

ASSIGNMENT 4

LIST COMPREHENSION ¶

- single line of code to construct powerful functionality.
- It consists of brackets containing the expression, which is executed for each element along with the for loop to iterate over each element.

Advantages of List Comprehension

- More time efficient and space efficient than loops.
- Require fewer lines of code.
- Transforms iterative statement into a formula.

List Comprehension with FOR Loop

```
In [4]: letters = []  
for letter in 'R15 is the best bike in yamaha':  
    letters.append(letter)  
print(letters)
```

executed in 11ms, finished 21:06:45 2021-10-21

```
['R', '1', '5', ' ', 'i', 's', ' ', 't', 'h', 'e', ' ', 'b', 'e', 's', 't', ' ',  
' ', 'b', 'i', 'k', 'e', ' ', ' ', 'i', 'n', ' ', ' ', 'y', 'a', 'm', 'a', 'h', 'a']
```

```
In [5]: letters = [letter for letter in 'R15 is the best bike in yamaha']  
print(letters)
```

executed in 14ms, finished 21:06:46 2021-10-21

```
['R', '1', '5', ' ', 'i', 's', ' ', 't', 'h', 'e', ' ', 'b', 'e', 's', 't', ' ',  
' ', 'b', 'i', 'k', 'e', ' ', ' ', 'i', 'n', ' ', ' ', 'y', 'a', 'm', 'a', 'h', 'a']
```

List Comprehension with for and if loop

```
In [6]: number_list = [ x for x in range(20) if x % 2 == 0 ]  
print(number_list)
```

executed in 5ms, finished 21:07:47 2021-10-21

```
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
```



```
In [7]: num_list = [y for y in range(100) if y % 2 == 0 if y % 5 == 0]  
print(num_list)
```

executed in 11ms, finished 21:08:11 2021-10-21

[0, 10, 20, 30, 40, 50, 60, 70, 80, 90]

In []: