



Бесплатная электронная книга

УЧУСЬ

loops

Free unaffiliated eBook created from
Stack Overflow contributors.

#loops

.....	1
1:	2
.....	2
Examples.....	2
.....	2
.....	2
.....	2
.....	2
.....	2
.....	3
Foreach.....	3
Goto	3
.....	3
2:	4
.....	4
Examples.....	4
.....	4
.....	6

Около

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [loops](#)

It is an unofficial and free loops ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official loops.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

глава 1: Начало работы с циклами

замечания

В этом разделе представлен обзор того, какие циклы и почему разработчик может захотеть его использовать.

Следует также упомянуть о любых крупных предметах в циклах и ссылки на связанные темы. Поскольку для циклов Documentation for new, вам может потребоваться создать начальные версии этих связанных тем.

Examples

Типы петель

Цикл представляет собой структуру потока управления, которая определенно или неопределенно запускает набор операторов, написанных только один раз в коде, до тех пор, пока не будет выполнено определенное условие или процесс не будет завершен.

Циклы состояния

Эти петли повторяются в зависимости от состояния их условий.

Для петель

Для циклов обычно выполняется переменная как объект итерации. Например, для циклов можно запускать целое число, чтобы ограничить количество циклов, в течение которых цикл должен быть запущен, или массив, чтобы перебирать его.

В то время как петли

Хотя циклы - это самый базовый тип цикла условий, который продолжает работать до тех пор, пока его состояние не будет изменено (или пока не будет выполнен оператор `break`).

Варианты

Вариантом `while` является цикл `do... while while`. Это то же самое, что и `while`, за исключением того, что содержимое цикла запускается один раз перед проверкой состояния.

Другим вариантом является `until` тех `until / do... until` циклов, что делает то же самое, что и их коллеги `while` за исключением того, что они проверяют условие в обратном порядке - `while` циклы выполняются до тех пор, пока условие не станет ложным, и `until` циклы не будут выполняться до тех пор, пока условие правда.

Циклы коллекции

Эти петли повторяются путем итерации по коллекциям, таким как массивы или итерации.

Циклы Foreach

Цикл `forEach` выполняется в коллекции, выполняя код один раз для каждого элемента в коллекции, предоставляя значение и / или ключ элемента в качестве параметра.

Goto

`goto` - это набор инструкций между меткой и операцией `goto`.

Рекурсивные петли

В функциональном программировании рекурсивные циклы могут использоваться для рекурсивного запуска функции до тех пор, пока условие не будет выполнено. Это обычная причина ошибок переполнения стека.

Прочитайте Начало работы с циклами онлайн: <https://riptutorial.com/ru/loops/topic/5080/начало-работы-с-циклами>

глава 2: Для петель

Синтаксис

- `for (init; condition; increment) {content_code (); } // общий синтаксис`
- `for (int i = 0; i <числоRuns; ++ i) {actions_with (i); } // запускаем действие для числа numberRuns`
- `for (int i = 0; i <sizeof (array); ++ i) {actions_with (array [i]); } // итерация по массиву`

Examples

Общие для цикла

Большинство языков программирования поддерживают `for` -loop структуры управления.

Он обычно реализуется таким образом:

```
for(init; condition; increment){
    content_code ();
}
```

Вышеупомянутый псевдокод идентичен псевдокоду ниже:

```
init;
start_loop:
if(condition){
    content_code ();
    increment;
    goto start_loop;
}
```

Это показывает, что:

- `init` запускается до цикла, используемого для инициализации объектов для запуска этого цикла
 - В некоторых языках программирования, таких как Java, переменные могут быть *объявлены* в `init`, а область объявленных переменных будет ограничена этим циклом.
- `condition` - это условие, чтобы определить, когда цикл может быть запущен. Если это значение равно `false`, цикл прекратит выполнение.
- `increment` обычно является оператором, используемым для управления параметрами, используемыми в `condition`, поэтому, когда `increment` выполняется определенное количество раз, `condition` становится `false` и цикл прерывается.
- `content_code()` - это основной код, который должен быть запущен в цикле.

Прочитайте Для петель онлайн: <https://riptutorial.com/ru/loops/topic/6135/для-петель>

кредиты

S. No	Главы	Contributors
1	Начало работы с циклами	Community , SOFe
2	Для петель	SOFe