) 0 a 1 b 2 c

d) none of the mentioned

Answer: b

Explanation: Loops over the values.

7. What will be the output of the following Python code?

d = {0: 'a', 1: 'b', 2: 'c'}

for x in d.values():

print(d[x])

a) 0 1 2

b) a b c

c) 0 a 1 b 2 c

d) none of the mentioned

Answer: d

Explanation: Causes a KeyError.

8. What will be the output of the following Python code?

d = {0, 1, 2}

for x in d.values():

print(x)

a) 0 1 2

b) None None None

c) error

d) none of the mentioned

Answer: c

Explanation: Objects of type set have no attribute values.

9. What will be the output of the following Python code?

d = {0, 1, 2}

for x in d:

print(x)

a) 0 1 2

b) {0, 1, 2} {0, 1, 2} {0, 1, 2}

c) error

d) none of the mentioned

Answer: a

Explanation: Loops over the elements of the set and prints them.

10. What will be the output of the following Python code?

d = {0, 1, 2}

for x in d:

print(d.add(x))

a) 0 1 2

b) 0 1 2 0 1 2 0 1 2 …

c) None None None

d) None of the mentioned

Answer: c

Explanation: Variable x takes the values 0, 1 and 2. set.add() returns None which is printed.

11. What will be the output of the following Python code?

for i in range(0):

print(i)

a) 0

b) no output

c) error

d) none of the mentioned

Answer: b

Explanation: range(0) is empty.

# While and For Loops – 5

This set of Python Questions and Answers for Experienced people focuses on “While and For Loops”.

1. What will be the output of the following Python code?

for i in range(2.0):

print(i)

a) 0.0 1.0

b) 0 1

c) error

d) none of the mentioned

Answer: c

Explanation: Object of type float cannot be interpreted as an integer.

2. What will be the output of the following Python code?

for i in range(int(2.0)):

print(i)

a) 0.0 1.0

b) 0 1

c) error

d) none of the mentioned

Answer: b

Explanation: range(int(2.0)) is the same as range(2).

3. What will be the output of the following Python code?

for i in range(float('inf')):

print (i)

a) 0.0 0.1 0.2 0.3 …

b) 0 1 2 3 …

c) 0.0 1.0 2.0 3.0 …

d) none of the mentioned

Answer: d

Explanation: Error, objects of type float cannot be interpreted as an integer.

4. What will be the output of the following Python code?

for i in range(int(float('inf'))):

print (i)

a) 0.0 0.1 0.2 0.3 …

b) 0 1 2 3 …

c) 0.0 1.0 2.0 3.0 …

d) none of the mentioned

Answer: d

Explanation: OverflowError, cannot convert float infinity to integer.

5. What will be the output of the following Python code snippet?

for i in [1, 2, 3, 4][::-1]:

print (i)

a) 1 2 3 4

b) 4 3 2 1

c) error

d) none of the mentioned

Answer: b

Explanation: [::-1] reverses the list.

6. What will be the output of the following Python code snippet?

for i in ''.join(reversed(list('abcd'))):

print (i)

a) a b c d

b) d c b a

c) error

d) none of the mentioned

Answer: b

Explanation: ‘ ‘.join(reversed(list(‘abcd’))) reverses a string.

7. What will be the output of the following Python code snippet?

for i in 'abcd'[::-1]:

print (i)

a) a b c d

b) d c b a

c) error

d) none of the mentioned

Answer: b

Explanation: [::-1] reverses the string.

8. What will be the output of the following Python code snippet?

for i in '':

print (i)

a) None

b) (nothing is printed)

c) error

d) none of the mentioned

Answer: b

Explanation: The string does not have any character to loop over.

9. What will be the output of the following Python code snippet?

x = 2

for i in range(x):

x += 1

print (x)

a) 0 1 2 3 4 …

b) 0 1

c) 3 4

d) 0 1 2 3

Answer: c

Explanation: Variable x is incremented and printed twice.

10. What will be the output of the following Python code snippet?

x = 2

for i in range(x):

x -= 2

print (x)

a) 0 1 2 3 4 …

b) 0 -2

c) 0

d) error

Answer: b

Explanation: The loop is entered twice.

# While and For Loops – 6

This set of Python Technical Questions & Answers focuses on “While/For Loops”.

1. What will be the output of the following Python code?

for i in range(10):

if i == 5:

break

else:

print(i)

else:

print("Here")

a) 0 1 2 3 4 Here

b) 0 1 2 3 4 5 Here

c) 0 1 2 3 4

d) 1 2 3 4 5

Answer: c

Explanation: The else part is executed if control doesn’t break out of the loop.

2. What will be the output of the following Python code?

for i in range(5):

if i == 5:

break

else:

print(i)

else:

print("Here")

a) 0 1 2 3 4 Here

b) 0 1 2 3 4 5 Here

c) 0 1 2 3 4

d) 1 2 3 4 5

Answer: a

Explanation: The else part is executed if control doesn’t break out of the loop.

3. What will be the output of the following Python code?

x = (i for i in range(3))

for i in x:

print(i)

a) 0 1 2

b) error

c) 0 1 2 0 1 2

d) none of the mentioned

Answer: a

Explanation: The first statement creates a generator object.

4. What will be the output of the following Python code?

x = (i for i in range(3))

for i in x:

print(i)

for i in x:

print(i)

a) 0 1 2

b) error

c) 0 1 2 0 1 2

d) none of the mentioned

Answer: a

Explanation: We can loop over a generator object only once.

5. What will be the output of the following Python code?

string = "my name is x"

for i in string:

print (i, end=", ")

a) m, y, , n, a, m, e, , i, s, , x,

b) m, y, , n, a, m, e, , i, s, , x

c) my, name, is, x,

d) error

Answer: a

Explanation: Variable i takes the value of one character at a time.

6. What will be the output of the following Python code?

string = "my name is x"

for i in string.split():

print (i, end=", ")

a) m, y, , n, a, m, e, , i, s, , x,

b) m, y, , n, a, m, e, , i, s, , x

c) my, name, is, x,

d) error

Answer: c

Explanation: Variable i takes the value of one word at a time.

7. What will be the output of the following Python code snippet?

a = [0, 1, 2, 3]

for a[-1] in a:

print(a[-1])

a) 0 1 2 3

b) 0 1 2 2

c) 3 3 3 3

d) error

Answer: b

Explanation: The value of a[-1] changes in each iteration.

8. What will be the output of the following Python code snippet?

a = [0, 1, 2, 3]

for a[0] in a:

print(a[0])

a) 0 1 2 3

b) 0 1 2 2

c) 3 3 3 3

d) error

Answer: a

Explanation: The value of a[0] changes in each iteration. Since the first value that it takes is itself, there is no visible error in the current example.

9. What will be the output of the following Python code snippet?

a = [0, 1, 2, 3]

i = -2

for i not in a:

print(i)

i += 1

a) -2 -1

b) 0

c) error

d) none of the mentioned

Answer: c

Explanation: SyntaxError, not in isn’t allowed in for loops.

10. What will be the output of the following Python code snippet?

string = "my name is x"

for i in ' '.join(string.split()):

print (i, end=", ")

a) m, y, , n, a, m, e, , i, s, , x,

b) m, y, , n, a, m, e, , i, s, , x

c) my, name, is, x,

d) error

Answer: a

Explanation: Variable i takes the value of one character at a time.

# Strings – 1

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Strings – 1”.

1. What will be the output of the following Python statement?

>>>"a"+"bc"

a) a

b) bc

c) bca

d) abc

Answer: d

Explanation: + operator is concatenation operator.

2. What will be the output of the following Python statement?

>>>"abcd"[2:]

a) a

b) ab

c) cd

d) dc

Answer: c

Explanation: Slice operation is performed on string.

3. The output of executing string.ascii\_letters can also be achieved by:

a) string.ascii\_lowercase\_string.digits

b) string.ascii\_lowercase+string.ascii\_upercase

c) string.letters

d) string.lowercase\_string.upercase

Answer: b

Explanation: Execute in shell and check.

4. What will be the output of the following Python code?

>>> str1 = 'hello'

>>> str2 = ','

>>> str3 = 'world'

>>> str1[-1:]

a) olleh

b) hello

c) h

d) o

Answer: d

Explanation: -1 corresponds to the last index.

5. What arithmetic operators cannot be used with strings?

a) +

b) \*

c) –

d) All of the mentioned

Answer: c

Explanation: + is used to concatenate and \* is used to multiply strings.

6. What will be the output of the following Python code?

>>>print (r"\nhello")

a) a new line and hello

b) \nhello

c) the letter r and then hello

d) error

Answer: b

Explanation: When prefixed with the letter ‘r’ or ‘R’ a string literal becomes a raw string and the escape sequences such as \n are not converted.

7. What will be the output of the following Python statement?

>>>print('new' 'line')

a) Error

b) Output equivalent to print ‘new\nline’

c) newline

d) new line

Answer: c

Explanation: String literal separated by whitespace are allowed. They are concatenated.

8. What will be the output of the following Python statement?

>>> print('x\97\x98')

a) Error

b)

97

98

c) x\97

d) \x97\x98

Answer: c

Explanation: \x is an escape sequence that means the following 2 digits are a hexadecimal number encoding a character.

9. What will be the output of the following Python code?

>>>str1="helloworld"

>>>str1[::-1]

a) dlrowolleh

b) hello

c) world

d) helloworld

Answer: a

Explanation: Execute in shell to verify.

10. print(0xA + 0xB + 0xC):

a) 0xA0xB0xC

b) Error

c) 0x22

d) 33

Answer: d

Explanation: 0xA and 0xB and 0xC are hexadecimal integer literals representing the decimal values 10, 11 and

12 respectively. There sum is 33.

# Strings – 2

This set of Advanced Python Interview Questions & Answers focuses on “Strings”.

1. What will be the output of the following Python code?

class father:

def \_\_init\_\_(self, param):

self.o1 = param

class child(father):

def \_\_init\_\_(self, param):

self.o2 = param

>>>obj = child(22)

>>>print "%d %d" % (obj.o1, obj.o2)

a) None None

b) None 22

c) 22 None

d) Error is generated

Answer: d

Explanation: self.o1 was never created.

2. What will be the output of the following Python code?

class tester:

def \_\_init\_\_(self, id):

self.id = str(id)

id="224"

>>>temp = tester(12)

>>>print(temp.id)

a) 224

b) Error

c) 12

d) None

Answer: c

Explanation: Id in this case will be the attribute of the class.

3. What will be the output of the following Python code?

>>>example = "snow world"

>>>print("%s" % example[4:7])

a) wo

b) world

c) sn

d) rl

Answer: a

Explanation: Execute in the shell and verify.

4. What will be the output of the following Python code?

>>>example = "snow world"

>>>example[3] = 's'

>>>print example

a) snow

b) snow world

c) Error

d) snos world

Answer: c

Explanation: Strings cannot be modified.

5. What will be the output of the following Python code?

>>>max("what are you")

a) error

b) u

c) t

d) y

Answer: d

Explanation: Max returns the character with the highest ascii value.

6. Given a string example=”hello” what is the output of example.count(‘l’)?

a) 2

b) 1

c) None

d) 0

Answer: a

Explanation: l occurs twice in hello.

7. What will be the output of the following Python code?

>>>example = "helle"

>>>example.find("e")

a) Error

b) -1

c) 1

d) 0

Answer: c

Explanation: Returns lowest index.

8. What will be the output of the following Python code?

>>>example = "helle"

>>>example.rfind("e")

a) -1

b) 4

c) 3

d) 1

Answer: b

Explanation: Returns highest index.

9. What will be the output of the following Python code?

>>>example="helloworld"

>>>example[::-1].startswith("d")

a) dlrowolleh

b) True

c) -1

d) None

Answer: b

Explanation: Starts with checks if the given string starts with the parameter that is passed.

10. To concatenate two strings to a third what statements are applicable?

a) s3 = s1 . s2

b) s3 = s1.add(s2)

c) s3 = s1.\_\_add\_\_(s2)

d) s3 = s1 \* s2

Answer: c

Explanation: \_\_add\_\_ is another method that can be used for concatenation.

# Strings – 3

This set of Python Technical Interview Questions & Answers focuses on “Strings”.

1. What will be the output of the following Python statement?

>>>chr(ord('A'))

a) A

b) B

c) a

d) Error

Answer: a

Explanation: Execute in shell to verify.

2. What will be the output of the following Python statement?

>>>print(chr(ord('b')+1))

a) a b) b c) c d) A

Answer: c

Explanation: Execute in the shell to verify.

3. Which of the following statement prints hello\example\test.txt?

a) print(“hello\example\test.txt”)

b) print(“hello\\example\\test.txt”)

c) print(“hello\”example\”test.txt”)

d) print(“hello”\example”\test.txt”)

Answer: b

Explanation: \is used to indicate that the next \ is not an escape sequence.

4. Suppose s is “\t\tWorld\n”, what is s.strip()?

a) \t\tWorld\n

b) \t\tWorld\n

c) \t\tWORLD\n

d) World

Answer: d

Explanation: Execute help(string.strip) to find details.

5. The format function, when applied on a string returns \_\_\_\_\_\_\_\_\_\_\_

a) Error b) int c) bool d) str

Answer: d

Explanation: Format function returns a string.

6. What will be the output of the “hello” +1+2+3?

a) hello123 b) hello c) Error d) hello6

Answer: c

Explanation: Cannot concatenate str and int objects.

7. What will be the output of the following Python code?

>>>print("D", end = ' ')

>>>print("C", end = ' ')

>>>print("B", end = ' ')

>>>print("A", end = ' ')

a) DCBA

b) A, B, C, D

c) D C B A

d) D, C, B, A will be displayed on four lines

Answer: c

Explanation: Execute in the shell.

8. What will be the output of the following Python statement?(python 3.xx)

>>>print(format("Welcome", "10s"), end = '#')

>>>print(format(111, "4d"), end = '#')

>>>print(format(924.656, "3.2f"))

a) Welcome# 111#924.66

b) Welcome#111#924.66

c) Welcome#111#.66

d) Welcome # 111#924.66

Answer: d

Explanation: Execute in the shell to verify.

9. What will be displayed by print(ord(‘b’) – ord(‘a’))?

a) 0

b) 1

c) -1

d) 2

Answer: b

Explanation: ASCII value of b is one more than a. Hence the output of this code is 98-97, which is equal to 1.

10. Say s=”hello” what will be the return value of type(s)?

a) int

b) bool

c) str

d) String

Answer: c

Explanation: str is used to represent strings in python.

# Strings – 4

This set of Python Coding Questions & Answers focuses on “Strings”.

1. What is “Hello”.replace(“l”, “e”)?

a) Heeeo

b) Heelo

c) Heleo

d) None

Answer: a

Explanation: Execute in shell to verify.

2. To retrieve the character at index 3 from string s=”Hello” what command do we execute (multiple answers allowed)?

a) s[]

b) s.getitem(3)

c) s.\_\_getitem\_\_(3)

d) s.getItem(3)

Answer: c

Explanation: \_\_getitem(..) can be used to get character at index specified as parameter.

3. To return the length of string s what command do we execute?

a) s.\_\_len\_\_()

b) len(s)

c) size(s)

d) s.size()

Answer: a

Explanation: Execute in shell to verify.

4. If a class defines the \_\_str\_\_(self) method, for an object obj for the class, you can use which command to invoke the \_\_str\_\_ method.

a) obj.\_\_str\_\_()

b) str(obj)

c) print obj

d) all of the mentioned

Answer: d

Explanation: Execute in shell to verify.

5. To check whether string s1 contains another string s2, use \_\_\_\_\_\_\_\_

a) s1.\_\_contains\_\_(s2)

b) s2 in s1

c) s1.contains(s2)

d) si.in(s2)

Answer: a

Explanation: s2 in s1 works in the same way as calling the special function \_\_contains\_\_ .

6. Suppose i is 5 and j is 4, i + j is same as \_\_\_\_\_\_\_\_

a) i.\_\_add(j)

b) i.\_\_add\_\_(j)

c) i.\_\_Add(j)

d) i.\_\_ADD(j)

Answer: b

Explanation: Execute in shell to verify.

7. What will be the output of the following Python code?

class Count:

def \_\_init\_\_(self, count = 0):

self.\_\_count = count

c1 = Count(2)

c2 = Count(2)

print(id(c1) == id(c2), end = " ")

s1 = "Good"

s2 = "Good"

print(id(s1) == id(s2))

a) True False

b) True True

c) False True

d) False False

Answer: c

Explanation: Execute in the shell objects cannot have same id, however in the case of strings its different.

8. What will be the output of the following Python code?

class Name:

def \_\_init\_\_(self, firstName, mi, lastName):

self.firstName = firstName

self.mi = mi

self.lastName = lastName

firstName = "John"

name = Name(firstName, 'F', "Smith")

firstName = "Peter"

name.lastName = "Pan"

print(name.firstName, name.lastName)

a) Peter Pan

b) John Pan

c) Peter Smith

d) John Smith

Answer: b

Explanation: Execute in the shell to verify.

9. What function do you use to read a string?

a) input(“Enter a string”)

b) eval(input(“Enter a string”))

c) enter(“Enter a string”)

d) eval(enter(“Enter a string”))

Answer: a

Explanation: Execute in shell to verify.

10. Suppose x is 345.3546, what is format(x, “10.3f”) (\_ indicates space).

a) \_\_345.355

b) \_\_\_345.355

c) \_\_\_\_345.355

d) \_\_\_\_\_345.354

Answer: b

Explanation: Execute in the shell to verify.

This set of Basic Python Questions & Answers focuses on “Strings”.

1. What will be the output of the following Python code?

print("abc DEF".capitalize())

a) abc def

b) ABC DEF

c) Abc def

d) Abc Def

Answer: c

Explanation: The first letter of the string is converted to uppercase and the others are converted to lowercase.

2. What will be the output of the following Python code?

print("abc. DEF".capitalize())

a) abc. def

b) ABC. DEF

c) Abc. def

d) Abc. Def

Answer: c

Explanation: The first letter of the string is converted to uppercase and the others are converted to lowercase.

3. What will be the output of the following Python code?

print("abcdef".center())

a) cd

b) abcdef

c) error

d) none of the mentioned

Answer: c

Explanation: The function center() takes at least one parameter.

4. What will be the output of the following Python code?

print("abcdef".center(0))

a) cd

b) abcdef

c) error

d) none of the mentioned

Answer: b

Explanation: The entire string is printed when the argument passed to center() is less than the length of the string.

5. What will be the output of the following Python code?

print('\*', "abcdef".center(7), '\*')

a) \* abcdef \*

b) \* abcdef \*

c) \*abcdef \*

d) \* abcdef\*

Answer: b

Explanation: Padding is done towards the left-hand-side first when the final string is of odd length. Extra spaces are present since we haven’t overridden the value of sep.

6. What will be the output of the following Python code?

print('\*', "abcdef".center(7), '\*', sep='')

a) \* abcdef \*

b) \* abcdef \*

c) \*abcdef \*

d) \* abcdef\*

Answer: d

Explanation: Padding is done towards the left-hand-side first when the final string is of odd length.

7. What will be the output of the following Python code?

print('\*', "abcde".center(6), '\*', sep='')

a) \* abcde \*

b) \* abcde \*

c) \*abcde \*

d) \* abcde\*

Answer: c

Explanation: Padding is done towards the right-hand-side first when the final string is of even length.

8. What will be the output of the following Python code?

print("abcdef".center(7, 1))

a) 1abcdef

b) abcdef1

c) abcdef

d) error

Answer: d

Explanation: TypeError, the fill character must be a character, not an int.

9. What will be the output of the following Python code?

print("abcdef".center(7, '1'))

a) 1abcdef

b) abcdef1

c) abcdef

d) error

Answer: a

Explanation: The character ‘1’ is used for padding instead of a space.

10. What will be the output of the following Python code?

print("abcdef".center(10, '12'))

a) 12abcdef12

b) abcdef1212

c) 1212abcdef

d) error

Answer: d

Explanation: The fill character must be exactly one character long.

This set of Python Quiz focuses on “Strings”.

1. What will be the output of the following Python code?

print("xyyzxyzxzxyy".count('yy'))

a) 2

b) 0

c) error

d) none of the mentioned

Answer: a

Explanation: Counts the number of times the substring ‘yy’ is present in the given string.

2. What will be the output of the following Python code?

print("xyyzxyzxzxyy".count('yy', 1))

a) 2

b) 0

c) 1

d) none of the mentioned

Answer: a

Explanation: Counts the number of times the substring ‘yy’ is present in the given string, starting from position 1.

3. What will be the output of the following Python code?

print("xyyzxyzxzxyy".count('yy', 2))

a) 2

b) 0

c) 1

d) none of the mentioned

Answer: c

Explanation: Counts the number of times the substring ‘yy’ is present in the given string, starting from position 2.

4. What will be the output of the following Python code?

print("xyyzxyzxzxyy".count('xyy', 0, 100))

a) 2

b) 0

c) 1

d) error

Answer: a

Explanation: An error will not occur if the end value is greater than the length of the string itself.

5. What will be the output of the following Python code?

print("xyyzxyzxzxyy".count('xyy', 2, 11))

a) 2

b) 0

c) 1

d) error

Answer: b

Explanation: Counts the number of times the substring ‘xyy’ is present in the given string, starting from position 2 and ending at position 11.

6. What will be the output of the following Python code?

print("xyyzxyzxzxyy".count('xyy', -10, -1))

a) 2

b) 0

c) 1

d) error

Answer: b

Explanation: Counts the number of times the substring ‘xyy’ is present in the given string, starting from position 2 and ending at position 11.

7. What will be the output of the following Python code?

print('abc'.encode())

a) abc

b) ‘abc’

c) b’abc’

d) h’abc’

Answer: c

Explanation: A bytes object is returned by encode.

8. What is the default value of encoding in encode()?

a) ascii

b) qwerty

c) utf-8

d) utf-16

Answer: c

Explanation: The default value of encoding is utf-8.

9. What will be the output of the following Python code?

print("xyyzxyzxzxyy".endswith("xyy"))

a) 1 b) True c) 3 d) 2

Answer: b

Explanation: The function returns True if the given string ends with the specified substring.

10. What will be the output of the following Python code?

print("xyyzxyzxzxyy".endswith("xyy", 0, 2))

a) 0 b) 1 c) True d) False

Answer: d

Explanation: The function returns False if the given string does not end with the specified substring.

This set of Online Python Quiz focuses on “Strings”.

1. What will be the output of the following Python code?

print("ab\tcd\tef".expandtabs())

a) ab  cd  ef

b) abcdef

c) ab\tcd\tef

d) ab cd ef

Answer: a

Explanation: Each \t is converted to 8 blank spaces by default.

2. What will be the output of the following Python code?

print("ab\tcd\tef".expandtabs(4))

a) ab  cd  ef

b) abcdef

c) ab\tcd\tef

d) ab cd ef

Answer: d

Explanation: Each \t is converted to 4 blank spaces.

3. What will be the output of the following Python code?

print("ab\tcd\tef".expandtabs('+'))

a) ab+cd+ef

b) ab++++++++cd++++++++ef

c) ab cd ef

d) none of the mentioned

Answer: d

Explanation: TypeError, an integer should be passed as an argument.

4. What will be the output of the following Python code?

print("abcdef".find("cd") == "cd" in "abcdef")

a) True

b) False

c) Error

d) None of the mentioned

Answer: b

Explanation: The function find() returns the position of the sunstring in the given string whereas the in keyword returns a value of Boolean type.

5. What will be the output of the following Python code?

print("abcdef".find("cd"))

a) True b) 2 c) 3 d) None of the mentioned

Answer: b

Explanation: The first position in the given string at which the substring can be found is returned.

6. What will be the output of the following Python code?

print("ccdcddcd".find("c"))

a) 4 b) 0 c) Error d) True

Answer: b

Explanation: The first position in the given string at which the substring can be found is returned.

7. What will be the output of the following Python code?

print("Hello {0} and {1}".format('foo', 'bin'))

a) Hello foo and bin

b) Hello {0} and {1} foo bin

c) Error

d) Hello 0 and 1

Answer: a

Explanation: The numbers 0 and 1 represent the position at which the strings are present.

8. What will be the output of the following Python code?

print("Hello {1} and {0}".format('bin', 'foo'))

a) Hello foo and bin

b) Hello bin and foo

c) Error

d) None of the mentioned

Answer: a

Explanation: The numbers 0 and 1 represent the position at which the strings are present.

9. What will be the output of the following Python code?

print("Hello {} and {}".format('foo', 'bin'))

a) Hello foo and bin

b) Hello {} and {}

c) Error

d) Hello and

Answer: a

Explanation: It is the same as Hello {0} and {1}.

10. What will be the output of the following Python code?

print("Hello {name1} and {name2}".format('foo', 'bin'))

a) Hello foo and bin

b) Hello {name1} and {name2}

c) Error

d) Hello and

Answer: c

Explanation: The arguments passed to the function format aren’t keyword arguments.

This set of Python Multiple Choice Questions and Answers focuses on “Strings”.

1. What will be the output of the following Python code?

print("Hello {name1} and {name2}".format(name1='foo', name2='bin'))

a) Hello foo and bin

b) Hello {name1} and {name2}

c) Error

d) Hello and

Answer: a

Explanation: The arguments are accessed by their names.

2. What will be the output of the following Python code?

print("Hello {0!r} and {0!s}".format('foo', 'bin'))

a) Hello foo and foo

b) Hello ‘foo’ and foo

c) Hello foo and ‘bin’

d) Error

Answer: b

Explanation: !r causes the characters ‘ or ” to be printed as well.

3. What will be the output of the following Python code?

print("Hello {0} and {1}".format(('foo', 'bin')))

a) Hello foo and bin

b) Hello (‘foo’, ‘bin’) and (‘foo’, ‘bin’)

c) Error

d) None of the mentioned

Answer: c

Explanation: IndexError, the tuple index is out of range.

4. What will be the output of the following Python code?

print("Hello {0[0]} and {0[1]}".format(('foo', 'bin')))

a) Hello foo and bin

b) Hello (‘foo’, ‘bin’) and (‘foo’, ‘bin’)

c) Error

d) None of the mentioned

Answer: a

Explanation: The elements of the tuple are accessed by their indices.

5. What will be the output of the following Python code snippet?

print('The sum of {0} and {1} is {2}'.format(2, 10, 12))

a) The sum of 2 and 10 is 12

b) Error

c) The sum of 0 and 1 is 2

d) None of the mentioned

Answer: a

Explanation: The arguments passed to the function format can be integers also.

6. What will be the output of the following Python code snippet?

print('The sum of {0:b} and {1:x} is {2:o}'.format(2, 10, 12))

a) The sum of 2 and 10 is 12

b) The sum of 10 and a is 14

c) The sum of 10 and a is c

d) Error

Answer: b

Explanation: 2 is converted to binary, 10 to hexadecimal and 12 to octal.

7. What will be the output of the following Python code snippet?

print('{:,}'.format(1112223334))

a) 1,112,223,334

b) 111,222,333,4

c) 1112223334

d) Error

Answer: a

Explanation: A comma is added after every third digit from the right.

8. What will be the output of the following Python code snippet?

print('{:,}'.format('1112223334'))

a) 1,112,223,334

b) 111,222,333,4

c) 1112223334

d) Error

Answer: d

Explanation: An integer is expected.

9. What will be the output of the following Python code snippet?

print('{:$}'.format(1112223334))

a) 1,112,223,334

b) 111,222,333,4

c) 1112223334

d) Error

Answer: d

Explanation: $ is an invalid format code.

10. What will be the output of the following Python code snippet?

print('{:#}'.format(1112223334))

a) 1,112,223,334

b) 111,222,333,4

c) 1112223334

d) Error

Answer: c

Explanation: The number is printed as it is.

This set of Python MCQs focuses on “Strings”.

1. What will be the output of the following Python code?

print('{0:.2}'.format(1/3))

a) 0.333333

b) 0.33

c) 0.333333:.2

d) Error

Answer: b

Explanation: .2 specifies the precision.

2. What will be the output of the following Python code?

print('{0:.2%}'.format(1/3))

a) 0.33

b) 0.33%

c) 33.33%

d) 33%

Answer: c

Explanation: The symbol % is used to represent the result of an expression as a percentage.

3. What will be the output of the following Python code?

print('ab12'.isalnum())

a) True

b) False

c) None

d) Error

Answer: a

Explanation: The string has only letters and digits.

4. What will be the output of the following Python code?

print('ab,12'.isalnum())

a) True

b) False

c) None

d) Error

Answer: b

Explanation: The character , is not a letter or a digit.

5. What will be the output of the following Python code?

print('ab'.isalpha())

a) True

b) False

c) None

d) Error

Answer: a

Explanation: The string has only letters.

6. What will be the output of the following Python code?

print('a B'.isalpha())

a) True

b) False

c) None

d) Error

Answer: b

Explanation: Space is not a letter.

7. What will be the output of the following Python code snippet?

print('0xa'.isdigit())

a) True

b) False

c) None

d) Error

Answer: b

Explanation: Hexadecimal digits aren’t considered as digits (a-f).

8. What will be the output of the following Python code snippet?

print(''.isdigit())

a) True

b) False

c) None

d) Error

Answer: b

Explanation: If there are no characters then False is returned.

9.What will be the output of the following Python code snippet?

print('my\_string'.isidentifier())

a) True

b) False

c) None

d) Error

Answer: a

Explanation: It is a valid identifier.

10. What will be the output of the following Python code snippet?

print('\_\_foo\_\_'.isidentifier())

a) True

b) False

c) None

d) Error

Answer: a

Explanation: It is a valid identifier.

This set of Python Test focuses on “Strings”.

1. What will be the output of the following Python code snippet?

print('for'.isidentifier())

a) True b) False c) None d) Error

Answer: a

Explanation: Even keywords are considered as valid identifiers.

2. What will be the output of the following Python code snippet?

print('abc'.islower())

a) True b) False c) None d) Error

Answer: a

Explanation: There are no uppercase letters.

3. What will be the output of the following Python code snippet?

print('a@ 1,'.islower())

a) True b) False c) None d) Error

Answer: a

Explanation: There are no uppercase letters.

4. What will be the output of the following Python code snippet?

print('11'.isnumeric())

a) True b) False c) None d) Error

Answer: a

Explanation: All the character are numeric.

5. What will be the output of the following Python code snippet?

print('1.1'.isnumeric())

a) True b) False c) None d) Error

Answer: b

Explanation: The character . is not a numeric character.

6. What will be the output of the following Python code snippet?

print('1@ a'.isprintable())

a) True b) False c) None d) Error

Answer: a

Explanation: All those characters are printable.

7. What will be the output of the following Python code snippet?

print(''''''.isspace())

a) True b) False c) None d) Error

Answer: b

Explanation: None.

8. What will be the output of the following Python code snippet?

print('\t'.isspace())

a) True b) False c) None d) Error

Answer: a

Explanation: Tab Spaces are considered as spaces.

9. What will be the output of the following Python code snippet?

print('HelloWorld'.istitle())

a) True b) False c) None d) Error

Answer: b

Explanation: The letter W is uppercased.

10. What will be the output of the following Python code snippet?

print('Hello World'.istitle())

a) True b) False c) None d) Error

Answer: a

Explanation: It is in title form.

This set of Online Python Test focuses on “Strings”.

1. What will be the output of the following Python code?

print('Hello!2@#World'.istitle())

a) True b) False c) None d) error

Answer: a

Explanation: It is in the form of a title.

2. What will be the output of the following Python code?

print('1Rn@'.lower())

a) n

b) 1rn@

c) rn

d) r

Answer: b

Explanation: Uppercase letters are converted to lowercase. The other characters are left unchanged.

3. What will be the output of the following Python code?

print('''

\tfoo'''.lstrip())

a) \tfoo

b) foo

c)   foo

d) none of the mentioned

Answer: b

Explanation: All leading whitespace is removed.

4. What will be the output of the following Python code?

print('xyyzxxyxyy'.lstrip('xyy'))

a) error

b) zxxyxyy

c) z

d) zxxy

Answer: b

Explanation: The leading characters containing xyy are removed.

5. What will be the output of the following Python code?

print('xyxxyyzxxy'.lstrip('xyy'))

a) zxxy

b) xyxxyyzxxy

c) xyxzxxy

d) none of the mentioned

Answer: a

Explanation: All combinations of the characters passed as an argument are removed from the left hand side.

6. What will be the output of the following Python code?

print('cba'.maketrans('abc', '123'))

a) {97: 49, 98: 50, 99: 51}

b) {65: 49, 66: 50, 67: 51}

c) 321

d) 123

Answer: a

Explanation: A translation table is returned by maketrans.

7. What will be the output of the following Python code?

print('a'.maketrans('ABC', '123'))

a) {97: 49, 98: 50, 99: 51}

b) {65: 49, 66: 50, 67: 51}

c) {97: 49}

d) 1

Answer: b

Explanation: maketrans() is a static method so it’s behaviour does not depend on the object from which it is being called.

8. What will be the output of the following Python code?

print('abcdef'.partition('cd'))

a) (‘ab’, ‘ef’)

b) (‘abef’)

c) (‘ab’, ‘cd’, ‘ef’)

d) 2

Answer: c

Explanation: The string is split into three parts by partition.

9. What will be the output of the following Python code?

print('abcdefcdgh'.partition('cd'))

a) (‘ab’, ‘cd’, ‘ef’, ‘cd’, ‘gh’)

b) (‘ab’, ‘cd’, ‘efcdgh’)

c) (‘abcdef’, ‘cd’, ‘gh’)

d) error

Answer: b

Explanation: The string is partitioned at the point where the separator first appears.

10. What will be the output of the following Python code?

print('abcd'.partition('cd'))

a) (‘ab’, ‘cd’, ”)

b) (‘ab’, ‘cd’)

c) error

d) none of the mentioned

Answer: a

Explanation: The last item is a null string.

This set of Python Problems focuses on “Strings”.

1. What will be the output of the following Python code snippet?

print('cd'.partition('cd'))

a) (‘cd’)

b) (”)

c) (‘cd’, ”, ”)

d) (”, ‘cd’, ”)

Answer: d

Explanation: The entire string has been passed as the separator hence the first and the last item of the tuple returned are null strings.

2. What will be the output of the following Python code snippet?

print('abef'.partition('cd'))

a) (‘abef’)

b) (‘abef’, ‘cd’, ”)

c) (‘abef’, ”, ”)

d) error

Answer: c

Explanation: The separator is not present in the string hence the second and the third elements of the tuple are null strings.

3. What will be the output of the following Python code snippet?

print('abcdef12'.replace('cd', '12'))

a) ab12ef12

b) abcdef12

c) ab12efcd

d) none of the mentioned

Answer: a

Explanation: All occurrences of the first substring are replaced by the second substring.

4. What will be the output of the following Python code snippet?

print('abef'.replace('cd', '12'))

a) abef

b) 12

c) error

d) none of the mentioned

Answer: a

Explanation: The first substring is not present in the given string and hence nothing is replaced.

5. What will be the output of the following Python code snippet?

print('abcefd'.replace('cd', '12'))

a) ab1ef2

b) abcefd

c) ab1efd

d) ab12ed2

Answer: b

Explanation: The first substring is not present in the given string and hence nothing is replaced.

6. What will be the output of the following Python code snippet?

print('xyyxyyxyxyxxy'.replace('xy', '12', 0))

a) xyyxyyxyxyxxy

b) 12y12y1212x12

c) 12yxyyxyxyxxy

d) xyyxyyxyxyx12

Answer: a

Explanation: The first 0 occurrences of the given substring are replaced.

7. What will be the output of the following Python code snippet?

print('xyyxyyxyxyxxy'.replace('xy', '12', 100))

a) xyyxyyxyxyxxy

b) 12y12y1212x12

c) none of the mentioned

d) error

Answer: b

Explanation: The first 100 occurrences of the given substring are replaced.

8. What will be the output of the following Python code snippet?

print('abcdefcdghcd'.split('cd'))

a) [‘ab’, ‘ef’, ‘gh’]

b) [‘ab’, ‘ef’, ‘gh’, ”]

c) (‘ab’, ‘ef’, ‘gh’)

d) (‘ab’, ‘ef’, ‘gh’, ”)

Answer: b

Explanation: The given string is split and a list of substrings is returned.

9. What will be the output of the following Python code snippet?

print('abcdefcdghcd'.split('cd', 0))

a) [‘abcdefcdghcd’]

b) ‘abcdefcdghcd’

c) error

d) none of the mentioned

Answer: a

Explanation: The given string is split at 0 occurances of the specified substring.

10. What will be the output of the following Python code snippet?

print('abcdefcdghcd'.split('cd', -1))

a) [‘ab’, ‘ef’, ‘gh’]

b) [‘ab’, ‘ef’, ‘gh’, ”]

c) (‘ab’, ‘ef’, ‘gh’)

d) (‘ab’, ‘ef’, ‘gh’, ”)

Answer: b

Explanation: Calling the function with a negative value for maxsplit is the same as calling it without any maxsplit specified. The string will be split into as many substring s as possible.

This set of Python Question Bank focuses on “Strings”.

1. What will be the output of the following Python code snippet?

print('abcdefcdghcd'.split('cd', 2))

a) [‘ab’, ‘ef’, ‘ghcd’]

b) [‘ab’, ‘efcdghcd’]

c) [‘abcdef’, ‘ghcd’]

d) none of the mentioned

Answer: a

Explanation: The string is split into a maximum of maxsplit+1 substrings.

2. What will be the output of the following Python code snippet?

print('ab\ncd\nef'.splitlines())

a) [‘ab’, ‘cd’, ‘ef’]

b) [‘ab\n’, ‘cd\n’, ‘ef\n’]

c) [‘ab\n’, ‘cd\n’, ‘ef’]

d) [‘ab’, ‘cd’, ‘ef\n’]

Answer: a

Explanation: It is similar to calling split(‘\n’).

3. What will be the output of the following Python code snippet?

print('Ab!2'.swapcase())

a) AB!@

b) ab12

c) aB!2

d) aB1@

Answer: c

Explanation: Lowercase letters are converted to uppercase and vice-versa.

4. What will be the output of the following Python code snippet?

print('ab cd ef'.title())

a) Ab cd ef

b) Ab cd eF

c) Ab Cd Ef

d) None of the mentioned

Answer: c

Explanation: The first letter of every word is capitalized.

5. What will be the output of the following Python code snippet?

print('ab cd-ef'.title())

a) Ab cd-ef

b) Ab Cd-ef

c) Ab Cd-Ef

d) None of the mentioned

Answer: c

Explanation: The first letter of every word is capitalized. Special symbols terminate a word.

6. What will be the output of the following Python code snippet?

print('abcd'.translate('a'.maketrans('abc', 'bcd')))

a) bcde

b) abcd

c) error

d) bcdd

Answer: d

Explanation: The output is bcdd since no translation is provided for d.

7. What will be the output of the following Python code snippet?

print('abcd'.translate({97: 98, 98: 99, 99: 100}))

a) bcde

b) abcd

c) error

d) none of the mentioned

Answer: d

Explanation: The output is bcdd since no translation is provided for d.

8. What will be the output of the following Python code snippet?

print('abcd'.translate({'a': '1', 'b': '2', 'c': '3', 'd': '4'}))

a) abcd

b) 1234

c) error

d) none of the mentioned

Answer: a

Explanation: The function translate expects a dictionary of integers. Use maketrans() instead of doing the above.

9. What will be the output of the following Python code snippet?

print('ab'.zfill(5))

a) 000ab

b) 00ab0

c) 0ab00

d) ab000

Answer: a

Explanation: The string is padded with zeros on the left hand side. It is useful for formatting numbers.

10. What will be the output of the following Python code snippet?

print('+99'.zfill(5))

a) 00+99

b) 00099

c) +0099

d) +++99

Answer: c

Explanation: zeros are filled in between the first sign and the rest of the string.

# Lists – 1

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Lists”.

1. Which of the following commands will create a list?

a) list1 = list()

b) list1 = []

c) list1 = list([1, 2, 3])

d) all of the mentioned

Answer: d

Explanation: Execute in the shell to verify

2. What is the output when we execute list(“hello”)?

a) [‘h’, ‘e’, ‘l’, ‘l’, ‘o’]

b) [‘hello’]

c) [‘llo’]

d) [‘olleh’]

Answer: a

Explanation: Execute in the shell to verify.

3. Suppose listExample is [‘h’,’e’,’l’,’l’,’o’], what is len(listExample)?

a) 5

b) 4

c) None

d) Error

Answer: a

Explanation: Execute in the shell and verify.

4. Suppose list1 is [2445,133,12454,123], what is max(list1)?

a) 2445

b) 133

c) 12454

d) 123

Answer: c

Explanation: Max returns the maximum element in the list.

5. Suppose list1 is [3, 5, 25, 1, 3], what is min(list1)?

a) 3

b) 5

c) 25

d) 1

Answer: d

Explanation: Min returns the minimum element in the list.

6. Suppose list1 is [1, 5, 9], what is sum(list1)?

a) 1

b) 9

c) 15

d) Error

Answer: c

Explanation: Sum returns the sum of all elements in the list.

7. To shuffle the list(say list1) what function do we use?

a) list1.shuffle()

b) shuffle(list1)

c) random.shuffle(list1)

d) random.shuffleList(list1)

Answer: c

Explanation: Execute in the shell to verify.

8. Suppose list1 is [4, 2, 2, 4, 5, 2, 1, 0], Which of the following is correct syntax for slicing operation?

a) print(list1[0])

b) print(list1[:2])

c) print(list1[:-2])

d) all of the mentioned

Answer: d

Explanation: Slicing is allowed in lists just as in the case of strings.

9. Suppose list1 is [2, 33, 222, 14, 25], What is list1[-1]?

a) Error

b) None

c) 25

d) 2

Answer: c

Explanation: -1 corresponds to the last index in the list.

10. Suppose list1 is [2, 33, 222, 14, 25], What is list1[:-1]?

a) [2, 33, 222, 14]

b) Error

c) 25

d) [25, 14, 222, 33, 2]

Answer: a

Explanation: Execute in the shell to verify.

1. What will be the output of the following Python code?

>>>names = ['Amir', 'Bear', 'Charlton', 'Daman']

>>>print(names[-1][-1])

a) A

b) Daman

c) Error

d) n

Answer: d

Explanation: Execute in the shell to verify.

