

✓ Повторение: list comprehension

List comprehension: инструмент для генерации списков на основе цикла `for`

```
x = [1, 2, 3, 4, 5]
y = [i for i in x]
print(y, x is y)
```

```
y = [i ** 2 for i in x]
print(y)
```

```
y = [i ** 2 for i in range(10)]
print(y)
```

Добавим условие:

```
y = [i ** 2 for i in range(10) if i % 2 == 0]
print(y)
```

Добавим тернарный оператор:

```
y = [i ** 2 if i % 3 == 0 else i ** 3 for i in range(10)]
print(y)
```

Добавим и тернарный оператор, и условие:

```
y = [i ** 2 if i % 3 == 0 else i ** 3 for i in range(10) if i % 2 == 0]
print(y)
```

Несколько вложенных циклов:

1) без list comprehension:

```
y = []
for j in range(1, 4):
    for i in range(1, 4):
        y.append(i * j)
print(y)
```

2. с list comprehension

```
y = [i * j for i in range(1, 4) for j in range(1, 4)]
print(y)
```

Порядок важен!

```
big_list = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
y = [el ** 2 for sub_list in big_list for el in sub_list]
print(y)
```

```
y = [[el ** 2 for el in sub_list] for sub_list in big_list]
print(y)
```

✓ Задания для выполнения в классе

1. Напишите программу, которая считывает с клавиатуры два целых числа M и N и создаёт список, в котором будут частные от деления на N всех чисел от 0 до M . Если число N равно 0 или число M меньше или равно 0, выведите на экран текст, сообщающий об этом.

2. Дан список слов. Создать новый список, где каждым элементом будет длина слова из первого списка.

```
words = ["apple", "banana", "orange", "pear", "pineapple"]
```

2a*. Создать словарь, в котором ключами будут слова, а значениями – их длины.

3. Дана переменная, содержащая фамилию, и список, содержащий имена. Создать новый список, где каждым элементом будет полное имя (имя и фамилия).

```
lastname = "Smith"  
firstnames = ["John", "Mary", "Peter", "Aerith", "Bob"]
```

4. Из списка полных имён создать новый, в котором будут только те, длина которых не превышает 10 символов.

5. Пусть программа принимает на вход целое число n . Создайте список длины n такого вида: ["a", "aa", "aaa", ...].

6. Дан список строк. Постройте новый список, в котором будут все те же строки, но написанные задом наперёд.

```
words = ["elbairav", "noitareti", "tsil", "nohtyp"]
```

7. Определите сумму всех чисел от 1 до 100, которые делятся на 7.

8. Дан список слов. Назовём серединой слова его часть без первой и последней буквы (гарантируется, что в каждом слове как минимум три буквы). Создать список слов, в середине которых есть последовательность "аб".

```
wordlist = ["абрикос", "крабам", "собака", "краб", "жаба"]
```

9. Дан список слов. Создайте новый список, где будут все слова, где есть как минимум две буквы "а", отсортированные по длине

```
words = [  
    "собака", "пират", "мама", "тарелка", "арба",  
    "стол", "парта", "ананас", "сад"  
]
```

