Python Questions and Answers – 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

View Answer

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’].

View Answer

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

View Answer

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

View Answer

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

View Answer

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

View Answer

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)

View Answer

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

View Answer

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

View Answer

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].

View Answer

Answer: a
Explanation: Execute in the shell to verify.

Python Questions and Answers – Lists – 2

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

1. What is the output when following code is executed ?

```
>>>names = ['Amir', 'Bear', 'Charlton', 'Daman']
>>>print(names[-1][-1])
```

a) A

b) Daman

c) Error

d) n

View Answer

Answer: d
Explanation: Execute in the shell to verify.

2. What is the output when following code is executed ?

```
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

View Answer

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].

View Answer

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].

View Answer

Answer: c
Explanation: Execute in the shell to verify.

5. What is the output when following code is executed ?

```
>>>list1 = [11, 2, 23]
>>>list2 = [11, 2, 2]
>>>list1 < list2 is
```

a) True

b) False

c) Error

d) None

View Answer

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)

View Answer

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)

View Answer

Answer: a
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”)

View Answer

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

View Answer

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

View Answer

Answer: d
Explanation: Execute in the shell to verify.

Python Questions and Answers – Lists – 3

This set of Python Programming Questions & Answers focuses on “Lists”.

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].
View Answer

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].
View Answer

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].
View Answer

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].
View Answer

Answer: a
Explanation: pop() by default will remove the last element.

5. What is the output when the following code is executed ?

```
>>>"Welcome to Python".split()
```

a) [“Welcome”, “to”, “Python”].

b) (“Welcome”, “to”, “Python”)

c) {“Welcome”, “to”, “Python”}

d) “Welcome”, “to”, “Python”

View Answer

Answer: a
Explanation: split() function returns the elements in a list.

6. What is the output when following code is executed ?

```
>>>list("a#b#c#d".split('#'))
```

a) [‘a’, ‘b’, ‘c’, ‘d’].

b) [‘a b c d’].

c) [‘a#b#c#d’].

d) [‘abcd’].

View Answer

Answer: a
Explanation: Execute in the shell to verify.

7. What is the output when following code is executed ?

```
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

View Answer

Answer: a
Explanation: First time the highest number is encountered is at index 1.

8. What is the output when following code is executed ?

```
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

View Answer

Answer: c
Explanation: Execute in the shell to verify.

9. What is the output when following code is executed ?

```
>>>list1 = [1, 3]
>>>list2 = list1
>>>list1[0] = 4
>>>print(list2)
```

a) [1, 3].

b) [4, 3].

c) [1, 4].

d) [1, 3, 4].

View Answer

Answer: b
Explanation: Lists should be copied by executing [:] operation.

10. What is the output when following code is executed ?

```
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].

View Answer

Answer: c
Explanation: Execute in the shell to verify.

Python Questions and Answers – Lists – 4

This set of Python Programming Interview Questions & Answers focuses on “Lists”.

1. What will be the output?

```
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

View Answer

Answer: c
Explanation: Execute in the shell to verify

2. What will be the output?

```
names1 = ['Amir', 'Bala', 'Chales']

if 'amir' in names1:
    print(1)
else:
    print(2)
```

a) None

b) 1

c) 2

d) Error

View Answer

Answer: c
Explanation: Execute in the shell to verify.

3. What will be the output?

```
names1 = ['Amir', 'Bala', 'Charlie']
names2 = [name.lower() for name in names1]

print(names2[2][0])
```

a) None

b) a

c) b

d) c

View Answer

Answer: d
Explanation: List Comprehension are a shorthand for creating new lists.

4. What will be the output?

```
numbers = [1, 2, 3, 4]

numbers.append([5,6,7,8])

print(len(numbers))
```

a) 4

b) 5

c) 8

d) 12

View Answer

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

View Answer

Answer: d
Explanation: In can be used in all data structures.

6. What will be the output?

```
list1 = [1, 2, 3, 4]
list2 = [5, 6, 7, 8]

print(len(list1 + list2))
```

a) 2

b) 4

c) 5

d) 8

View Answer

Answer: d
Explanation: + appends all the elements individually into a new list.

7. What will be the output?

```
def addItem(listParam):
    listParam += [1]

mylist = [1, 2, 3, 4]
addItem(mylist)
print(len(mylist))
```

a) 1

b) 4

c) 5

d) 8

View Answer

Answer: c
Explanation: + will append the element to the list.

8. What will be the output?

```
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

View Answer

Answer: a
Explanation: Execute in the shell to verify.

