

Содержание

Ветвление	2
Задания по $\text{if} \dots \text{else}$	5
Несколько условий	6
Задания по условиям $(\text{if} \dots \text{else} \dots \text{else})$	7
Оператор <code>elif</code>	8
Задания по <code>elif</code> $\text{if} \dots \text{elif} \dots \text{else}$	9
Вложенные условия	10
Задания на вложенные условия $\text{if} \dots \text{if} \dots$	11
Логический оператор <code>not</code>	12
Задания на оператор <code>not</code> $\text{if} \dots \text{not} \dots$	13



Ветвление

Чтобы программа не просто выполняла команды, а еще и могла думать, надо ли ей выполнять тот или иной кусок кода, используют операторы ветвления

if УСЛОВИЕ:



После двоеточия код
отделяется табуляцией
(кнопка Tab)

Что такое условие? Это сравнение, из которого получается логический вывод (**bool**)

Знак сравнения	Описание	Условие	Результат
>	Больше	5 > 9 7 > 3 19 > 19	False True False
<	Меньше	12 < 4 5 < 9 3 < 3	False True False
==	Равно	5 == 5 13 == 5	True False
!=	Не равно	5 != 5 13 != 5	False True

Если условие после *if* выполнится, тогда код будет выполняться. Например:

```
x = 6
if x > 0:
    print("x больше нуля")
```

```
C:\Users\dima2\AppData\Local\Program
x больше нуля

Process finished with exit code 0
```

Если условие не выполняется, то и код тоже не выполняется

```
x = -8

if x > 0:
    print("x больше нуля")
```

```
C:\Users\dima2\AppData\Local\Progr
Process finished with exit code 0
```

Чтобы предусмотреть оба случая выполнения и невыполнения условия используется **оператор else**

if УСЛОВИЕ:

КОД_ЕСЛИ_ВЫПОЛНЯЕТСЯ_УСЛОВИЕ

После двоеточия код отделяется табуляцией (кнопка Tab)

else:

КОД_ЕСЛИ_НЕ_ВЫПОЛНЯЕТСЯ_УСЛОВИЕ

Пример с else

```
x = -8

if x > 0:
    print("x больше нуля")
else:
    print("x меньше нуля")
```

```
C:\Users\dima2\AppData\Local\Progr  
х меньше нуля
```

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



1. Спроси у пользователя число.

Если оно больше нуля – выведи "Положительное", если меньше нуля – "Отрицательное", если равно нулю – выведи "Ноль"

2. Спроси у пользователя возраст.

Если меньше 18 – выведи "Доступ запрещён", иначе "Добро пожаловать".

3. Спроси у пользователя два числа.

Выведи большее из них.

4. Спроси у пользователя число.

Если оно делится на 2 – выведи "Чётное", иначе "Нечётное"

5. Спроси у пользователя пароль.

Если он совпадает с "qwerty", выведи "Пароль верный", иначе – "Неверный пароль"



Несколько условий

Можно указывать несколько условий с помощью специальных слов:

- **or** (или)
- **and** (и)

if УСЛОВИЕ_1 or УСЛОВИЕ_2:

КОД

Если между условиями стоит **and**, то код **if** будет выполнен ТОЛЬКО ЕСЛИ ОБА УСЛОВИЯ ВЕРНЫЕ!

```
x = 2
y = 3

if x == 2 and y == 3:
    print("Условие выполняется")
else:
    print("Не прокатило")
```

```
C:\Users\dima2\AppData\Local\Programs\Python\Python38-32\python.exe
Условие выполняется

Process finished with exit code 0
```

Если между условиями стоит **or**, то код **if** будет выполнен ЕСЛИ ХОТЬ ЧТО-ТО ИЗ УСЛОВИЙ ВЫПОЛНИТСЯ!

```
x = 2
y = 1

if x == 2 and y == 3:
    print("Условие выполняется")
else:
    print("Не прокатило")
```

```
C:\Users\dima2\AppData\Local\Progr  
Не прокатило(  
  
Process finished with exit code 0
```

Задания по условиям

(^">~<"/^)

1. Спроси у пользователя число

Если оно находится в диапазоне от 1 до 10 включительно
выведи "Число от 1 до 10"

Иначе выведи "Не подходит"

2. Спроси у пользователя имя и пароль

Если имя равно "admin" и пароль равен "1234", выведи
"Доступ разрешён", иначе "Ошибка входа".

3. Спроси у пользователя число.

