



EBook Gratis

APRENDIZAJE random

Free unaffiliated eBook created from
Stack Overflow contributors.

#random

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con el azar.....	2
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
Fisher-Yates Shuffle.....	2
Creditos.....	4

Acerca de

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

Capítulo 1: Empezando con el azar

Observaciones

Esta sección proporciona una descripción general de qué es aleatorio y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro del azar, y vincular a los temas relacionados. Dado que la Documentación para aleatorios es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Instalación o configuración

Instrucciones detalladas sobre cómo configurar o instalar al azar.

Fisher-Yates Shuffle

También conocido como el shuffle de Knuth y el shuffle de Durstenfeld-Fisher-Yates. Este orden aleatorio toma una serie de n elementos y lo baraja. El algoritmo es verdaderamente aleatorio ya que, después de barajar, cada permutación de la matriz es igualmente 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;  
    }  
}
```

Tenga en cuenta: es necesario que el elemento de reemplazo venga de $[0, i]$ inclusive y no de $[0, i)$ exclusivo: de lo contrario, las permutaciones de la matriz donde los elementos permanecen en su lugar son imposibles, lo que no es realmente aleatorio.

Suponiendo que asumiendo que los números aleatorios toman $O(1)$ para generar, el algoritmo opera en su lugar y toma $O(n)$ tiempo y espacio. Un arreglo barajado de esta manera puede usarse para recuperar elementos no repetitivos en $O(1)$ tiempo amortizado por 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++];
}
```

Lea Empezando con el azar en línea: <https://riptutorial.com/es/random/topic/9484/empezando-con-el-azar>

Creditos

S. No	Capítulos	Contributors
1	Empezando con el azar	Community , Mauve Ranger