2. What will be the output of the following Python code?

names1 = ['Amir', 'Bear', 'Charlton', 'Daman']

names2 = names1

names3 = names1[:]

names2[0] = 'Alice'

names3[1] = 'Bob'

sum = 0

for ls in (names1, names2, names3):

if ls[0] == 'Alice':

sum += 1

if ls[1] == 'Bob':

sum += 10

print sum

a) 11

b) 12

c) 21

d) 22

Answer: b

Explanation: When assigning names1 to names2, we create a second reference to the same list. Changes to names2 affect names1. When assigning the slice of all elements in names1 to names3, we are creating a full copy of names1 which can be modified independently.

3. Suppose list1 is [1, 3, 2], What is list1 \* 2?

a) [2, 6, 4]

b) [1, 3, 2, 1, 3]

c) [1, 3, 2, 1, 3, 2]

d) [1, 3, 2, 3, 2, 1]

Answer: c

Explanation: Execute in the shell and verify.

4. Suppose list1 = [0.5 \* x for x in range(0, 4)], list1 is:

a) [0, 1, 2, 3]

b) [0, 1, 2, 3, 4]

c) [0.0, 0.5, 1.0, 1.5]

d) [0.0, 0.5, 1.0, 1.5, 2.0]

Answer: c

Explanation: Execute in the shell to verify.

5. What will be the output of the following Python code?

>>>list1 = [11, 2, 23]

>>>list2 = [11, 2, 2]

>>>list1 < list2 is

a) True

b) False

c) Error

d) None

Answer: b

Explanation: Elements are compared one by one.

6. To add a new element to a list we use which command?

a) list1.add(5)

b) list1.append(5)

c) list1.addLast(5)

d) list1.addEnd(5)

Answer: b

Explanation: We use the function append to add an element to the list.

7. To insert 5 to the third position in list1, we use which command?

a) list1.insert(3, 5)

b) list1.insert(2, 5)

c) list1.add(3, 5)

d) list1.append(3, 5)

Answer: b

Explanation: Execute in the shell to verify.

8. To remove string “hello” from list1, we use which command?

a) list1.remove(“hello”)

b) list1.remove(hello)

c) list1.removeAll(“hello”)

d) list1.removeOne(“hello”)

Answer: a

Explanation: Execute in the shell to verify.

9. Suppose list1 is [3, 4, 5, 20, 5], what is list1.index(5)?

a) 0

b) 1

c) 4

d) 2

Answer: d

Explanation: Execute help(list.index) to get details.

10. Suppose list1 is [3, 4, 5, 20, 5, 25, 1, 3], what is list1.count(5)?

a) 0

b) 4

c) 1

d) 2

Answer: d

Explanation: Execute in the shell to verify.

1. Suppose list1 is [3, 4, 5, 20, 5, 25, 1, 3], what is list1 after list1.reverse()?

a) [3, 4, 5, 20, 5, 25, 1, 3]

b) [1, 3, 3, 4, 5, 5, 20, 25]

c) [25, 20, 5, 5, 4, 3, 3, 1]

d) [3, 1, 25, 5, 20, 5, 4, 3]

Answer: d

Explanation: Execute in the shell to verify.

2. Suppose listExample is [3, 4, 5, 20, 5, 25, 1, 3], what is list1 after listExample.extend([34, 5])?

a) [3, 4, 5, 20, 5, 25, 1, 3, 34, 5]

b) [1, 3, 3, 4, 5, 5, 20, 25, 34, 5]

c) [25, 20, 5, 5, 4, 3, 3, 1, 34, 5]

d) [1, 3, 4, 5, 20, 5, 25, 3, 34, 5]

Answer: a

Explanation: Execute in the shell to verify.

3. Suppose listExample is [3, 4, 5, 20, 5, 25, 1, 3], what is list1 after listExample.pop(1)?

a) [3, 4, 5, 20, 5, 25, 1, 3]

b) [1, 3, 3, 4, 5, 5, 20, 25]

c) [3, 5, 20, 5, 25, 1, 3]

d) [1, 3, 4, 5, 20, 5, 25]

Answer: c

Explanation: pop() removes the element at the position specified in the parameter.

4. Suppose listExample is [3, 4, 5, 20, 5, 25, 1, 3], what is list1 after listExample.pop()?

a) [3, 4, 5, 20, 5, 25, 1]

b) [1, 3, 3, 4, 5, 5, 20, 25]

c) [3, 5, 20, 5, 25, 1, 3]

d) [1, 3, 4, 5, 20, 5, 25]

Answer: a

Explanation: pop() by default will remove the last element.

5. What will be the output of the following Python code?

>>>"Welcome to Python".split()

a) [“Welcome”, “to”, “Python”]

b) (“Welcome”, “to”, “Python”)

c) {“Welcome”, “to”, “Python”}

d) “Welcome”, “to”, “Python”

Answer: a

Explanation: split() function returns the elements in a list.

6. What will be the output of the following Python code?

>>>list("a#b#c#d".split('#'))

a) [‘a’, ‘b’, ‘c’, ‘d’]

b) [‘a b c d’]

c) [‘a#b#c#d’]

d) [‘abcd’]

Answer: a

Explanation: Execute in the shell to verify.

7. What will be the output of the following Python code?

myList = [1, 5, 5, 5, 5, 1]

max = myList[0]

indexOfMax = 0

for i in range(1, len(myList)):

if myList[i] > max:

max = myList[i]

indexOfMax = i

>>>print(indexOfMax)

a) 1

b) 2

c) 3

d) 4

Answer: a

Explanation: First time the highest number is encountered is at index 1.

8. What will be the output of the following Python code?

myList = [1, 2, 3, 4, 5, 6]

for i in range(1, 6):

myList[i - 1] = myList[i]

for i in range(0, 6):

print(myList[i], end = " ")

a) 2 3 4 5 6 1

b) 6 1 2 3 4 5

c) 2 3 4 5 6 6

d) 1 1 2 3 4 5

Answer: c

Explanation: Execute in the shell to verify.

9. What will be the output of the following Python code?

>>>list1 = [1, 3]

>>>list2 = list1

>>>list1[0] = 4

>>>print(list2)

a) [1, 3]

b) [4, 3]

c) [1, 4]

d) [1, 3, 4]

Answer: b

Explanation: Lists should be copied by executing [:] operation.

10. What will be the output of the following Python code?

def f(values):

values[0] = 44

v = [1, 2, 3]

f(v)

print(v)

a) [1, 44]

b) [1, 2, 3, 44]

c) [44, 2, 3]

d) [1, 2, 3]

Answer: c

Explanation: Execute in the shell to verify.

1. What will be the output of the following Python code?

def f(i, values = []):

values.append(i)

return values

f(1)

f(2)

v = f(3)

print(v)

a) [1] [2] [3]

b) [1] [1, 2] [1, 2, 3]

c) [1, 2, 3]

d) 1 2 3

Answer: c

Explanation: Execute in the shell to verify

2. What will be the output of the following Python code?

names1 = ['Amir', 'Bala', 'Chales']

if 'amir' in names1:

print(1)

else:

print(2)

a) None

b) 1

c) 2

d) Error

Answer: c

Explanation: Execute in the shell to verify.

3. What will be the output of the following Python code?

names1 = ['Amir', 'Bala', 'Charlie']

names2 = [name.lower() for name in names1]

print(names2[2][0])

a) None

b) a

c) b

d) c

Answer: d

Explanation: List Comprehension are a shorthand for creating new lists.

4. What will be the output of the following Python code?

numbers = [1, 2, 3, 4]

numbers.append([5,6,7,8])

print(len(numbers))

a) 4

b) 5

c) 8

d) 12

Answer: b

Explanation: A list is passed in append so the length is 5.

5. To which of the following the “in” operator can be used to check if an item is in it?

a) Lists

b) Dictionary

c) Set

d) All of the mentioned

Answer: d

Explanation: In can be used in all data structures.

6. What will be the output of the following Python code?

list1 = [1, 2, 3, 4]

list2 = [5, 6, 7, 8]

print(len(list1 + list2))

a) 2

b) 4

c) 5

d) 8

Answer: d

Explanation: + appends all the elements individually into a new list.

7. What will be the output of the following Python code?

def addItem(listParam):

listParam += [1]

mylist = [1, 2, 3, 4]

addItem(mylist)

print(len(mylist))

a) 1

b) 4

c) 5

d) 8

Answer: c

Explanation: + will append the element to the list.

8. What will be the output of the following Python code?

def increment\_items(L, increment):

i = 0

while i < len(L):

L[i] = L[i] + increment

i = i + 1

values = [1, 2, 3]

print(increment\_items(values, 2))

print(values)

a)

None

[3, 4, 5]

b)

None

[1, 2, 3]

c)

[3, 4, 5]

[1, 2, 3]

d)

[3, 4, 5]

None

Answer: a

Explanation: Execute in the shell to verify.

9. What will be the output of the following Python code?

def example(L):

''' (list) -> list

'''

i = 0

result = []

while i < len(L):

result.append(L[i])

i = i + 3

return result

a) Return a list containing every third item from L starting at index 0

b) Return an empty list

c) Return a list containing every third index from L starting at index 0

d) Return a list containing the items from L starting from index 0, omitting every third item

Answer: a

Explanation: Run the code to get a better understanding with many arguments.

10. What will be the output of the following Python code?

veggies = ['carrot', 'broccoli', 'potato', 'asparagus']

veggies.insert(veggies.index('broccoli'), 'celery')

print(veggies)

a) [‘carrot’, ‘celery’, ‘broccoli’, ‘potato’, ‘asparagus’] Correct 1.00

b) [‘carrot’, ‘celery’, ‘potato’, ‘asparagus’]

c) [‘carrot’, ‘broccoli’, ‘celery’, ‘potato’, ‘asparagus’]

d) [‘celery’, ‘carrot’, ‘broccoli’, ‘potato’, ‘asparagus’]

Answer: a

Explanation: Execute in the shell to verify.

1. What will be the output of the following Python code?

>>>m = [[x, x + 1, x + 2] for x in range(0, 3)]

a) [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

b) [[0, 1, 2], [1, 2, 3], [2, 3, 4]]

c) [1, 2, 3, 4, 5, 6, 7, 8, 9]

d) [0, 1, 2, 1, 2, 3, 2, 3, 4]

Answer: b

Explanation: Execute in the shell to verify.

2. How many elements are in m?

m = [[x, y] for x in range(0, 4) for y in range(0, 4)]

a) 8

b) 12

c) 16

d) 32

Answer: c

Explanation: Execute in the shell to verify.

3. What will be the output of the following Python code?

values = [[3, 4, 5, 1], [33, 6, 1, 2]]

v = values[0][0]

for row in range(0, len(values)):

for column in range(0, len(values[row])):

if v < values[row][column]:

v = values[row][column]

print(v)

a) 3

b) 5

c) 6

d) 33

Answer: d

Explanation: Execute in the shell to verify.

4. What will be the output of the following Python code?

values = [[3, 4, 5, 1], [33, 6, 1, 2]]

v = values[0][0]

for lst in values:

for element in lst:

if v > element:

v = element

print(v)

a) 1

b) 3

c) 5

d) 6

Answer: a

Explanation: Execute in the shell to verify.

5. What will be the output of the following Python code?

values = [[3, 4, 5, 1 ], [33, 6, 1, 2]]

for row in values:

row.sort()

for element in row:

print(element, end = " ")

print()

a) The program prints two rows 3 4 5 1 followed by 33 6 1 2

b) The program prints on row 3 4 5 1 33 6 1 2

c) The program prints two rows 3 4 5 1 followed by 33 6 1 2

d) The program prints two rows 1 3 4 5 followed by 1 2 6 33

Answer: d

Explanation: Execute in the shell to verify.

6. What will be the output of the following Python code?

matrix = [[1, 2, 3, 4],

[4, 5, 6, 7],

[8, 9, 10, 11],

[12, 13, 14, 15]]

for i in range(0, 4):

print(matrix[i][1], end = " ")

a) 1 2 3 4

b) 4 5 6 7

c) 1 3 8 12

d) 2 5 9 13

Answer: d

Explanation: Execute in the shell to verify.

7. What will be the output of the following Python code?

def m(list):

v = list[0]

for e in list:

if v < e: v = e

return v

values = [[3, 4, 5, 1], [33, 6, 1, 2]]

for row in values:

print(m(row), end = " ")

a) 3 33

b) 1 1

c) 5 6

d) 5 33

Answer: d

Explanation: Execute in the shell to verify.

8. What will be the output of the following Python code?

data = [[[1, 2], [3, 4]], [[5, 6], [7, 8]]]

print(data[1][0][0])

a) 1

b) 2

c) 4

d) 5

Answer: d

Explanation: Execute in the shell to verify.

9. What will be the output of the following Python code?

data = [[[1, 2], [3, 4]], [[5, 6], [7, 8]]]

def ttt(m):

v = m[0][0]

for row in m:

for element in row:

if v < element: v = element

return v

print(ttt(data[0]))

a) 1

b) 2

c) 4

d) 5

Answer: c

Explanation: Execute in the shell to verify.

10. What will be the output of the following Python code?

points = [[1, 2], [3, 1.5], [0.5, 0.5]]

points.sort()

print(points)

a) [[1, 2], [3, 1.5], [0.5, 0.5]]

b) [[3, 1.5], [1, 2], [0.5, 0.5]]

c) [[0.5, 0.5], [1, 2], [3, 1.5]]

d) [[0.5, 0.5], [3, 1.5], [1, 2]]

Answer: c

Explanation: Execute in the shell to verify.

1. What will be the output of the following Python code?

a=[10,23,56,[78]]

b=list(a)

a[3][0]=95

a[1]=34

print(b)

a) [10,34,56,[95]]

b) [10,23,56,[78]]

c) [10,23,56,[95]]

d) [10,34,56,[78]]

Answer: c

Explanation: The above copy is a type of shallow copy and only changes made in sublist is reflected in the copied list.

2. What will be the output of the following Python code?

print(list(zip((1,2,3),('a'),('xxx','yyy'))))

print(list(zip((2,4),('b','c'),('yy','xx'))))

a)

[(1,2,3),(‘a’),(‘xxx’,’yyy’)]

[(2,4),(‘b’,’c’),(‘yy’,’xx’)]

b)

[(1, 'a', 'xxx'),(2,’ ‘,’yyy’),(3,’ ‘,’ ‘)]

[(2, 'b', 'yy'), (4, 'c', 'xx')]

c) Syntax error

d)

[(1, 'a', 'xxx')]

[(2, 'b', 'yy'), (4, 'c', 'xx')]

Answer: d

Explanation: The zip function combines the individual attributes of the lists into a list of tuples.

3. What will be the output of the following Python code?

import copy

a=[10,23,56,[78]]

b=copy.deepcopy(a)

a[3][0]=95

a[1]=34

print(b)

a) [10,34,56,[95]]

b) [10,23,56,[78]]

c) [10,23,56,[95]]

d) [10,34,56,[78]]

Answer: b

Explanation: The above copy is deepcopy. Any change made in the original list isn’t reflected.

4. What will be the output of the following Python code?

s="a@b@c@d"

a=list(s.partition("@"))

print(a)

b=list(s.split("@",3))

print(b)

a)

[‘a’,’b’,’c’,’d’]

[‘a’,’b’,’c’,’d’]

b)

[‘a’,’@’,’b’,’@’,’c’,’@’,’d’]

[‘a’,’b’,’c’,’d’]

c)

[‘a’,’@’,’b@c@d’]

[‘a’,’b’,’c’,’d’]

d)

[‘a’,’@’,’b@c@d’]

[‘a’,’@’,’b’,’@’,’c’,’@’,’d’]

Answer: c

Explanation: The partition function only splits for the first parameter along with the separator while split function splits for the number of times given in the second argument but without the separator.

5. What will be the output of the following Python code?

a=[1,2,3,4]

b=[sum(a[0:x+1]) for x in range(0,len(a))]

print(b)

a) 10

b) [1,3,5,7]

c) 4

d) [1,3,6,10]

Answer: d

Explanation: The above code returns the cumulative sum of elements in a list.

6. What will be the output of the following Python code?

a="hello"

b=list((x.upper(),len(x)) for x in a)

print(b)

a) [(‘H’, 1), (‘E’, 1), (‘L’, 1), (‘L’, 1), (‘O’, 1)]

b) [(‘HELLO’, 5)]

c) [(‘H’, 5), (‘E’, 5), (‘L’, 5), (‘L’, 5), (‘O’, 5)]

d) Syntax error

Answer: a

Explanation: Variable x iterates over each letter in string a hence the length of each letter is 1.

7. What will be the output of the following Python code?

a=[1,2,3,4]

b=[sum(a[0:x+1]) for x in range(0,len(a))]

print(b)

a) 10

b) [1,3,5,7]

c) 4

d) [1,3,6,10]

Answer: d

Explanation: The above code returns the cumulative sum of elements in a list.

8. What will be the output of the following Python code?

a=[[]]\*3

a[1].append(7)

print(a)

a) Syntax error

b) [[7], [7], [7]]

c) [[7], [], []]

d) [[],7, [], []]

Answer: b

Explanation: The first line of the code creates multiple reference copies of sublist. Hence when 7 is appended, it gets appended to all the sublists.

9. What will be the output of the following Python code?

b=[2,3,4,5]

a=list(filter(lambda x:x%2,b))

print(a)

a) [2,4]

b) [ ]

c) [3,5]

d) Invalid arguments for filter function

Answer: c

Explanation: The filter function gives value from the list b for which the condition is true, that is, x%2==1.

10. What will be the output of the following Python code?

lst=[3,4,6,1,2]

lst[1:2]=[7,8]

print(lst)

a) [3, 7, 8, 6, 1, 2]

b) Syntax error

c) [3,[7,8],6,1,2]

d) [3,4,6,7,8]

Answer: a

Explanation: In the piece of code, slice assignment has been implemented. The sliced list is replaced by the assigned elements in the list. Type in python shell to verify.

1. What will be the output of the following Python code?

a=[1,2,3]

b=a.append(4)

print(a)

print(b)

a)

[1,2,3,4]

[1,2,3,4]

b)

[1, 2, 3, 4]

None

c) Syntax error

d)

[1,2,3]

[1,2,3,4]

Answer: b

Explanation: Append function on lists doesn’t return anything. Thus the value of b is None.

2. What will be the output of the following Python code?

>>> a=[14,52,7]

>>>> b=a.copy()

>>> b is a

a) True

b) False

Answer: b

Explanation: List b is just a copy of the original list. Any copy made in list b will not be reflected in list a.

3. What will be the output of the following Python code?

a=[13,56,17]

a.append([87])

a.extend([45,67])

print(a)

a) [13, 56, 17, [87], 45, 67]

b) [13, 56, 17, 87, 45, 67]

c) [13, 56, 17, 87,[ 45, 67]]

d) [13, 56, 17, [87], [45, 67]]

Answer: a

Explanation: The append function simply adds its arguments to the list as it is while extend function extends its arguments and later appends it.

4. What is the output of the following piece of code?

a=list((45,)\*4)

print((45)\*4)

print(a)

a)

180

[(45),(45),(45),(45)]

b)

(45,45,45,45)

[45,45,45,45]

c)

180

[45,45,45,45]

d) Syntax error

Answer: c

Explanation: (45) is an int while (45,) is a tuple of one element. Thus when a tuple is multiplied, it created references of itself which is later converted to a list.

5. What will be the output of the following Python code?

lst=[[1,2],[3,4]]

print(sum(lst,[]))

a) [[3],[7]]

b) [1,2,3,4]

c) Error

d) [10]

Answer: b

Explanation: The above piece of code is used for flattening lists.

6. What will be the output of the following Python code?

word1="Apple"

word2="Apple"

list1=[1,2,3]

list2=[1,2,3]

print(word1 is word2)

print(list1 is list2)

a)

True

True

b)

False

True

c)

False

False

d)

True

False

Answer: d

Explanation: In the above case, both the lists are equivalent but not identical as they have different objects.

7. What will be the output of the following Python code?

def unpack(a,b,c,d):

print(a+d)

x = [1,2,3,4]

unpack(\*x)

a) Error

b) [1,4]

c) [5]

d) 5

Answer: d

Explanation: unpack(\*x) unpacks the list into the separate variables. Now, a=1 and d=4. Thus 5 gets printed.

8. What will be the output of the following Python code?

places = ['Bangalore', 'Mumbai', 'Delhi']

<br class="blank" />places1 = places

places2 = places[:]

<br class="blank" />places1[1]="Pune"

places2[2]="Hyderabad"

print(places)

a) [‘Bangalore’, ‘Pune’, ‘Hyderabad’]

b) [‘Bangalore’, ‘Pune’, ‘Delhi’]

c) [‘Bangalore’, ‘Mumbai’, ‘Delhi’]

d) [‘Bangalore’, ‘Mumbai’, ‘Hyderabad’]

Answer: b

Explanation: places1 is an alias of the list places. Hence, any change made to places1 is reflected in places. places2 is a copy of the list places. Thus, any change made to places2 isn’t reflected in places.

9. What will be the output of the following Python code?

x=[[1],[2]]

print(" ".join(list(map(str,x))))

a) [1] [2]

b) [49] [50]

c) Syntax error

d) [[1]] [[2]]

Answer: a

Explanation: The elements 1 and 2 are first put into separate lists and then combined with a space in between using the join attribute.

10. What will be the output of the following Python code?

a=165

b=sum(list(map(int,str(a))))

print(b)

a) 561

b) 5

c) 12

d) Syntax error

Answer: c

Explanation: First, map converts the number to string and then places the individual digits in a list. Then, sum finds the sum of the digits in the list. The code basically finds the sum of digits in the number.

11. What will be the output of the following Python code?

a= [1, 2, 3, 4, 5]

for i in range(1, 5):

a[i-1] = a[i]

for i in range(0, 5):

print(a[i],end = " ")

a) 5 5 1 2 3

b) 5 1 2 3 4

c) 2 3 4 5 1

d) 2 3 4 5 5

Answer: d

Explanation: The items having indexes from 1 to 4 are shifted forward by one index due to the first for-loop and the item of index four is printed again because of the second for-loop.

12. What will be the output of the following Python code?

def change(var, lst):

var = 1

lst[0] = 44

k = 3

a = [1, 2, 3]

change(k, a)

print(k)

print(a)

a)

3

[44, 2, 3]

b)

1

[1,2,3]

c)

3

[1,2,3]

d)

1

[44,2,3]

Answer: a

Explanation: A list is mutable, hence it’s value changes after function call. However, integer isn’t mutable. Thus its value doesn’t change.

13. What will be the output of the following Python code?

a = [1, 5, 7, 9, 9, 1]

<br class="blank" />b=a[0]

<br class="blank" />x= 0

for x in range(1, len(a)):

if a[x] > b:

b = a[x]

b= x

print(b)

a) 5

b) 3

c) 4

d) 0

Answer: c

Explanation: The above piece of code basically prints the index of the largest element in the list.

14. What will be the output of the following Python code?

a=["Apple","Ball","Cobra"]

<br class="blank" />a.sort(key=len)

print(a)

a) [‘Apple’, ‘Ball’, ‘Cobra’] b) [‘Ball’, ‘Apple’, ‘Cobra’]

c) [‘Cobra’, ‘Apple’, ‘Ball’] d) Invalid syntax for sort()

Answer: b

Explanation: The syntax isn’t invalid and the list is sorted according to the length of the strings in the list since key is given as len.

15. What will be the output of the following Python code?

num = ['One', 'Two', 'Three']

for i, x in enumerate(num):

print('{}: {}'.format(i, x),end=" ")

a) 1: 2: 3: b) Exception is thrown c) One Two Three d) 0: One 1: Two 2: Three

Answer: d

Explanation: enumerate(iterator,start=0) is a built-in function which returns (0,lst[0]),(1,lst[1]) and so on where lst is a list(iterator).

# List Comprehension

1. What will be the output of the following Python code snippet?

k = [print(i) for i in my\_string if i not in "aeiou"]

a) prints all the vowels in my\_string

b) prints all the consonants in my\_string

c) prints all characters of my\_string that aren’t vowels

d) prints only on executing print(k)

Answer: c

Explanation: print(i) is executed if the given character is not a vowel.

2. What is the output of print(k) in the following Python code snippet?

k = [print(i) for i in my\_string if i not in "aeiou"]

print(k)

a) all characters of my\_string that aren’t vowels

b) a list of Nones

c) list of Trues

d) list of Falses

Answer: b

Explanation: print() returns None.

3. What will be the output of the following Python code snippet?

my\_string = "hello world"

k = [(i.upper(), len(i)) for i in my\_string]

print(k)

a) [(‘HELLO’, 5), (‘WORLD’, 5)]

b) [(‘H’, 1), (‘E’, 1), (‘L’, 1), (‘L’, 1), (‘O’, 1), (‘ ‘, 1), (‘W’, 1), (‘O’, 1), (‘R’, 1), (‘L’, 1), (‘D’, 1)]

c) [(‘HELLO WORLD’, 11)]

d) none of the mentioned

Answer: b

Explanation: We are iterating over each letter in the string.

4. Which of the following is the correct expansion of list\_1 = [expr(i) for i in list\_0 if func(i)]?

a)

list\_1 = []

for i in list\_0:

if func(i):

list\_1.append(i)

b)

for i in list\_0:

if func(i):

list\_1.append(expr(i))

c)

list\_1 = []

for i in list\_0:

if func(i):

list\_1.append(expr(i))

d) none of the mentioned

Answer: c

Explanation: We have to create an empty list, loop over the contents of the existing list and check if a condition is satisfied before performing some operation and adding it to the new list.

5. What will be the output of the following Python code snippet?

x = [i\*\*+1 for i in range(3)]; print(x);

a) [0, 1, 2]

b) [1, 2, 5]

c) error, \*\*+ is not a valid operator

d) error, ‘;’ is not allowed

Answer: a

Explanation: i\*\*+1 is evaluated as (i)\*\*(+1).

6. What will be the output of the following Python code snippet?

print([i.lower() for i in "HELLO"])

a) [‘h’, ‘e’, ‘l’, ‘l’, ‘o’]

b) ‘hello’

c) [‘hello’]

d) hello

Answer: a

Explanation: We are iterating over each letter in the string.

7. What will be the output of the following Python code snippet?

print([i+j for i in "abc" for j in "def"])

a) [‘da’, ‘ea’, ‘fa’, ‘db’, ‘eb’, ‘fb’, ‘dc’, ‘ec’, ‘fc’]

b) [[‘ad’, ‘bd’, ‘cd’], [‘ae’, ‘be’, ‘ce’], [‘af’, ‘bf’, ‘cf’]]

c) [[‘da’, ‘db’, ‘dc’], [‘ea’, ‘eb’, ‘ec’], [‘fa’, ‘fb’, ‘fc’]]

d) [‘ad’, ‘ae’, ‘af’, ‘bd’, ‘be’, ‘bf’, ‘cd’, ‘ce’, ‘cf’]

Answer: d

Explanation: If it were to be executed as a nested for loop, i would be the outer loop and j the inner loop.

8. What will be the output of the following Python code snippet?

print([[i+j for i in "abc"] for j in "def"])

a) [‘da’, ‘ea’, ‘fa’, ‘db’, ‘eb’, ‘fb’, ‘dc’, ‘ec’, ‘fc’]

b) [[‘ad’, ‘bd’, ‘cd’], [‘ae’, ‘be’, ‘ce’], [‘af’, ‘bf’, ‘cf’]]

c) [[‘da’, ‘db’, ‘dc’], [‘ea’, ‘eb’, ‘ec’], [‘fa’, ‘fb’, ‘fc’]]

d) [‘ad’, ‘ae’, ‘af’, ‘bd’, ‘be’, ‘bf’, ‘cd’, ‘ce’, ‘cf’]

Answer: b

Explanation: The inner list is generated once for each value of j.

9. What will be the output of the following Python code snippet?

print([if i%2==0: i; else: i+1; for i in range(4)])

a) [0, 2, 2, 4]

b) [1, 1, 3, 3]

c) error

d) none of the mentioned

Answer: c

Explanation: Syntax error.

10. Which of the following is the same as list(map(lambda x: x\*\*-1, [1, 2, 3]))?

a) [x\*\*-1 for x in [(1, 2, 3)]]

b) [1/x for x in [(1, 2, 3)]]

c) [1/x for x in (1, 2, 3)]

d) error

Answer: c

Explanation: x\*\*-1 is evaluated as (x)\*\*(-1).

1. What will be the output of the following Python code?

l=[1,2,3,4,5]

[x&1 for x in l]

a) [1, 1, 1, 1, 1]

b) [1, 0, 1, 0, 1]

c) [1, 0, 0, 0, 0]

d) [0, 1, 0, 1, 0]

Answer: b

Explanation: In the code shown above, each of the numbers of the list, that is, 1, 2, 3, 4 and 5 are AND-ed with 1 and the result is printed in the form of a list. Hence the output is [1, 0, 1, 0, 1].

2. What will be the output of the following Python code?

l1=[1,2,3]

l2=[4,5,6]

[x\*y for x in l1 for y in l2]

a) [4, 8, 12, 5, 10, 15, 6, 12, 18]

b) [4, 10, 18]

c) [4, 5, 6, 8, 10, 12, 12, 15, 18]

d) [18, 12, 6, 15, 10, 5, 12, 8, 4]

Answer: c

Explanation: The code shown above returns x\*y, where x belongs to the list l1 and y belongs to the list l2. Therefore, the output is: [4, 5, 6, 8, 10, 12, 12, 15, 18].

3. Write the list comprehension to pick out only negative integers from a given list ‘l’.

a) [x<0 in l]

b) [x for x<0 in l]

c) [x in l for x<0]

d) [x for x in l if x<0]

Answer: d

Explanation: To pick out only the negative numbers from a given list ‘l’, the correct list comprehension statement would be: [x for x in l if x<0].

For example if we have a list l=[-65, 2, 7, -99, -4, 3]

>>> [x for x in l if x<0]

The output would be: [-65, -99, -4].

4. What will be the output of the following Python code?

s=["pune", "mumbai", "delhi"]

[(w.upper(), len(w)) for w in s]

a) Error

b) [‘PUNE’, 4, ‘MUMBAI’, 6, ‘DELHI’, 5]

c) [PUNE, 4, MUMBAI, 6, DELHI, 5]

d) [(‘PUNE’, 4), (‘MUMBAI’, 6), (‘DELHI’, 5)]

Answer: d

Explanation: If we need to generate two results, we need to put it in the form of a tuple. The code shown above returns each word of list in uppercase, along with the length of the word. Hence the output of the code is: [(‘PUNE’, 4), (‘MUMBAI’, 6), (‘DELHI’, 5)].

5. What will be the output of the following Python code?

l1=[2,4,6]

l2=[-2,-4,-6]

for i in zip(l1, l2):

print(i)

a)

2, -2

4, -4

6, -6

b) [(2, -2), (4, -4), (6, -6)]

c)

(2, -2)

(4, -4)

(6, -6)

d) [-4, -16, -36]

Answer: c

Explanation: The output of the code shown will be:

(2, -2)

(4, -4)

(6, -6)

This format is due to the statement print(i).

6. What will be the output of the following Python code?

l1=[10, 20, 30]

l2=[-10, -20, -30]

l3=[x+y for x, y in zip(l1, l2)]

l3

a) Error

b) 0

c) [-20, -60, -80]

d) [0, 0, 0]

Answer: d

Explanation: The code shown above returns x+y, for x belonging to the list l1 and y belonging to the list l2. That is, l3=[10-10, 20-20, 30-20], which is, [0, 0, 0].

7. Write a list comprehension for number and its cube for l=[1, 2, 3, 4, 5, 6, 7, 8, 9].

a) [x\*\*3 for x in l]

b) [x^3 for x in l]

c) [x\*\*3 in l]

d) [x^3 in l]

Answer: a

Explanation: The list comprehension to print a list of cube of the numbers for the given list is: [x\*\*3 for x in l].

8. What will be the output of the following Python code?

l=[[1 ,2, 3], [4, 5, 6], [7, 8, 9]]

[[row[i] for row in l] for i in range(3)]

a) Error

b) [[1, 4, 7], [2, 5, 8], [3, 6, 9]]

c)

1 4 7

2 5 8

3 6 9

d)

(1 4 7)

(2 5 8)

(3 6 9)

Answer: b

Explanation: In the code shown above, ‘3’ is the index of the list. Had we used a number greater than 3, it would result in an error. The output of this code is: [[1, 4, 7], [2, 5, 8], [3, 6, 9]].

9. What will be the output of the following Python code?

import math

[str(round(math.pi)) for i in range (1, 6)]

a) [‘3’, ‘3’, ‘3’, ‘3’, ‘3’, ‘3’]

b) [‘3.1’, ‘3.14’, ‘3.142’, ‘3.1416’, ‘3.14159’, ‘3.141582’]

c) [‘3’, ‘3’, ‘3’, ‘3’, ‘3’]

d) [‘3.1’, ‘3.14’, ‘3.142’, ‘3.1416’, ‘3.14159’]

Answer: c

Explanation: The list comprehension shown above rounds off pi(3.141) and returns its value, that is 3. This is done 5 times. Hence the output is: [‘3’, ‘3’, ‘3’, ‘3’, ‘3’].

10. What will be the output of the following Python code?

l1=[1,2,3]

l2=[4,5,6]

l3=[7,8,9]

for x, y, z in zip(l1, l2, l3):

print(x, y, z)

a)

1 4 7

2 5 8

3 6 9

b)

(1 4 7)

(2 5 8)

(3 6 9)

c) [(1, 4, 7), (2, 5, 8), (3, 6, 9)]

d) Error

Answer: a

Explanation: The output of the code shown above is:

1 4 7

2 5 8

3 6 9

This is due to the statement: print(x, y,z).

1. Read the information given below carefully and write a list comprehension such that the output is: [‘e’, ‘o’]

w="hello"

v=('a', 'e', 'i', 'o', 'u')

a) [x for w in v if x in v]

b) [x for x in w if x in v]

c) [x for x in v if w in v]

d) [x for v in w for x in w]

Answer: b

Explanation: The tuple ‘v’ is used to generate a list containing only vowels in the string ‘w’. The result is a list containing only vowels present in the string “hello”. Hence the required list comprehension is: [x for x in w if x in v].

2. What will be the output of the following Python code?

[ord(ch) for ch in 'abc']

a) [97, 98, 99]

b) [‘97’, ‘98’, ‘99’]

c) [65, 66, 67]

d) Error

Answer: a

Explanation: The list comprehension shown above returns the ASCII value of each alphabet of the string ‘abc’. Hence the output is: [97, 98, 99]. Had the string been ‘ABC’, the output would be: [65, 66, 67].

3. What will be the output of the following Python code?

t=32.00

[round((x-32)\*5/9) for x in t]

a) [0]

b) 0

c) [0.00]

d) Error

Answer: d

Explanation: The value of t in the code shown above is equal to 32.00, which is a floating point value. ‘Float’ objects are not iterable. Hence the code results in an error.

4. Write a list comprehension for producing a list of numbers between 1 and 1000 that are divisible by 3.

a) [x in range(1, 1000) if x%3==0]

b) [x for x in range(1000) if x%3==0]

c) [x%3 for x in range(1, 1000)]

d) [x%3=0 for x in range(1, 1000)]

Answer: b

Explanation: The list comprehension [x for x in range(1000) if x%3==0] produces a list of numbers between 1 and 1000 that are divisible by 3.

5. Write a list comprehension equivalent for the Python code shown below.

for i in range(1, 101):

if int(i\*0.5)==i\*0.5:

print(i)

a) [i for i in range(1, 100) if int(i\*0.5)==(i\*0.5)]

b) [i for i in range(1, 101) if int(i\*0.5)==(i\*0.5)]

c) [i for i in range(1, 101) if int(i\*0.5)=(i\*0.5)]

d) [i for i in range(1, 100) if int(i\*0.5)=(i\*0.5)]

Answer: b

Explanation: The code shown above prints the value ‘i’ only if it satisfies the condition: int(i\*0.5) is equal to (i\*0.5). Hence the required list comprehension is: [i for i in range(1, 101) if int(i\*0.5)==(i\*0.5)].

6. What is the list comprehension equivalent for: list(map(lambda x:x\*\*-1, [1, 2, 3]))?

a) [1|x for x in [1, 2, 3]]

b) [-1\*\*x for x in [1, 2, 3]]

c) [x\*\*-1 for x in [1, 2, 3]]

d) [x^-1 for x in range(4)]

Answer: c

Explanation: The output of the function list(map(lambda x:x\*\*-1, [1, 2, 3])) is [1.0, 0.5, 0.3333333333333333] and that of the list comprehension [x\*\*-1 for x in [1, 2, 3]] is [1.0, 0.5, 0.3333333333333333]. Hence the answer is: [x\*\*-1 for x in [1, 2, 3]].

7. Write a list comprehension to produce the list: [1, 2, 4, 8, 16……212].

a) [(2\*\*x) for x in range(0, 13)]

b) [(x\*\*2) for x in range(1, 13)]

c) [(2\*\*x) for x in range(1, 13)]

d) [(x\*\*2) for x in range(0, 13)]

Answer: a

Explanation: The required list comprehension will print the numbers from 1 to 12, each raised to 2. The required answer is thus, [(2\*\*x) for x in range(0, 13)].

8. What is the list comprehension equivalent for?

{x : x is a whole number less than 20, x is even} (including zero)

a) [x for x in range(1, 20) if (x%2==0)]

b) [x for x in range(0, 20) if (x//2==0)]

c) [x for x in range(1, 20) if (x//2==0)]

d) [x for x in range(0, 20) if (x%2==0)]

Answer: d

Explanation: The required list comprehension will print a whole number, less than 20, provided that the number is even. Since the output list should contain zero as well, the answer to this question is: [x for x in range(0, 20) if (x%2==0)].

9. What will be the output of the following Python list comprehension?

[j for i in range(2,8) for j in range(i\*2, 50, i)]

a) A list of prime numbers up to 50

b) A list of numbers divisible by 2, up to 50

c) A list of non prime numbers, up to 50

d) Error

Answer: c

Explanation: The list comprehension shown above returns a list of non-prime numbers up to 50. The logic behind this is that the square root of 50 is almost equal to 7. Hence all the multiples of 2-7 are not prime in this range.

10. What will be the output of the following Python code?

l=["good", "oh!", "excellent!", "#450"]

[n for n in l if n.isalpha() or n.isdigit()]

a) [‘good’, ‘oh’, ‘excellent’, ‘450’ ]

b) [‘good’]

c) [‘good’, ‘#450’]

d) [‘oh!’, ‘excellent!’, ‘#450’]

Answer: b

Explanation: The code shown above returns a new list containing only strings which do not have any punctuation in them. The only string from the list which does not contain any punctuation is ‘good’. Hence the output of the code shown above is [‘good’].

# Matrix List Comprehension

1. Which of the following matrices will throw an error in Python?

a)

A = [[1, 2, 3],

[4, 5, 6],

[7, 8, 9]]

b)

B = [[3, 3, 3]

[4, 4, 4]

[5, 5, 5]]

c)

C = [(1, 2, 4),

(5, 6, 7),

(8, 9, 10)]

d)

D = [2, 3, 4,

3, 3, 3,

4, 5, 6]

Answer: b

Explanation: In matrix B will result in an error because in the absence of a comma at the end of each row, it behaves like three separate lists. The error thrown states that the list integers must be integers or slices, not tuples.

2. What will be the output of the following Python code?

A = [[1, 2, 3],

[4, 5, 6],

[7, 8, 9]]

A[1]

a) [4, 5, 6]

b) [3, 6, 9]

c) [1, 4, 7]

d) [1, 2, 3]

Answer: a

Explanation: We can index the rows and columns using normal index operations. The statement A[1] represents the second row, that is, the middle row. Hence the output of the code will be: [4, 5, 6].

3. Which of the following Python statements will result in the output: 6?

A = [[1, 2, 3],

[4, 5, 6],

[7, 8, 9]]

a) A[2][3]

b) A[2][1]

c) A[1][2]

d) A[3][2]

Answer: c

Explanation: The output that is required is 6, that is, row 2, item 3. This position is represented by the statement: A[1][2].

4. What will be the output of the following Python code?

A = [[1, 2, 3],

[4, 5, 6],

[7, 8, 9]]

[A[row][1] for row in (0, 1, 2)]

a) [7, 8, 9]

b) [4, 5, 6]

c) [2, 5, 8]

d) [1, 4, 7]

Answer: c

Explanation: To get a particular column as output, we can simple iterate across the rows and pull out the desired column, or iterate through positions in rows and index as we go. Hence the output of the code shown above is: [2, 5, 8].

5. What will be the output of the following Python code?

A = [[1, 2, 3],

[4, 5, 6],

[7, 8, 9]]

[A[i][i] for i in range(len(A))]

a) [1, 5, 9]

b) [3, 5, 7]

c) [4, 5, 6]

d) [2, 5, 8]

Answer: a

Explanation: We can also perform tasks like pulling out a diagonal. The expression shown above uses range to generate the list of offsets and the indices with the row and column the same, picking out A[0][0], then A[1][1] and so on. Hence the output of the code is: [1, 5, 9].

6. What will be the output of the following Python code?

l=[[1, 2, 3], [4, 5, 6]]

for i in range(len(l)):

for j in range(len(l[i])):

l[i][j]+=10

l

a) No output

b) Error

c) [[1, 2, 3], [4, 5, 6]]

d) [[11, 12, 13], [14, 15, 16]]

Answer: d

Explanation: We use range twice if the shapes differ. Each element of list l is increased by 10. Hence the output is: [[11, 12, 13], [14, 15, 16]]

7. What will be the output of the following Python code?

A = [[1, 2, 3],

[4, 5, 6],

[7, 8, 9]]

[[col + 10 for col in row] for row in A]

a) [[11, 12, 13], [14, 15, 16], [17, 18, 19]]

b) Error

c) [11, 12, 13], [14, 15, 16], [17, 18, 19]

d) [11, 12, 13, 14, 15, 16, 17, 18, 19]

Answer: a

Explanation: The code shown above shows a list comprehension which adds 10 to each element of the matrix A and prints it row-wise. Hence the output of the code is: [[11, 12, 13], [14, 15, 16], [17, 18, 19]]

8. What will be the output of the following Python code?

A = [[1, 2, 3],

[4, 5, 6],

[7, 8, 9]]

[A[i][len(A)-1-i] for i in range(len(A))]

a) [1, 5, 9]

b) [4, 5, 6]

c) [3, 5, 7]

d) [2, 5, 8]

Answer: c

Explanation: This expression scales the common index to fetch A[0][2], A[1][1], etc. We assume the matrix has the same number of rows and columns.

9. What will be the output of the following Python code?

A = [[1, 2, 3],

[4, 5, 6],

[7, 8, 9]]

B = [[3, 3, 3],

[4, 4, 4],

[5, 5, 5]]

[B[row][col]\*A[row][col] for row in range(3) for col in range(3)]

a) [3, 6, 9, 16, 20, 24, 35, 40, 45]

b) Error

c) [0, 30, 60, 120, 160, 200, 300, 350, 400]

d) 0

Answer: a

Explanation: In the code shown above, we have used list comprehension to combine values of multiple matrices. We have multiplied the elements of the matrix B with that of the matrix A, in the range(3). Hence the output of this code is: [3, 6, 9, 16, 20, 24, 35, 40, 45].

10. What will be the output of the following Python code?

r = [11, 12, 13, 14, 15, 16, 17, 18, 19]

A = [[0, 10, 20],

[30, 40, 50],

[60, 70, 80]]

for row in A:

for col in row:

r.append(col+10)

r

a) [11, 12, 13, 14, 15, 16, 17, 18, 19, 10, 20, 30, 40, 50, 60, 70, 80, 90]

b) [10, 20, 30, 40, 50, 60, 70, 80, 90]

c) [11, 12, 13, 14, 15, 16, 17, 18, 19]

d) [0, 10, 20, 30, 40, 50, 60, 70, 80]

Answer: a

Explanation: The code shown above adds 10 to each element of the matrix and prints the output row-wise. Since the list l already contains some elements, the new elements are appended to it. Hence the output of this code is: [11, 12, 13, 14, 15, 16, 17, 18, 19, 10, 20, 30, 40, 50, 60, 70, 80, 90].

11. What will be the output of the following Python code?

A = [[1, 2, 3],

[4, 5, 6],

[7, 8, 9]]

B = [[3, 3, 3],

[4, 4, 4],

[5, 5, 5]]

[[col1 \* col2 for (col1, col2) in zip(row1, row2)] for (row1, row2) in zip(A, B)]

a) [0, 30, 60, 120, 160, 200, 300, 350, 400]

b) [[3, 6, 9], [16, 20, 24], [35, 40, 45]]

c) No output

d) Error

Answer: b

Explanation: The list comprehension shown above results in the output: [[3, 6, 9], [16, 20, 24], [35, 40, 45]].

12. What will be the output of the following Python code?

A = [[1, 2, 3],

[4, 5, 6],

[7, 8, 9]]

B = [[3, 3, 3],

[4, 4, 4],

[5, 5, 5]]

zip(A, B)

a) Address of the zip object

b) Address of the matrices A and B

c) No output

d) [3, 6, 9, 16, 20, 24, 35, 40, 45]

Answer: a

Explanation: The output of the code shown above returns the address of the zip object. If we print it in the form of a list, we get:

>>> list(zip(A, B))

[([1, 2, 3], [3, 3, 3]), ([4, 5, 6], [4, 4, 4]), ([7, 8, 9], [5, 5, 5])]

# Tuples – 1

1. Which of the following is a Python tuple?

a) [1, 2, 3]

b) (1, 2, 3)

c) {1, 2, 3}

d) {}

Answer: b

Explanation: Tuples are represented with round brackets.

2. Suppose t = (1, 2, 4, 3), which of the following is incorrect?

a) print(t[3])

b) t[3] = 45

c) print(max(t))

d) print(len(t))

Answer: b

Explanation: Values cannot be modified in the case of tuple, that is, tuple is immutable.

3. What will be the output of the following Python code?

>>>t=(1,2,4,3)

>>>t[1:3]

a) (1, 2)

b) (1, 2, 4)

c) (2, 4)

d) (2, 4, 3)

Answer: c

Explanation: Slicing in tuples takes place just as it does in strings.

4. What will be the output of the following Python code?

>>>t=(1,2,4,3)

>>>t[1:-1]

a) (1, 2)

b) (1, 2, 4)

c) (2, 4)

d) (2, 4, 3)

Answer: c

Explanation: Slicing in tuples takes place just as it does in strings.

5. What will be the output of the following Python code?

>>>t = (1, 2, 4, 3, 8, 9)

>>>[t[i] for i in range(0, len(t), 2)]

a) [2, 3, 9]

b) [1, 2, 4, 3, 8, 9]

c) [1, 4, 8]

d) (1, 4, 8)

Answer: c

Explanation: Execute in the shell to verify.

6. What will be the output of the following Python code?

d = {"john":40, "peter":45}

d["john"]

a) 40

b) 45

c) “john”

d) “peter”

Answer: a

Explanation: Execute in the shell to verify.

7. What will be the output of the following Python code?

>>>t = (1, 2)

>>>2 \* t

a) (1, 2, 1, 2)

b) [1, 2, 1, 2]

c) (1, 1, 2, 2)

d) [1, 1, 2, 2]

Answer: a

Explanation: \* operator concatenates tuple.

8. What will be the output of the following Python code?

>>>t1 = (1, 2, 4, 3)

>>>t2 = (1, 2, 3, 4)

>>>t1 < t2

a) True

b) False

c) Error

d) None

Answer: b

Explanation: Elements are compared one by one in this case.

9. What will be the output of the following Python code?

>>>my\_tuple = (1, 2, 3, 4)

>>>my\_tuple.append( (5, 6, 7) )

>>>print len(my\_tuple)

a) 1

b) 2

c) 5

d) Error

Answer: d

Explanation: Tuples are immutable and don’t have an append method. An exception is thrown in this case.

10. What will be the output of the following Python code?

numberGames = {}

numberGames[(1,2,4)] = 8

numberGames[(4,2,1)] = 10

numberGames[(1,2)] = 12

sum = 0

for k in numberGames:

sum += numberGames[k]

print len(numberGames) + sum

a) 30

b) 24

c) 33

d) 12

Answer: c

Explanation: Tuples can be used for keys into dictionary. The tuples can have mixed length and the order of the items in the tuple is considered when comparing the equality of the keys.

1. What is the data type of (1)?

a) Tuple

b) Integer

c) List

d) Both tuple and integer

Answer: b

Explanation: A tuple of one element must be created as (1,).

2. If a=(1,2,3,4), a[1:-1] is \_\_\_\_\_\_\_\_\_

a) Error, tuple slicing doesn’t exist

b) [2,3]

c) (2,3,4)

d) (2,3)

Answer: d

Explanation: Tuple slicing exists and a[1:-1] returns (2,3).

3. What will be the output of the following Python code?

>>> a=(1,2,(4,5))

>>> b=(1,2,(3,4))

>>> a<b

a) False

b) True

c) Error, < operator is not valid for tuples

d) Error, < operator is valid for tuples but not if there are sub-tuples

Answer: a

Explanation: Since the first element in the sub-tuple of a is larger that the first element in the subtuple of b, False is printed.

4. What will be the output of the following Python code?

>>> a=("Check")\*3

>>> a

a) (‘Check’,’Check’,’Check’)

b) \* Operator not valid for tuples

c) (‘CheckCheckCheck’)

d) Syntax error

Answer: c

Explanation: Here (“Check”) is a string not a tuple because there is no comma after the element.

5. What will be the output of the following Python code?

>>> a=(1,2,3,4)

>>> del(a[2])

a) Now, a=(1,2,4)

b) Now, a=(1,3,4)

c) Now a=(3,4)

d) Error as tuple is immutable

Answer: d

Explanation: ‘tuple’ object doesn’t support item deletion.

6. What will be the output of the following Python code?

>>> a=(2,3,4)

>>> sum(a,3)

a) Too many arguments for sum() method

b) The method sum() doesn’t exist for tuples

c) 12

d) 9

Answer: c

Explanation: In the above case, 3 is the starting value to which the sum of the tuple is added to.

7. Is the following Python code valid?

>>> a=(1,2,3,4)

>>> del a

a) No because tuple is immutable

b) Yes, first element in the tuple is deleted

c) Yes, the entire tuple is deleted

d) No, invalid syntax for del method

Answer: c

Explanation: The command del a deletes the entire tuple.

8. What type of data is: a=[(1,1),(2,4),(3,9)]?

a) Array of tuples

b) List of tuples

c) Tuples of lists

d) Invalid type

Answer: b

Explanation: The variable a has tuples enclosed in a list making it a list of tuples.

9. What will be the output of the following Python code?

>>> a=(0,1,2,3,4)

>>> b=slice(0,2)

>>> a[b]

a) Invalid syntax for slicing

b) [0,2]

c) (0,1)

d) (0,2)

Answer: c

Explanation: The method illustrated in the above piece of code is that of naming of slices.

10. Is the following Python code valid?

>>> a=(1,2,3)

>>> b=('A','B','C')

>>> c=tuple(zip(a,b))

a) Yes, c will be ((1, ‘A’), (2, ‘B’), (3, ‘C’))

b) Yes, c will be ((1,2,3),(‘A’,’B’,’C’))

c) No because tuples are immutable

d) No because the syntax for zip function isn’t valid

Answer: a

Explanation: Zip function combines individual elements of two iterables into tuples. Execute in Python shell to verify.

1. Is the following Python code valid?

>>> a,b,c=1,2,3

>>> a,b,c

a) Yes, [1,2,3] is printed

b) No, invalid syntax

c) Yes, (1,2,3) is printed

d) 1 is printed

Answer: c

Explanation: A tuple needn’t be enclosed in parenthesis.

2. What will be the output of the following Python code?

a = ('check',)

n = 2

for i in range(int(n)):

a = (a,)

print(a)

a) Error, tuples are immutable

b)

(('check',),)

((('check',),),)

c) ((‘check’,)’check’,)

d)

(('check',)’check’,)

((('check',)’check’,)’check’,)

Answer: b

Explanation: The loop runs two times and each time the loop runs an extra parenthesis along with a comma is added to the tuple (as a=(a’)).

3. Is the following Python code valid?

>>> a,b=1,2,3

a) Yes, this is an example of tuple unpacking. a=1 and b=2

b) Yes, this is an example of tuple unpacking. a=(1,2) and b=3

c) No, too many values to unpack

d) Yes, this is an example of tuple unpacking. a=1 and b=(2,3)

Answer: c

Explanation: For unpacking to happen, the number of values of the right hand side must be equal to the number of variables on the left hand side.

4. What will be the output of the following Python code?

>>> a=(1,2)

>>> b=(3,4)

>>> c=a+b

>>> c

a) (4,6)

b) (1,2,3,4)

c) Error as tuples are immutable

d) None

Answer: b

Explanation: In the above piece of code, the values of the tuples aren’t being changed. Both the tuples are simply concatenated.

5. What will be the output of the following Python code?

>>> a,b=6,7

>>> a,b=b,a

>>> a,b

a) (6,7)

b) Invalid syntax

c) (7,6)

d) Nothing is printed

Answer: c

Explanation: The above piece of code illustrates the unpacking of variables.

6. What will be the output of the following Python code?

>>> import collections

>>> a=collections.namedtuple('a',['i','j'])

>>> obj=a(i=4,j=7)

>>> obj

a) a(i=4, j=7)

b) obj(i=4, j=7)

c) (4,7)

d) An exception is thrown

Answer: a

Explanation: The above piece of code illustrates the concept of named tuples.

7. Tuples can’t be made keys of a dictionary.

a) True

b) False

Answer: b

Explanation: Tuples can be made keys of a dictionary because they are hashable.

8. Is the following Python code valid?

>>> a=2,3,4,5

>>> a

a) Yes, 2 is printed

b) Yes, [2,3,4,5] is printed

c) No, too many values to unpack

d) Yes, (2,3,4,5) is printed

Answer: d

Explanation: A tuple needn’t be enclosed in parenthesis.

9. What will be the output of the following Python code?

>>> a=(2,3,1,5)

>>> a.sort()

>>> a

a) (1,2,3,5)

b) (2,3,1,5)

c) None

d) Error, tuple has no attribute sort

Answer: d

Explanation: A tuple is immutable thus it doesn’t have a sort attribute.

10. Is the following Python code valid?

>>> a=(1,2,3)

>>> b=a.update(4,)

a) Yes, a=(1,2,3,4) and b=(1,2,3,4)

b) Yes, a=(1,2,3) and b=(1,2,3,4)

c) No because tuples are immutable

d) No because wrong syntax for update() method

Answer: c

Explanation: Tuple doesn’t have any update() attribute because it is immutable.

11. What will be the output of the following Python code?

>>> a=[(2,4),(1,2),(3,9)]

>>> a.sort()

>>> a

a) [(1, 2), (2, 4), (3, 9)]

b) [(2,4),(1,2),(3,9)]

c) Error because tuples are immutable

d) Error, tuple has no sort attribute

Answer: a

Explanation: A list of tuples is a list itself. Hence items of a list can be sorted.

# Sets – 1

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Sets – 1”.

1. Which of these about a set is not true?

a) Mutable data type

b) Allows duplicate values

c) Data type with unordered values

d) Immutable data type

Answer: d

Explanation: A set is a mutable data type with non-duplicate, unordered values, providing the usual mathematical set operations.

2. Which of the following is not the correct syntax for creating a set?

a) set([[1,2],[3,4]])

b) set([1,2,2,3,4])

c) set((1,2,3,4))

d) {1,2,3,4}

Answer: a

Explanation: The argument given for the set must be an iterable.

3. What will be the output of the following Python code?

nums = set([1,1,2,3,3,3,4,4])

print(len(nums))

a) 7

b) Error, invalid syntax for formation of set

c) 4

d) 8

Answer: c

Explanation: A set doesn’t have duplicate items.

4. What will be the output of the following Python code?

a = [5,5,6,7,7,7]

b = set(a)

def test(lst):

if lst in b:

return 1

else:

return 0

for i in filter(test, a):

print(i,end=" ")

a) 5 5 6

b) 5 6 7

c) 5 5 6 7 7 7

d) 5 6 7 7 7

Answer: c

Explanation: The filter function will return all the values from list a which are true when passed to function test. Since all the members of the set are non-duplicate members of the list, all of the values will return true. Hence all the values in the list are printed.

5. Which of the following statements is used to create an empty set?

a) { }

b) set()

c) [ ]

d) ( )

Answer: b

Explanation: { } creates a dictionary not a set. Only set() creates an empty set.

6. What will be the output of the following Python code?

>>> a={5,4}

>>> b={1,2,4,5}

>>> a<b

a) {1,2}

b) True

c) False

d) Invalid operation

Answer: b

Explanation: a<b returns True if a is a proper subset of b.

7. If a={5,6,7,8}, which of the following statements is false?

a) print(len(a))

b) print(min(a))

c) a.remove(5)

d) a[2]=45

Answer: d

Explanation: The members of a set can be accessed by their index values since the elements of the set are unordered.

8. If a={5,6,7}, what happens when a.add(5) is executed?

a) a={5,5,6,7}

b) a={5,6,7}

c) Error as there is no add function for set data type

d) Error as 5 already exists in the set

Answer: b

Explanation: There exists add method for set data type. However 5 isn’t added again as set consists of only non-duplicate elements and 5 already exists in the set. Execute in python shell to verify.

9. What will be the output of the following Python code?

>>> a={4,5,6}

>>> b={2,8,6}

>>> a+b

a) {4,5,6,2,8}

b) {4,5,6,2,8,6}

c) Error as unsupported operand type for sets

d) Error as the duplicate item 6 is present in both sets

Answer: c

Explanation: Execute in python shell to verify.

10. What will be the output of the following Python code?

>>> a={4,5,6}

>>> b={2,8,6}

>>> a-b

a) {4,5}

b) {6}

c) Error as unsupported operand type for set data type

d) Error as the duplicate item 6 is present in both sets

Answer: a

Explanation: – operator gives the set of elements in set a but not in set b.

11. What will be the output of the following Python code?

>>> a={5,6,7,8}

>>> b={7,8,10,11}

>>> a^b

a) {5,6,7,8,10,11}

b) {7,8}

c) Error as unsupported operand type of set data type

d) {5,6,10,11}

Answer: d

Explanation: ^ operator returns a set of elements in set A or set B, but not in both (symmetric difference).

12. What will be the output of the following Python code?

>>> s={5,6}

>>> s\*3

a) Error as unsupported operand type for set data type

b) {5,6,5,6,5,6}

c) {5,6}

d) Error as multiplication creates duplicate elements which isn’t allowed

Answer: a

Explanation: The multiplication operator isn’t valid for the set data type.

13. What will be the output of the following Python code?

>>> a={5,6,7,8}

>>> b={7,5,6,8}

>>> a==b

a) True

b) False

Answer: a

Explanation: It is possible to compare two sets and the order of elements in both the sets doesn’t matter if the values of the elements are the same.

14. What will be the output of the following Python code?

>>> a={3,4,5}

>>> b={5,6,7}

>>> a|b

a) Invalid operation

b) {3, 4, 5, 6, 7}

c) {5}

d) {3,4,6,7}

Answer: d

Explanation: The operation in the above piece of code is union operation. This operation produces a set of elements in both set a and set b.

15. Is the following Python code valid?

a={3,4,{7,5}}

print(a[2][0])

a) Yes, 7 is printed

b) Error, elements of a set can’t be printed

c) Error, subsets aren’t allowed

d) Yes, {7,5} is printed

Answer: c

Explanation: In python, elements of a set must not be mutable and sets are mutable. Thus, subsets can’t exist.

1. Which of these about a frozenset is not true?

a) Mutable data type

b) Allows duplicate values

c) Data type with unordered values

d) Immutable data type

Answer: a

Explanation: A frozenset is an immutable data type.

2. What is the syntax of the following Python code?

>>> a=frozenset(set([5,6,7]))

>>> a

a) {5,6,7}

b) frozenset({5,6,7})

c) Error, not possible to convert set into frozenset

d) Syntax error

Answer: b

Explanation: The above piece of code is the correct syntax for creating a frozenset.

3. Is the following Python code valid?

>>> a=frozenset([5,6,7])

>>> a

>>> a.add(5)

a) Yes, now a is {5,5,6,7}

b) No, frozen set is immutable

c) No, invalid syntax for add method

d) Yes, now a is {5,6,7}

Answer: b

Explanation: Since a frozen set is immutable, add method doesn’t exist for frozen method.

4. Set members must not be hashable.

a) True

b) False

Answer: b

Explanation: Set members must always be hashable.

5. What will be the output of the following Python code?

>>> a={3,4,5}

>>> a.update([1,2,3])

>>> a

a) Error, no method called update for set data type

b) {1, 2, 3, 4, 5}

c) Error, list can’t be added to set

d) Error, duplicate item present in list

Answer: b

Explanation: The method update adds elements to a set.

6. What will be the output of the following Python code?

>>> a={1,2,3}

>>> a.intersection\_update({2,3,4,5})

>>> a

a) {2,3}

b) Error, duplicate item present in list

c) Error, no method called intersection\_update for set data type

d) {1,4,5}

Answer: a

Explanation: The method intersection\_update returns a set which is an intersection of both the sets.

7. What will be the output of the following Python code?

>>> a={1,2,3}

>>> b=a

>>> b.remove(3)

>>> a

a) {1,2,3}

b) Error, copying of sets isn’t allowed

c) {1,2}

d) Error, invalid syntax for remove

Answer: c

Explanation: Any change made in b is reflected in a because b is an alias of a.

8. What will be the output of the following Python code?

>>> a={1,2,3}

>>> b=a.copy()

>>> b.add(4)

>>> a

a) {1,2,3}

b) Error, invalid syntax for add

c) {1,2,3,4}

d) Error, copying of sets isn’t allowed

Answer: a

Explanation: In the above piece of code, b is barely a copy and not an alias of a. Hence any change made in b isn’t reflected in a.

9. What will be the output of the following Python code?

>>> a={1,2,3}

>>> b=a.add(4)

>>> b

a) 0

b) {1,2,3,4}

c) {1,2,3}

d) Nothing is printed

Answer: d

Explanation: The method add returns nothing, hence nothing is printed.

10. What will be the output of the following Python code?

>>> a={1,2,3}

>>> b=frozenset([3,4,5])

>>> a-b

a) {1,2}

b) Error as difference between a set and frozenset can’t be found out

c) Error as unsupported operand type for set data type

d) frozenset({1,2})

Answer: a

Explanation: – operator gives the set of elements in set a but not in set b.

11. What will be the output of the following Python code?

>>> a={5,6,7}

>>> sum(a,5)

a) 5

b) 23

c) 18

d) Invalid syntax for sum method, too many arguments

Answer: b

Explanation: The second parameter is the start value for the sum of elements in set a. Thus, sum(a,5) = 5+(5+6+7)=23.

12. What will be the output of the following Python code?

>>> a={1,2,3}

>>> {x\*2 for x in a|{4,5}}

a) {2,4,6}

b) Error, set comprehensions aren’t allowed

c) {8, 2, 10, 4, 6}

d) {8,10}

Answer: c

Explanation: Set comprehensions are allowed.

13. What will be the output of the following Python code?

>>> a={5,6,7,8}

>>> b={7,8,9,10}

>>> len(a+b)

a) 8

b) Error, unsupported operand ‘+’ for sets

c) 6

d) Nothing is displayed

Answer: b

Explanation: Duplicate elements in a+b is eliminated and the length of a+b is computed.

14. What will be the output of the following Python code?

a={1,2,3}

b={1,2,3}

c=a.issubset(b)

print(c)

a) True

b) Error, no method called issubset() exists

c) Syntax error for issubset() method

d) False

Answer: a

Explanation: The method issubset() returns True if b is a proper subset of a.

15. Is the following Python code valid?

a={1,2,3}

b={1,2,3,4}

c=a.issuperset(b)

print(c)

a) False

b) True

c) Syntax error for issuperset() method

d) Error, no method called issuperset() exists

Answer: a

Explanation: The method issubset() returns True if b is a proper subset of a.

1. What will be the output of the following Python code?

s=set()

type(s)

a) <’set’>

b) <class ‘set’>

c) set

d) class set

Answer: b

Explanation: When we find the type of a set, the output returned is: .

2. The following Python code results in an error.

s={2, 3, 4, [5, 6]}

a) True

b) False

Answer: a

Explanation: The set data type makes use of a principle known as hashing. This means that each item in the set should be hashable. Hashable in this context means immutable. List is mutable and hence the line of code shown above will result in an error.

3. Set makes use of \_\_\_\_\_\_\_\_\_\_

Dictionary makes use of \_\_\_\_\_\_\_\_\_\_\_\_

a) keys, keys

b) key values, keys

c) keys, key values

d) key values, key values

Answer: c

Explanation: Set makes use of keys.

Dictionary makes use of key values.

4. Which of the following lines of code will result in an error?

a) s={abs}

b) s={4, ‘abc’, (1,2)}

c) s={2, 2.2, 3, ‘xyz’}

d) s={san}

Answer: d

Explanation: The line: s={san} will result in an error because ‘san’ is not defined. The line s={abs} does not result in an error because abs is a built-in function. The other sets shown do not result in an error because all the items are hashable.

5. What will be the output of the following Python code?

s={2, 5, 6, 6, 7}

s

a) {2, 5, 7}

b) {2, 5, 6, 7}

c) {2, 5, 6, 6, 7}

d) Error

Answer: b

Explanation: Duplicate values are not allowed in sets. Hence, the output of the code shown above will be a set containing the duplicate value only once. Therefore the output is: {2, 5, 6, 7}

6. Input order is preserved in sets.

a) True

b) False

Answer: b

Explanation: The input order in sets is not maintained. This is demonstrated by the code shown below:

>>> s={2, 6, 8, 1, 5}

>>> s

{8, 1, 2, 5, 6}

7. Write a list comprehension for number and its cube for:

l=[1, 2, 3, 4, 5, 6, 7, 8, 9]

a) [x\*\*3 for x in l]

b) [x^3 for x in l]

c) [x\*\*3 in l]

d) [x^3 in l]

Answer: a

Explanation: The list comprehension to print a list of cube of the numbers for the given list is: [x\*\*3 for x in l].

8. What will be the output of the following Python code?

s={1, 2, 3}

s.update(4)

s

a) {1, 2, 3, 4}

b) {1, 2, 4, 3}

c) {4, 1, 2, 3}

d) Error

Answer: d

Explanation: The code shown above will result in an error because the argument given to the function update should necessarily be an iterable. Hence if we write this function as: s.update([4]), there will be no error.

9. Which of the following functions cannot be used on heterogeneous sets?

a) pop

b) remove

c) update

d) sum

Answer: d

Explanation: The functions sum, min and max cannot be used on mixed type (heterogeneous) sets. The functions pop, remove, update etc can be used on homogenous as well as heterogeneous sets. An example of heterogeneous sets is: {‘abc’, 4, (1, 2)}

10. What will be the output of the following Python code?

s={4>3, 0, 3-3}

all(s)

any(s)

a)

True

False

b)

False

True

c)

True

True

d)

False

False

Answer: b

Explanation: The function all returns true only if all the conditions given are true. But in the example shown above, we have 0 as a value. Hence false is returned. Similarly, any returns true if any one condition is true. Since the condition 4>3 is true, true is returned.

1. Which of the following functions will return the symmetric difference between two sets, x and y?

a) x | y

b) x ^ y

c) x & y

d) x – y

Answer: b

Explanation: The function x ^ y returns the symmetric difference between the two sets x and y. This is basically an XOR operation being performed on the two sets.

2. What will be the output of the following Python code snippet?

z=set('abc$de')

'a' in z

a) True

b) False

c) No output

d) Error

Answer: a

Explanation: The code shown above is used to check whether a particular item is a part of a given set or not. Since ‘a’ is a part of the set z, the output is true. Note that this code would result in an error in the absence of the quotes.

3. What will be the output of the following Python code snippet?

z=set('abc')

z.add('san')

z.update(set(['p', 'q']))

z

a) {‘abc’, ‘p’, ‘q’, ‘san’}

b) {‘a’, ‘b’, ‘c’, [‘p’, ‘q’], ‘san}

c) {‘a’, ‘c’, ‘c’, ‘p’, ‘q’, ‘s’, ‘a’, ‘n’}

d) {‘a’, ‘b’, ‘c’, ‘p’, ‘q’, ‘san’}

Answer: d

Explanation: The code shown first adds the element ‘san’ to the set z. The set z is then updated and two more elements, namely, ‘p’ and ‘q’ are added to it. Hence the output is: {‘a’, ‘b’, ‘c’, ‘p’, ‘q’, ‘san’}

4. What will be the output of the following Python code snippet?

s=set([1, 2, 3])

s.union([4, 5])

s|([4, 5])

a)

{1, 2, 3, 4, 5}

{1, 2, 3, 4, 5}

b)

Error

{1, 2, 3, 4, 5}

c)

{1, 2, 3, 4, 5}

Error

d)

Error

Error

Answer: c

Explanation: The first function in the code shown above returns the set {1, 2, 3, 4, 5}. This is because the method of the function union allows any iterable. However the second function results in an error because of unsupported data type, that is list and set.

5. What will be the output of the following Python code snippet?

for x in set('pqr'):

print(x\*2)

a)

pp

qq

rr

b)

pqr

pqr

c) ppqqrr

d) pqrpqr

Answer: a

Explanation: The code shown above prints each element of the set twice separately. Hence the output of this code is:

pp

qq

rr

6. What will be the output of the following Python code snippet?

{a\*\*2 for a in range(4)}

a) {1, 4, 9, 16}

b) {0, 1, 4, 9, 16}

c) Error

d) {0, 1, 4, 9}

Answer: d

Explanation: The code shown above returns a set containing the square of values in the range 0-3, that is 0, 1, 2 and 3. Hence the output of this line of code is: {0, 1, 4, 9}.

7. What will be the output of the following Python function?

{x for x in 'abc'}

{x\*3 for x in 'abc'}

a)

{abc}

aaa

bbb

ccc

b)

abc

abc abc abc

c)

{‘a’, ‘b’, ‘c’}

{‘aaa’, ‘bbb’, ‘ccc’}

d)

{‘a’, ‘b’, ‘c’}

abc

abc

abc

Answer: c

Explanation: The first function prints each element of the set separately, hence the output is: {‘a’, ‘b’, ‘c’}. The second function prints each element of the set thrice, contained in a new set. Hence the output of the second function is: {‘aaa’, ‘bbb’, ‘ccc’}. (Note that the order may not be the same)

8. The output of the following code is: class<’set’>.

type({})

a) True

b) False

Answer: b

Explanation: The output of the line of code shown above is: class<’dict’>. This is because {} represents an empty dictionary, whereas set() initializes an empty set. Hence the statement is false.

9. What will be the output of the following Python code snippet?

a=[1, 4, 3, 5, 2]

b=[3, 1, 5, 2, 4]

a==b

set(a)==set(b)

a)

True

False

b)

False

False

c)

False

True

d)

True

True

Answer: c

Explanation: In the code shown above, when we check the equality of the two lists, a and b, we get the output false. This is because of the difference in the order of elements of the two lists. However, when these lists are converted to sets and checked for equality, the output is true. This is known as order-neutral equality. Two sets are said to be equal if and only if they contain exactly the same elements, regardless of order.

10. What will be the output of the following Python code snippet?

l=[1, 2, 4, 5, 2, 'xy', 4]

set(l)

l

a)

{1, 2, 4, 5, 2, ‘xy’, 4}

[1, 2, 4, 5, 2, ‘xy’, 4]

b)

{1, 2, 4, 5, ‘xy’}

[1, 2, 4, 5, 2, ‘xy’, 4]

c)

{1, 5, ‘xy’}

[1, 5, ‘xy’]

d)

{1, 2, 4, 5, ‘xy’}

[1, 2, 4, 5, ‘xy’]

Answer: b

Explanation: In the code shown above, the function set(l) converts the given list into a set. When this happens, all the duplicates are automatically removed. Hence the output is: {1, 2, 4, 5, ‘xy’}. On the other hand, the list l remains unchanged. Therefore the output is: [1, 2, 4, 5, 2, ‘xy’, 4].

Note that the order of the elements may not be the same.

1. What will be the output of the following Python code?

s1={3, 4}

s2={1, 2}

s3=set()

i=0

j=0

for i in s1:

for j in s2:

s3.add((i,j))

i+=1

j+=1

print(s3)

a) {(3, 4), (1, 2)}

b) Error

c) {(4, 2), (3, 1), (4, 1), (5, 2)}

d) {(3, 1), (4, 2)}

Answer: c

Explanation: The code shown above finds the Cartesian product of the two sets, s1 and s2. The Cartesian product of these two sets is stored in a third set, that is, s3. Hence the output of this code is: {(4, 2), (3, 1), (4, 1), (5, 2)}.

2. The \_\_\_\_\_\_\_\_\_\_\_\_ function removes the first element of a set and the last element of a list.

a) remove b) pop c) discard d) dispose

Answer: b

Explanation: The function pop removes the first element when used on a set and the last element when used to a list.

3. The difference between the functions discard and remove is that:

a) Discard removes the last element of the set whereas remove removes the first element of the set

b) Discard throws an error if the specified element is not present in the set whereas remove does not throw an error in case of absence of the specified element

c) Remove removes the last element of the set whereas discard removes the first element of the set

d) Remove throws an error if the specified element is not present in the set whereas discard does not throw an error in case of absence of the specified element

Answer: d

Explanation: The function remove removes the element if it is present in the set. If the element is not present, it throws an error. The function discard removes the element if it is present in the set. If the element is not present, no action is performed (Error is not thrown).

4. What will be the output of the following Python code?

s1={1, 2, 3}

s2={3, 4, 5, 6}

s1.difference(s2)

s2.difference(s1)

a)

{1, 2}

{4, 5, 6}

b)

{1, 2}

{1, 2}

c)

{4, 5, 6}

{1, 2}

d)

{4, 5, 6}

{4, 5, 6}

Answer: a

Explanation: The function s1.difference(s2) returns a set containing the elements which are present in the set s1 but not in the set s2. Similarly, the function s2.difference(s1) returns a set containing elements which are present in the set s2 but not in the set s1. Hence the output of the code shown above will be:

{1, 2}

{4, 5, 6}.

5. What will be the output of the following Python code?

s1={1, 2, 3}

s2={4, 5, 6}

s1.isdisjoint(s2)

s2.isdisjoint(s1)

a)

True

False

b)

False

True

c)

True

True

d)

False

False

Answer: c

Explanation: The function isdisjoint returns true the two sets in question are disjoint, that is if they do not have even a single element in common. The two sets s1 and s2 do not have any elements in common, hence true is returned in both the cases.

6. If we have two sets, s1 and s2, and we want to check if all the elements of s1 are present in s2 or not, we can use the function:

a) s2.issubset(s1) b) s2.issuperset(s1)

c) s1.issuperset(s2) d) s1.isset(s2)

Answer: b

Explanation: Since we are checking whether all the elements present in the set s1 are present in the set s2. This means that s1 is the subset and s1 is the superset. Hence the function to be used is: s2.issuperset(s1). This operation can also be performed by the function: s1.issubset(s2).

7. What will be the output of the following Python code?

s1={1, 2, 3, 8}

s2={3, 4, 5, 6}

s1|s2

s1.union(s2)

a)

{3}

{1, 2, 3, 4, 5, 6, 8}

b)

{1, 2, 4, 5, 6, 8}

{1, 2, 4, 5, 6, 8}

c)

{3}

{3}

d)

{1, 2, 3, 4, 5, 6, 8}

{1, 2, 3, 4, 5, 6, 8}

Answer: d

Explanation: The function s1|s2 as well as the function s1.union(s2) returns a union of the two sets s1 and s2. Hence the output of both of these functions is: {1, 2, 3, 4, 5, 6, 8}.

8. What will be the output of the following Python code?

a=set('abc')

b=set('def')

b.intersection\_update(a)

a

b

a)

set()

(‘e’, ‘d’, ‘f’}

b)

{}

{}

c)

{‘b’, ‘c’, ‘a’}

set()

d)

set()

set()

Answer: c

Explanation: The function b.intersection\_update(a) puts those elements in the set b which are common to both the sets a and b. The set a remains as it is. Since there are no common elements between the sets a and b, the output is:

‘b’, ‘c’, ‘a’}

set().

9. What will be the output of the following Python code, if s1= {1, 2, 3}?

s1.issubset(s1)

a) True b) Error c) No output d) False

Answer: a

Explanation: Every set is a subset of itself and hence the output of this line of code is true.

10. What will be the output of the following Python code?

x=set('abcde')

y=set('xyzbd')

x.difference\_update(y)

x

y

a)

{‘a’, ‘b’, ‘c’, ‘d’, ‘e’}

{‘x’, ‘y’, ‘z’}

b)

{‘a’, ‘c’, ‘e’}

{‘x’, ‘y’, ‘z’, ‘b’, ‘d’}

c)

{‘b’, ‘d’}

{‘b’, ‘d’}

d)

{‘a’, ‘c’, ‘e’}

{‘x’, ‘y’, ‘z’}

Answer: b

Explanation: The function x.difference\_update(y) removes all the elements of the set y from the set x. Hence the output of the code is:

{‘a’, ‘c’, ‘e’}

{‘x’, ‘y’, ‘z’, ‘b’, ‘d’}.

# Dictionary – 1

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Dictionaries”.

1. Which of the following statements create a dictionary?

a) d = {}

b) d = {“john”:40, “peter”:45}

c) d = {40:”john”, 45:”peter”}

d) All of the mentioned

Answer: d

Explanation: Dictionaries are created by specifying keys and values.

2. What will be the output of the following Python code snippet?

d = {"john":40, "peter":45}

a) “john”, 40, 45, and “peter”

b) “john” and “peter”

c) 40 and 45

d) d = (40:”john”, 45:”peter”)

Answer: b

Explanation: Dictionaries appear in the form of keys and values.

3. What will be the output of the following Python code snippet?

d = {"john":40, "peter":45}

"john" in d

a) True

b) False

c) None

d) Error

Answer: a

Explanation: In can be used to check if the key is int dictionary.

4. What will be the output of the following Python code snippet?

d1 = {"john":40, "peter":45}

d2 = {"john":466, "peter":45}

d1 == d2

a) True

b) False

c) None

d) Error

Answer: b

Explanation: If d2 was initialized as d2 = d1 the answer would be true.

5. What will be the output of the following Python code snippet?

d1 = {"john":40, "peter":45}

d2 = {"john":466, "peter":45}

d1 > d2

a) True

b) False

c) Error

d) None

Answer: c

Explanation: Arithmetic > operator cannot be used with dictionaries.

6. What will be the output of the following Python code snippet?

d = {"john":40, "peter":45}

d["john"]

a) 40

b) 45

c) “john”

d) “peter”

Answer: a

Explanation: Execute in the shell to verify.

7. Suppose d = {“john”:40, “peter”:45}, to delete the entry for “john” what command do we use?

a) d.delete(“john”:40)

b) d.delete(“john”)

c) del d[“john”]

d) del d(“john”:40)

Answer: c

Explanation: Execute in the shell to verify.

8. Suppose d = {“john”:40, “peter”:45}. To obtain the number of entries in dictionary which command do we use?

a) d.size()

b) len(d)

c) size(d)

d) d.len()

Answer: b

Explanation: Execute in the shell to verify.

9. What will be the output of the following Python code snippet?

d = {"john":40, "peter":45}

print(list(d.keys()))

a) [“john”, “peter”]

b) [“john”:40, “peter”:45]

c) (“john”, “peter”)

d) (“john”:40, “peter”:45)

Answer: a

Explanation: The output of the code shown above is a list containing only keys of the dictionary d, in the form of a list.

10. Suppose d = {“john”:40, “peter”:45}, what happens when we try to retrieve a value using the expression d[“susan”]?

a) Since “susan” is not a value in the set, Python raises a KeyError exception

b) It is executed fine and no exception is raised, and it returns None

c) Since “susan” is not a key in the set, Python raises a KeyError exception

d) Since “susan” is not a key in the set, Python raises a syntax error

Answer: c

Explanation: Execute in the shell to verify.

1. Which of these about a dictionary is false?

a) The values of a dictionary can be accessed using keys

b) The keys of a dictionary can be accessed using values

c) Dictionaries aren’t ordered

d) Dictionaries are mutable

Answer: b

Explanation: The values of a dictionary can be accessed using keys but the keys of a dictionary can’t be accessed using values.

2. Which of the following is not a declaration of the dictionary?

a) {1: ‘A’, 2: ‘B’}

b) dict([[1,”A”],[2,”B”]])

c) {1,”A”,2”B”}

d) { }

Answer: c

Explanation: Option c is a set, not a dictionary.

3. What will be the output of the following Python code snippet?

a={1:"A",2:"B",3:"C"}

for i,j in a.items():

print(i,j,end=" ")

a) 1 A 2 B 3 C

b) 1 2 3

c) A B C

d) 1:”A” 2:”B” 3:”C”

Answer: a

Explanation: In the above code, variables i and j iterate over the keys and values of the dictionary respectively.

4. What will be the output of the following Python code snippet?

a={1:"A",2:"B",3:"C"}

print(a.get(1,4))

a) 1

b) A

c) 4

d) Invalid syntax for get method

Answer: b

Explanation: The get() method returns the value of the key if the key is present in the dictionary and the default value(second parameter) if the key isn’t present in the dictionary.

5. What will be the output of the following Python code snippet?

a={1:"A",2:"B",3:"C"}

print(a.get(5,4))

a) Error, invalid syntax

b) A

c) 5

d) 4

Answer: d

Explanation: The get() method returns the default value(second parameter) if the key isn’t present in the dictionary.

6. What will be the output of the following Python code snippet?

a={1:"A",2:"B",3:"C"}

print(a.setdefault(3))

a) {1: ‘A’, 2: ‘B’, 3: ‘C’}

b) C

c) {1: 3, 2: 3, 3: 3}

d) No method called setdefault() exists for dictionary

Answer: b

Explanation: setdefault() is similar to get() but will set dict[key]=default if key is not already in the dictionary.

7. What will be the output of the following Python code snippet?

a={1:"A",2:"B",3:"C"}

a.setdefault(4,"D")

print(a)

a) {1: ‘A’, 2: ‘B’, 3: ‘C’, 4: ‘D’}

b) None

c) Error

d) [1,3,6,10]

Answer: a

Explanation: setdefault() will set dict[key]=default if key is not already in the dictionary.

8. What will be the output of the following Python code?

a={1:"A",2:"B",3:"C"}

b={4:"D",5:"E"}

a.update(b)

print(a)

a) {1: ‘A’, 2: ‘B’, 3: ‘C’}

b) Method update() doesn’t exist for dictionaries

c) {1: ‘A’, 2: ‘B’, 3: ‘C’, 4: ‘D’, 5: ‘E’}

d) {4: ‘D’, 5: ‘E’}

Answer: c

Explanation: update() method adds dictionary b’s key-value pairs to dictionary a. Execute in python shell to verify.

9. What will be the output of the following Python code?

a={1:"A",2:"B",3:"C"}

b=a.copy()

b[2]="D"

print(a)

a) Error, copy() method doesn’t exist for dictionaries

b) {1: ‘A’, 2: ‘B’, 3: ‘C’}

c) {1: ‘A’, 2: ‘D’, 3: ‘C’}

d) “None” is printed

Answer: b

Explanation: Changes made in the copy of the dictionary isn’t reflected in the original one.

10. What will be the output of the following Python code?

a={1:"A",2:"B",3:"C"}

a.clear()

print(a)

a) None

b) { None:None, None:None, None:None}

c) {1:None, 2:None, 3:None}

d) { }

Answer: d

Explanation: The clear() method clears all the key-value pairs in the dictionary.

11. Which of the following isn’t true about dictionary keys?

a) More than one key isn’t allowed

b) Keys must be immutable

c) Keys must be integers

d) When duplicate keys encountered, the last assignment wins

Answer: c

Explanation: Keys of a dictionary may be any data type that is immutable.

12. What will be the output of the following Python code?

a={1:5,2:3,3:4}

a.pop(3)

print(a)

a) {1: 5}

b) {1: 5, 2: 3}

c) Error, syntax error for pop() method

d) {1: 5, 3: 4}

Answer: b

Explanation: pop() method removes the key-value pair for the key mentioned in the pop() method.

