deeplearning.ai

Machine Learning Engineers Test - (1h45)

22m
to test end

☆ Python coding

What's the output of this code?

```
def op(x):
   if (x % 3 == 1):
       return x % 3
   else:
       return x + op(x-1)

print(op(316))
```

Pick one of the choices

- 1
- 316
- 317
- 318
- It will output an error.
- None of the above.

Clear selection

☆ Python coding: Overlapping elements

You were asked to write a function that takes as input two lists of integers, and returns the list of only the integers that are common between the two lists (without the duplicates.)

You came up with the following code:

```
def overlapping_elements(list_a, list_b):
   out = [i for i in list_a if i in list_b]
   return out
```

What can we say about this solution?

Pick one of the choices

- The code will output an error.
- The code will not always output the correct output list.
- The code will work, and perform the goal for all input lists.
- The code will work properly only if the two lists are none empty.

Clear selection

☆ Python Coding: Class Inheritance

Consider the following code:

```
class One(object):
    def method1(self):
        print("Method1, class One")
    def method2(self):
        print("Method2, class One")
    def method3(self):
        raise Exception("Empty method")

class Two(One):
    def method1(self):
        super(Two, self).method1()
        print("Method 1, class Two")

class Three(One):
    def method1(self):
        super(Three, self).method1()
        print("Method 1, class Three")
    def method2(self):
        super(Three, self).method2()
        print("Method2, class Three")

class Fourth(Three, Two):
    def method1(self):
        super(Fourth, self).method1()
        print("Method 1, class Fourth")
    def method2(self):
        super(Fourth, self).method2()
        print("Method 2, class Fourth")
```

```
### Object definition ###

firstObject = One()
secondObject = Two()
thirdObject = Three()
fourthObject = Fourth()

fourthObject.method2()
```

What will the code output?

Pick one of the choices

- Output: "Method 2, class One" "Method 2, class Two" "Method 2, class Three" "Method 2, class Fourth"
- Output: "Method 2, class One" "Method 2, class Three" "Method 2, class Fourth"
- Output: "Method 2, class One" "Method 2, class Three" "Method 2, class Two" "Method 2, class Fourth"
- Output: "Method 2, class Two" "Method 2, class Three" "Method 2, class Fourth"

Clear selection

☆ Python coding: Function composition

What is the value of out?

```
def f(x):
    x /= 2
    return x + 3

def g(x, y):
    x *= y + 2
    return x

def h(x, y):
    return x*y

out = g(f(4), h(3,2)))
```

Pick one of the choices

- 20
- 40
- 24
- 14
- 10
- None of the above.

Clear selection

☆ Python coding: Reverse Integer

Fill-in the missing line in the code below. (You can assume that there's no overflow.)

```
def reverse_integer(num):
"""This function takes as input an integer number, and returns an
integer number with all the numbers in the reversed order."""
   rev = 0
   while (num != 0):
      fig = num % 10
      num = num // 10
      ### MISSING LINE ###
   return rev
```

Pick one of the choices

- rev = 10 * fig + rev
- rev = 10 * rev + fig
- rev = 10 * fig + num
- rev = 10 * num + rev
- None of the above.

Clear selection

☆ Python coding: Target Sum

Given a list of integers, you want to return the two elements of the list that add up to a specific value named “target”. Assume that, in the input list, there is only a single pair of elements that satisfy this property. Plus, the two elements should be distinct (e.g. they can have the same value but have a distinct index in the list.) Here’s a starter code:

```
def target_sum(input_list, target):
     length = len(input_list)

     if length <= 1:
               raise Exception(“There’s no solution”)

     hash_map = dict()

     ### START CODE HERE ###



     ### END CODE HERE ###
```

Select one of the following code snippets to fill the gap in the function:

```
### a)

for i in range(length):
       if input_list[i] in hash_map:
                 return [hash_map[input_list[i]], input_list[i]]
       else:
             hashMap[target – input_list[i]] = input_list[i]


### b)

for i in range(length):
       if input_list[i] in hash_map:
                     return [hash_map[input_list[i]], i]
       else:
             hash_map[target – input_list[i]] = i

### c)

for i in range(length):
      if i in hash_map:
                 return [hash_map[input_list[i]], i]
      else:
             hash_map[target – input_list[i]] = i
```

```
### d)

for i in range(length):
      if i in hash_map:
             return [hash_map[input_list[i]], input_list[i]]
      else:
             hash_map[target – input_list[i]] = input_list[i]
```

Pick one of the choices

- a)
- b)
- c)
- d)
- None of the above.

Clear selection

☆ Python coding: Divisors

The divisors of a number are all the numbers which divide evenly this number into another number. For instance, the divisors of 12 are 1, 2, 3, 4 and 12 because 12/1 = 12, 12/2 = 6, 12/3 = 4, 12/4 = 3, 12/12 = 1. 5 is not a divisor of 12 because 12/5 = 2.4 which is not an integer.

In order for the following code function to work, what should be the missing line of code? (Check all that apply.)

```
def divisors(num):
"""This function takes as input an integer, and returns the list of
positive divisors of this integer."""
     list_of_div = []
     ### MISSING LINE ###
        if num % i == 0:
            list_of_div.append(i)
     return list_of_div
```

Pick the correct choices

- [ ] for i in range(0, num):
- [ ] for i in range(0, num + 1):
- [ ] for i in range(1, num):
- [ ] for i in range(1, num + 1):
- [ ] None of the above.

Clear selection

☆ Python coding: Reserved Keyword "continue"

Let's imagine you have the following loop:

```
a = [1,3,5,6,8,9,10,1,2,4,5,6,1,3,5,29,391,192,21]
list1 = []
list2 = []

for i, val in enumerate(a):
    if i % 2 == 0 and type(val) == float:
          list1.append(val)
          continue
    if i % 3 == 0 and type(val) == float:
          list2.append(val)
```

In the context of the code above, which of the following is true about the reserved keyword "continue"?

Pick one of the choices

- ○ It speeds up the code.
- ○ It forces the execution of the next "elif" statement.
- ○ It is equivalent to using the keyword "pass".
- ○ None of the above.

Clear selection

☆ Python coding: Find First repetition

You were asked to implement a function that returns the value of the first repeated item in a list. You came up with the following code:

```
def first_repetition(input_list):

    temp_list = []

    for i, val in enumerate(input_list):
        if val not in temp_list :
            temp_list.append(val)
            if i+1 == len(input_list):
                return -1
            continue
        elif val in temp_list :
            return val
```

Which of the following changes would make the code run faster without undermining the output?

Pick one of the choices

- Removing enumerate(), because it is not necessary for the code to work.
- Using a set instead of a list for temp_list
- Using an array instead of a list for temp_list
- Removing the second "if" statement, because it is redundant.
- None of the above.

Clear selection

☆ Python coding: String Preprocessing Function

Let "text" be a string of characters. What does the following code function do?

```
def function(text):
     return text == text[::-1]
```

Pick one of the choices

- It returns True if the text stays the same when shifted one character on the right.
- It returns True if the text stays the same when shifted one character on the left.
- It returns True if the first and the last character of the text match.
- It returns True if the text reads the same forwards and backwards.
- This code doesn't make sense.

Clear selection

☆ Python coding: Draw a Board Game

You want to draw a nice 4x4 game board, fill-in the missing line.

```
A = '———'.join([' ', ' ', ' ', ' ', ' '])
### MISSING LINE ###
print('\n'.join((A,B,A,B,A,B,A,B,A)))
```

Pick one of the choices

- B = ' '.join(['|', '|', '|', '|'])
- B = ' '.join(['|', '|', '|', '|', '|'])
- B = '██'.join(['–', '–', '–', '–'])
- B = '█'.join(['–', '–', '–', '–'])
- There's no need to add a line of code.
- None of the above.

Clear selection

☆ Python coding: Draw a Board Game I

You'd like to write a code that takes as input an integer "n", and prints an "n" by "n" game board as follows:

```
Input: n = 4

Output:

*** *** *** ***
|   |   |   |   |
*** *** *** ***
|   |   |   |   |
*** *** *** ***
|   |   |   |   |
*** *** *** ***
|   |   |   |   |
*** *** *** ***
```

Additional information:

- Two "|" are separated by 5 spaces.
- Two "***" are separated by 1 space.
- There is no space characters out of the edges of the board.

