



Universidad del  
**Rosario**

Educación Continua  
y Consultoría

# Diplomado en Ciencia de Datos



#URSolucionesInnovadoras

#URConsultoría



Universidad del  
**Rosario**

Educación Continua  
y Consultoría

## PROGRAMACIÓN 3

# Numpy y Pandas



Universidad del  
**Rosario**

Educación Continua  
y Consultoría

---

# Contenido

- Pandas dataframe y series
- Manipulación de dataframes
- El objeto Series
- Algunas operaciones matemáticas
- Series y funciones integradas de Python
- Retos de programación
- Numpy





Universidad del  
**Rosario**

Educación Continua  
y Consultoría

---

# Contenido

- **Pandas dataframe y series**
- Manipulación de dataframes
- El objeto Series
- Algunas operaciones matemáticas
- Series y funciones integradas de Python
- Retos de programación
- Numpy





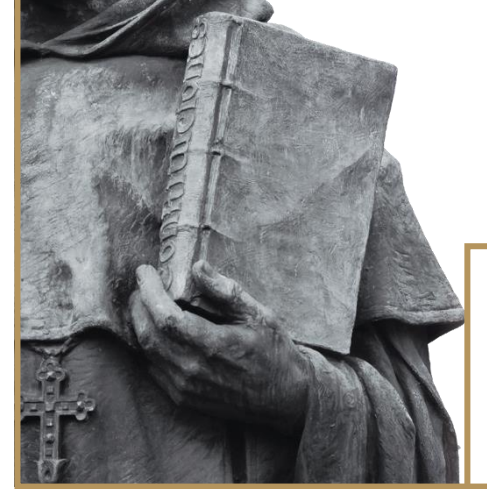
## Pandas dataframe y series

# Dataframes y series

- Estructura de datos principal en pandas.
- Es una tabla de datos con múltiples columnas (series).
- Una serie es una matriz etiquetada unidimensional, una sola columna de datos.

Series 1		Series 2		Series 3		DataFrame			
Mango		Apple		Banana		Mango Apple Banana			
0	4	0	5	0	2	0	4	5	2
1	5	1	4	1	3	1	5	4	3
2	6	2	3	2	5	2	6	3	5
3	3	3	0	3	2	3	3	0	2
4	1	4	2	4	7	4	1	2	7

<https://www.datasciencemadesimple.com/create-series-in-python-pandas/>





---

# Contenido

- Pandas dataframe y series
- **Manipulación de dataframes**
- El objeto Series
- Algunas operaciones matemáticas
- Series y funciones integradas de Python
- Retos de programación
- Numpy





## Manipulación de dataframes

# Análisis básico

### Información general del dataframe

- head y tail
- Número de filas, columnas y celdas
- Tipos de datos en cada columna

### Acceder a una fila en una serie o datarame

- Por su número de fila
- Por su etiqueta de índice

### Ordenar datos

- Por una o varias columnas
- Por el índice





Universidad del  
**Rosario**

Educación Continua  
y Consultoría

Manipulación de dataframes

# Análisis un poco más sofisticado

- Extraer columnas
  - Una
  - Varias
- Filtro por diferentes criterios
- Agrupación de datos







## Manipulación de dataframes

# Análisis un poco más sofisticado



Se pueden agrupar las filas del dataframe en función de los valores de una columna

Algunas operaciones de agregación sobre los grupos resultantes

- Conteos
- Sumas
- Promedios
- ...



---

# Contenido

- Pandas dataframe y series
- Manipulación de dataframes
- **El objeto Series**
- Algunas operaciones matemáticas
- Series y funciones integradas de Python
- Retos de programación
- Numpy





## El objeto Series

# ¿Qué es una Serie?

- Es un array unidimensional similar a una lista [ ]
- Tiene una etiqueta (label)
- Los datos son homogéneos
- Cada valor tiene una posición y una etiqueta que permite identificar cada valor (como un diccionario)
- Agrega más de 180 métodos para manipulación de datos





## El objeto Series

# Instanciar una serie

El constructor recibe como primer parámetro (data) una lista de elementos

- Se puede asignar a cada valor de serie una etiqueta de índice.
- Las etiquetas de índice pueden ser de cualquier tipo de datos inmutables: cadenas, tuplas, fechas y horas y más.
- Se puede referenciar un valor por su orden o por una clave/etiqueta. Cada valor tiene dos identificadores.

