



In collaboration with
IBM SkillsBuild

IBM SkillsBuild y SkillUp Online presentan:

Programa de inteligencia artificial



Sesión

Fundamentos de Inteligencia Artificial y Python para IA

HUGO RAMALLO



In collaboration with
IBM SkillsBuild

Programa

Introducción a la IA

- Introducción a la IA
- Aplicaciones y salidas profesionales de la IA
- Por qué Python para la IA
- Bibliotecas Python para IA
- Fundamentos de Python: práctica con variables y tipos de datos
- Práctica: Listas, tuplas y diccionarios
- Práctica: Escribir comandos básicos en Python para realizar operaciones aritméticas, manipular series y asignar variables

Fundamentos de la IA

DEFINICIÓN

Disciplina científica que se ocupa de crear programas informáticos que ejecutan operaciones *comparables a las que realiza la mente humana*, como el aprendizaje o el razonamiento lógico. RAE



ALAN TURING
1950

ÁREAS DE LA IA



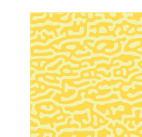
MACHINE LEARNING



NLP



SISTEMAS EXPERTOS



STT



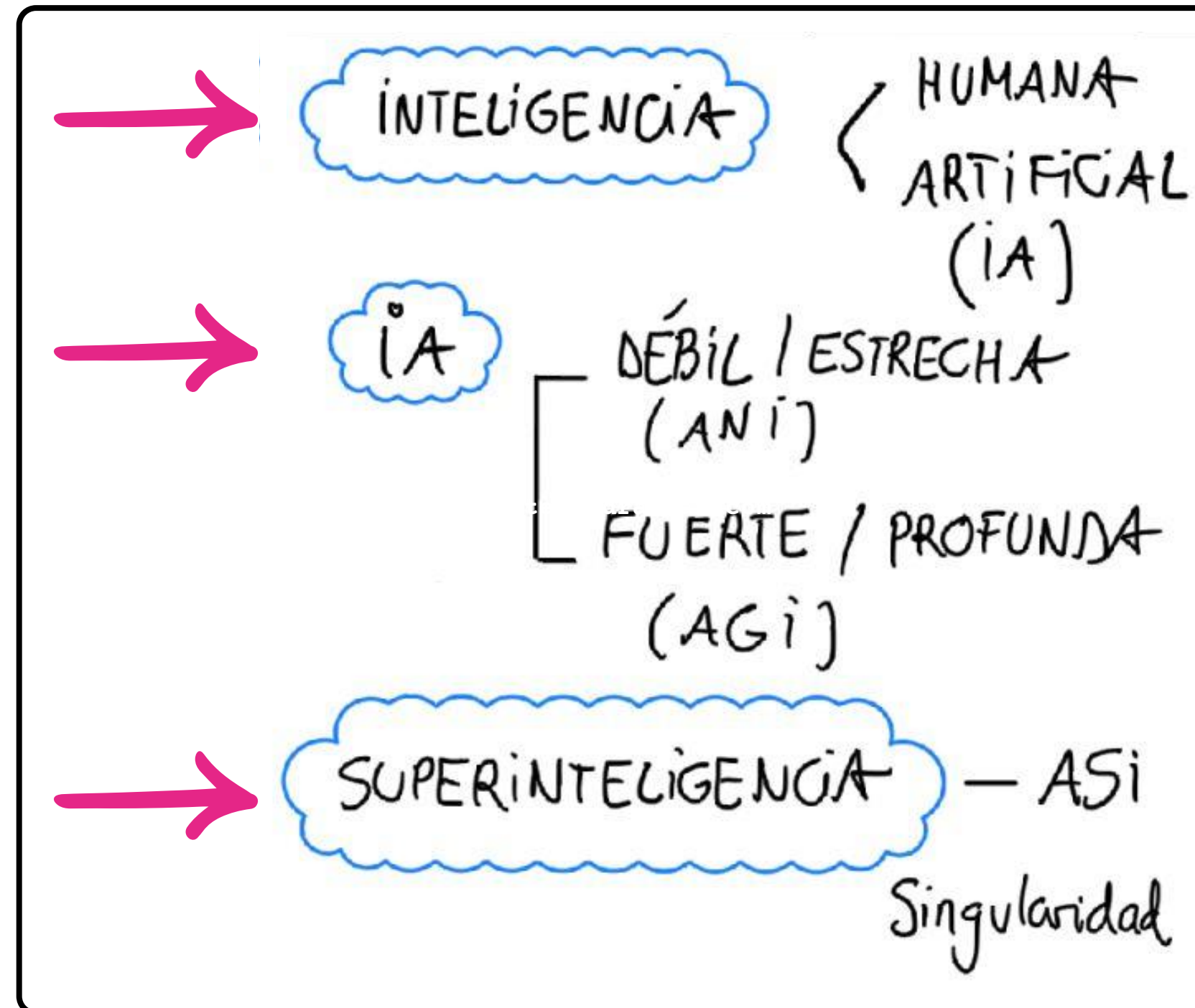
VISIÓN ARTIFICIAL



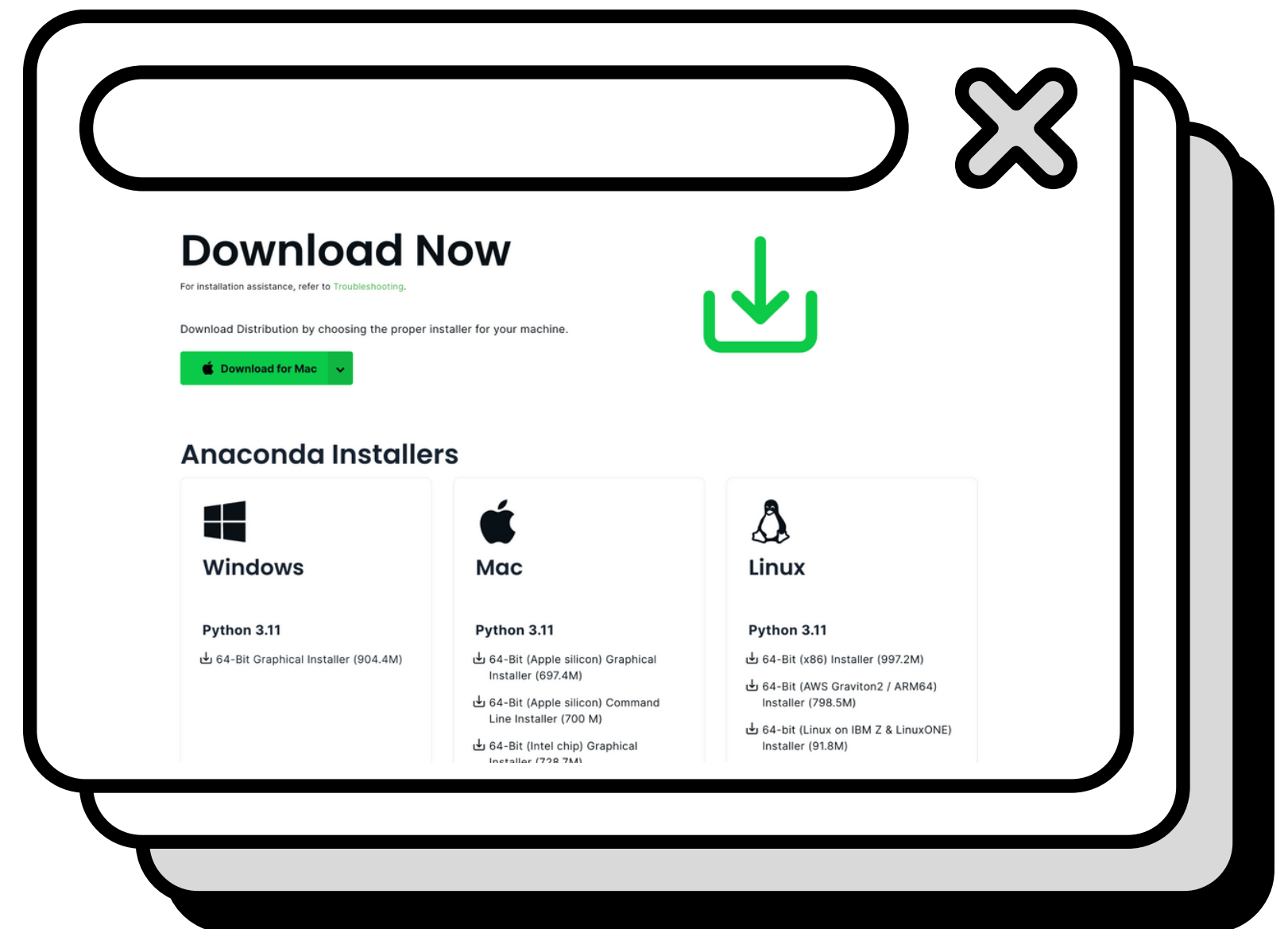
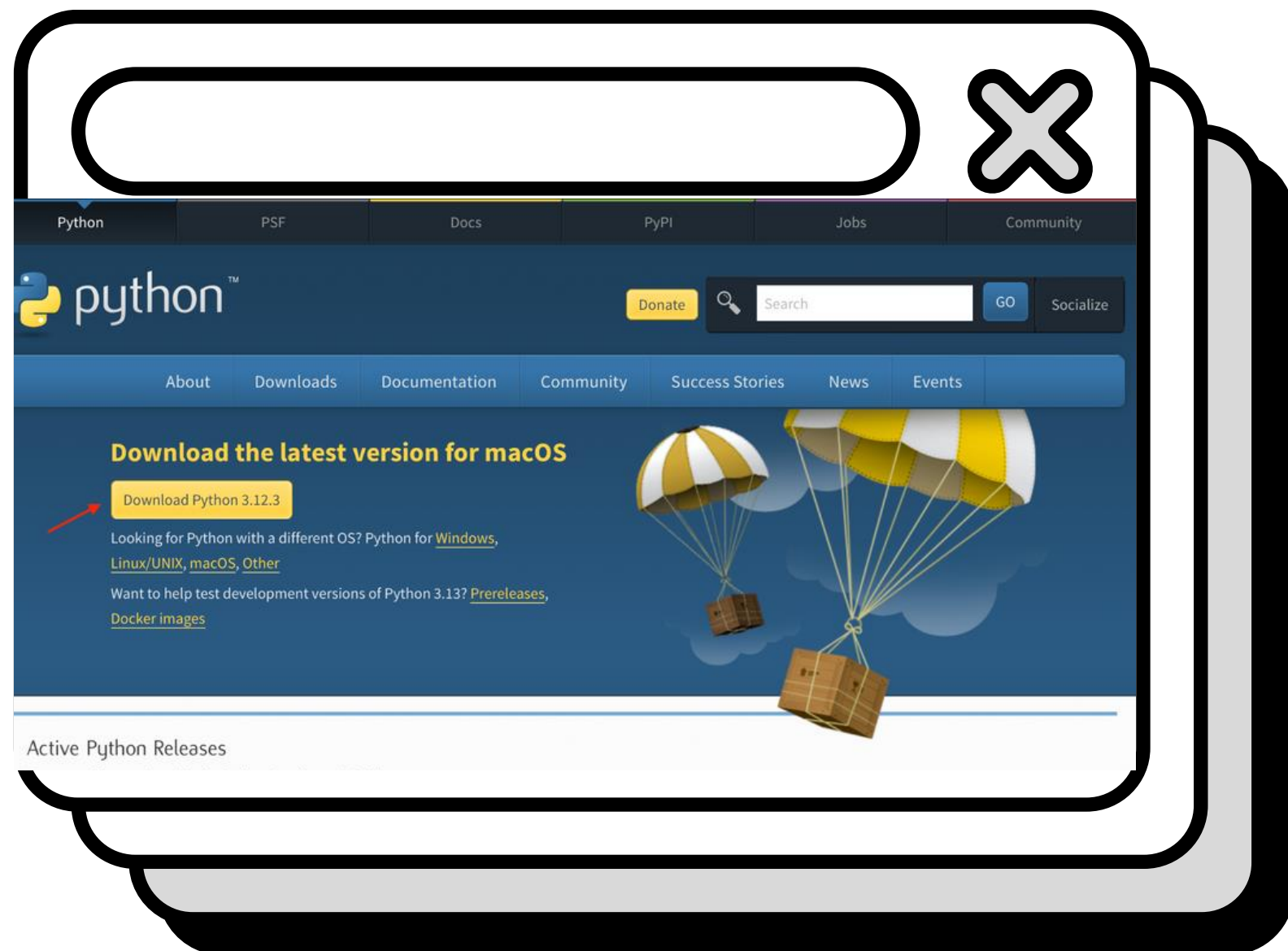
ROBÓTICA

Fundamentos de la IA

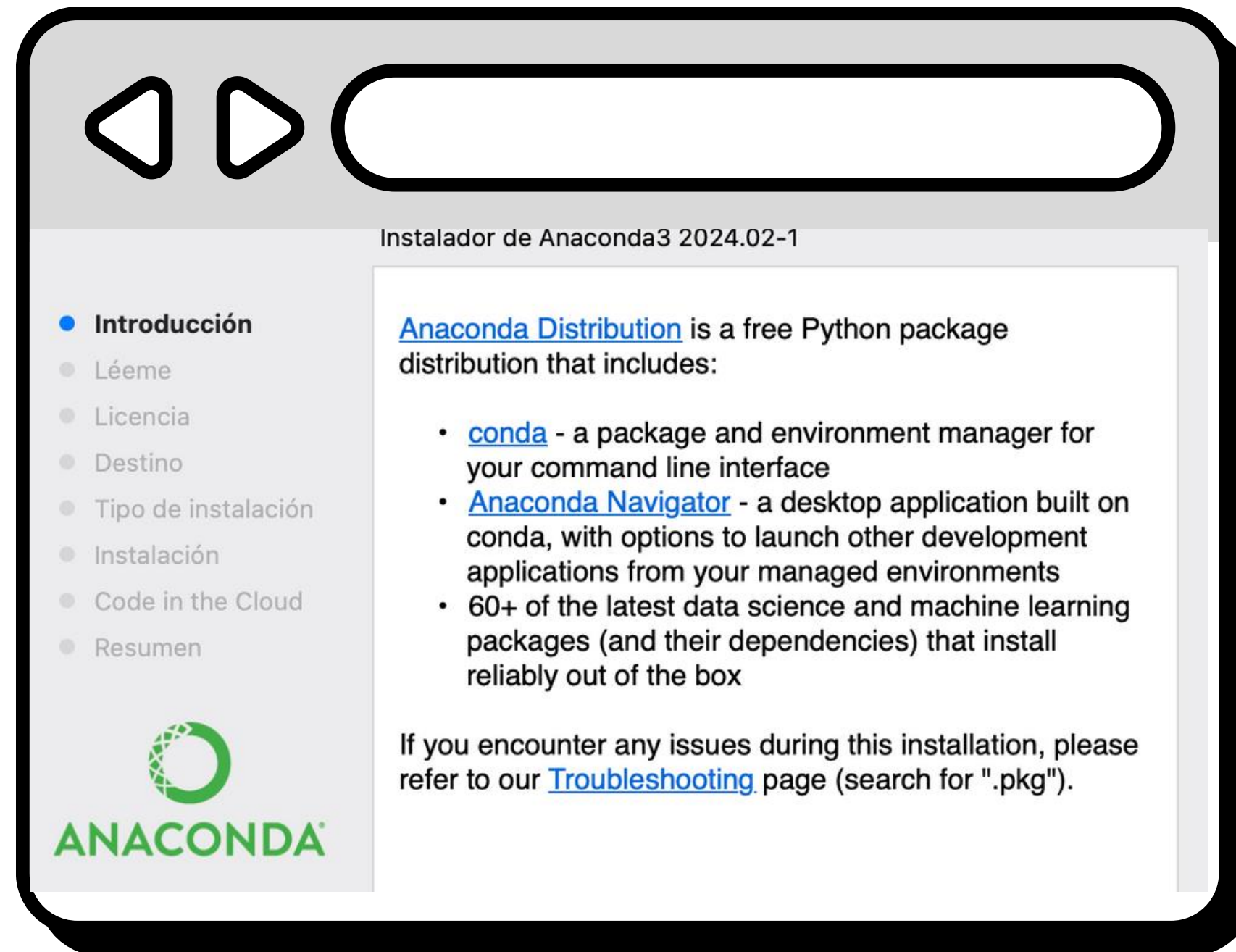
TIPOS DE IA



PYTHON – ANACONDA



Instalación ANACONDA



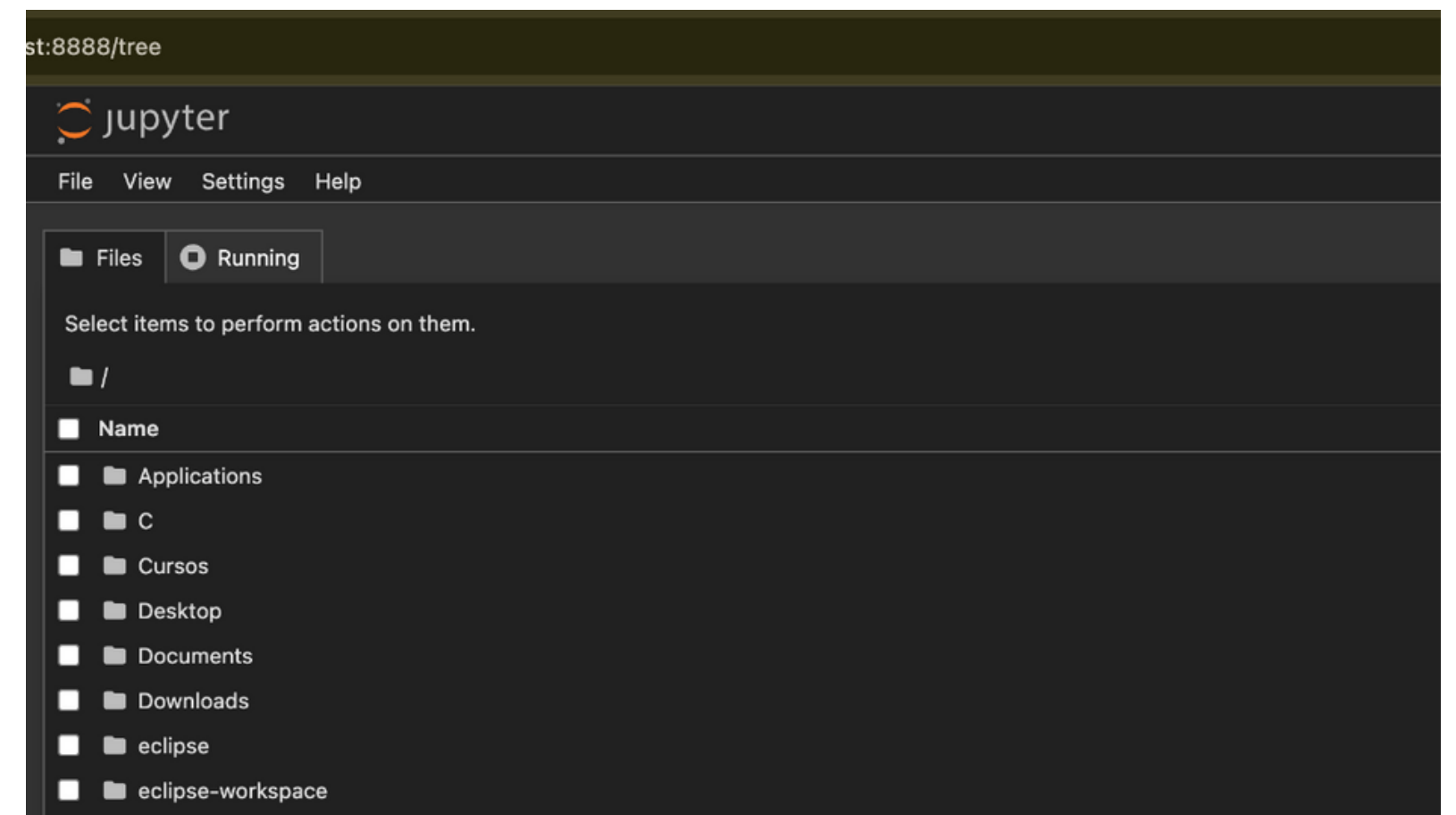
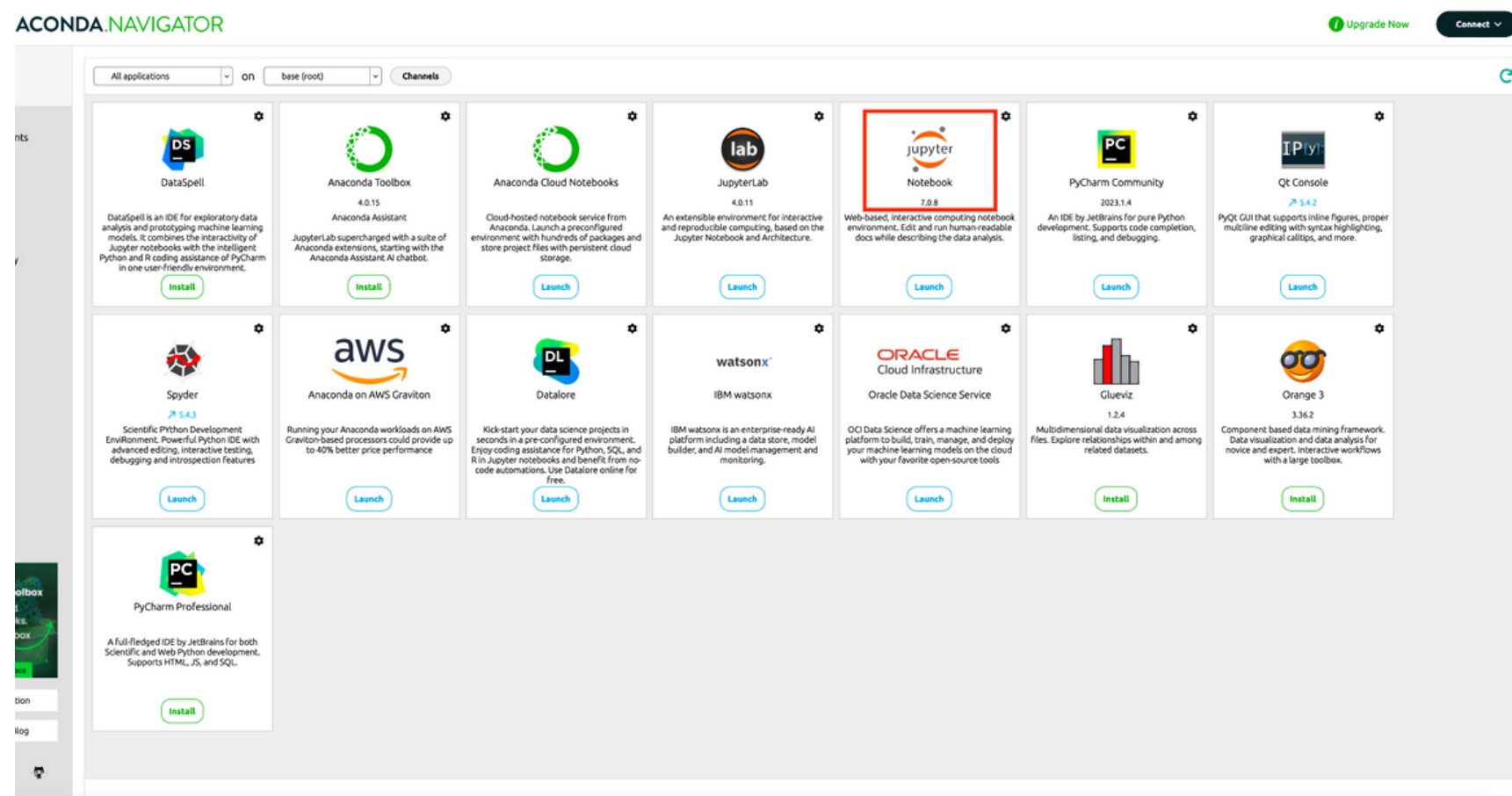
Una vez descargado el paquete Anaconda, el siguiente paso será proceder con la instalación.



JUPYTER NOTEBOOK

● Al ejecutar el Anaconda-Navigator podremos ver la siguiente ventana

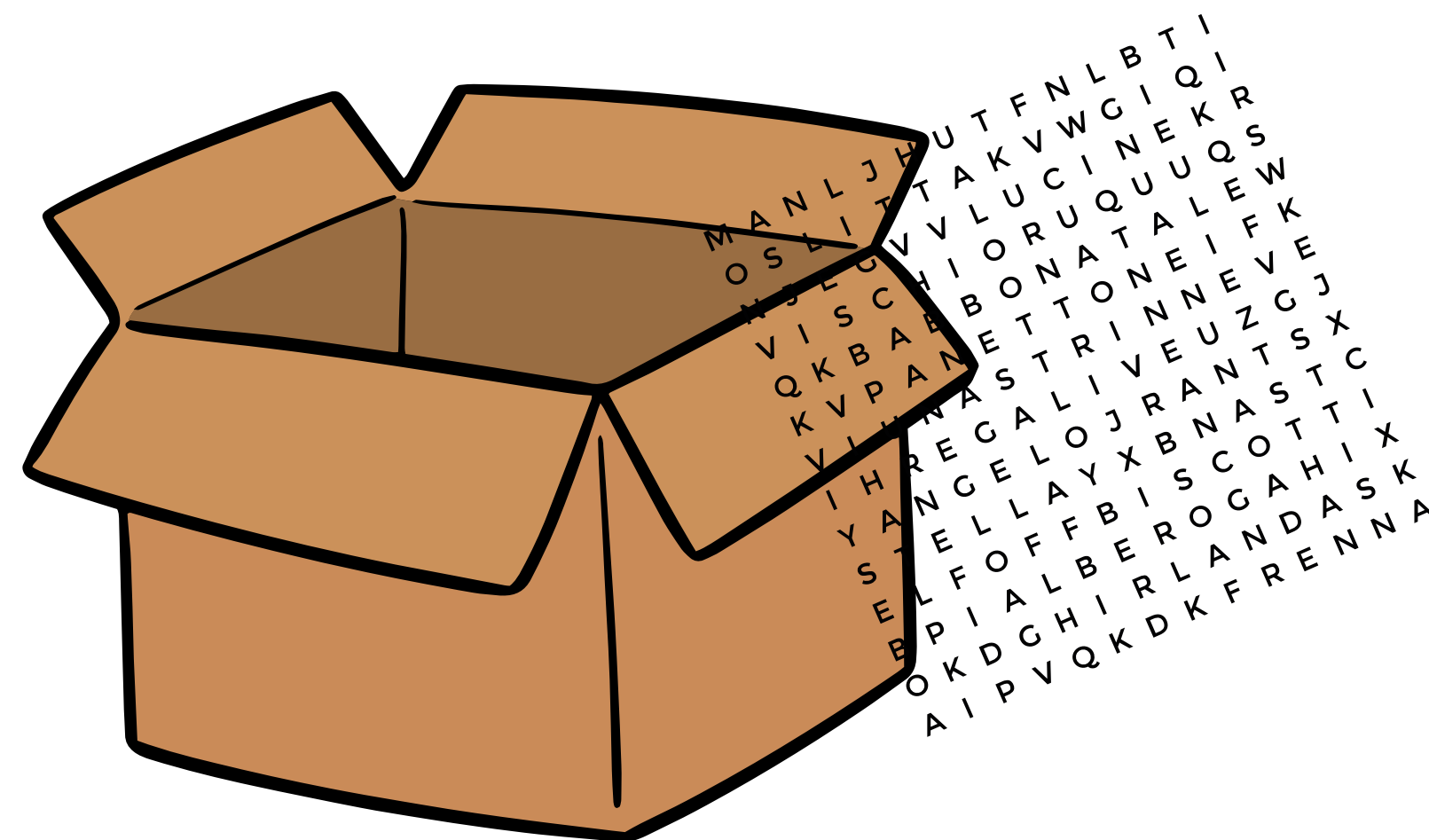
Ejecutar Jupyter Notebooks

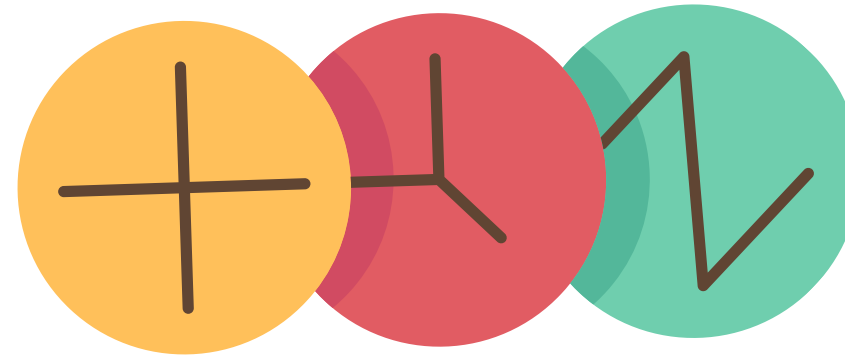


VARIABLES Y TIPOS DE DATOS

Variables

Tipos de datos





Variables

Una variable en Python es un nombre que se asocia con un valor almacenado en la memoria. Las variables se utilizan para almacenar datos que pueden ser manipulados y utilizados a lo largo de un programa.

