



Операторы сравнения

<	Меньше
>	Больше
<=	Меньше или равно
>=	Больше или равно
==	Равенство
!=	Неравенство

Побитовые логические операторы

~x	Побитовое "НЕ"
	Побитовое "ИЛИ"
^	Побитовое "ИСКЛЮЧИТЕЛЬНОЕ ИЛИ"
&	Побитовое "И"

Логические операторы

not	Логическое НЕ
or	Логическое ИЛИ
and	Логическое И
in, not in	Проверка принадлежности
is, is not	Проверка тождественности

Логическое НЕ

x	not x
True	False
False	True

Логическое И

x	y	x and y
False (0)	False (0)	False (0)
False (0)	True (1)	False (0)
True (1)	False (0)	False (0)
True (1)	True (1)	True (1)

Логическое ИЛИ

x	y	x or y
False (0)	False (0)	False (0)
False (0)	True (1)	True (1)
True (1)	False (0)	True (1)
True (1)	True (1)	True (1)

Условный оператор

```
if Условие:  
    Блок инструкций 1  
else:  
    Блок инструкций 2
```

if-elif-else

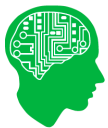
```
if Условие 1:  
    Блок инструкций 1  
elif Условие 2:  
    Блок инструкций 2  
else:  
    Блок инструкций 3
```

Инлайновый if

```
A = 5 if X else -5
```

Цикл for

```
for value in iterator:  
    # Начало блока кода с телом цикла  
    ...  
    ...  
    ...  
    # Конец блока кода с телом цикла  
# Код который будет выполняться после  
цикла
```



Цикл while с предусловием

```
while условие:
    # Начало блока кода с телом цикла
    # пока условие истинно, цикл
    выполняется
    ...
    ...
    ...
    # Конец блока кода с телом цикла
# Код который будет выполняться после
цикла
```

Цикл while с постусловием

```
while True:
    # Начало блока кода с телом цикла
    # пока условие истинно, цикл
    выполняется
    ...
    ...
    ...
    # Конец блока кода с телом цикла
    if условие:
        break
# Код который будет выполняться после
цикла
```

Best practice

```
# Хорошо
a is None

# Плохо
a == None
```

```
# Хорошо
if not seq:
if seq:

# Плохо
if len(seq)
if not len(seq)
```

```
# Хорошо
if greeting:

# Плохо
if greeting == True:

# Совсем не правильно
if greeting is True:
```

```
a <= b <= c

a == b == c
```