Índice



```
0    Chocolate
1     Vanilla
2  Strawberry
3  Rum Raisin
dtype: object
```





## El objeto Series

# Modificar el índice



El constructor recibe como segundo parámetro un índice personalizado

Índice

```
pd.Series(data = ice_cream_flavors, index = days_of_week)
```

Out [5]	Monday	Chocolate
	Wednesday	Vanilla
	Friday	Strawberry
	Saturday	Rum Raisin
	dtype: object	

- A diferencia de los diccionarios, el índice permite llaves duplicadas (hay que evitarlo)



## El objeto Series

# Tipos de datos

Para la mayoría de tipos de datos, pandas detecta el tipo de datos

- Bool
- Float
- Int

```
Open      985.32  
Close     950.44  
dtype: float64
```

Para otros tipos de datos como strings u objetos complejos, se muestra dtype:object

Se puede especificar el tipo con el parámetro **dtype**



```
0      4  
1      8  
2     15  
3     16  
4     23  
5     42  
dtype: int64
```

```
0      True  
1     False  
2     False  
dtype: bool
```



## El objeto Series

# Valores faltantes

- Cuando pandas ve un valor faltante durante la importación de un archivo, la biblioteca sustituye el objeto nan de NumPy.
- El acrónimo nan es la abreviatura de no un número y es un término general para un valor indefinido.
- NaN es un objeto marcador de posición que representa nulidad o ausencia.
- Cuando la serie tiene un nulo, pandas convierte la serie a float



```
import numpy as np

temperatures = [94, 88, np.nan, 91]
pd.Series(data = temperatures)
```



Universidad del  
**Rosario**

Educación Continua  
y Consultoría

El objeto Series

# Crear Series a partir de objetos de Python

El parámetro data acepta varios tipos de datos

- Diccionarios
- Tuplas
- Conjuntos (transformados a listas)
- Arrays de numpy





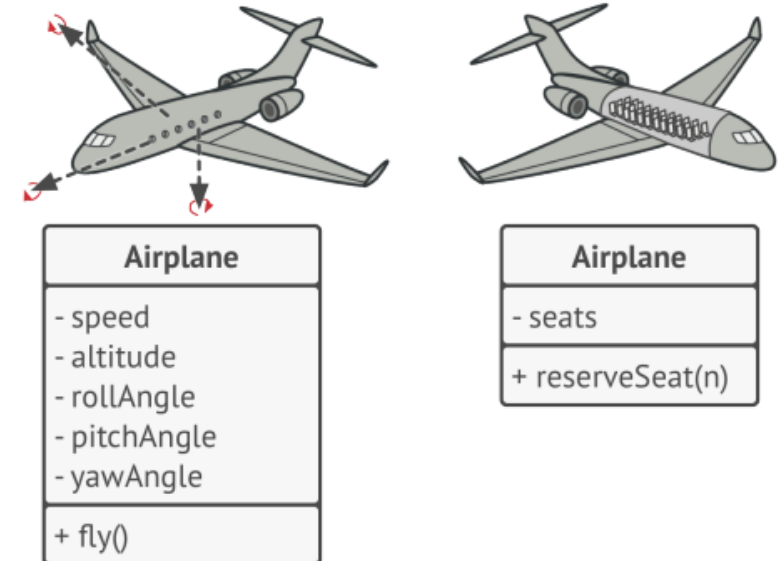


## El objeto Series

# Un poco de programación orientada a objetos

## Conceptos principales

- **Encapsulamiento:** evitar que cualquier código pueda modificar una estructura de datos
- **Abstracción:** propiedades relevantes a observar



*Different models of the same real-world object.*



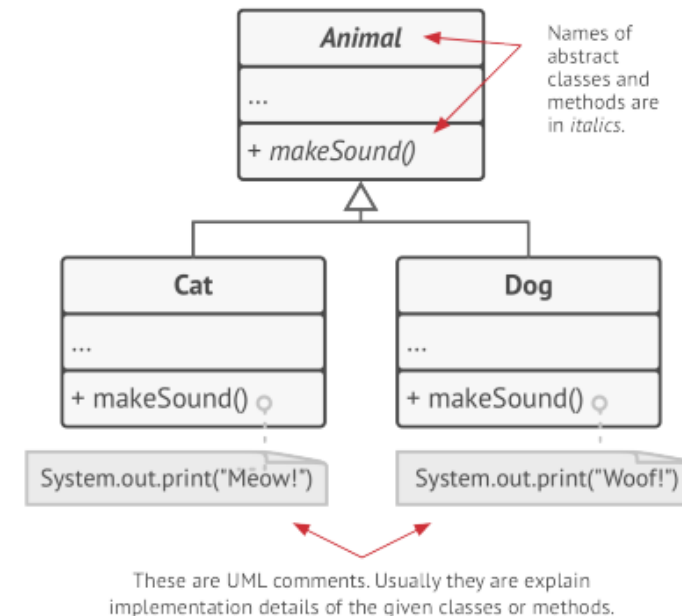
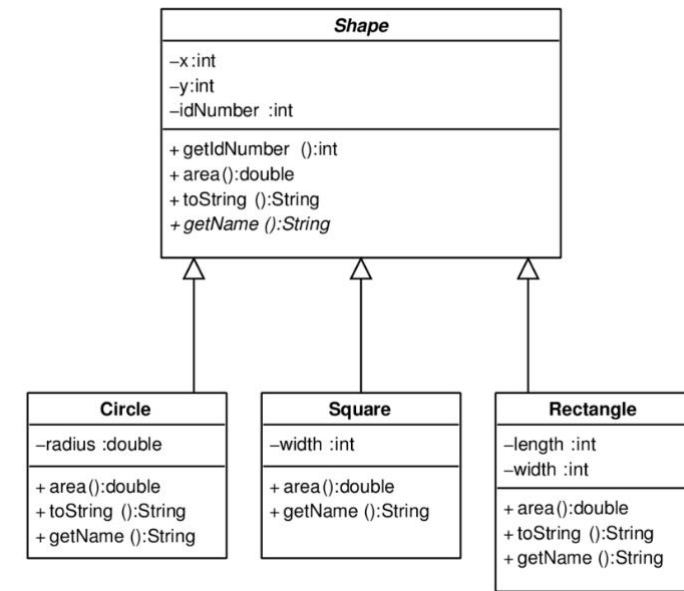


## El objeto Series

# Un poco de programación orientada a objetos

## Conceptos principales

- **Herencia:** reusar el código heredando el comportamiento de otros objetos
- **Polimorfismo:** el método invocado varía en función de la clase de la instancia de un objeto





El objeto Series

# Creación de una clase

```
class Employee:

    # constructor
    def __init__(self, name, salary, project):

        # atributos
        self.name = name
        self.salary = salary
        self.project = project

    #métodos

    def show(self):
        print("Name:", self.name, "Salary:", self.salary)

    def work(self):
        print(self.name, "is working on", self.project)
```





Universidad del  
**Rosario**

Educación Continua  
y Consultoría

El objeto Series

# Instanciación

```
emp = Employee("John", 1000, "Project X")
```

```
emp.show()
```

```
emp.work()
```





## El objeto Series

# Atributos de las serie

Una serie se compone de objetos más pequeños

- Objeto **Index** de pandas para guardar el índice
- Objeto **ndarray** para guardar los datos





## El objeto Series

# Métodos de las serie

- Una serie tiene funciones/acciones
- Usualmente involucran análisis, cálculos, manipulación de los atributos
- Se identifican por los paréntesis para ingresar parámetros al final de la función (si existen)





---

# Contenido

- Pandas dataframe y series
- Manipulación de dataframes
- El objeto Series
- **Algunas operaciones matemáticas**
- Series y funciones integradas de Python
- Retos de programación
- Numpy





## Algunas operaciones matemáticas

# Operaciones estadísticas

- Conteo
- Suma
- Producto
- Suma acumulativa
- Cambio porcentual
- Media
- Mediana
- Desviación estándar
- Máximo
- Mínimo
- Describe
- Muestreo
- Único y conteo de únicos







Universidad del  
**Rosario**

Educación Continua  
y Consultoría

Algunas operaciones matemáticas

# Operaciones aritméticas

- Suma
- Resta
- Multiplicación
- División
- División entera
- Módulo





## Algunas operaciones matemáticas

# Broadcasting

- Se delegan las operaciones aritméticas a numpy
- Aplica la misma operación a todos los valores de la serie
- Cuando se suman dos series, se suman los valores que tienen el mismo índice
- Cuando se comparan por igualdad (`==`), pandas considera que un **NaN** no es igual a otro **NaN**





Universidad del  
**Rosario**

Educación Continua  
y Consultoría

---

# Contenido

- Pandas dataframe y series
- Manipulación de dataframes
- El objeto Series
- Algunas operaciones matemáticas
- **Series y funciones integradas de Python**
- Retos de programación
- Numpy





## Series y funciones integradas de Python

# Integración fluída



- A los desarrolladores de Python les gusta respetar ciertos principios de diseño para garantizar la coherencia entre las bases de código.
- Un ejemplo es la integración perfecta entre los objetos de la biblioteca y las funciones integradas de Python. Pandas no es una excepción.
- Podemos pasar una serie a cualquiera de las funciones integradas de Python y obtener un resultado predecible.