A = 10

TIPOS DE DATOS

NÚMEROS (ENTEROS, REALES,
COMPLEJOS)

CADENAS

BOOLEANOS

***LISTAS**

***TUPLAS**

***DICCIONARIOS**

SETS

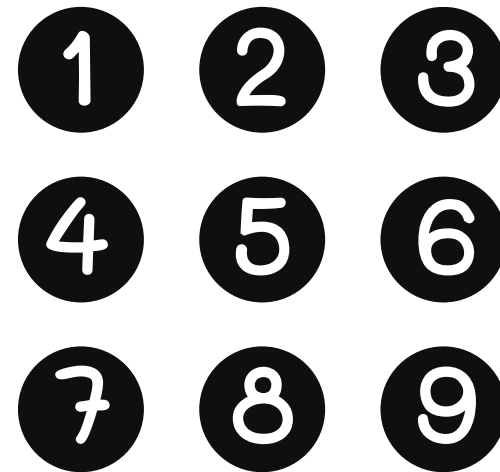




Tipos de datos

⇒ Los tipos de datos que se manejan en Python son básicamente: números (enteros, reales y complejos), strings (cadenas), booleanos, listas, tuplas, diccionarios, conjuntos(sets).

En Python no hace falta definir el tipo de dato (ya lo detecta Python), como si es necesario en otros lenguajes de programación, por ejemplo en Java, donde si hay que especificar si es un tipo entero (int), flotante (float), flotante extendido (double), booleano(boolean)



Enteros y flotantes

En Python, los números enteros (int) son valores sin decimales que pueden ser positivos, negativos o cero y tienen un rango prácticamente ilimitado. Los números reales (float) son valores con decimales que también pueden ser positivos, negativos o cero y se utilizan para representar números fraccionarios.

Entero = 5

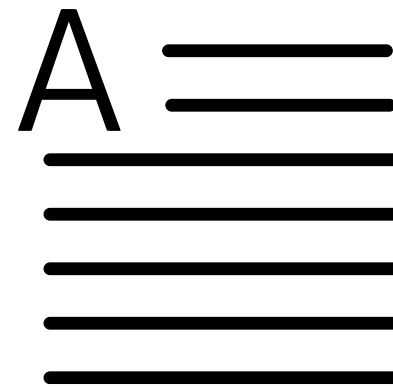
Flotante = 2,5

Cadenas

Una cadena, o string, en Python es una secuencia de caracteres encerrada entre comillas simples (' ') o comillas dobles (" "). Las cadenas son utilizadas para almacenar y manipular texto.

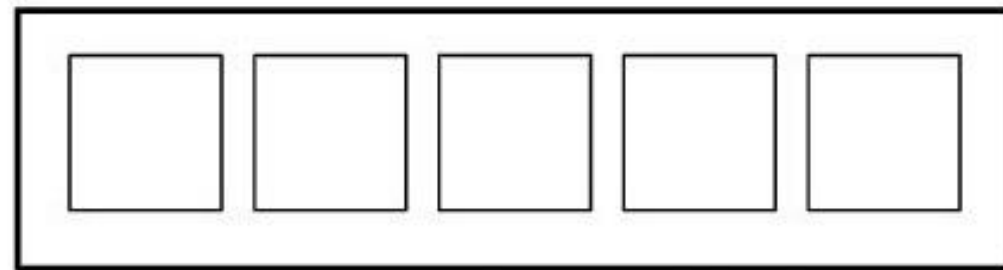
"Esto es una cadena de texto"

A



PRÁCTICA 1: LISTAS

```
miLista = [1, 2, 4, 2.1 , "Hola"]
```



```
print( miLista[0] )
```

FLEXIBLES Y MODIFICABLES

Se puede añadir, quitar y cambiar elementos.



Listas

Son secuencias mutables de elementos, que pueden ser de cualquier tipo y se pueden modificar después de su creación. Se usan normalmente para crear colecciones de datos. Se crean con corchetes [].

Características Principales

- Ordenadas
- Mutables
- Permiten elementos duplicados
- Heterogéneas (pueden tener elementos de diferentes tipos)

Ejemplos de uso

- Almacenar y manipular una colección de datos dinámicos.
- Iterar sobre elementos en bucles.
- Agregar, eliminar o modificar elementos de la colección.



Lista = [1, 2, 3, "a", "b", "c"]

PRÁCTICA 1: TUPLAS

$T = (20, \text{'Juan'}, 35.75, [30, 60, 90])$

\uparrow \uparrow \uparrow \uparrow
 $T[0]$ $T[1]$ $T[2]$ $T[3]$

Ordenado: Mantiene el orden de la inserción de datos.

