?

El equipo docente subió contenido extra del encuentro 🛠

# Funciones para no reinventar la rueda

"A complex system that works is invariably found to have evolved from a simple system that worked" -John Gall.

### Cuando Python no alcanza, aparecen las librerías Ahora que ya dimos nuestros primeros pasos en el mundo de la programación,

es hora de que nos interioricemos cada vez más para comprender cómo sacarle mayor provecho a esta herramienta que, como todo, tiene sus limitaciones. A veces, las estructuras de datos que vienen con Python —y sus funcionalidades

asociadas— no son suficientes para lo que queremos hacer. En el desafío de la

bitácora anterior te propusimos que tomes una lista de números y le sumes un

valor a ellos. Por las dudas, te recordamos la consigna y dejamos una resolución

acá. Habrás notado que la solución a ese problema no es ni muy interpretable, ni muy cómoda. Y, además, si evaluáramos su desempeño —el tiempo que lleva a la computadora hacer esa operación— veríamos que no es muy eficiente. ¿Habrá una forma de hacerlo mejor?

Por suerte, existen *complementos* que podemos sumarle a nuestro lenguaje de programación que se llaman librerías o bibliotecas. Una librería es, en el fondo, funcionalidades que otras personas programaron y que nosotros sumamos a nuestro espacio de trabajo para aprovecharlas. Siguiendo con el ejemplo, si una persona programó una forma muy eficiente de sumarle un número a todos los

elementos de una lista —o a algo que se parezca mucho a una lista—, esa

persona puede compartir ese código en forma de librería.

todo. A lo largo de la carrera, iremos viendo distintas librerías que nos permitirán: • Trabajar con datos de manera eficiente (NumPy, Pandas),

Las librerías abren el panorama de una manera abismal. Existen librerías para

• Implementar herramientas de Aprendizaje Automático (Scikit-Learn)

paquetes que puedes usar para resolver necesidades específicas.

• Graficar (Matplotlib, Seaborn),

Nuevamente, la existencia de librerías no es algo que sólo ocurre en Python. Casi todos los lenguajes de programación cuentan con complementos y

Nuestra primera librería: NumPy

Python), la librería por defecto para hacer operaciones numéricas de manera

En este encuentro trabajaremos con NumPy (proveniente de Numerical

### eficiente en Python. Como muchas librerías, NumPy viene con una estructura

de datos propia: los arrays o arreglos. Un arreglo, a primer orden, es como una lista. De hecho, se pueden crear a partir de una lista. Pero un diferencia crucial es que, mientras las listas pueden contener dentro cualquier tipo de dato básico, los arreglos funcionan mejor cuando lo que contiene son solamente números. De hecho, de esta forma NumPy nos permitirá hacer operaciones matemáticas en una gran cantidad de números de la forma que esperamos. En el notebook de práctica ya quedará claro a qué nos referimos con eso. Otras características de NumPy son: • Muchas librerías tienen, en el corazón de sus operaciones, a NumPy.

• Cuenta con muy buena documentación. Recomendamos siempre consultarla.

- Más adelante veremos en detalle cómo se arman estos objetos nuevos que no
- vienen nativamente en Python, sino que los traemos con una librería. Pero lo importante es tener en mente que, en general, las librerías incorporan nuevas estructuras de datos u objetos, que estos objetos satisfacen una necesidad

específica, y que vienen con funcionalidades propias que, en general, están muy bien hechas y nos ahorran mucho tiempo de trabajo si tuviéramos que programarlas nosotros/as desde cero. Antes de comenzar a trabajar en el notebook, veamos brevemente qué son las operaciones lógicas.

Un poco más de programación: Operaciones lógicas

en programación y que usaremos mucho a lo largo de la carrera. No

Las operaciones lógicas son aquellas que dan como resultado un

Representan un tipo muy importante de operación que aparece frecuentemente

### pretendemos en esta sección hacer una exposición exhaustiva ni rigurosa de ellas, pero es importante que veamos algunos conceptos básicos.

booleano. Recordemos que un booleano es un tipo de dato básico que puede tomar dos valores, Verdadero (True) o Falso (False). En general, los usamos como condiciones para tomar decisiones en el flujo de programación. Supongamos que queremos elegir entre dos opciones, A ó B.

Esta es la base de las estructuras de decisión de control (if, elif y else) que veremos en el notebook de hoy.

• Si no se cumple una condición (False), elegimos B.

• Si se cumple una condición (True), elegimos A;

necesitamos de las operaciones lógicas.

condiciones originales (A,B) sean verdaderas.

necesitamos la operación OR.

**False** 

**False** 

Α

**False** 

**False** 

True

de verdad" para cada una de estas operaciones: A AND B В Α

**False** 

True

В

**False** 

¿Cómo hacemos si queremos combinar más de una condición? Para ello,

• Si queremos que se cumplan dos condiciones, es decir la condición A y la

condición B, necesitamos la operación AND, tal como lo muestra la "tabla

**False** 

**False** 

A OR B

**False** 

	True	False	False	
	True	True	True	
Pedir	que se cumplan dos co	ndiciones, A y B, da por	resultado una nueva	
condi	ción, A y B. Esta condic	ión será verdadera únic	camente cuando las dos	

• Si, en cambio, queremos que se cumpla la condición A ó la condición B,

**False** True True **False** True True True True True

Pedir que se cumplan una condición u otra, A o B, da por resultado una nueva

Δ	Not A			
• Para finalizar, dejamos una tabla de verdad sencilla, la negación.				
dos condiciones originales (A,B) sea verdadera.				
condición, A or B. Esta condición será verdadera siempre y cuando alguna de la				

True

**False** 

utilizaremos con NumPy. Empieza a trabajar en el notebook

Las operaciones lógicas son parte de todos los lenguajes de programación. En el

notebook veremos cómo implementar lo que vimos en Python y, luego, cómo lo

### 1. Instala NumPy Estas instrucciones te servirán para instalar NumPy y, prácticamente, cualquier

Challenge: empieza a trabajar en el notebook

**Instala NumPy** 

computadoras. Para el encuentro, **lleva resuelta la Sección 1**:

Notebook: DS\_Bitácora\_03\_NumPy

#### 2. Empieza a resolver el notebook Una vez que esté instalado, trabaja en este notebook, ya sea en Colab o en sus

librería:

"Programación", el resto del notebook lo trabajaremos en clase.

# ¿Quieres saber más? **Profundiza**

Te invitamos a conocer más sobre el tema de esta bitácora.



contenido necesario para sobrevivir.

## Contenido extra del encuentro



DS-ONLINE-65 2/2/21 12/8/21 Mar/Jue Tema: DS-ONLINE-65 2/2/21 12/8/21 Mar/Jue Hora de inicio de la reunión : 9 feb. 2021 07:01 Grabación de la reunión: https://acamica.zoom.us/rec/share/fM\_vjAn4A41BX-CmLKhHt5BX8QTmBG7SXL3UHfAxqKD8Lq-S00spzEl6-fciDJNr.LJYkVkb0tVox5Gue Código de acceso

اننزز Matemática??!! No te asustes... aquí tienes herramientas para revisar el

#### Profundiza

Lista de recursos

Glosario

Rincón Matemático Notebook: DS\_Bitácora\_03\_NumPy

### DS-ONLINE-65 2/2/21

Contenido extra

12/8/21 Mar/Jue