

List Comprehensions

1. A Pythonic approach to a frequent operation
 - Turns a multi-line for-loop into a one liner.
 - Basic syntax:

```
[transformation iteration filter]    OR  
[expression for item in list if conditional]
```

- Equivalent to:

```
for item in list:  
    if conditional:  
        expression
```

2. Unconditional list comprehensions

```
1  numbers = [1, 2, 3, 4, 5]  
2  
3  #using for loop  
4  doubled_numbers = []  
5  - for n in numbers:  
6      doubled_numbers.append(n * 2)  
7  print(doubled_numbers)  
8  
9  print()  
10 #using list comprehension  
11 doubled_numbers = [n * 2 for n in numbers]  
12 print(doubled_numbers)
```

3. Conditional list comprehensions

```
1 numbers = [1, 2, 3, 4, 5]
2
3 #using for loop
4 doubled_odds = []
5 - for n in numbers:
6 -     if n % 2 == 1:
7         doubled_odds.append(n * 2)
8 print(doubled_odds)
9
10 print()
11 #using list comprehension
12 doubled_odds = [n * 2 for n in numbers if n % 2 == 1]
13 print(doubled_odds)
```

4. Create list of first 10 square numbers

```
1 #create list of first 10 square numbers
2 #using for loop
3 squares = []
4 - for s in range(1, 11):
5     squares.append(s * s)
6 print(squares)
7
8 print()
9 #using list comprehension
10 numbers = list(range(1, 11))
11 squares = [s * s for s in numbers]
12 print(squares)
13
14 print()
15 #using list comprehension with range
16 squares = [s * s for s in range(1, 11)]
17 print(squares)
```

5. Multiply each item in list with 3

```
1 #multiply each item in list with 3
2 list1 = [3, 4, 5]
3 multiplied = [item * 3 for item in list1]
4 print(multiplied)
```

6. Function call in list comprehension

```
1 #function call in list comprehension
2 #double of 0 to 9
3 = def double(x):
4     return x * 2
5
6 doubled = [double(x) for x in range(10)]
7 print(doubled)
```

8. Using list comprehension with string list.

```
1 #extract first letter of each String
2 fname = ["Arthur", "King", "Of", "The", "Britons"]
3 initials = [word[0] for word in fname]
4 print(initials)
5
6 print()
7 #change all the letter to lower case
8 lowered = [x.lower() for x in ["F", "O", "P"] ]
9 print(lowered)
```

The End