



Tipos y estructuras de datos (parte I)

Clase sincrónica

Ideas fuerza



Las listas y diccionarios corresponden a estructuras flexibles para el uso de **diferentes tipos de datos**



Utilizamos **NumPy Arrays** para optimizar las posibilidades de **cálculo**, sobre estructuras de **datos numéricos**.



Pandas nos provee de poderosas herramientas para el manejo, creación y presentación de estructuras de datos de **diferentes tipos**.

Recursos asincrónicos

- ¿Revisaste los recursos asincrónicos?
- ¿Alguno de ellos te dejó dudas?



`/*Listas y diccionarios*/`

Listas y diccionarios

Agregar y mostrar elementos

```
frutas_favoritas = ['manzana', 'banana', 'uva', 'fresa']
amigos = {"Juan": "Bogotá", "Maria": "Medellín", "Luis": "Cali", "Ana": "Barranquilla"}

# agregar nuevo elemento
frutas_favoritas.append(nueva_fruta)
amigos["Raúl"] = "Caracas"

# Mostrar elementos
print("Lista de frutas favoritas:")
for fruta in frutas_favoritas:
    print("-", fruta)

print("Este es tu diccionario de amigos y ciudades:")
for amigo, ciudad in amigos.items():
    print(f"- {amigo} vive en {ciudad}")
```

{desafío}
latam_

Listas y diccionarios

Quitar y filtrar elementos

```
frutas_favoritas = ['manzana', 'banana', 'uva', 'fresa', 'mango', 'piña', 'maracuyá']
amigos = {"Juan": 30, "Maria": 24, "Luis": 30, "Ana": 25, "José": 30}

#quitar elementos
frutas_favoritas.remove('fresa')
del amigos["José"]

# filtros
frutas_filtradas = [fruta for fruta in frutas_favoritas if fruta.lower().startswith("m")]

for fruta in frutas_filtradas:
    print("-", fruta)

for amigo, edad in amigos.items():
    if edad == 30:
        print("-", amigo)
```

- manzana
- mango
- maracuyá
- Juan
- Luis

{desafío}
latam_

Listas y diccionarios

Ordenamiento

```
frutas_favoritas = ['manzana', 'banana', 'uva', 'mango', 'piña', 'maracuyá']
amigos = {"Juan": 30, "Maria": 24, "Luis": 30, "Ana": 25, "José": 30}

frutas_favoritas.sort()
amigos_ordenados = dict(sorted(amigos.items()))

for i in frutas_favoritas:
    print(i)

for amigo, edad in amigos_ordenados.items():
    print(f"Mi amigo(a) {amigo} tiene {edad} años")
```

```
banana
mango
manzana
maracuyá
piña
uva
Mi amigo(a) Ana tiene 25 años
Mi amigo(a) José tiene 30 años
Mi amigo(a) Juan tiene 30 años
Mi amigo(a) Luis tiene 30 años
Mi amigo(a) Maria tiene 24 años
```

Listas y diccionarios

Transformaciones

```
nombres = ["Juan", "María", "Pedro"]
edades = [28, 30, 25]

diccionario_edades = dict(zip(nombres, edades))

nombres = list(diccionario_edades.keys())
edades = list(diccionario_edades.values())

print(diccionario_edades)
print(nombres)
print(edades)
```

```
{'Juan': 28, 'María': 30, 'Pedro': 25}
['Juan', 'María', 'Pedro']
[28, 30, 25]
```


/*NumPy Arrays*/

NumPy Arrays

Arrays y listas

```
import numpy as np

lista = [0, 1, 2, 3, 4]
my_array = np.array(lista)

my_array

array([0, 1, 2, 3, 4])
```

```
# Operación sobre una lista.
lista*2

[0, 1, 2, 3, 4, 0, 1, 2, 3, 4]
```

```
# Operación sobre un Array.
my_array*2

array([0, 2, 4, 6, 8])
```

NumPy Arrays

Operaciones vectorizadas

```
lista1 = [1,2,3]  
lista2 = [10,20,30]  
lista3 = [100, 200, 300]  
  
array1 = np.array(lista1)  
array2 = np.array(lista2)  
array3 = np.array(lista3)
```

```
lista1 + lista2 + lista3
```

```
[1, 2, 3, 10, 20, 30, 100, 200, 300]
```

```
array1 + array2 + array3
```

```
array([111, 222, 333])
```

NumPy Arrays

Filtrado de datos

```
import numpy as np

lista = [1, 6, 3, 8, 2, 9, 10]

arr = np.array(lista)

arr[arr > 5]

array([ 6,  8,  9, 10])
```

```
arr[np.logical_and(arr > 5, arr < 10)]

array([6, 8, 9])
```

```
arr[np.logical_or(arr <= 3, arr > 8)]

array([ 1,  3,  2,  9, 10])
```

NumPy Arrays

Conteo y búsqueda

```
import numpy as np
personas = ["Claudia", "Marcela", "Luis", "Antonio", "Laura", "José", "Pedro", "Mariela"]
edades = np.array([5,12,17,3,8,41,24,37])

print(lista.argmax())
print(personas[lista.argmin()])
print(edades[edades>=18])
print(edades[edades>=18].shape)
```

```
5
Antonio
[41 24 37]
(3,)
```

{desafío}
latam_

NumPy Arrays

Cálculo de indicadores estadísticos

```
import numpy as np
lista = [3,5,6,8,2,6,3,7,3,8,7,9,7,4,3,7,2,4,6]
lista1 = np.array(lista)

print(np.mean(lista1)) #promedio
print(np.median(lista1)) # mediana
print(np.std(lista1)) # desviación estándar
print(np.var(lista1)) #varianza
print(np.percentile(lista1,20)) #percentil 20
```

```
5.2631578947368425
6.0
2.1482454349430053
4.6149584487534625
3.0
```

/*Carga de datos con NumPy*/

Carga de datos con NumPy

Creando listas desde archivos

NumPy nos permite cargar archivos csv para crear arrays y realizar cálculos.

```
nombres = np.loadtxt("nombre_astronautas.csv", delimiter=',', dtype = str, encoding='utf-8')
edades  = np.loadtxt("edad_astronautas.csv", delimiter=',', dtype = int)
alturas = np.loadtxt("altura_astronautas.csv", delimiter=',', dtype = float)
```


Desafío - Tipos y estructuras de datos



Desafío

"Conociendo Python II"

- ¿Hay contenidos que necesitas repasar antes de comenzar el desafío?
- ¿Comprendes bien qué te están solicitando en cada caso?





Próxima sesión...

- *Utilizar y operar datos con diferentes estructuras.*
- *Cargar diferentes tipos de datos.*

{desafío}
latam_

*Academia de
talentos digitales*

