

# Составные инструкции: условия

---

ЛЕКЦИЯ 3





# Содержание

---

- ☐ Составные инструкции
- ☐ Условная инструкция if
- ☐ Условные операторы
- ☐ Группировка условий
- ☐ Множественное ветвление
- ☐ Аналог тернарного оператор

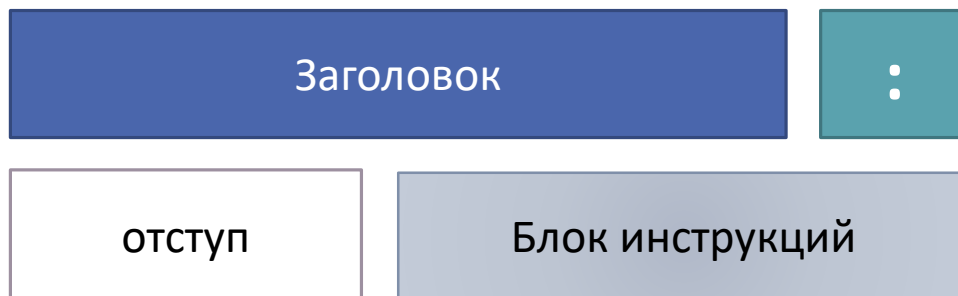


# Составные инструкции

**Составная инструкция** – это инструкция или блок инструкций, встроенных в другую инструкцию

- ❑ В языке Python отсутствуют фигурные скобки или разделители begin/end, окружающие блоки программного кода. Принадлежность инструкций к вложенному блоку определяется по величине отступов
- ❑ Пустые строки, пробелы и комментарии в Языке Python игнорируются и не нарушают блок

Составные инструкции строятся как:



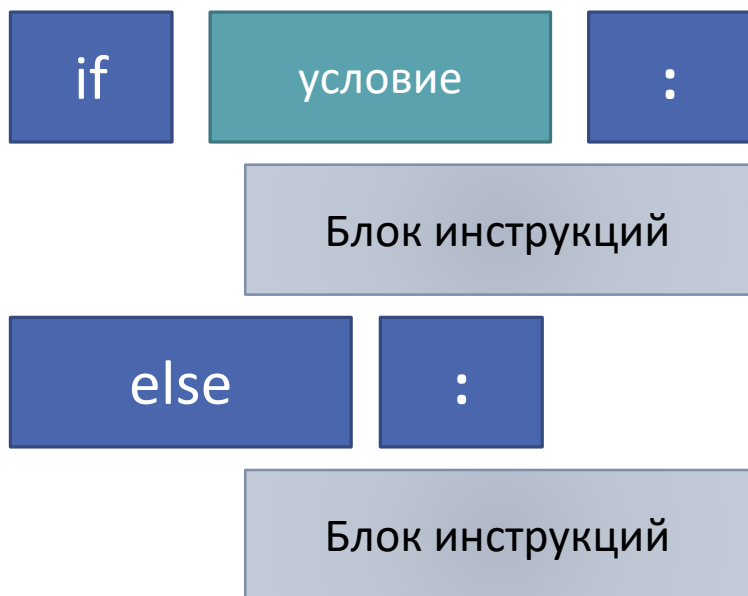
**Пример:**

```
if x > y:  
    print(x)
```



# Условная инструкция if

В языке Python условная инструкция if («если-тогда-иначе») организована следующим образом:



*Примечание:*

*В отличие от языка C++ необходимости в заключении условия в круглые скобки в языке Python нет*

**Пример:**

```
if X > Y:
    print(X)
else:
    print(Y)
```



# Условные операторы

Условные операторы в языке Python аналогичны операторам в языке Си:

==	!=	<	>	<=	>=
Равно?	Не равно?	Меньше?	Больше?	Меньше или равно?	Больше или равно?

Проверка истинности заданного значения отличается от своего аналога в Си:

~~`if !Y:  
 print(X)`~~

**Не верно**

```
if not Y:  
    print(X)
```

**Верно**



# Условные операторы

Результатом вывода выражений от условных операторов в Python с помощью функции `print()` является **True** или **False**

```
print(X > Y)
```

```
True
```

```
Process finished with exit code 0
```

Однако условные операторы можно использовать в составных выражениях. В этом случае их значение вычисляется аналогично языку Си ( 0 или 1)

```
print((X > Y) + 1)
```

```
2
```

```
Process finished with exit code 0
```



# Группировка условий

---

В качестве логического «И» используется ключевое слово **and**

```
if x == 1 and x != 3:  
    print(x)
```

В качестве логического «ИЛИ» используется ключевое слово **or**

```
if x == 1 or x != 3:  
    print(x)
```

Для определения порядка операций можно использовать скобки:

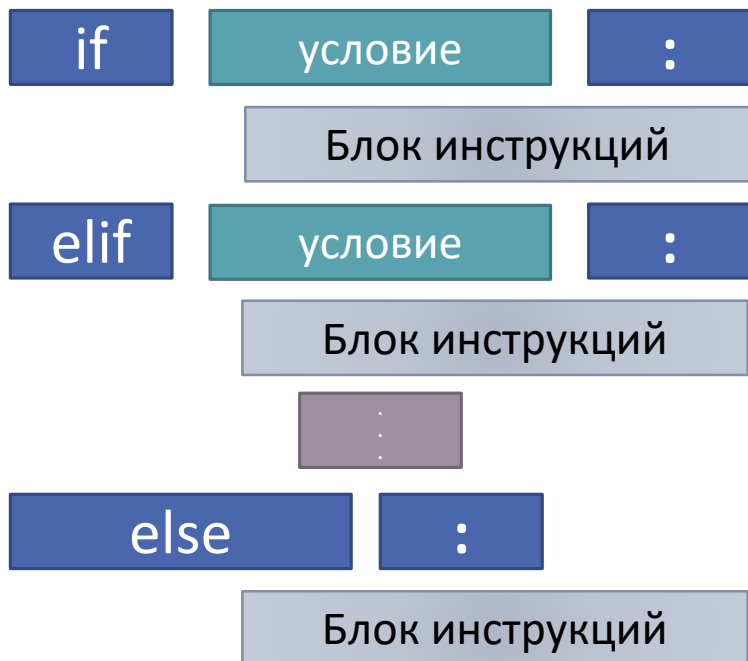
```
if (x == 1 and x != 3) or x > 1:  
    print(x)
```



# Множественное ветвление

В языке Python **нет** инструкции **switch**

Для множественного ветвления в инструкции **if** – **else** следует использовать дополнительные блоки **elif**



```
x = 3

if x == 1:
    print(x)
elif x == 2:
    print(x**2)
elif x == 3:
    print(x*4)
else:
    print(x*2)
```

12

Process finished with exit code 0





# Аналог тернарного оператора

---

Конструкцию вида:

```
if a > b:  
    c = a  
else:  
    c = b
```

Можно привести к упрощенной форме за счет выражения

Результат

if

Условие

else

Результат

```
c = a if a > b else b
```



# Переход на следующую строку в инструкции

---

Выражения в скобках в Python можно продолжать на разных строках

```
c = (2 +  
     3)
```

Кроме того в инструкции можно использовать специальный символ «\» для перехода на новую строку

```
if (X == 1 and  
    X != 3) \  
    or X > 1:  
    print(X)
```