

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.

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.

Programa: ejercicio179.py

[Ver video](#)

```
import random

dato1=random.randint(1,6)
dato2=random.randint(1,6)
print("Primer dado:",dato1)
print("Segundo dado:",dato2)
suma=dato1+dato2
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
```

Como dijimos la biblioteca estándar de Python se instala junto con Python.

Si disponemos un nombre de módulo inexistente aparecerá un error:

```
Traceback (most recent call last):  
  File "C:/programaspython/ejercicio179.py", line 1, in  
    import ran  
ImportError: No module named 'ran'
```

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:

```
dadol=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  
    dadol=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.

La ejecución del programa tiene una salida similar a esta:

The image shows a Python IDE window titled 'ejercicio179.py - C:/programasp/python/ejercicio179.py (3.8.0)'. The code in the editor is as follows:

```
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")
```

The status bar at the bottom right of the IDE shows 'Ln: 12 Col: 0'.

Below the IDE is a 'Python 3.8.0 Shell' window. It shows the execution of the script. The output is:

```
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/programasp/python/ejercicio179.py =====
Primer dado: 6
Segundo dado: 3
Perdio
>>> |
```

The status bar at the bottom right of the shell window shows 'Ln: 8 Col: 4'. The output lines 'Primer dado: 6', 'Segundo dado: 3', and 'Perdio' are highlighted with a red box.

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.

Programa: ejercicio180.py

[Ver video](#)

```
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)
```

No hay ningún problema de llamar a las funciones de un módulo dentro de una función propia siempre y cuando indiquemos el import respectivo:

```
import random

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

Desde el bloque principal procedemos a llamar a las funciones que hemos codificado:

```
# bloque principal

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

En la documentación oficial de Python podemos consultar todas las funciones que nos provee el módulo random [random](#).

Y en general podemos también consultar todos los módulos de la [Biblioteca estándar de Python](#).

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.
- [Ver video](#)
- 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.