



Universidad del  
**Rosario**

Educación Continua  
y Consultoría

# Diplomado en Ciencia de Datos



#URSolucionesInnovadoras

#URConsultoría



Universidad del  
**Rosario**

Educación Continua  
y Consultoría

PROGRAMACIÓN 1

# **Introducción a la Ciencia de Datos con Python**



---

# Contenido

- ¿Qué es Python?
- Instalación
- Ejecución de código
- Variables, expresiones y operaciones básicas
- Ejecuciones condicionales y cíclicas
- Datos en el Siglo XXI
- La librería pandas
- Análisis movies.csv





Universidad del  
**Rosario**

Educación Continua  
y Consultoría

---

# Contenido

- **¿Qué es Python?**
- **Instalación**
- **Ejecución de código**
- **Variables, expresiones y operaciones básicas**
- **Ejecuciones condicionales y cíclicas**
- **Datos en el Siglo XXI**
- **La librería pandas**
- **Análisis movies.csv**



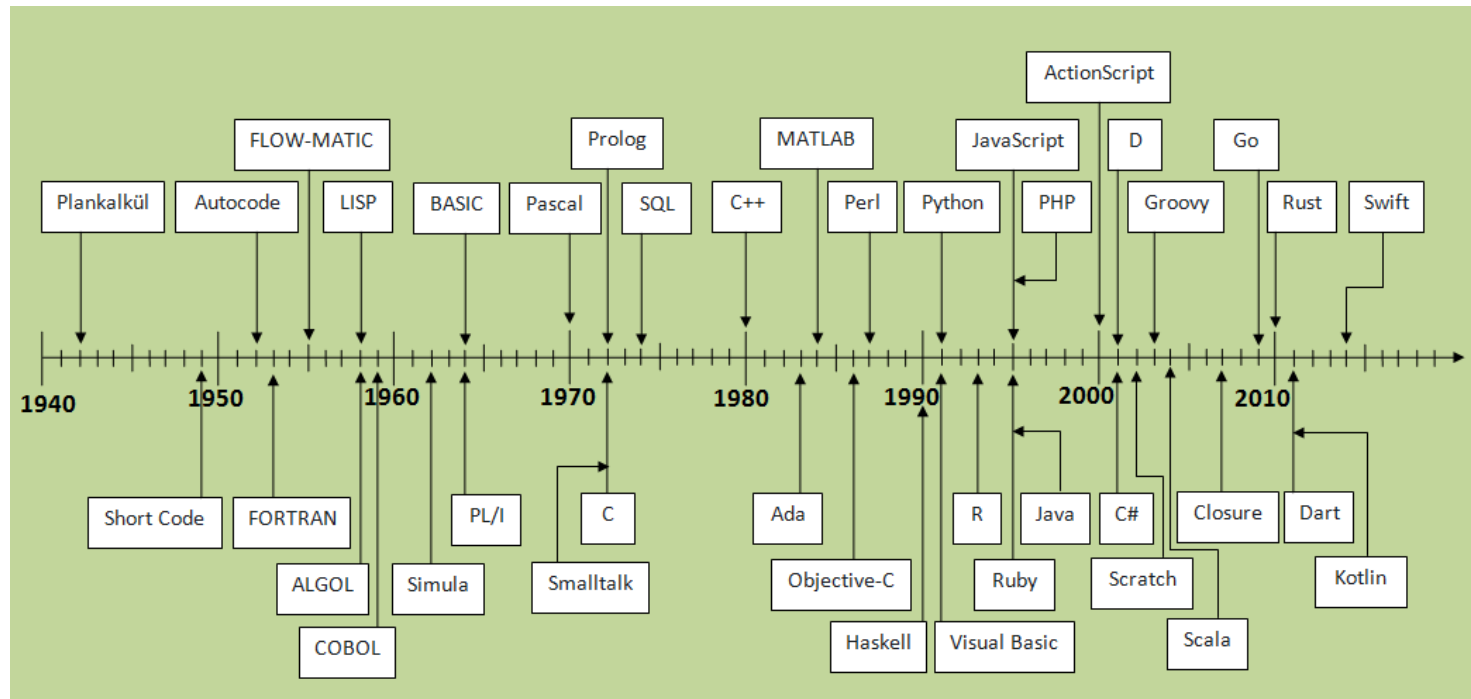


Universidad del  
**Rosario**

Educación Continua  
y Consultoría

¿Qué es Python?

# Historia de los lenguajes de programación



<https://javaconceptoftheday.com/history-of-programming-languages/>



## ¿Qué es Python?

# Características de Python

- Lenguaje de programación **interpretado** de alto nivel

```
package main  
  
import "fmt"  
  
func main() {  
    fmt.Printf("hello, world")  
}
```

Lenguaje de alto nivel que  
entiende el programador



```
0101010111101110001101  
0100010100010101001010  
0101010010101010000101  
0011010001010100011110  
0110010100101010101001  
1110001101010010010001
```

Lenguaje de máquina que  
entiende el procesador

- Multipropósito
  - Data analytics/Big data/Data science/IA/Blockchain
  - Desarrollo web
  - Desarrollo de videojuegos
  - Glue code





---

# Contenido

- ¿Qué es Python?
- **Instalación**
- Ejecución de código
- Variables, expresiones y operaciones básicas
- Ejecuciones condicionales y cíclicas
- Datos en el Siglo XXI
- La librería pandas
- Análisis movies.csv





Universidad del  
**Rosario**

Educación Continua  
y Consultoría

## Instalación

# Python y Entornos de Desarrollo



### Intérprete de Python

- <https://www.python.org/>

### Entornos Integrados de Desarrollo (IDEs)

- <https://www.spyder-ide.org/>
- <https://jupyter.org/try>
- <https://www.jetbrains.com/es-es/pycharm/>
- <https://code.visualstudio.com/>
- <https://www.sublimetext.com/>
- ...

### Entornos Cloud

- <https://colab.research.google.com/>







Universidad del  
**Rosario**

Educación Continua  
y Consultoría

Instalación

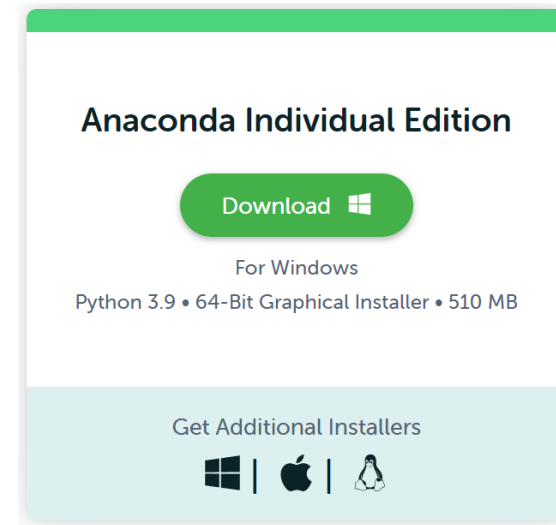
# Anaconda

Intérprete de Python

+

Entornos Integrados de Desarrollo  
(IDEs)

<https://www.anaconda.com/>



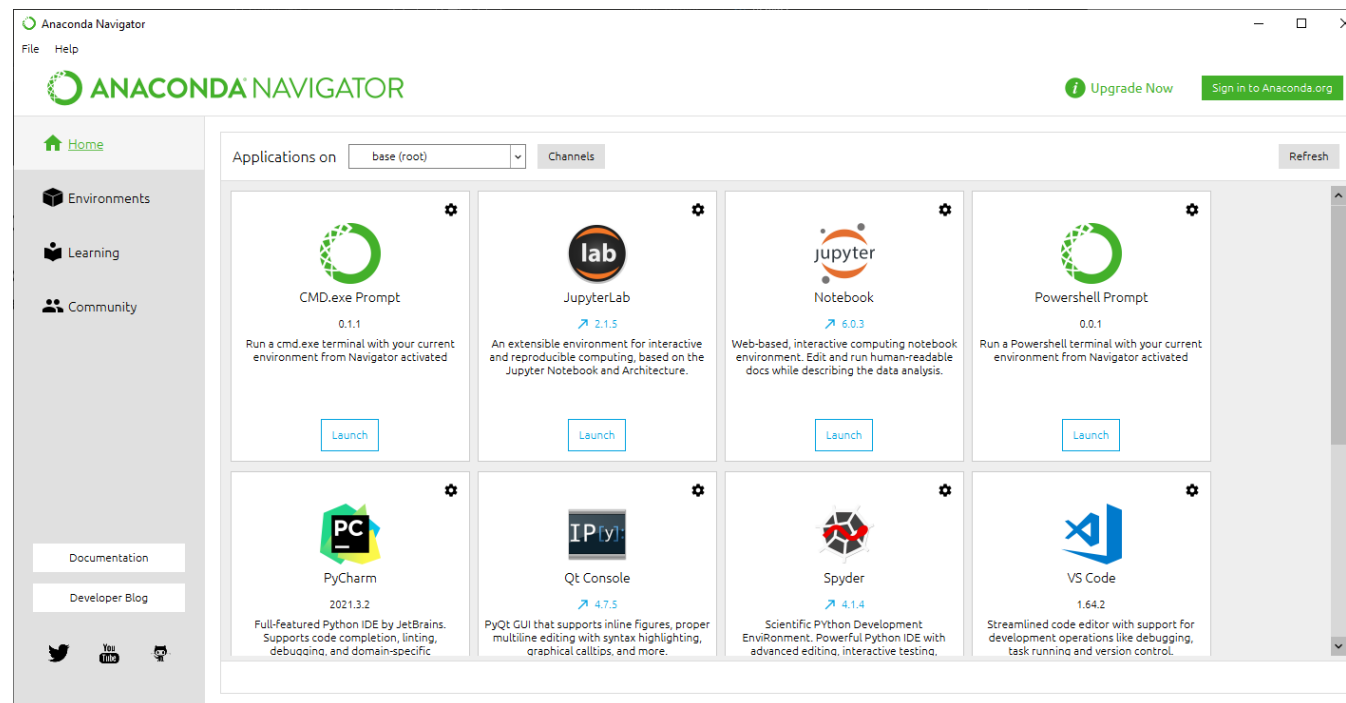
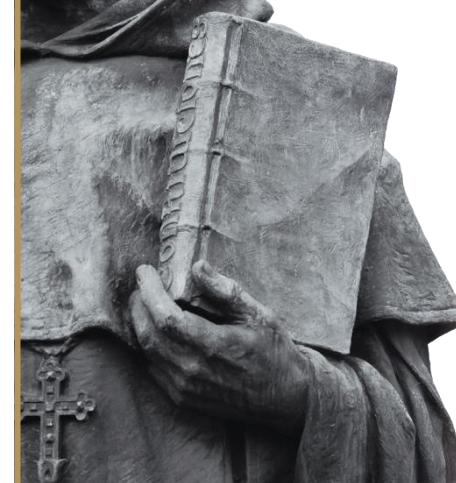


Universidad del  
**Rosario**

Educación Continua  
y Consultoría

## Instalación

# Anaconda





---

# Contenido

- ¿Qué es Python?
- Instalación
- **Ejecución de código**
- Variables, expresiones y operaciones básicas
- Ejecuciones condicionales y cíclicas
- Datos en el Siglo XXI
- La librería pandas
- Análisis movies.csv





