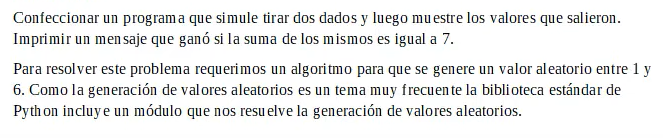
# **Biblioteca Estándar de Python**

Todos los conceptos que hemos visto hasta ahora lo hemos resuelto utilizando solo la sintaxis que nos provee Python y un conjunto de funciones básicas que se incluyen automáticamente como, por ejemplo: 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, estadística, comprensión 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.

**Problema1:**



import random

dado1=random.randint(1,6)

dado2=random.randint(1,6)

print("Primer dado:",dado1)

print("Segundo dato:",dado2)

suma=dado1+dado2

if suma==7:

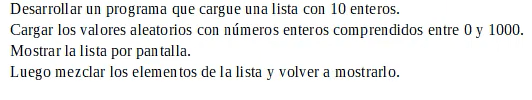
print("Gano")

else:

print("Perdio")

#Para poder usar funcionalidades de la biblioteca estándar, tenemos que importar El módulo desde donde se encuentra

**Problema2:**

****

import random

def cargar():

lista=[]

for x in range(10):

lista.append(random.randint(0,1000))

return lista

def imprimir():

print(lista)

def mezclar():

random.shuffle(lista)

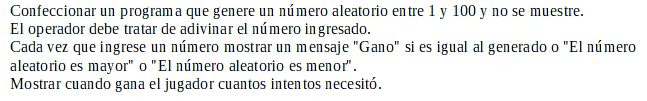
lista=cargar()

imprimir()

mezclar()

imprimir()

**Problema3:**

****

import random

print("Intente adivinar el numero que pense entre 1 y 100")

aleatorio=random.randint(1,100)

intentos=0

elegido=0

while elegido!=aleatorio:

elegido=int(input("Cual numero elige?: "))

if aleatorio>elegido:

print("Pense un numero mayor")

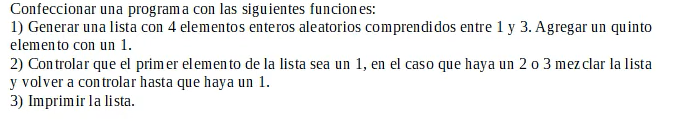
elif aleatorio<elegido:

print("Pense un numero menor")

intentos=intentos+1

print("Ganaste en",intentos,"intentos")

**Problema4:**

****

import random

def cargar():

lista=[]

for i in range(4):

lista.append(random.randint(1,3))

lista.append(1)

return lista

def controlar\_primero():

while lista[0]!=1:

random.shuffle(lista)

def imprimir():

print(lista)

#Bloque principal

lista=cargar()

imprimir()

controlar\_primero()

imprimir()

# **Importar algunas funcionalidades de un modulo de la biblioteca estándar de Python**

Hemos visto que para importar toda la funcionalidad de un modulo de la Biblioteca estándar de Python utilizamos la palabra clave import y seguidamente el nombre del módulo:

Import random

Con esa sintaxis todas las funcionalidades del módulo “random” pueden ser accedidas desde nuestro modulo.

Ahora veremos que en Python tenemos otra sintaxis para las situaciones que queremos acceder a una o pocas funcionalidades de un módulo.

Por ejemplo, si queremos acceder solo a la función randint del modulo random en Python lo podemos expresar con la siguiente sintaxis:

From random import randint

Utilizamos la palabra clave from y seguidamente el nombre del modulo de donde queremos importar funcionalidades del mismo. Luego indicamos la palabra clave import y la funcionalidad que queremos importar, en nuestro ejemplo la función randint.

También cambia como utilizamos la función randint dentro de nuestro modulo:

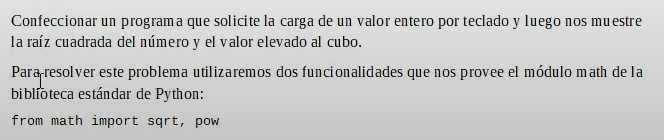
Valor =randint(1,10)  
print(valor)

Como vemos no le antecedemos ningún nombre de modulo y hacemos referencia directamente a la función importada.

