

# Циклы

Пример цикла `for` в Python:

```
In bremen_musicians = ['Трубадур', 'Петух',  
                        'Кот', 'Пёс', 'Осёл']  
print('Представляем музыкантов:')  
for musician in bremen_musicians:  
    print(musician)
```

```
Out Представляем музыкантов:  
Трубадур  
Петух  
Кот  
Пёс  
Осёл
```

Переменная `musician` принимает последовательно значения всех элементов из списка `bremen_musicians`.

Ещё пример. Как пройти по числам подряд:

```
In for i in range(1, 6): # range – диапазон  
    print(i)  
    print('я иду тебя искать')
```

```
Out 1  
2  
3  
4  
5  
я иду тебя искать
```

Внимание: `range(a, b)` возвращает числа от `a` до `b-1`.  
Функция `reversed()` «переворачивает» списки и диапазоны значений:

```
In for i in reversed(range(1, 11)):  
    print(i)  
    print('Поехали!')
```

```
Out 10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
Поехали!
```

# Ветвления

Логические выражения могут принимать логические значения `True` («истина») и `False` («ложь»).

Условный оператор `if` для записи ветвления «если – то»:

```
# в переменной beaufort хранится  
# скорость ветра по шкале Бофорта  
if beaufort == 0:  
    print('штиль')
```

Конструкция `if-else` для записи ветвления «если – то – иначе»:

```
if beaufort == 0:  
    print('штиль')  
else:  
    print('есть ветер')
```

Множественное ветвление:

```
if beaufort == 0:  
    print('штиль')  
elif beaufort == 1:  
    print('тихий ветер')  
elif beaufort == 2:  
    print('лёгкий ветер')  
elif beaufort == 3:  
    print('слабый ветер')  
elif beaufort == 4:  
    print('умеренный ветер')  
elif beaufort == 5:  
    print('свежий ветер')  
elif beaufort == 6:  
    print('сильный ветер')
```

Как только выполняется одно из условий — все нижеследующие `elif` и `else` пропускаются.

# Логические выражения

Операторы сравнения:

- равно `==`
- меньше `<`
- больше `>`
- больше или равно `>=`
- меньше или равно `<=`
- не равно `!=`

Логические операторы:

- `or` («или») — логическое сложение

```
if beaufort == 7 or beaufort == 8:  
    print('крепкий ветер')
```

- `and` («и») — логическое умножение
- `not` («не») — отрицание