

## Generar números aleatorios en Java

Se puede hacer con:

- el método `Math.random()`. Observa que `random()` es un método de la clase `Math`
- un objeto de la clase `Random`

Crear objetos de la clase `Random` tiene más posibilidades, además como nosotros ahora mismo lo que estamos es observando una diversidad de ejemplos de objetos muy dispares, lo que nos interesa es usar la clase `Random`.

Ejercicio: localiza en el API el método `random()` de la clase `Math` y el método `nextInt()` de la clase `Random`.

Aunque les llamamos "aleatorios" lo correcto es "pseudoaleatorio" ya que no es posible imitar un proceso de generación de números 100% aleatorio, tan sólo podemos aproximarnos

## La clase `Random`. El método `nextInt()`

Observa que la clase `Random` tiene una variedad de métodos, a nosotros nos basta con manejar el método `nextInt()`.

Igual que ocurría con `Scanner`, debemos incorporar una sentencia `import` al principio del archivo.

`nextInt()` devuelve un número aleatorio en el rango de `int` (de -2,147,483,648 a +2,147,483,647)

Ejemplo:

```
import java.util.Random;
class Unidad2{
    public static void main(String[] args){
        Random numeroAleatorio= new Random();
        int aleatorio1= numeroAleatorio.nextInt();
        int aleatorio2=numeroAleatorio.nextInt();
        int aleatorio3=numeroAleatorio.nextInt();
        System.out.println("aleatorio1: "+aleatorio1);
        System.out.println("aleatorio2: "+aleatorio2);
        System.out.println("aleatorio3: "+aleatorio3);
    }
}
```

Ejecuta varias veces el programa y comprueba que ni dentro de cada ejecución ni entre ejecuciones se repite ningún número. Ya que al ser números aleatorios todos los números tienen la misma probabilidad de salir, un número `x` podría salir dos veces, pero en este rango la probabilidad es tan baja, que si te ocurriera debes salir pitando a comprarte un billete de lotería.

## Modificar el rango de valores de generación de números aleatorios.

En el API observa que hay una versión de `nextInt` a la que se le puede pasar como parámetro un `int`.

```
public int nextInt(int n)
```

Con esta versión, obtenemos un entero en el rango de 0 a `n-1` en lugar de en el rango rango de `int` (de -2,147,483,648 a +2,147,483,647)

Ejemplo:

```
import java.util.Random;
```

```
class Unidad2{
    public static void main(String[] args){
        Random numeroAleatorio= new Random();
        int jugada1= numeroAleatorio.nextInt(2);
        int jugada2=numeroAleatorio.nextInt(2);
        System.out.println("en la jugada1 salió "+ (jugada1==0?"cara":"cruz"));
        System.out.println("en la jugada2 salió "+ (jugada2==0?"cara":"cruz"));
    }
}
```

### **Ejercicio 1**

Escribe algo similar a la prueba anterior para obtener tiradas aleatorias de un dado de 6 caras. Como `nextInt(5)` genera números de 0 a 5 súmale 1 al valor que te de este método para obtener números aleatorios en el rango 1-6

### **Ejercicio 2**

Modifica el ejemplo de cara cruz para que interactúe con el usuario (que elija cara o cruz y le informamos si acertó o no). El usuario especifica con teclado su elección "cara" o "cruz". Te hará falta el método `equals()` de la clase `String`. Consulta el API para utilizarla en tu programa.