Universidad del  
**Rosario**

Educación Continua  
y Consultoría

---

# Contenido

- Pandas dataframe y series
- Manipulación de dataframes
- El objeto Series
- Algunas operaciones matemáticas
- Series y funciones integradas de Python
- **Retos de programación**
- Numpy





## Retos de programación

# Actividad

```
superheroes = [  
    "Batman",  
    "Superman",  
    "Spider-Man",  
    "Iron Man",  
    "Captain America",  
    "Wonder Woman"  
]
```

```
strength_levels = (100, 120, 90, 95, 110, 120)
```

1. Use la lista de **superheroes** para completar un nuevo objeto Serie.
2. Use la tupla de **strength\_levels** para llenar un nuevo objeto Serie.
3. Cree una serie con los superhéroes como etiquetas de índice y los niveles de fuerza como valores. Asigne la serie a una variable llamada **heroes**.
4. Extraiga las dos primeras filas de la Serie de **heroes**.
5. Extraiga las últimas cuatro filas de la Serie de **heroes**.
6. Determina el número de valores únicos en la serie de **heroes**.
7. Calcula la fuerza promedio de los superhéroes en **heroes**.
8. Calcula las fuerzas máximas y mínimas de los **heroes**.
9. Calcula cuál sería el nivel de fuerza de cada **heroes** si se duplicara.
10. Convierta la Serie de **heroes** en un diccionario de Python.



Universidad del  
**Rosario**

Educación Continua  
y Consultoría

---

# Contenido

- Pandas dataframe y series
- Manipulación de dataframes
- El objeto Series
- Algunas operaciones matemáticas
- Series y funciones integradas de Python
- Retos de programación
- **Numpy**





Numpy

# Arrays n-dimensionales



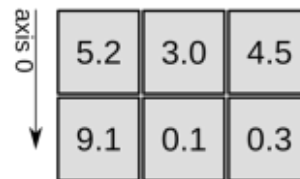
1D array



axis 0 →

shape: (4,)

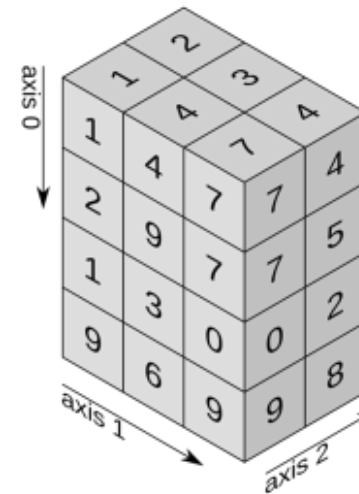
2D array



axis 1 →

shape: (2, 3)

3D array



shape: (4, 3, 2)





Numpy

# Dimensiones

- Número de puntos de referencia necesarios para extraer un valor de la estructura de datos



Temperature	
New York	38
Chicago	36
San Francisco	51
Miami	73



Numpy

# Dimensiones

- Número de puntos de referencia necesarios para extraer un valor de la estructura de datos

	Monday	Tuesday	Wednesday	Thursday	Friday
New York	38	41	35	32	35
Chicago	36	39	31	27	25
San Francisco	51	52	50	49	53
Miami	73	74	72	71	74





Numpy

# Dimensiones



- Número de puntos de referencia necesarios para extraer un valor de la estructura de datos

Week 1

	Monday	Tuesday	Wednesday	Thursday	Friday
New York	38	41	35	32	35
Chicago	36	39	31	27	25
San Francisco	51	52	50	49	53
Miami	73	74	72	71	74

Week 2

	Monday	Tuesday	Wednesday	Thursday	Friday
New York	40	42	38	36	28
Chicago	32	28	25	31	25
San Francisco	49	55	54	51	48
Miami	75	78	73	76	71



---

## Numpy

# El objeto ndarray

La función **arange** retorna un ndarray unidimensional con una secuencia de valores numéricos

## Parámetros

- start
- stop
- step





Universidad del  
**Rosario**

Educación Continua  
y Consultoría

---

Numpy

# Atributos y métodos

## Atributos

- shape
- ndim
- size

## Métodos

- reshape





Numpy

# Generadoras de números aleatorios

Permiten generar números aleatorios donde se puede personalizar

- La distribución de probabilidad de los números
- El número más pequeño posible
- El número más grande posible
- La cantidad de números aleatorios
- El número de dimensiones

Se puede usar una semilla para el algoritmo generador



Unidad de Educación Continua y Consultoría  
construimos país desde

**#URSolucionesInnovadoras**  
**#URConsultoría**



@RosarioContinua



/EduContinuaURosario



@RosarioContinua