

ХОЧУ ПОМОЧЬ
ПРОЕКТУ

Циклы и управление ветвлением

Яндекс Взгляд · Опрос

Выберите 1 или несколько ответов

Какие сервисы проверки истории автомобилей вы знаете?

Автотека/Авито

Avtocod

Avinfobot

ПроАвто/Auto.ru

Ни один из вариантов

1 из 3 вопросов

Продолжить

[Справочник по языку Python3.](#) / Циклы и управление ветвлением

Конструкции, управляющие ветвлением программы Python v.3 имеют свои особенности по сравнению с другими языками программирования.

Алгоритм ветвления должен содержать хотя бы одну проверку условия. Для проверки условий в Python есть оператор `if ... elif ... else`.

Циклы служат для многократного повторения каких-либо действий. Для этого в Python есть операторы `for ... in` и `while`.

Также разберем поведение операторов `break` и `continue`.

[Проверка условий, конструкция if/elif/else в Python](#)

Оператор `if` используется для условного исполнения. Ключевое слово `elif` является сокращением от 'else if'. Часть конструкции `elif` не является обязательной, а так же может повторяться сколько угодно

[Конструкция if/else в одну строку в Python](#)

Иногда `if/else` настолько просты, что кажется лишнем тратить на них четыре строки и хочется вложить такую конструкцию внутрь другой. трехместное выражение вида `x = a if item else b` позволяет это сделать

[Имитация оператора switch/case в Python](#)

Случай с оператором `switch/case` очень похож на оператор `if/elif/else`. Приведенный выше код просто читать и понимать. Но это все еще не оператор `switch`. `switch` оценивает точность, но также и сходство слова. В python есть объект, который действует как селектор.

[while, цикл по условию в Python](#)

Конструкция цикла `while` многократно проверяет выражение `expression`, при этом выполняет код внутри блока `while`, и если выражение стало ложно (`False`), то выполняется код внутри блока `else`, если он присутствует

[Цикл for в Python](#)

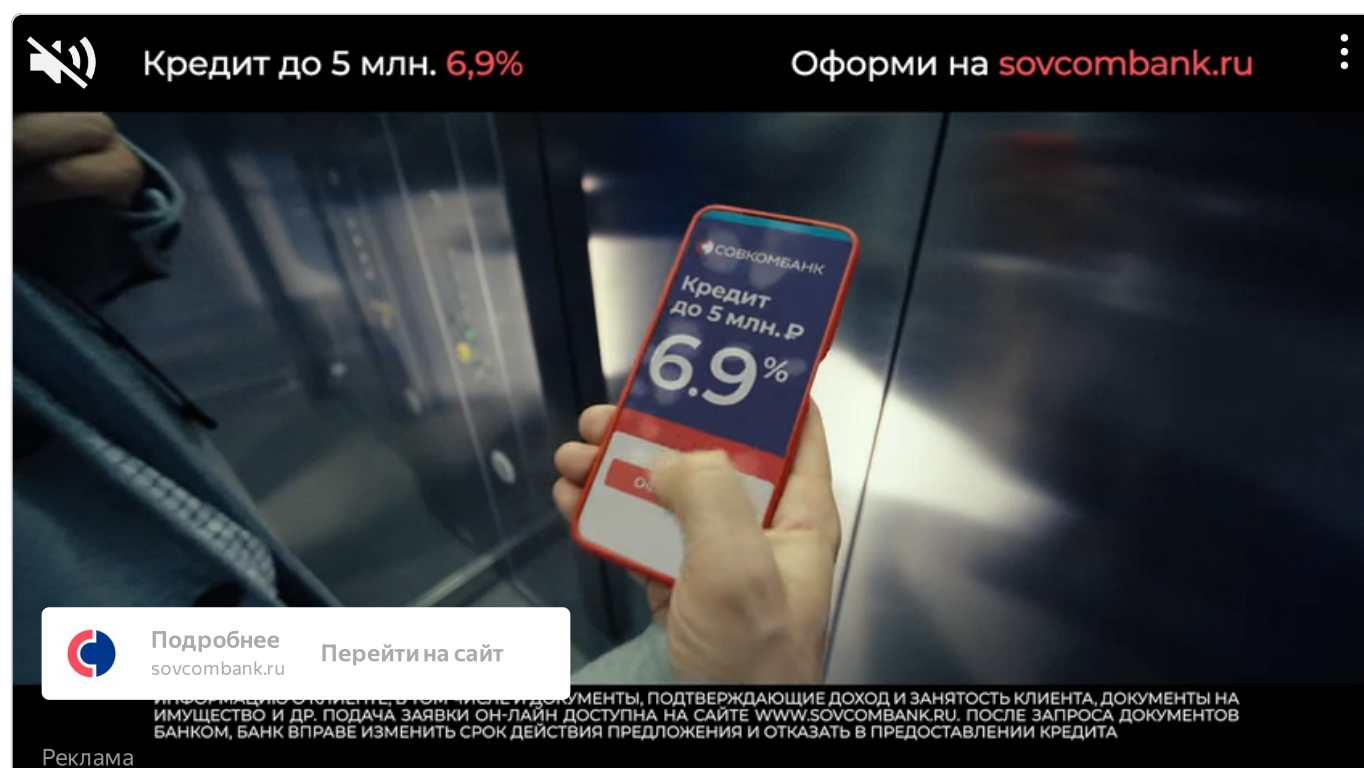
Цикл `for/in` в Python перебирает элементы любой итерируемой последовательности (список `list`, строку `string`, кортеж `tuple`, словарь `dict` или другого итерационного объекта) в том порядке, в котором они появляются.

[Числовые последовательности для цикла Python](#)

Если вам нужно запустить цикла `for ... in` определенное число раз, в этом вам поможет встроенная функция `range()`. Эта функция генерирует арифметические прогрессии

[Практическое применение `for/in/else` циклов в Python](#)

Рассмотрены разные случаи итерации по разным типам данных. Приведены подробные примеры кода.



[Инструкции `break`, `continue` и блок `else` в циклах Python](#)

Заявление `break`, как и в C, прерывает выполнение блока `for` или `while` с выходом из него. Операторы цикла могут иметь блок `else`, который выполняется, когда цикл заканчивается без прерывания `break`

[Оператор `pass` в Python](#)

Оператор `pass` ни ничего не делает. Он используется тогда, когда в коде требуется какой то синтаксис, но никаких действий производить не надо. Оператор `pass` можно использовать в качестве заполнителя для функции или условного блока

[Конструкция `match/case` в Python, сопоставление с образцом](#)

В Python 3.10 введена новая конструкция `match/case`, которая называется **Structural pattern matching** (соответствие структуре шаблона). Сопоставление списков, словарей, сложных структур и классов.