9. What will be the output?

```
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

View Answer

Answer: a
Explanation: Run the code to get a better understanding with many arguements.

10. What will be the output?

```
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’].

View Answer

Answer: a
Explanation: Execute in the shell to verify.

Python Questions and Answers – Lists – 5

This set of Python Question Paper focuses on “Lists”.

1. What will be the output?

```
>>>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].

View Answer

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

View Answer

Answer: c
Explanation: Execute in the shell to verify.

3. What will be the output?

```
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

View Answer

Answer: d
Explanation: Execute in the shell to verify.

4. What will be the output?

```
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

View Answer

Answer: a
Explanation: Execute in the shell to verify.

5. What will be the output?

```
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

View Answer

Answer: d
Explanation: Execute in the shell to verify.

6. What is the output?

```
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

View Answer

Answer: d
Explanation: Execute in the shell to verify.

7. What will be the output?

```
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]]

```

```python
for row in values:
    print(m(row), end = " ")
```

a) 3 33

b) 1 1

c) 5 6

d) 5 33

View Answer

Answer: d
Explanation: Execute in the shell to verify.

8. What will be the output?

```python
data = [[[1, 2], [3, 4]], [[5, 6], [7, 8]]]

print(data[1][0][0])
```

a) 1

b) 2

c) 4

d) 5

View Answer

Answer: d
Explanation: Execute in the shell to verify.

9. What will be the output?

```python
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

View Answer

Answer: c
Explanation: Execute in the shell to verify.

10. What will be the output?

```
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]].

View Answer

Answer: c
Explanation: Execute in the shell to verify.

Python Questions and Answers – Lists – 6

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Lists-6”.

1. What is the output of the following 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]].

View Answer

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 does the following piece of code do?

```
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’)].

View Answer

Answer: d
Explanation: The zip function combines the individual attributes of the lists into a list of tuples.

3. What is the output of the following 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]].

View Answer

Answer: b
Explanation: The above copy is deepcopy. Any change made in the original list isn’t reflected.

4. What is the output of the following piece of 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’].

View Answer

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 is the output of the following 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].

View Answer

Answer: d
Explanation: The above code returns the cumulative sum of elements in a list.

6. What is the output of the following 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

View Answer

Answer: a
Explanation: Variable x iterates over each letter in string a hence the length of each letter is 1.

7. What is the output of the following 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].

View Answer

Answer: d
Explanation: The above code returns the cumulative sum of elements in a list.

8. What is the output of the following code?

```
a=[[]]*3
a[1].append(7)
print(a)
```

a) Syntax error

b) [[7], [7], [7]].

c) [[7], [], []].

d) [[],7, [], []].

View Answer

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 is the output of the following 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

View Answer

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 is the output of the following 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].

View Answer

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.

Python Questions and Answers – Lists – 7

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Lists-7”.

1. What is the output of the following 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].

View Answer

Answer: b
Explanation: Append function on lists doesn’t return anything. Thus the value of b is None.

2. What will be the output when executed in python shell?

```
>>> a=[14,52,7]
>>>> b=a.copy()
>>> b is a
```

a) True

b) False

View Answer

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 is the output of the following 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]].

View Answer

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

View Answer

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 is the output of the following code?

```
lst=[[1,2],[3,4]]
print(sum(lst,[]))
```

a) [[3],[7]].

b) [1,2,3,4].

c) Error

d) [10].

View Answer

Answer: b
Explanation: The above piece of code is used for flattening lists.

6. What is the output of the following 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

View Answer

Answer: d
Explanation: In the above case, both the lists are equivalent but not identical as they have different objects.

7. What is the output of the following 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

View Answer

Answer: d
Explanation: unpack(*x) unpacks the list into the separate variables. Now, a=1 and d=4. Thus 5 gets printed.

8. What is the output of the following 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’].

View Answer

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 is the output of the following piece of code?

```
x=[[1],[2]]
print(" ".join(list(map(str,x))))
```

a) [1] [2].

b) [49] [50].

c) Syntax error

d) [[1]] [[2]].

View Answer

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 is the output of the following code?

```
a=165
b=sum(list(map(int,str(a))))
print(b)
```

a) 561

b) 5

c) 12

d) Syntax error

View Answer

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 is the output of the following 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

View Answer

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 is the output of the following 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].

View Answer

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 is the output of the following 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

View Answer

Answer: c
Explanation: The above piece of code basically prints the index of the largest element in the list.

14. What is the output of the following 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().

View Answer

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 is the output of the following 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

View Answer

Answer: d
Explanation: enumerate(iterator,start=0) is a buit-in function which returns (0,lst[0]),(1,lst[1]) and so on where lst is a list(iterator).

Python Questions and Answers – List Comprehension

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

1. What is the output of the following?

```
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)

View Answer

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?

```
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

View Answer

Answer: b
Explanation: print() returns None.

3. What is the output of the following?

```
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

View Answer

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

View Answer

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 is the output of the following?

```
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

View Answer

Answer: a
Explanation: i**+1 is evaluated as (i)**(+1).

6. What is the output of the following?

```
print([i.lower() for i in "HELLO"])
```

a) [‘h’, ‘e’, ‘l’, ‘l’, ‘o’].

b) ‘hello’

c) [‘hello’].

d) hello

View Answer

Answer: a
Explanation: We are iterating over each letter in the string.

7. What is the output of the following?

```
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’].

View Answer

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 is the output of the following?

```
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’].

View Answer

Answer: b
Explanation: The inner list is generated once for each value of j.

9. What is the output of the following?

```
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

View Answer

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

View Answer

Answer: c
Explanation: x**-1 is evaluated as (x)**(-1).

Python Questions and Answers – List Comprehension – 1

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

1. What is the output of the code shown?

```
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]

View Answer

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 is the output of the code shown below?

```
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]

View Answer

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]

View Answer

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 is the output of the code shown?

```
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)]

View Answer

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 is the output of the code shown below?

```
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]

View Answer

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 is the output of the following 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]

View Answer

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]

View Answer

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 is the output of the code shown below?

```
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)

View Answer

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 is the output of the code shown below?

```
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’]

View Answer

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 is the output of the code shown below?

```
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

View Answer

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).

Python Questions and Answers – List Comprehension – 2

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

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]

View Answer

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 is the output of the code shown below?

```
[ord(ch) for ch in 'abc']
```

a) [97, 98, 99]

b) [‘97’, ‘98’, ‘99’]

c) [65, 66, 67]

d) Error

View Answer

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 is the output of the code shown below?

```
t=32.00
[round((x-32)*5/9) for x in t]
```

a) [0]

b) 0

c) [0.00]

d) Error

View Answer

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)]

View Answer

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 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)]

View Answer

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)]

View Answer

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)]

View Answer

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)]

View Answer

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 is the output of the list comprehension shown below?

```
[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

View Answer

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 is the output of the code shown below?

```
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’]

View Answer

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’].

Python Questions and Answers – Matrix List Comprehension

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “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]
```

View Answer

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 is the output of the snippet of code shown below?

```
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]

View Answer

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 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]

View Answer

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 is the output of the code shown?

```
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]

View Answer

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 is the output of the code shown below?

```
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]

View Answer

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 is the output of the following 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]]

View Answer

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 is the output of the code shown?

```
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]

View Answer

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 is the output of the code shown below?

```
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]

View Answer

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. The service in which the cloud consumer does not manage the underlying cloud infrastructure nor the application(s) contained in the instances as the service provides user with all of it is:

```
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

View Answer

Answer: c
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: [0, 30, 60, 120, 160, 200, 300, 350, 400].

10. What is the output of the code shown?

```
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]

View Answer

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 is the output of the code shown?

```
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) [[0, 30, 60], [120, 160, 200], [300, 350, 400]]

c) No output

d) Error

View Answer

Answer: b
Explanation: The list comprehension shown above results in the output: [[0, 30, 60], [120, 160, 200], [300, 350, 400]].

12. What is the output of the code shown?

```
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]

View Answer

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))
[([0, 10, 20], [3, 3, 3]), ([30, 40, 50], [4, 4, 4]), ([60, 70, 80], [5, 5, 5])]

Python Questions and Answers – Tuples – 1

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

1. Which of the following is a Python tuple?

a) [1, 2, 3].

b) (1, 2, 3)

c) {1, 2, 3}

d) {}

View Answer

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))

View Answer

Answer: b
Explanation: Values cannot be modified in the case of tuple, that is, tuple is immutable.

3. What will be the output?

```
>>>t=(1,2,4,3)
>>>t[1:3]
```

a) (1, 2)

b) (1, 2, 4)

c) (2, 4)

d) (2, 4, 3)

View Answer

Answer: c
Explanation: Slicing in tuples takes place just as it does in strings.

4. What will be the output?

```
>>>t=(1,2,4,3)
>>>t[1:-1]
```

a) (1, 2)

b) (1, 2, 4)

c) (2, 4)

d) (2, 4, 3)

View Answer

Answer: c
Explanation: Slicing in tuples takes place just as it does in strings.

5. What will be the output?

```
>>>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)

View Answer

Answer: c
Explanation: Execute in the shell to verify.

6. What will be the output?

```
d = {"john":40, "peter":45}
d["john"]
```

a) 40

b) 45

c) “john”

d) “peter”

View Answer

Answer: a
Explanation: Execute in the shell to verify.

7. What will be the output?

```
>>>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].

View Answer

Answer: a
Explanation: * operator concatenates tuple.

8. What will be the output?

```
>>>t1 = (1, 2, 4, 3)
>>>t2 = (1, 2, 3, 4)
>>>t1 < t2
```

a) True

b) False

c) Error

d) None

View Answer

Answer: b
Explanation: Elements are compared one by one in this case.

9. What will be the output?

```
>>>my_tuple = (1, 2, 3, 4)
>>>my_tuple.append( (5, 6, 7) )
>>>print len(my_tuple)
```

a) 1

b) 2

c) 5

d) Error

View Answer

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?

```
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

View Answer

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.

Python Questions and Answers – Tuples – 2

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

1. What is the data type of (1)?

a) Tuple

b) Integer

c) List

d) Both tuple and integer

View Answer

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)

View Answer

Answer: d
Explanation: Tuple slicing exists and a[1:-1] returns (2,3).

3. What is the output of the following 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

View Answer

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 is the output of the following piece of code when executed in Python shell?

```
>>> a=("Check")*3
>>> a
```

a) (‘Check’,’Check’,’Check’)

b) * Operator not valid for tuples

c) (‘CheckCheckCheck’)

d) Syntax error

View Answer

Answer: c
Explanation: Here (“Check”) is a string not a tuple because there is no comma after the element.

5. What is the output of the following 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

View Answer

Answer: d
Explanation: ‘tuple’ object doesn’t support item deletion.

6. What is the output of the following 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

View Answer

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 piece of 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

View Answer

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

View Answer

Answer: b
Explanation: The variable a has tuples enclosed in a list making it a list of tuples.

9. What is the output of the following piece of 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)

View Answer

Answer: c
Explanation: The method illustrated in the above piece of code is that of naming of slices.

10. Is the following piece of code valid?

```
>>> a=(1,2,3)
>>> b=('A','B','C')
>>> c=zip(a,b)
```

a) Yes, c will be ((1,2,3),(‘A’,’B’,’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

View Answer

Answer: a
Explanation: Zip function combines individual elements of two iterrables into tuples. Execute in Python shell to verify.

Python Questions and Answers – Tuples-3

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Tuples – 3”.

1. Is the following piece of 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

View Answer

Answer: c
Explanation: A tuple needn’t be enclosed in parenthesis.

2. What is the output of the following piece of 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’,)

View Answer

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 line of 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)

View Answer

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 is the output of the following piece of code when executed in Python shell?

```
>>> 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

View Answer

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 is the output of the following code?

```
>>> a,b=6,7
>>> a,b=b,a
>>> a,b
```

a) (6,7)

b) Invalid syntax

c) (7,6)

d) Nothing is printed

View Answer

Answer: c
Explanation: The above piece of code illustrates the unpacking of variables.

6. What is the output of the following 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

View Answer

Answer: a
Explanation: The above piece of code illustrates the concept of named tuples.

7. Tuples can't be made keys of a dictionary. True or False?

a) True

b) False

View Answer

Answer: b
Explanation: Tuples can be made keys of a dictionary because they are hashable.

8. Is the following piece of 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

View Answer

Answer: d
Explanation: A tuple needn't be enclosed in parenthesis.

9. What is the output of the following piece of 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

View Answer

Answer: d
Explanation: A tuple is immutable thus it doesn't have a sort attribute.

10. Is the following piece of 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

View Answer

Answer: c
Explanation: Tuple doesn't have any update() attribute because it is immutable.

11. What is the output of the following piece of 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

View Answer

Answer: d
Explanation: A list of tuples is a list itself. Hence items of a list can be sorted.

Python Questions and Answers – 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

View Answer

Answer: d
Explanation: Dictionaries are created by specifying keys and values.

2. Read the code shown below carefully and pick out the keys?

```
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”)

View Answer

Answer: b
Explanation: Dictionaries appear in the form of keys and values.

3. What will be the output?

```
d = {"john":40, "peter":45}
"john" in d
```

a) True

b) False

c) None

d) Error

View Answer

Answer: a
Explanation: In can be used to check if the key is int dictionary.

4. What will be the output?

```
d1 = {"john":40, "peter":45}
d2 = {"john":466, "peter":45}
d1 == d2
```

a) True

b) False

c) None

d) Error

View Answer

Answer: b
Explanation: If d2 was initialized as d2 = d1 the answer would be true.

5. What will be the output?