Universidad del  
**Rosario**

Educación Continua  
y Consultoría

Ejecución de código

# Directamente en el intérprete

1. Asegurarse de tener el Python instalado
2. Abrir una terminal de línea de comandos

En Windows



3. Ejecutar el comando `python`
4. Salir con el comando `exit()`



En MacOS

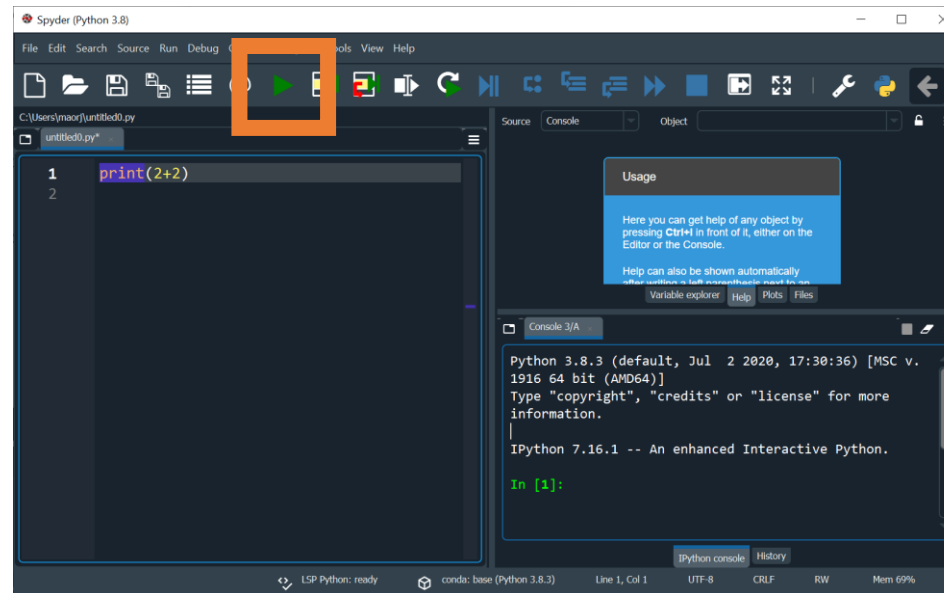




## Ejecución de código

# En spyder

- Asegurarse de tener el Python instalado
- Dar en el botón Play





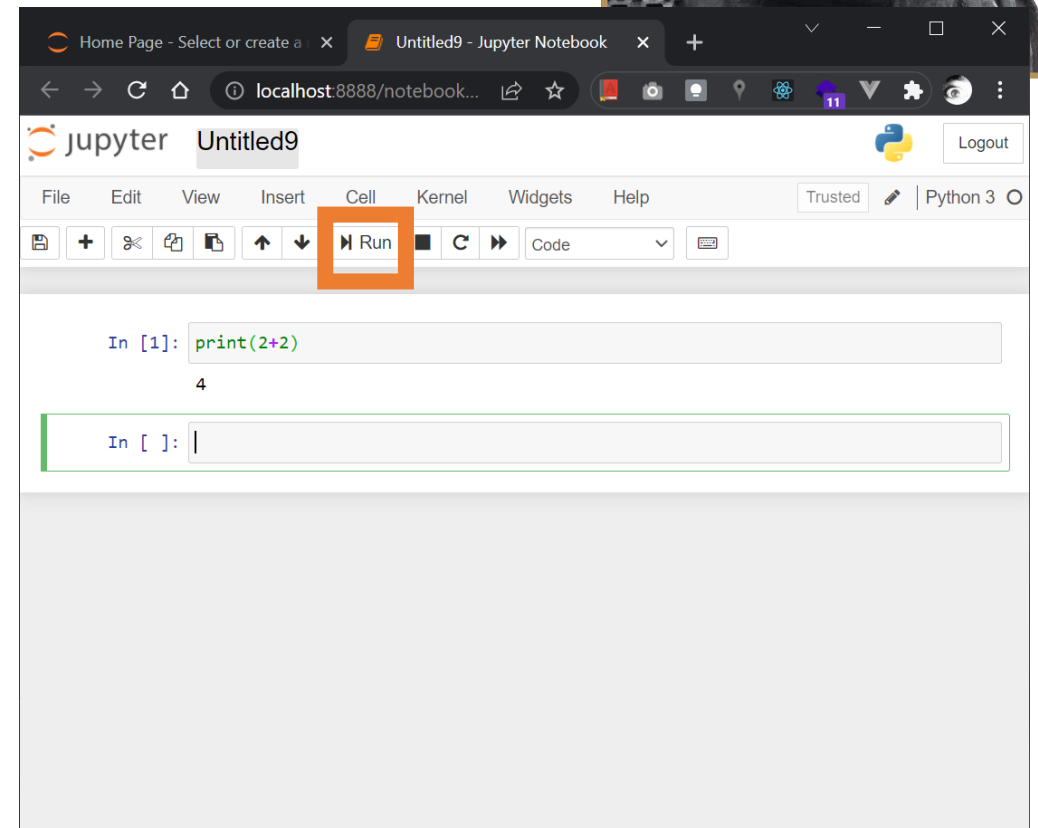
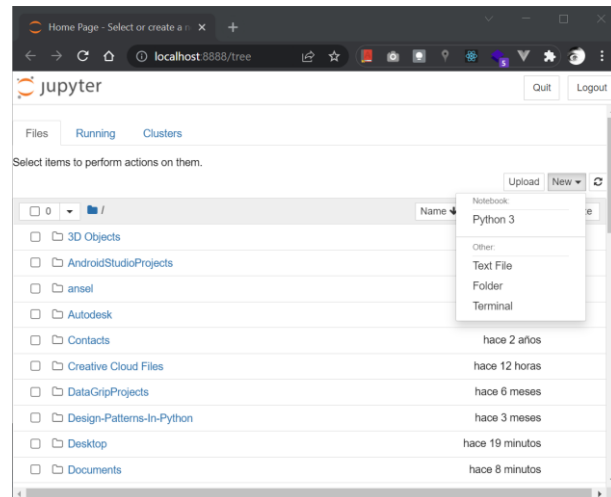
Universidad del  
**Rosario**

Educación Continua  
y Consultoría

## Ejecución de código

# En Jupyter

- Asegurarse de tener el Python instalado
- Dar en el botón Play





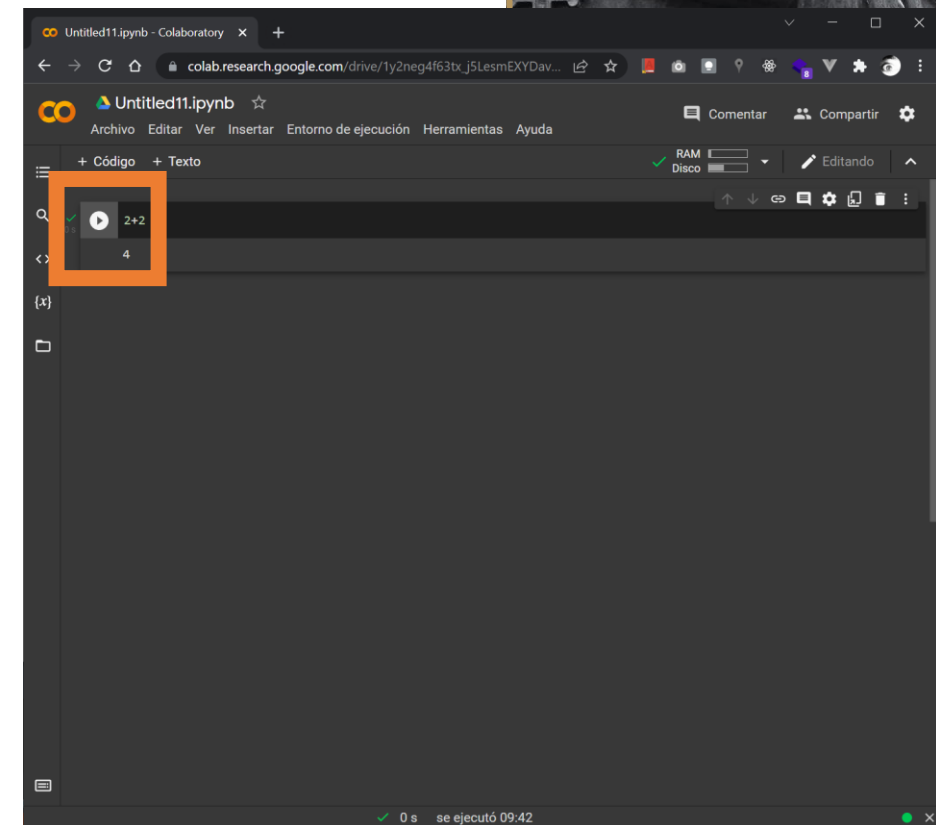
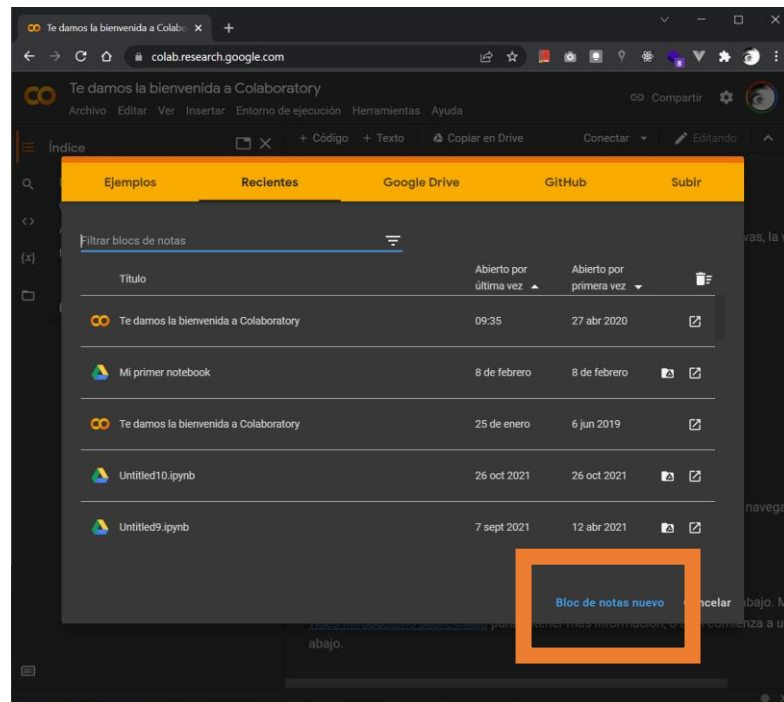
Universidad del  
**Rosario**

Educación Continua  
y Consultoría

## Ejecución de código

# En Colab

■ Igual que Jupyter





---

# Contenido

- ¿Qué es Python?
- Instalación
- Ejecución de código
- **Variables, expresiones y operaciones básicas**
- Ejecuciones condicionales y cíclicas
- Datos en el Siglo XXI
- La librería pandas
- Análisis movies.csv





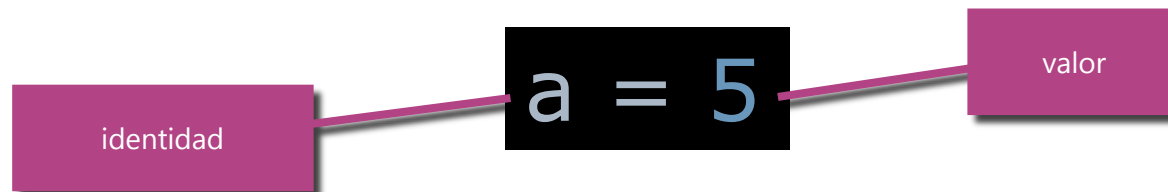


## Variables

# Objetos en Python



- Los objetos son la abstracción de Python para los datos.
- Todos los datos en un programa de Python están representados por **objetos** o por **relaciones entre objetos**.



Cada objeto tiene una *identidad*, un *tipo* y un *valor* asociado



## Variables

# Crear variables

- Cada variable tiene un tipo de datos adecuado para el dato que representa

- Es posible convertir un dato a otro tipo

```
mensaje = "Hola mundo"  
print (mensaje)  
print(type(mensaje))
```

```
n = 10  
print(n)  
print(type(n))
```

```
pi = 3.141592  
print(pi)  
print(type(pi))
```

```
int(pi)  
float(15)  
float("2.7182")
```





## Expresiones

# ¿Qué es una expresión?

Combinación de constantes, variables o funciones, que es interpretada de acuerdo a la sintaxis de un lenguaje particular



```
# Esto es un comentario  
m = 2  
b = 3.5  
x = 3  
y = (m*x)+b  
print(y)
```

```
'''  
Este es un bloque  
de comentarios  
'''
```

```
m = 2  
b = 3.5  
x = 3  
y = (m*x)+b  
print(y)
```



## Expresiones

# Operadores

### Aritméticos

OPERATOR	DESCRIPTION	SYNTAX
+	Addition: adds two operands	$x + y$
-	Subtraction: subtracts two operands	$x - y$
*	Multiplication: multiplies two operands	$x * y$
/	Division (float): divides the first operand by the second	$x / y$
//	Division (floor): divides the first operand by the second	$x // y$
%	Modulus: returns the remainder when first operand is divided by the second	$x \% y$



a = 3 + 2  
b = 5 - 3  
c = 4 \* 8  
d = 7 / 4  
e = 7 // 4  
f = 7 % 4  
g = 3 \*\* 4



## Expresiones

# Entrada de datos



```
n = input("Digite un numero ")  
print(type(n))  
print("El numero es:", n)
```



## Expresiones

# Expresiones booleanas y valores lógicos

### Operadores relacionales

OPERATOR	DESCRIPTION	SYNTAX
>	Greater than: True if left operand is greater than the right	$x > y$
<	Less than: True if left operand is less than the right	$x < y$
==	Equal to: True if both operands are equal	$x == y$
!=	Not equal to - True if operands are not equal	$x != y$
>=	Greater than or equal to: True if left operand is greater than or equal to the right	$x >= y$
<=	Less than or equal to: True if left operand is less than or equal to the right	$x <= y$

### Operadores lógicos

OPERATOR	DESCRIPTION	SYNTAX
and	Logical AND: True if both the operands are true	$x \text{ and } y$
or	Logical OR: True if either of the operands is true	$x \text{ or } y$
not	Logical NOT: True if operand is false	$\text{not } x$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Operador Y

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Operador Ó

A	B
0	1
1	0

Operador NO

```
x = 10, y = 20, z = 30
x < y and z % 2 == 0
x % y == 0 or z < y
```



---

# Contenido

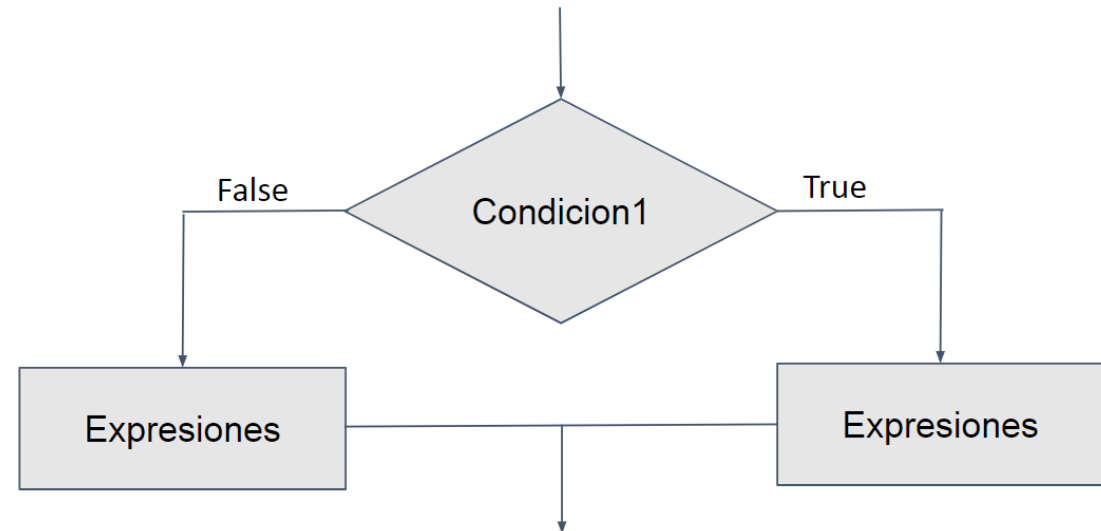
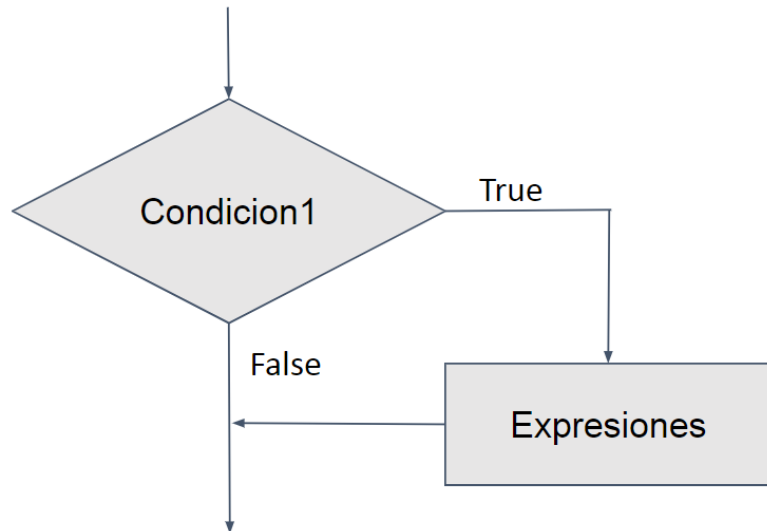
- ¿Qué es Python?
- Instalación
- Ejecución de código
- Variables, expresiones y operaciones básicas
- **Ejecuciones condicionales y cíclicas**
- Datos en el Siglo XXI
- La librería pandas
- Análisis movies.csv





