



EBook Gratuito

APPENDIMENTO

random

Free unaffiliated eBook created from
Stack Overflow contributors.

#random

Sommario

Di.....	1
Capitolo 1: Iniziare con random.....	2
Osservazioni.....	2
Examples.....	2
Installazione o configurazione.....	2
Fisher-Yates mescola.....	2
Titoli di coda.....	4

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

It is an unofficial and free random 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 random.

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 random

Osservazioni

Questa sezione fornisce una panoramica di cosa è casuale e perché uno sviluppatore potrebbe volerlo utilizzare.

Dovrebbe anche menzionare qualsiasi argomento di grandi dimensioni in modo casuale e collegarsi agli argomenti correlati. Poiché la documentazione per random è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

Examples

Installazione o configurazione

Istruzioni dettagliate su come impostare o installare casualmente.

Fisher-Yates mescola

Conosciuto anche come Knuth shuffle e il Durstenfeld-Fisher-Yates shuffle. Questo shuffle prende una serie di n elementi e lo meschia. L'algoritmo è veramente casuale in quanto, dopo aver mescolato, ogni permutazione dell'array è ugualmente probabile.

In java:

```
public static void shuffle(E[] deck) {  
  
    //From the end, swap each card with a random card from the unswapped portion.  
    for(int i = deck.length - 1; i > 0; i--)  
    {  
        //Pick an element from [0,i], inclusive.  
        int chosenCard = (int) (Math.random() * (i + 1));  
  
        E temp = deck[i];  
        deck[i] = deck[chosenCard];  
        deck[chosenCard] = temp;  
    }  
}
```

Nota: è necessario che l'elemento di sostituzione provenga da $[0, i]$ incluso e non $[0, i)$ esclusivo: altrimenti, le permutazioni dell'array in cui gli elementi rimangono sul posto sono impossibili, il che non è veramente casuale.

Supponendo che assumere numeri casuali prenda $O(1)$ per generare, l'algoritmo funziona sul posto e prende $O(n)$ tempo e spazio. Un array meschiato in questo modo può essere utilizzato per recuperare elementi non ripetitivi in tempo di ammortamento $O(1)$ per elemento.

```
E[] deck;
```

```
int drawIndex;

//Elements are taken from an index that advances.
public E drawUniqueCard()
{
    //Once all cards have been drawn, reshuffle the deck and draw from the top.
    if(drawIndex == deck.length)
    {
        shuffle(deck);
        drawIndex = 0;
    }
    //Pull the next card off the deck.
    return deck[drawIndex++];
}
```

Leggi Iniziare con random online: <https://riptutorial.com/it/random/topic/9484/iniziare-con-random>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con random	Community , Mauve Ranger