

## More list comprehension examples

```
In [10]: import numpy as np
np.random.seed(10)
data = np.random.randint(0, 10, size=25)
print(data)

[9 4 0 1 9 0 1 8 9 0 8 6 4 3 0 4 6 8 1 8 4 1 3 6 5]
```

```
In [11]: only_even = [i for i in data if (i % 2) == 0]
print(only_even)

[4, 0, 0, 8, 0, 8, 6, 4, 0, 4, 6, 8, 8, 4, 6]
```

```
In [12]: data_sq = [i ** 2 for i in data]
print(data_sq)

[81, 16, 0, 1, 81, 0, 1, 64, 81, 0, 64, 36, 16, 9, 0, 16, 36, 64, 1, 6
4, 16, 1, 9, 36, 25]
```

```
In [13]: combo = [i ** 2 for i in [j * 2 for j in data]]
print(combo)

[324, 64, 0, 4, 324, 0, 4, 256, 324, 0, 256, 144, 64, 36, 0, 64, 144, 2
56, 4, 256, 64, 4, 36, 144, 100]
```

```
In [14]: even_sq = [i ** 2 for i in data if (i % 2) == 0]
print(even_sq)

[16, 0, 0, 64, 0, 64, 36, 16, 0, 16, 36, 64, 64, 16, 36]
```

```
In [15]: def sq(x):
    return x*x
def keepEven(x):
    return x % 2 == 0
out2 = [sq(d) for d in data if keepEven(d)]

print(out2)

[16, 0, 0, 64, 0, 64, 36, 16, 0, 16, 36, 64, 64, 16, 36]
```

```
In [16]: numbers = [1, 2, 3, 4, 5, 6, 18, 20]
squares = [
    "small" if number < 10 else "big"
    for number in numbers
    if (number % 2 == 0 and number % 3 == 0) ]
print(squares)

['small', 'big']
```

```
In [17]: matrix = [  
    [(b+1)*(i+1) for b in range(5)]  
    for i in range(3)  
    ]  
print(matrix)
```

```
[[1, 2, 3, 4, 5], [2, 4, 6, 8, 10], [3, 6, 9, 12, 15]]
```

```
In [18]: matrix = [[1, 2], [3, 4, 5], [6, 7, 8]]  
new_matrix = [i for s in matrix for i in s]  
print(new_matrix)
```

```
[1, 2, 3, 4, 5, 6, 7, 8]
```

```
In [ ]:
```