course_3_assessment_2

April 27, 2021

0.1 Q1) Write code to assign to the variable map_testing all the elements in lst_check while adding the string "Fruit:" to the beginning of each element using mapping.

[1]:
```
lst_check = ['plums', 'watermelon', 'kiwi', 'strawberries', 'blueberries',␣
 ↪'peaches', 'apples', 'mangos', 'papaya']

map_testing = map(lambda item : "Fruit: " + item , lst_check)
```

0.2 Q2) Below, we have provided a list of strings called countries. Use filter to produce a list called b_countries that only contains the strings from countries that begin with B.

[2]:
```
countries = ['Canada', 'Mexico', 'Brazil', 'Chile', 'Denmark', 'Botswana',␣
 ↪'Spain', 'Britain', 'Portugal', 'Russia', 'Thailand', 'Bangladesh',␣
 ↪'Nigeria', 'Argentina', 'Belarus', 'Laos', 'Australia', 'Panama', 'Egypt',␣
 ↪'Morocco', 'Switzerland', 'Belgium']

b_countries = filter(lambda item : item[0] == 'B' , countries )
```

0.3 Q3) Below, we have provided a list of tuples that contain the names of Game of Thrones characters. Using list comprehension, create a list of strings called first_names that contains only the first names of everyone in the original list.

[4]:
```
people = [('Snow', 'Jon'), ('Lannister', 'Cersei'), ('Stark', 'Arya'),␣
 ↪('Stark', 'Robb'), ('Lannister', 'Jamie'), ('Targaryen', 'Daenerys'),␣
 ↪('Stark', 'Sansa'), ('Tyrell', 'Margaery'), ('Stark', 'Eddard'),␣
 ↪('Lannister', 'Tyrion'), ('Baratheon', 'Joffrey'), ('Bolton', 'Ramsey'),␣
 ↪('Baelish', 'Peter')]

first_names = [  tup[1] for tup in people ]
```

0.4 Q4) Use list comprehension to create a list called lst2 that doubles each element in the list, lst.

[5]:

```
lst = [["hi", "bye"], "hello", "goodbye", [9, 2], 4]

lst2 = [ item * 2 for item in lst]
```

0.5 Q5) Below, we have provided a list of tuples that contain students' names and their final grades in PYTHON 101. Using list comprehension, create a new list passed that contains the names of students who passed the class (had a final grade of 70 or greater).

[9]:

```
l1 = ['left', 'up', 'front']
l2 = ['right', 'down', 'back']

l3 = list(zip(l1,l2))
opposites  = filter(lambda tup : len(tup[0]) > 3 and len(tup[1]) > 3 ,l3)
```

0.6 Q6) Write code using zip and filter so that these lists (l1 and l2) are combined into one big list and assigned to the variable opposites if they are both longer than 3 characters each.

[10]:

```
l1 = ['left', 'up', 'front']
l2 = ['right', 'down', 'back']

opposites = [(x1,x2) for (x1,x2) in list(zip(l1,l2)) if len(x1) > 3 and len(x2)␣
 ↪> 3 ]
```

0.7 Q7) Below, we have provided a species list and a population list. Use zip to combine these lists into one list of tuples called pop_info. From this list, create a new list called endangered that contains the names of species whose populations are below 2500.

[11]:

```
species = ['golden retriever', 'white tailed deer', 'black rhino', 'brown␣
 ↪squirrel', 'field mouse', 'orangutan', 'sumatran elephant', 'rainbow trout',␣
 ↪'black bear', 'blue whale', 'water moccasin', 'giant panda', 'green turtle',␣
 ↪'blue jay', 'japanese beetle']

population = [10000, 90000, 1000, 2000000, 500000, 500, 1200, 8000, 12000,␣
 ↪2300, 7500, 100, 1800, 9500, 125000]

pop_info  = list(zip(species,population))
print(pop_info)

endangered = [ tup[0] for tup in pop_info if tup[1] < 2500 ]
```

```
[('golden retriever', 10000), ('white tailed deer', 90000), ('black rhino',
1000), ('brown squirrel', 2000000), ('field mouse', 500000), ('orangutan', 500),
('sumatran elephant', 1200), ('rainbow trout', 8000), ('black bear', 12000),
('blue whale', 2300), ('water moccasin', 7500), ('giant panda', 100), ('green
turtle', 1800), ('blue jay', 9500), ('japanese beetle', 125000)]
```

[ ]:

[ ]:

[ ]: