

2024-1

Python 2

Teoría de Lenguajes de Programación

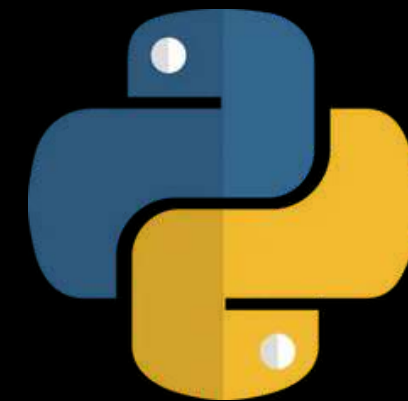
Sara Acevedo Maya
saacevedom@unal.edu.co
Universidad Nacional de Colombia



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Contenido

01/ Listas, tuplas,
diccionarios, arreglos
02/ Analítica de datos
(Pandas y Numpy)



UNIVERSIDAD
NACIONAL
DE COLOMBIA

01 - Listas, tuplas, diccionarios, arreglos

Listas

Las listas son colecciones **ordenadas y mutables** en Python. Pueden almacenar cualquier tipo de dato y se pueden modificar después de su creación.

Propiedades: mutable, indexada.

Operaciones comunes: añadir (append), eliminar (remove), ordenar (sort).



Listas

● ● ●

```
# Crear una lista
mi_lista = [1, 2, 3, 4, 5]
print(mi_lista) # Imprime: [1, 2, 3, 4, 5]

# Acceder a elementos
print(mi_lista[0]) # Imprime: 1 (primer elemento de la lista)

# Modificar un elemento
mi_lista[1] = 'dos'
print(mi_lista) # Imprime: [1, 'dos', 3, 4, 5] (se cambia el segundo elemento)

# Añadir un elemento
mi_lista.append(6)
print(mi_lista) # Imprime: [1, 'dos', 3, 4, 5, 6] (se añade el 6 al final)

# Eliminar un elemento
mi_lista.remove(3)
print(mi_lista) # Imprime: [1, 'dos', 4, 5, 6] (elimina el 3 de la lista)
```



Tuplas

Las tuplas son colecciones **ordenadas e inmutables**. Una vez creada una tupla, no se puede modificar.

```
● ● ●  
  
# Crear una tupla  
mi_tupla = (1, 2, 3, 4, 5)  
print(mi_tupla)  
  
# Acceder a elementos  
print(mi_tupla[0])  
  
# No se puede modificar ni añadir elementos a una tupla.
```

Propiedades: mutable, indexada.

Operaciones comunes: cuando se necesita una colección de datos que no deba cambiar.



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Diccionarios

Los diccionarios son colecciones **no ordenadas de pares clave-valor**. Son útiles para almacenar datos que se pueden identificar mediante una clave

Propiedades: mutable, clave-valor, no indexado

Operaciones comunes: añadir (`mi_diccionario[clave] = valor`), eliminar (`del`), acceder a valores.



```
● ● ●

# Crear un diccionario
mi_diccionario = {
    "nombre": "Juan",
    "edad": 30,
    "ciudad": "Medellín"
}

print(mi_diccionario)
# Imprime: {'nombre': 'Juan', 'edad': 30, 'ciudad': 'Medellín'}
# (el diccionario completo)

# Acceder a un valor usando su clave
print(mi_diccionario["nombre"])
# Imprime: Juan (el valor asociado a la clave "nombre")

# Modificar un valor
mi_diccionario["edad"] = 31
print(mi_diccionario)
# Imprime: {'nombre': 'Juan', 'edad': 31, 'ciudad': 'Medellín'}
# (el valor de "edad" ha sido modificado a 31)

# Añadir un nuevo par clave-valor
mi_diccionario["profesión"] = "Ingeniero"
print(mi_diccionario)
# Imprime: {'nombre': 'Juan', 'edad': 31, 'ciudad': 'Medellín', 'profesión': 'Ingeniero'}
# (se añade el par clave-valor "profesión": "Ingeniero")

# Eliminar un par clave-valor
del mi_diccionario["ciudad"]
print(mi_diccionario)
# Imprime: {'nombre': 'Juan', 'edad': 31, 'profesión': 'Ingeniero'}
# (se elimina el par clave-valor con la clave "ciudad")
```



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Arreglos (Arrays)

Los arreglos en Python son gestionados mediante el módulo array. Son **más eficientes** en memoria que las listas, pero solo pueden almacenar **elementos de un solo tipo**

Propiedades: mutable, indexado, solo un tipo de dato

usos comunes: cuando se necesita eficiencia en memoria y velocidad, pero solo se requiere un tipo de dato

```
...  
  
import array as arr  
  
mi_arreglo = arr.array('i', [1, 2, 3, 4, 5])
```



Arreglos (Arrays)

'i': Es el tipo de datos que especifica que el arreglo contendrá enteros con signo.