Fill-in the missing line in the code below.

```
def draw_game_board(n):
    for i in range(n):
        print("*** " * (n-1) + "*")
        ### MISSING LINE OF CODE ###
    print("*** " * (n-1) + "*")
```

Pick one of the choices

- print("| " * (n+1))
- print("| " * n + "|")
- print("||" * n)
- print("||" * (n+1))
- There's no need to add a line of code.
- None of the above.

Clear selection

☆ Python Coding: Binary Trees and Classes

Consider the following code:

```
class NodeTree(object):

        def __init__(self, name):
                self.name_node = name
                self.child_nodes = []

        def __repr__(self):
                return “<Node ‘{}’>”.format(self.name_node)

        def append(self, *args, **kwargs):
                self.child_nodes.append(*args, **kwargs)

        def printWrapper(self):

                def gen(node):
                        list_nodes = [node]
                        while list_nodes:
                                next_node = list_nodes.pop(0)  #pops
first childNode

list_nodes.extend(next_node.child_nodes)
                                yield next_node

        for node in gen(self):
                        print(node)


### Tree definition using the NodeTree class ###

root_node = NodeTree("root_node")
node1 = NodeTree("node1")
node2 = NodeTree("node2")
node3 = NodeTree("node3")
node4 = NodeTree("node4")
node5 = NodeTree("node5")
node6 = NodeTree("node6")


root_node.append(node1)
root_node.append(node2)
root_node.append(node3)
node1.append(node4)
node2.append(node5)
node3.append(node6)
```

What will the code output? And, how is the function **printWrapper** traversing the tree?

Pick one of the choices

- Output: <Node 'root_node'> <Node 'child1'> <Node 'child2'> <Node 'child3'> <Node 'child4'> <Node 'child5'> <Node 'child6'> printWrapper traverses the tree in a depth-first search (DFS) manner
- Output: <Node 'root_node'> <Node 'child1'> <Node 'child4'> <Node 'child2'> <Node 'child5'> <Node 'child3'> <Node 'child6'> printWrapper traverses the tree in a depth-first search (DFS) manner
- Output: <Node 'root_node'> <Node 'child1'> <Node 'child2'> <Node 'child3'> <Node 'child4'> <Node 'child5'> <Node 'child6'> printWrapper traverses the tree in a breath-first search (BFS) manner
- Output: <Node 'root_node'> <Node 'child1'> <Node 'child4'> <Node 'child2'> <Node 'child5'> <Node 'child3'> <Node 'child6'> printWrapper traverses the tree in a breath-first search (BFS) manner

Clear selection

☆

Python Coding: Answering the Zoo Manager's Questions

A zoo gives you a labelled images dataset used for a classification task. The zoo manager often asks you the questions: "In the entire dataset, how many pictures contain a x?" (x could be any animal.)

In order to be ready at all time to answer the manager's question, you decide to write a code to count the presence of each animal in the dataset. What data structure seems the most appropriate to use?

Pick one of the choices

- A set
- A list
- A map
- A LinkedList
- A queue
- A stack

Clear selection

☆ Python coding: Process List of Integers

Let "numbers" be a list of integers. What does the following code function do?

```
def function(numbers):
        return [i for i in numbers if i % 3 == 0]
```

Pick one of the choices

- It returns a list equal to numbers, but without elements that are divisible by 3.
- It returns a list of only the elements in numbers that are divisible by 3.
- It returns a list made of one every 3 elements in numbers.
- It returns a list equal to numbers, minus 1 element every 3 element.
- This code doesn't make sense.

Clear selection

☆ Python coding: Data Labelling Script for the Zoo

After collecting images for a classification task, a zoo has labeled the images by listing the animal present in image in a text file "labels.txt". Line "k" indicates the label of the kth image.

There is one animal per line.

```
Iguana
Giraffe
Lion
Iguana
Lion
Turtle
...
```

What will be printed by this code?

```
animals = {}
with open('labels.txt') as f:
   line = f.readline()
   while line:
        if line in animals:
             animals[line] = animals[line] + 1
       else:
             animals[line] = 1
       line = f.readline()

print(animals)
```

Pick one of the choices

- Only the list of animals present in the zoo images.
- The list of animals present in the zoo images, and how many times they appeared.
- The total number of animals present in images.
- The number of distinct animals in the zoo.
- An error.

Clear selection

Continue