## Ejecuciones condicionales y cíclicas

# Ejecución condicional

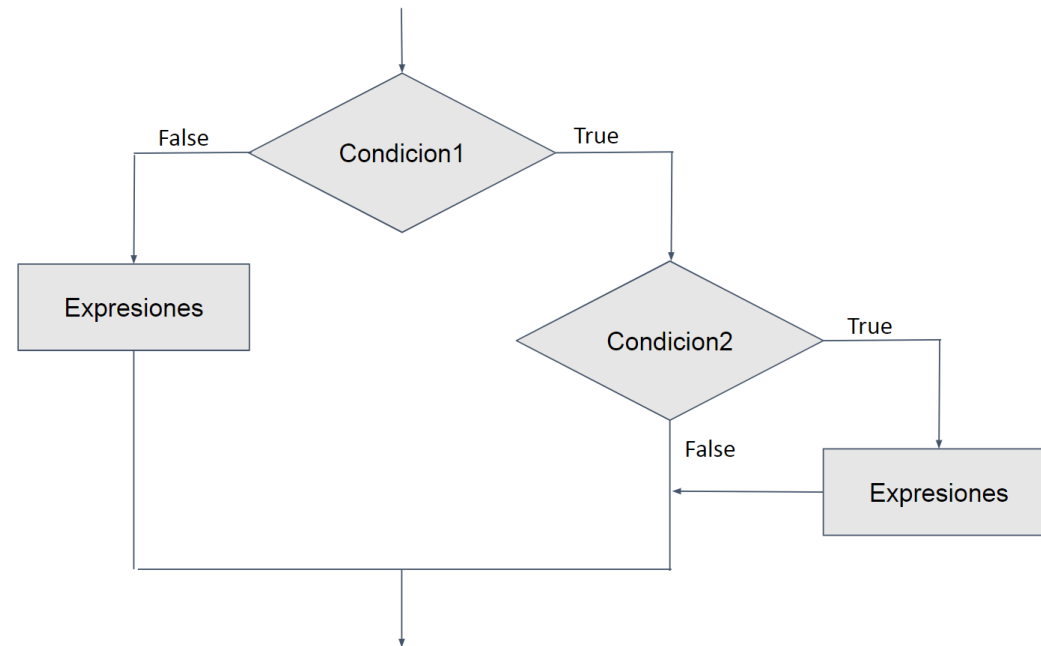






## Ejecuciones condicionales y cíclicas

# Ejecución condicional





## Ejecuciones condicionales y cíclicas

# Ejemplos: Ejecución condicional



```
x = int(input("Digite numero: "))
if x % 2 == 0:
    print(x, " es par.")
else:
    print(x, " es impar.")
```

```
x = int(input("Digite numero: "))
if x % 2 == 0:
    print(x, " es divisible por 2.")
elif x % 3 == 0:
    print(x, " es divisible por 3.")
else:
    print(x, " no es divisible por 2 ni por 3.")
```

```
edad = int(input("Digite su edad: "))
if edad <= 18:
    print("Usted es menor de edad")
elif edad > 18 and edad < 60:
    print("Usted es mayor de edad")
else:
    print("Usted es adulto mayor")
```

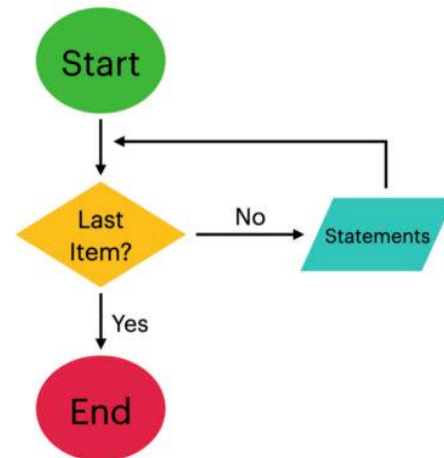


## Ejecuciones condicionales y cíclicas

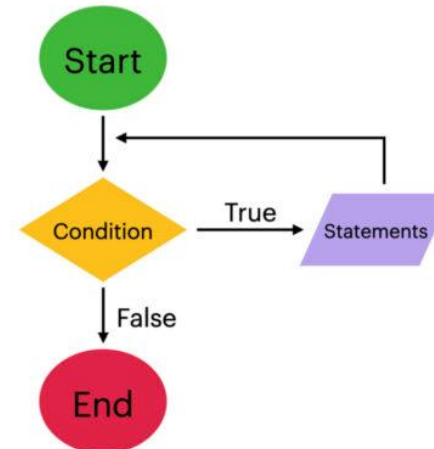
# Ejecución cíclica



### For Loop



### While Loop





## Ejecuciones condicionales y cíclicas

# Ejemplos: Ejecución cíclica

```
for i in range(0, 10):  
    print (i)  
  
for i in range(10):  
    print (i)  
  
for i in range(0, 10, 2):  
    print (i)  
  
lista = [10, 20, 30, 40]  
for i in lista:  
    print (i)
```

```
i = 1  
while i < 6:  
    print(i)  
    i += 1
```





Universidad del  
**Rosario**

Educación Continua  
y Consultoría

# Algunos retos





Universidad del  
**Rosario**

Educación Continua  
y Consultoría

---

Algunos retos

## Reto # 1

- Encontrar la sumatoria de todos los múltiplos de 3 o 5 menores a 1000





Universidad del  
**Rosario**

Educación Continua  
y Consultoría

---

Algunos retos

## Reto # 2

- Encontrar los números primos menores que 100





## Reto # 3

■ Er  
dc  
té  
in



1 pareja

1 pareja

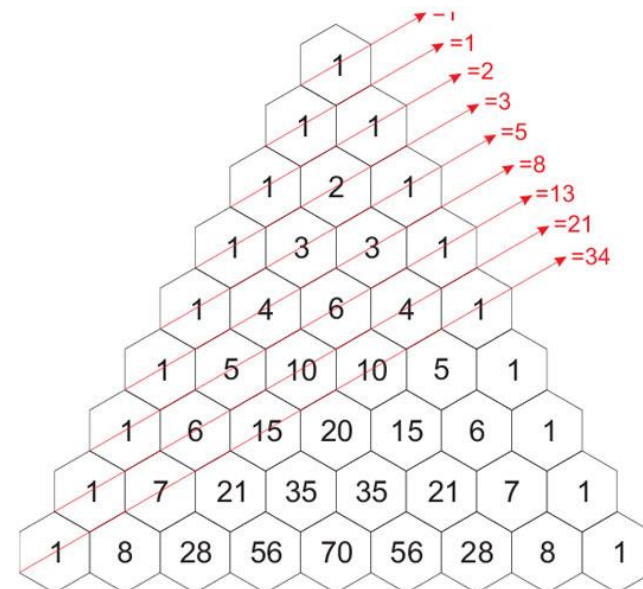
2 parejas

3 parejas

5 parejas

8 parejas

a térr  
sidera  
térmir



do los  
os son





Universidad del  
**Rosario**

Educación Continua  
y Consultoría

---

# Contenido

- ¿Qué es Python?
- Instalación
- Ejecución de código
- Variables, expresiones y operaciones básicas
- Ejecuciones condicionales y cíclicas
- **Datos en el Siglo XXI**
- La librería pandas
- Análisis movies.csv





Universidad del  
**Rosario**

Educación Continua  
y Consultoría

## Datos en el siglo XXI

# The world's most valuable resource is no longer oil, but data

The data economy demands a new approach to antitrust rules



May 6th 2017 (Updated May 11th 2017)

Share

A NEW commodity spawns a lucrative, fast-growing industry, prompting antitrust regulators to step in to restrain those who control its flow. A century ago, the resource in question was oil. Now similar concerns are being raised by

<https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data>





Universidad del  
**Rosario**

Educación Continua  
y Consultoría

---

# Contenido

- ¿Qué es Python?
- Instalación
- Ejecución de código
- Variables, expresiones y operaciones básicas
- Ejecuciones condicionales y cíclicas
- Datos en el Siglo XXI
- **La librería pandas**
- Análisis movies.csv





Universidad del  
**Rosario**

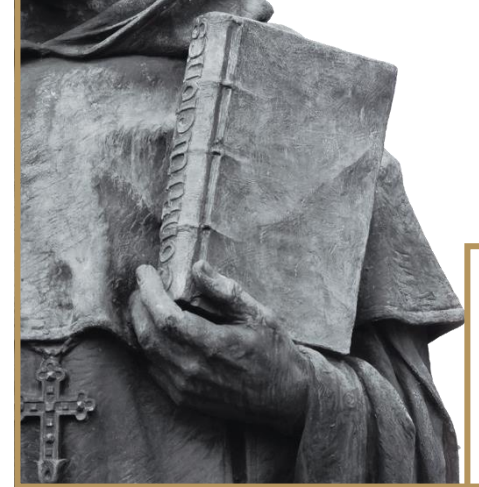
Educación Continua  
y Consultoría

La librería pandas

# ¿Qué es pandas?



<https://pandas.pydata.org/>



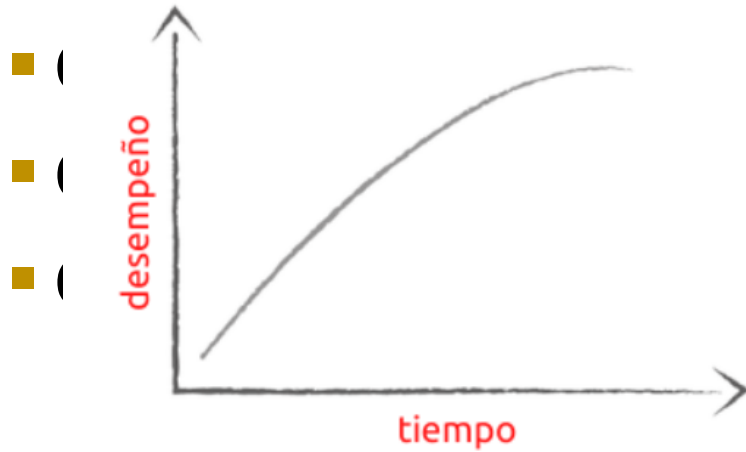


Universidad del  
**Rosario**

Educación Continua  
y Consultoría

Pandas vs. Excel

# Diferencias en mindsets





Universidad del  
**Rosario**

Educación Continua  
y Consultoría

## Pandas vs. Excel

# Similitudes en mindset

## Uso de funciones

- Identificar la función correcta
- Darle los inputs correctos y en orden

## Ejemplos en

- SUMIF
- VLOOKUP







Universidad del  
**Rosario**

Educación Continua  
y Consultoría

Pandas vs. Excel

# El poder de Python + pandas

- Lenguaje de comunicación con el programa más verboso
- Jupyter o Colab
- Ecosistema de data science

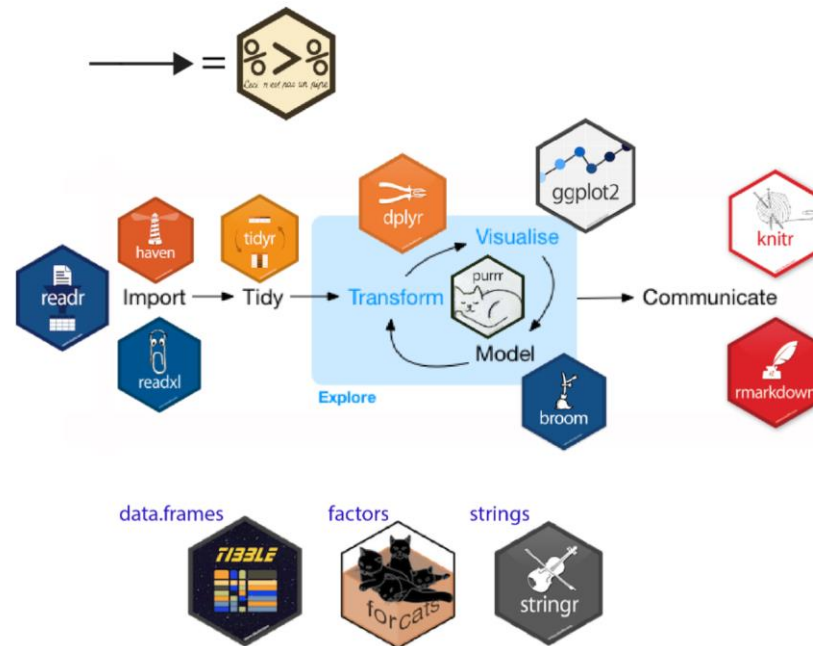




## Pandas vs. Excel

# Competidores de pandas

- RStudio y su tidyverse







## Instalación

# Una simple línea de código

Se usa la terminal (MacOS, Linux) o el CMD (Windows)

- Para instalaciones básicas de Python  
`pip install pandas`
- Si se cuenta con anaconda  
`conda install pandas`

```
Terminal — top — Pro — 94x24
Processes: 421 total, 2 running, 419 sleeping, 1383 threads
Load Avg: 1.49, 1.20, 1.14 CPU usage: 0.54% user, 0.96% sys, 98.49% idle
SharedLibs: 515M resident, 86M data, 305M linkedit.
MemRegions: 98214 total, 2353M resident, 242M private, 1558M shared.
PhysMem: 14G used (2687M wired), 18G unused.
VM: 2461G vsize, 2277M framework vsize, 0(0) swaptins, 0(0) swapouts.
Networks: packets: 168275/66M in, 191369/56M out.
Disks: 246661/3854M read, 146809/3327M written.

PID  COMMAND   %CPU  TIME    #TH  #PR  TommyPowerShell \(\P\) www.thomasmaurer.ch x
147  WindowServer 6.7  15:40.52 14  5  0  PowerShell 6.2.1
3725 top         3.2  00:17.37 1  0  0  Copyright (c) Microsoft Corporation. All rights reserved.
3724 top         3.1  00:17.41 1  0  0
3723 top         3.0  00:17.39 1/1 0  0
2947 Terminal   1.7  00:28.08 6  1  0  https://aka.ms/pscore6-docs
0    kernel_task 1.7  03:48.16 270/16 0  0  type 'help' to get help.
260  mtrecorder.i 1.2  01:46.76 2  1  0
3123 com.apple.Ap 0.9  00:37.87 3  2  0  OS Version: Microsoft Windows 10.0.18922 PS Version: 6.2.1
81   powerd      0.1  00:07.37 3  2  0  PS C:\Users\thmaure> .\important.ps1

Emoji support 🤖
Testing and getting results

Running:
Pre-requisites ..... 3/3 ✓ Pass
Validate Data ..... 5/5 ✓ Pass
Process Data ..... 8/8 ✓ Pass
Generate Results ..... 2/3 🟡 Partial
Validate Results ..... 1/3 🔴 Fail

PS C:\Users\thmaure>
```





Universidad del  
**Rosario**

Educación Continua  
y Consultoría

---

# Contenido

- ¿Qué es Python?
- Instalación
- Ejecución de código
- Variables, expresiones y operaciones básicas
- Ejecuciones condicionales y cíclicas
- Datos en el Siglo XXI
- La librería pandas
- **Análisis movies.csv**





▪Análisis movies.csv

# Actividad

1. Verificar la instalación de pandas
2. Abrir algún IDE de Python
3. Cargar pandas
4. Cargar el archivo de datos movies.csv  
<https://drive.google.com/file/d/18cigQy12DAqdwOd07iW0aiM9ZWE1JfUM/view?usp=sharing>
5. Analizar el dataset de películas





Universidad del  
**Rosario**

Educación Continua  
y Consultoría

## ■ Análisis movies.csv

# Fuentes de ayuda

- Las hojas de trucos (Google: pandas cheatsheet)

## 1. The Most Comprehensive Cheat Sheet



- Stackoverflow

**Data Wrangling with pandas Cheat Sheet**  
<http://pandas.pydata.org>

**Tidy Data** – A foundation for wrangling in pandas

In a tidy data set:

- Each variable is saved in its own column
- Each observation is saved in its own row

Tidy data complements pandas's vectorized operations. pandas will automatically preserve observations as you manipulate variables. No other format works as intuitively with pandas.

**Reshaping Data** – Change the layout of a data set

**Syntax – Creating DataFrames**

```
df = pd.DataFrame({
    "a": [4, 5, 6],
    "b": [7, 8, 9],
    "c": [10, 11, 12]},
    index = [1, 2, 3])
```

Specify values for each column.

```
df = pd.DataFrame(
    [[4, 7, 10],
     [5, 8, 11],
     [6, 9, 12]],
    index=[1, 2, 3],
    columns=['a', 'b', 'c'])
```

**pd.melt(df)**  
Gather columns into rows.

**df.pivot(columns='var', values='val')**  
Spread rows into columns.

**pd.concat([df1, df2])**  
Append rows of DataFrames

**pd.concat([df1, df2], axis=1)**  
Append columns of DataFrames

**df.sort\_values('mpg')**  
Order rows by values of a column (low to high).

**df.sort\_values('mpg', ascending=False)**  
Order rows by values of a column (high to low).

**df.rename(columns = {'y':'year'})**  
Rename the columns of a DataFrame

**df.sort\_index()**  
Sort the Index of a DataFrame

**df.reset\_index()**  
Reset index of DataFrame to row numbers, moving index to columns.

**df.drop(columns=['Length', 'Height'])**  
Drop columns from DataFrame



Unidad de Educación Continua y Consultoría  
construimos país desde

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



@RosarioContinua



/EduContinuaURosario



@RosarioContinua