GOBIERNO DE
EL SALVADOR

SECRETARÍA DE
INNOVACIÓN

CIENCIA DE DATOS

Agenda
Sesión 3/18

Bibliotecas estadísticas y numéricas.

- ¿Qué es una biblioteca?
- ¿Como crear una biblioteca?
- ¿Como se llama una biblioteca?
- Bibliotecas estadísticas y numéricas
    - Biblioteca **math**
    - Biblioteca **random**
    - Biblioteca **statistics**
    - Biblioteca **datetime**
- Pip como gestor de paquétes

¿Qué es una biblioteca?

Considere que un módulo es lo mismo que una biblioteca de código.

Un archivo que contiene un conjunto de funciones que desea incluir en su aplicación.

Ejemplos:

Pygame

Math

Json

¿Como crear una biblioteca?

Para crear un módulo, simplemente guarde el código que desea en un archivo con la extensión de archivo .py

Ejemplo

Guarde este código en un archivo llamado mymodule.py

```
def greeting(name):
  print("Hello, " + name)
```

¿Como se llama una biblioteca?

Ahora podemos usar el módulo que acabamos de crear, usando la declaración **import**:

```
import mymodule
mymodule.greeting(“Manuel Calderón")
```

Nota: cuando utilice una función de un módulo, utilice la sintaxis:

nombre-módulo.nombre-función

¿Como se llama una biblioteca?

Variables en el módulo

El módulo puede contener funciones, como ya se describió, pero también variables de todo tipo (matrices, diccionarios, objetos, etc.):

Guarde este código en el archivo **mymodule.py**

```
person1 = {
  "name": "John",
  "age": 36,
  "country": "Norway"
}
```

¿Como se llama una biblioteca?

Importe el módulo llamado mymodule y acceda al diccionario person1:

```
import mymodule

a = mymodule.person1["age"]
print(a)
```

Módulos numéricos y matemáticos

Los módulos descritos en este capítulo proporcionan funciones y tipos de datos numéricos y relacionados con las matemáticas. El módulo **numbers** define una jerarquía abstracta de tipos numéricos. Los módulos **math** y **cmath** contienen varias funciones matemáticas para números complejos y de punto flotante.

El módulo decimal admite representaciones exactas de números decimales, utilizando aritmética de precisión arbitraria.

Biblioteca math

Python tiene un conjunto de funciones matemáticas integradas, incluido un módulo matemático extenso, que le permite realizar tareas matemáticas con números.

Funciones matemáticas integradas

Las funciones **min()** y **max()** se pueden usar para encontrar el valor más bajo o más alto en un iterable:

```
x = min(5, 10, 25)
y = max(5, 10, 25)
print(x)
print(y)
```

Biblioteca math

La función **abs()** devuelve el valor absoluto (positivo) del número especificado:

```
x = abs(-7.25)
print(x)
```

La función devuelve el valor de x elevado a y (x^y). **pow(x, y)**

Devuelve el valor de 4 a la potencia de 3 (igual que 4 * 4 * 4):

```
x = pow(4, 3)
print(x)
```

Biblioteca math

Python también tiene un módulo incorporado llamado math, que amplía la lista de funciones matemáticas.

Para usarlo, debes importar el módulo math:

```
import math
```

Cuando haya importado el mathmódulo, puede comenzar a usar métodos y constantes del módulo.

El método **math.sqrt()**, por ejemplo, devuelve la raíz cuadrada de un número:

Biblioteca math

```
import math
x = math.sqrt(64)
print(x)
```

El método math.ceil() redondea un número hacia arriba a su entero más cercano, y el método math.floor() redondea un número hacia abajo a su entero más cercano y devuelve el resultado:

```
import math

x = math.ceil(1.4) # returns 2
y = math.floor(1.4) # returns 1
```

Biblioteca math

La math.piconstante, devuelve el valor de PI (3,14 ...):

```
import math
x = math.pi
print(x)
```

Biblioteca random

Python tiene un módulo incorporado que puede usar para hacer números aleatorios.

El módulo random tiene un conjunto de métodos:

- **random()**

método devuelve un número flotante aleatorio entre 0 y 1.

- **randint()**

método devuelve un elemento seleccionado de número entero del rango especificado

- **randrange()**

método devuelve un elemento seleccionado aleatoriamente del rango especificado

Biblioteca random

```
import random

print(random.random())
print(random.randint(3, 9))
print(random.randrange(3, 9))


mylist = ["apple", "banana", "cherry"]
random.shuffle(mylist)

print(mylist)
```

Biblioteca statistics

Este módulo proporciona funciones para calcular estadísticas matemáticas de datos numéricos

Este módulo no pretende ser competidor o sustituto de bibliotecas de terceros como NumPy o SciPy, ni de paquetes completos de software propietario para profesionales como Minitab, SAS o Matlab. Este módulo se ubica a nivel de calculadoras científicas gráficas.

Estas funciones calculan el promedio o el valor típico de una población o muestra.

Biblioteca random

mean()

Media aritmética («promedio») de los datos.

median()

Mediana (valor central) de los datos.

mode()

Moda única (valor más común) de datos discretos o nominales.

Biblioteca random

```
# Import statistics Library
import statistics

# Calculate average values
print(statistics.mean([1, 3, 5, 7, 9, 11, 13]))
print(statistics.mean([1, 3, 5, 7, 9, 11]))
print(statistics.mean([-11, 5.5, -3.4, 7.1, -9, 22]))
```

Biblioteca random

```
# Import statistics Library
import statistics

# Calculate middle values
print(statistics.median([1, 3, 5, 7, 9, 11, 13]))
print(statistics.median([1, 3, 5, 7, 9, 11]))
print(statistics.median([-11, 5.5, -3.4, 7.1, -9, 22]))
```

Biblioteca random

```
# Import statistics Library
import statistics

# Calculate the mode
print(statistics.mode([1, 3, 3, 3, 5, 7, 7 9, 11]))
print(statistics.mode([1, 1, 3, -5, 7, -9, 11]))
print(statistics.mode(['red', 'green', 'blue', 'red']))
```

Biblioteca datetime

Una fecha en Python no es un tipo de datos en sí misma, pero podemos importar un módulo nombrado **datetime** para trabajar con fechas como objetos de fecha.

```
import datetime
x = datetime.datetime.now()
print(x)
```

Salida de fecha

Cuando ejecutamos el código del ejemplo anterior, el resultado será:

2021-07-13 16:48:30.507132

Biblioteca datetime

Una fecha en Python no es un tipo de datos en sí misma, pero podemos importar un módulo nombrado **datetime** para trabajar con fechas como objetos de fecha.

```
import datetime
x = datetime.datetime.now()
print(x)
```

Salida de fecha

Cuando ejecutamos el código del ejemplo anterior, el resultado será:

2021-07-13 16:48:30.507132

Biblioteca datetime

La fecha contiene año, mes, día, hora, minuto, segundo y microsegundo.

El módulo datetime tiene muchos métodos para devolver información sobre el objeto de fecha.

Aquí hay algunos ejemplos, aprenderá más sobre ellos más adelante en este capítulo:

```
import datetime
x = datetime.datetime.now()
print(x.year)
print(x.strftime("%A"))
```

Biblioteca datetime

Creación de objetos de fecha

Para crear una fecha, podemos usar la clase datetime() (constructor) del módulo datetime.

La clase datetime() requiere tres parámetros para crear una fecha: año, mes, día.

```
import datetime
x = datetime.datetime(2020, 5, 17)
print(x)
```

Pip como gestor de paquétes

¿Qué es PIP?

PIP es un administrador de paquetes para paquetes de Python, o módulos si lo desea.

¿Qué es un paquete?

Un paquete contiene todos los archivos necesarios para un módulo.

Los módulos son bibliotecas de código Python que puede incluir en su proyecto.

Pip como gestor de paquétes

Compruebe si PIP está instalado

Navegue por su línea de comando hasta la ubicación del directorio de secuencias de comandos de Python y escriba lo siguiente:

```
pip --version
```

Listar paquetes

Use el comando list para listar todos los paquetes instalados en su sistema:

```
pip list
```

RESUMEN DE SESIÓN

GOBIERNO DE
EL SALVADOR

SECRETARÍA DE
INNOVACIÓN