13. What will be the output of the following Python code?

a={1:5,2:3,3:4}

print(a.pop(4,9))

a) 9

b) 3

c) Too many arguments for pop() method

d) 4

Answer: a

Explanation: pop() method returns the value when the key is passed as an argument and otherwise returns the default value(second argument) if the key isn’t present in the dictionary.

14. What will be the output of the following Python code?

a={1:"A",2:"B",3:"C"}

for i in a:

print(i,end=" ")

a) 1 2 3

b) ‘A’ ‘B’ ‘C’

c) 1 ‘A’ 2 ‘B’ 3 ‘C’

d) Error, it should be: for i in a.items():

Answer: a

Explanation: The variable i iterates over the keys of the dictionary and hence the keys are printed.

15. What will be the output of the following Python code?

>>> a={1:"A",2:"B",3:"C"}

>>> a.items()

a) Syntax error

b) dict\_items([(‘A’), (‘B’), (‘C’)])

c) dict\_items([(1,2,3)])

d) dict\_items([(1, ‘A’), (2, ‘B’), (3, ‘C’)])

Answer: d

Explanation: The method items() returns list of tuples with each tuple having a key-value pair.

1. Which of the statements about dictionary values if false?

a) More than one key can have the same value

b) The values of the dictionary can be accessed as dict[key]

c) Values of a dictionary must be unique

d) Values of a dictionary can be a mixture of letters and numbers

Answer: c

Explanation: More than one key can have the same value.

2. What will be the output of the following Python code snippet?

>>> a={1:"A",2:"B",3:"C"}

>>> del a

a) method del doesn’t exist for the dictionary b) del deletes the values in the dictionary

c) del deletes the entire dictionary d) del deletes the keys in the dictionary

Answer: c

Explanation: del deletes the entire dictionary and any further attempt to access it will throw an error.

3. If a is a dictionary with some key-value pairs, what does a.popitem() do?

a) Removes an arbitrary element

b) Removes all the key-value pairs

c) Removes the key-value pair for the key given as an argument

d) Invalid method for dictionary

Answer: a

Explanation: The method popitem() removes a random key-value pair.

4. What will be the output of the following Python code snippet?

total={}

def insert(items):

if items in total:

total[items] += 1

else:

total[items] = 1

insert('Apple')

insert('Ball')

insert('Apple')

print (len(total))

a) 3 b) 1 c) 2 d) 0

Answer: c

Explanation: The insert() function counts the number of occurrences of the item being inserted into the dictionary. There are only 2 keys present since the key ‘Apple’ is repeated. Thus, the length of the dictionary is 2.

5. What will be the output of the following Python code snippet?

a = {}

a[1] = 1

a['1'] = 2

a[1]=a[1]+1

count = 0

for i in a:

count += a[i]

print(count)

a) 1 b) 2 c) 4 d) Error, the keys can’t be a mixture of letters and numbers

Answer: c

Explanation: The above piece of code basically finds the sum of the values of keys.

6. What will be the output of the following Python code snippet?

numbers = {}

letters = {}

comb = {}

numbers[1] = 56

numbers[3] = 7

letters[4] = 'B'

comb['Numbers'] = numbers

comb['Letters'] = letters

print(comb)

a) Error, dictionary in a dictionary can’t exist

b) ‘Numbers’: {1: 56, 3: 7}

c) {‘Numbers’: {1: 56}, ‘Letters’: {4: ‘B’}}

d) {‘Numbers’: {1: 56, 3: 7}, ‘Letters’: {4: ‘B’}}

Answer: d

Explanation: Dictionary in a dictionary can exist.

7. What will be the output of the following Python code snippet?

test = {1:'A', 2:'B', 3:'C'}

test = {}

print(len(test))

a) 0 b) None c) 3 d) An exception is thrown

Answer: a

Explanation: In the second line of code, the dictionary becomes an empty dictionary. Thus, length=0.

8. What will be the output of the following Python code snippet?

test = {1:'A', 2:'B', 3:'C'}

del test[1]

test[1] = 'D'

del test[2]

print(len(test))

a) 0

b) 2

c) Error as the key-value pair of 1:’A’ is already deleted

d) 1

Answer: b

Explanation: After the key-value pair of 1:’A’ is deleted, the key-value pair of 1:’D’ is added.

9. What will be the output of the following Python code snippet?

a = {}

a[1] = 1

a['1'] = 2

a[1.0]=4

count = 0

for i in a:

count += a[i]

print(count)

a) An exception is thrown b) 3 c) 6 d) 2

Answer: c

Explanation: The value of key 1 is 4 since 1 and 1.0 are the same. Then, the function count() gives the sum of all the values of the keys (2+4).

10. What will be the output of the following Python code snippet?

a={}

a['a']=1

a['b']=[2,3,4]

print(a)

a) Exception is thrown

b) {‘b’: [2], ‘a’: 1}

c) {‘b’: [2], ‘a’: [3]}

d) {‘b’: [2, 3, 4], ‘a’: 1}

Answer: d

Explanation: Mutable members can be used as the values of the dictionary but they cannot be used as the keys of the dictionary.

11. What will be the output of the following Python code snippet?

>>>import collections

>>> a=collections.Counter([1,1,2,3,3,4,4,4])

>>> a

a) {1,2,3,4}

b) Counter({4, 1, 3, 2})

c) Counter({4: 3, 1: 2, 3: 2, 2: 1})

d) {4: 3, 1: 2, 3: 2, 2: 1}

Answer: c

Explanation: The statement a=collections.OrderedDict() generates a dictionary with the number as the key and the count of times the number appears as the value.

12. What will be the output of the following Python code snippet?

>>>import collections

>>> b=collections.Counter([2,2,3,4,4,4])

>>> b.most\_common(1)

a) Counter({4: 3, 2: 2, 3: 1})

b) {3:1}

c) {4:3}

d) [(4, 3)]

Answer: d

Explanation: The most\_common() method returns the n number key-value pairs where the value is the most recurring.

13. What will be the output of the following Python code snippet?

>>>import collections

>>> b=collections.Counter([2,2,3,4,4,4])

>>> b.most\_common(1)

a) Counter({4: 3, 2: 2, 3: 1})

b) {3:1}

c) {4:3}

d) [(4, 3)]

Answer: d

Explanation: The most\_common() method returns the n number key-value pairs where the value is the most recurring.

14. What will be the output of the following Python code snippet?

>>> import collections

>>> a=collections.Counter([2,2,3,3,3,4])

>>> b=collections.Counter([2,2,3,4,4])

>>> a|b

a) Counter({3: 3, 2: 2, 4: 2})

b) Counter({2: 2, 3: 1, 4: 1})

c) Counter({3: 2})

d) Counter({4: 1})

Answer: a

Explanation: a|b returns the pair of keys and the highest recurring value.

15. What will be the output of the following Python code snippet?

>>> import collections

>>> a=collections.Counter([3,3,4,5])

>>> b=collections.Counter([3,4,4,5,5,5])

>>> a&b

a) Counter({3: 12, 4: 1, 5: 1})

b) Counter({3: 1, 4: 1, 5: 1})

c) Counter({4: 2})

d) Counter({5: 1})

Answer: b

Explanation: a&b returns the pair of keys and the lowest recurring value.

1. The following Python code is invalid.

class demo(dict):

def \_\_test\_\_(self,key):

return []

a = demo()

a['test'] = 7

print(a)

a) True

b) False

Answer: b

Explanation: The output of the code is: {‘test’:7}.

2. What will be the output of the following Python code?

count={}

count[(1,2,4)] = 5

count[(4,2,1)] = 7

count[(1,2)] = 6

count[(4,2,1)] = 2

tot = 0

for i in count:

tot=tot+count[i]

print(len(count)+tot)

a) 25

b) 17

c) 16

d) Tuples can’t be made keys of a dictionary

Answer: c

Explanation: Tuples can be made keys of a dictionary. Length of the dictionary is 3 as the value of the key (4,2,1) is modified to 2. The value of the variable tot is 5+6+2=13.

3. What will be the output of the following Python code?

a={}

a[2]=1

a[1]=[2,3,4]

print(a[1][1])

a) [2,3,4]

b) 3

c) 2

d) An exception is thrown

Answer: b

Explanation: Now, a={1:[2,3,4],2:1} . a[1][1] refers to second element having key 1.

4. What will be the output of the following Python code?

>>> a={'B':5,'A':9,'C':7}

>>> sorted(a)

a) [‘A’,’B’,’C’]

b) [‘B’,’C’,’A’]

c) [5,7,9]

d) [9,5,7]

Answer: a

Explanation: Return a new sorted list of keys in the dictionary.

5. What will be the output of the following Python code?

>>> a={i: i\*i for i in range(6)}

>>> a

a) Dictionary comprehension doesn’t exist

b) {0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6:36}

c) {0: 0, 1: 1, 4: 4, 9: 9, 16: 16, 25: 25}

d) {0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

Answer: d

Explanation: Dictionary comprehension is implemented in the above piece of code.

6. What will be the output of the following Python code?

>>> a={}

>>> a.fromkeys([1,2,3],"check")

a) Syntax error

b) {1:”check”,2:”check”,3:”check”}

c) “check”

d) {1:None,2:None,3:None}

Answer: b

Explanation: The dictionary takes values of keys from the list and initializes it to the default value (value given in the second parameter). Execute in Python shell to verify.

7. What will be the output of the following Python code?

>>> b={}

>>> all(b)

a) { }

b) False

c) True

d) An exception is thrown

Answer: c

Explanation: Function all() returns True if all keys of the dictionary are true or if the dictionary is empty.

8. If b is a dictionary, what does any(b) do?

a) Returns True if any key of the dictionary is true

b) Returns False if dictionary is empty

c) Returns True if all keys of the dictionary are true

d) Method any() doesn’t exist for dictionary

Answer: a

Explanation: Method any() returns True if any key of the dictionary is true and False if the dictionary is empty.

9. What will be the output of the following Python code?

>>> a={"a":1,"b":2,"c":3}

>>> b=dict(zip(a.values(),a.keys()))

>>> b

a) {‘a’: 1, ‘b’: 2, ‘c’: 3}

b) An exception is thrown

c) {‘a’: ‘b’: ‘c’: }

d) {1: ‘a’, 2: ‘b’, 3: ‘c’}

Answer: d

Explanation: The above piece of code inverts the key-value pairs in the dictionary.

10. What will be the output of the following Python code?

>>> a={i: 'A' + str(i) for i in range(5)}

>>> a

a) An exception is thrown

b) {0: ‘A0’, 1: ‘A1’, 2: ‘A2’, 3: ‘A3’, 4: ‘A4’}

c) {0: ‘A’, 1: ‘A’, 2: ‘A’, 3: ‘A’, 4: ‘A’}

d) {0: ‘0’, 1: ‘1’, 2: ‘2’, 3: ‘3’, 4: ‘4’}

Answer: b

Explanation: Dictionary comprehension and string concatenation is implemented in the above piece of code.

11. What will be the output of the following Python code?

>>> a=dict()

>>> a[1]

a) An exception is thrown since the dictionary is empty

b) ‘ ‘

c) 1

d) 0

Answer: a

Explanation: The values of a dictionary can be accessed through the keys only if the keys exist in the dictionary.

12. What will be the output of the following Python code?

>>> import collections

>>> a=dict()

>>> a=collections.defaultdict(int)

>>> a[1]

a) 1

b) 0

c) An exception is thrown

d) ‘ ‘

Answer: b

Explanation: The statement a=collections.defaultdict(int) gives the default value of 0

(since int data type is given within the parenthesis) even if the keys don’t exist in the dictionary.

13. What will be the output of the following Python code?

>>> import collections

>>> a=dict()

>>> a=collections.defaultdict(str)

>>> a['A']

a) An exception is thrown since the dictionary is empty

b) ‘ ‘

c) ‘A’

d) 0

Answer: b

Explanation: The statement a=collections.defaultdict(str) gives the default value of ‘ ‘ even if the keys don’t exist in the dictionary.

14. What will be the output of the following Python code?

>>> import collections

>>> b=dict()

>>> b=collections.defaultdict(lambda: 7)

>>> b[4]

a) 4

b) 0

c) An exception is thrown

d) 7

Answer: d

Explanation: The statement a=collections.defaultdict(lambda: x) gives the default value of x even if the keys don’t exist in the dictionary.

15. What will be the output of the following Python code?

>>> import collections

>>> a=collections.OrderedDict((str(x),x) for x in range(3))

>>> a

a) {‘2’:2, ‘0’:0, ‘1’:1}

b) OrderedDict([(‘0’, 0), (‘1’, 1), (‘2’, 2)])

c) An exception is thrown

d) ‘ ‘

Answer: b

Explanation: The line of code a=collections.OrderedDict() generates a dictionary satisfying the conditions given within the parenthesis and in an ascending order of the keys.

# Built-in Functions – 1

<https://www.sanfoundry.com/python-questions-answers-built-in-functions-1/>

1. Which of the following functions is a built-in function in python?

a) seed()

b) sqrt()

c) factorial()

d) print()

Answer: d

Explanation: The function seed is a function which is present in the random module. The functions sqrt and factorial are a part of the math module. The print function is a built-in function which prints a value directly to the system output.

2. What will be the output of the following Python expression?

round(4.576)

a) 4.5

b) 5

c) 4

d) 4.6

Answer: b

Explanation: This is a built-in function which rounds a number to give precision in decimal digits. In the above case, since the number of decimal places has not been specified, the decimal number is rounded off to a whole number. Hence the output will be 5.

3. The function pow(x,y,z) is evaluated as:

a) (x\*\*y)\*\*z

b) (x\*\*y) / z

c) (x\*\*y) % z

d) (x\*\*y)\*z

Answer: c

Explanation: The built-in function pow() can accept two or three arguments. When it takes in two arguments, they are evaluated as x\*\*y. When it takes in three arguments, they are evaluated as (x\*\*y)%z.

4. What will be the output of the following Python function?

all([2,4,0,6])

a) Error

b) True

c) False

d) 0

Answer: c

Explanation: The function all returns false if any one of the elements of the iterable is zero and true if all the elements of the iterable are non zero. Hence the output of this function will be false.

5. What will be the output of the following Python expression?

round(4.5676,2)?

a) 4.5

b) 4.6

c) 4.57

d) 4.56

Answer: c

Explanation: The function round is used to round off the given decimal number to the specified decimal places. In this case, the number should be rounded off to two decimal places. Hence the output will be 4.57.

6. What will be the output of the following Python function?

any([2>8, 4>2, 1>2])

a) Error

b) True

c) False

d) 4>2

Answer: b

Explanation: The built-in function any() returns true if any or more of the elements of the iterable is true (non zero), If all the elements are zero, it returns false.

7. What will be the output of the following Python function?

import math

abs(math.sqrt(25))

a) Error

b) -5

c) 5

d) 5.0

Answer: d

Explanation: The abs() function prints the absolute value of the argument passed. For example: abs(-5)=5. Hence, in this case we get abs(5.0)=5.0.

8. What will be the output of the following Python function?

sum(2,4,6)

sum([1,2,3])

a) Error, 6

b) 12, Error

c) 12, 6

d) Error, Error

Answer: a

Explanation: The first function will result in an error because the function sum() is used to find the sum of iterable numbers. Hence the outcomes will be Error and 6 respectively.

9. What will be the output of the following Python function?

all(3,0,4.2)

a) True

b) False

c) Error

d) 0

Answer: c

Explanation: The function all() returns ‘True’ if any one or more of the elements of the iterable are non zero. In the above case, the values are not iterable, hence an error is thrown.

10. What will be the output of the following Python function?

min(max(False,-3,-4), 2,7)

a) 2

b) False

c) -3

d) -4

Answer: b

Explanation: The function max() is being used to find the maximum value from among -3, -4 and false. Since false amounts to the value zero, hence we are left with min(0, 2, 7) Hence the output is 0 (false).

1. What will be the output of the following Python functions?

chr(‘97’)

chr(97)

a)

a

Error

b)

‘a’

a

c)

Error

a

d)

Error

Error

Answer: c

Explanation: The built-in function chr() returns the alphabet corresponding to the value given as an argument. This function accepts only integer type values. In the first function, we have passed a string. Hence the first function throws an error.

2. What will be the output of the following Python function?

complex(1+2j)

a) Error

b) 1

c) 2j

d) 1+2j

Answer: d

Explanation: The built-in function complex() returns the argument in a complex form. Hence the output of the function shown above will be 1+2j.

3. What is the output of the function complex()?

a) 0j

b) 0+0j

c) 0

d) Error

Answer: a

Explanation: The complex function returns 0j if both of the arguments are omitted, that is, if the function is in the form of complex() or complex(0), then the output will be 0j.

4. The function divmod(a,b), where both ‘a’ and ‘b’ are integers is evaluated as:

a) (a%b, a//b)

b) (a//b, a%b)

c) (a//b, a\*b)

d) (a/b, a%b)

Answer: b

Explanation: The function divmod(a,b) is evaluated as a//b, a%b, if both ‘a’ and ‘b’ are integers.

5. What will be the output of the following Python function?

divmod(10.5,5)

divmod(2.4,1.2)

a)

(2.00, 0.50)

(2.00, 0.00)

b)

(2, 0.5)

(2, 0)

c)

(2.0, 0.5)

(2.0, 0.0)

d)

(2, 0.5)

(2)

Answer: c

Explanation: See python documentation for the function divmod.

6. The function complex(‘2-3j’) is valid but the function complex(‘2 – 3j’) is invalid.

a) True b) False

Answer: a

Explanation: When converting from a string, the string must not contain any blank spaces around the + or – operator. Hence the function complex(‘2 – 3j’) will result in an error.

7. What will be the output of the following Python function?

list(enumerate([2, 3]))

a) Error

b) [(1, 2), (2, 3)]

c) [(0, 2), (1, 3)]

d) [(2, 3)]

Answer: c

Explanation: The built-in function enumerate() accepts an iterable as an argument. The function shown in the above case returns containing pairs of the numbers given, starting from 0. Hence the output will be: [(0, 2), (1,3)].

8. What will be the output of the following Python functions?

x=3

eval('x^2')

a) Error

b) 1

c) 9

d) 6

Answer: b

Explanation: The function eval is use to evaluate the expression that it takes as an argument. In the above case, the eval() function is used to perform XOR operation between 3 and 2. Hence the output is 1.

9. What will be the output of the following Python functions?

float('1e-003')

float('2e+003')

a)

3.00

300

b)

0.001

2000.0

c)

0.001

200

d)

Error

2003

Answer: b

Explanation: The output of the first function will be 0.001 and that of the second function will be 2000.0. The first function created a floating point number up to 3 decimal places and the second function adds 3 zeros after the given number.

10. Which of the following functions does not necessarily accept only iterables as arguments?

a) enumerate()

b) all()

c) chr()

d) max()

Answer: c

Explanation: The functions enumerate(), all() and max() accept iterables as arguments whereas the function chr() throws an error on receiving an iterable as an argument. Also note that the function chr() accepts only integer values.

1. Which of the following functions accepts only integers as arguments?

a) ord() b) min() c) chr() d) any()

Answer: c

Explanation: The function chr() accepts only integers as arguments. The function ord() accepts only strings. The functions min() and max() can accept floating point as well as integer arguments.

2. Suppose there is a list such that: l=[2,3,4]. If we want to print this list in reverse order, which of the following methods should be used?

a) reverse(l) b) list(reverse[(l)]) c) reversed(l) d) list(reversed(l))

Answer: d

Explanation: The built-in function reversed() can be used to reverse the elements of a list. This function accepts only an iterable as an argument. To print the output in the form of a list, we use: list(reversed(l)). The output will be: [4,3,2].

3. What will be the output of the following Python function?

float(' -12345\n')

(Note that the number of blank spaces before the number is 5)

a)   -12345.0 (5 blank spaces before the number)

b) -12345.0

c) Error

d) -12345.000000000…. (infinite decimal places)

Answer: b

Explanation: The function float() will remove all the blank spaces and convert the integer to a floating point number. Hence the output will be: -12345.0.

4. What will be the output of the following Python function?

ord(65)

ord(‘A’)

a)

A

65

b)

Error

65

c)

A

Error

d)

Error

Error

Answer: b

Explanation: The built-in function ord() is used to return the ASCII value of the alphabet passed to it as an argument. Hence the first function results in an error and the output of the second function is 65.

5. What will be the output of the following Python function?

float(‘-infinity’)

float(‘inf’)

a)

–inf

inf

b)

–infinity

inf

c)

Error

Error

d)

Error

Junk value

Answer: a

Explanation: The output of the first function will be –inf and that of the second function will be inf.

6. Which of the following functions will not result in an error when no arguments are passed to it?

a) min() b) divmod() c) all() d) float()

Answer: d

Explanation: The built-in functions min(), max(), divmod(), ord(), any(), all() etc throw an error when no arguments are passed to them. However there are some built-in functions like float(), complex() etc which do not throw an error when no arguments are passed to them. The output of float() is 0.0.

7. What will be the output of the following Python function?

hex(15)

a) f

b) 0xF

c) 0Xf

d) 0xf

Answer: d

Explanation: The function hex() is used to convert the given argument into its hexadecimal representation, in lower case. Hence the output of the function hex(15) is 0xf.

8. Which of the following functions does not throw an error?

a) ord() b) ord(‘ ‘) c) ord(”) d) ord(“”)

Answer: b

Explanation: The function ord() accepts a character. Hence ord(), ord(”) and ord(“”) throw errors. However the function ord(‘ ‘) does not throw an error because in this case, we are actually passing a blank space as an argument. The output of ord(‘ ‘) is 32 (ASCII value corresponding to blank space).

9. What will be the output of the following Python function?

len(["hello",2, 4, 6])

a) 4 b) 3 c) Error d) 6

Answer: a

Explanation: The function len() returns the length of the number of elements in the iterable. Therefore the output of the function shown above is 4.

10. What will be the output of the following Python function?

oct(7)

oct(‘7’)

a)

Error

07

b)

0o7

Error

c)

0o7

Error

d)

07

0o7

Answer: c

Explanation: The function oct() is used to convert its argument into octal form. This function does not accept strings. Hence the second function results in an error while the output of the first function is 0o7.

**Function – 1**

<https://www.sanfoundry.com/python-mcqs-function-1/>

1. Which of the following is the use of function in python?

a) Functions are reusable pieces of programs

b) Functions don’t provide better modularity for your application

c) you can’t also create your own functions

d) All of the mentioned

Answer: a

Explanation: Functions are reusable pieces of programs. They allow you to give a name to a block of statements, allowing you to run that block using the specified name anywhere in your program and any number of times.

2. Which keyword is used for function?

a) Fun b) Define c) Def d) Function

Answer: c

Explanation: None.

3. What will be the output of the following Python code?

def sayHello():

print('Hello World!')

sayHello()

sayHello()

a)

Hello World!

Hello World!

b)

'Hello World!'

'Hello World!'

c)

Hello

Hello

d) None of the mentioned

Answer: a

Explanation: Functions are defined using the def keyword. After this keyword comes an identifier name for the function, followed by a pair of parentheses which may enclose some names of variables, and by the final colon that ends the line. Next follows the block of statements that are part of this function.

def sayHello():

print('Hello World!') # block belonging to the function

# End of function #

sayHello() # call the function

sayHello() # call the function again

4. What will be the output of the following Python code?

def printMax(a, b):

if a > b:

print(a, 'is maximum')

elif a == b:

print(a, 'is equal to', b)

else:

print(b, 'is maximum')

printMax(3, 4)

a) 3 b) 4 c) 4 is maximum d) None of the mentioned

Answer: c

Explanation: Here, we define a function called printMax that uses two parameters called a and b. We find out the greater number using a simple if..else statement and then print the bigger number.

5. What will be the output of the following Python code?

x = 50

def func(x):

print('x is', x)

x = 2

print('Changed local x to', x)

func(x)

print('x is now', x)

a) x is now 50 b) x is now 2 c) x is now 100 d) None of the mentioned

Answer: a

Explanation: The first time that we print the value of the name x with the first line in the function’s body, Python uses the value of the parameter declared in the main block, above the function definition.

Next, we assign the value 2 to x. The name x is local to our function. So, when we change the value of x in the function, the x defined in the main block remains unaffected.

With the last print function call, we display the value of x as defined in the main block, thereby confirming that it is actually unaffected by the local assignment within the previously called function.

6. What will be the output of the following Python code?

x = 50

def func():

global x

print('x is', x)

x = 2

print('Changed global x to', x)

func()

print('Value of x is', x)

a)

x is 50

Changed global x to 2

Value of x is 50

b)

x is 50

Changed global x to 2

Value of x is 2

c)

x is 50

Changed global x to 50

Value of x is 50

d) None of the mentioned

Answer: b

Explanation: The global statement is used to declare that x is a global variable – hence, when we assign a value to x inside the function, that change is reflected when we use the value of x in the main block.

7. What will be the output of the following Python code?

def say(message, times = 1):

print(message \* times)

say('Hello')

say('World', 5)

a)

Hello

WorldWorldWorldWorldWorld

b)

Hello

World 5

c)

Hello

World,World,World,World,World

d)

Hello

HelloHelloHelloHelloHello

Answer: a

Explanation: For some functions, you may want to make some parameters optional and use default values in case the user does not want to provide values for them. This is done with the help of default argument values. You can specify default argument values for parameters by appending to the parameter name in the function definition the assignment operator (=) followed by the default value.

The function named say is used to print a string as many times as specified. If we don’t supply a value, then by default, the string is printed just once. We achieve this by specifying a default argument value of 1 to the parameter times.

In the first usage of say, we supply only the string and it prints the string once. In the second usage of say, we supply both the string and an argument 5 stating that we want to say the string message 5 times.

8. What will be the output of the following Python code?

def func(a, b=5, c=10):

print('a is', a, 'and b is', b, 'and c is', c)

func(3, 7)

func(25, c = 24)

func(c = 50, a = 100)

a)

a is 7 and b is 3 and c is 10

a is 25 and b is 5 and c is 24

a is 5 and b is 100 and c is 50

b)

a is 3 and b is 7 and c is 10

a is 5 and b is 25 and c is 24

a is 50 and b is 100 and c is 5

c)

a is 3 and b is 7 and c is 10

a is 25 and b is 5 and c is 24

a is 100 and b is 5 and c is 50

d) None of the mentioned

Answer: c

Explanation: If you have some functions with many parameters and you want to specify only some of them, then you can give values for such parameters by naming them – this is called keyword arguments – we use the name (keyword) instead of the position (which we have been using all along) to specify the arguments to the function.

The function named func has one parameter without a default argument value, followed by two parameters with default argument values.

In the first usage, func(3, 7), the parameter a gets the value 3, the parameter b gets the value 7 and c gets the default value of 10.

In the second usage func(25, c=24), the variable a gets the value of 25 due to the position of the argument. Then, the parameter c gets the value of 24 due to naming i.e. keyword arguments. The variable b gets the default value of 5.

In the third usage func(c=50, a=100), we use keyword arguments for all specified values. Notice that we are specifying the value for parameter c before that for a even though a is defined before c in the function definition.

9. What will be the output of the following Python code?

def maximum(x, y):

if x > y:

return x

elif x == y:

return 'The numbers are equal'

else:

return y

print(maximum(2, 3))

a) 2

b) 3

c) The numbers are equal

d) None of the mentioned

Answer: b

Explanation: The maximum function returns the maximum of the parameters, in this case the numbers supplied to the function. It uses a simple if..else statement to find the greater value and then returns that value.

10. Which of the following is a feature of DocString?

a) Provide a convenient way of associating documentation with Python modules, functions, classes, and methods

b) All functions should have a docstring

c) Docstrings can be accessed by the \_\_doc\_\_ attribute on objects

d) All of the mentioned

Answer: d

Explanation: Python has a nifty feature called documentation strings, usually referred to by its shorter name docstrings. DocStrings are an important tool that you should make use of since it helps to document the program better and makes it easier to understand.

1. Which are the advantages of functions in python?

a) Reducing duplication of code b) Decomposing complex problems into simpler pieces

c) Improving clarity of the code d) All of the mentioned

Answer: d

Explanation: None.

2. What are the two main types of functions?

a) Custom function

b) Built-in function & User defined function

c) User function

d) System function

Answer: b

Explanation: Built-in functions and user defined ones. The built-in functions are part of the Python language. Examples are: dir(), len() or abs(). The user defined functions are functions created with the def keyword.

3. Where is function defined?

a) Module b) Class c) Another function d) All of the mentioned

Answer: d

Explanation: Functions can be defined inside a module, a class or another function.

4. What is called when a function is defined inside a class?

a) Module b) Class c) Another function d) Method

Answer: d

Explanation: None.

5. Which of the following is the use of id() function in python?

a) Id returns the identity of the object

b) Every object doesn’t have a unique id

c) All of the mentioned

d) None of the mentioned

Answer: a

Explanation: Each object in Python has a unique id. The id() function returns the object’s id.

6. Which of the following refers to mathematical function?

a) sqrt b) rhombus c) add d) rhombus

Answer: a

Explanation: Functions that are always available for usage, functions that are contained within external modules, which must be imported and functions defined by a programmer with the def keyword.

Eg: math import sqrt

A sqrt() function is imported from the math module.

7. What will be the output of the following Python code?

def cube(x):

return x \* x \* x

x = cube(3)

print x

a) 9

b) 3

c) 27

d) 30

Answer: c

Explanation: A function is created to do a specific task. Often there is a result from such a task. The return keyword is used to return values from a function. A function may or may not return a value. If a function does not have a return keyword, it will send a none value.

8. What will be the output of the following Python code?

def C2F(c):

return c \* 9/5 + 32

print C2F(100)

print C2F(0)

a)

212

32

b)

314

24

c)

567

98

d) None of the mentioned

Answer: a

Explanation: The code shown above is used to convert a temperature in degree celsius to fahrenheit.

9. What will be the output of the following Python code?

def power(x, y=2):

r = 1

for i in range(y):

r = r \* x

return r

print power(3)

print power(3, 3)

a)

212

32

b)

9

27

c)

567

98

d) None of the mentioned

Answer: b

Explanation: The arguments in Python functions may have implicit values. An implicit value is used, if no value is provided. Here we created a power function. The function has one argument with an implicit value. We can call the function with one or two arguments.

10. What will be the output of the following Python code?

def sum(\*args):

'''Function returns the sum

of all values'''

r = 0

for i in args:

r += i

return r

print sum.\_\_doc\_\_

print sum(1, 2, 3)

print sum(1, 2, 3, 4, 5)

a)

6

15

b)

6

100

c)

123

12345

d) None of the mentioned

Answer: a

Explanation: We use the \* operator to indicate, that the function will accept arbitrary number of arguments. The sum() function will return the sum of all arguments. The first string in the function body is called the function documentation string. It is used to document the function. The string must be in triple quotes.

1. Python supports the creation of anonymous functions at runtime, using a construct called \_\_\_\_\_\_\_\_\_\_

a) lambda b) pi c) anonymous d) none of the mentioned

Answer: a

Explanation: Python supports the creation of anonymous functions (i.e. functions that are not bound to a name) at runtime, using a construct called lambda. Lambda functions are restricted to a single expression. They can be used wherever normal functions can be used.

2. What will be the output of the following Python code?

y = 6

z = lambda x: x \* y

print z(8)

a) 48 b) 14 c) 64 d) None of the mentioned

Answer: a

Explanation: The lambda keyword creates an anonymous function. The x is a parameter, that is passed to the lambda function. The parameter is followed by a colon character. The code next to the colon is the expression that is executed, when the lambda function is called. The lambda function is assigned to the z variable.

The lambda function is executed. The number 8 is passed to the anonymous function and it returns 48 as the result. Note that z is not a name for this function. It is only a variable to which the anonymous function was assigned.

3. What will be the output of the following Python code?

lamb = lambda x: x \*\* 3

print(lamb(5))

a) 15 b) 555 c) 125 d) None of the mentioned

Answer: c

Explanation: None.

4. Does Lambda contains return statements?

a) True b) False

Answer: b

Explanation: lambda definition does not include a return statement. it always contains an expression which is returned. Also note that we can put a lambda definition anywhere a function is expected. We don’t have to assign it to a variable at all.

5. Lambda is a statement.

a) True b) False

Answer: b

Explanation: lambda is an anonymous function in Python. Hence this statement is false.

6. Lambda contains block of statements.

a) True b) False

Answer: b

Explanation: None.

7. What will be the output of the following Python code?

def f(x, y, z): return x + y + z

f(2, 30, 400)

a) 432 b) 24000 c) 430 d) No output

Answer: a

Explanation: None.

8. What will be the output of the following Python code?

def writer():

title = 'Sir'

name = (lambda x:title + ' ' + x)

return name

who = writer()

who('Arthur')

a) Arthur Sir b) Sir Arthur c) Arthur d) None of the mentioned

Answer: b

Explanation: None.

9. What will be the output of the following Python code?

L = [lambda x: x \*\* 2,

lambda x: x \*\* 3,

lambda x: x \*\* 4]

for f in L:

print(f(3))

a)

27

81

343

b)

6

9

12

c)

9

27

81

d) None of the mentioned

Answer: c

Explanation: None.

10. What will be the output of the following Python code?

min = (lambda x, y: x if x < y else y)

min(101\*99, 102\*98)

a) 9997

b) 9999

c) 9996

d) None of the mentioned

Answer: c

Explanation: None.

1. What is a variable defined outside a function referred to as?

a) A static variable

b) A global variable

c) A local variable

d) An automatic variable

Answer: b

Explanation: The value of a variable defined outside all function definitions is referred to as a global variable and can be used by multiple functions of the program.

2. What is a variable defined inside a function referred to as?

a) A global variable

b) A volatile variable

c) A local variable

d) An automatic variable

Answer: c

Explanation: The variable inside a function is called as local variable and the variable definition is confined only to that function.

3. What will be the output of the following Python code?

i=0

def change(i):

i=i+1

return i

change(1)

print(i)

a) 1

b) Nothing is displayed

c) 0

d) An exception is thrown

Answer: c

Explanation: Any change made in to an immutable data type in a function isn’t reflected outside the function.

4. What will be the output of the following Python code?

def a(b):

b = b + [5]

c = [1, 2, 3, 4]

a(c)

print(len(c))

a) 4

b) 5

c) 1

d) An exception is thrown

Answer: b

Explanation: Since a list is mutable, any change made in the list in the function is reflected outside the function.

5. What will be the output of the following Python code?

a=10

b=20

def change():

global b

a=45

b=56

change()

print(a)

print(b)

a)

10

56

b)

45

56

c)

10

20

d) Syntax Error

Answer: a

Explanation: The statement “global b” allows the global value of b to be accessed and changed. Whereas the variable a is local and hence the change isn’t reflected outside the function.

6. What will be the output of the following Python code?

def change(i = 1, j = 2):

i = i + j

j = j + 1

print(i, j)

change(j = 1, i = 2)

a) An exception is thrown because of conflicting values

b) 1 2

c) 3 3

d) 3 2

Answer: d

Explanation: The values given during function call is taken into consideration, that is, i=2 and j=1.

7. What will be the output of the following Python code?

def change(one, \*two):

print(type(two))

change(1,2,3,4)

a) Integer

b) Tuple

c) Dictionary

d) An exception is thrown

Answer: b

Explanation: The parameter two is a variable parameter and consists of (2,3,4). Hence the data type is tuple.

8. If a function doesn’t have a return statement, which of the following does the function return?

a) int

b) null

c) None

d) An exception is thrown without the return statement

Answer: c

Explanation: A function can exist without a return statement and returns None if the function doesn’t have a return statement.

9. What will be the output of the following Python code?

def display(b, n):

while n > 0:

print(b,end="")

n=n-1

display('z',3)

a) zzz

b) zz

c) An exception is executed

d) Infinite loop

Answer: a

Explanation: The loop runs three times and ‘z’ is printed each time.

10. What will be the output of the following Python code?

def find(a, \*\*b):

print(type(b))

find('letters',A='1',B='2')

a) String

b) Tuple

c) Dictionary

d) An exception is thrown

Answer: c

Explanation: b combines the remaining parameters into a dictionary.

**Argument Parsing 1**

<https://www.sanfoundry.com/python-puzzles/>

This set of Python Puzzles focuses on “Argument Parsing”.

1. What is the type of each element in sys.argv?

a) set

b) list

c) tuple

d) string

Answer: d

Explanation: It is a list of strings.

2. What is the length of sys.argv?

a) number of arguments

b) number of arguments + 1

c) number of arguments – 1

d) none of the mentioned

Answer: b

Explanation: The first argument is the name of the program itself. Therefore the length of sys.argv is one more than the number arguments.

3. What will be the output of the following Python code?

def foo(k):

k[0] = 1

q = [0]

foo(q)

print(q)

a) [0]

b) [1]

c) [1, 0]

d) [0, 1]

Answer: b

Explanation: Lists are passed by reference.

4. How are keyword arguments specified in the function heading?

a) one-star followed by a valid identifier

b) one underscore followed by a valid identifier

c) two stars followed by a valid identifier

d) two underscores followed by a valid identifier

Answer: c

Explanation: Refer documentation.

5. How many keyword arguments can be passed to a function in a single function call?

a) zero b) one c) zero or more d) one or more

Answer: c

Explanation: Zero keyword arguments may be passed if all the arguments have default values.

6. What will be the output of the following Python code?

def foo(fname, val):

print(fname(val))

foo(max, [1, 2, 3])

foo(min, [1, 2, 3])

a) 3 1

b) 1 3

c) error

d) none of the mentioned

Answer: a

Explanation: It is possible to pass function names as arguments to other functions.

7. What will be the output of the following Python code?

def foo():

return total + 1

total = 0

print(foo())

a) 0

b) 1

c) error

d) none of the mentioned

Answer: b

Explanation: It is possible to read the value of a global variable directly.

8. What will be the output of the following Python code?

def foo():

total += 1

return total

total = 0

print(foo())

a) 0

b) 1

c) error

d) none of the mentioned

Answer: c

Explanation: It is not possible to change the value of a global variable without explicitly specifying it.

9. What will be the output of the following Python code?

def foo(x):

x = ['def', 'abc']

return id(x)

q = ['abc', 'def']

print(id(q) == foo(q))

a) True

b) False

c) None

d) Error

Answer: b

Explanation: A new object is created in the function.

10. What will be the output of the following Python code?

def foo(i, x=[]):

x.append(i)

return x

for i in range(3):

print(foo(i))

a) [0] [1] [2]

b) [0] [0, 1] [0, 1, 2]

c) [1] [2] [3]

d) [1] [1, 2] [1, 2, 3]

Answer: b

Explanation: When a list is a default value, the same list will be reused.

1. What will be the output of the following Python code?

def foo(k):

k = [1]

q = [0]

foo(q)

print(q)

a) [0]

b) [1]

c) [1, 0]

d) [0, 1]

Answer: a

Explanation: A new list object is created in the function and the reference is lost. This can be checked by comparing the id of k before and after k = [1].

2. How are variable length arguments specified in the function heading?

a) one star followed by a valid identifier

b) one underscore followed by a valid identifier

c) two stars followed by a valid identifier

d) two underscores followed by a valid identifier

Answer: a

Explanation: Refer documentation.

3. Which module in the python standard library parses options received from the command line?

a) getopt

b) os

c) getarg

d) main

Answer: a

Explanation: getopt parses options received from the command line.

4. What is the type of sys.argv?

a) set

b) list

c) tuple

d) string

Answer: b

Explanation: It is a list of elements.

5. What is the value stored in sys.argv[0]?

a) null

b) you cannot access it

c) the program’s name

d) the first argument

Answer: c

Explanation: Refer documentation.

6. How are default arguments specified in the function heading?

a) identifier followed by an equal to sign and the default value

b) identifier followed by the default value within backticks (“)

c) identifier followed by the default value within square brackets ([])

d) identifier

Answer: a

Explanation: Refer documentation.

7. How are required arguments specified in the function heading?

a) identifier followed by an equal to sign and the default value

b) identifier followed by the default value within backticks (“)

c) identifier followed by the default value within square brackets ([])

d) identifier

Answer: d

Explanation: Refer documentation.

8. What will be the output of the following Python code?

def foo(x):

x[0] = ['def']

x[1] = ['abc']

return id(x)

q = ['abc', 'def']

print(id(q) == foo(q))

a) True b) False c) None d) Error

Answer: a

Explanation: The same object is modified in the function.

9. Where are the arguments received from the command line stored?

a) sys.argv b) os.argv c) argv d) none of the mentioned

Answer: a

Explanation: Refer documentation.

10. What will be the output of the following Python code?

def foo(i, x=[]):

x.append(x.append(i))

return x

for i in range(3):

y = foo(i)

print(y)

a) [[[0]], [[[0]], [1]], [[[0]], [[[0]], [1]], [2]]]

b) [[0], [[0], 1], [[0], [[0], 1], 2]]

c) [0, None, 1, None, 2, None]

d) [[[0]], [[[0]], [1]], [[[0]], [[[0]], [1]], [2]]]

Answer: c

Explanation: append() returns None.

**Global vs Local Variables – 1**

<https://www.sanfoundry.com/python-questions-answers-global-local-variables-1/>

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Global vs Local Variables – 1”.

1. What will be the output of the following Python code?

def f1():

x=15

print(x)

x=12

f1()

a) Error

b) 12

c) 15

d) 1512

Answer: c

Explanation: In the code shown above, x=15 is a local variable whereas x=12 is a global variable. Preference is given to local variable over global variable. Hence the output of the code shown above is 15.

2. What will be the output of the following Python code?

def f1():

x=100

print(x)

x=+1

f1()

a) Error

b) 100

c) 101

d) 99

Answer: b

Explanation: The variable x is a local variable. It is first printed and then modified. Hence the output of this code is 100.

3. What will be the output of the following Python code?

def san(x):

print(x+1)

x=-2

x=4

san(12)

a) 13

b) 10

c) 2

d) 5

Answer: a

Explanation: The value passed to the function san() is 12. This value is incremented by one and printed. Hence the output of the code shown above is 13.

4. What will be the output of the following Python code?

def f1():

global x

x+=1

print(x)

x=12

print("x")

a) Error

b) 13

c)

13

x

d) x

Answer: d

Explanation: In the code shown above, the variable ‘x’ is declared as global within the function. Hence the output is ‘x’. Had the variable ‘x’ been a local variable, the output would have been:

13

X

5. What will be the output of the following Python code?

def f1(x):

global x

x+=1

print(x)

f1(15)

print("hello")

a) error

b) hello

c) 16

d)

16

hello

Answer: a

Explanation: The code shown above will result in an error because ‘x’ is a global variable. Had it been a local variable, the output would be: 16

hello

6. What will be the output of the following Python code?

x=12

def f1(a,b=x):

print(a,b)

x=15

f1(4)

a) Error

b) 12 4

c) 4 12

d) 4 15

Answer: c

Explanation: At the time of leader processing, the value of ‘x’ is 12. It is not modified later. The value passed to the function f1 is 4. Hence the output of the code shown above is 4 12.

7. What will be the output of the following Python code?

def f():

global a

print(a)

a = "hello"

print(a)

a = "world"

f()

print(a)

a)

hello

hello

world

b)

world

hello

hello

c)

hello

world

world

d)

world

hello

world

Answer: b

Explanation: Since the variable ‘a’ has been explicitly specified as a global variable, the value of a passed to the function is ‘world’. Hence the output of this code is:

world

hello

hello

8. What will be the output of the following Python code?

def f1(a,b=[]):

b.append(a)

return b

print(f1(2,[3,4]))

a) [3,2,4]

b) [2,3,4]

c) Error

d) [3,4,2]

Answer: d

Explanation: In the code shown above, the integer 2 is appended to the list [3,4]. Hence the output of the code is [3,4,2]. Both the variables a and b are local variables.

9. What will be the output of the following Python code?

def f(p, q, r):

global s

p = 10

q = 20

r = 30

s = 40

print(p,q,r,s)

p,q,r,s = 1,2,3,4

f(5,10,15)

a) 1 2 3 4

b) 5 10 15 4

c) 10 20 30 40

d) 5 10 15 40

Answer: c

Explanation: The above code shows a combination of local and global variables. The output of this code is: 10 20 30 40

10. What will be the output of the following Python code?

def f(x):

print("outer")

def f1(a):

print("inner")

print(a,x)

f(3)

f1(1)

a)

outer

error

b)

inner

error

c)

outer

inner

d) error

Answer: a

Explanation: The error will be caused due to the statement f1(1) because the function is nested. If f1(1) had been called inside the function, the output would have been different and there would be no error.

11. What will be the output of the following Python code?

x = 5

def f1():

global x

x = 4

def f2(a,b):

global x

return a+b+x

f1()

total = f2(1,2)

print(total)

a) Error

b) 7

c) 8

d) 15

Answer: b

Explanation: In the code shown above, the variable ‘x’ has been declared as a global variable under both the functions f1 and f2. The value returned is a+b+x = 1+2+4 = 7.

12. What will be the output of the following Python code?

x=100

def f1():

global x

x=90

def f2():

global x

x=80

print(x)

a) 100

b) 90

c) 80

d) Error

Answer: a

Explanation: The output of the code shown above is 100. This is because the variable ‘x’ has been declared as global within the functions f1 and f2.

13. Read the following Python code carefully and point out the global variables?

y, z = 1, 2

def f():

global x

x = y+z

a) x

b) y and z

c) x, y and z

d) Neither x, nor y, nor z

Answer: c

Explanation: In the code shown above, x, y and z are global variables inside the function f. y and z are global because they are not assigned in the function. x is a global variable because it is explicitly specified so in the code. Hence, x, y and z are global variables.

1. Which of the following data structures is returned by the functions globals() and locals()?

a) list

b) set

c) dictionary

d) tuple

Answer: c

Explanation: Both the functions, that is, globals() and locals() return value of the data structure dictionary.

2. What will be the output of the following Python code?

x=1

def cg():

global x

x=x+1

cg()

x

a) 2

b) 1

c) 0

d) Error

Answer: a

Explanation: Since ‘x’ has been declared a global variable, it can be modified very easily within the function. Hence the output is 2.

3. On assigning a value to a variable inside a function, it automatically becomes a global variable.

a) True

b) False

Answer: b

Explanation: On assigning a value to a variable inside a function, t automatically becomes a local variable. Hence the above statement is false.

4. What will be the output of the following Python code?

e="butter"

def f(a): print(a)+e

f("bitter")

a) error

b)

butter

error

c)

bitter

error

d) bitterbutter

Answer: c

Explanation: The output of the code shown above will be ‘bitter’, followed by an error. The error is because the operand ‘+’ is unsupported on the types used above.

5. What happens if a local variable exists with the same name as the global variable you want to access?

a) Error

b) The local variable is shadowed

c) Undefined behavior

d) The global variable is shadowed

Answer: d

Explanation: If a local variable exists with the same name as the local variable that you want to access, then the global variable is shadowed. That is, preference is given to the local variable.

6. What will be the output of the following Python code?

a=10

globals()['a']=25

print(a)

a) 10

b) 25

c) Junk value

d) Error

Answer: b

Explanation: In the code shown above, the value of ‘a’ can be changed by using globals() function. The dictionary returned is accessed using key of the variable ‘a’ and modified to 25.

7. What will be the output of the following Python code?

def f(): x=4

x=1

f()

x

a) Error

b) 4

c) Junk value

d) 1

Answer: d

Explanation: In the code shown above, when we call the function f, a new namespace is created. The assignment x=4 is performed in the local namespace and does not affect the global namespace. Hence the output is 1.

8. \_\_\_\_\_\_\_\_\_\_\_\_\_\_ returns a dictionary of the module namespace.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ returns a dictionary of the current namespace.

a)

locals()

globals()

b)

locals()

locals()

c)

globals()

locals()

d)

globals()

globals()

Answer: c

Explanation: The function globals() returns a dictionary of the module namespace, whereas the function locals() returns a dictionary of the current namespace.

**Recursion**

<https://www.sanfoundry.com/python-questions-answers-recursion/>

1. Which is the most appropriate definition for recursion?

a) A function that calls itself

b) A function execution instance that calls another execution instance of the same function

c) A class method that calls another class method

d) An in-built method that is automatically called

Answer: b

Explanation: The appropriate definition for a recursive function is a function execution instance that calls another execution instance of the same function either directly or indirectly.

2. Only problems that are recursively defined can be solved using recursion.

a) True b) False

Answer: b

Explanation: There are many other problems can also be solved using recursion.

3. Which of these is false about recursion?

a) Recursive function can be replaced by a non-recursive function

b) Recursive functions usually take more memory space than non-recursive function

c) Recursive functions run faster than non-recursive function

d) Recursion makes programs easier to understand

Answer: c

Explanation: The speed of a program using recursion is slower than the speed of its non-recursive equivalent.

4. Fill in the line of the following Python code for calculating the factorial of a number.

def fact(num):

if num == 0:

return 1

else:

return \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

a) num\*fact(num-1) b) (num-1)\*(num-2) c) num\*(num-1) d) fact(num)\*fact(num-1)

Answer: a

Explanation: Suppose n=5 then, 5\*4\*3\*2\*1 is returned which is the factorial of 5.

5. What will be the output of the following Python code?

def test(i,j):

if(i==0):

return j

else:

return test(i-1,i+j)

print(test(4,7))

a) 13 b) 7 c) Infinite loop d) 17

Answer: d

Explanation: The test(i-1,i+j) part of the function keeps calling the function until the base condition of the function is satisfied.

6. What will be the output of the following Python code?

l=[]

def convert(b):

if(b==0):

return l

dig=b%2

l.append(dig)

convert(b//2)

convert(6)

l.reverse()

for i in l:

print(i,end="")

a) 011 b) 110 c) 3 d) Infinite loop

Answer: b

Explanation: The above code gives the binary equivalent of the number.

7. What is tail recursion?

a) A recursive function that has two base cases

b) A function where the recursive functions leads to an infinite loop

c) A recursive function where the function doesn’t return anything and just prints the values

d) A function where the recursive call is the last thing executed by the function

Answer: d

Explanation: A recursive function is tail recursive when recursive call is executed by the function in the last.

8. Observe the following Python code?

def a(n):

if n == 0:

return 0

else:

return n\*a(n - 1)

def b(n, tot):

if n == 0:

return tot

else:

return b(n-2, tot-2)

a) Both a() and b() aren’t tail recursive b) Both a() and b() are tail recursive

c) b() is tail recursive but a() isn’t d) a() is tail recursive but b() isn’t

Answer: c

Explanation: A recursive function is tail recursive when recursive call is executed by the function in the last.

9. Which of the following statements is false about recursion?

a) Every recursive function must have a base case

b) Infinite recursion can occur if the base case isn’t properly mentioned

c) A recursive function makes the code easier to understand

d) Every recursive function must have a return value

Answer: d

Explanation: A recursive function needn’t have a return value.

10. What will be the output of the following Python code?

def fun(n):

if (n > 100):

return n - 5

return fun(fun(n+11));

print(fun(45))

a) 50 b) 100 c) 74 d) Infinite loop

Answer: b

Explanation: The fun(fun(n+11)) part of the code keeps executing until the value of n becomes greater than 100, after which n-5 is returned and printed.

11. Recursion and iteration are the same programming approach.

a) True b) False

Answer: b

Explanation: In recursion, the function calls itself till the base condition is reached whereas iteration means repetition of process for example in for-loops.

12. What happens if the base condition isn’t defined in recursive programs?

a) Program gets into an infinite loop

b) Program runs once

c) Program runs n number of times where n is the argument given to the function

d) An exception is thrown

Answer: a

Explanation: The program will run until the system gets out of memory.

13. Which of these is not true about recursion?

a) Making the code look clean b) A complex task can be broken into sub-problems

c) Recursive calls take up less memory d) Sequence generation is easier than a nested iteration

Answer: c

Explanation: Recursive calls take up a lot of memory and time as memory is taken up each time the function is called.

14. Which of these is not true about recursion?

a) It’s easier to code some real-world problems using recursion than non-recursive equivalent

b) Recursive functions are easy to debug

c) Recursive calls take up a lot of memory

d) Programs using recursion take longer time than their non-recursive equivalent

Answer: b

Explanation: Recursive functions may be hard to debug as the logic behind recursion may be hard to follow.

15. What will be the output of the following Python code?

def a(n):

if n == 0:

return 0

elif n == 1:

return 1

else:

return a(n-1)+a(n-2)

for i in range(0,4):

print(a(i),end=" ")

a) 0 1 2 3 b) An exception is thrown c) 0 1 1 2 3 d) 0 1 1 2

Answer: d

Explanation: The above piece of code prints the Fibonacci series.

**Shallow copy vs Deep copy**

<https://www.sanfoundry.com/python-questions-answers-shallow-copy-deep-copy/>

1. Which type of copy is shown in the following python code?

l1=[[10, 20], [30, 40], [50, 60]]

ls=list(l1)

ls

[[10, 20], [30, 40], [50, 60]]

a) Shallow copy b) Deep copy c) memberwise d) All of the mentioned

Answer: a

Explanation: The code shown above depicts shallow copy. For deep copy, the command given is: l2 = l1.copy().

2. What will be the output of the following Python code?

l=[2, 3, [4, 5]]

l2=l.copy()

l2[0]=88

l

l2

a)

[88, 2, 3, [4, 5]]

[88, 2, 3, [4, 5]]

b)

[2, 3, [4, 5]]

[88, 2, 3, [4, 5]]

c)

[88, 2, 3, [4, 5]]

[2, 3, [4, 5]]

d)

[2, 3, [4, 5]]

[2, 3, [4, 5]]

Answer: b

Explanation: The code shown above depicts deep copy. In deep copy, the base address of the objects is not copied. Hence the modification done on one list does not affect the other list.

3. In \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ copy, the base address of the objects are copied. In \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ copy, the base address of the objects are not copied.

a) deep. Shallow b) memberwise, shallow c) shallow, deep d) deep, memberwise

Answer: c

Explanation: In shallow copy, the base address of the objects are copied.

In deep copy, the base address of the objects are not copied.

Note that memberwise copy is another name for shallow copy.

4. The nested list undergoes shallow copy even when the list as a whole undergoes deep copy.

a) True b) False

Answer: a

Explanation: A nested list undergoes shallow copy even when the list as a whole undergoes deep copy. Hence, this statement is true.

5. What will be the output of the following Python code and state the type of copy that is depicted?

l1=[2, 4, 6, 8]

l2=[1, 2, 3]

l1=l2

l2

a) [2, 4, 6, 8], shallow copy b) [2, 4, 6, 8], deep copy

c) [1, 2, 3], shallow copy d) [1, 2, 3], deep copy

Answer: c

Explanation: The code shown above depicts shallow copy and the output of the code is: [1, 2, 3].

6. What will be the output of the following Python code?

l1=[10, 20, 30]

l2=l1

id(l1)==id(l2)

l2=l1.copy()

id(l1)==id(l2)

a) False, False b) False, True c) True, True d) True, False

Answer: d

Explanation: The first code shown above represents shallow copy. Hence the output of the expression id(l1)==id(l2) is True. The second code depicts deep copy. Hence the output of the expression id(l1)==id(l2) in the second case is False.

7. What will be the output of the following Python code?

l1=[1, 2, 3, [4]]

l2=list(l1)

id(l1)==id(l2)

a) True b) False c) Error d) Address of l1

Answer: b

Explanation: The code shown above shows a nested list. A nested list will undergo shallow copy when the list as a whole undergoes deep copy. Hence the output of this code is False.

8. What will be the output of the following Python code?

l1=[10, 20, 30, [40]]

l2=copy.deepcopy(l1)

l1[3][0]=90

l1

l2

a)

[10, 20, 30, [40]]

[10, 20, 30, 90]

b) Error

c)

[10, 20, 30 [90]]

[10, 20, 30, [40]]

d)

[10, 20, 30, [40]]

[10, 20, 30, [90]]

Answer: c

Explanation: The code shown above depicts deep copy. Hence at the end of the code, l1=[10, 20, 30, [90]] and l2=[10, 20, 30, [40]].

9. In \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ copy, the modification done on one list affects the other list. In \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ copy, the modification done on one list does not affect the other list.

a) shallow, deep b) memberwise, shallow

c) deep, shallow d) deep, memberwise

Answer: a

Explanation: In shallow copy, the modification done on one list affects the other list. In deep copy, the modification done on one list does not affect the other list.

10. What will be the output of the following Python code?

l1=[1, 2, 3, (4)]

l2=l1.copy()

l2

l1

a)

[1, 2, 3, (4)]

[1, 2, 3, 4]

b)

[1, 2, 3, 4]

[1, 2, 3, (4)]

c)

[1, 2, 3, 4]

[1, 2, 3, 4]

d)

[1, 2, 3, (4)]

[1, 2, 3, (4)]

Answer: c =>>In the code shown above, the list l1 is enclosed in a tuple. When we print this list, it is printed as [1, 2, 3, 4]. Note the absence of the tuple. The code shown depicts deep copy. Hence the output of this program is: l1=[1, 2, 3, 4] and l2=[1, 2, 3, 4].

11. What will be the output of the following Python code?

def check(n):

if n < 2:

return n % 2 == 0

return check(n - 2)

print(check(11))

a) False b) True c) 1 d) An exception is thrown

Answer: a =>The above piece of code checks recursively whether a number is even or odd.

12. What is the base case in the Merge Sort algorithm when it is solved recursively?

a) n=0 b) n=1 c) A list of length one d) An empty list

Answer: c =>Merge Sort algorithm implements the recursive algorithm and when the recursive function receives a list of length 1 which is the base case, the list is returned.

13. What will be the output of the following Python code?

a = [1, 2, 3, 4, 5]

b = lambda x: (b (x[1:]) + x[:1] if x else [])

print(b (a))

a) 1 2 3 4 5 b) [5,4,3,2,1] c) [] d) Error, lambda functions can’t be called recursively

Answer: c =>The above piece of code appends the first element of the list to a reversed sublist and reverses the list using recursion.

**Functional Programming Tools**

<https://www.sanfoundry.com/python-questions-answers-functional-programming-tools/>

1. What will be the output of the following Python code?

odd=lambda x: bool(x%2)

numbers=[n for n in range(10)]

print(numbers)

n=list()

for i in numbers:

if odd(i):

continue

else:

break

a) [0, 2, 4, 6, 8, 10] b) [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

c) [1, 3, 5, 7, 9] d) Error

Answer: b

Explanation: The code shown above returns a new list containing whole numbers up to 10 (excluding 10). Hence the output of the code is: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9].

2. What will be the output of the following Python code?

f=lambda x:bool(x%2)

print(f(20), f(21))

a) False True b) False False c) True True d) True False

Answer: a

Explanation: The code shown above will return true if the given argument is an odd number, and false if the given argument is an even number. Since the arguments are 20 and 21 respectively, the output of this code is: False True.

3. What will be the output of the following Python code?

import functools

l=[1,2,3,4]

print(functools.reduce(lambda x,y:x\*y,l))

a) Error b) 10 c) 24 d) No output

Answer: c

Explanation: The code shown above returns the product of all the elements of the list. Hence the output is 1\*2\*3\*4 = 24.

4. What will be the output of the following Python code?

l=[1, -2, -3, 4, 5]

def f1(x):

return x<2

m1=filter(f1, l)

print(list(m1))

a) [1, 4, 5 ] b) Error c) [-2, -3] d) [1, -2, -3]

Answer: d

Explanation: The code shown above returns only those elements from the list, which are less than 2. The functional programming tool used to achieve this operation is filter. Hence the output of the code is:[1, -2, -3].

5. What will be the output of the following Python code?

l=[-2, 4]

m=map(lambda x:x\*2, l)

print(m)

a) [-4, 16]

b) Address of m

c) Error

d)

-4

16

Answer: b

Explanation: The code shown above returns the address of m. Had we used the statement: print(list(m)), the output would have been: [-4, 16].

6. What will be the output of the following Python code?

l=[1, -2, -3, 4, 5]

def f1(x):

return x<-1

m1=map(f1, l)

print(list(m1))

a) [False, False, False, False, False] b) [False, True, True, False, False]

c) [True, False, False, True, True] d) [True, True, True, True, True]

Answer: b

Explanation: This code shown returns a list which contains True if the corresponding element of the list is less than -1, and false if the corresponding element is greater than -1. Hence the output of the code shown above: [False, True, True, False, False].

7. What will be the output of the following Python code?

l=[1, 2, 3, 4, 5]

m=map(lambda x:2\*\*x, l)

print(list(m))

a) [1, 4, 9, 16, 25 ] b) [2, 4, 8, 16, 32 ]

c) [1, 0, 1, 0, 1] d) Error

Answer: b

Explanation: The code shown above prints a list containing each element of the list as the power of two. That is, the output is: [2, 4, 8, 16, 32].

8. What will be the output of the following Python code?

import functools

l=[1, 2, 3, 4, 5]

m=functools.reduce(lambda x, y:x if x>y else y, l)

print(m)

a) Error

b) Address of m

c) 1

d) 5

Answer: d

Explanation: The code shown above can be used to find the maximum of the elements from the given list. In the above code, this operation is achieved by using the programming tool reduce. Hence the output of the code shown above is 5.

9. What will be the output of the following Python code?

l=[n for n in range(5)]

f=lambda x:bool(x%2)

print(f(3), f(1))

for i in range(len(l)):

if f(l[i]):

del l[i]

print(i)

a)

True True

1

2

Error

b)

False False

1

2

c)

True False

1

2

Error

d)

False True

1

2

Answer: a

Explanation: The code shown above prints true if the value entered as an argument is odd, else false is printed. Hence the output: True True. The error is due to the list index being out of range.

10. What will be the output of the following Python code?

m=reduce(lambda x: x-3 in range(4, 10))

print(list(m))

a) [1, 2, 3, 4, 5, 6, 7] b) No output

c) [1, 2, 3, 4, 5, 6] d) Error

Answer: b

Explanation: The code shown above will result in an error. This is because e have not imported functools. Further, ‘reduce’, as such is not defined. We should use functools.reduce to remove the error.

11. Which of the following numbers will not be a part of the output list of the following Python code?

def sf(a):

return a%3!=0 and a%5!=0

m=filter(sf, range(1, 31))

print(list(m))

a) 1 b) 29 c) 6 d) 10

Answer: d

Explanation: The output list of the code shown above will not contain any element that is divisible by 3 or 5. Hence the number which is not present in the output list is 10. The output list: [1, 2, 4, 7, 8, 11, 13, 14, 16, 17, 19, 22, 23, 26, 28, 29]

12. The single line equivalent of the following Python code?

l=[1, 2, 3, 4, 5]

def f1(x):

return x<0

m1=filter(f1, l)

print(list(m1))

a) filter(lambda x:x<0, l) b) filter(lambda x, y: x<0, l)

c) filter(reduce x<0, l) d) reduce(x: x<0, l)

Answer: a

Explanation: The code shown above returns a new list containing only those elements from list l, which are less than 0. Since there are no such elements in the list l, the output of this code is: []. The single line equivalent of this code is filter(lambda x:x<0, l).

13. What will be the output of the following Python code?

list(map((lambda x:x^2), range(10)))

a) [0, 1, 4, 9, 16, 25, 36, 49, 64, 81] b) Error

c) [2, 3, 0, 1, 6, 7, 4, 5, 10, 11] d) No output

Answer: c

Explanation: The line of code shown above returns a list of each number from 1 to 10, after an XOR operation is performed on each of these numbers with 2. Hence the output of this code is: [2, 3, 0, 1, 6, 7, 4, 5, 10, 11]

14. What will be the output of the following Python code?

list(map((lambda x:x\*\*2), filter((lambda x:x%2==0), range(10))))

a) [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] b) [0, 4, 16, 36, 64]

c) Error d) No output

Answer: b

Explanation: The output list will contain each number up to 10 raised to 2, except odd numbers, that is, 1, 3, 5, 9. Hence the output of the code is: [0, 4, 16, 36, 64].

15. The output of the following codes are the same.

[x\*\*2 for x in range(10)]

list(map((lambda x:x\*\*2), range(10)))

a) True b) False

Answer: a

Explanation: Both of the codes shown above print each whole number up to 10, raised to the power 2. Hence the output of both of these codes is: [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]. Therefore, the statement is true.

**Mapping Functions – 1**

<https://www.sanfoundry.com/python-interview-questions-answers/>

1. What will be the output of the following Python code?

elements = [0, 1, 2]

def incr(x):

return x+1

print(list(map(elements, incr)))

a) [1, 2, 3]

b) [0, 1, 2]

c) error

d) none of the mentioned

Answer: c

Explanation: The list should be the second parameter to the mapping function.

2. What will be the output of the following Python code?

elements = [0, 1, 2]

def incr(x):

return x+1

print(list(map(incr, elements)))

a) [1, 2, 3]

b) [0, 1, 2]

c) error

d) none of the mentioned

Answer: a

Explanation: Each element of the list is incremented.

3. What will be the output of the following Python code?

x = ['ab', 'cd']

print(list(map(upper, x)))

a) [‘AB’, ‘CD’]

b) [‘ab’, ‘cd’]

c) error

d) none of the mentioned

Answer: c

Explanation: A NameError occurs because upper is a class method.

4. What will be the output of the following Python code?

def to\_upper(k):

return k.upper()

x = ['ab', 'cd']

print(list(map(upper, x)))

a) [‘AB’, ‘CD’]

b) [‘ab’, ‘cd’]

c) none of the mentioned

d) error

Answer: d

Explanation: A NameError occurs because upper is a class method.

5. What will be the output of the following Python code?

def to\_upper(k):

return k.upper()

x = ['ab', 'cd']

print(list(map(to\_upper, x)))

a) [‘AB’, ‘CD’] b) [‘ab’, ‘cd’] c) none of the mentioned d) error

Answer: a

Explanation: Each element of the list is converted to uppercase.

6. What will be the output of the following Python code?

def to\_upper(k):

k.upper()

x = ['ab', 'cd']

print(list(map(to\_upper, x)))

a) [‘AB’, ‘CD’] b) [‘ab’, ‘cd’] c) none of the mentioned d) error

Answer: c

Explanation: A list of Nones is printed as to\_upper() returns None.

7. What will be the output of the following Python code?

x = ['ab', 'cd']

print(map(len, x))

a) [‘ab’, ‘cd’]

b) [2, 2]

c) [‘2’, ‘2’]

d) none of the mentioned

Answer: d

Explanation: A map object is generated by map(). We must convert this to a list to be able to print it in a human readable form.

8. What will be the output of the following Python code?

x = ['ab', 'cd']

print(list(map(len, x)))

a) [‘ab’, ‘cd’] b) [2, 2] c) [‘2’, ‘2’] d) none of the mentioned

Answer: b

Explanation: The length of each string is 2.

9. What will be the output of the following Python code?

x = ['ab', 'cd']

print(len(map(list, x)))

a) [2, 2]

b) 2

c) 4

d) none of the mentioned

Answer: d

Explanation: A TypeError occurs as map has no len().

10. What will be the output of the following Python code?

x = ['ab', 'cd']

print(len(list(map(list, x))))

a) 2

b) 4

c) error

d) none of the mentioned

Answer: a

Explanation: The outer list has two lists in it. So it’s length is 2.

1. What will be the output of the following Python code?

x = ['ab', 'cd']

print(len(list(map(list, x))))))

a) 2

b) 4

c) error

d) none of the mentioned

Answer: c

Explanation: SyntaxError, unbalanced parenthesis.

2. What will be the output of the following Python code?

x = ['ab', 'cd']

print(list(map(list, x)))

a) [‘a’, ‘b’, ‘c’, ‘d’]

b) [[‘ab’], [‘cd’]]

c) [[‘a’, ‘b’], [‘c’, ‘d’]]

d) none of the mentioned

Answer: c

Explanation: Each element of x is converted into a list.

3. What will be the output of the following Python code?

x = [12, 34]

print(len(list(map(len, x))))

a) 2

b) 1

c) error

d) none of the mentioned

Answer: c

Explanation: TypeError, int has no len().

4. What will be the output of the following Python code?

x = [12, 34]

print(len(list(map(int, x))))

a) 2

b) 1

c) error

d) none of the mentioned

Answer: a

Explanation: list(map()) returns a list of two items in this example.

5. What will be the output of the following Python code?

x = [12, 34]

print(len(''.join(list(map(int, x)))))

a) 4

b) 2

c) error

d) none of the mentioned

Answer: c

Explanation: Cannot perform join on a list of ints.

6. What will be the output of the following Python code?

x = [12, 34]

print(len(''.join(list(map(str, x)))))

a) 4

b) 5

c) 6

d) error

Answer: a

Explanation: Each number is mapped into a string of length 2.

7. What will be the output of the following Python code?

x = [12, 34]

print(len(' '.join(list(map(int, x)))))

a) 4

b) 5

c) 6

d) error

Answer: d

Explanation: TypeError. Execute in shell to verify.

8. What will be the output of the following Python code?

x = [12.1, 34.0]

print(len(' '.join(list(map(str, x)))))

a) 6

b) 8

c) 9

d) error

Answer: c

Explanation: The floating point numbers are converted to strings and joined with a space between them.

9. What will be the output of the following Python code?

x = [12.1, 34.0]

print(' '.join(list(map(str, x))))

a) 12 1 34 0

b) 12.1 34

c) 121 340

d) 12.1 34.0

Answer: d

Explanation: str(ab.c) is ‘ab.c’.

10. What will be the output of the following Python code?

x = [[0], [1]]

print(len(' '.join(list(map(str, x)))))

a) 2

b) 3

c) 7

d) 8

Answer: c

Explanation: map() is applied to the elements of the outer loop.

1. What will be the output of the following Python code?

x = [[0], [1]]

print((' '.join(list(map(str, x)))))

a) (‘[0] [1]’,)

b) (’01’,)

c) [0] [1]

d) 01

Answer: c

Explanation: (element) is the same as element. It is not a tuple with one item.

2. What will be the output of the following Python code?

x = [[0], [1]]

print((' '.join(list(map(str, x))),))

a) (‘[0] [1]’,)

b) (’01’)

c) [0] [1]

d) 01

Answer: a

Explanation: (element,) is not the same as element. It is a tuple with one item.

3. What will be the output of the following Python code?

x = [34, 56]

print((''.join(list(map(str, x))),))

a) 3456

b) (3456)

c) (‘3456’)

d) (‘3456’,)

Answer: d

Explanation: We have created a tuple with one string in it.

4. What will be the output of the following Python code?

x = [34, 56]

print((''.join(list(map(str, x)))),)

a) 3456

b) (3456)

c) (‘3456’)

d) (‘3456’,)

Answer: a

Explanation: We have just created a string.

5. What will be the output of the following Python code?

x = [34, 56]

print(len(map(str, x)))

a) [34, 56]

b) [’34’, ’56’]

c) 34 56

d) error

Answer: d

Explanation: TypeError, map has no len.

6. What will be the output of the following Python code?

x = 'abcd'

print(list(map(list, x)))

a) [‘a’, ‘b’, ‘c’, ‘d’]

b) [‘abcd’]

c) [[‘a’], [‘b’], [‘c’], [‘d’]]

d) none of the mentioned

Answer: c

Explanation: list() is performed on each character in x.

7. What will be the output of the following Python code?

x = abcd

print(list(map(list, x)))

a) [‘a’, ‘b’, ‘c’, ‘d’]

b) [‘abcd’]

c) [[‘a’], [‘b’], [‘c’], [‘d’]]

d) none of the mentioned

Answer: d

Explanation: NameError, we have not defined abcd.

8. What will be the output of the following Python code?

x = 1234

print(list(map(list, x)))

a) [1, 2, 3, 4]

b) [1234]

c) [[1], [2], [3], [4]]

d) none of the mentioned

Answer: d

Explanation: TypeError, int is not iterable.

9. What will be the output of the following Python code?

x = 1234

print(list(map(list, [x])))

a) [1, 2, 3, 4]

b) [1234]

c) [[1], [2], [3], [4]]

d) none of the mentioned

Answer: d

Explanation: TypeError, int is not iterable.

10. What will be the output of the following Python code?

x = 'abcd'

print(list(map([], x)))

a) [‘a’, ‘b’, ‘c’, ‘d’]

b) [‘abcd’]

c) [[‘a’], [‘b’], [‘c’], [‘d’]]

d) none of the mentioned

Answer: d

Explanation: TypeError, list object is not callable.

11. Is Python code compiled or interpreted?

a) Python code is only compiled

b) Python code is both compiled and interpreted

c) Python code is only interpreted

d) Python code is neither compiled nor interpreted

Answer: b

Explanation: Many languages have been implemented using both compilers and interpreters, including C, Pascal, and Python.

12. Which of these is the definition for packages in Python?

a) A folder of python modules

b) A set of programs making use of Python modules

c) A set of main modules

d) A number of files containing Python definitions and statements

Answer: a

Explanation: A folder of python programs is called as a package of modules.

13. Which of these is false about a package?

a) A package can have subfolders and modules

b) Each import package need not introduce a namespace

c) import folder.subfolder.mod1 imports packages

d) from folder.subfolder.mod1 import objects imports packages

Answer: b

Explanation: Packages provide a way of structuring Python namespace. Each import package introduces a namespace.

**Python Modules**

<https://www.sanfoundry.com/python-questions-answers-modules/>

1. Which of these definitions correctly describes a module?

a) Denoted by triple quotes for providing the specification of certain program elements

b) Design and implementation of specific functionality to be incorporated into a program

c) Defines the specification of how it is to be used

d) Any program that reuses code

Answer: b

Explanation: The term “module” refers to the implementation of specific functionality to be incorporated into a program.

2. Which of the following is not an advantage of using modules?

a) Provides a means of reuse of program code b) Provides a means of dividing up tasks

c) Provides a means of reducing the size of the program d) Provides a means of testing individual parts of the program

Answer: c

Explanation: The total size of the program remains the same regardless of whether modules are used or not. Modules simply divide the program.

3. Program code making use of a given module is called a \_\_\_\_\_\_ of the module.

a) Client b) Docstring c) Interface d) Modularity

Answer: a

Explanation: Program code making use of a given module is called the client of the module. There may be multiple clients for a module.

4. \_\_\_\_\_\_ is a string literal denoted by triple quotes for providing the specifications of certain program elements.

a) Interface b) Modularity c) Client d) Docstring

Answer: d

Explanation: Docstring used for providing the specifications of program elements.

5. Which of the following is true about top-down design process?

a) The details of a program design are addressed before the overall design

b) Only the details of the program are addressed

c) The overall design of the program is addressed before the details

d) Only the design of the program is addressed

Answer: c

Explanation: Top-down design is an approach for deriving a modular design in which the overall design.

6. In top-down design every module is broken into same number of submodules.

a) True b) False

Answer: b

Explanation: In top-down design every module can even be broken down into different number of submodules.

7. All modular designs are because of a top-down design process.

a) True b) False

Answer: b

Explanation: The details of the program can be addressed before the overall design too. Hence, all modular designs are not because of a top-down design process.

8. What will be the output of the following Python code?

#mod1

def change(a):

b=[x\*2 for x in a]

print(b)

#mod2

def change(a):

b=[x\*x for x in a]

print(b)

from mod1 import change

from mod2 import change

#main

s=[1,2,3]

change(s)

a) [2,4,6]

b) [1,4,9]

c)

[2,4,6]

[1,4,9]

d) There is a name clash

Answer: d

Explanation: A name clash is when two different entities with the same identifier become part of the same scope. Since both the modules have the same function name, there is a name clash.

9. Which of the following isn’t true about main modules?

a) When a python file is directly executed, it is considered main module of a program

b) Main modules may import any number of modules

c) Special name given to main modules is: \_\_main\_\_

d) Other main modules can import main modules

Answer: d

Explanation: Main modules are not meant to be imported into other modules.

10. Which of the following is not a valid namespace?

a) Global namespace b) Public namespace

c) Built-in namespace d) Local namespace

Answer: b

Explanation: During a Python program execution, there are as many as three namespaces – built-in namespace, global namespace and local namespace.

11. Which of the following is false about “import modulename” form of import?

a) The namespace of imported module becomes part of importing module

b) This form of import prevents name clash

c) The namespace of imported module becomes available to importing module

d) The identifiers in module are accessed as: modulename.identifier

Answer: a

Explanation: In the “import modulename” form of import, the namespace of imported module becomes available to, but not part of, the importing module.

12. Which of the following is false about “from-import” form of import?

a) The syntax is: from modulename import identifier

b) This form of import prevents name clash

c) The namespace of imported module becomes part of importing module

d) The identifiers in module are accessed directly as: identifier

Answer: b

Explanation: In the “from-import” form of import, there may be name clashes because names of the imported identifiers aren’t specified along with the module name.

13. Which of the statements about modules is false?

a) In the “from-import” form of import, identifiers beginning with two underscores are private and aren’t imported

b) dir() built-in function monitors the items in the namespace of the main module

c) In the “from-import” form of import, all identifiers regardless of whether they are private or public are imported

d) When a module is loaded, a compiled version of the module with file extension .pyc is automatically produced

Answer: c

Explanation: In the “from-import” form of import, identifiers beginning with two underscores are private and aren’t imported.

14. What will be the output of the following Python code?

from math import factorial

print(math.factorial(5))

a) 120

b) Nothing is printed

c) Error, method factorial doesn’t exist in math module

d) Error, the statement should be: print(factorial(5))

Answer: d

Explanation: In the “from-import” form of import, the imported identifiers (in this case factorial()) aren’t specified along with the module name.

15. What is the order of namespaces in which Python looks for an identifier?

a) Python first searches the global namespace, then the local namespace and finally the built-in namespace

b) Python first searches the local namespace, then the global namespace and finally the built-in namespace

c) Python first searches the built-in namespace, then the global namespace and finally the local namespace

d) Python first searches the built-in namespace, then the local namespace and finally the global namespace

Answer: b

Explanation: Python first searches for the local, then the global and finally the built-in namespace.

**Math – 1**

<https://www.sanfoundry.com/python-objective-questions-answers/>

1. What is returned by math.ceil(3.4)?

a) 3 b) 4 c) 4.0 d) 3.0

Answer: b

Explanation: The ceil function returns the smallest integer that is bigger than or equal to the number itself.

2. What is the value returned by math.floor(3.4)?

a) 3 b) 4 c) 4.0 d) 3.0

Answer: a

Explanation: The floor function returns the biggest number that is smaller than or equal to the number itself.

3. What will be the output of print(math.copysign(3, -1))?

a) 1 b) 1.0 c) -3 d) -3.0

Answer: d

Explanation: The copysign function returns a float whose absolute value is that of the first argument and the sign is that of the second argument.

4. What is displayed on executing print(math.fabs(-3.4))?

a) -3.4 b) 3.4 c) 3 d) -3

Answer: b

Explanation: A negative floating point number is returned as a positive floating point number.

5. Is the output of the function abs() the same as that of the function math.fabs()?

a) sometimes b) always c) never d) none of the mentioned

Answer: a

Explanation: math.fabs() always returns a float and does not work with complex numbers whereas the return type of abs() is determined by the type of value that is passed to it.

6. What is the value returned by math.fact(6)?

a) 720 b) 6 c) [1, 2, 3, 6] d) error

Answer: d

Explanation: NameError, fact() is not defined.

7. What is the value of x if x = math.factorial(0)?

a) 0 b) 1 c) error d) none of the mentioned

Answer: b

Explanation: Factorial of 0 is 1.

8. What is math.factorial(4.0)?

a) 24 b) 1 c) error d) none of the mentioned

Answer: a

Explanation: The factorial of 4 is returned.

9. What will be the output of print(math.factorial(4.5))?

a) 24 b) 120 c) error d) 24.0

Answer: c

Explanation: Factorial is only defined for non-negative integers.

10. What is math.floor(0o10)?

a) 8 b) 10 c) 0 d) 9

Answer: a

Explanation: 0o10 is 8 and floor(8) is 8.

1. What does the function math.frexp(x) return?

a) a tuple containing the mantissa and the exponent of x

b) a list containing the mantissa and the exponent of x

c) a tuple containing the mantissa of x d) a list containing the exponent of x

Answer: a

Explanation: It returns a tuple with two elements. The first element is the mantissa and the second element is the exponent.

2. What is the result of math.fsum([.1 for i in range(20)])?

a) 2.0 b) 20 c) 2 d) 2.0000000000000004

Answer: a

Explanation: The function fsum returns an accurate floating point sum of the elements of its argument.

3. What is the result of sum([.1 for i in range(20)])?

a) 2.0 b) 20 c) 2 d) 2.0000000000000004

Answer: d

Explanation: There is some loss of accuracy when we use sum with floating point numbers. Hence the function fsum is preferable.

4. What is returned by math.isfinite(float(‘inf’))?

a) True b) False c) None d) error

Answer: b

Explanation: float(‘inf’) is not a finite number.

5. What is returned by math.isfinite(float(‘nan’))?

a) True b) False c) None d) error

Answer: b

Explanation: float(‘nan’) is not a finite number.

6. What is x if x = math.isfinite(float(‘0.0’))?

a) True b) False c) None d) error

Answer: a

Explanation: float(‘0.0’) is a finite number.

7. What will be the output of the following Python code?

>>> -float('inf') + float('inf')

a) inf b) nan c) 0 d) 0.0

Answer: b

Explanation: The result of float(‘inf’)-float(‘inf’) is undefined.

8. What will be the output of the following Python code?

print(math.isinf(float('-inf')))

a) error, the minus sign shouldn’t have been inside the brackets

b) error, there is no function called isinf

c) True

d) False

Answer: c

Explanation: -float(‘inf’) is the same as float(‘-inf’).

9. What is the value of x if x = math.ldexp(0.5, 1)?

a) 1 b) 2.0 c) 0.5 d) none of the mentioned

Answer: d

Explanation: The value returned by ldexp(x, y) is x \* (2 \*\* y). In the current case x is 1.0.

10. What is returned by math.modf(1.0)?

a) (0.0, 1.0) b) (1.0, 0.0) c) (0.5, 1) d) (0.5, 1.0)

Answer: a

Explanation: The first element is the fractional part and the second element is the integral part of the argument.

1. What is the result of math.trunc(3.1)?

a) 3.0 b) 3 c) 0.1 d) 1

Answer: b

Explanation: The integral part of the floating point number is returned.

2. What is the output of print(math.trunc(‘3.1’))?

a) 3 b) 3.0 c) error d) none of the mentioned

Answer: c

Explanation: TypeError, a string does not have \_\_trunc\_\_ method.

3. Which of the following is the same as math.exp(p)?

a) e \*\* p

b) math.e \*\* p

c) p \*\* e

d) p \*\* math.e

Answer: b

Explanation: math.e is the constant defined in the math module.

4. What is returned by math.expm1(p)?

a) (math.e \*\* p) – 1

b) math.e \*\* (p – 1)

c) error

d) none of the mentioned

Answer: a

Explanation: One is subtracted from the result of math.exp(p) and returned.

5. What is the default base used when math.log(x) is found?

a) e b) 10 c) 2 d) none of the mentioned

Answer: a

Explanation: The natural log of x is returned by default.

6. Which of the following aren’t defined in the math module?

a) log2()

b) log10()

c) logx()

d) none of the mentioned

Answer: c

Explanation: log2() and log10() are defined in the math module.

7. What is returned by int(math.pow(3, 2))?

a) 6

b) 9

c) error, third argument required

d) error, too many arguments

Answer: b

Explanation: math.pow(a, b) returns a \*\* b.

8. What is output of print(math.pow(3, 2))?

a) 9

b) 9.0

c) None

d) None of the mentioned

Answer: b

Explanation: math.pow() returns a floating point number.

9. What is the value of x if x = math.sqrt(4)?

a) 2

b) 2.0

c) (2, -2)

d) (2.0, -2.0)

Answer: b

Explanation: The function returns one floating point number.

10. What does math.sqrt(X, Y) do?

a) calculate the Xth root of Y

b) calculate the Yth root of X

c) error

d) return a tuple with the square root of X and Y

Answer: c

Explanation: The function takes only one argument.

**Datetime Module – 1**

<https://www.sanfoundry.com/python-questions-answers-datetime-module-1/>

1. What will be the output of the following Python code?

import datetime

d=datetime.date(2016,7,24)

print(d)

a) Error

b) 2017-07-24

c) 2017-7-24

d) 24-7-2017

Answer: b

Explanation: In the snippet of code shown above, we are simply printing the date entered by us. We enter the date in the format: yyyy,m,dd. The date is then printed in the format: yyyy-mm-dd. Hence the output is: 2017-07-24.

