

 eBook Gratuit

APPRENEZ random

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#random

Table des matières

À propos	1
Chapitre 1: Commencer avec random	2
Remarques.....	2
Exemples.....	2
Installation ou configuration.....	2
Fisher-Yates mélange.....	2
Crédits	4

À propos

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

Chapitre 1: Commencer avec random

Remarques

Cette section fournit une vue d'ensemble de ce qu'est le hasard et de la raison pour laquelle un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les grands sujets au hasard et établir un lien avec les sujets connexes. La documentation de random étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

Exemples

Installation ou configuration

Instructions détaillées sur la configuration ou l'installation aléatoire.

Fisher-Yates mélange

Aussi connu sous le nom de Knuth Shuffle et le mélange de Durstenfeld-Fisher-Yates. Ce mélange prend un tableau de n éléments et le mélange. L'algorithme est vraiment aléatoire en ce sens que, après le brassage, chaque permutation du tableau est également probable.

En 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;  
    }  
}
```

Remarque: il est nécessaire que l'élément de remplacement vienne de $[0, i]$ inclus et non de $[0, i)$ exclusif: sinon, les permutations du tableau où les éléments restent en place sont impossibles, ce qui n'est pas vraiment aléatoire.

En supposant que des nombres aléatoires prennent $O(1)$ pour générer, l'algorithme fonctionne en place et prend $O(n)$ temps et espace. Un tableau mélangé de cette manière peut être utilisé pour récupérer des éléments non répétitifs dans $O(1)$ temps amorti par élément.

```
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++];
}
```

Lire Commencer avec random en ligne: <https://riptutorial.com/fr/random/topic/9484/commencer-avec-random>

Crédits

S. No	Chapitres	Contributeurs
1	Commencer avec random	Community , Mauve Ranger