

{desafío}
latam_

API _

Sesión Presencial 1



Itinerario

Activación de conceptos

Desarrollo
Desafío

Panel de discusión

Activación de conceptos

¿Cuál es la convención para importar pandas?

1. From pandas import dataframe
2. Use pandas as pd
3. Import pandas as pd
4. Import pandas as pan
5. From pandas import pd

¿Cómo haría un subset que incluya toda las filas y las columnas adfert, chldmort y life??

	Unnamed: 0	country	region	gdp	school	adfert	chldmort	life	pop	urban	femlab	literacy	co2	gini
0	1	Algeria	Africa	7300.399902	6.716667	7.300000	34.75	72.316666	34172236	64.933334	0.4522	72.599998	15.0	NaN
1	2	Benin	Africa	1338.800049	3.100000	111.699997	122.75	54.733334	8237634	41.000000	0.8482	41.700001	1.2	NaN
2	3	Botswana	Africa	12307.400391	8.600000	52.099998	60.25	52.250000	1941233	59.250000	0.8870	84.099998	9.2	NaN

1. `df[colums= ["adfert", "chldmort", "life"]]`
2. `df.loc[:, ["adfert":"life"]]`
3. `df[["adfert", "chldmort", "life"]]`
4. `df.loc[:,["adfert","chldmort","life"]]`
5. `pd.Series(df[["adfert":"life"]])`

¿Cuál de las siguientes expresiones no genera un ndarray?

1. `np.arange(50)`
2. `np.array([4, 10, 20])`
3. `np.random.randn(2,2)`
4. `np.sqrt(9)`
5. `np.ones(5)`

¿Cómo es la sintaxis de np.where()?

1. `np.where(<condicion>,<valor positivo>,<valor negativo>)`
2. `np.where(<valor positivo>, <valor negativo>, <condicion>)`
3. `np.where(<condicion>,<valor negativo>,<valor positivo>)`
4. `np.where(<condicion>:<valor positivo>:<valor negativo>)`
5. `np.where(<valor positivo>/<valor negativo>/<condicion>)`

Listas vs Diccionarios

Lista

- Se accede a los elementos por medio de la posición o índice.
- Los índices se definen implícitamente.
- Los índices siempre son int

Diccionario

- Se accede a los elementos por medio de la clave.
- Las claves se definen explícitamente.
- Las claves suelen ser string

```
lista = [25, 31, "hola"]  
lista[2] # "hola"
```

```
diccionario = {"a": 25, "b": 31, "c": "hola"}  
diccionario["c"] # "hola"
```


Acciones sobre los diccionarios

- Definir un diccionario

```
notas = {"Camila": 7, "Antonio": 5, "Felipe": 6, "Antonia": 7}
```

- Acceder a un elemento

```
notas["Felipe"] # 6
```

- Agregar un elemento

```
diccionario["llave 2"] = 9
```

- Cambiar un elemento

```
diccionario = {"llave 1": 5, "llave 2": 7}  
diccionario["llave 2"] = 9
```

Iterando diccionarios

- 1 iterador = Se itera la **clave**

```
1 diccionario = {"nombre": "Juan", "apellido": "Pérez", "edad": 33, "altura": 1.75}
2
3 for clave in diccionario:
4     print("Esta clave es: {}".format(clave))
```

Esta clave es: nombre

Esta clave es: apellido

Esta clave es: edad

Esta clave es: altura

Iterando diccionarios

- 2 iteradores: Clave, valor
- Se debe usar la función items()

```
: 1 for clave, valor in diccionario.items():  
  2     print("La clave es {} y el valor es {}".format(clave, valor))
```

La clave es nombre y el valor es Juan
La clave es apellido y el valor es Pérez
La clave es edad y el valor es 33
La clave es altura y el valor es 1.75

