Forward School

Program Code: J620-002-4:2020

Program Name: FRONT-END SOFTWARE DEVELOPMENT

Title : Exercise 3 - List Comprehension & Lambda

Name: Chuay Xiang Ze

IC Number: 021224070255

Date : 22/06/2023

Introduction : Learning how to implement list comprehension and lambda

Conclusion : Managed to solve problems with codes learnt.

EXERCISE 3

List Comprehension & Lambda Exercise

In [18]:

```
# write list comprehension to determine the length of each word
# except 'the' and store as 'word_lengths'
sentence = "the quick brown fox jumps over the lazy dog"

word_lengths = [len(word) for word in sentence.split() if word != "the"]
print(word_lengths)
```

```
[5, 5, 3, 5, 4, 4, 3]
```

In [27]:

```
# write a list comprehension to extract the
# negative numbers from the list as integers and store as newlist
numbers = [34.6, -203.4, 44.9, -68.3, -12.2, 44.6, 12.7]
newlist = [round(n) for n in numbers if n < 0]
print(newlist)
```

```
[-203, -68, -12]
```

In [30]:

```
# Convert the following code to list comprehension

coords = []
for x in range(4):
    for y in range(2):
        coordinate = (x, y)
        coords.append(coordinate)
print(coords)

newCoords = [(x, y) for x in range(4) for y in range(2)]
print(newCoords)
```

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

In [41]:

```
# write a list comprehension to list all the combinations
# for the two sets of words

set1 = ['ball','cheese','round']
set2 = ['cake','rice','ham']

fullList = [(set1, set2) for i in range(len(set1)) for i in range(len(set2))]
print (fullList)
print (len(fullList))
```

```
[(['ball', 'cheese', 'round'], ['cake', 'rice', 'ham']), (['ball', 'chees
e', 'round'], ['cake', 'rice', 'ham']), (['ball', 'cheese', 'round'], ['ca
ke', 'rice', 'ham']), (['ball', 'cheese', 'round'], ['cake', 'rice', 'ha
m']), (['ball', 'cheese', 'round'], ['cake', 'rice', 'ham']), (['ball', 'c
heese', 'round'], ['cake', 'rice', 'ham']), (['ball', 'cheese', 'round'],
['cake', 'rice', 'ham']), (['ball', 'cheese', 'round'], ['cake', 'rice',
'ham']), (['ball', 'cheese', 'round'], ['cake', 'rice', 'ham'])]
9
```

In [66]:

```
# write a lambda function that squares the number
# for all odd numbers from 1 to 100
x = range(1,101)

oddSquared = [i*i for i in filter(lambda num: num%2 != 0, x)]
print(list(oddSquared))
```

```
[1, 9, 25, 49, 81, 121, 169, 225, 289, 361, 441, 529, 625, 729, 841, 961,
1089, 1225, 1369, 1521, 1681, 1849, 2025, 2209, 2401, 2601, 2809, 3025, 32
49, 3481, 3721, 3969, 4225, 4489, 4761, 5041, 5329, 5625, 5929, 6241, 656
1, 6889, 7225, 7569, 7921, 8281, 8649, 9025, 9409, 9801]
```

In [67]:

```
# write a list comprehension that squares number
# for all odd numbers from 1 to 100
x = range(1,101)

oddSquared = [i*i for i in x if i%2 != 0]
print(list(oddSquared))
```

```
[1, 9, 25, 49, 81, 121, 169, 225, 289, 361, 441, 529, 625, 729, 841, 961,
1089, 1225, 1369, 1521, 1681, 1849, 2025, 2209, 2401, 2601, 2809, 3025, 32
49, 3481, 3721, 3969, 4225, 4489, 4761, 5041, 5329, 5625, 5929, 6241, 656
1, 6889, 7225, 7569, 7921, 8281, 8649, 9025, 9409, 9801]
```

In [72]:

```
# write a lambda function to extract names that begin with 'A'
names = ['Anne', 'Amy', 'Bob', 'David', 'Carrie', 'Barbara', 'Zach']

aName = filter(lambda name: name[0]=="A", names)
print(list(aName))
```

```
['Anne', 'Amy']
```

In [65]:

```
# write a list comprehension to extract names that begin with 'B'
names = ['Anne', 'Amy', 'Bob', 'David', 'Carrie', 'Barbara', 'Zach']

bName = [n for n in names if n[0] == 'B']
print(list(bName))
```

```
['Bob', 'Barbara']
```

In [ ]:

```

```