

Списочные выражения

Создание списков без явного использования циклов и вызова списочного метода `append()`

1. Обычные списковые выражения

Без считывания данных:

```
my_list = [1, 2, 3, 4]
```

Со считыванием данных:

```
my_list = [int(x) for x in input("Введите числа, разделенные пробелами:").split()]  
# Пример ввода: "1 2 3 4", результат: [1, 2, 3, 4]
```

2. Генератор списков

Без считывания данных:

```
squares = [x**2 for x in range(5)]  
# Результат: [0, 1, 4, 9, 16]
```

Со считыванием данных:

```
squares = [int(x)**2 for x in input("Введите числа, разделенные пробелами:").split()]  
# Пример ввода: "1 2 3", результат: [1, 4, 9]
```

3. Генератор списков с условием

Без считывания данных:

```
evens = [x for x in range(10) if x % 2 == 0]  
# Результат: [0, 2, 4, 6, 8]
```

Со считыванием данных:

```
evens = [int(x) for x in input("Введите числа, разделенные пробелами: ").split()
if int(x) % 2 == 0]
# Пример ввода: "1 2 3 4 5 6", результат: [2, 4, 6]
```

4. Генератор списков с несколькими вложенными циклами

Без считывания данных:

```
lists = [[x + y for y in range(3)] for x in range(3)]
# Результат: [[0, 1, 2], [1, 2, 3], [2, 3, 4]]
```

Со считыванием данных:

```
n = int(input("Введите количество строк: "))
m = int(input("Введите количество столбцов: "))
lists = [[i + j for j in range(m)] for i in range(n)]
# Пример ввода: 2 (строки), 3 (столбцы), результат: [[0, 1, 2], [1, 2, 3]]
```

5. Генератор списков с условием и несколькими вложенными циклами

Без считывания данных:

```
distinct_lists = [[x, y] for x in range(3) for y in range(3) if x != y]
# Результат: [[0, 1], [0, 2], [1, 0], [1, 2], [2, 0], [2, 1]]
```

Со считыванием данных:

```
n = int(input("Введите предел диапазона: "))
distinct_lists = [[x, y] for x in range(n) for y in range(n) if x != y]
# Пример ввода: 3, результат: [[0, 1], [0, 2], [1, 0], [1, 2], [2, 0], [2, 1]]
```

6. Генератор списков с использованием функции

Без считывания данных:

```
strings = [str(x) for x in range(5)]
# Результат: ['0', '1', '2', '3', '4']
```

Со считыванием данных:

```
strings = [str(int(x)) for x in input("Введите числа, разделенные пробелами: ").split()]  
# Пример ввода: "1 2 3", результат: ['1', '2', '3']
```

7. Генератор списков с использованием вложенных списков (разворачивание)

Без считывания данных:

```
flattened = [item for sublist in [[1, 2], [3, 4]] for item in sublist]  
# Результат: [1, 2, 3, 4]
```

Со считыванием данных:

```
rows = int(input("Введите количество списков: "))  
flattened = [int(item) for _ in range(rows) for item in input("Введите числа, разделенные пробелами: ").split()]  
# Пример ввода: 2 (списка), затем: "1 2" и "3 4", результат: [1, 2, 3, 4]
```

8. Генератор списков для извлечения данных из строк

Без считывания данных:

```
lengths = [len(s) for s in ['apple', 'banana', 'cherry']]  
# Результат: [5, 6, 6]
```

Со считыванием данных:

```
lengths = [len(s) for s in input("Введите строки, разделенные запятыми: ").split(',')]  
# Пример ввода: "apple,banana,cherry", результат: [5, 6, 6]
```

9. Генератор списков для матрицы

Без считывания данных:

```
matrix = [[x + y for x in range(3)] for y in range(3)]  
# Результат: [[0, 1, 2], [1, 2, 3], [2, 3, 4]]
```

Со считыванием данных:

```
n = int(input("Введите количество строк: "))
m = int(input("Введите количество столбцов: "))
matrix = [[input(f"Введите элемент ({i+1},{j+1}): ") for j in range(m)] for i in range(n)]
# Пример ввода:
# 2 (строки)
# 3 (столбцы)
# 1
# 2
# 3
# 4
# 5
# 6
# Результат: [['1', '2', '3'], ['4', '5', '6']]
```

10. Вложенные списковые выражения (двумерные списки)

Без считывания данных:

```
nested_lists = [[x for x in range(3)] for y in range(3)]
# Результат: [[0, 1, 2], [0, 1, 2], [0, 1, 2]]
```

Со считыванием данных:

```
n = int(input("Введите количество строк: "))
nested_lists = [[int(x) for x in input(f"Введите числа строки {i+1}, разделенные пробелами: ").split()] for i in range(n)]
# Пример ввода:
# 2 (строки)
# 1 2 3
# 4 5 6
# Результат: [[1, 2, 3], [4, 5, 6]]
```