Si necesitamos importar mas de una funcionalidad de un modulo debemos separar por comas las funcionalidades importadas:

From random import randint, shuffle

**Problema1:**



**Primera Forma:**

from math import sqrt, pow

def cargar():

valor=int(input("Ingresa un valor: "))

print(sqrt(valor))

print(pow(valor,3))

resultado=cargar()

**Segunda Forma:**

import math

def cargar():

valor=int(input("Ingresa un valor: "))

resultado1=math.sqrt(valor)

resultado2=math.pow(valor,3)

print(resultado1,resultado2)

resultado=cargar()

# **Definición de alias para una funcionalidad**

Podemos definir un nombre distinto para una funcionalidad que importamos de otro modulo. Esto puede tener como objetivo que nuestro programa sea más legible o evitar que un nombre de función que importamos colisione con un nombre de función de nuestro propio modulo.

Resolveremos el mismo problema anterior, pero definiendo dos alias para las funciones sqrt y pow del modulo math.

**Problema1:**

****

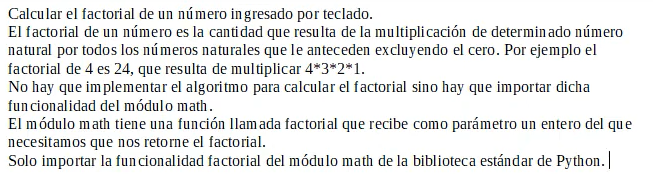
from math import sqrt as raiz\_cuadrada, pow as elevar

valor = int(input("Ingrese un valor entero: "))

print("La raiz cuadrada",raiz\_cuadrada(valor))

print("El cubo",elevar(valor,3))

**Problema2:**

****

from math import factorial as XD

n=5

resultado=XD(n)

print(resultado)

# **Aplicaciones propias con varios módulos**

A medida que una aplicación sea más grande vimos que disponer todo el algoritmo en forma secuencial es difícil de mantener y desarrollar. Para eso vimos que podemos resolver por partes utilizando funciones.

Pero siempre dispusimos todas las funciones en un único archivo.

Pensemos que nuestro problema podría ser lo suficientemente complejo y grande que también por mas que lo dividamos en funciones un único archivo sea complejo de mantener.

En Python podemos agrupar funciones en distintos archivos (módulos) y luego importarlos en otros módulos.

Así como importamos los módulos de la biblioteca estándar de Python podemos importar otros módulos que codifiquemos nosotros.

**Problema1:**

Confeccionar una aplicación que permita cargar por teclado una lista de 5 enteros, obtener y mostrar el mayor y calcular su suma. Definir un modulo con las funciones de cargar, identificar el mayor y suma. En el modulo principal del programa importar el otro modulo y llamar sus funciones.

Para ser mas ordenados crearemos una carpeta llamada proyecto1 y dentro de la misma crearemos los dos módulos llamados:

Operacioneslista.py

Principal.py

El modulo operacioneslista.py contiene todas las funciones que nos permiten cargar una lista, imprimir el mayor de una lista y sumar todos los elementos y mostrar dicho valor.

**Desarrollo:**

**Modulo local:** operacioneslistas

def cargar():

lista=[]

for i in range(5):

valor=int(input("Ingrese un valor: "))

lista.append(valor)

return lista

def imprimir\_mayor(lista):

mayor=lista[0]

for i in range(1,5):

if lista[i]>mayor:

mayor=lista[i]

print("Mayor elemento de la lista:",mayor)

def imprimir\_suma(lista):

suma=0

for i in lista:

suma+=i

print("La suma de sus elementos es: ",suma)

**Modulo principal:**

import operacioneslistas

lista=operacioneslistas.cargar()

operacioneslistas.imprimir\_mayor(lista)

operacioneslistas.imprimir\_suma(lista)

**Problema2:**

Confeccionar un método que implemente dos funciones, una que retorne el mayor de dos enteros y otro que retorno el menor de dos enteros.

En el modulo principal importar solo la función que retorne el mayor, luego cargar dos enteros y mostrar el mayor de ellos.

Crear una carpeta llamada proyecto2 y dentro de la misma crear dos módulos llamados: