

Python Programming Exam Questions

Time: 1 hr.

1. Given below are a list of positive and negative words. Also a list of comments is provided. Segregate the comments based on positive and negative sentiments.

```
positive = ['good','awesome', 'best', 'nice']
```

```
negative = ['worst','awful']
```

```
comments = ['He is a good boy',  
            'Food is the worst here',  
            'He is an awesome player',  
            'She is the best',  
            'This pizza tastes awful',  
            'These burger are really nice']
```

2. Create a dictionary containing three lambda functions square, cube and square root.

E.g. dict = {'Square': function for squaring, 'Cube': function for cube, 'Squareroot': function for square root}

Pass the values (input from the user) to the functions in the dictionary respectively. Then add the outputs of each function and print it.

3. Find the fruits that are *sour* in taste from the tuple given below.

```
fruits = (('Lemon','sour'),  
          ('DragonFruit', 'Sweet'),  
          ('Grapes','soUr'),  
          ('Kiwi','Sour'),  
          ('Apples','sweet'),  
          ('Orange','sour'),  
          ('Blueberries','sweet'),  
          ('Limes','Sour')  
)
```

Output:

Sour Fruits: ['Lemon', 'Grapes', 'Kiwi', 'Orange', 'Limes']

4. A list of words is given. Find the words from the list that have their second character in uppercase.

ls = ['hello', 'Dear', 'hOw', 'ARe', 'You']

Output: ['hOw', 'ARe']

5. A dictionary of names and their weights on earth is given. Find how much they will weigh on the moon. (Use map and lambda functions)

Formula : $w_{\text{Moon}} = (w_{\text{Earth}} * G_{\text{Moon}}) / G_{\text{Earth}}$

Weight of people in kg

WeightOnEarth = {'John':45, 'Shelly':65, 'Marry':35}

Gravitational force on the Moon: 1.622 m/s²

G_{Moon} = 1.622

Gravitational force on the Earth: 9.81 m/s²

G_{Earth} = 9.81

Output:

Weight on Moon: {'John': 7.44, 'Shelly': 10.75, 'Marry': 5.79}

6. Write a program to fetch the words from the given list which have their first character in uppercase.

namesList = ['santa Maria', 'Hello World', 'Merry christmas', 'tHank You']

Output: ['Maria', 'Hello', 'World', 'Merry', 'You']

7. A list containing multiple lists is given. Convert each inner list into sets and find the intersection of all the sets.

Use reduce function.

given_sets = [[1, 2, 3, 4, 8], [2, 3, 8, 5, 6], [8, 4, 5, 3, 7], [6, 9, 8, 3], [9, 12, 3, 7, 6, 8, 4, 6, 21, 1, 6]]

Output: {8, 3}

8. Find the cumulative average of the list [9,8,7,6,5] using accumulate() and lambda function.

Input list: [9, 5, 7, 8, 5]

9. A list of words is given. Convert the words into uppercase. Use lambda and map functions.

lsbool = ['True', 'FALse', 'tRUe', 'tRue', 'False', 'faLse']

Output: ['TRUE', 'FALSE', 'TRUE', 'TRUE', 'FALSE', 'FALSE']

10. A list of dates (dd-mm-yyyy) in the form of string is given below. Create a new list that stores years i.e. the 'yyyy' part from the dates in the given list.

datesList = ['17-12-1997', '22-04-2011', '01-05-1993', '19-06-2020']

Output: ['1997', '2011', '1993', '2020']