LIST COMPREHENSION EXERCISE SOLUTIONS

TASK 1: CONVERT FOR LOOP TO LIST COMPREHENSION

WORK WITH

```
simple = []
for i in range(10):
    s = i + 30
    simple.append(s)
simple
```

DESIRED OUT

[30, 31, 32, 33, 34, 35, 36, 37, 38, 39]

HINTS?

1. List comprehension doesn't have any append function or empty list.

In [2]:

```
[i + 30 for i in range(10)]
```

Out[2]:

```
[30, 31, 32, 33, 34, 35, 36, 37, 38, 39]
```

TASK 2: USE FOR LOOP WITH LIST COMPREHENSION

WORK WITH

```
x= 0
case = []
while x < 10:
    s = "The number is " + str(x)
    case.append(s)
    x += 2
case
```

DESIRED OUTPUT

['The number is 0',
'The number is 2',
'The number is 4',

'The number is 6',
'The number is 8']

HINTS?

1. Like task 1, remove the empty list and append function.

In [30]:

```
["The number is " + str(x) for x in range(0, 10, 2)]
```

Out[30]:

```
['The number is 0',
 'The number is 2',
 'The number is 4',
 'The number is 6',
 'The number is 8']
```

TASK 3:CONTROL FLOW WITH MODULUS IN LIST COMPREHENSION

DESIRED OUTPUT

[1728, 2744, 4096, 5832]

HINTS?

1. Mulitply d by **3 or **pow**(d, 3).
2. Use **range**.
3. Use an **if** condition
4. Use modulus where d % 2
5. Use d > 10

In [35]:

```
[d**3 for d in range(20) if d %2 == 0 and d > 10]
```

Out[35]:

```
[1728, 2744, 4096, 5832]
```

TASK 4: CREATE TWO NESTED LIST COMPREHENSIONS INSIDE A DICTIONARY

DESIRED OUTPUT

```
d2 = {'k1': [0, 21, 42, 63, 84],
 'k2': [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]}

d3 = {'k3': [495]}
```

HINTS?

1. Use the built-in **sum()** function twice for **k3** dictionary.
2. For k1, use g + 20 *g with **range** and g < 5
3. For k2, e*e with **range** and e < 10
4. Use an **if** condition.

Remember, solutions are in the PDF resource for section 6!

In [1]:

```
d2 = {"k1": [(g + 20 * g) for g in range(10) if g < 5],
      "k2": [e*e for e in range(20) if e < 10]}

d3 = {"k3": sum(d2["k1"]) + sum(d2["k2"])}
d3
```

Out[1]:

```
{'k3': 495}
```

END OF SECTION