



EBook Gratuito

APPENDIMENTO

loops

Free unaffiliated eBook created from
Stack Overflow contributors.

#loops

Sommario

Di.....	1
Capitolo 1: Iniziare con i loop.....	2
Osservazioni.....	2
Examples.....	2
Tipi di anelli.....	2
Circuiti di condizione.....	2
Per i loop.....	2
Mentre i loop.....	2
varianti.....	2
Cicli di raccolta.....	2
Cicli Foreach.....	3
Goto loop.....	3
Loop ricorsivi.....	3
Capitolo 2: Per i loop.....	4
Sintassi.....	4
Examples.....	4
Generale per loop.....	4
Titoli di coda.....	5

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

Capitolo 1: Iniziare con i loop

Osservazioni

Questa sezione fornisce una panoramica dei loop e del motivo per cui uno sviluppatore potrebbe volerlo utilizzare.

Dovrebbe anche menzionare eventuali soggetti di grandi dimensioni all'interno di loop e collegarsi agli argomenti correlati. Poiché la Documentazione per cicli è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

Examples

Tipi di anelli

Un ciclo è una struttura di flusso di controllo per eseguire definitivamente o indefinitamente un insieme di istruzioni scritte una sola volta nel codice, fino a quando non viene soddisfatta una determinata condizione o il processo viene terminato.

Circuiti di condizione

Questi anelli vengono ripetuti in base allo stato delle loro condizioni.

Per i loop

I loop di solito vengono eseguiti su una variabile come soggetto dell'iterazione. Ad esempio, per i cicli può essere eseguito su un numero intero per limitare il numero di volte che il ciclo deve essere eseguito, o su un array per scorrere su di esso.

Mentre i loop

Mentre loop è il tipo più elementare di condition loop che continua a funzionare fino a quando non viene modificata la sua condizione (o finché non viene eseguita un'istruzione `break`).

varianti

Una variante di loop `while` è il `do... while` loop. È uguale a `while` loop, tranne per il fatto che il contenuto del ciclo viene eseguito una volta prima di controllare la condizione.

Un'altra variante è il `until / do... until` anse, che fa la stessa come le loro controparti in `while` tranne che controllano la condizione in modo opposto - `while` cicli eseguiti finché la condizione è falsa, e `until` loop corrono fino alla condizione è vero.

Cicli di raccolta

Questi cicli vengono ripetuti iterando su raccolte, come array o iterables.

Cicli Foreach

Un ciclo `forEach` viene eseguito su una raccolta eseguendo il codice una volta per elemento nella raccolta, fornendo il valore e / o la chiave dell'elemento come parametro.

Goto loop

`goto` cicli `goto` sono un insieme di istruzioni tra un'etichetta e un'istruzione `goto`.

Loop ricorsivi

Nella programmazione funzionale, i loop ricorsivi possono essere utilizzati per eseguire una funzione in modo ricorsivo finché non viene soddisfatta una condizione. Questa è una causa comune per errori di overflow dello stack.

Leggi Iniziare con i loop online: <https://riptutorial.com/it/loops/topic/5080/iniziare-con-i-loop>

Capitolo 2: Per i loop

Sintassi

- `for (init; condition; increment) {content_code (); } // sintassi generale`
- `for (int i = 0; i < numberRuns; ++ i) {actions_with (i); } // esegue un'azione per un numero di volte`
- `for (int i = 0; i < sizeof (array); ++ i) {actions_with (array [i]); } // iterazione su un array`

Examples

Generale per loop

La maggior parte dei linguaggi di programmazione supporta la struttura di controllo `for` -loop.

In genere è implementato in questo modo:

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

Lo pseudocodice precedente è identico allo pseudocodice seguente:

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

Questo dimostra che:

- `init` viene eseguito prima del ciclo, utilizzato per inizializzare le cose per l'esecuzione di quel ciclo
 - In alcuni linguaggi di programmazione come Java, le variabili possono essere *dichiarate* in `init` e l'ambito delle variabili dichiarate sarà limitato a quel ciclo.
- `condition` è una condizione per determinare quando è possibile eseguire il ciclo. Se questo è `false`, il ciclo smetterà di essere eseguito.
- `increment` è solitamente un'istruzione usata per manipolare i parametri utilizzati in `condition`, quindi quando l' `increment` viene eseguito un certo numero di volte, la `condition` diventa `false` e il ciclo si interrompe.
- `content_code()` è il codice principale da eseguire all'interno del ciclo.

Leggi Per i loop online: <https://riptutorial.com/it/loops/topic/6135/per-i-loop>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con i loop	Community , SOFe
2	Per i loop	SOFe