

# Циклические конструкции

## Резюме

### 1. Понятие циклических конструкций

- Виды простейших алгоритмов:
  - Линейный алгоритм — набор команд (указаний), выполняемых последовательно во времени друг за другом.
  - Разветвляющийся алгоритм — алгоритм, содержащий хотя бы одно условие, в результате проверки которого может осуществляться разделение на несколько параллельных ветвей алгоритма.
  - Циклический алгоритм — алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций) над новыми исходными данными. К циклическим алгоритмам сводится большинство методов вычислений, перебора вариантов. Цикл программы — последовательность команд (серия, тело цикла), которая может выполняться многократно (для новых исходных данных) до удовлетворения некоторого условия.
- Цикл — это управляющая конструкция, многократно исполняющая блок кода.
- Существуют три классических разновидности циклов:
  - цикл с предусловием;
  - цикл с постусловием;
  - цикл со счётчиком.
- Также для современных языков программирования характерен `foreach`-цикл (совместный цикл), который выполняет определённые действия для всех элементов из какого-то набора.
- Первые два варианта циклических конструкций выполняют участок кода, пока указанное условие истинно. Они отличаются тем, когда проверяется условие: в начале каждого шага (итерации) или в конце. Циклы с предусловием и с постусловием взаимозаменяемы и любой из них можно переписать с использованием другого, при этом (по мнению создателей языка Python) циклы с постусловием используются

значительно реже и хуже читаются из-за места указания условия, поэтому в Python существуют только циклы с предусловием (циклы “while”).

- Цикл со счётчиком (цикл “for”) используется для повторения определённых действий конкретное количество раз, при этом доступен счётчик – номер текущей итерации.
- В Python цикл “for” является по своей сути циклом совместным циклом, то есть цикл со счётчиком реализуется при помощи foreach-цикла и функции, которая возвращает последовательность целых чисел. Как на самом деле работает цикл “for” в Python мы изучим позже (в общих чертах рассмотрим в седьмом уроке данного курса и основательно изучим в курсе Python Essential при изучении коллекций и последовательностей). Пока что мы будем рассматривать цикл “for” как цикл со счётчиком.
- Цикл while в Python – цикл с предусловием. Синтаксис:

```
while условие:
    операторы
```

- Эта конструкция похожа на оператор if, который был изучен в предыдущем уроке, только выполняет блок операторов не один раз, а несколько, пока условие истинно. Все, что было сказано про условия, отступы и вложенные операторы справедливо и для оператора while (а также for, который будет рассмотрен в этом уроке позже).
- Цикл while не выполнится ни разу, если условие изначально было ложным.
- Условие должно зависеть от какого-либо состояния, которое изменяется, иначе цикл будет выполняться бесконечно (пока пользователь не прервёт выполнение программы), если оно изначально было истинным.
- Пример цикла while:

```
x = 0
while x <= 0:
    x = int(input('Введите положительное число: '))
```

- Если необходимо создать бесконечный цикл, в качестве условия принято использовать True:

```
print('Все натуральные числа:')
n = 1
while True:
    print(n)
    n += 1
```

- Если необходимо досрочно завершить выполнение цикла, то можно прервать его при помощи оператора break.
- Один проход цикла через блок операторов называется итерацией.
- Если необходимо прервать выполнение текущей итерации и перейти к началу следующей, то используется оператор continue.
- Оператор while также может иметь ветку else (по аналогии с if). В начале каждой итерации интерпретатор проверяет истинность условия выполнения цикла, и если оно истинно, то исполняет ветку while, иначе он исполняет ветку else (если она присутствует) и завершает выполнение цикла, причём это может произойти и перед первой итерацией, если условие изначально было ложным. Однако если цикл был прерван оператором break, то ветка else не исполняется.
- Цикл со счётчиком (цикл for) – это цикл, в котором переменная – счётчик итераций изменяет своё значение от начального до конечного с определённым шагом.
- Цикл for со счётчиком в Python:

```
for переменная in range(конечное_значение):
    операторы
```

или

```
for переменная in range(начальное_значение, конечное_значение):
    операторы
```

или

```
for переменная in range(начальное_значение, конечное_значение, шаг):
    операторы
```

- В первом варианте переменная-счётчик принимает значение от нуля включительно до заданного значения, не включая его; во втором задаётся начальное значение вместо нуля; в третьем указывается ещё и шаг изменения переменной-счётчика (по умолчанию

- Можно задать начальное значение больше, чем конечное, и шаг, равный -1, и получить обратный цикл.
- Перед началом каждой итерации переменная-счётчик принимает следующее значение из сгенерированной интерпретатором последовательности. Таким образом, любые изменения переменной-счётчика будут потеряны на следующей итерации. После завершения цикла переменная-счётчик сохраняется и равна последнему значению, которое она принимала.
- Операторы break и continue в цикле for имеют такой же смысл, как и в цикле while.
- Аналогично циклу while, цикл for также может иметь ветку else.
- Вложенные циклы – это циклы, находящиеся внутри других циклов. Цикл, который вложен в тело другого, называется внутренним циклом. Цикл, в тело которого вложен другой цикл, называется внешним.

### Закрепление материала

- Что такое цикл?
- Какие типы циклов вы знаете?
- Какие типы циклов существуют в языке Python?
- Охарактеризуйте цикл с предусловием. Как реализовать его на Python?
- Охарактеризуйте цикл со счётчиком. Как реализовать его на Python?
- Как завершить выполнение цикла из его тела?
- Как пропустить итерацию в цикле?
- Что такое вложенные циклы?

### Дополнительное задание

#### Задание

Создайте программу, которая рисует на экране прямоугольник из звёздочек заданной пользователем ширины и высоты.

### Самостоятельная деятельность учащегося

#### Задание 1

Даны числа  $a$  и  $b$  ( $a < b$ ). Выведите сумму всех натуральных чисел от  $a$  до  $b$  (включительно).

#### Задание 2

Факториалом числа  $n$  называется число  $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$ . Создайте программу, которая вычисляет факториал введённого пользователем числа.

#### Задание 3

Используя вложенные циклы и функции `print('*', end='')`, `print(' ', end='')` и `print()` выведите на экран прямоугольный треугольник.

### Рекомендуемые ресурсы

Документация по Python

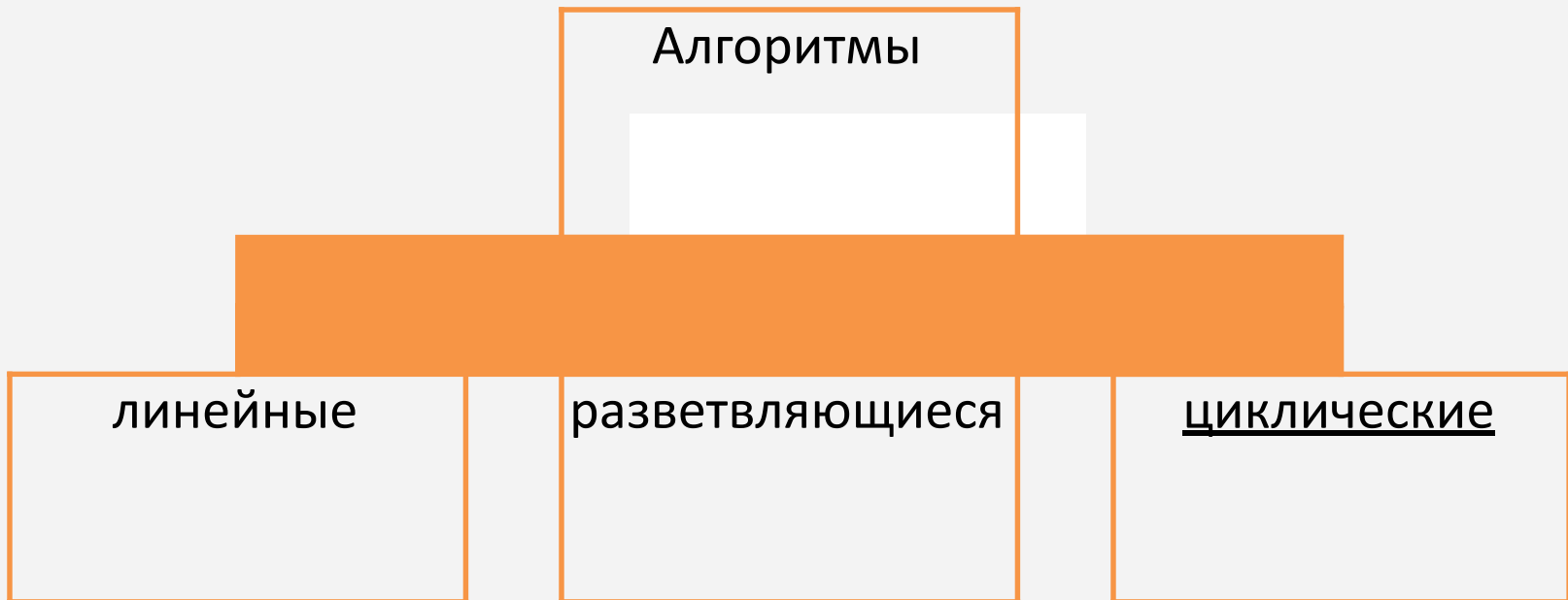
[https://docs.python.org/3/reference/compound\\_stmts.html#the-while-statement](https://docs.python.org/3/reference/compound_stmts.html#the-while-statement)

[https://docs.python.org/3/reference/compound\\_stmts.html#the-for-statement](https://docs.python.org/3/reference/compound_stmts.html#the-for-statement)

[https://docs.python.org/3/reference/simple\\_stmts.html#break](https://docs.python.org/3/reference/simple_stmts.html#break)

[https://docs.python.org/3/reference/simple\\_stmts.html#continue](https://docs.python.org/3/reference/simple_stmts.html#continue)

# Основные алгоритмические конструкции

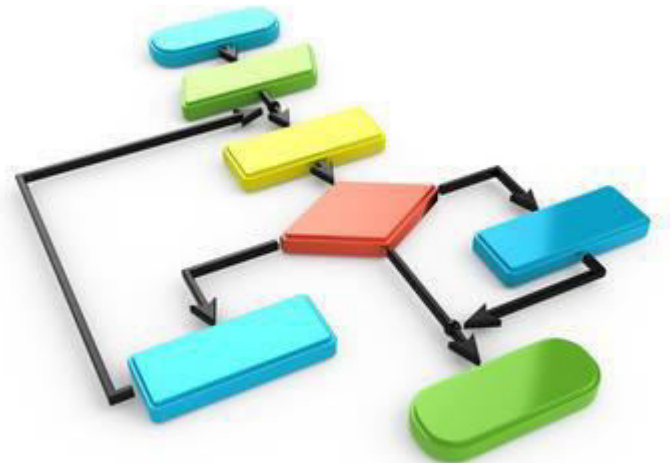


# Понятие цикла

Цикл — это управляющая конструкция, предназначенная для многократного исполнения набора инструкций.

Виды циклов:

- цикл с предусловием
- цикл с постусловием
- цикл со счётчиком
- совместный цикл



Цикл с предусловием

## Цикл while

`while` условие:  
операторы

Здесь:

- условие – это логическое выражение,
- операторы – это последовательность каких-либо других команд.

Блок операторов не может быть пустым. Если такая необходимость возникает, можно воспользоваться оператором `pass`, который не делает ничего.

Блок операторов выполняется до тех пор, пока условие продолжения цикла истинно. Если оно изначально было ложным, то он не выполняется ни разу.

## Операторы break и continue

Если необходимо прервать выполнение цикла досрочно, то используется оператор break.

```
while True:
    response = input('Введите команду')
    if response == 'exit':
        break
```

Если необходимо завершить выполнение текущей итерации цикла и перейти к следующей, то используется оператор continue.

```
x = 0
while x < 10:
    x += 1
    if x == 5:
        continue
    print('Текущее число равно', x)
    print('(но число 5 мы не выводим)')
```



## Цикл while с веткой else

```
while условие:  
    блок_операторов_1  
else:  
    блок_операторов_2
```

Оператор while также может иметь ветку else (по аналогии с if). В начале каждой итерации интерпретатор проверяет истинность условия выполнения цикла, и если оно истинно, то исполняет ветку while, иначе он исполняет ветку else (если она присутствует) и завершает выполнение цикла, причём это может произойти и перед первой итерацией, если условие изначально было ложным. Однако если цикл был прерван оператором break, то ветка else не исполняется.

Цикл со счётчиком

---

## Цикл for

Цикл со счётчиком (цикл for) – это цикл, в котором переменная – счётчик итераций изменяет своё значение от начального до конечного с определённым шагом.

```
for переменная in  
    range(конечное_значение):  
    операторы
```

или

```
for переменная in range(начальное_значение, конечное_значение):  
    операторы
```

или

```
for переменная in range(начальное_значение, конечное_значение,  
шаг):  
    операторы
```

Начальное значение включается в диапазон, конечное – нет.

# Вложенные циклы

Вложенные циклы – это циклы, находящиеся внутри других циклов. Цикл, который вложен в тело другого, называется внутренним циклом. Цикл, в тело которого вложен другой цикл, называется внешним.

```
for i in range(10):
```

```
for j in range(30):
    print('*', end='')
```

```
print()
```

A screenshot of a Python 3.4.1 Shell window. The title bar shows "Python 3.4.1 Shell" and standard window controls. The menu bar includes File, Edit, Shell, Debug, Options, Windows, and Help. The code being executed is a nested loop: >>> for i in range(10):  
 for j in range(30):  
 print('\*', end='')  
 print(). This prints a grid of 10 rows and 30 columns of asterisks. The output consists of 10 lines, each containing 30 asterisks. At the bottom right, a status bar indicates "Ln: 30 Col: 4".