

if - проверка условия

Оператор **if** используется для выполнения каких-либо действий при выполнении условия:

```
In [1]: company = "my.com"

if "my" in company:
    print("Условие выполнено!")
```

Условие выполнено!

```
In [2]: company = "example.net"

if "my" in company or company.endswith(".net"):
    print("Условие выполнено!")
```

Условие выполнено!

if - else

Оператор **else** позволяет выполнить какой-либо код, если условие в блоке **if** не выполнилось:

```
In [1]: company = "google.com"

if "my" in company:
    print("Условие выполнено!")
else:
    print("Условие не выполнено!")
```

Условие не выполнено!

if - elif - else

Оператор **elif** используется, когда нужно проверить несколько разных условий друг за другом:

```
In [2]: company = "google.com"

if "my" in company:
    print("Подстрока my найдена")
elif "google" in company:
    print("Подстрока google найдена")
else:
    print("Подстрока не найдена")
```

Подстрока google найдена

Аналог тернарного оператора

```
In [3]: score_1 = 5
score_2 = 0

winner = "Argentina" if score_1 > score_2 else "Jama
ica"

print(winner)
```

Argentina

while

Оператор **while** позволяет выполнять блок кода до тех пор пока выполняется условие:

```
In [4]:
```

```
i = 0
```

```
100  
while i < 100:  
    i += 1
```

Цикл for, объект range

Выражение **for .. in** это еще один способ выполнить блок кода - но оно позволяет выполнить блок кода для каждого из элементов из последовательности:

```
In [5]: name = "Alex"  
  
for letter in name:  
    print(letter)
```

```
A  
l  
e  
x
```

Встроенный объект **range** позволяет итерироваться по целым числам:

```
In [9]: for i in range(3):  
        print(i)
```

```
0  
1  
2
```

```
In [6]: result = 0

        for i in range(101):
            result += i

        print(result)
```

5050

```
In [8]: for i in range(5, 8):
        print(i)
```

5
6
7

```
In [9]: for i in range(1, 10, 2):
        print(i)
```

1
3
5
7
9

```
In [10]: for i in range(10, 5, -1):
        print(i)
```

10
9
8
7
6

pass

Определяет пустой блок, который ничего не делает

```
In [11]: for i in range(100):  
         pass
```

break

Оператор **break** позволяет выйти из цикла досрочно:

```
In [15]: result = 0  
  
         while True:  
             result += 1  
             if result >= 100:  
                 break  
  
         print(result)
```

100

```
In [16]: for i in range(10):  
         if i == 5:  
             break  
         print(i)
```

0
1
2
3
4

continue

Оператор **continue** используется, когда в блоке цикла нужно перейти к следующей итерации цикла без выполнения оставшихся инструкций в блоке:

```
In [17]:
```

```
result = 0

for i in range(10):
    if i % 2 != 0:
        continue
    result += i

print(result)
20
```

Применим на практике

```
In [16]: import random

number = random.randint(0, 100)

while True:
    answer = input('Угадайте число: ')
    if answer == "" or answer == "exit":
        print("Выход из программы")
        break

    if not answer.isdigit():
        print("Введите правильное число")
        continue

    answer = int(answer)

    if answer == number:
        print('Совершенно верно!')
        break

    elif answer < number:
        print('Загаданное число больше')
    else:
        print('Загаданное число меньше')
```

Угадайте число: ора

Введите правильное число