Clase 14 – Aleatoriedad (random)

Fundamentación

Biblioteca estándar de Python

Todos los conceptos que hemos visto hasta ahora los hemos resuelto utilizando solo la sintaxis que nos provee Python y un conjunto de funciones básicas que se incluyen automáticamente como por ejemplo son print, range, len etc.

En Python se incluye una biblioteca extra de funciones, variables, clases etc. que nos facilitan la resolución de problemas en una gran diversidad de áreas como matemáticas, estadísticas, compresión de datos, internet, interfaces visuales etc.

Veremos en este concepto como se importa un módulo de la biblioteca estándar y como se accede a su funcionalidad.

La **aleatoriedad** es esencial en programación cuando se quiere introducir el factor sorpresa. Se utiliza en juegos, simulaciones, experimentos y gráficos. En Python, la librería **random** nos permite generar números o elegir elementos al azar, lo que fomenta la creatividad y el pensamiento lógico en los proyectos de los estudiantes.

Objetivos

- Comprender el uso de la librería random.
- Generar números enteros, decimales y colores aleatorios.

¿Por qué usar la biblioteca random?

En programación, a veces necesitamos que **los resultados no sean siempre los mismos**, sino que cambien de manera impredecible. Ahí entra en juego la **aleatoriedad**.

Razones principales:

1. **Simular situaciones de la vida real**
    - El clima, el lanzamiento de un dado o una moneda, o el comportamiento de una multitud no son predecibles.
    - Con random podemos recrear estos fenómenos en un programa.
2. **Hacer juegos más interesantes**
    - Un juego en el que los enemigos siempre aparezcan en el mismo lugar sería aburrido.
    - Gracias a random, cada partida es diferente.
3. **Practicar pensamiento lógico**
    - Al usar random, los estudiantes deben **pensar en distintas posibilidades**, ya que nunca saben exactamente qué valor saldrá.
4. **Aplicaciones reales**
    - Sorteos y rifas digitales.
    - Simulación de experimentos científicos (por ejemplo, probabilidades).
    - Seguridad informática (contraseñas o números aleatorios para cifrado).

Marco teórico

- import random → importar la librería.
- random.randint(a, b) → devuelve un **entero aleatorio** entre *a* y *b*.
- random.choice(lista) → devuelve un **elemento aleatorio** de una lista.
- random.random() → devuelve un **decimal entre 0 y 1**.

Problema 1:

Confeccionar un programa que simule tirar dos dados y luego muestre los valores que salieron.

Imprimir un mensaje que ganó si la suma de los mismos es igual a 7.

Para resolver este problema requerimos un algoritmo para que se genere un valor aleatorio entre 1 y 6. Como la generación de valores aleatorios es un tema muy frecuente

la biblioteca estándar de Python incluye un módulo que nos resuelve la generación de valores aleatorios.

```
import random
dado1=random.randint(1,6)
dado2=random.randint(1,6)
print("Primer dado:",dado1)
print("Segundo dado:",dado2)
suma=dado1+dado2
if suma==7:
 print("Gano")
else:
 print("Perdio")
```

Para importar un módulo de la biblioteca estándar de Python utilizamos la palabra clave import seguida por el nombre del módulo que necesitamos importar:

import random

Para acceder a todas las funciones contenidas en el módulo random es necesario primero importar el módulo y luego dentro del algoritmo de nuestro programa anteceder primero el nombre del módulo y seguidamente la función que queremos acceder:

dado1=random.randint(1,6)

Si tratamos de acceder directamente al nombre de la función sin disponer el nombre del módulo se produce un error:

```
Traceback (most recent call last):
 File "C:/programaspython/ejercicio179.py", line 3, in
 dado1=randint(1,6)
NameError: name 'randint' is not defined
```

Este error se produce porque la función randint no es una función integrada en Python como print, range, len etc.

Entonces la sintaxis para acceder a la funcionalidad de un módulo requiere que dispongamos primero el nombre del módulo y seguidamente el nombre de la función.

Como podemos imaginar la función randint retorna un valor aleatorio comprendido entre los dos valores indicados en los parámetros.

Problema 2:

Desarrollar un programa que cargue una lista con 10 enteros.

Cargar los valores aleatorios con números enteros comprendidos entre 0 y 1000.

Mostrar la lista por pantalla.

Luego mezclar los elementos de la lista y volver a mostrarlo.

```
import random
def cargar():
        lista=[ ]
        for x in range(10):
                lista.append(random.randint(0,1000))
        return lista


def imprimir(lista):
        print(lista)
def mezclar(lista):
        random.shuffle(lista)


# bloque principal
lista=cargar()
print("Lista generada aleatoriamente")
imprimir(lista)
mezclar(lista)
print("La misma lista luego de mezclar")
imprimir(lista)
```

El módulo random cuenta con otra función llamada shuffle que le pasamos como parámetro una lista y nos la devuelve con los elementos mezclados (pensemos esto nos podría servir si estamos desarrollando un juego de naipes y necesitamos mezclarlos):

```
def mezclar(lista):
        random.shuffle(lista)
```

En la documentación oficial de Python podemos consultar todas las funciones que nos provee el módulo random https://docs.python.org/3/library/random.html

Y en general podemos también consultar todos los módulos de la Biblioteca estándar de Python https://docs.python.org/3/library/index.html

Actividades de guía

1. **Número aleatorio entre 1 y 10**

```
import random
numero = random.randint(1, 10)
print("Número generado:", numero)
```

2. **Elegir un color al azar de una lista**

```
colores = ["rojo", "azul", "verde", "amarillo", "violeta"]
color_azar = random.choice(colores)
print("Color elegido:", color_azar)
```

Actividad final

Opciones para elegir (según el interés de los alumnos):

1. **Juego de adivinanza**
    - El programa genera un número secreto entre 1 y 20.
    - El usuario tiene que adivinarlo en un máximo de 5 intentos.

```
import random

secreto = random.randint(1, 20)
intentos = 0
max_intentos = 5

print("Adivina el número secreto entre 1 y 20")

while intentos < max_intentos:
    numero = int(input("Tu intento: "))
    intentos += 1

    if numero == secreto:
        print("¡Adivinaste en", intentos, "intentos!")
        break
    elif numero < secreto:
        print("El número secreto es mayor.")
    else:
```

```
            print("El número secreto es menor.")

    if numero != secreto:
        print("Lo siento, el número era:", secreto)
```

Problemas propuestos

• Confeccionar un programa que genere un número aleatorio entre 1 y 100 y no se muestre.

El operador debe tratar de adivinar el número ingresado.

Cada vez que ingrese un número mostrar un mensaje "Gano" si es igual al generado o "El número aleatorio es mayor" o "El número aleatorio es menor".

Mostrar cuando gana el jugador cuantos intentos necesitó.

Confeccionar una programa con las siguientes funciones:

1) Generar una lista con 4 elementos enteros aleatorios comprendidos entre 1 y 3. Agregar un quinto elemento con un 1.

2) Controlar que el primer elemento de la lista sea un 1, en el caso que haya un 2 o 3 mezclar la lista y volver a controlar hasta que haya un 1.

3) Imprimir la lista.