```
d1 = {"john":40, "peter":45}
d2 = {"john":466, "peter":45}
d1 > d2
```

a) True

b) False

c) Error

d) None

View Answer

Answer: c
Explanation: Arithmetic > operator cannot be used with dictionaries.

6. What is the output?

```
d = {"john":40, "peter":45}
d["john"]
```

a) 40

b) 45

c) “john”

d) “peter”

View Answer

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)

View Answer

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()

View Answer

Answer: b
Explanation: Execute in the shell to verify.

9. What will be the output?

```
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)

View Answer

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

View Answer

Answer: c
Explanation: Execute in the shell to verify.

Python Questions and Answers – Dictionary – 2

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

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

View Answer

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) { }

View Answer

Answer: c
Explanation: Option c is a set, not a dictionary.

3. What is the output of the following code?

```
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"

View Answer

Answer: a
Explanation: In the above code, variables i and j iterate over the keys and values of the dictionary respectively.

4. What is the output of the following piece of code?

```
a={1:"A",2:"B",3:"C"}
print(a.get(1,4))
```

a) 1

b) A

c) 4

d) Invalid syntax for get method

View Answer

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 is the output of the following code?

```
a={1:"A",2:"B",3:"C"}
```

```
print(a.get(5,4))
```

a) Error, invalid syntax

b) A

c) 5

d) 4

View Answer

Answer: d
Explanation: The get() method returns the default value(second parameter) if the key isn’t present in the dictionary.

6. What is the output of the following code?

```
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

View Answer

Answer: b
Explanation: setdefault() is similar to get() but will set dict[key]=default if key is not already in the dictionary.

7. What is the output of the following code?

```
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].

View Answer

Answer: a
Explanation: setdefault() will set dict[key]=default if key is not already in the dictionary.

8. What is the output of the following 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’}

View Answer

Answer: c
Explanation: update() method adds dictionary b’s key-value pairs to dictionary a. Execute in python shell to verify.

9. What is the output of the following 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

View Answer

Answer: b
Explanation: Changes made in the copy of the dictionary isn’t reflected in the original one.

10. What is the output of the following 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) { }

View Answer

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

View Answer

Answer: c
Explanation: Keys of a dictionary may be any data type that is immutable.

12. What is the output of the following 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}

View Answer

Answer: b
Explanation: pop() method removes the key-value pair for the key mentioned in the pop() method.

13. What is the output of the following 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

View Answer

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 is the output of the following 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():

View Answer

Answer: a
Explanation: The variable i iterates over the keys of the dictionary and hence the keys are printed.

15. Execute the following in Python shell?

```
>>> 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’)])

View Answer

Answer: d
Explanation: The method items() returns list of tuples with each tuple having a key-value pair.

Python Questions and Answers – Dictionary – 3

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Dictionary – 3”.

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

View Answer

Answer: c
Explanation: More than one key can have the same value.

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

```
>>> 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

View Answer

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

View Answer

Answer: a
Explanation: The method popitem() removes a random key-value pair.

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

```
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

View Answer

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 is the output of the following snippet of code?

```
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

View Answer

Answer: c
Explanation: The above piece of code basically finds the sum of the values of keys.

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

```
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’}}

View Answer

Answer: d

Explanation: Dictionary in a dictionary can exist.

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

```
test = {1:'A', 2:'B', 3:'C'}
test = {}
print(len(test))
```

a) 0

b) None

c) 3

d) An exception is thrown

View Answer

Answer: a

Explanation: In the second line of code, the dictionary becomes an empty dictionary. Thus, length=0.

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

```
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

View Answer

Answer: b

Explanation: After the key-value pair of 1:’A’ is deleted, the key-value pair of 1:’D’ is added.

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

```
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

View Answer

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 is the output of the following snippet of code?

```
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}

View Answer

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 is the output of the following piece of code?

```
>>>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}

View Answer

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 is the output of the following piece of code?

```
>>>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)].

View Answer

Answer: d
Explanation: The most_common() method returns the n number key-value pairs where the value is the most recurring.

13. What is the output of the following piece of code?

```
>>>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)].

View Answer

Answer: d
Explanation: The most_common() method returns the n number key-value pairs where the value is the most recurring.

14. What is the output of the following piece of code?

```
>>> 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})

View Answer

Answer: a
Explanation: a|b returns the pair of keys and the highest recurring value.

15. What is the output of the following piece of code?

```
>>> 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})

View Answer

Answer: b
Explanation: a&b returns the pair of keys and the lowest recurring value.