2. What will be the output of the following Python code?

import datetime

d=datetime.date(2017,06,18)

print(d)

a) Error

b) 2017-06-18

c) 18-06-2017

d) 06-18-2017

Answer: a

Explanation: The code shown above will result in an error because of the format of the date entered. Had the date been entered as: d=datetime.date(2017,6,18), no error would have been thrown.

3. What will be the output of the following Python code if the system date is 18th August, 2016?

tday=datetime.date.today()

print(tday.month())

a) August

b) Aug

c) 08

d) 8

Answer: d

Explanation: The code shown above prints the month number from the system date. Therefor the output will be 8 if the system date is 18th August, 2016.

4. What will be the output of the following Python code if the system date is 18th June, 2017 (Sunday)?

import datetime

tday=datetime.date.today()

print(tday)

a) 18-06-2017

b) 06-18-2017

c) 2017-06-18

d) Error

Answer: c

Explanation: The code shown above prints the system date in the format yyyy-mm-dd. Hence the output of this code is: 2017-06-18.

5. What will be the output of the following Python code if the system date is 18th June, 2017 (Sunday)?

tday=datetime.date.today()

print(tday.weekday())

a) 6

b) 1

c) 0

d) 7

Answer: a

Explanation: The code shown above prints an integer depending on which day of the week it is. Monday-0, Tuesday-1, Wednesday-2, Thursday-3, Friday-4, Saturday-5, Sunday-6. Hence the output is 6 in the case shown above.

6. What will be the output of the following Python code if the system date is 21st June, 2017 (Wednesday)?

tday=datetime.date.today()

print(tday.isoweekday())

a) Wed

b) Wednesday

c) 2

d) 3

Answer: d

Explanation: This code prints an integer depending on which day of the week it is. Monday-1, Tuesday-2, Wednesday-3, Thursday-4, Friday-5, Saturday-6, Sunday-7. Hence the output of the code shown above is 3.

7. Point out the error (if any) in the code shown below if the system date is 18th June, 2017?

tday=datetime.date.today()

bday=datetime.date(2017,9,18)

till\_bday=bday-tday

print(till\_bday)

a) 3 months, 0:00:00

b) 90 days, 0:00:00

c) 3 months 2 days, 0:00:00

d) 92 days, 0:00:00

Answer: d

Explanation: The code shown above can be used to find the number of days between two given dates. The output of the code shown above will thus be 92.

8. The value returned when we use the function isoweekday() is \_\_\_\_\_\_ and that for the function weekday() is \_\_\_\_\_\_\_\_ if the system date is 19th June, 2017 (Monday).

a) 0,0

b) 0,1

c) 1,0

d) 1,1

Answer: c

Explanation: The value returned when we use the function isoweekday() is 1 and that for the function weekday() is 0 if the system date is 19th June, 2017 (Monday).

9. Which of the following will throw an error if used after the following Python code?

tday=datetime.date.today()

bday=datetime.date(2017,9,18)

t\_day=bday-tday

a) print(t\_day.seconds)

b) print(t\_day.months)

c) print(t\_day.max)

d) print(t\_day.resolution)

Answer: b

Explanation: The statement: print(t\_day.months) will throw an error because there is no function such as t\_day.months, whereas t\_day.seconds, t\_day.max and t\_day.resolution are valid, provided that t\_day is defined.

10. What will be the output of the following Python code if the system date is: 6/19/2017

tday=datetime.date.today()

tdelta=datetime.timedelta(days=10)

print(tday+tdelta)

a) 2017-16-19

b) 2017-06-9

c) 2017-06-29

d) Error

Answer: c

Explanation: The code shown above will add the specified number of days to the current date and print the new date. On adding ten days to 6/19/2017, we get 6/29/2017. Hence the output is: 2017-06-29.

1. The output of both of the print statements is the same.

import datetime

dt\_1 = datetime.datetime.today()

dt\_2 = datetime.datetime.now()

print(dt\_1)

print(dt\_2)

a) True

b) False

Answer: b

Explanation: The output of the two print statements is not the same because of the difference in time between the execution of the two print statements. There is a difference in the order of milliseconds between the two statements and this is reflected in the output.

2. Which of the following functions can be used to find the coordinated universal time, assuming that the datetime module has already been imported?

a) datetime.utc()

b) datetime.datetime.utc()

c) datetime.utcnow()

d) datetime.datetime.utcnow()

Answer: d

Explanation: The function datetime.datetime.utcnow() can be used to find the UTC (Coordinated Universal Time), assuming that the datetime module has already been imported. The other function s shown above are invalid.

3. What will be the output of the following Python code?

import time

time.time()

a) The number of hours passed since 1st January, 1970

b) The number of days passed since 1st January, 1970

c) The number of seconds passed since 1st January, 1970

d) The number of minutes passed since 1st January, 1970

Answer: c

Explanation: The code shown above will return the number of seconds passed since 1st January, 1970.

4. What will be the output of the following Python code, if the time module has already been imported?

def num(m):

t1 = time.time()

for i in range(0,m):

print(i)

t2 = time.time()

print(str(t2-t1))

num(3)

a)

1

2

3

The time taken for the execution of the code

b)

3

The time taken for the execution of the code

c)

1

2

3

UTC time

d)

3

UTC time

Answer: a

Explanation: The code shown above will return the numbers 1, 2, 3, followed by the time taken in the execution of the code.

Output:

1

2

3

The time taken for the execution of the code

5. What will be the output of the following Python code?

import time

time.asctime()

a) Current date only

b) UTC time

c) Current date and time

d) Current time only

Answer: c

Explanation: The function time.asctime(), present if the time module can be used to return the current date and time. It can also accept a parameter and return the date and time in a particular format. However in the above code, since we have not passed any parameters in the above code, the current date and time is returned.

6. What will be the output of the following Python code?

import time

t=(2010, 9, 20, 8, 15, 12, 6)

time.asctime(t)

a) ‘20 Sep 2010 8:15:12 Sun’

b) ‘2010 20 Sept 08:15:12 Sun’

c) ‘Sun Sept 20 8:15:12 2010’

d) Error

Answer: d

Explanation: The code shown above results in an error because this function accepts exactly 9 arguments (including day of the year and DST), but only 7 are given. Hence an error is thrown.

7. What will be the output of the following Python code?

import time

t=(2010, 9, 20, 8, 45, 12, 6, 0, 0)

time.asctime(t)

a) ‘Sep 20 2010 08:45:12 Sun’

b) ‘Sun Sep 20 08:45:12 2010’

c) ’20 Sep 08:45:12 Sun 2010’

d) ‘2010 20 Sep 08:45:12 Sun’

Answer: b

Explanation: The code shown above returns the given date and time in a particular format. Hence the output of the code shown above will be: ‘Sun Sep 20 08:45:12 2010’.

8. The sleep function (under the time module) is used to \_\_\_\_\_\_\_\_\_\_\_

a) Pause the code for the specified number of seconds

b) Return the specified number of seconds, in terms of milliseconds

c) Stop the execution of the code

d) Return the output of the code had it been executed earlier by the specified number of seconds

Answer: a

Explanation: The sleep function (under the time module) is used to pause the code for the specified number of seconds. The number of seconds is taken as an argument by this function.

9. What will be the output of the following Python code?

import time

for i in range(0,5):

print(i)

time.sleep(2)

a) After an interval of 2 seconds, the numbers 1, 2, 3, 4, 5 are printed all together

b) After an interval of 2 seconds, the numbers 0, 1, 2, 3, 4 are printed all together

c) Prints the numbers 1, 2, 3, 4, 5 at an interval of 2 seconds between each number

d) Prints the numbers 0, 1, 2, 3, 4 at an interval of 2 seconds between each number

Answer: d

Explanation: The output of the code shown above will be the numbers 0, 1, 2, 3, 4 at an interval of 2 seconds each.

10. What will be the output if we try to extract only the year from the following Python code? (time.struct\_time(tm\_year=2017, tm\_mon=6, tm\_mday=25, tm\_hour=18, tm\_min=26, tm\_sec=6, tm\_wday=6, tm\_yday=176, tm\_isdst=0))

import time

t=time.localtime()

print(t)

a) t[1]

b) tm\_year

c) t[0]

d) t\_year

Answer: c

Explanation: To extract the year from the code shown above, we use the command t[0]. The command t[1] will return the month number (6 in the above case). The commands tm\_year and t\_year will result in errors.

11. State whether true or false.

s = time.time()

t= time.time()

s == t

a) True

b) False

Answer: b

Explanation: The variables ‘s’ and ‘t’ will not be equal due to the slight difference in the time of their execution. Hence the output of this code will be: False.

**Random module – 1**

<https://www.sanfoundry.com/python-questions-answers-random-module/>

1. To include the use of functions which are present in the random library, we must use the option:

a) import random

b) random.h

c) import.random

d) random.random

Answer: a

Explanation: The command import random is used to import the random module, which enables us to use the functions which are present in the random library.

2. The output of the following Python code is either 1 or 2.

import random

random.randint(1,2)

a) True

b) False

Answer: a

Explanation: The function random.randint(a,b) helps us to generate an integer between ‘a’ and ‘b’, including ‘a’ and ‘b’. In this case, since there are no integers between 1 and 2, the output will necessarily be either 1 or 2’.

3. What will be the output of the following Python code?

import random

random.choice(2,3,4)

a) An integer other than 2, 3 and 4

b) Either 2, 3 or 4

c) Error

d) 3 only

Answer: c

Explanation: The code shown above displays the incorrect syntax of the function random.choice(). This functions takes its numeric parameter in the form of a list. Hence the correct syntax world be: random.choice([2,3,4]).

4. What will be the output of the following Python code?

import random

random.choice([10.4, 56.99, 76])

a) Error

b) Either 10.4, 56.99 or 76

c) Any number other than 10.4, 56.99 and 76

d) 56.99 only

Answer: b

Explanation: The function random.choice(a,b,c,d) returns a random number which is selected from a, b, c and d. The output can be either a, b, c or d. Hence the output of the snippet of code shown above can be either 10.4, 56.99 or 76.

5. What will be the output of the following Python function (random module has already been imported)?

random.choice('sun')

a) sun

b) u

c) either s, u or n

d) error

Answer: c

Explanation: The above function works with alphabets just as it does with numbers. The output of this expression will be either s, u or n.

6. What will be the output of the following Python function, assuming that the random module has already been imported?

random.uniform(3,4)

a) Error

b) Either 3 or 4

c) Any integer other than 3 and 4

d) Any decimal value between 3 and 4

Answer: d

Explanation: This question depicts the basic difference between the functions random.randint(a, b) and random.uniform(a, b). While random.randint(a,b) generates an integer between ‘a’ and ‘b’, including ‘a’ and ‘b’, the function random.uniform(a,b) generates a decimal value between ‘a’ and ‘b’.

7. What will be the output of the following Python function if the random module has already been imported?

random.randint(3.5,7)

a) Error

b) Any integer between 3.5 and 7, including 7

c) Any integer between 3.5 and 7, excluding 7

d) The integer closest to the mean of 3.5 and 7

Answer: a

Explanation: The function random.randint() does not accept a decimal value as a parameter. Hence the function shown above will throw an error.

8. Which of the following functions helps us to randomize the items of a list?

a) seed

b) randomise

c) shuffle

d) uniform

Answer: c

Explanation: The function shuffle, which is included in the random module, helps us to randomize the items of a list. This function takes the list as a parameter.

9. What will be the output of the following Python code?

random.seed(3)

random.randint(1,5)

2

random.seed(3)

random.randint(1,5)

a) 3

b) 2

c) Any integer between 1 and 5, including 1 and 5

d) Any integer between 1 and 5, excluding 1 and 5

Answer: b

Explanation: We use the seed function when we want to use the same random number once again in our program. Hence the output of the code shown above will be 2, since 2 was generated previously following which we used the seed function.

10. What is the interval of the value generated by the function random.random(), assuming that the random module has already been imported?

a) (0,1) b) (0,1] c) [0,1] d) [0,1)

Answer: d

Explanation: The function random.random() generates a random value in the interval [0,1), that is, including zero but excluding one.

11. What will be the output of the following Python code?

random.randrange(0,91,5)

a) 10 b) 18 c) 79 d) 95

Answer: a

Explanation: The function shown above will generate an output which is a multiple of 5 and is between 0 and 91. The only option which satisfies these criteria is 10. Hence the only possible output of this function is 10.

12. Both the functions randint and uniform accept \_\_\_\_\_\_\_\_\_\_\_\_ parameters.

a) 0 b) 1 c) 3 d) 2

Answer: c

Explanation: Both of these functions, that is, randint and uniform are included in the random module and both of these functions accept 3 parameters. For example: random.uniform(self,a,b) where ‘a’ and ‘b’ specify the range and self is an imaginary parameter.

13. The randrange function returns only an integer value.

a) True b) False

Answer: a

Explanation: The function randrange returns only an integer value. Hence this statement is true.

14. What will be the output of the following Python code?

random.randrange(1,100,10)

a) 32 b) 67 c) 91 d) 80

Answer: c

Explanation: The output of this function can be any value which is a multiple of 10, plus 1. Hence a value like 11, 21, 31, 41…91 can be the output. Also, the value should necessarily be between 1 and 100. The only option which satisfies this criteria is 91.

15. What will be the output of the following Python function, assuming that the random library has already been included?

random.shuffle[1,2,24]

a) Randomized list containing the same numbers in any order b) The same list, that is [1,2,24]

c) A list containing any random numbers between 1 and 24 d) Error

Answer: d

Explanation: The function shown above will result in an error because this is the incorrect syntax for the usage of the function shuffle(). The list should be previously declared and then passed to this function to get an output.

An example of the correct syntax:

>>> l=['a','b','c','d']

>>> random.shuffle(l)

>>> print(l)

1. What the does random.seed(3) return?

a) True b) None c) 3 d) 1

Answer: b

Explanation: The function random.seed() always returns a None.

2. Which of the following cannot be returned by random.randrange(4)?

a) 0 b) 3 c) 2.3 d) none of the mentioned

Answer: c

Explanation: Only integers can be returned.

3. Which of the following is equivalent to random.randrange(3)?

a) range(3) b) random.choice(range(0, 3))

c) random.shuffle(range(3)) d) random.select(range(3))

Answer: b

Explanation: It returns one number from the given range.

4. The function random.randint(4) can return only one of the following values. Which?

a) 4 b) 3.4 c) error d) 5

Answer: c

Explanation: Error, the function takes two arguments.

5. Which of the following is equivalent to random.randint(3, 6)?

a) random.choice([3, 6]) b) random.randrange(3, 6)

c) 3 + random.randrange(3) d) 3 + random.randrange(4)

Answer: d

Explanation: random.randint(3, 6) can return any one of 3, 4, 5 and 6.

6. Which of the following will not be returned by random.choice(“1 ,”)?

a) 1 b) (space) c) , d) none of the mentioned

Answer: d

Explanation: Any of the characters present in the string may be returned.

7. Which of the following will never be displayed on executing print(random.choice({0: 1, 2: 3}))?

a) 0 b) 1 c) KeyError: 1 d) none of the mentioned

Answer: a

Explanation: It will not print 0 but dict[0] i.e. 1 may be printed.

8. What does random.shuffle(x) do when x = [1, 2, 3]?

a) error

b) do nothing, it is a placeholder for a function that is yet to be implemented

c) shuffle the elements of the list in-place

d) none of the mentioned

Answer: c

Explanation: The elements of the list passed to it are shuffled in-place.

9. Which type of elements are accepted by random.shuffle()?

a) strings b) lists c) tuples d) integers

Answer: b

Explanation: Strings and tuples are immutable and an integer has no len().

10. What is the range of values that random.random() can return?

a) [0.0, 1.0] b) (0.0, 1.0] c) (0.0, 1.0) d) [0.0, 1.0)

Answer: d

Explanation: Any number that is greater than or equal to 0.0 and lesser than 1.0 can be returned.

**Sys Module**

<https://www.sanfoundry.com/python-questions-answers-sys-module/>

1. Which of the following functions can help us to find the version of python that we are currently working on?

a) sys.version

b) sys.version()

c) sys.version(0)

d) sys.version(1)

Answer: a

Explanation: The function sys.version can help us to find the version of python that we are currently working on. For example, 3.5.2, 2.7.3 etc. this function also returns the current date, time, bits etc along with the version.

2. Which of the following functions is not defined under the sys module?

a) sys.platform

b) sys.path

c) sys.readline

d) sys.argv

Answer: c

Explanation: The functions sys.platform, sys.path and sys.argv are defined under the sys module. The function sys.readline is not defined. However, sys.stdin.readline is defined.

3. The output of the functions len(“abc”) and sys.getsizeof(“abc”) will be the same.

a) True b) False

Answer: b

Explanation: The function len returns the length of the string passed, and hence it’s output will be 3. The function getsizeof, present under the sys module returns the size of the object passed. It’s output will be a value much larger than 3. Hence the above statement is false.

4. What will be the output of the following Python code, if the code is run on Windows operating system?

import sys

if sys.platform[:2]== 'wi':

print("Hello")

a) Error

b) Hello

c) No output

d) Junk value

Answer: b

Explanation: The output of the function sys.platform[:2] is equal to ‘wi’, when this code is run on windows operating system. Hence the output printed is ‘hello’.

5. What will be the output of the following Python code, if the sys module has already been imported?

sys.stdout.write("hello world")

a) helloworld

b) hello world10

c) hello world11

d) error

Answer: c

Explanation: The function shown above prints the given string along with the length of the string. Hence the output of the function shown above will be hello world11.

6. What will be the output of the following Python code?

import sys

sys.stdin.readline()

Sanfoundry

a) ‘Sanfoundry\n’

b) ‘Sanfoundry’

c) ‘Sanfoundry10’

d) Error

Answer: a

Explanation: The function shown above works just like raw\_input. Hence it automatically adds a ‘\n’ character to the input string. Therefore, the output of the function shown above will be: Sanfoundry\n.

7. What will be the output of the following Python code?

import sys

eval(sys.stdin.readline())

"India"

a) India5

b) India

c) ‘India\n’

d) ‘India’

Answer: d

Explanation: The function shown above evaluates the input into a string. Hence if the input entered is enclosed in double quotes, the output will be enclosed in single quotes. Therefore, the output of this code is ‘India’.

8. What will be the output of the following Python code?

import sys

eval(sys.stdin.readline())

Computer

a) Error

b) ‘Computer\n’

c) Computer8

d) Computer

Answer: a

Explanation: The code shown above will result in an error. This is because this particular function accepts only strings enclosed in single or double inverted quotes, or numbers. Since the string entered above is not enclosed in single or double inverted quotes, an error will be thrown.

9. What will be the output of the following Python code?

import sys

sys.argv[0]

a) Junk value

b) ‘ ‘

c) No output

d) Error

Answer: b

Explanation: The output of the function shown above will be a blank space enclosed in single quotes. Hence the output of the code shown above is ‘ ‘.

10. What will be the output of the following Python code?

import sys

sys.stderr.write(“hello”)

a) ‘hello’

b) ‘hello\n’

c) hello

d) hello5

Answer: d

Explanation: The code shown above returns the string, followed by the length of the string. Hence the output of the code shown above is hello5.

11. What will be the output of the following Python code?

import sys

sys.argv

a) ‘ ‘

b) [ ]

c) [‘ ‘]

d) Error

Answer: c

Explanation: The output of the code shown above is a blank space inserted in single quotes, which is enclosed by square brackets. Hence the output will be [‘ ‘].

12. To obtain a list of all the functions defined under sys module, which of the following functions can be used?

a) print(sys)

b) print(dir.sys)

c) print(dir[sys])

d) print(dir(sys))

Answer: d

Explanation: The function print(dir(sys)) helps us to obtain a list of all the functions defined under the sys module. The function can be used to obtain the list of functions under any given module in Python.

13. The output of the function len(sys.argv) is \_\_\_\_\_\_\_\_\_\_\_\_

a) Error

b) 1

c) 0

d) Junk value

Answer: b

Explanation: The output of the function sys.argv is [‘ ‘]. When we execute the function len([‘ ‘]), the output is 1. Hence the output of the function len(sys.argv) is also 1.

**Operating System**

<https://www.sanfoundry.com/python-exam-questions-answers/>

1. What does os.name contain?

a) the name of the operating system dependent module imported

b) the address of the module os

c) error, it should’ve been os.name()

d) none of the mentioned

Answer: a

Explanation: It contains the name of the operating system dependent module imported such as ‘posix’, ‘java’ etc.

2. What does print(os.geteuid()) print?

a) the group id of the current process

b) the user id of the current process

c) both the group id and the user of the current process

d) none of the mentioned

Answer: b

Explanation: os.geteuid() gives the user id while the os.getegid() gives the group id.

3. What does os.getlogin() return?

a) name of the current user logged in

b) name of the superuser

c) gets a form to login as a different user

d) all of the mentioned

Answer: a

Explanation: It returns the name of the user who is currently logged in and is running the script.

4. What does os.close(f) do?

a) terminate the process f

b) terminate the process f if f is not responding

c) close the file descriptor f

d) return an integer telling how close the file pointer is to the end of file

Answer: c

Explanation: When a file descriptor is passed as an argument to os.close() it will be closed.

5. What does os.fchmod(fd, mode) do?

a) change permission bits of the file

b) change permission bits of the directory

c) change permission bits of either the file or the directory

d) none of the mentioned

Answer: a

Explanation: The arguments to the function are a file descriptor and the new mode.

6. Which of the following functions can be used to read data from a file using a file descriptor?

a) os.reader()

b) os.read()

c) os.quick\_read()

d) os.scan()

Answer: b

Explanation: None of the other functions exist.

7. Which of the following returns a string that represents the present working directory?

a) os.getcwd()

b) os.cwd()

c) os.getpwd()

d) os.pwd()

Answer: a

Explanation: The function getcwd() (get current working directory) returns a string that represents the present working directory.

8. What does os.link() do?

a) create a symbolic link

b) create a hard link

c) create a soft link

d) none of the mentioned

Answer: b

Explanation: os.link(source, destination) will create a hard link from source to destination.

9. Which of the following can be used to create a directory?

a) os.mkdir()

b) os.creat\_dir()

c) os.create\_dir()

d) os.make\_dir()

Answer: a

Explanation: The function mkdir() creates a directory in the path specified.

10. Which of the following can be used to create a symbolic link?

a) os.symlink()

b) os.symb\_link()

c) os.symblin()

d) os.ln()

Answer: a

Explanation: It is the function that allows you to create a symbolic link.

**Turtle Module – 1**

<https://www.sanfoundry.com/python-questions-answers-turtle-module-1/>

1. What will be the output shape of the following Python code?

import turtle

t=turtle.Pen()

for i in range(0,4):

t.forward(100)

t.left(120)

a) square b) rectangle c) triangle d) kite

Answer: c

Explanation: According to the code shown above, 4 lines will be drawn. Three lines will be in the shape of a triangle. The fourth line will trace the base, which is already drawn. Hence the base will be slightly thicker than the rest of the lines. However there will be no change in the shape due to this extra line. Hence the output shape will be a triangle.

2. The number of lines drawn in each case, assuming that the turtle module has been imported:

Case 1:

for i in range(0,10):

turtle.forward(100)

turtle.left(90)

Case 2:

for i in range(1,10):

turtle.forward(100)

turtle.left(90)

a) 10, 9

b) 9, 10

c) 9, 9

d) 10, 10

Answer: a

Explanation: The number of lines drawn in the first case is 10, while that in the second case is 9.

3. The command which helps us to reset the pen (turtle):

a) turtle.reset

b) turtle.penreset

c) turtle.penreset()

d) turtle.reset()

Answer: d

Explanation: The command turtle.reset() helps us to reset the pen. After the execution of this command, we get a blank page with an arrow on it. We can then perform any desired operation on this page.

4. Fill in the blank such that the following Python code results in the formation of an inverted, equilateral triangle.

import turtle

t=turtle.Pen()

for i in range(0,3):

t.forward(150)

t.right(\_\_\_\_\_)

a) -60

b) 120

c) -120

d) 60

Answer: b

Explanation: An angle of -120 will result in the formation of an upright, equilateral triangle. An angle of 120 will result in the formation of an inverted triangle. The angles of 60 and -60 do not result in the formation of a triangle.

5. What will be the output shape of the following Python code?

import turtle

t=turtle.Pen()

for i in range(1,4):

t.forward(60)

t.left(90)

a) Rectangle b) Trapezium c) Triangle d) Square

Answer: d

Explanation: The code shown above will result in the formation of a square, with each of side 60.

6. What will be the output of the following Python code?

import turtle

t=turtle.Pen()

for i in range(0,4):

t.forward(100)

t.left(90)

t.penup()

t.left(90)

t.forward(200)

for i in range(0,4):

t.forward(100)

t.left(90)

a) Error

b) 1 square

c) 2 squares, at a separation of100 units, joined by a straight line

d) 2 squares, at a separation of 100 units, without a line joining them

Answer: b

Explanation: The output of the code shown above will be a single square. This is because the function t.penup() is used to lift the pen after the construction of the first square. However, the function t.pendown() has not been used to put the pen back down. Hence, the output shape of this code is one square, of side 100 units.

7. Which of the following functions does not accept any arguments?

a) position b) fillcolor c) goto d) setheading()

Answer: a

Explanation: The functions fillcolor(), goto() and setheading() accept arguments, whereas the function position() does not accept any arguments. The function position() returns the current position of the turtle.

8. What will be the output of the following Python code?

import turtle

t=turtle.Pen()

t.goto(300,9)

t.position()

a) 300.00, 9.00

b) 9, 300

c) 300, 9

d) 9.00, 300.00

Answer: a

Explanation: The goto functions takes the arrow to the position specified by the user as arguments. The position function returns the current position of the arrow. Hence the output of the code shown above will be: 300.00, 9.00.

9. What will be the output of the following Python code?

import turtle

t=turtle.Pen()

for i in range(0,5):

t.left(144)

t.forward(100)

a) Trapezium b) Parallelepiped c) Tetrahedron d) Star

Answer: d

Explanation: It is clear from the above code that 5 lines will be drawn on the canvas, at an angle of 144 degrees. The only shape which fits this description is star. Hence the output of the code shown above is star.

10. What will be the output of the following Python functions?

import turtle

t=turtle.Pen()

for i in range(0,3):

t.forward(100)

t.left(120)

t.back(100)

for i in range(0,3):

t.forward(100)

t.left(120)

a) Error

b) Two triangles, joined by a straight line

c) Two triangles, joined at one vertex

d) Two separate triangles, not connected by a line

Answer: c

Explanation: The output of the code shown above is two equilateral triangles (of side 100 units), joined at the vertex.

1. What will be the output of the following Python code?

import turtle

t=turtle.Pen()

t.color(0,0,1)

t.begin\_fill()

t.circle(15)

t.end\_fill()

a) Error

b) A circle filled in with the colour red

c) A circle filled in with the colour blue

d) A circle filled in with the colour green

Answer: c

Explanation: The function t.colour(0, 0, 1) is used to fill in the colour blue into any given shape. Hence the output of the code shown above will be a circle filled in with the colour blue.

2. Which of the following functions can be used to make the arrow black?

a) turtle.color(0,1,0)

b) turtle.color(1,0,0)

c) turtle.color(0,0,1)

d) turtle.color(0,0,0)

Answer: d

Explanation: The function turtle.color(0,0,0) can change the colour of the arrow. The function turtle.color(0,1,0) will make the arrow green. The function turtle.color(1,0,0) will make the arrow red. The function turtle.color(0,0,1) will make the arrow blue. The function turtle.color(0,0,0) will make the arrow black.

3. What will be the output of the following Python code?

import turtle

t=turtle.Pen()

t.color(1,1,1)

t.begin\_fill()

for i in range(0,3):

t.forward(100)

t.right(120)

t.end\_fill()

a) Blank page b) A triangle filled in with the colour yellow

c) A triangle which is not filled in with any colour d) Error

Answer: a

Explanation: The code shown above will result in a blank page. This is because the command turtle.color(1,1,1) eliminates the arrow from the page. Hence all the commands after this command are ineffective.

4. What will be the output of the following Python code?

import turtle

t=turtle.Pen()

t.color(0,1,0)

t.begin\_fill()

for i in range(0,4):

t.forward(100)

t.right(90)

a) A square filled in with the colour green b) A square outlined with the colour green

c) Blank canvas d) Error

Answer: c

Explanation: The output shape of the code shown above is a square, outlined with the colour green, but not filled in with any colour. This is because we have not used the command t.end\_fill() at the end.

5. In which direction is the turtle pointed by default?

a) North b) South c) East d) West

Answer: c

Explanation: By default, the turtle is pointed towards the east direction. We can change the direction of the turtle by using certain commands. However, whenever the turtle is reset, it points towards east.

6. The command used to set only the x coordinate of the turtle at 45 units is:

a) reset(45)

b) setx(45)

c) xset(45)

d) xreset(45)

Answer: b

Explanation: The command setx(45) is used to set the x coordinate of the turtle. Similarly, the command sety() is used to set the y coordinate of the turtle. The function reset() takes two values as arguments, one for the x-coordinate and the other for the y-coordinate.

7. Which of the following functions returns a value in degrees, counterclockwise from the horizontal right?

a) heading() b) degrees() c) position() d) window\_height()

Answer: a

Explanation: The function heading() returns the heading of the turtle, which is a value in degrees counterclockwise from the horizontal right. This measure will be in radians if radians() has been called.

8. What will be the output of the following Python code?

import turtle

t=turtle.Pen()

t.right(90)

t.forward(100)

t.heading()

a) 0.0

b) 90.0

c) 270.0

d) 360.0

Answer: c

Explanation: The output of the code shown above will be 270.0. The function heading() returns the heading of the turtle, a value in degrees, counterclockwise from the horizontal right. The output shape of this code is a straight line pointing downwards.

9. What will be the output of the following Python code?

import turtle

t=turtle.Pen()

t.clear()

t.isvisible()

a) Yes b) True c) No d) False

Answer: b

Explanation: The function t.clear() returns a blank canvas, without changing the position of the turtle. Since the turtle is visible on the blank canvas, the output of this code is: Yes.

10. What will be the output of the following Python code?

import turtle

t=turtle.Pen()

t.forward(100)

t.left(90)

t.clear()

t.position()

a) 0.00, 90.00

b) 0.00, 0.00

c) 100.00, 90.00

d) 100.00, 100.00

Answer: d

Explanation: The output of the code shown above is 100.00, 100.00. The function clear() is used to erase the entire canvas and redraw the turtle. However, the position of the turtle is not changed.

1. Which of the following functions results in an error?

a) turtle.shape(“turtle”) b) turtle.shape(“square”)

c) turtle.shape(“triangle”) d) turtle.shape(“rectangle”)

Answer: d

Explanation: The functions shown above will change the arrow to the shape mentioned. The functions turtle.shape(“turtle”), turtle.shape(“square”) and turtle.shape(“triangle”) are valid whereas the function turtle.shape(“rectangle”) is invalid.

2. What will be the output of the following Python code?

import turtle

t=turtle.Pen

t.tilt(75)

t.forward(100)

a) A straight line of 100 units tiled at 75 degrees from the horizontal

b) A straight line of 100 units tilted at 15 degrees from the horizontal

c) A straight line of 100 units lying along the horizontal

d) Error

Answer: c

Explanation: The function turtle.tilt(75) will tilt the turtle. But the straight line (of 100 units) is drawn along the horizontal. Hence the output of the code shown above is a straight line of 100 units lying along the horizontal.

3. What will be the output of the following Python code?

import turtle

t=turtle.Pen()

t.backward(100)

t.penup()

t.right(45)

t.isdown()

a) True b) False c) Yes d) No

Answer: b

Explanation: In the code shown above, we have used the function t.penup() to life the pen from the canvas. However, we have not used the function t.pendown() to keep the pen back down. The function turtle.isdown() returns True if the pen is down and False if the pen is not down. Hence the output is False.

4. The function used to alter the thickness of the pen to ‘x’ units:

a) turtle.width(x) b) turtle.span(x) c) turtle.girth(x) d) turtle.thickness(x)

Answer: a

Explanation: The function turtle.width(x) is used to alter the thickness of the pen to ‘x’ units. The function turtle.span(x), turtle.girth(x) and turtle.thickness(x) are invalid.

5. What will be the output of the following Python code if the system date is 18th June, 2017 (Sunday)?

import turtle

t=turtle.Pen()

t.goto(100,0)

t.towards(0,0)

a) 0.0

b) 180.0

c) 270.0

d) 360.0

Answer: b

Explanation: The function t.towards(x,y) returns the angle between the line to the line specified by (x,y). Hence the output will be 180.0.

6. What will be the output of the following Python code?

import turtle

t=turtle.Pen()

t.position()

(100.00,0.00)

t.goto(100,100)

t.distance(100,0)

a) 0.0

b) Error

c) 100.0, 100.0

d) 100.0

Answer: d

Explanation: The distance() function returns the distance between the turtle to the given vector. Hence the output of the code shown above is 100.0.

7. The output of the following Python code will result in a shape similar to the alphabet \_\_\_\_\_\_\_\_\_\_\_

import turtle

t=turtle.Turtle()

t1=turtle.Turtle()

t.left(45)

t1.left(135)

t.forward(100)

t1.forward(100)

a) V

b) Inverted V

c) X

d) T

Answer: a

Explanation: In the code shown above, two pens have been used to create a shape similar to the alphabet ‘V’. The angle between the two straight lines is 90 degrees.

8. The output of the following Python code is similar to the alphabet \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

import turtle

t=turtle.Pen()

t1=turtle.Pen()

t2=turtle.Pen()

t.forward(100)

t1.forward(100)

t2.forward(100)

t1.left(90)

t1.forward(75)

t2.right(90)

t2.forward(75)

a) X

b) N

c) T

d) M

Answer: c

Explanation: In the above code, three pens have been used to create a shape similar to the letter ‘T’. All the three straight lines are mutually perpendicular.

9. The following Python code will result in an error.

import turtle

t=turtle.Pen()

t.speed(-45)

t.circle(30)

a) True b) False

Answer: b

Explanation: Although a negative speed is not possible, the code shown above does not result in an error. Hence, the answer is False.

10. What will be the output of the following Python code?

import turtle()

t=turtle.Pen()

t.goto(50,60)

t1=t.clone()

t1.ycor()

a) 0.0

b) 50.0

c) 60.0

d) Error

Answer: c

Explanation: The function clone() is used to create a clone of the turtle, having the same properties such as position, coordinates etc. Hence, the properties of the t and t1 are the same in the code shown above. The function ycor() returns the y-coordinate of the turtle. Hence the output of the code is 60.0.

11. What will be the output shape of the following Python code?

import turtle

t=turtle.Pen()

for i in range(0,6):

t.forward(100)

t.left(60)

a) Hexagon b) Octagon c) Pentagon d) Heptagon

Answer: a =>The code shown above creates a six-sided polygon. The output shape of the code shown above is will be a hexagon.

12. What will be the output of the following Python code?

import turtle

t=turtle.Pen()

t.resizemode(“sanfoundry”)

t.resizemode()

a) user b) auto c) nonresize d) error

Answer: c =>When not explicitly specified as auto or user, no adaption of the turtle’s appearance takes place and the mode is ‘noresize’. Hence the output of the code is: noresize.

**Pickle Module**

<https://www.sanfoundry.com/python-questions-answers-pickle-module/>

1. The process of pickling in Python includes:

a) conversion of a list into a datatable

b) conversion of a byte stream into Python object hierarchy

c) conversion of a Python object hierarchy into byte stream

d) conversion of a datatable into a list

Answer: c

Explanation: Pickling is the process of sterilizing a Python object, that is, conversion of a byte stream into Python object hierarchy. The reverse of this process is known as unpickling.

2. To sterilize an object hierarchy, the \_\_\_\_\_\_\_\_\_\_\_\_\_ function must be called. To desterilize a data stream, the \_\_\_\_\_\_\_\_\_\_\_\_\_\_ function must be called.

a) dumps(), undumps() b) loads(), unloads()

c) loads(), dumps() d) dumps(), loads()

Answer: d

Explanation: To sterilize an object hierarchy, the dumps() function must be called. To desterilize a data stream, the loads function must be called.

3. Pick the correct statement regarding pickle and marshal modules.

a) The pickle module supports primarily .pyc files whereas marshal module is used to sterilize Python objects

b) The pickle module keeps track of the objects that have already been sterilized whereas the marshal module does not do this

c) The pickle module cannot be used to sterilize user defined classes and their instances whereas marshal module can be used to perform this task

d) The format of sterilization of the pickle module is not guaranteed to be supported across all versions of Python. The marshal module sterilization is compatible across all the versions of Python

Answer: b

Explanation: The correct statement from among the above options is that the python module keeps track of the objects that have already been sterilized whereas the marshal module does not do this. The rest of the statements are incorrect.

4. What will be the output of the following Python code?

pickle.HIGHEST\_PROTOCOL

a) 4

b) 5

c) 3

d) 6

Answer: a

Explanation: There are five protocol versions available of the pickle module, namely, 0, 1, 2, 3 and 4. In the code shown above, the highest protocol version is returned, that is, 4.

5. Which of the following Python codes will result in an error?

object = ‘a’

a) >>> pickle.dumps(object)

b) >>> pickle.dumps(object, 3)

c) >>> pickle.dumps(object, 3, True)

d) >>> pickle.dumps(‘a’, 2)

Answer: c

Explanation: The function pickle.dumps requires either 1 or 2 arguments. If any other number of arguments are passed to it, it results in an error. An error is thrown even when no arguments are passed to it.

6. Which of the following functions can be used to find the protocol version of the pickle module currently being used?

a) pickle.DEFAULT b) pickle.CURRENT

c) pickle.CURRENT\_PROTOCOL d) pickle.DEFAULT\_PROTOCOL

Answer: d

Explanation: The function pickle.DEFAULT\_PROTOCOL can be used to find the protocol version of the pickle module currently being used by the system.

7. The output of the following two Python codes is exactly the same.

object

'a'

CODE 1

>>> pickle.dumps('a', 3)

CODE 2

>>> pickle.dumps(object, 3)

a) True b) False

Answer: a

Explanation: The two codes shown above result in the same output, that is, b’\x80\x03X\x01\x00\x00\x00aq\x00.’ Hence this statement is true.

8. Which of the following functions can accept more than one positional argument?

a) pickle.dumps b) pickle.loads

c) pickle.dump d) pickle.load

Answer: a

Explanation: The functions pickle.loads, pickle.dump and pickle.load accept only one argument. The function pickle.dumps can accept more than one argument.

9. Which of the following functions raises an error when an unpicklable object is encountered by Pickler?

a) pickle.PickleError b) pickle.PicklingError

c) pickle.UnpickleError d) pickle.UnpicklingError

Answer: b

Explanation: The function pickle.PicklingError raises an error when an unpickable object is encountered by Pickler.

10. The pickle module defines \_\_\_\_\_\_ exceptions and exports \_\_\_\_\_\_\_ classes.

a) 2, 3

b) 3, 4

c) 3, 2

d) 4, 3

Answer: c

Explanation: The pickle module defines three exceptions, namely, pickle.PickleError, pickle.PicklingError, pickle.UnpickleError and exports two classes, namely, pickle.Pickler and pickle.Unpickler.

11. Which of the following cannot be pickled?

a) Functions which are defined at the top level of a module with lambda

b) Functions which are defined at the top level of a module with def

c) Built-in functions which are defined at the top level of a module

d) Classes which are defined at the top level of a module

Answer: a

Explanation: Functions which are defined at the top level of a module with lambda cannot be pickled.

12. If \_\_getstate\_\_() returns \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ the \_\_setstate\_\_() module will not be called on pickling.

a) True value b) False value

c) ValueError d) OverflowError

Answer: b

Explanation: If getstate\_\_() returns a false value, the \_\_setstate\_\_() module will not be called on pickling.

13. Lambda functions cannot be pickled because:

a) Lambda functions only deal with binary values, that is, 0 and 1

b) Lambda functions cannot be called directly

c) Lambda functions cannot be identified by the functions of the pickle module

d) All lambda functions have the same name, that is, <lambda>

Answer: d

Explanation: Lambda functions cannot be pickled because all the lambda functions have the same name, that is, <lambda>

14. The module \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is a comparatively faster implementation of the pickle module.

a) cPickle b) nPickle c) gPickle d) tPickle

Answer: a

Explanation: The module cPickle is a comparatively faster implementation of the pickle module.

15. The copy module uses the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ protocol for shallow and deep copy.

a) pickle b) marshal c) shelve d) copyreg

Answer: a

Explanation: The copy module uses the pickle protocol for shallow and deep copy.

**Regular Expressions**

<https://www.sanfoundry.com/python-questions-answers-regular-expressions/>

1. Which module in Python supports regular expressions?

a) re

b) regex

c) pyregex

d) none of the mentioned

Answer: a

Explanation: re is a part of the standard library and can be imported using: import re.

2. Which of the following creates a pattern object?

a) re.create(str) b) re.regex(str)

c) re.compile(str) d) re.assemble(str)

Answer: c

Explanation: It converts a given string into a pattern object.

3. What does the function re.match do?

a) matches a pattern at the start of the string

b) matches a pattern at any position in the string

c) such a function does not exist

d) none of the mentioned

Answer: a

Explanation: It will look for the pattern at the beginning and return None if it isn’t found.

4. What does the function re.search do?

a) matches a pattern at the start of the string b) matches a pattern at any position in the string

c) such a function does not exist d) none of the mentioned

Answer: b

Explanation: It will look for the pattern at any position in the string.

5. What will be the output of the following Python code?

sentence = 'we are humans'

matched = re.match(r'(.\*) (.\*?) (.\*)', sentence)

print(matched.groups())

a) (‘we’, ‘are’, ‘humans’) b) (we, are, humans)

c) (‘we’, ‘humans’) d) ‘we are humans’

Answer: a

Explanation: This function returns all the subgroups that have been matched.

6. What will be the output of the following Python code?

sentence = 'we are humans'

matched = re.match(r'(.\*) (.\*?) (.\*)', sentence)

print(matched.group())

a) (‘we’, ‘are’, ‘humans’) b) (we, are, humans)

c) (‘we’, ‘humans’) d) ‘we are humans’

Answer: d

Explanation: This function returns the entire match.