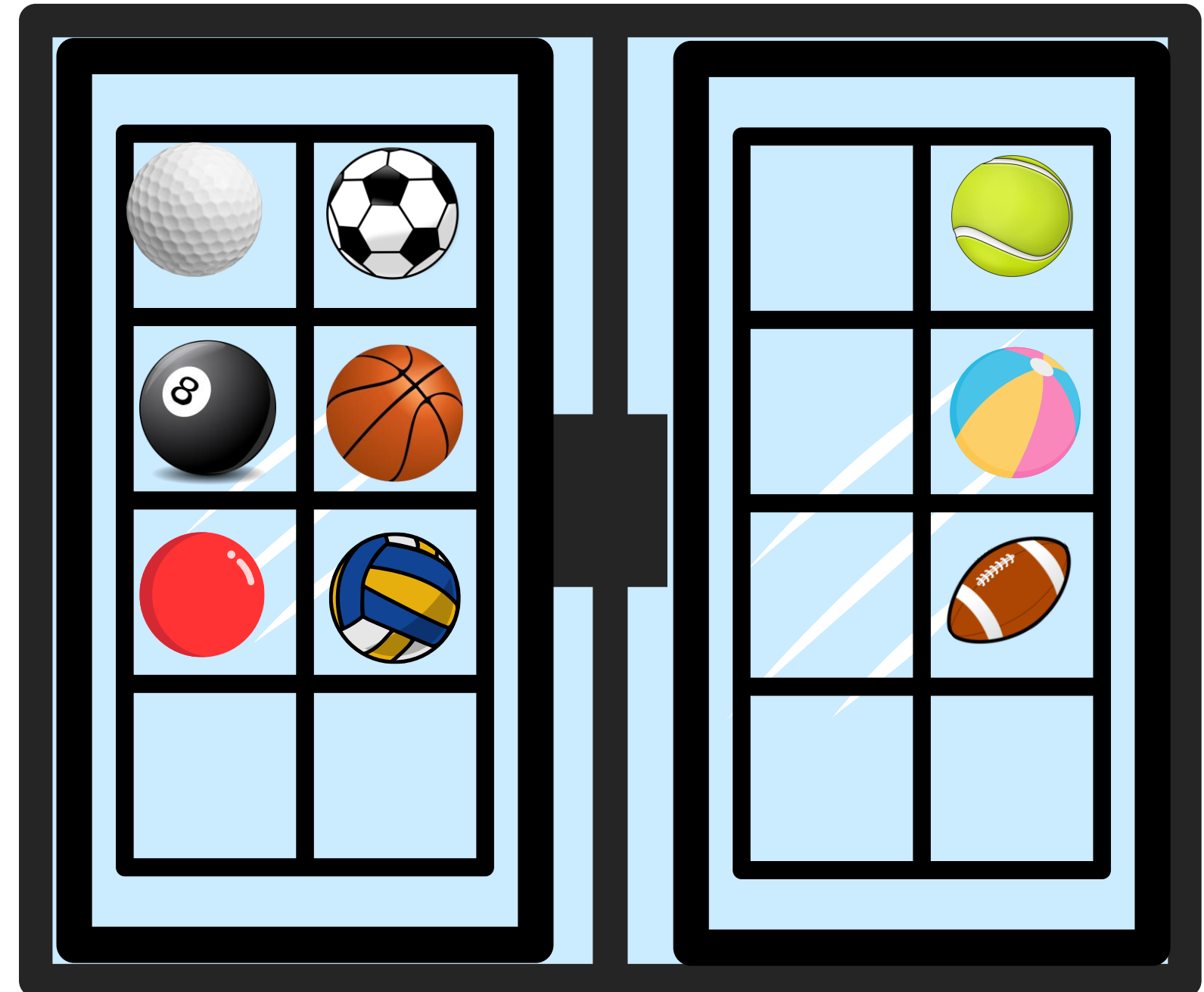
Inmutable: Las tuplas son inmutables y no podemos modificar los elementos.

Heterogéneo: Las tuplas pueden contener datos de diferentes tipos.

Contiene duplicados: Permite datos duplicados.

FIJAS E INMUTABLES

No puedes cambiar, añadir ni quitar elementos.



Tuplas

Son secuencias inmutables de elementos, que también pueden ser de cualquier tipo, pero no se pueden modificar después de su creación.

Características Principales

- Ordenadas
- Inmutables
- Permiten elementos duplicados
- Heterogéneas

Ejemplos de uso:

- Almacenar datos constantes (días de la semana).
- Retornar múltiples valores desde una función.
- Utilizarlas como claves en un diccionario (debido a su inmutabilidad).

Tupla = (1, 2, 3, "a", "b", "c")



PRÁCTICA 1: DICCIONARIOS

```
d = {'a': 10, 'b': 20, 'c': 30}
```

↑ ↑ ↑
d['a'] d['b'] d['c']

Desordenado: Los elementos en el diccionario se almacenan sin ningún valor de índice.
Único: Las claves en los diccionarios deben ser únicas.
Mutable: Podemos agregar/modificar/eliminar pares clave-valor después de la creación.

MODIFICABLES Y NO ORDENADOS

Un diccionario es una estructura de datos que almacena pares clave-valor

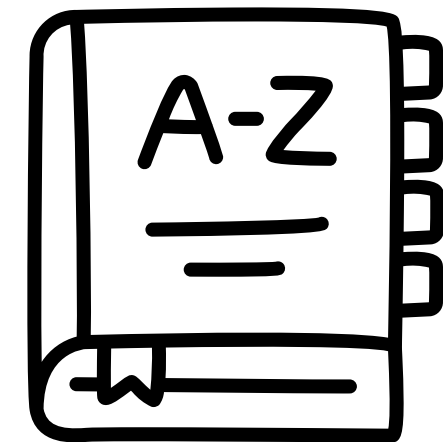


Diccionarios

Son colecciones desordenadas de pares clave-valor, donde cada clave única está asociada a un valor. Los diccionarios permiten un acceso rápido a los valores a través de sus claves. Se crean con las llaves {}.

Características Principales

- Colección de pares clave-valor.
- Mutable (se pueden modificar).
- Ordenados a partir de Python 3.7
- Claves inmutables y únicas.



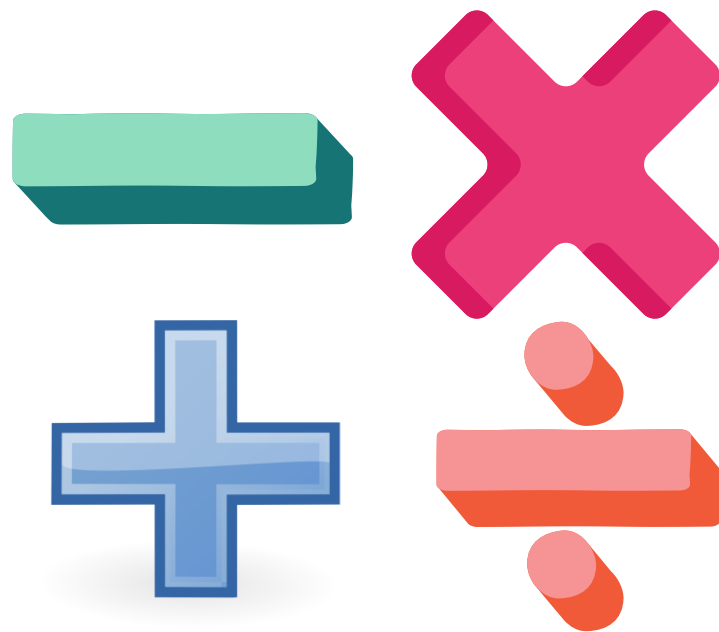
Ejemplos de uso:

