Unit 14

Loops & List Comprehensions

Asg 14.5 (Coding)

In [1]:
```
# set up notebook to display multiple output in one cell

from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

print('The notebook is set up to display multiple output in one cell.')
```

```
The notebook is set up to display multiple output in one cell.
```

LIST COMPREHENSIONS - In Python, an easy way to create a list from a sequence of values based on some selection criteriai is to use a **list comprehension**. - List comprehensions are concise ways to create lists. The general syntax is: ! [list%20comprehension.PNG](attachment:list%20comprehension.PNG)
... where the if clause is optional. - The **expression** describes each element of the list that is being built. - The **for clause** iterates through each item in a sequence. - The items are filtered by the **if clause** if there is one. - In the example above, the for statement lets item take on all the values in the list startlist. Each item is then cubed before it is added to the list that is being built. The result is a list of cubes of the values in endlist.

Problem #1: Write code that uses a list comprehension that takes the given list input_list and creates a new list that consists of the cube values of only the odd numbers in input_list.

input_list = [5, 9, 12, 16, 19, 23, 26, 29]

Desired Output
![list%20comp%201.PNG](attachment:list%20comp%201.PNG)

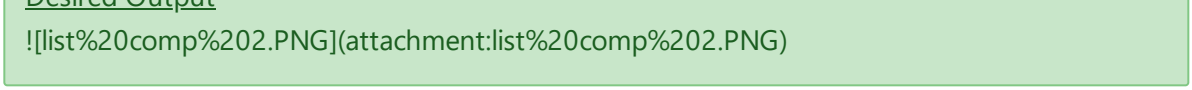
In [2]:
```
input_list = [5, 9, 12, 16, 19, 23, 26, 29]

[num **3 for num in input_list if num % 2 != 0]
```

Out[2]:
```
[125, 729, 6859, 12167, 24389]
```

Problem #2: Write code that uses a list comprehension to create a new list which consists of

the square values of all numbers between 11 and 133 that are multiples of 7..

Desired Output



In [6]:
```
print([x ** 2 for x in range(11, 134) if x % 7 == 0])
```

```
[196, 441, 784, 1225, 1764, 2401, 3136, 3969, 4900, 5929, 7056, 8281, 9604, 11025, 1254
4, 14161, 15876, 17689]
```

Problem #3: Write code that uses a list comprehension to iterate through the string 'School District of Elmbrook' and returns a new list that consists only of the vowels in the string 'School District of Elmbrook'.

Desired Output

In [11]:
```
def isVowel(y):
    x = y.lower()
    if (x == 'a' or x == 'e' or x == 'i' or
        x == 'o' or x == 'u'):
        return True
    else:
        return False

string = 'School District of Elmbrook'
[ch for ch in string if isVowel(ch) ]
```

Out[11]:
```
['o', 'o', 'i', 'i', 'o', 'E', 'o', 'o']
```

Problem #4:
Write code that uses a list comprehension to create a new list that cosists of all positive integers less than 310 that have 3 and 7 as factors

Desired Outsput

In [13]:
```
def isFactorOf(factor, number):
    if number % factor == 0:
        return True
    else:
        return False
[num for num in range(310) if isFactorOf(3, num) and isFactorOf(7, num)]
```

Out[13]:
```
[0, 21, 42, 63, 84, 105, 126, 147, 168, 189, 210, 231, 252, 273, 294]
```

DICTIONARY COMPREHENSIONS Read the article found below:

[10 Examples to Master Python Dictionary Comprehensions]
(https://towardsdatascience.com/10-examples-to-master-python-dictionary-comprehensions-7aaa536f5960)