

Циклы в Python

Цикл While

while условное_выражение:
инструкции

while условное_выражение:
инструкции
else:
инструкции

Пример 1

```
number = 1
```

```
while number < 5:
```

```
    print(f"number = {number}")
```

```
    number += 1
```

```
print("Работа программы завершена")
```

Пример 2

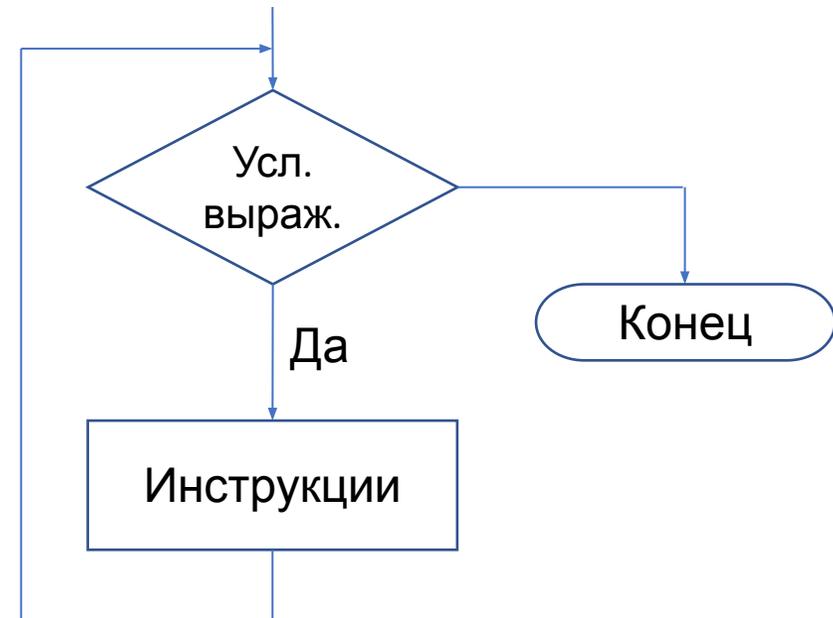
```
while i <= 10:
```

```
    print(i)
```

```
    i += 1
```

```
else:
```

```
    print('Цикл окончен, i =', i)
```



Цикл For

```
for переменная in набор_значений:  
    инструкции
```

```
for переменная in набор_значений:  
    инструкции  
else:  
    инструкции
```

Пример 1

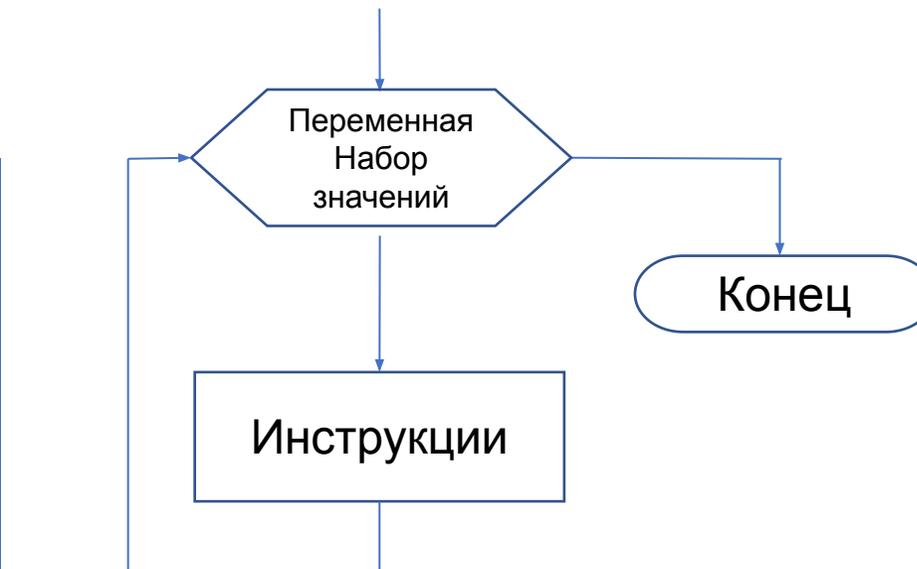
```
message = "Hello"  
for c in message:  
    print(c)
```

Пример 2

```
for i in range(4): # 0, 1, 2, 3:  
    print(i)
```

Пример 3

```
for i in range(1, 5): # 1,2,3,4,5  
    sum += 1  
print(sum)
```



Пример 4

```
a = int(input())  
while a != 0:  
    if a < 0:  
        print('Встретилось отрицательное число', a)  
        break  
    a = int(input())  
else:  
    print('Ни одного отрицательного числа не  
встретилось')
```

Выход из цикла. `break` и `continue`

Оператор `break` осуществляет выход из цикла

```
number = 0
while number < 5:
    number += 1
    if number == 3 : # если number = 3,
                    ВЫХОДИМ ИЗ ЦИКЛА
        break
print(f"number = {number}")
```

В отличие от оператора `break` оператор `continue` выполняет переход к следующей итерации цикла без его завершения.

```
number = 0
while number < 5:
    number += 1
    if number == 3 : # если number = 3,
                    переходим к новой итерации цикла
        continue
print(f"number = {number}")
```

Вложенные циклы

while условие_выражение:

инструкции

while условие_выражение:

инструкции

for переменная **in** набор_значений:

инструкции

for переменная **in** набор_значений:

инструкции

while условие_выражение:

инструкции

for переменная **in** набор_значений:

инструкции

Пример 1.

```
n = int(input())
```

```
m = int(input())
```

```
for i in range(n):
```

```
    for j in range(m): # вложенный цикл
```

```
        print('*', end='')
```

```
    print()
```

Пример 2.

```
i = 1
```

```
while i < 10:
```

```
    j = 1
```

```
    while j < 10:
```

```
        print(i, '*', j, '=', i * j, end='\t')
```

```
        j += 1
```

```
    i += 1
```

```
    print()
```

Самостоятельные задания

Цикл While

1. Вводится N натуральных чисел. Найти количество четных цифр в каждом из них.
2. Подсчитать количество двузначных чисел, кратных 3.
3. Дано натуральное число n . Вычислить $n!$ ($n! = 1 * 2 * 3 * \dots * n$)
4. Готовясь к соревнованиям, лыжник в первый день пробежал 10 км, затем каждый день увеличивал расстояние на 10%. Сколько километров пробежал он за неделю тренировок? На какой день он пробежал больше 15 км?
5. Определить, является ли число n простым.

Цикл For

1. Найти: а) сумму квадратов всех целых чисел от a до 50 (значение a вводится с клавиатуры; $a \leq 50$); б) произведение всех целых чисел от a до b (значения a и b вводится с клавиатуры; $b \geq a$).
2. Дано целое число N (> 0). Вывести произведение $1 \cdot 2 \cdot \dots \cdot N$. Чтобы избежать целочисленного переполнения, вычислять это произведение с помощью вещественной переменной и выводить его как вещественное число.
3. Найти среднее арифметическое квадратов всех целых чисел от 100 до b (значение b вводится с клавиатуры; $b \geq 100$);
4. Определить суммарный объем в литрах 12 вложенных друг в друга шаров со стенками 5 мм. Внутренний диаметр внутреннего шара равен 10 см. Считать, что шары вкладываются друг в друга без зазора.