- Almacenar y acceder a datos estructurados.
- Implementar tablas de búsqueda y mapas.
- Organizar datos de una manera que permita el acceso rápido a través de claves.

```
Diccionario = {"clave1": "valor1", "clave2":  
               "valor2"}
```

PRÁCTICA 2 : Comandos básicos

OPERACIONES ARITMÉTICAS



Operadores Aritméticos En Python

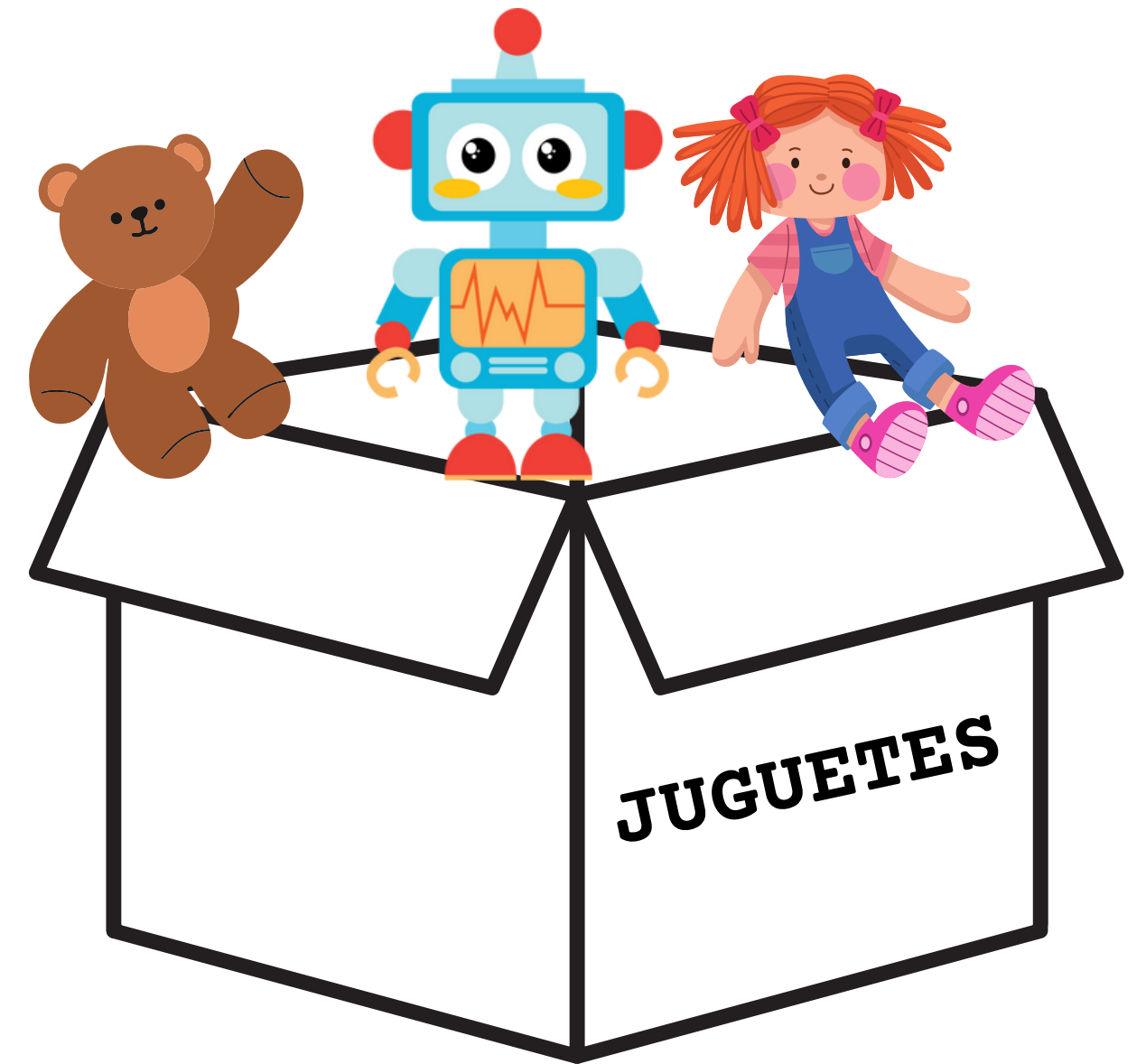
	Operador	Significado	Ejemplo
1	+	Suma	$4 + 7 \rightarrow 11$
2	-	Resta	$12 - 5 \rightarrow 7$
3	*	Multiplicación	$6 * 6 \rightarrow 36$
4	/	División	$30 / 5 \rightarrow 6$
5	%	Módulo	$10 \% 4 \rightarrow 2$
6	//	Cociente	$18 // 5 \rightarrow 3$
7	**	Exponente	$3 ** 5 \rightarrow 243$

PRÁCTICA 2: Comandos básicos

ASIGNAR VARIABLES

Vincular un nombre (etiqueta) a un valor (contenido) utilizando el operador de asignación "="

```
# Asignar una variable en Python
juguetes = 3
print(juguetes) # Imprime 3
```



PRÁCTICA 2 : Comandos básicos

MANIPULAR SERIES



Biblioteca



Trabajar con una **colección de datos** ordenados para realizar operaciones como **añadir**, **cambiar**, **eliminar** y **analizar** los elementos.



Manipular series y asignar variables

Una Series en Pandas es una estructura de datos unidimensional similar a una columna en una tabla o una lista en Python, pero con funcionalidades adicionales. Las series son parte fundamental de la biblioteca pandas y están diseñadas para manejar y manipular datos etiquetados de manera eficiente.

Características Principales

- Operaciones estadísticas
- Métodos y funciones integradas
- Eficientes

Ejemplos de uso

- Análisis de precios de acciones
- Filtrado de temperaturas
- Análisis de series temporales





Próxima sesión

Fundamentos de Python y del aprendizaje automático



In collaboration with
IBM SkillsBuild

¡Gracias!

Envíanos tus preguntas a
ibmskillsbuild.eu@skillup.online