7. What will be the output of the following Python code?

sentence = 'we are humans'

matched = re.match(r'(.\*) (.\*?) (.\*)', sentence)

print(matched.group(2))

a) ‘are’ b) ‘we’ c) ‘humans’ d) ‘we are humans’

Answer: c

Explanation: This function returns the particular subgroup.

8. What will be the output of the following Python code?

sentence = 'horses are fast'

regex = re.compile('(?P<animal>\w+) (?P<verb>\w+) (?P<adjective>\w+)')

matched = re.search(regex, sentence)

print(matched.groupdict())

a) {‘animal’: ‘horses’, ‘verb’: ‘are’, ‘adjective’: ‘fast’}

b) (‘horses’, ‘are’, ‘fast’)

c) ‘horses are fast’

d) ‘are’

Answer: a

Explanation: This function returns a dictionary that contains all the matches.

9. What will be the output of the following Python code?

sentence = 'horses are fast'

regex = re.compile('(?P<animal>\w+) (?P<verb>\w+) (?P<adjective>\w+)')

matched = re.search(regex, sentence)

print(matched.groups())

a) {‘animal’: ‘horses’, ‘verb’: ‘are’, ‘adjective’: ‘fast’}

b) (‘horses’, ‘are’, ‘fast’)

c) ‘horses are fast’

d) ‘are’

Answer: b

Explanation: This function returns all the subgroups that have been matched.

10. What will be the output of the following Python code?

sentence = 'horses are fast'

regex = re.compile('(?P<animal>\w+) (?P<verb>\w+) (?P<adjective>\w+)')

matched = re.search(regex, sentence)

print(matched.group(2))

a) {‘animal’: ‘horses’, ‘verb’: ‘are’, ‘adjective’: ‘fast’}

b) (‘horses’, ‘are’, ‘fast’)

c) ‘horses are fast’

d) ‘are’

Answer: d

Explanation: This function returns the particular subgroup.

<https://www.sanfoundry.com/python-questions-answers-regular-expressions-1/>

1. The character Dot (that is, ‘.’) in the default mode, matches any character other than \_\_\_\_\_\_\_\_\_\_\_\_\_

a) caret b) ampersand c) percentage symbol d) newline

Answer: d

Explanation: The character Dot (that is, ‘,’) in the default mode, matches any character other than newline. If DOTALL flag is used, then it matches any character other than newline.

2. The expression a{5} will match \_\_\_\_\_\_\_\_\_\_\_\_\_ characters with the previous regular expression.

a) 5 or less b) exactly 5 c) 5 or more d) exactly 4

Answer: b

Explanation: The character {m} is used to match exactly m characters to the previous regular expression. Hence the expression a{5} will match exactly 5 characters and not less than that.

3. \_\_\_\_\_\_\_\_ matches the start of the string.

\_\_\_\_\_\_\_\_ matches the end of the string.

a) ‘^’, ‘$’

b) ‘$’, ‘^’

c) ‘$’, ‘?’

d) ‘?’, ‘^’

Answer: a => ‘^’ (carat) matches the start of the string.

‘$’ (dollar sign) matches the end of the string.

4. Which of the following will result in an error?

a)>>> p = re.compile("d")

>>> p.search("door")

b) >>> p = re.escape(‘hello’)

c) >>> p = re.subn()

d) >>> p = re.purge()

Answer: c =>The function re.subn() will result in an error. This is because subn() requires 3 positional arguments while we have entered none.

5. What will be the output of the following Python code?

re.split('\W+', 'Hello, hello, hello.')

a) [‘Hello’, ‘hello’, ‘hello.’]

b) [‘Hello, ‘hello’, ‘hello’]

c) [‘Hello’, ‘hello’, ‘hello’, ‘.’]

d) [‘Hello’, ‘hello’, ‘hello’, ”]

Answer: d

Explanation: In the code shown above, the function split() splits the string based on the pattern given as an argument in the parenthesis. Note: split will never split a string on an empty pattern match. Hence the output of this code is: [‘Hello’, ‘hello’, ‘hello’, ”].

6. What will be the output of the following Python function?

re.findall("hello world", "hello", 1)

a) [“hello”] b) [ ] c) hello d) hello world

Answer: b => The function findall returns the word matched if and only if both the pattern and the string match completely, that is, they are exactly the same. Observe the example shown below:

>>> re.findall(“hello”, “hello”, 1) The output is: [‘hello’] Hence the output of the code shown in this question is [].

7. Choose the function whose output can be: <\_sre.SRE\_Match object; span=(4, 8), match=’aaaa’>.

a) >>> re.search(‘aaaa’, “alohaaaa”, 0) b) >>> re.match(‘aaaa’, “alohaaaa”, 0)

c) >>> re.match(‘aaa’, “alohaaa”, 0) d) >>> re.search(‘aaa’, “alohaaa”, 0)

Answer: a

Explanation: The output shown above is that of a search function, whose pattern is ‘aaaa’ and the string is that of 8 characters. The only option which matches all these criteria is:

>>> re.search(‘aaaa’, “alohaaaa”, 0)

8. Which of the following functions clears the regular expression cache?

a) re.sub() b) re.pos() c) re.purge() d) re.subn()

Answer: c => The function which clears the regular expression cache is re.purge(). Note that this function takes zero positional arguments.

9. What will be the output of the following Python code?

import re

re.ASCII

a) 8 b) 32 c) 64 d) 256

Answer: d

Explanation: The expression re.ASCII returns the total number of ASCII characters that are present, that is 256. This can also be abbreviated as re.A, which results in the same output (that is, 256).

10. Which of the following functions results in case insensitive matching?

a) re.A b) re.U c) re.I d) re.X

Answer: c => The function re.I (that is, re.IGNORECASE) results in case-insensitive matching. That is, expressions such as [A-Z] will match lowercase characters too.

1. What will be the output of the following Python code?

re.compile('hello', re.X)

a) [‘h’, ‘e’, ‘l’, ‘l’, ‘o’] b) re.compile(‘hello’, re.VERBOSE)

c) Error d) Junk value

Answer: b

Explanation: The compile function compiles a pattern of regular expression into an object of regular expression. Re.X is a flag which is also used as re.VERBOSE. Hence the output of this code is: re.compile(‘hello’, re.VERBOSE).

2. What will be the output of the following Python code?

re.split('[a-c]', '0a3B6', re.I)

a) Error

b) [‘a’, ‘B’]

c) [‘0’, ‘3B6’]

d) [‘a’]

Answer: c

Explanation: The function re.split() splits the string on the basis of the pattern given in the parenthesis. Since we have used the flag e.I (that is, re.IGNORECASE), the output is: [‘0’, ‘3B6’].

3. What will be the output of the following Python code?

re.sub('morning', 'evening', 'good morning')

a) ‘good evening’ b) ‘good’ c) ‘morning’ d) ‘evening’

Answer: a

Explanation: The code shown above first searches for the pattern ‘morning’ in the string ‘good morning’ and then replaces this pattern with ‘evening’. Hence the output of this code is: ‘good evening’.

4. The function re.error raises an exception if a particular string contains no match for the given pattern.

a) True b) False

Answer: b => The function re.error raises an exception when a string passed to one of its functions here is not a valid regular expression. It does not raise an exception if a particular string does not contain a match for the given pattern.

5. What will be the output of the following Python code?

re.escape('new\*\*world')

a) ‘new world’

b) ‘new\\\*\\\*world’

c) ‘\*\*’

d) ‘new’, ‘\*’, ‘\*’, ‘world’

Answer: b => The function re.escape escapes all the characters in the pattern other than ASCII letters and numbers. Hence the output of the code shown above is: ‘new\\\*\\\*world’.

6. What will be the output of the following Python code?

re.fullmatch('hello', 'hello world')

a) No output

b) []

c) <\_sre.SRE\_Match object; span=(0, 5), match='hello'>

d) Error

Answer: a

Explanation: The function re.fullmatch applies the pattern to the entire string and returns an object if match is found and none if match in not found. In the code shown above, match is not found. Hence there is no output.

7. Choose the option wherein the two choices do not refer to the same option.

a)

re.I

re.IGNORECASE

b)

re.M

re.MULTILINE

c)

re.X

re.VERBOSE

d)

re.L

re.LOWERCASE

Answer: d => The function re.L is also written as re.LOCALE. There is no function such as re.LOWERCASE in the re module of Python.

8. The difference between the functions re.sub and re.subn is that re.sub returns a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ whereas re.subn returns a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

a) string, list b) list, tuple

c) string, tuple d) tuple, list

Answer: c => The difference the functions re.sub and re.subn is that re.sub returns a string whereas re.subn returns a tuple.

9. What will be the output of the following Python code?

re.split('mum', 'mumbai\*', 1)

a) Error

b) [”, ‘bai\*’]

c) [”, ‘bai’]

d) [‘bai\*’]

Answer: b

Explanation: The code shown above splits the string based on the pattern given as an argument. Hence the output of the code is: [”, ‘bai\*’].

10. What will be the output of the following Python code?

re.findall('good', 'good is good')

re.findall('good', 'bad is good')

a)

[‘good’, ‘good’]

[‘good’]

b)

(‘good’, ‘good’)

(good)

c)

(‘good’)

(‘good’)

d)

[‘good’]

[‘good’]

Answer: a

Explanation: The function findall returns a list of all the non overlapping matches in a string. Hence the output of the first function is: [‘good’, ‘good’] and that of the second function is: [‘good’].

1. What will be the output of the following Python code?

re.split(r'(n\d)=', 'n1=3.1, n2=5, n3=4.565')

a) Error

b) [”, ‘n1’, ‘3.1, ‘, ‘n2’, ‘5, ‘, ‘n3’, ‘4.565’]

c) [‘n1’, ‘3.1, ‘, ‘n2’, ‘5, ‘, ‘n3’, ‘4.565’]

d) [‘3.1, ‘, ‘5, ‘, ‘4.565’]

Answer: b

Explanation: In the snippet of code shown above, we extract the numbers as a list of floating point values, including the initial empty string. The example shown above demonstrate how groups in the regular expression influence the result of re.split. Hence the output of the code shown above is:

[”, ‘n1’, ‘3.1, ‘, ‘n2’, ‘5, ‘, ‘n3’, ‘4.565’].

2. The function of re.search is \_\_\_\_\_\_\_\_\_\_

a) Matches a pattern at the start of the string

b) Matches a pattern at the end of the string

c) Matches a pattern from any part of a string

d) Such a function does not exist

Answer: c

Explanation: The re module of Python consists of a function re.search. It’s function is to match a pattern from anywhere in a string.

3. Which of the following functions creates a Python object?

a) re.compile(str) b) re.assemble(str)

c) re.regex(str) d) re.create(str)

Answer: a

Explanation: The function re.compile(srt) compiles a pattern of regular expression into an object of regular expression. Hence re.compile(str) is the only function from the above options which creates an object.

4. Which of the following pattern matching modifiers permits whitespace and comments inside the regular expression?

a) re.L

b) re.S

c) re.U

d) re.X

Answer: d

Explanation: The modifier re.X allows whitespace and comments inside the regular expressions.

5. What will be the output of the following Python code?

s = 'welcome home'

m = re.match(r'(.\*)(.\*?)', s)

print(m.group())

a) (‘welcome’, ‘home’)

b) [‘welcome’, ‘home’]

c) welcome home

d) [‘welcome’ // ‘home’ ]

Answer: c

Explanation: The code shown above shows the function re.match combined with the use of special characters. Hence the output of this code is: welcome home.

6. The function of re.match is \_\_\_\_\_\_\_\_\_\_\_\_

a) Error

b) Matches a pattern anywhere in the string

c) Matches a pattern at the end of the string

d) Matches a pattern at the start of the string

Answer: d

Explanation: The function of re.match matches a pattern at the start of the string.

7. The special character \B matches the empty string, but only when it is \_\_\_\_\_\_\_\_\_\_\_\_\_

a) at the beginning or end of a word b) not at the beginning or end of a word

c) at the beginning of the word d) at the end of the word

Answer: b

Explanation: The special character \B matches the empty string, but only when it is not at the beginning or end of a word.

8. What will be the output of the following Python code?

import re

s = "A new day"

m = re.match(r'(.\*)(.\*?)', s)

print(m.group(2))

print(m.group(0))

a)

No output

A new day

b)

No output

No output

c)

[‘A’, ‘new’, ‘day’]

(‘A’, ‘new’, ‘day’)

d)

Error

[‘A’, ‘new’, ‘day’]

Answer: a

Explanation: The code shown above demonstrates the use of the function re.match, with different arguments given to the group method. Hence the first function does not return any output whereas the second function returns the output: A new day

9. Which of the following special characters matches a pattern only at the end of the string?

a) \B

b) \X

c) \Z

d) \A

Answer: c => \B matches a pattern which is not at the beginning or end of a string. \X refers to re.VERBOSE. \A matches a pattern only at the start of a string. \Z matches a pattern only at the end of a string.

10. The output of the following two Python codes are the same.

p = re.compile('hello')

r = p.match('hello everyone')

print(r.group(0))

r = re.match('hello', 'hello everyone')

print(r.group(0))

a) True b) False

Answer: a

Explanation: The two codes shown above are equivalent. Both of these codes result in the same output, that is: hello. Hence this statement is true.

11. What will be the output of the following Python code?

re.match('sp(.\*)am', 'spam')

a) <\_sre.SRE\_Match object; span=(1, 4), match=’spam’>

b) <\_sre.SRE\_Match object; span=(0, 4), match=’spam’>

c) No output

d) Error

Answer: b

Explanation: The code shown above demonstrates the function re.match, combined with a special character. The output of the code shown is: <\_sre.SRE\_Match object; span=(0, 4), match=’spam’>

12. Which of the following special characters represents a comment (that is, the contents of the parenthesis are simply ignores)?

a) (?:…)

b) (?=…)

c) (?!…)

d) (?#…)

Answer: d

Explanation: The special character (?#…) represent a comment, that is, the contents of the parenthesis are simply ignored.

13. Which of the codes shown below results in a match?

a) re.match(‘George(?=Washington)’, ‘George Washington’)

b) re.match(‘George(?=Washington)’, ‘George’)

c) re.match(‘George(?=Washington)’, ‘GeorgeWashington’)

d) re.match(‘George(?=Washington)’, ‘Georgewashington’)

Answer: c

Explanation: The code shown above demonstrates the use of the function re.match, along with the special character ?=. This results in a match only when ‘George’ is immediately followed by ‘Washington’. Also, we have not used the module to ignore case. Hence the match is case-sensitive. Therefore the only option which results in a match is:

re.match(‘George(?=Washington)’, ‘GeorgeWashington’)

1. What will be the output of the following Python code?

re.split(r'(a)(t)', 'Maths is a difficult subject')

a) [‘M a t h s i s a d i f f i c u l t s u b j e c t’]

b) [‘Maths’, ‘is’, ‘a’, ‘difficult’, ‘subject’]

c) ‘Maths is a difficult subject’

d) [‘M’, ‘a’, ‘t’, ‘hs is a difficult subject’]

Answer: d

Explanation: The code shown above demonstrates the use of the function re.match. The first argument of this function specifies the pattern. Since the pattern contains groups, those groups are incorporated in the resultant list as well. Hence the output of the code shown above is [‘M’, ‘a’, ‘t’, ‘hs is a difficult subject’].

2. The output of the following two Python codes are the same.

CODE 1

>>> re.split(r'(a)(t)', 'The night sky')

CODE 2

>>> re.split(r'\s+', 'The night sky')

a) True b) False

Answer: b

Explanation: The output of the first code is: [‘The night sky’] whereas the output of the second code is:[‘The’, ‘night’, ‘sky’]. Clearly, the outputs of the two codes are different. Hence the statement given above is a false one.

3. What will be the output of the following Python code?

import re

s = 'abc123 xyz666 lmn-11 def77'

re.sub(r'\b([a-z]+)(\d+)', r'\2\1:', s)

a) ‘123abc: 666xyz: lmn-11 77def:’

b) ‘77def: lmn-11: 666xyz: 123abc’

c) ‘abc123:’, ‘xyz666:’, ‘lmn-11:’, ‘def77:’

d) ‘abc123: xyz666: lmn-11: def77’

Answer: a => The function re.sub returns a string produced by replacing every non overlapping occurrence of the first argument with the second argument in the third argument. Hence the output is: ‘123abc: 666xyz: lmn-11 77def:’

4. What will be the output of the following Python code?

re.subn('A', 'X', 'AAAAAA', count=4)

a) ‘XXXXAA, 4’

b) (‘AAAAAA’, 4)

c) (‘XXXXAA’, 4)

d) ‘AAAAAA, 4’

Answer: c

Explanation: The line of code shown above demonstrates the function re.subn. This function is very similar to the function re.sub except that in the former, a tuple is returned instead of a string. The output of the code shown above is: (‘XXXXAA’, 4).

5. What will be the output of the following Python code?

n = re.sub(r'\w+', 'Hello', 'Cats and dogs')

a)

Hello

Hello

Hello

b) ‘Hello Hello Hello’

c) [‘Hello’, ‘Hello’, ‘Hello’]

d) (‘Hello’, ‘Hello’, ‘Hello’)

Answer: b

Explanation: The code shown above demonstrates the function re.sub. Since the string given as an argument consists of three words. The output of the code is: ‘Hello Hello Hello’. Had the string consisted of 4 words, the output would be: ‘Hello Hello Hello Hello’

6. What will be the output of the following Python code?

w = re.compile('[A-Za-z]+')

w.findall('It will rain today')

a) ‘It will rain today’

b) (‘It will rain today’)

c) [‘It will rain today’]

d) [‘It’, ‘will’, ‘rain’, ‘today’]

Answer: d => The code shown above demonstrates the function re.findall. Since all the words in the string match the criteria, the output of the code is: [‘It’, ‘will’, ‘rain’, ‘today’].

7. In the functions re.search.start(group) and re.search.end(group), if the argument groups not specified, it defaults to \_\_\_\_\_\_\_\_\_\_

a) Zero b) None c) One d) Error

Answer: a => In the functions re.search.start(group) and re.search.end(group), if the argument groups not specified, it defaults to Zero.

8. What will be the output of the following Python code?

re.split(r'\s+', 'Chrome is better than explorer', maxspilt=3)

a) [‘Chrome’, ‘is’, ‘better’, ‘than’, ‘explorer’] b) [‘Chrome’, ‘is’, ‘better’, ‘than explorer’]

c) (‘Chrome’, ‘is’, ‘better’, ‘than explorer’) d) ‘Chrome is better’ ‘than explorer’

Answer: b => The code shown above demonstrates the use of the function re.split, including the use of maxsplit. Since maxsplit is equal to 3, the output of the code shown above is:[‘Chrome’, ‘is’, ‘better’, ‘than explorer’]

9. What will be the output of the following Python code?

a=re.compile('[0-9]+')

a.findall('7 apples and 3 mangoes')

a) [‘apples’ ‘and’ ‘mangoes’]

b) (7, 4)

c) [‘7’, ‘4’]

d) Error

Answer: c

Explanation: The code shown above demonstrates the use of the functions re.compile and re.findall. Since we have specified in the code that only digits from 0-9 be found, hence the output of this code is: [‘7’, ‘4’].

1. Which of the following functions returns a dictionary mapping group names to group numbers?

a) re.compile.group

b) re.compile.groupindex

c) re.compile.index

d) re.compile.indexgroup

Answer: b

Explanation: The function re.compile.groupindex returns a dictionary mapping group names to group numbers.

2. Which of the following statements regarding the output of the function re.match is incorrect?

a) ‘pq\*’ will match ‘pq’

b) ‘pq?’ matches ‘p’

c) ‘p{4}, q’ does not match ‘pppq’

d) ‘pq+’ matches ‘p’

Answer: d

Explanation: All of the above statements are correct except that ‘pq+’ match ‘p’. ‘pq+’ will match ‘p’ followed by any non-zero number of q’s, but it will not match ‘p’.

3. The following Python code snippet results in an error.

c=re.compile(r'(\d+)(\[A-Z]+)([a-z]+)')

c.groupindex

a) True

b) False

Answer: b

Explanation: In the code shown above, none of the group names match the group numbers. In such a case, no error is thrown. The output of the code is an empty dictionary, that is, {}.

4. Which of the following functions does not accept any argument?

a) re.purge

b) re.compile

c) re.findall

d) re.match

Answer: a

Explanation: The function re.purge is used to clear the cache and it does not accept any arguments.

5. What will be the output of the following Python code?

a = re.compile('0-9')

a.findall('3 trees')

a) []

b) [‘3’]

c) Error

d) [‘trees’]

Answer: c

Explanation: The output of the code shown above is an empty list. This is due to the way the arguments have been passed to the function re.compile. Carefully read the code shown below in order to understand the correct syntax:

>>> a = re.compile(‘[0-9]’)

>>> a.findall(‘3 trees’)

[‘3’].

6. Which of the following lines of code will not show a match?

a) >>> re.match(‘ab\*’, ‘a’) b) >>> re.match(‘ab\*’, ‘ab’)

c) >>> re.match(‘ab\*’, ‘abb’) d) >>> re.match(‘ab\*’, ‘ba’)

Answer: d =>In the code shown above, ab\* will match to ‘a’ or ‘ab’ or ‘a’ followed by any number of b’s. Hence the only line of code from the above options which does not result in a match is:

>>> re.match(‘ab\*’, ‘ba’).

7. What will be the output of the following Python code?

m = re.search('a', 'The blue umbrella')

m.re.pattern

a) {} b) ‘The blue umbrella’ c) ‘a’ d) No output

Answer: c => The PatternObject is used to produce the match. The real regular expression pattern string must be retrieved from the PatternObject’s pattern method. Hence the output of this code is: ‘a’.

8. What will be the output of the following Python code?

re.sub('Y', 'X', 'AAAAAA', count=2)

a) ‘YXAAAA’ b) (‘YXAAAA’) c) (‘AAAAAA’) d) ‘AAAAAA’

Answer: d =>The code shown above demonstrates the function re.sub, which returns a string. The pattern specified is substituted in the string and returned. Hence the output of the code shown above is: ‘AAAAAA’.

**Files – 1**

<https://www.sanfoundry.com/python-mcqs-files-1/>

1. To open a file c:\scores.txt for reading, we use \_\_\_\_\_\_\_\_\_\_\_\_\_

a) infile = open(“c:\scores.txt”, “r”)

b) infile = open(“c:\\scores.txt”, “r”)

c) infile = open(file = “c:\scores.txt”, “r”)

d) infile = open(file = “c:\\scores.txt”, “r”)

Answer: b

Explanation: Execute help(open) to get more details.

2. To open a file c:\scores.txt for writing, we use \_\_\_\_\_\_\_\_\_\_\_\_

a) outfile = open(“c:\scores.txt”, “w”)

b) outfile = open(“c:\\scores.txt”, “w”)

c) outfile = open(file = “c:\scores.txt”, “w”)

d) outfile = open(file = “c:\\scores.txt”, “w”)

Answer: b

Explanation: w is used to indicate that file is to be written to.

3. To open a file c:\scores.txt for appending data, we use \_\_\_\_\_\_\_\_\_\_\_\_

a) outfile = open(“c:\\scores.txt”, “a”)

b) outfile = open(“c:\\scores.txt”, “rw”)

c) outfile = open(file = “c:\scores.txt”, “w”)

d) outfile = open(file = “c:\\scores.txt”, “w”)

Answer: a

Explanation: a is used to indicate that data is to be appended.

4. Which of the following statements are true?

a) When you open a file for reading, if the file does not exist, an error occurs

b) When you open a file for writing, if the file does not exist, a new file is created

c) When you open a file for writing, if the file exists, the existing file is overwritten with the new file

d) All of the mentioned

Answer: d

Explanation: The program will throw an error.

5. To read two characters from a file object infile, we use \_\_\_\_\_\_\_\_\_\_\_\_

a) infile.read(2)

b) infile.read()

c) infile.readline()

d) infile.readlines()

Answer: a

Explanation: Execute in the shell to verify.

6. To read the entire remaining contents of the file as a string from a file object infile, we use \_\_\_\_\_\_\_\_\_\_\_\_

a) infile.read(2)

b) infile.read()

c) infile.readline()

d) infile.readlines()

Answer: b

Explanation: read function is used to read all the lines in a file.

7. What will be the output of the following Python code?

f = None

for i in range (5):

with open("data.txt", "w") as f:

if i > 2:

break

print(f.closed)

a) True b) False c) None d) Error

Answer: a => The WITH statement when used with open file guarantees that the file object is closed when the with block exits.

8. To read the next line of the file from a file object infile, we use \_\_\_\_\_\_\_\_\_\_\_\_

a) infile.read(2)

b) infile.read()

c) infile.readline()

d) infile.readlines()

Answer: c

Explanation: Execute in the shell to verify.

9. To read the remaining lines of the file from a file object infile, we use \_\_\_\_\_\_\_\_\_\_\_\_

a) infile.read(2)

b) infile.read()

c) infile.readline()

d) infile.readlines()

Answer: d

Explanation: Execute in the shell to verify.

10. The readlines() method returns \_\_\_\_\_\_\_\_\_\_\_\_

a) str

b) a list of lines

c) a list of single characters

d) a list of integers

Answer: b

Explanation: Every line is stored in a list and returned.

1. Which are the two built-in functions to read a line of text from standard input, which by default comes from the keyboard?

a) Raw\_input & Input

b) Input & Scan

c) Scan & Scanner

d) Scanner

Answer: a

Explanation: Python provides two built-in functions to read a line of text from standard input, which by default comes from the keyboard. These functions are:

raw\_input and input

2. What will be the output of the following Python code?

str = raw\_input("Enter your input: ");

print "Received input is : ", str

a)

Enter your input: Hello Python

Received input is : Hello Python

b)

Enter your input: Hello Python

Received input is : Hello

c)

Enter your input: Hello Python

Received input is : Python

d) None of the mentioned

Answer: a

Explanation: The raw\_input([prompt]) function reads one line from standard input and returns it as a string. This would prompt you to enter any string and it would display same string on the screen. When I typed “Hello Python!”

3. What will be the output of the following Python code?

str = input("Enter your input: ");

print "Received input is : ", str

a)

Enter your input: [x\*5 for x in range(2,10,2)]

Received input is : [x\*5 for x in range(2,10,2)]

b)

Enter your input: [x\*5 for x in range(2,10,2)]

Received input is : [10, 30, 20, 40]

c)

Enter your input: [x\*5 for x in range(2,10,2)]

Received input is : [10, 10, 30, 40]

d) None of the mentioned

Answer: a

Explanation: None.

4. Which one of the following is not attributes of file?

a) closed b) softspace c) rename d) mode

Answer: c

Explanation: rename is not the attribute of file rest all are files attributes.

Attribute Description

file.closed Returns true if file is closed, false otherwise.

file.mode Returns access mode with which file was opened.

file.name Returns name of the file.

file.softspace Returns false if space explicitly required with print, true otherwise.

5. What is the use of tell() method in python?

a) tells you the current position within the file b) tells you the end position within the file

c) tells you the file is opened or not d) none of the mentioned

Answer: a

Explanation: The tell() method tells you the current position within the file; in other words, the next read or write will occur at that many bytes from the beginning of the file.

6. What is the current syntax of rename() a file?

a) rename(current\_file\_name, new\_file\_name) b) rename(new\_file\_name, current\_file\_name,)

c) rename(()(current\_file\_name, new\_file\_name)) d) none of the mentioned

Answer: a

Explanation: This is the correct syntax which has shown below.

rename(current\_file\_name, new\_file\_name)

7. What is the current syntax of remove() a file?

a) remove(file\_name) b) remove(new\_file\_name, current\_file\_name,)

c) remove(() , file\_name)) d) none of the mentioned

Answer: a

Explanation: remove(file\_name)

8. What will be the output of the following Python code?

fo = open("foo.txt", "rw+")

print "Name of the file: ", fo.name

# Assuming file has following 5 lines

# This is 1st line

# This is 2nd line

# This is 3rd line

# This is 4th line

# This is 5th line

for index in range(5):

line = fo.next()

print "Line No %d - %s" % (index, line)

# Close opened file

fo.close()

a) Compilation Error b) Syntax Error c) Displays Output d) None of the mentioned

Answer: c

Explanation: It displays the output as shown below. The method next() is used when a file is used as an iterator, typically in a loop, the next() method is called repeatedly. This method returns the next input line, or raises StopIteration when EOF is hit.

Output:

Name of the file: foo.txt

Line No 0 - This is 1st line

Line No 1 - This is 2nd line

Line No 2 - This is 3rd line

Line No 3 - This is 4th line

Line No 4 - This is 5th line

9. What is the use of seek() method in files?

a) sets the file’s current position at the offset b) sets the file’s previous position at the offset

c) sets the file’s current position within the file d) none of the mentioned

Answer: a

Explanation: Sets the file’s current position at the offset. The method seek() sets the file’s current position at the offset.

Following is the syntax for seek() method:

fileObject.seek(offset[, whence])

Parameters

offset — This is the position of the read/write pointer within the file.

whence — This is optional and defaults to 0 which means absolute file positioning, other values are 1 which means seek relative to the current position and 2 means seek relative to the file’s end.

10. What is the use of truncate() method in file?

a) truncates the file size b) deletes the content of the file

c) deletes the file size d) none of the mentioned

Answer: a

Explanation: The method truncate() truncates the file size. Following is the syntax for truncate() method:

fileObject.truncate( [ size ])

Parameters

size — If this optional argument is present, the file is truncated to (at most) that size.

1. Which is/are the basic I/O connections in file?

a) Standard Input b) Standard Output

c) Standard Errors d) All of the mentioned

Answer: d

Explanation: Standard input, standard output and standard error. Standard input is the data that goes to the program. The standard input comes from a keyboard. Standard output is where we print our data with the print keyword. Unless redirected, it is the terminal console. The standard error is a stream where programs write their error messages. It is usually the text terminal.

2. What will be the output of the following Python code? (If entered name is sanfoundry)

import sys

print 'Enter your name: ',

name = ''

while True:

c = sys.stdin.read(1)

if c == '\n':

break

name = name + c

print 'Your name is:', name

a) sanfoundry b) sanfoundry, sanfoundry

c) San d) None of the mentioned

Answer: a

Explanation: In order to work with standard I/O streams, we must import the sys module. The read() method reads one character from the standard input. In our example we get a prompt saying “Enter your name”. We enter our name and press enter. The enter key generates the new line character: \n.

Output:

Enter your name: sanfoundry

Your name is: sanfoundry

3. What will be the output of the following Python code?

import sys

sys.stdout.write(' Hello\n')

sys.stdout.write('Python\n')

a) Compilation Error

b) Runtime Error

c) Hello Python

d)

Hello

Python

Answer: d

Explanation: None

Output:

Hello

Python

4. Which of the following mode will refer to binary data?

a) r

b) w

c) +

d) b

Answer:d

Explanation: Mode Meaning is as explained below:

r Reading

w Writing

a Appending

b Binary data

+ Updating.

5. What is the pickling?

a) It is used for object serialization b) It is used for object deserialization

c) None of the mentioned d) All of the mentioned

Answer: a

Explanation: Pickle is the standard mechanism for object serialization. Pickle uses a simple stack-based virtual machine that records the instructions used to reconstruct the object. This makes pickle vulnerable to security risks by malformed or maliciously constructed data, that may cause the deserializer to import arbitrary modules and instantiate any object.

6. What is unpickling?

a) It is used for object serialization b) It is used for object deserialization

c) None of the mentioned d) All of the mentioned

Answer: b

Explanation: We have been working with simple textual data. What if we are working with objects rather than simple text? For such situations, we can use the pickle module. This module serializes Python objects. The Python objects are converted into byte streams and written to text files. This process is called pickling. The inverse operation, reading from a file and reconstructing objects is called deserializing or unpickling.

7. What is the correct syntax of open() function?

a) file = open(file\_name [, access\_mode][, buffering])

b) file object = open(file\_name [, access\_mode][, buffering])

c) file object = open(file\_name)

d) none of the mentioned

Answer: b

Explanation: Open() function correct syntax with the parameter details as shown below:

file object = open(file\_name [, access\_mode][, buffering])

Here is parameters’ detail:

file\_name: The file\_name argument is a string value that contains the name of the file that you want to access.

access\_mode: The access\_mode determines the mode in which the file has to be opened, i.e., read, write, append, etc. A complete list of possible values is given below in the table. This is optional parameter and the default file access mode is read (r).

buffering: If the buffering value is set to 0, no buffering will take place. If the buffering value is 1, line buffering will be performed while accessing a file. If you specify the buffering value as an integer greater than 1, then buffering action will be performed with the indicated buffer size. If negative, the buffer size is the system default(default behavior).

8. What will be the output of the following Python code?

fo = open("foo.txt", "wb")

print "Name of the file: ", fo.name

fo.flush()

fo.close()

a) Compilation Error b) Runtime Error

c) No Output d) Flushes the file when closing them

Answer: d =>The method flush() flushes the internal buffer. Python automatically flushes the files when closing them. But you may want to flush the data before closing any file.

9. Correct syntax of file.writelines() is?

a) file.writelines(sequence) b) fileObject.writelines()

c) fileObject.writelines(sequence) d) none of the mentioned

Answer: c

Explanation: The method writelines() writes a sequence of strings to the file. The sequence can be any iterable object producing strings, typically a list of strings. There is no return value.

Syntax

Following is the syntax for writelines() method:

fileObject.writelines( sequence ).

10. Correct syntax of file.readlines() is?

a) fileObject.readlines( sizehint ); b) fileObject.readlines();

c) fileObject.readlines(sequence) d) none of the mentioned

Answer: a =>The method readlines() reads until EOF using readline() and returns a list containing the lines. If the optional sizehint argument is present, instead of reading up to EOF, whole lines totalling approximately sizehint bytes (possibly after rounding up to an internal buffer size) are read.

Syntax

Following is the syntax for readlines() method:

fileObject.readlines( sizehint );

Parameters

sizehint — This is the number of bytes to be read from the file.

1. In file handling, what does this terms means “r, a”?

a) read, append b) append, read

c) write, append d) none of the mentioned

Answer: a => r- reading, a-appending.

2. What is the use of “w” in file handling?

a) Read b) Write c) Append d) None of the mentioned

Answer: b =>This opens the file for writing. It will create the file if it doesn’t exist, and if it does, it will overwrite it.

fh = open(“filename\_here”, “w”).

3. What is the use of “a” in file handling?

a) Read b) Write c) Append d) None of the mentioned

Answer: c

Explanation: This opens the fhe file in appending mode. That means, it will be open for writing and everything will be written to the end of the file.

fh =open(“filename\_here”, “a”).

4. Which function is used to read all the characters?

a) Read() b) Readcharacters()

c) Readall() d) Readchar()

Answer: a

Explanation: The read function reads all characters fh = open(“filename”, “r”)

content = fh.read().

5. Which function is used to read single line from file?

a) Readline() b) Readlines()

c) Readstatement() d) Readfullline()

Answer: b

Explanation: The readline function reads a single line from the file fh = open(“filename”, “r”)

content = fh.readline().

6. Which function is used to write all the characters?

a) write() b) writecharacters()

c) writeall() d) writechar()

Answer: a

Explanation: To write a fixed sequence of characters to a file

fh = open(“hello.txt”,”w”)

write(“Hello World”).

7. Which function is used to write a list of string in a file?

a) writeline() b) writelines()

c) writestatement() d) writefullline()

Answer: a

Explanation: With the writeline function you can write a list of strings to a file

fh = open(“hello.txt”, “w”)

lines\_of\_text = [“a line of text”, “another line of text”, “a third line”] fh.writelines(lines\_of\_text).

8. Which function is used to close a file in python?

a) Close() b) Stop()

c) End() d) Closefile()

Answer: a

Explanation: f.close()to close it and free up any system resources taken up by the open file.

9. Is it possible to create a text file in python?

a) Yes b) No

c) Machine dependent d) All of the mentioned

Answer: a

Explanation: Yes we can create a file in python. Creation of file is as shown below.

file = open(“newfile.txt”, “w”)

file.write(“hello world in the new file\n”)

file.write(“and another line\n”)

file.close().

10. Which of the following are the modes of both writing and reading in binary format in file?

a) wb+

b) w

c) wb

d) w+

Answer: a

Explanation: Here is the description below

“w” Opens a file for writing only. Overwrites the file if the file exists. If the file does not exist, creates a new file for writing.

“wb” Opens a file for writing only in binary format. Overwrites the file if the file exists. If the file does not exist, creates a new file for writing.

“w+” Opens a file for both writing and reading. Overwrites the existing file if the file exists. If the file does not exist, creates a new file for reading and writing.

“wb+” Opens a file for both writing and reading in binary format. Overwrites the existing file if the file exists. If the file does not exist, creates a new file for reading and writing.

1. Which of the following is not a valid mode to open a file?

a) ab

b) rw

c) r+

d) w+

Answer: b

Explanation: Use r+, w+ or a+ to perform both read and write operations using a single file object.

2. What is the difference between r+ and w+ modes?

a) no difference

b) in r+ the pointer is initially placed at the beginning of the file and the pointer is at the end for w+

c) in w+ the pointer is initially placed at the beginning of the file and the pointer is at the end for r+

d) depends on the operating system

Answer: b

Explanation: none.

3. How do you get the name of a file from a file object (fp)?

a) fp.name

b) fp.file(name)

c) self.\_\_name\_\_(fp)

d) fp.\_\_name\_\_()

Answer: a

Explanation: name is an attribute of the file object.

4. Which of the following is not a valid attribute of a file object (fp)?

a) fp.name

b) fp.closed

c) fp.mode

d) fp.size

Answer: d

Explanation: fp.size has not been implemented.

5. How do you close a file object (fp)?

a) close(fp)

b) fclose(fp)

c) fp.close()

d) fp.\_\_close\_\_()

Answer: c

Explanation: close() is a method of the file object.

6. How do you get the current position within the file?

a) fp.seek()

b) fp.tell()

c) fp.loc

d) fp.pos

Answer: b

Explanation: It gives the current position as an offset from the start of file.

7. How do you rename a file?

a) fp.name = ‘new\_name.txt’

b) os.rename(existing\_name, new\_name)

c) os.rename(fp, new\_name)

d) os.set\_name(existing\_name, new\_name)

Answer: b

Explanation: os.rename() is used to rename files.

8. How do you delete a file?

a) del(fp)

b) fp.delete()

c) os.remove(‘file’)

d) os.delete(‘file’)

Answer: c

Explanation: os.remove() is used to delete files.

9. How do you change the file position to an offset value from the start?

a) fp.seek(offset, 0)

b) fp.seek(offset, 1)

c) fp.seek(offset, 2)

d) none of the mentioned

Answer: a

Explanation: 0 indicates that the offset is with respect to the start.

10. What happens if no arguments are passed to the seek function?

a) file position is set to the start of file

b) file position is set to the end of file

c) file position remains unchanged

d) error

Answer: d

Explanation: seek() takes at least one argument.

**Operator Overloading**

1. Which function is called when the following Python code is executed?

f = foo()

format(f)

a) format()

b) \_\_format\_\_()

c) str()

d) \_\_str\_\_()

Answer: d

Explanation: Both str(f) and format(f) call f.\_\_str\_\_().

2. Which of the following Python code will print True?

a = foo(2)

b = foo(3)

print(a < b)

a)

class foo:

def \_\_init\_\_(self, x):

self.x = x

def \_\_lt\_\_(self, other):

if self.x < other.x:

return False

else:

return True

b)

class foo:

def \_\_init\_\_(self, x):

self.x = x

def \_\_less\_\_(self, other):

if self.x > other.x:

return False

else:

return True

c)

class foo:

def \_\_init\_\_(self, x):

self.x = x

def \_\_lt\_\_(self, other):

if self.x < other.x:

return True

else:

return False

d)

class foo:

def \_\_init\_\_(self, x):

self.x = x

def \_\_less\_\_(self, other):

if self.x < other.x:

return False

else:

return True

Answer: c

Explanation: \_\_lt\_\_ overloads the < operator>.

3. Which function overloads the + operator?

a) \_\_add\_\_()

b) \_\_plus\_\_()

c) \_\_sum\_\_()

d) none of the mentioned

Answer: a

Explanation: Refer documentation.

4. Which operator is overloaded by \_\_invert\_\_()?

a) !

b) ~

c) ^

d) –

Answer: b

Explanation: \_\_invert\_\_() overloads ~.

5. Which function overloads the == operator?

a) \_\_eq\_\_()

b) \_\_equ\_\_()

c) \_\_isequal\_\_()

d) none of the mentioned

Answer: a

Explanation: The other two do not exist.

6. Which operator is overloaded by \_\_lg\_\_()?

a) <

b) >

c) !=

d) none of the mentioned

Answer: d

Explanation: \_\_lg\_\_() is invalid.

7. Which function overloads the >> operator?

a) \_\_more\_\_()

b) \_\_gt\_\_()

c) \_\_ge\_\_()

d) none of the mentioned

Answer: d

Explanation: \_\_rshift\_\_() overloads the >> operator.

8. Let A and B be objects of class Foo. Which functions are called when print(A + B) is executed?

a) \_\_add\_\_(), \_\_str\_\_()

b) \_\_str\_\_(), \_\_add\_\_()

c) \_\_sum\_\_(), \_\_str\_\_()

d) \_\_str\_\_(), \_\_sum\_\_()

Answer: a => The function \_\_add\_\_() is called first since it is within the bracket. The function \_\_str\_\_() is then called on the object that we received after adding A and B.

9. Which operator is overloaded by the \_\_or\_\_() function?

a) ||

b) |

c) //

d) /

Answer: b => The function \_\_or\_\_() overloads the bitwise OR operator |.

10. Which function overloads the // operator?

a) \_\_div\_\_()

b) \_\_ceildiv\_\_()

c) \_\_floordiv\_\_()

d) \_\_truediv\_\_()

Answer: c => \_\_floordiv\_\_() is for //.

**Classes and Objects – 1**

<https://www.sanfoundry.com/python-questions-answers-classes-objects-1/>

1. \_\_\_\_\_ represents an entity in the real world with its identity and behaviour.

a) A method b) An object

c) A class d) An operator

Answer: b

Explanation: An object represents an entity in the real world that can be distinctly identified. A class may define an object.

2. \_\_\_\_\_ is used to create an object.

a) class b) constructor

c) User-defined functions d) In-built functions

Answer: b

Explanation: The values assigned by the constructor to the class members is used to create the object.

3. What will be the output of the following Python code?

class test:

def \_\_init\_\_(self,a="Hello World"):

self.a=a

def display(self):

print(self.a)

obj=test()

obj.display()

a) The program has an error because constructor can’t have default arguments

b) Nothing is displayed

c) “Hello World” is displayed

d) The program has an error display function doesn’t have parameters

Answer: c

Explanation: The program has no error. “Hello World” is displayed. Execute in python shell to verify.

4. What is setattr() used for?

a) To access the attribute of the object b) To set an attribute

c) To check if an attribute exists or not d) To delete an attribute

Answer: b

Explanation: setattr(obj,name,value) is used to set an attribute. If attribute doesn’t exist, then it would be created.

5. What is getattr() used for?

a) To access the attribute of the object b) To delete an attribute

c) To check if an attribute exists or not d) To set an attribute

Answer: a

Explanation: getattr(obj,name) is used to get the attribute of an object.

6. What will be the output of the following Python code?

class change:

def \_\_init\_\_(self, x, y, z):

self.a = x + y + z

x = change(1,2,3)

y = getattr(x, 'a')

setattr(x, 'a', y+1)

print(x.a)

a) 6

b) 7

c) Error

d) 0

Answer: b

Explanation: First, a=1+2+3=6. Then, after setattr() is invoked, x.a=6+1=7.

7. What will be the output of the following Python code?

class test:

def \_\_init\_\_(self,a):

self.a=a

def display(self):

print(self.a)

obj=test()

obj.display()

a) Runs normally, doesn’t display anything

b) Displays 0, which is the automatic default value

c) Error as one argument is required while creating the object

d) Error as display function requires additional argument

Answer: c

Explanation: Since, the \_\_init\_\_ special method has another argument a other than self, during object creation, one argument is required. For example: obj=test(“Hello”)

8. Is the following Python code correct?

>>> class A:

def \_\_init\_\_(self,b):

self.b=b

def display(self):

print(self.b)

>>> obj=A("Hello")

>>> del obj

a) True

b) False

Answer: a

Explanation: It is possible to delete an object of the class. On further typing obj in the python shell, it throws an error because the defined object has now been deleted.

9. What will be the output of the following Python code?

class test:

def \_\_init\_\_(self):

self.variable = 'Old'

self.Change(self.variable)

def Change(self, var):

var = 'New'

obj=test()

print(obj.variable)

a) Error because function change can’t be called in the \_\_init\_\_ function

b) ‘New’ is printed

c) ‘Old’ is printed

d) Nothing is printed

Answer: c

Explanation: This is because strings are immutable. Hence any change made isn’t reflected in the original string.

10. What is Instantiation in terms of OOP terminology?

a) Deleting an instance of class b) Modifying an instance of class

c) Copying an instance of class d) Creating an instance of class

Answer: d

Explanation: Instantiation refers to creating an object/instance for a class.

11. What will be the output of the following Python code?

class fruits:

def \_\_init\_\_(self, price):

self.price = price

obj=fruits(50)

obj.quantity=10

obj.bags=2

print(obj.quantity+len(obj.\_\_dict\_\_))

a) 12

b) 52

c) 13

d) 60

Answer: c => In the above code, obj.quantity has been initialised to 10. There are a total of three items in the dictionary, price, quantity and bags. Hence, len(obj.\_\_dict\_\_) is 3.

12. What will be the output of the following Python code?

class Demo:

def \_\_init\_\_(self):

pass

def test(self):

print(\_\_name\_\_)

obj = Demo()

obj.test()

a) Exception is thrown

b) \_\_main\_\_

c) Demo

d) test

Answer: b => Since the above code is being run not as a result of an import from another module, the variable will have value “\_\_main\_\_”.

1. The assignment of more than one function to a particular operator is \_\_\_\_\_\_\_

a) Operator over-assignment b) Operator overriding

c) Operator overloading d) Operator instance

Answer: c => The assignment of more than one function to a particular operator is called as operator overloading.

2. Which of the following is not a class method?

a) Non-static b) Static

c) Bounded d) Unbounded

Answer: a => The three different class methods in Python are static, bounded and unbounded methods.

3. What will be the output of the following Python code?

def add(c,k):

c.test=c.test+1

k=k+1

class A:

def \_\_init\_\_(self):

self.test = 0

def main():

Count=A()

k=0

for i in range(0,25):

add(Count,k)

print("Count.test=", Count.test)

print("k =", k)

main()

a) Exception is thrown

b)

Count.test=25

k=25

c)

Count.test=25

k=0

d)

Count.test=0

k=0

Answer: c

Explanation: The program has no error. Here, test is a member of the class while k isn’t. Hence test keeps getting incremented 25 time while k remains 0.

4. Which of the following Python code creates an empty class?

a)

class A:

return

b)

class A:

pass

c)

class A:

d) It is not possible to create an empty class

Answer: b => Execute in python shell to verify.

5. Is the following Python code valid?

class B(object):

def first(self):

print("First method called")

def second():

print("Second method called")

ob = B()

B.first(ob)

a) It isn’t as the object declaration isn’t right

b) It isn’t as there isn’t any \_\_init\_\_ method for initializing class members

c) Yes, this method of calling is called unbounded method call

d) Yes, this method of calling is called bounded method call

Answer: c

Explanation: The method may be created in the method demonstrated in the code as well and this is called as the unbounded method call. Calling the method using obj.one() is the bounded method call.

6. What are the methods which begin and end with two underscore characters called?

a) Special methods b) In-built methods

c) User-defined methods d) Additional methods

Answer: a => Special methods like \_\_init\_\_ begin and end with two underscore characters.

7. Special methods need to be explicitly called during object creation.

a) True b) False

Answer: b => Special methods are automatically called during object creation.

8. What will be the output of the following Python code?

>>> class demo():

def \_\_repr\_\_(self):

return '\_\_repr\_\_ built-in function called'

def \_\_str\_\_(self):

return '\_\_str\_\_ built-in function called'

>>> s=demo()

>>> s

a) Error b) Nothing is printed

c) \_\_str\_\_ called d) \_\_repr\_\_ called

Answer: c

Explanation: \_\_str\_\_ is used for producing a string representation of an object’s value that Python can evaluate. Execute in python shell to verify.

9. What will be the output of the following Python code?

>>> class demo():

def \_\_repr\_\_(self):

return '\_\_repr\_\_ built-in function called'

def \_\_str\_\_(self):

return '\_\_str\_\_ built-in function called'

>>> s=demo()

>>> print(s)

a) \_\_str\_\_ called b) \_\_repr\_\_ called

c) Error d) Nothing is printed

Answer: a

Explanation: \_\_str\_\_ is used for producing a string representation of an object’s value that is most readable for humans. Execute in python shell to verify.

10. What is hasattr(obj,name) used for?

a) To access the attribute of the object b) To delete an attribute

c) To check if an attribute exists or not d) To set an attribute

Answer: c => hasattr(obj,name) checks if an attribute exists or not and returns True or False.

11. What will be the output of the following Python code?

class stud:

def \_\_init\_\_(self, roll\_no, grade):

self.roll\_no = roll\_no

self.grade = grade

def display (self):

print("Roll no : ", self.roll\_no, ", Grade: ", self.grade)

stud1 = stud(34, ‘S’)

stud1.age=7

print(hasattr(stud1, 'age'))

a) Error as age isn’t defined b) True

c) False d) 7

Answer: a => Execute in python shell to verify.

12. What is delattr(obj,name) used for?

a) To print deleted attribute b) To delete an attribute

c) To check if an attribute is deleted or not d) To set an attribute

Answer: b => delattr(obj,name) deletes an attribute in a class.

13. \_\_del\_\_ method is used to destroy instances of a class.

a) True b) False

Answer: a => \_\_\_del\_\_ method acts as a destructor and is used to destroy objects of classes.

14. What will be the output of the following Python code?

class stud:

‘Base class for all students’

def \_\_init\_\_(self, roll\_no, grade):

self.roll\_no = roll\_no

self.grade = grade

def display (self):

print("Roll no : ", self.roll\_no, ", Grade: ", self.grade)

print(student.\_\_doc\_\_)

a) Exception is thrown b) \_\_main\_\_

c) Nothing is displayed d) Base class for all students

Answer: d =>\_\_\_doc\_\_ built-in class attribute is used to print the class documentation string or none, if undefined.

15. What does print(Test.\_\_name\_\_) display (assuming Test is the name of the class)?

a) () b) Exception is thrown

c) Test d) \_\_main\_\_

Answer: c => \_\_name\_\_ built-in class attribute is used to display the class name.

**Inheritance – 1**

<https://www.sanfoundry.com/python-questions-answers-inheritance-1/>

1. Which of the following best describes inheritance?

a) Ability of a class to derive members of another class as a part of its own definition

b) Means of bundling instance variables and methods in order to restrict access to certain class members

c) Focuses on variables and passing of variables to functions

d) Allows for implementation of elegant software that is well designed and easily modified

Answer: a => If the class definition is class B(A): then class B inherits the methods of class A. This is called inheritance.

2. Which of the following statements is wrong about inheritance?

a) Protected members of a class can be inherited

b) The inheriting class is called a subclass

c) Private members of a class can be inherited and accessed

d) Inheritance is one of the features of OOP

Answer: c => Any changes made to the private members of the class in the subclass aren’t reflected in the original members.

3. What will be the output of the following Python code?

class Demo:

def \_\_new\_\_(self):

self.\_\_init\_\_(self)

print("Demo's \_\_new\_\_() invoked")

def \_\_init\_\_(self):

print("Demo's \_\_init\_\_() invoked")

class Derived\_Demo(Demo):

def \_\_new\_\_(self):

print("Derived\_Demo's \_\_new\_\_() invoked")

def \_\_init\_\_(self):

print("Derived\_Demo's \_\_init\_\_() invoked")

def main():

obj1 = Derived\_Demo()

obj2 = Demo()

main()

a)

Derived\_Demo’s \_\_init\_\_() invoked

Derived\_Demo's \_\_new\_\_() invoked

Demo's \_\_init\_\_() invoked

Demo's \_\_new\_\_() invoked

b)

Derived\_Demo's \_\_new\_\_() invoked

Demo's \_\_init\_\_() invoked

Demo's \_\_new\_\_() invoked

c)

Derived\_Demo's \_\_new\_\_() invoked

Demo's \_\_new\_\_() invoked

d)

Derived\_Demo’s \_\_init\_\_() invoked

Demo's \_\_init\_\_() invoked

Answer: b

Explanation: Since the object for the derived class is declared first, \_\_new\_\_() method of the derived class is invoked first, followed by the constructor and the \_\_new\_\_() method of main class.

4. What will be the output of the following Python code?

class Test:

def \_\_init\_\_(self):

self.x = 0

class Derived\_Test(Test):

def \_\_init\_\_(self):

self.y = 1

def main():

b = Derived\_Test()

print(b.x,b.y)

main()

a) 0 1

b) 0 0

c) Error because class B inherits A but variable x isn’t inherited

d) Error because when object is created, argument must be passed like Derived\_Test(1)

Answer: c

Explanation: Since the invoking method, Test.\_\_init\_\_(self), isn’t present in the derived class, variable x can’t be inherited.

5. What will be the output of the following Python code?

class A():

def disp(self):

print("A disp()")

class B(A):

pass

obj = B()

obj.disp()

a) Invalid syntax for inheritance

b) Error because when object is created, argument must be passed

c) Nothing is printed

d) A disp()

Answer: d

Explanation: Class B inherits class A hence the function disp () becomes part of class B’s definition. Hence disp() method is properly executed and the line is printed.

6. All subclasses are a subtype in object-oriented programming.

a) True b) False

Answer: b => A subtype is something that be substituted for and behave as its parent type. All subclass may not be a subtype in object-oriented programming.

7. When defining a subclass in Python that is meant to serve as a subtype, the subtype Python keyword is used.

a) True b) False

Answer: b => B is a subtype of B if instances of type B can substitute for instances of type A without affecting semantics.

8. Suppose B is a subclass of A, to invoke the \_\_init\_\_ method in A from B, what is the line of code you should write?

a) A.\_\_init\_\_(self)

b) B.\_\_init\_\_(self)

c) A.\_\_init\_\_(B)

d) B.\_\_init\_\_(A)

Answer: a

Explanation: To invoke the \_\_init\_\_ method in A from B, either of the following should be written: A.\_\_init\_\_(self) or super().\_\_init\_\_(self).

9. What will be the output of the following Python code?

class Test:

def \_\_init\_\_(self):

self.x = 0

class Derived\_Test(Test):

def \_\_init\_\_(self):

Test.\_\_init\_\_(self)

self.y = 1

def main():

b = Derived\_Test()

print(b.x,b.y)

main()

a) Error because class B inherits A but variable x isn’t inherited

b) 0 0

c) 0 1

d) Error, the syntax of the invoking method is wrong

Answer: c

Explanation: Since the invoking method has been properly invoked, variable x from the main class has been properly inherited and it can also be accessed.

10. What will be the output of the following Python code?

class A:

def \_\_init\_\_(self, x= 1):

self.x = x

class der(A):

def \_\_init\_\_(self,y = 2):

super().\_\_init\_\_()

self.y = y

def main():

obj = der()

print(obj.x, obj.y)

main()

a) Error, the syntax of the invoking method is wrong

b) The program runs fine but nothing is printed

c) 1 0

d) 1 2

Answer: d

Explanation: In the above piece of code, the invoking method has been properly implemented and hence x=1 and y=2.

11. What does built-in function type do in context of classes?

a) Determines the object name of any value

b) Determines the class name of any value

c) Determines class description of any value

d) Determines the file name of any value

Answer: b => For example: >>> type((1,)) gives <class ‘tuple’>.

12. Which of the following is not a type of inheritance?

a) Double-level b) Multi-level c) Single-level d) Multiple

Answer: a => Multiple, multi-level, single-level and hierarchical inheritance are all types of inheritance.

13. What does built-in function help do in context of classes?

a) Determines the object name of any value

b) Determines the class identifiers of any value

c) Determines class description of any built-in type

d) Determines class description of any user-defined built-in type

Answer: c =>help() usually gives information of the class on any built-in type or function.

14. What will be the output of the following Python code?

class A:

def one(self):

return self.two()

def two(self):

return 'A'

class B(A):

def two(self):

return 'B'

obj1=A()

obj2=B()

print(obj1.two(),obj2.two())

a) A A

b) A B

c) B B

d) An exception is thrown

Answer: b

Explanation: obj1.two() invokes the method two() in class A which returns ‘A’ and obj2.two() invokes the method two() in class B which returns ‘B’.

15. What type of inheritance is illustrated in the following Python code?

class A():

pass

class B():

pass

class C(A,B):

pass

a) Multi-level inheritance b) Multiple inheritance

c) Hierarchical inheritance d) Single-level inheritance

Answer: b

Explanation: In multiple inheritance, two or more subclasses are derived from the superclass as shown in the above piece of code.

1. What type of inheritance is illustrated in the following Python code?

class A():

pass

class B(A):

pass

class C(B):

pass

a) Multi-level inheritance b) Multiple inheritance

c) Hierarchical inheritance d) Single-level inheritance

Answer: a => In multi-level inheritance, a subclass derives from another class which itself is derived from another class.

2. What does single-level inheritance mean?

a) A subclass derives from a class which in turn derives from another class

b) A single superclass inherits from multiple subclasses

c) A single subclass derives from a single superclass

d) Multiple base classes inherit a single derived class

Answer: c

Explanation: In single-level inheritance, there is a single subclass which inherits from a single superclass. So the class definition of the subclass will be: class B(A): where A is the superclass.

3. What will be the output of the following Python code?

class A:

def \_\_init\_\_(self):

self.\_\_i = 1

self.j = 5

def display(self):

print(self.\_\_i, self.j)

class B(A):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.\_\_i = 2

self.j = 7

c = B()

c.display()

a) 2 7

b) 1 5

c) 1 7

d) 2 5

Answer: c => Any change made in variable i isn’t reflected as it is the private member of the superclass.

4. Which of the following statements isn’t true?

a) A non-private method in a superclass can be overridden

b) A derived class is a subset of superclass

c) The value of a private variable in the superclass can be changed in the subclass

d) When invoking the constructor from a subclass, the constructor of superclass is automatically invoked

Answer: c

Explanation: If the value of a private variable in a superclass is changed in the subclass, the change isn’t reflected.

5. What will be the output of the following Python code?

class A:

def \_\_init\_\_(self,x):

self.x = x

def count(self,x):

self.x = self.x+1

class B(A):

def \_\_init\_\_(self, y=0):

A.\_\_init\_\_(self, 3)

self.y = y

def count(self):

self.y += 1

def main():

obj = B()

obj.count()

print(obj.x, obj.y)

main()

a) 3 0

b) 3 1

c) 0 1

d) An exception in thrown

Answer: b => Initially x=3 and y=0. When obj.count() is called, y=1.

6. What will be the output of the following Python code?

>>> class A:

pass

>>> class B(A):

pass

>>> obj=B()

>>> isinstance(obj,A)

a) True b) False

c) Wrong syntax for isinstance() method d) Invalid method for classes

Answer: a => isinstance(obj,class) returns True if obj is an object class.

7. Which of the following statements is true?

a) The \_\_new\_\_() method automatically invokes the \_\_init\_\_ method

b) The \_\_init\_\_ method is defined in the object class

c) The \_\_eq(other) method is defined in the object class

d) The \_\_repr\_\_() method is defined in the object class

Answer: c => The \_\_eq(other) method is called if any comparison takes place and it is defined in the object class.

8. Method issubclass() checks if a class is a subclass of another class.

a) True b) False

Answer: a => Method issubclass() returns True if a class is a subclass of another class and False otherwise.

9. What will be the output of the following Python code?

class A:

def \_\_init\_\_(self):

self.\_\_x = 1

class B(A):

def display(self):

print(self.\_\_x)

def main():

obj = B()

obj.display()

main()

a) 1

b) 0

c) Error, invalid syntax for object declaration

d) Error, private class member can’t be accessed in a subclass

Answer: d => Private class members in the superclass can’t be accessed in the subclass.

10. What will be the output of the following Python code?

class A:

def \_\_init\_\_(self):

self.\_x = 5

class B(A):

def display(self):

print(self.\_x)

def main():

obj = B()

obj.display()

main()

a) Error, invalid syntax for object declaration

b) Nothing is printed

c) 5

d) Error, private class member can’t be accessed in a subclass

Answer: c

Explanation: The class member x is protected, not private and hence can be accessed by subclasses.

11. What will be the output of the following Python code?

class A:

def \_\_init\_\_(self,x=3):

self.\_x = x

class B(A):

def \_\_init\_\_(self):

super().\_\_init\_\_(5)

def display(self):

print(self.\_x)

def main():

obj = B()

obj.display()

main()

a) 5

b) Error, class member x has two values

c) 3

d) Error, protected class member can’t be accessed in a subclass

Answer: a

Explanation: The super() method re-assigns the variable x with value 5. Hence 5 is printed.

12. What will be the output of the following Python code?

class A:

def test1(self):

print(" test of A called ")

class B(A):

def test(self):

print(" test of B called ")

class C(A):

def test(self):

print(" test of C called ")

class D(B,C):

def test2(self):

print(" test of D called ")

obj=D()

obj.test()

a)

test of B called

test of C called

b)

test of C called

test of B called

c) test of B called

d) Error, both the classes from which D derives has same method test()

Answer: c => Execute in Python shell to verify. If class D(B,C): is switched is class D(C,B): test of C is called.

13. What will be the output of the following Python code?

class A:

def test(self):

print("test of A called")

class B(A):

def test(self):

print("test of B called")

super().test()

class C(A):

def test(self):

print("test of C called")

super().test()

class D(B,C):

def test2(self):

print("test of D called")

obj=D()

obj.test()

a) test of B called

test of C called

test of A called

b) test of C called

test of B called

c) test of B called

test of C called

d) Error, all the three classes from which D derives has same method test()

Answer: a => Since the invoking method, super().test() is called in the subclasses, all the three methods of test() in three different classes is called.

**Polymorphism**

<https://www.sanfoundry.com/python-questions-answers-polymorphism/>

1. Which of the following best describes polymorphism?

a) Ability of a class to derive members of another class as a part of its own definition

b) Means of bundling instance variables and methods in order to restrict access to certain class members

c) Focuses on variables and passing of variables to functions

d) Allows for objects of different types and behaviour to be treated as the same general type

Answer: d

Explanation: Polymorphism is a feature of object-oriented programming languages. It allows for the implementation of elegant software that is well designed and easily modified.

2. What is the biggest reason for the use of polymorphism?

a) It allows the programmer to think at a more abstract level

b) There is less program code to write

c) The program will have a more elegant design and will be easier to maintain and update

d) Program code takes up less space

Answer: c

Explanation: Polymorphism allows for the implementation of elegant software.

3. What is the use of duck typing?

a) More restriction on the type values that can be passed to a given method

b) No restriction on the type values that can be passed to a given method

c) Less restriction on the type values that can be passed to a given method

d) Makes the program code smaller

Answer: c

Explanation: In Python, any set of classes with a common set of methods can be treated similarly. This is called duck typing. Hence duck typing imposes less restrictions.

4. What will be the output of the following Python code?

class A:

def \_\_str\_\_(self):

return '1'

class B(A):

def \_\_init\_\_(self):

super().\_\_init\_\_()

class C(B):

def \_\_init\_\_(self):

super().\_\_init\_\_()

def main():

obj1 = B()

obj2 = A()

obj3 = C()

print(obj1, obj2,obj3)

main()

a) 1 1 1

b) 1 2 3

c) ‘1’ ‘1’ ‘1’

d) An exception is thrown

Answer: a

Explanation: The super().\_\_init\_\_() in the subclasses has been properly invoked and none of other subclasses return any other value. Hence 1 is returned each time the object is created and printed.

5. What will be the output of the following Python code?

class Demo:

def \_\_init\_\_(self):

self.x = 1

def change(self):

self.x = 10

class Demo\_derived(Demo):

def change(self):

self.x=self.x+1

return self.x

def main():

obj = Demo\_derived()

print(obj.change())

main()

a) 11

b) 2

c) 1

d) An exception is thrown

Answer: b

Explanation: The derived class method change() overrides the base class method.

6. A class in which one or more methods are only implemented to raise an exception is called an abstract class.

a) True b) False

Answer: a => A class in which one or more methods are unimplemented or implemented for the methods throw an exception is called an abstract class.

7. Overriding means changing behavior of methods of derived class methods in the base class.

a) True b) False

Answer: b => Overriding means if there are two same methods present in the superclass and the subclass, the contents of the subclass method are executed.

8. What will be the output of the following Python code?

class A:

def \_\_repr\_\_(self):

return "1"

class B(A):

def \_\_repr\_\_(self):

return "2"

class C(B):

def \_\_repr\_\_(self):

return "3"

o1 = A()

o2 = B()

o3 = C()

print(obj1, obj2, obj3)

a) 1 1 1

b) 1 2 3

c) ‘1’ ‘1’ ‘1’

d) An exception is thrown

Answer: b => When different objects are invoked, each of the individual classes return their individual values and hence it is printed.

9. What will be the output of the following Python code?

class A:

def \_\_init\_\_(self):

self.multiply(15)

print(self.i)

def multiply(self, i):

self.i = 4 \* i;

class B(A):

def \_\_init\_\_(self):

super().\_\_init\_\_()

def multiply(self, i):

self.i = 2 \* i;

obj = B()

a) 15

b) 60

c) An exception is thrown

d) 30

Answer: d

Explanation: The derived class B overrides base class A.

10. What will be the output of the following Python code?

class Demo:

def check(self):

return " Demo's check "

def display(self):

print(self.check())

class Demo\_Derived(Demo):

def check(self):

return " Derived's check "

Demo().display()

Demo\_Derived().display()

a) Demo’s check Derived’s check

b) Demo’s check Demo’s check

c) Derived’s check Demo’s check

d) Syntax error

Answer: a

Explanation: Demo().display() invokes the display() method in class Demo and Demo\_Derived().display() invokes the display() method in class Demo\_Derived.

11. What will be the output of the following Python code?

class A:

def \_\_init\_\_(self):

self.multiply(15)

def multiply(self, i):

self.i = 4 \* i;

class B(A):

def \_\_init\_\_(self):

super().\_\_init\_\_()

print(self.i)

def multiply(self, i):

self.i = 2 \* i;

obj = B()

a) 15

b) 30

c) An exception is thrown

d) 60

Answer: b

Explanation: The derived class B overrides base class A.

12. What will be the output of the following Python code?

class Demo:

def \_\_check(self):

return " Demo's check "

def display(self):

print(self.check())

class Demo\_Derived(Demo):

def \_\_check(self):

return " Derived's check "

Demo().display()

Demo\_Derived().display()

a) Demo’s check Derived’s check

b) Demo’s check Demo’s check

c) Derived’s check Demo’s check

d) Syntax error

Answer: b

Explanation: The method check is private so it can’t be accessed by the derived class. Execute the code in the Python shell.

13. What will be the output of the following Python code?

class A:

def \_\_init\_\_(self, x, y):

self.x = x

self.y = y

def \_\_str\_\_(self):

return 1

def \_\_eq\_\_(self, other):

return self.x \* self.y == other.x \* other.y

obj1 = A(5, 2)

obj2 = A(2, 5)

print(obj1 == obj2)

a) False

b) 1

c) True

d) An exception is thrown

Answer: c

Explanation: Since 5\*2==2\*5, True is printed. Execute it in the Python shell to verify.

14. What will be the output of the following Python code?

class A:

def one(self):

return self.two()

def two(self):

return 'A'

class B(A):

def two(self):

return 'B'

obj2=B()

print(obj2.two())

a) A

b) An exception is thrown

c) A B

d) B

Answer: d

Explanation: The derived class method two() overrides the method two() in the base class A.

15. Which of the following statements is true?

a) A non-private method in a superclass can be overridden

b) A subclass method can be overridden by the superclass

c) A private method in a superclass can be overridden

d) Overriding isn’t possible in Python

Answer: a

Explanation: A public method in the base class can be overridden by the same named method in the subclass.

**Encapsulation**

<https://www.sanfoundry.com/python-questions-answers-encapsulation/>

1. Which of these is not a fundamental features of OOP?

a) Encapsulation b) Inheritance

c) Instantiation d) Polymorphism

Answer: c => Instantiation simply refers to creation of an instance of class. It is not a fundamental feature of OOP.

2. Which of the following is the most suitable definition for encapsulation?

a) Ability of a class to derive members of another class as a part of its own definition

b) Means of bundling instance variables and methods in order to restrict access to certain class members

c) Focuses on variables and passing of variables to functions

d) Allows for implementation of elegant software that is well designed and easily modified

Answer: b => The values assigned by the constructor to the class members is used to create the object.

3. What will be the output of the following Python code?

class Demo:

def \_\_init\_\_(self):

self.a = 1

self.\_\_b = 1

def display(self):

return self.\_\_b

obj = Demo()

print(obj.a)

a) The program has an error because there isn’t any function to return self.a

b) The program has an error because b is private and display(self) is returning a private member

c) The program runs fine and 1 is printed

d) The program has an error as you can’t name a class member using \_\_b

Answer: c

Explanation: The program has no error because the class member which is public is printed. 1 is displayed. Execute in python shell to verify.

4. What will be the output of the following Python code?

class Demo:

def \_\_init\_\_(self):

self.a = 1

self.\_\_b = 1

def display(self):

return self.\_\_b

obj = Demo()

print(obj.\_\_b)

a) The program has an error because there isn’t any function to return self.a

b) The program has an error because b is private and display(self) is returning a private member

c) The program has an error because b is private and hence can’t be printed

d) The program runs fine and 1 is printed

Answer: c

Explanation: Variables beginning with two underscores are said to be private members of the class and they can’t be accessed directly.

5. Methods of a class that provide access to private members of the class are called as \_\_\_\_\_\_ and \_\_\_\_\_\_

a) getters/setters

b) \_\_repr\_\_/\_\_str\_\_

c) user-defined functions/in-built functions

d) \_\_init\_\_/\_\_del\_\_

Answer: a

Explanation: The purpose of getters and setters is to get(return) and set(assign) private instance variables of a class.

6. Which of these is a private data field?

def Demo:

def \_\_init\_\_(self):

\_\_a = 1

self.\_\_b = 1

self.\_\_c\_\_ = 1

\_\_d\_\_= 1

a) \_\_a

b) \_\_b

c) \_\_c\_\_

d) \_\_d\_\_

Answer: b

Explanation: Variables such as self.\_\_b are private members of the class.

7. What will be the output of the following Python code?

class Demo:

def \_\_init\_\_(self):

self.a = 1

self.\_\_b = 1

def get(self):

return self.\_\_b

obj = Demo()

print(obj.get())

a) The program has an error because there isn’t any function to return self.a

b) The program has an error because b is private and display(self) is returning a private member

c) The program has an error because b is private and hence can’t be printed

d) The program runs fine and 1 is printed

Answer: d => Here, get(self) is a member of the class. Hence, it can even return a private member of the class. Because of this reason, the program runs fine and 1 is printed.

8. What will be the output of the following Python code?

class Demo:

def \_\_init\_\_(self):

self.a = 1

self.\_\_b = 1

def get(self):

return self.\_\_b

obj = Demo()

obj.a=45

print(obj.a)

a) The program runs properly and prints 45

b) The program has an error because the value of members of a class can’t be changed from outside the class

c) The program runs properly and prints 1

d) The program has an error because the value of members outside a class can only be changed as self.a=45

Answer: a

Explanation: It is possible to change the values of public class members using the object of the class.

9. Private members of a class cannot be accessed.

a) True b) False

Answer: b

Explanation: Private members of a class are accessible if written as follows: obj.\_Classname\_\_privatemember. Such renaming of identifiers is called as name mangling.

10. The purpose of name mangling is to avoid unintentional access of private class members.

a) True b) False

Answer: a => Name mangling prevents unintentional access of private members of a class, while still allowing access when needed. Unless the variable is accessed with its mangled name, it will not be found.

11. What will be the output of the following Python code?

class fruits:

def \_\_init\_\_(self):

self.price = 100

self.\_\_bags = 5

def display(self):

print(self.\_\_bags)

obj=fruits()

obj.display()

a) The program has an error because display() is trying to print a private class member

b) The program runs fine but nothing is printed

c) The program runs fine and 5 is printed

d) The program has an error because display() can’t be accessed

Answer: c => Private class members can be printed by methods which are members of the class.

12. What will be the output of the following Python code?

class student:

def \_\_init\_\_(self):

self.marks = 97

self.\_\_cgpa = 8.7

def display(self):

print(self.marks)

obj=student()

print(obj.\_student\_\_cgpa)

a) The program runs fine and 8.7 is printed

b) Error because private class members can’t be accessed

c) Error because the proper syntax for name mangling hasn’t been implemented

d) The program runs fine but nothing is printed

Ans: a => Name mangling has been properly implemented in the code given above and hence the program runs properly.

13. Which of the following is false about protected class members?

a) They begin with one underscore

b) They can be accessed by subclasses

c) They can be accessed by name mangling method

d) They can be accessed within a class

Answer: c => Protected class members can’t be accessed by name mangling.

14. What will be the output of the following Python code?

class objects:

def \_\_init\_\_(self):

self.colour = None

self.\_shape = "Circle"

def display(self, s):

self.\_shape = s

obj=objects()

print(obj.\_objects\_shape)

a) The program runs fine because name mangling has been properly implemented

b) Error because the member shape is a protected member

c) Error because the proper syntax for name mangling hasn’t been implemented

d) Error because the member shape is a private member

Answer: b

Explanation: Protected members begin with one underscore and they can only be accessed within a class or by subclasses.

**Exception Handling – 1**

<https://www.sanfoundry.com/python-questions-answers-exception-handling/>

1. How many except statements can a try-except block have?

a) zero b) one

c) more than one d) more than zero

Answer: d => There has to be at least one except statement.

2. When will the else part of try-except-else be executed?

a) always

b) when an exception occurs

c) when no exception occurs

d) when an exception occurs in to except block

Answer: c => The else part is executed when no exception occurs.

3. Is the following Python code valid?

try:

# Do something

except:

# Do something

finally:

# Do something

a) no, there is no such thing as finally

b) no, finally cannot be used with except

c) no, finally must come before except

d) yes

Answer: b

Explanation: Refer documentation.

4. Is the following Python code valid?

try:

# Do something

except:

# Do something

else:

# Do something

a) no, there is no such thing as else

b) no, else cannot be used with except

c) no, else must come before except

d) yes

Answer: d => Refer documentation.

5. Can one block of except statements handle multiple exception?

a) yes, like except TypeError, SyntaxError [,…]

b) yes, like except [TypeError, SyntaxError]

c) no

d) none of the mentioned

Answer: a => Each type of exception can be specified directly. There is no need to put it in a list.

6. When is the finally block executed?

a) when there is no exception

b) when there is an exception

c) only if some condition that has been specified is satisfied

d) always

Answer: d => The finally block is always executed.

7. What will be the output of the following Python code?

def foo():

try:

return 1

finally:

return 2

k = foo()

print(k)

a) 1

b) 2

c) 3

d) error, there is more than one return statement in a single try-finally block

Answer: b => The finally block is executed even there is a return statement in the try block.

8. What will be the output of the following Python code?

def foo():

try:

print(1)

finally:

print(2)

foo()

a) 1 2

b) 1

c) 2

d) none of the mentioned

Ans: a => No error occurs in the try block so 1 is printed. Then the finally block is executed and 2 is printed.

9. What will be the output of the following Python code?

try:

if '1' != 1:

raise "someError"

else:

print("someError has not occurred")

except "someError":

print ("someError has occurred")

a) someError has occurred b) someError has not occurred

c) invalid code d) none of the mentioned

Answer: c => A new exception class must inherit from a BaseException. There is no such inheritance here.

10. What happens when ‘1’ == 1 is executed?

a) we get a True b) we get a False

c) an TypeError occurs d) a ValueError occurs

Answer: b => It simply evaluates to False and does not raise any exception.

1. The following Python code will result in an error if the input value is entered as -5.

assert False, 'Spanish'

a) True b) False

Answer: a

Explanation: The code shown above results in an assertion error. The output of the code is:

Traceback (most recent call last):

File “<pyshell#0>”, line 1, in <module>

assert False, ‘Spanish’

AssertionError: Spanish

Hence, this statement is true.

2. What will be the output of the following Python code?

x=10

y=8

assert x>y, 'X too small'

a) Assertion Error

b) 10 8

c) No output

d) 108

Ans: c => The code shown above results in an error if and only if x<y. However, in the above case, since x>y, there is no error. Since there is no print statement, hence there is no output.

3. What will be the output of the following Python code?

#generator

def f(x):

yield x+1

g=f(8)

print(next(g))

a) 8

b) 9

c) 7

d) Error

Ans: b => The code shown above returns the value of the expression x+1, since we have used to keyword yield. The value of x is 8. Hence the output of the code is 9.

4. What will be the output of the following Python code?

def f(x):

yield x+1

print("test")

yield x+2

g=f(9)

a) Error

b) test

c)

test

10

12

d) No output

Ans: d => The code shown above will not yield any output. This is because when we try to yield 9, and there is no next(g), the iteration stops. Hence there is no output.

5. What will be the output of the following Python code?

def f(x):

yield x+1

print("test")

yield x+2

g=f(10)

print(next(g))

print(next(g))

a) No output

b)

11

test

12

c)

11

test

d) 11

Answer: b => The code shown above results in the output:

11

test

12

This is because we have used next(g) twice. Had we not used next, there would be no output.

6. What will be the output of the following Python code?

def a():

try:

f(x, 4)

finally:

print('after f')

print('after f?')

a()

a) No output b) after f?

c) error d) after f

Ans: c => This code shown above will result in an error simply because ‘f’ is not defined. ‘try’ and ‘finally’ are keywords used in exception handling.

7. What will be the output of the following Python code?

def f(x):

for i in range(5):

yield i

g=f(8)

print(list(g))

a) [0, 1, 2, 3, 4]

b) [1, 2, 3, 4, 5, 6, 7, 8]

c) [1, 2, 3, 4, 5]

d) [0, 1, 2, 3, 4, 5, 6, 7]

Ans: a => The output of the code shown above is a list containing whole numbers in the range (5). Hence the output of this code is: [0, 1, 2, 3, 4].

8. The error displayed in the following Python code is?

import itertools

l1=(1, 2, 3)

l2=[4, 5, 6]

l=itertools.chain(l1, l2)

print(next(l1))

a) ‘list’ object is not iterator b) ‘tuple’ object is not iterator

c) ‘list’ object is iterator d) ‘tuple’ object is iterator

Ans: b => The error raised in the code shown above is that: ‘tuple’ object is not iterator. Had we given l2 as argument to next, the error would have been: ‘list’ object is not iterator.

9. Which of the following is not an exception handling keyword in Python?

a) try b) except c) accept d) finally

Ans: c => The keywords ‘try’, ‘except’ and ‘finally’ are exception handling keywords in python whereas the word ‘accept’ is not a keyword at all.

10. What will be the output of the following Python code?

g = (i for i in range(5))

type(g)

a) class <’loop’> b) class <‘iteration’>

c) class <’range’> d) class <’generator’>

Ans: d => Another way of creating a generator is to use parenthesis. Hence the output of the code shown above is: class<’generator’>.

1. What happens if the file is not found in the following Python code?

a=False

while not a:

try:

f\_n = input("Enter file name")

i\_f = open(f\_n, 'r')

except:

print("Input file not found")

a) No error b) Assertion error

c) Input output error d) Name error

Ans: a => In the code shown above, if the input file in not found, then the statement: “Input file not found” is printed on the screen. The user is then prompted to reenter the file name. Error is not thrown.

2. What will be the output of the following Python code?

lst = [1, 2, 3]

lst[3]

a) NameError b) ValueError c) IndexError d) TypeError

Ans: c => The snippet of code shown above throws an index error. This is because the index of the list given in the code, that is, 3 is out of range. The maximum index of this list is 2.

3. What will be the output of the following Python code?

t[5]

a) IndexError b) NameError c) TypeError d) ValeError

Ans: b => The expression shown above results in a name error. This is because the name ‘t’ is not defined.

4. What will be the output of the following Python code, if the time module has already been imported?

4 + '3'

a) NameError b) IndexError c) ValueError d) TypeError

Ans: d => The line of code shown above will result in a type error. This is because the operand ‘+’ is not supported when we combine the data types ‘int’ and ‘str’. Sine this is exactly what we have done in the code shown above, a type error is thrown.

5. What will be the output of the following Python code?

int('65.43')

a) ImportError b) ValueError c) TypeError d) NameError

Ans: b => The snippet of code shown above results in a value error. This is because there is an invalid literal for int() with base 10: ’65.43’.

6. Compare the following two Python codes shown below and state the output if the input entered in each case is -6?

CODE 1

import math

num=int(input("Enter a number of whose factorial you want to find"))

print(math.factorial(num))

CODE 2

num=int(input("Enter a number of whose factorial you want to find"))

print(math.factorial(num))

a) ValueError, NameError b) AttributeError, ValueError

c) NameError, TypeError d) TypeError, ValueError

Ans: a => The first code results in a ValueError. This is because when we enter the input as -6, we are trying to find the factorial of a negative number, which is not possible. The second code results in a NameError. This is because we have not imported the math module. Hence the name ‘math’ is undefined.

7. What will be the output of the following Python code?

def getMonth(m):

if m<1 or m>12:

raise ValueError("Invalid")

print(m)

getMonth(6)

a) ValueError b) Invalid

c) 6 d) ValueError(“Invalid”)

Ans: c => In the code shown above, since the value passed as an argument to the function is between 1 and 12 (both included), hence the output is the value itself, that is 6. If the value had been above 12 and less than 1, a ValueError would have been thrown.

8. What will be the output of the following Python code if the input entered is 6?

valid = False

while not valid:

try:

n=int(input("Enter a number"))

while n%2==0:

print("Bye")

valid = True

except ValueError:

print("Invalid")

a) Bye (printed once) b) No output

c) Invalid (printed once) d) Bye (printed infinite number of times)

Ans: d => The code shown above results in the word “Bye” being printed infinite number of times. This is because an even number has been given as input. If an odd number had been given as input, then there would have been no output.

9. Identify the type of error in the following Python codes?

Print(“Good Morning”)

print(“Good night)

a) Syntax, Syntax b) Semantic, Syntax

c) Semantic, Semantic d) Syntax, Semantic

Ans: b => The first code shows an error detected during execution. This might occur occasionally. The second line of code represents a syntax error. When there is deviation from the rules of a language, a syntax error is thrown.

10. Which of the following statements is true?

a) The standard exceptions are automatically imported into Python programs

b) All raised standard exceptions must be handled in Python

c) When there is a deviation from the rules of a programming language, a semantic error is thrown

d) If any exception is thrown in try block, else block is executed

Ans: a => When any exception is thrown in try block, except block is executed. If exception in not thrown in try block, else block is executed. When there is a deviation from the rules of a programming language, a syntax error is thrown. The only true statement above is: The standard exceptions are automatically imported into Python programs.

11. Which of the following is not a standard exception in Python?

a) NameError b) IOError c) AssignmentError d) ValueError

Ans: c => NameError, IOError and ValueError are standard exceptions in Python whereas Assignment error is not a standard exception in Python.

12. Syntax errors are also known as parsing errors.

a) True b) False

Ans: a => Syntax errors are known as parsing errors. Syntax errors are raised when there is a deviation from the rules of a language. Hence the statement is true.

13. An exception is \_\_\_\_\_\_\_\_\_\_\_\_

a) an object b) a special function

c) a standard module d) a module

Ans: a => An exception is an object that is raised by a function signaling that an unexpected situation has occurred, that the function itself cannot handle.

14. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ exceptions are raised as a result of an error in opening a particular file.

a) ValueError b) TypeError c) ImportError d) IOError

Ans: d => IOError exceptions are raised as a result of an error in opening or closing a particular file.

15. Which of the following blocks will be executed whether an exception is thrown or not?

a) except b) else c) finally d) assert

Ans: c => The statements in the finally block will always be executed, whether an exception is thrown or not. This clause is used to close the resources used in a code.