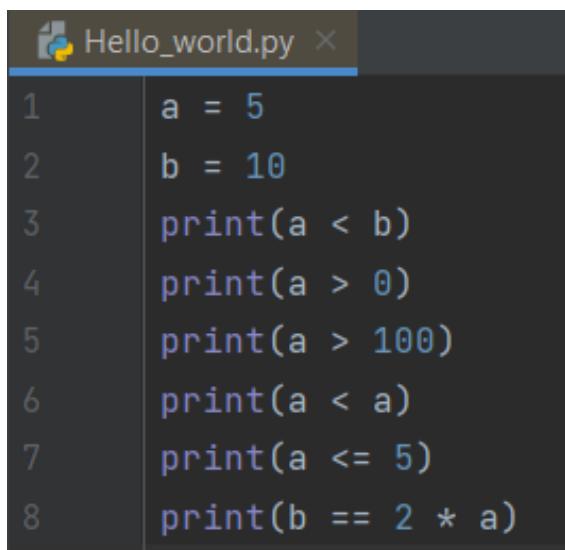


Логически изрази и проверки, оператори за сравнение

Оператори за сравнение

| Оператор | Означение |
|----------------------------|--------------------|
| равенство | <code>==</code> |
| различно | <code>!=</code> |
| По-голямо | <code>></code> |
| По голямо или равно | <code>>=</code> |
| По-малко | <code><</code> |
| По-малко или равно | <code><=</code> |

В програмирането можем да сравняваме стойности, резултатът от логическите изрази е **True** или **False** (фиг.1)



```

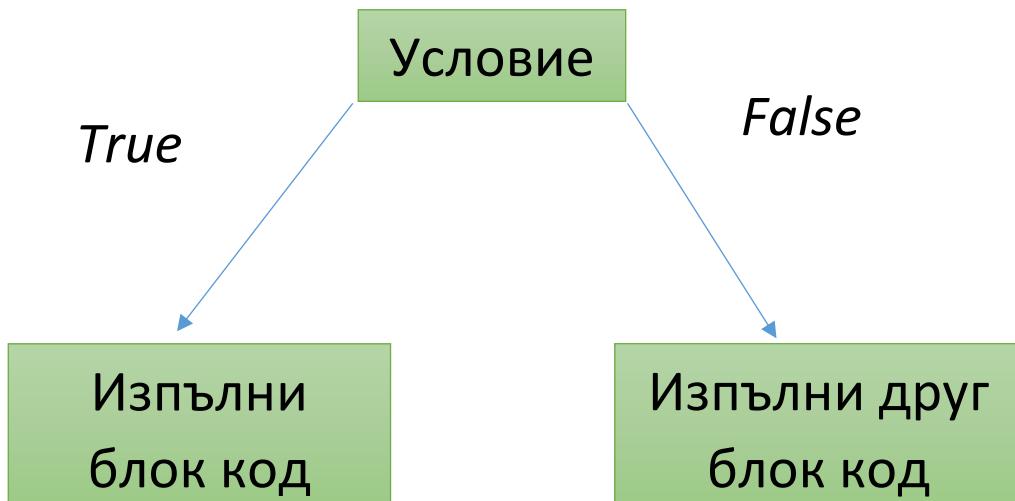
1 a = 5
2 b = 10
3 print(a < b)           True
4 print(a > 0)           True
5 print(a > 100)          False
6 print(a < a)            False
7 print(a <= 5)           True
8 print(b == 2 * a)        True

```

Фиг. 1

УСЛОВНИ КОНСТРУКЦИИ

Условен оператор – това е команда в програмирането, чрез която казваме на компютъра да вземе решение. Той работи като въпрос: „Вярно ли е това?“. Ако отговорът е „Да“, компютърът изпълнява една задача, а ако е „Не“ – изпълнява друга или просто продължава напред. Това е „мозъкът“ на нашата програма, който ѝ позволява да избира между различни пътища. В Python тези оператори са **if**, **elif**, **else**



if (ако): „Ако си избрал опция 1...“

elif (иначе ако): „Ако не е 1, но пък е 2...“

else (във всички останали случаи): „Ако не е нито 1, нито 2...“

Синтаксис на условна конструкция:

if <условие>:

код за изпълнение

else:

код за изпълнение

Прие невярност (false) на условието, можем да изпълним други действия – чрез **else** конструкция.

Индентациите (отстъпът) въвеждат блок от код (група команди), които се изпълняват в редът, който отговаря на условието.

```
color = 'red'
if color == 'red':
    print('red')
else:
    print('yellow')
print('bye')
```



Без индентация ще се изпълнява и последният ред, независимо от условието

```
1 color = 'red'
2 if color == 'red':
3     print('red')
4 else:
5     print('yellow')
6 print('bye')
7
```

```
C:\Users\SatellitePro\PycharmProject
red
bye

Process finished with exit code 0
```

Алгоритъм, който определя дали зададено число е четно или нечетно.

```
1 num = int(input())
2 if num % 2 == 0:
3     print('even')
4 else:
5     print('odd')
6
```

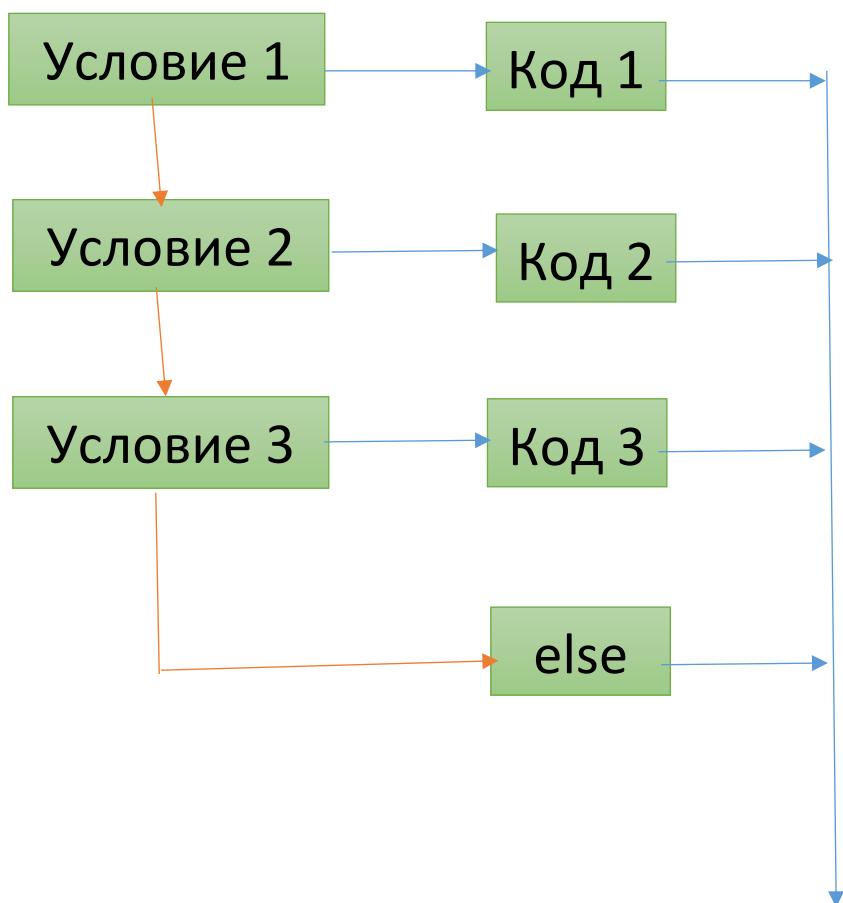
Тази програма показва, че условието за проверка може да бъде дори и числов израз

Серии от проверки

Това са по-сложни условни конструкции, при които имаме повече от едно разклонение, тук ключов е операторът **elif** (else if), при истинност на едно условие, не се продължава към проверяване на следващите условия.

Синтаксис

```
if <условие>:  
    блок от код  
elif <условие>:  
    блок от код  
elif <условие>:  
    блок от код  
...  
else:  
    блок от код
```



Програмата проверява условията последователно, установява кое е вярното и приключва, без да провери останалите.

Нека се върнем към нашия мини-проект и да дадем избор на потребителя, като добавим логиката на менютата. Използвайки следния код, допишете сами останалата част.

if izbor == 1:

```
chislo = input("Въведете двоично число: ")
```

```
print(f"Избрахте преобразуване от Двоична към Десетична за: {chislo}")
```

Програмата трябва да ни отпечата съобщение с изборът направен от потребителя, ако той е избрал невалидна опция да се отпечатва следното съобщение "Грешка! Моля, изберете число между 1 и 4.".

Краен резултат:

```
# Условните оператори, отговарящи за менюто
if izbor == 1:
    chislo = input("Въведете двоично число: ")
    print(f"Избрахте преобразуване от Двоична към Десетична за: {chislo}")

elif izbor == 2:
    chislo = int(input("Въведете десетично число: "))
    print(f"Избрахте преобразуване от Десетична към Двоична за: {chislo}")

elif izbor == 3:
    chislo = input("Въведете шестнадесетично число: ")
    print(f"Избрахте преобразуване от Шестнадесетична към Десетична за: {chislo}")

elif izbor == 4:
    chislo = int(input("Въведете десетично число: "))
    print(f"Избрахте преобразуване от Десетична към Шестнадесетична за: {chislo}")

else:
    print("Грешка! Моля, изберете число между 1 и 4.")
```