

Data Fundamentals



red.es

Centro de
Referencia Nacional
en Comercio Electrónico
y Marketing

CRN
Digital

GARANTÍA
JUVENIL



Barrabés

The Valley

"El FSE invierte en tu futuro"
Fondo Social Europeo

Data Fundamentals



Data Fundamentals



red.es

Centro de
Referencia Nacional
en Comercio Electrónico
y Marketing

CRN
Digital

GARANTÍA
JUVENIL



Barrabés

The Valley

"El FSE invierte en tu futuro"
Fondo Social Europeo

Índice

1. Python

1. Lenguaje de programación
2. ¿Por qué Python?

2. IDE

1. ¿Qué es un IDE?
2. Ventajas de usar un IDE
3. CRISP-DM

3. Variables

4. Funciones

5. Librerías y PIP

1. Funciones (tipos)
2. Librerías
3. PIP

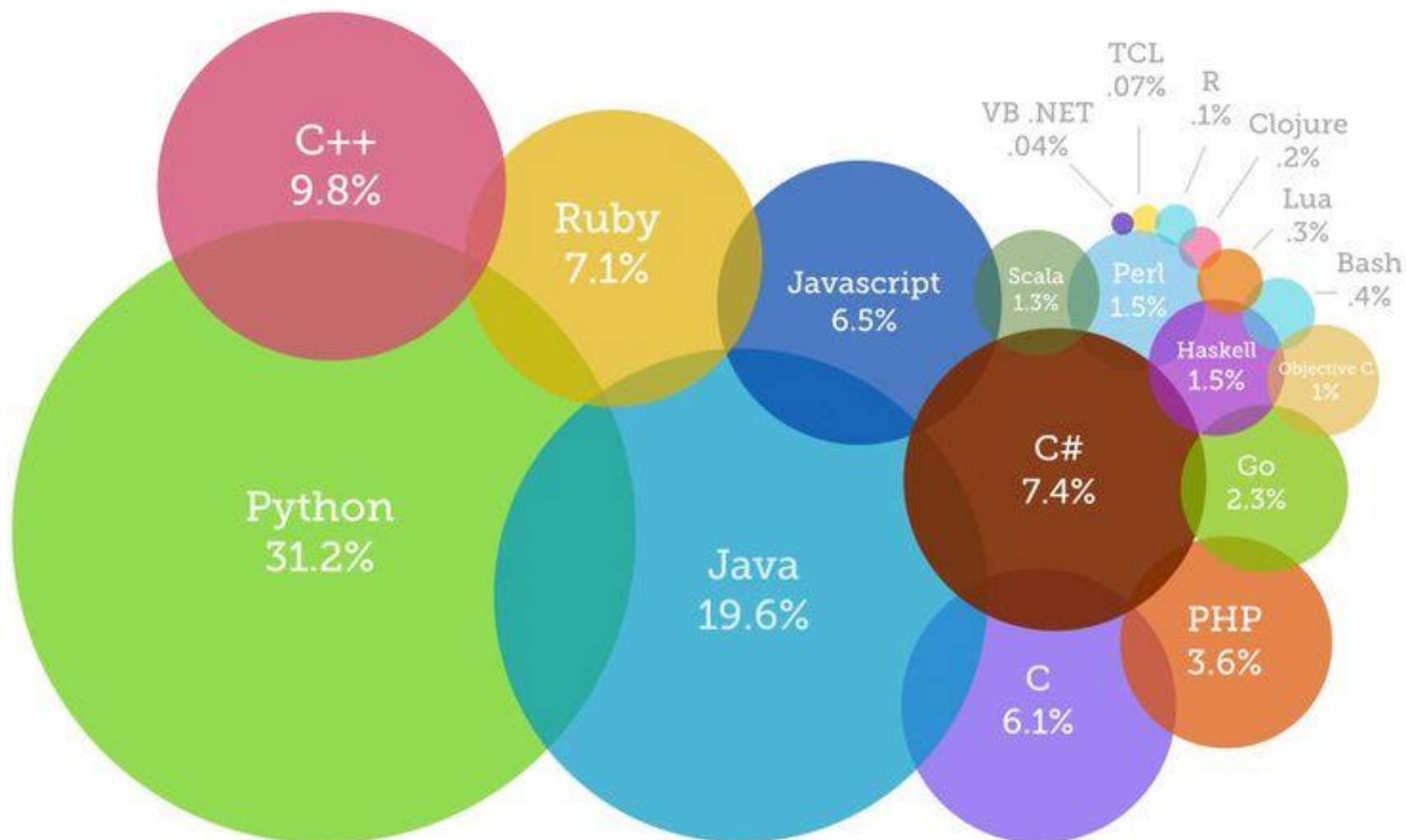


TÍTULO DE LA PRESENTACIÓN