[1, 2, 3, 4, 5]: Es la lista de valores iniciales que se almacenarán en el arreglo.

Código	Tipo de dato	Descripción
'b'	byte	Entero con signo de 1 byte
'B'	unsigned byte	Entero sin signo de 1 byte
'i'	integer	Entero con signo de 4 bytes
'I'	unsigned integer	Entero sin signo de 4 bytes
'f'	float	Flotante de 4 bytes
'd'	double	Flotante de 8 bytes
'h'	short	Entero con signo de 2 bytes
'H'	unsigned short	Entero sin signo de 2 bytes




```
import array as arr

# Crear un arreglo de enteros
mi_arreglo = arr.array('i', [1, 2, 3, 4, 5])
print(mi_arreglo)
# Imprime: array('i', [1, 2, 3, 4, 5])
# (un arreglo de enteros con los valores 1, 2, 3, 4, 5)

# Acceder a un elemento
print(mi_arreglo[2])
# Imprime: 3
# (el elemento en la posición 2 del arreglo)

# Modificar un elemento
mi_arreglo[0] = 10
print(mi_arreglo)
# Imprime: array('i', [10, 2, 3, 4, 5])
# (se modifica el primer elemento a 10)

# Añadir un elemento
mi_arreglo.append(6)
print(mi_arreglo)
# Imprime: array('i', [10, 2, 3, 4, 5, 6])
# (se añade el 6 al final del arreglo)
```



Comparación entre las estructuras:

Estructura	Mutable	Indexada	Ordenada	Permite duplicados
Listas	Sí	Sí	Sí	Sí
Tuplas	No	Sí	Sí	Sí
Diccionarios	Sí	No	No	No (en las claves)
Arreglos	Sí	Sí	Sí	Sí (del mismo tipo)



02 - Analítica de datos (Pandas y Numpy)



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Analítica de Datos

La analítica de datos es el proceso de **examinar** grandes volúmenes de datos para descubrir patrones, tendencias, y relaciones que puedan apoyar la toma de decisiones informadas.

NumPy:

Biblioteca de Python diseñada para realizar **operaciones numéricas eficientes** en arreglos y matrices multidimensionales, ofreciendo funciones matemáticas avanzadas

Pandas:

Biblioteca de Python para la manipulación y análisis de datos estructurados, que utiliza estructuras como **DataFrames** para trabajar con datos tabulares.



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Analítica de Datos

Dataset: Conjunto de datos recopilados y organizados, típicamente en forma de tabla, que contiene información sobre diferentes variables y observaciones

DataFrame: Estructura de datos en Pandas (y en otros frameworks) que organiza los datos en formato tabular, con filas y columnas, permitiendo manipulación y análisis eficientes



https://www.instagram.com/p/C9vpr-gBPpJ/?img_index=1

Find Open Datasets and Machine Learning Projects

Download Open Datasets on 1000s of Projects + Share Projects on One Platform. Explore Popular Topics Like Government, Sports, Medicine, Fintech, Food, More. Flexible Data Ingestion.

 Kaggle

<https://www.kaggle.com/datasets>



UNIVERSIDAD
NACIONAL
DE COLOMBIA



Create beautiful images of your code

```
const pluckDeep = key => obj =>  
  key.split('.').reduce((accum, key) => accum[key], obj)  
  
const compose = (...fns) => res => fns.reduce((accum, next)  
  => next(accum), res)  
  
const unfold = (f, seed) => {
```

Create beautiful images of your code

Turn your code into beautiful images. Choose from a range of syntax colors, hide or show the background, and toggle between a dark and light window.



<https://ray.so>

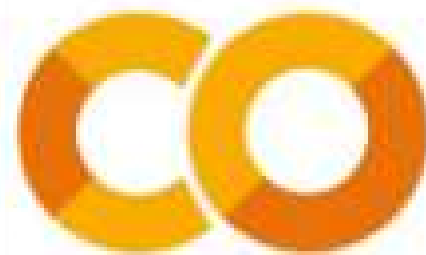


UNIVERSIDAD
NACIONAL
DE COLOMBIA

COLAB

es una plataforma en línea gratuita proporcionada por Google para escribir y ejecutar código Python en un entorno de cuadernos **Jupyter**

Permite a los usuarios realizar análisis de datos, machine learning, y visualización sin necesidad de instalar nada en sus computadoras locales



Google Colab

 google.com

<https://colab.research.google.com/drive/1XhFA1S4OwnqwoLkwVlggSO81BsqeTE7I?usp=sharing>



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Tarea

Grupo 1,2,3:

Plazo hasta el Domingo 15 de Septiembre a las 11:59pm:

<https://forms.gle/8ockjkVQTiJpubJt6>



UNIVERSIDAD
NACIONAL
DE COLOMBIA