Operaciones típicas

Eliminar elementos

Pueden usarse “del” o pop()

```
1 diccionario = {"celular": 140000, "notebook": 489990, "tablet": 120000, "cargador": 12400}
```

```
1 del diccionario["celular"]  
2  
3 print(diccionario)
```

```
{'notebook': 489990, 'tablet': 120000, 'cargador': 12400}
```

```
1 eliminado = diccionario.pop("tablet")  
2  
3 print(eliminado)  
4 print(diccionario)
```

```
120000  
{'notebook': 489990, 'cargador': 12400}
```

Unir diccionarios

```
1 diccionario_a = {"nombre": "Alejandra", "apellido": "López", "edad": 33, "altura": 1.55}
2 diccionario_b = { "mascota": "miti", "ejercicio": "bicicleta", "altura": 155}
3
4 # Union de diccionario_a y diccionario_b
5 diccionario_a.update(diccionario_b)
6
7 # Se sobrescribió el valor de altura por el del diccionario_b
8 print(diccionario_a)
```

```
{'nombre': 'Alejandra', 'apellido': 'López', 'edad': 33, 'altura': 155, 'mascota': 'miti', 'ejercicio': 'bicicleta'}
```

Invertir

```
1 colors = {"red": "#cc0000", "green": "#00cc00", "blue": "#0000cc"}
2 colors_inv = {v: k for k, v in colors.items()}
3 colors_inv["#cc0000"] # red
```

'red'

Obtener claves y valores

```
1 colors = {"red": "#cc0000", "green": "#00cc00", "blue": "#0000cc"}  
2 print(colors.keys())  
3 print(colors.values())
```

```
dict_keys(['red', 'green', 'blue'])  
dict_values(['#cc0000', '#00cc00', '#0000cc'])
```


Construir un diccionario a partir de claves y valores

```
1 dict(zip(["k1", "k2", "k3"], [1, 2, 3]))  
{'k1': 1, 'k2': 2, 'k3': 3}
```

Convertir estructuras

- Convertir un diccionario en una lista

```
list({"k1": 5, "k2": 7}.items()) # [('k1', 5), ('k2', 7)]
```

- Convertir una lista en un diccionario

```
dict([('k1', 5), ('k2', 7)]) # {"k1": 5, "k2": 7}
```

- Unir 2 listas en un diccionario

```
1 nombres = ['Alumno1', 'Alumno2', 'Alumno3']  
2 notas = [10, 3, 8]  
3  
4 notas_por_alumno = dict(zip(nombres, notas))  
5  
6 print(notas_por_alumno)
```

```
{'Alumno1': 10, 'Alumno2': 3, 'Alumno3': 8}
```

Uso de groupby

- Se importa como “from itertools import groupby”
- Permite agrupar elementos de una estructura bajo un criterio
- Retorna llaves consecutivas y sus grupos asociados
- Cada grupo se creará según los datos que tengan igual valor del criterio
- Se requiere que los datos estén ordenados antes de agruparlos

Groupby sobre un string

```
1 from itertools import groupby
2 {k: len(list(g)) for k, g in groupby('AAAABBBCCD')}
```

{'A': 4, 'B': 3, 'C': 2, 'D': 1}

Groupby sobre una lista

```
1 # Agrupar palabras por su largo
2
3 from itertools import groupby
4 words = ["hola", "a", "todos", "y", "cada", "uno"]
5
6 # Primero ordenamos
7 words.sort(key=lambda x: len(x))
8
9 # Luego agrupamos
10 {k: list(v) for k, v in groupby(words, key=len)}
```

```
{1: ['a', 'y'], 3: ['uno'], 4: ['hola', 'cada'], 5: ['todos']}
```

Agrupar y contar

```
1 lista = [1, 2, 6, 7, 2, 5, 8, 9, 1, 2, 9, 7]
2
3 lista.sort()
4 diccionario = {k: len(list(v)) for k, v in groupby(lista)}
5
6 print(diccionario)
```

```
{1: 2, 2: 3, 5: 1, 6: 1, 7: 2, 8: 1, 9: 2}
```

/* Desafíos */

Panel de discusión

{desafío}
latam_

*Academia de
talentos digitales*

www.desafiolatam.com