Python Questions and Answers – Dictionary – 4

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Dictionary – 4”.

1. The following piece of code is invalid. True or False?

```
class demo(dict):
  def __test__(self,key):
    return []
a = demo()
a['test'] = 7
print(a)
```

a) True

b) False

View Answer

Answer: b
Explanation: The output of the code is: {‘test’:7}.

2. What is the output of the following 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

View Answer

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 is the output of the following 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

View Answer

Answer: b
Explanation: Now, a={1:[2,3,4],2:1} . a[1][1] refers to second element having key 1.

4. What is the output of the following piece of 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].

View Answer

Answer: a
Explanation: Return a new sorted list of keys in the dictionary.

5. What is the output of the following snippet of 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}

View Answer

Answer: d
Explanation: Dictionary comprehension is implemented in the above piece of code.

6. What is the output of the following piece of 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}

View Answer

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 is the output of the following snippet of code?

```
>>> b={}
>>> all(b)
```

a) { }

b) False

c) True

d) An exception is thrown

View Answer

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

View Answer

Answer: a
Explanation: Method any() returns True if any key of the dictionary is true and False if the dictionary is empty.

9. What is the output of the following 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'}

View Answer

Answer: d
Explanation: The above piece of code inverts the key-value pairs in the dictionary.

10. What is the output of the following piece of code when executed in Python shell?

```
>>> 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’}

View Answer

Answer: d
Explanation: Dictionary comprehension and string concatenation is implemented in the above piece of code.

11. What is the output of the following piece of code when executed in Python shell?

```
>>> a=dict()
>>> a[1]
```

a) An exception is thrown since the dictionary is empty

b) ‘ ‘

c) 1

d) 0

View Answer

Answer: a
Explanation: The values of a dictionary can be accessed through the keys only if the keys exist in the dictionary.

12. What is the output of the following piece of code when executed in Python shell?

```
>>> import collections
>>> a=dict()
>>> a=collections.defaultdict(int)
>>> a[1]
```

a) 1

b) 0

c) An exception is thrown

d) ‘ ‘

View Answer

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 is the output of the following piece of code when executed in Python shell?

```
>>> 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

View Answer

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 is the output of the following piece of code when executed in Python shell?

```
>>> import collections
>>> b=dict()
>>> b=collections.defaultdict(lambda: 7)
>>> b[4]
```

a) 4

b) 0

c) An exception is thrown

d) 7

View Answer

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 is the output of the following piece of code when executed in Python shell?

```
>>> 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) ‘ ‘

View Answer

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.

Python Question and Answers – Built-in Functions – 1

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

View Answer

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 is the output of the expression:

```
round(4.576)
```

a) 4.5

b) 5

c) 4

d) 4.6

View Answer

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

View Answer

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 is the output of the function shown below?

```
all([2,4,0,6])
```

a) Error

b) True

c) False

c) 0

View Answer

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 is the output of the expression?

```
round(4.5676,2)?
```

a) 4.5

b) 4.6

c) 4.57

d) 4.56

View Answer

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 is the output of the following function?

```
any([2>8, 4>2, 1>2])
```

a) Error

b) True

c) False

d) 4>2

View Answer

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 is the output of the function shown below?

```
import math
abs(math.sqrt(25))
```

a) Error

b) -5

c) 5

d) 5.0

View Answer

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 are the outcomes of the functions shown below?

```
sum(2,4,6)
sum([1,2,3])
```

a) Error, 6

b) 12, Error

c) 12, 6

d) Error, Error

View Answer

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 is the output of the function:

```
all(3,0,4.2)
```

a) True

b) False

c) Error

d) 0

View Answer

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 is the output of the functions shown below?

```
min(max(False,-3,-4), 2,7)
```

a) 2

b) False

c) -3

d) -4

View Answer

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).

Python Question and Answers – Built-in Functions – 2

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on "Built-in Functions – 2".

1. What are the outcomes of the following functions?

```
chr('97')
chr(97)
```

a) a

Error

b) 'a'

a

c) Error

a

d) Error

Error

View Answer

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 is the output of the following function?

```
complex(1+2j)
```

a) Error

b) 1

c) 2j

d) 1+2j

View Answer

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

View Answer

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)

c) (a/b, a%b)

View Answer

Answer: b
Explanation: The function divmod(a,b) is evaluated as a//b, a%b, if both ‘a’ and ‘b’ are integers.

5. What is the output of the functions shown below?

```
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)

View Answer

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. State whether this statement is true or false.

a) True

b) False

View Answer

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 is the output of the function shown below?

```
list(enumerate([2, 3]))
```

a) Error

b) [(1, 2), (2, 3)]

c) [(0, 2), (1, 3)]

d) [(2, 3)]

View Answer

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 are the outcomes of the function shown below?

```
x=3
eval('x^2')
```

a) Error

b) 1

c) 9

d) 6

View Answer

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 is the output of the functions shown below?

```
float('1e-003')
float('2e+003')
```

a) 3.00

300

b) 0.001

2000.0

c) 0.001

200

d) Error

2003

View Answer

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()

View Answer

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.

Python Question and Answers – Built-in Functions – 3

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Built-in Functions – 3”.

1. Which of the following functions accepts only integers as arguments?

a) ord()

b) min()

c) chr()

d) any()

View Answer

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))

View Answer

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. The output of the 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)

View Answer

Answer: b
Explanation: The function float() will remove all the blank spaces and covert the integer to a floating point number. Hence the output will be: -12345.0.

4. What is the output of the functions shown below?

```
ord(65)
ord(‘A’)
```

a) A

65

b) Error

65

c) A

Error

c) Error

Error

View Answer

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 is the output of the functions shown below?

```
float(‘-infinity’)
float(‘inf’)
```

a) –inf

inf

b) –infinity

inf

c) Error

Error

d) Error

Junk value

View Answer

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()

View Answer

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 is the output of the function shown below?

```
hex(15)
```

a) f

b) 0xF

c) 0Xf

d) 0xf

View Answer

8. Which of the following functions does not throw an error?

a) ord()

b) ord(‘ ‘)

c) ord(”)

d) ord(“”)

View Answer

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 is the output of the function:

```
len(["hello",2, 4, 6])
```

a) 4

b) 3

c) Error

d) 6

View Answer

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 is the output of the function shown below?

```
oct(7)
oct(‘7’)
```

a) Error
07

b) 0o7
Error

c) 0o7
Error

d) 07
0o7

View Answer

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.

Python Questions and Answers – Function – 1

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

View Answer

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 use for function?

a) Fun

b) Define

c) Def

d) Function

View Answer

Answer: c
Explanation: None.

3. What is the output of the below program?

```
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

View Answer

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 is the output of the below program?

```
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

View Answer

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 is the output of the below program ?

```
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

View Answer

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 is the output of the below program?

```python
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

View Answer

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 is the output of below program?

```python
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

View Answer

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 is the output of the below program?

```
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

View Answer

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 is the output of below program?

```python
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
View Answer

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 features 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
View Answer

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.

Python Questions and Answers – Function – 2

This set of Python Questions for entrance examinations focuses on “Functions”.

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

View Answer

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

View Answer

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

View Answer

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

View Answer

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

View Answer

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

View Answer

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
The sqrt() function is imported from the math module.

7. What is the output of below program?

```
def cube(x):
    return x * x * x
x = cube(3)
print x
```

a) 9

b) 3

c) 27

d) 30

View Answer

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 is the output of the below program?

```
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

View Answer

Answer: a
Explanation: The code shown above is used to convert a temperature in degree celsius to fahrenheit.

9. What is the output of the below program?

```
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

View Answer

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 is the output of the below program?

```
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

View Answer

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.

Python Questions and Answers – Function – 3

This set of Python Questions for campus interview focuses on “Functions”.

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

View Answer

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 is the output of this program?

```
y = 6
z = lambda x: x * y
print z(8)
```

a) 48

b) 14

c) 64

d) None of the mentioned

View Answer

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 is the output of below program?

```
lamb = lambda x: x ** 3
print(lamb(5))
```

a) 15

b) 555

c) 125

d) None of the mentioned

View Answer

Answer: c
Explanation: None.

4. Does Lambda contains return statements?

a) True

b) False

View Answer

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

View Answer

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

View Answer

Answer: b
Explanation: None.

7. What is the output of below program?

```
def f(x, y, z): return x + y + z
f(2, 30, 400)
```

a) 432

b) 24000

c) 430

d) No output

View Answer

Answer: a
Explanation: None.

8. What is the output of below program?

```
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

View Answer

Answer: b
Explanation: None.

9. What is the output of this program?

```
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

View Answer

Answer: c
Explanation: None.

10. What is the output of this program?

```
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

View Answer

Answer: c
Explanation: None.

Python Questions and Answers – Function – 4

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Function – 4”.

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

View Answer

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

View Answer

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 is the output of the following 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

View Answer

Answer: c
Explanation: Any change made in to an immutable data type in a function isn’t reflected outside the function.

4. What is the output of the following piece of 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

View Answer

Answer: b
Explanation: Since a list is mutable, any change made in the list in the function is reflected outside the function.

5. What is the output of the following 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

View Answer

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 is the output of the following 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

View Answer

Answer: d
Explanation: The values given during function call is taken into consideration, that is, i=2 and j=1.

7. What is the output of the following code?

```
def change(one, *two):
   print(type(two))
change(1,2,3,4)
```

a) Integer

b) Tuple

c) Dictionary

d) An exception is thrown

View Answer

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

View Answer

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 is the output of the following 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

View Answer

Answer: a
Explanation: The loop runs three times and ‘z’ is printed each time.

10. What is the output of the following piece of 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

View Answer

Answer: c
Explanation: b combines the remaining parameters into a dictionary.

Python Questions and Answers – Argument Parsing 1

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

View Answer

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

View Answer

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 is the output of the following code?

```
def foo(k):
    k[0] = 1
q = [0]
foo(q)
print(q)
```

a) [0].

b) [1].

c) [1, 0].

d) [0, 1].

View Answer

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

View Answer

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

View Answer

Answer: c
Explanation: zero keyword arguments may be passed if all the arguments have default values.

6. What is the output of the following 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

View Answer

Answer: a
Explanation: It is possible to pass function names as arguments to other functions.

7. What is the output of the following code?

```
def foo():
    return total + 1
total = 0
print(foo())
```

a) 0

b) 1

c) error

d) none of the mentioned

View Answer

Answer: b
Explanation: It is possible to read the value of a global variable directly.

8. What is the output of the following code?

```
def foo():
    total += 1
    return total
total = 0
print(foo())
```

a) 0

b) 1

c) error

d) none of the mentioned

View Answer

Answer: c
Explanation: It is not possible to change the value of a global variable without explicitly specifying it.

9. What is the output of the following 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

View Answer

Answer: b
Explanation: A new object is created in the function.

10. What is the output of the following 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].

View Answer

Answer: b
Explanation: When a list is a default value, the same list will be reused.

Python Questions and Answers – Argument Parsing 2

This set of Tricky Python Questions & Answers focuses on “Argument Parsing”.

1. What is the output of the following code?

```
def foo(k):
    k = [1]
q = [0]
foo(q)
print(q)
```

a) [0].

b) [1].

c) [1, 0].

d) [0, 1].

View Answer

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

View Answer

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

View Answer

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

View Answer

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

View Answer

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 back-ticks (“)

c) identifier followed by the default value within square brackets ([])

d) identifier

View Answer

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 back-ticks (“)

c) identifier followed by the default value within square brackets ([])

d) identifier

View Answer

Answer: d
Explanation: Refer documentation.

8. What is the output of the following 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

View Answer

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

View Answer

Answer: a
Explanation: Refer documentation.

10. What is the output of the following?

```
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]]].

View Answer

Answer: c
Explanation: append() returns None.

Python Questions and Answers – Global vs Local Variables – 1

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Global vs Local Variables – 1”.

1. The output of the code shown below is:

```
def f1():
    x=15
    print(x)
x=12
f1()
```

a) Error

b) 12

c) 15

d) 1512

View Answer

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 is the output of the code shown below?

```
def f1():
    x=100
    print(x)
x=+1
f1()
```

a) Error

b) 100

c) 101

d) 99

View Answer

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 is the output of the code shown below?

```
def san(x):
    print(x+1)
x=-2
x=4
san(12)
```

a) 13

b) 10

c) 2

d) 5

View Answer

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 is the output of the code shown?

```
def f1():
    global x
    x+=1
    print(x)
x=12
print("x")
```

a) Error

b) 13

c) 13

x

d) x

View Answer

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 is the output of the code shown below?

```
def f1(x):
    global x
    x+=1
    print(x)
f1(15)
print("hello")
```

a) error

b) hello

c) 16

d) 16

hello

View Answer

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 is the output of the following 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

View Answer

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 is the output of the code shown?

```
def f():
    global a
    print(a)
    a = "hello"
    print(a)
a = "world"
f()
print(a)
```

a) hello

hello

world

b) world

world

hello

c) hello

world

world

d) world

hello

world

View Answer

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
world
hello

8. What is the output of the code shown below?

```
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]

View Answer

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 is the output of the code shown below?

```
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

View Answer

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 is the output of the code shown below?

```
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

View Answer

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. The output of code shown below is:

```
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

View Answer

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 is the output of the code shown below?

```
x=100
def f1():
    global x
    x=90
def f2():
    global x
    x=80
print(x)
```

a) 100

b) 90

c) 80

d) Error

View Answer

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 code shown below 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

View Answer

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.