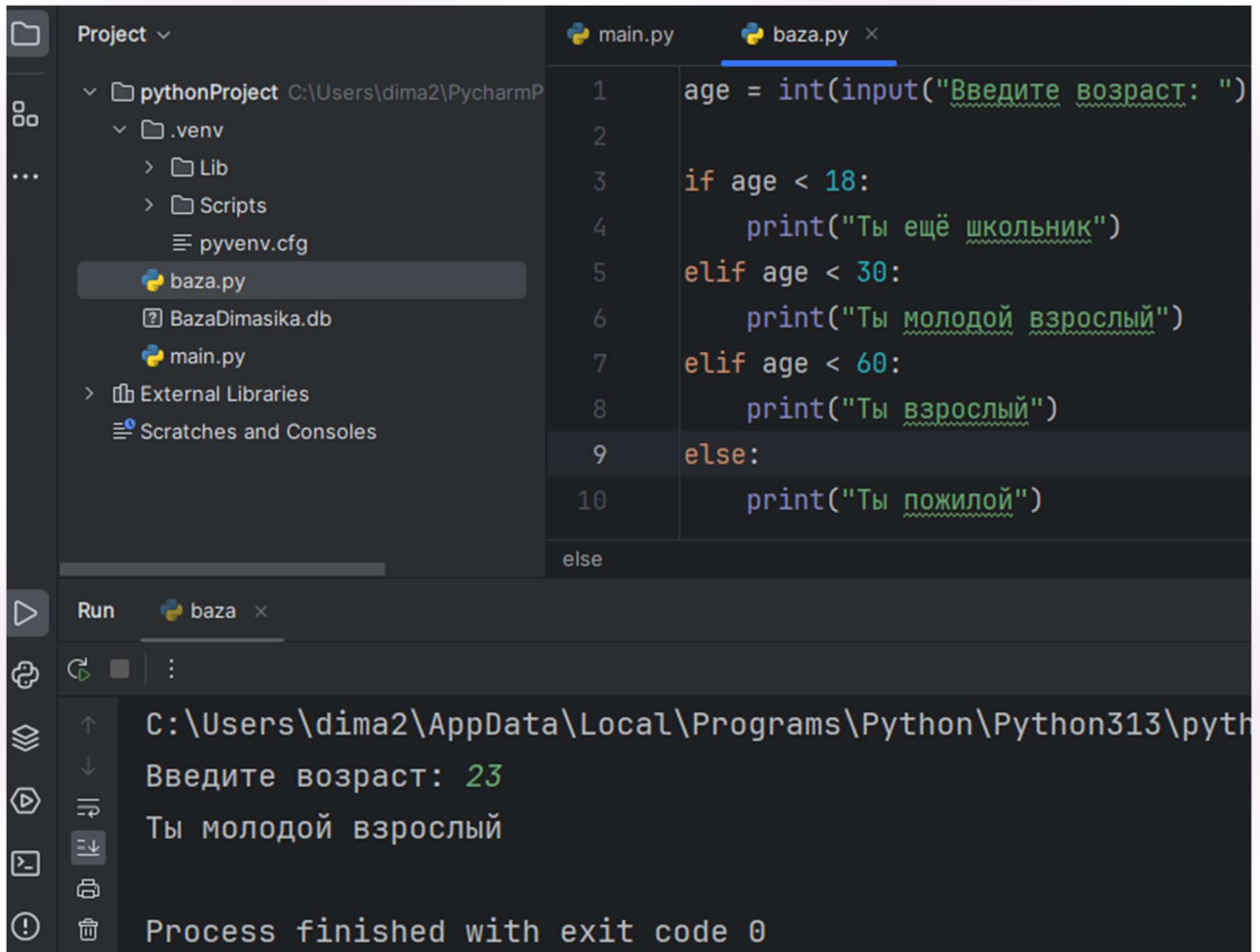
Если оно меньше 0 или больше 100 → "Выход за пределы",
иначе "Норма".



Оператор elif

Когда вариантов больше двух можно использовать `elif` вместо `if`

`elif` – сокращенно от `else if`, то есть если `if` не выполнен, то мы дальше проверяем уже по другому условию `if`



The screenshot shows the PyCharm IDE interface. On the left is the Project Explorer showing a project named 'pythonProject' with files like 'baza.py' and 'main.py'. The main editor displays the code in 'baza.py':

```
1 age = int(input("Введите возраст: "))
2
3 if age < 18:
4     print("Ты ещё школьник")
5 elif age < 30:
6     print("Ты молодой взрослый")
7 elif age < 60:
8     print("Ты взрослый")
9 else:
10    print("Ты пожилой")
```

Below the editor is the Run console. It shows the command 'Run' for 'baza.py'. The output of the program is:

```
C:\Users\dima2\AppData\Local\Programs\Python\Python313\python.exe
Введите возраст: 23
Ты молодой взрослый
Process finished with exit code 0
```

Так удобнее, чем писать много вложенных `if`



1. Спроси у пользователя его оценку по 5-балльной шкале.

- Если это 5 - выведи "Отлично!"
- 4 - выведи "Хорошо"
- 3 - выведи "Удовлетворительно"
- 2 или 1 - выведи "Неудовлетворительно"

2. Спроси у пользователя число.

- Если оно больше 100 выведи "Очень большое"
- Если от 10 до 100 выведи "Среднее"
- Если меньше 10 выведи "Маленькое"

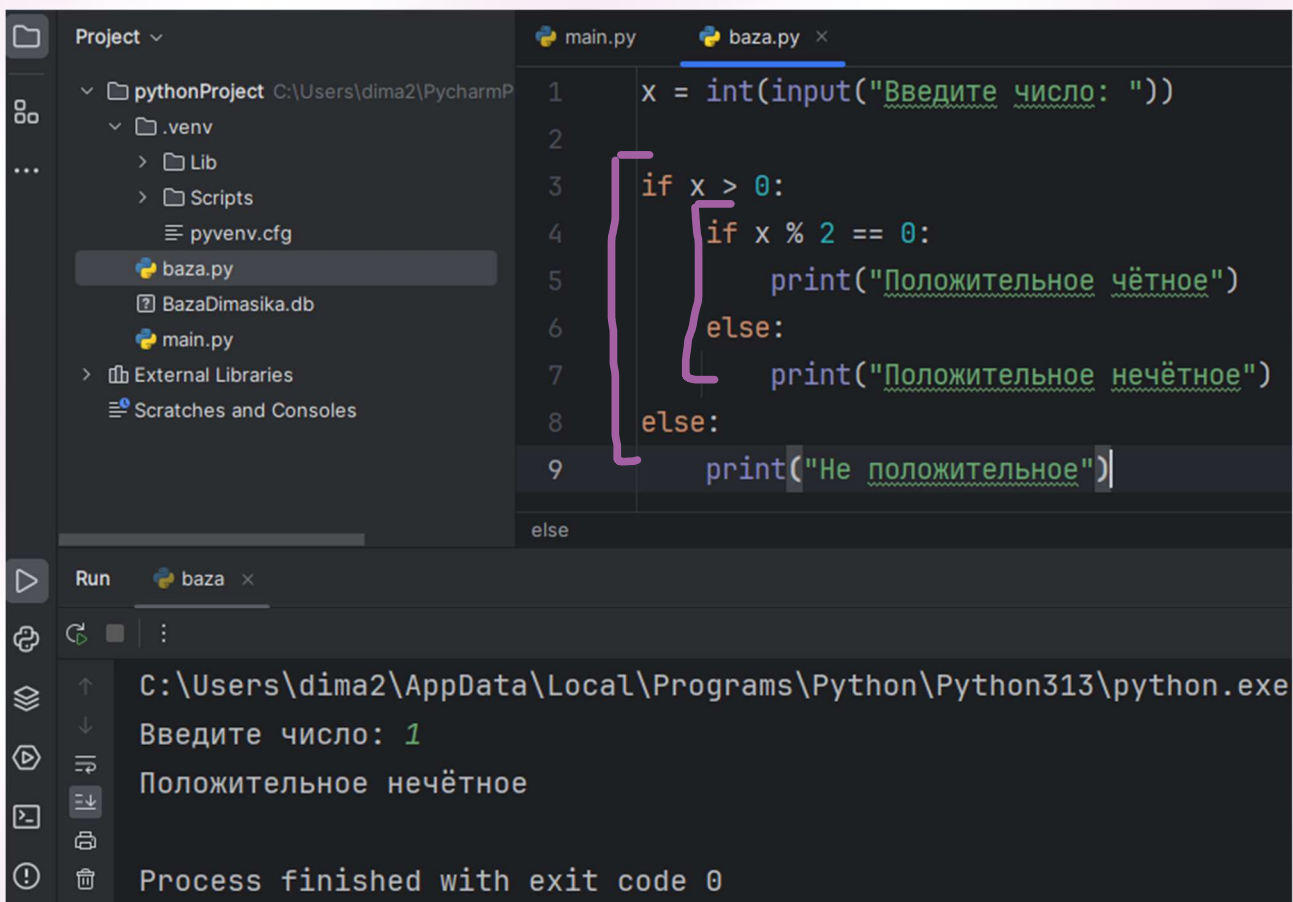


Вложенные условия

В if можно засунуть еще один if, чтобы у тебя было условие внутри условия

```
x = int(input("Введите число: "))

if x > 0:
    if x % 2 == 0:
        print("Положительное чётное")
    else:
        print("Положительное нечётное")
else:
    print("Не положительное")
```



The screenshot shows the PyCharm IDE interface. The left sidebar displays the project structure for 'pythonProject', including files like 'baza.py' and 'main.py'. The main editor window shows the code for 'baza.py', which implements the nested if-else logic for checking if a number is positive and even. The code is as follows:

```
1 x = int(input("Введите число: "))
2
3 if x > 0:
4     if x % 2 == 0:
5         print("Положительное чётное")
6     else:
7         print("Положительное нечётное")
8 else:
9     print("Не положительное")
```

Below the editor, the 'Run' tab shows the execution output. The command prompt displays the input '1' and the resulting output 'Положительное нечётное'. The process finished with exit code 0.

Главное не забывать проставлять табуляцию – парные if и else должны быть на одной линии!



1. Спроси у пользователя число.

- Если оно положительное:
 - Если чётное выведи "Положительное чётное"
 - Иначе выведи "Положительное нечётное"
- Если отрицательное выведи "Отрицательное"
- Если ноль выведи "Ноль"

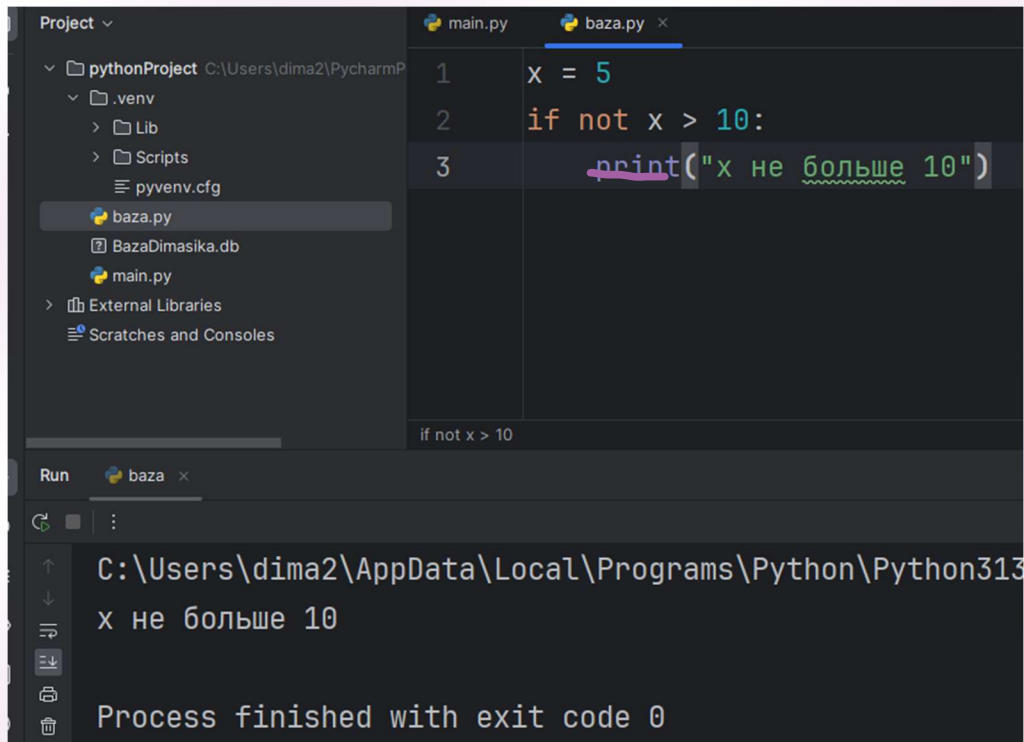
2. Спроси у пользователя логин.

- Если "admin": спроси пароль
 - Если пароль "1234" → "Добро пожаловать, админ!"
 - Иначе → "Неверный пароль"
- Если не "admin" → "Нет доступа"



Логический оператор not

Проверяемое условие можно перевернуть, если прямо перед ним указать оператор **not**



The screenshot shows the PyCharm IDE with a project named 'pythonProject'. The file explorer on the left shows the project structure. The main editor displays a file named 'baza.py' with the following code:

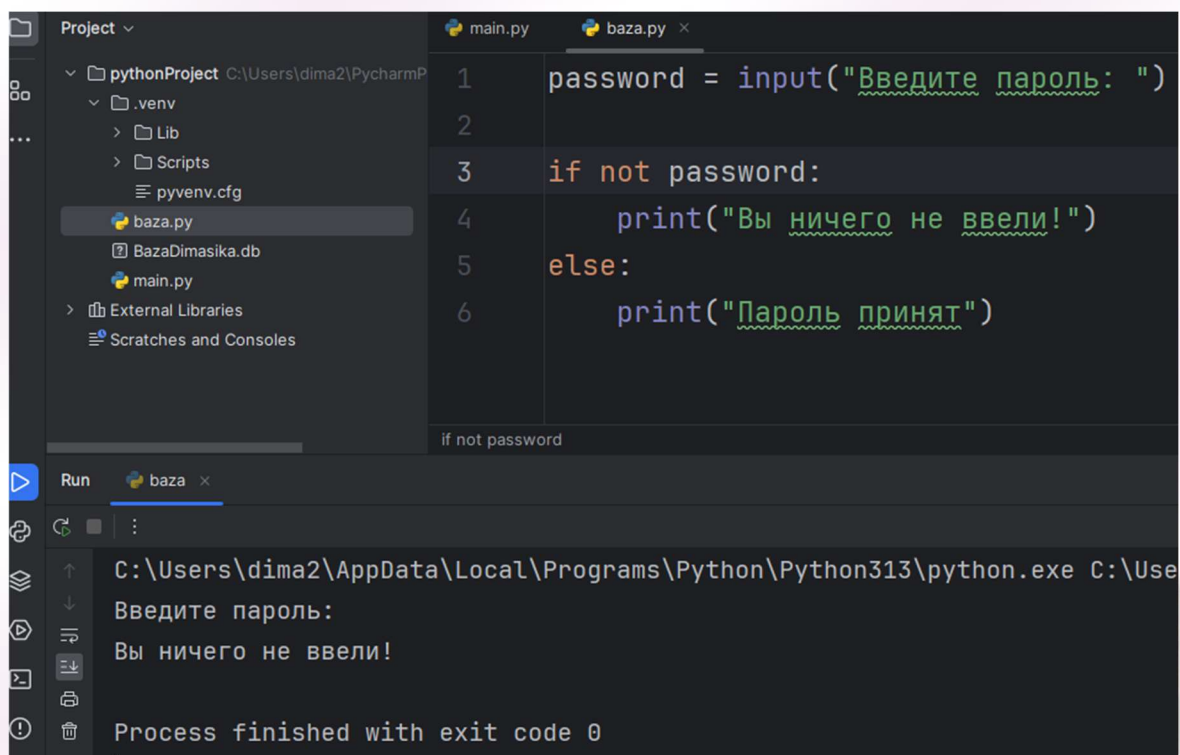
```
1 x = 5
2 if not x > 10:
3     print("x не больше 10")
```

The output console at the bottom shows the execution of the script:

```
C:\Users\dima2\AppData\Local\Programs\Python\Python313\python.exe C:\Users\dima2\AppData\Local\Programs\Python\Python313\python.exe
x не больше 10
Process finished with exit code 0
```

not превращает:

- **True -> False**
- **False -> True**



The screenshot shows the PyCharm IDE with a project named 'pythonProject'. The file explorer on the left shows the project structure. The main editor displays a file named 'baza.py' with the following code:

```
1 password = input("Введите пароль: ")
2
3 if not password:
4     print("Вы ничего не ввели!")
5 else:
6     print("Пароль принят")
```

The output console at the bottom shows the execution of the script:

```
C:\Users\dima2\AppData\Local\Programs\Python\Python313\python.exe C:\Users\dima2\AppData\Local\Programs\Python\Python313\python.exe
Введите пароль:
Вы ничего не ввели!
Process finished with exit code 0
```

1. Проверка пустого ввода

Спроси у пользователя логин.

Если он пустой (строка ""), выведи: «Что бубнишь? Не слышу!»

2. Спроси у пользователя имя

Если оно не "admin", выведи: «Я запрещаю вам»

Иначе выведи: «Я разрешаю вам админить»

3. Не положительное число

Спроси у пользователя число.

Если оно не положительное (меньше или равно нулю), выведи: «Число не положительное»