Python



Selección del lenguaje de programación



¿Por qué Python?

*Lenguaje de **alto nivel**, fácil de aprender*

Expresivo y legible

Sintaxis elegante y tipado dinámico y fuerte

Multiparadigma

Interpretado

Multiplataforma

“APRENDER A MONTAR EN BICICLETA”



PYTHON

Aprender un lenguaje de programación



IDE:

¿Qué es un IDE?



¿Qué es un IDE?

IDE

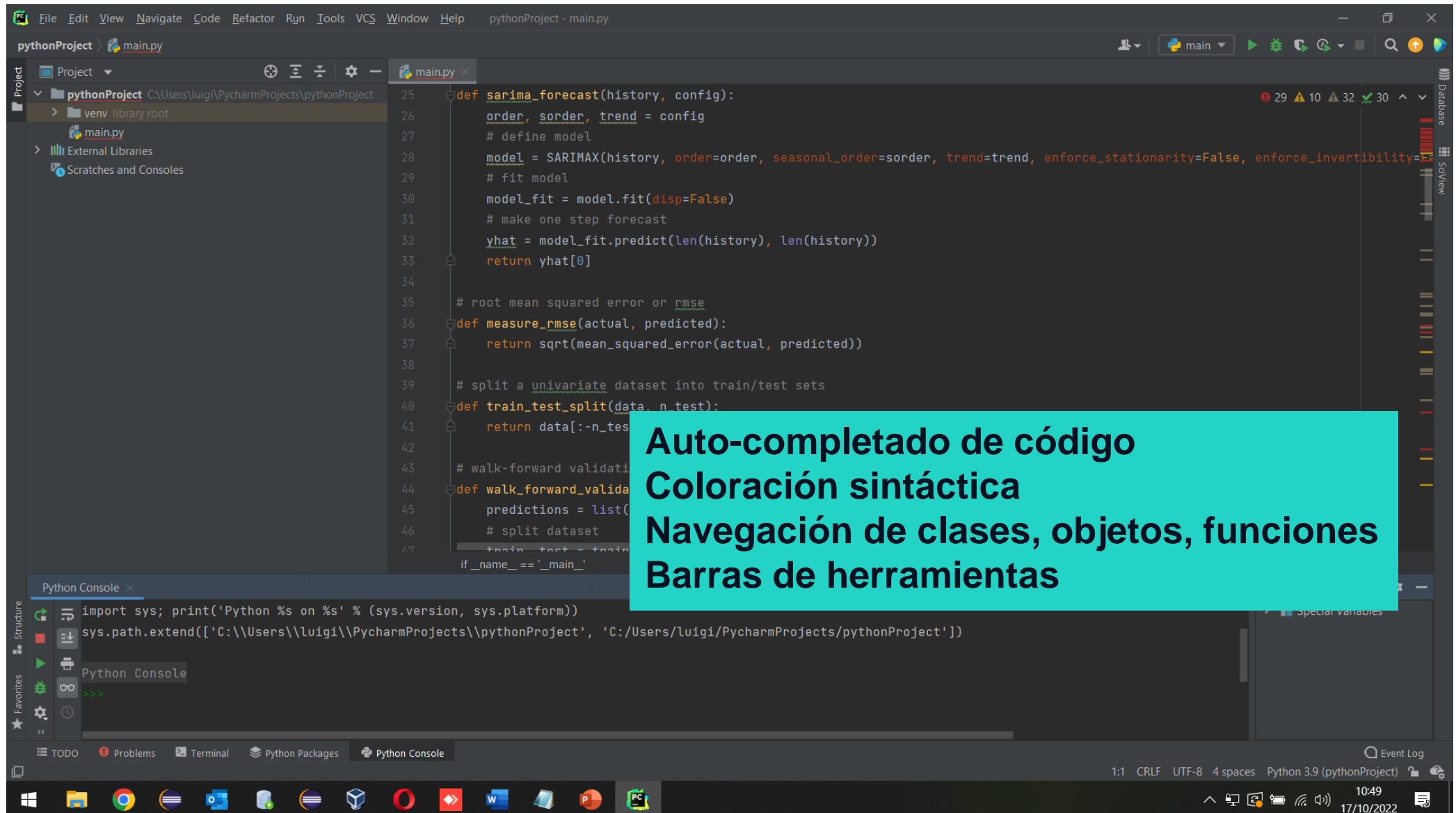
Es un software que nos permite desarrollar de una forma **más fácil**.



```
C:\Windows\system32\cmd.exe - py
D:\>py
Python 3.7.0 (default, Jun 28 2018, 08:04:48) [MSC v.1912 64 bit (AMD64)]
conda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 10 + 8
18
>>>
>>> 90 / 7
12.857142857142858
>>>
>>> 90 // 7
12
>>>
```



Ventajas de usar un IDE:



CRISP-DM

| Business Understanding | Data Understanding | Data Preparation | Modeling | Evaluation | Deployment |
|--|--|---|---|--|--|
| Determine Business Objectives <i>Background</i> <i>Business Objectives</i> <i>Business Success Criteria</i> | Collect Initial Data <i>Initial Data Collection Report</i> | Select Data <i>Rationale for Inclusion/Exclusion</i> | Select Modeling Techniques <i>Modeling Technique</i> <i>Modeling Assumptions</i> | Evaluate Results <i>Assessment of Data Mining Results w.r.t. Business Success Criteria</i> <i>Approved Models</i> | Plan Deployment <i>Deployment Plan</i> |
| Assess Situation <i>Inventory of Resources</i> <i>Requirements, Assumptions, and Constraints</i> <i>Risks and Contingencies</i> <i>Terminology</i> <i>Costs and Benefits</i> | Describe Data <i>Data Description Report</i> | Clean Data <i>Data Cleaning Report</i> | Generate Test Design <i>Test Design</i> | Review Process <i>Review of Process</i> | Plan Monitoring and Maintenance <i>Monitoring and Maintenance Plan</i> |
| Determine Data Mining Goals <i>Data Mining Goals</i> <i>Data Mining Success Criteria</i> | Explore Data <i>Data Exploration Report</i> | Construct Data <i>Derived Attributes</i> <i>Generated Records</i> | Build Model <i>Parameter Settings</i> <i>Models</i> <i>Model Descriptions</i> | Determine Next Steps <i>List of Possible Actions</i> <i>Decision</i> | Produce Final Report <i>Final Report</i> <i>Final Presentation</i> |
| Produce Project Plan <i>Project Plan</i> <i>Initial Assessment of Tools and Techniques</i> | Verify Data Quality <i>Data Quality Report</i> | Integrate Data <i>Merged Data</i> | Assess Model <i>Model Assessment</i> <i>Revised Parameter Settings</i> | | Review Project <i>Experience</i> <i>Documentation</i> |
| | | Format Data <i>Reformatted Data</i> <i>Dataset</i> <i>Dataset Description</i> | | | |

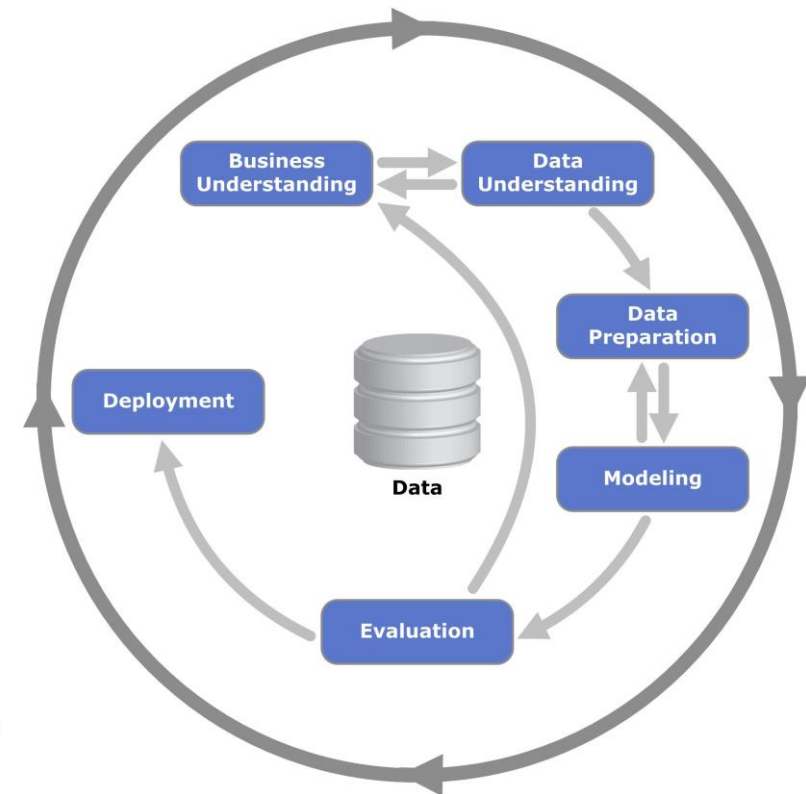


Figure 3: Generic tasks (bold) and outputs (italic) of the CRISP-DM reference model

VARIABLES

Una variable es donde se almacenan y se recuperan los datos de un programa.

Existen distintos tipos de variables: texto, número, booleanos...



```
1 #Ejemplos de Variables
2 nombre = 'Luis'
3 dia = 2
4 soleado = False
```

```
nombreVariable1 = valor1
nombreVariable2 = valor2
...
```


LISTAS

```
>>> jugadores = ['Batalla', 'Driussi', 'Alario', 'Mora', 'Auzqui', 'Rojas', 'Ponzio']
>>> jugadores
['Batalla', 'Driussi', 'Alario', 'Mora', 'Auzqui', 'Rojas', 'Ponzio']
>>> len(jugadores)
7
>>> jugadores[0]
'Batalla'
>>> jugadores[6]
'Ponzio'
>>> jugadores[-1]
'Ponzio'
>>> jugadores[3]
'Mora'
>>>
```

```
>>> lista = ["RIN", "LOIRA", "TAJO", "DANUBIO", "ELBA"]
>>> lista.insert(2, "TÍBER")
>>> lista
['RIN', 'LOIRA', 'TÍBER', 'TAJO', 'DANUBIO', 'ELBA']
>>> lista = ["RIN", "LOIRA", "TAJO", "DANUBIO", "ELBA"]
>>> lista + ["TÍBER"] ①
['RIN', 'LOIRA', 'TAJO', 'DANUBIO', 'ELBA', 'TÍBER']
```

DATA FUNDAMENTALS

FUNCIONES



MALAMADO

Receta: Inés De Los Santos

2 oz de Whisky Bourbon
1 oz de Malamado Viognier

1 dash de almíbar

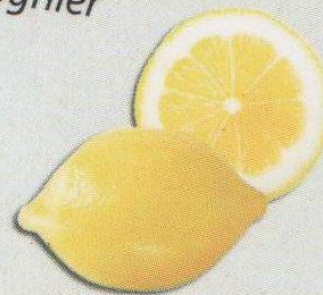
2 gajos de lima

2 gajos de limón

Gaseosa limón

Método: batido

Vaso super highball



En una coctelera aplastar los gajos de lima y limón, luego colocar los hielos y batir todos los ingredientes (menos la gaseosa). Servir en un vaso super highball y terminar con gaseosa lima-limón.

FUNCIONES

Una función te permite definir un bloque de **código reutilizable** que se puede ejecutar muchas veces dentro de tu programa.

Una de las grandes ventajas de usar funciones en tu código es que reduce el número total de líneas de código en tu proyecto.

```
1 #Ejemplo de función
2 def cuadrado(n):
3     return n*n
4
5 cuadrado(2)
```

4



```
def nombreFuncion( ):
    instruc1
    instruc2
    ...
```

¿Para qué repetir una y otra vez bloques de código una y otra vez?

```
# función sin parámetros o retorno de valores
def diHola():
    print("Hello!")

diHola() # llamada a la función, 'Hello!' se muestra en la consola

# función con un parámetro
def holaConNombre(name):
    print("Hello " + name + "!")

holaConNombre("Ada") # llamada a la función, 'Hello Ada!' se muestra en la consola
```

Por parámetro y sin parámetro (ARGUMENTOS)

FUNCIONES (Python)

Funciones Built-in

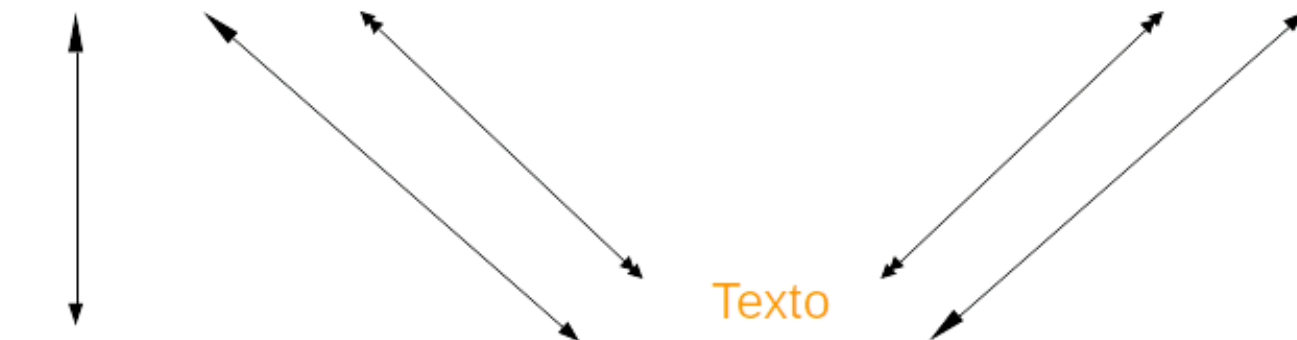
El intérprete de Python tiene una serie de funciones y tipos incluidos en él que están siempre disponibles. Están listados aquí en orden alfabético.

| | | Funciones Built-in | | |
|---------------|--|--------------------|--------------|--------------|
| abs() | delattr() | hash() | memoryview() | set() |
| all() | dict() | help() | min() | setattr() |
| any() | dir() | hex() | next() | slice() |
| ascii() | <u>Funciones nativas y propias:</u> | | | |
| bin() | | | | |
| bool() | eval() | int() | open() | str() |
| breakpoint() | exec() | isinstance() | ord() | sum() |
| bytearray() | <u>¡YA PODEMOS USARLAS!</u> | | | |
| bytes() | | | | |
| callable() | format() | len() | property() | type() |
| chr() | frozenset() | list() | range() | vars() |
| classmethod() | getattr() | locals() | repr() | zip() |
| compile() | globals() | map() | reversed() | __import__() |
| complex() | hasattr() | max() | round() | |

FUNCIONES (Python)

Función Print (imprimir):

Print ("Hola, esto es una cadena de texto")



Función

Texto

Parámetros

Función ()
Función (Parámetros)
Función ("TEXTO")



DATA FUNDAMENTALS

FUNCIONES (Python)

Pip es nuestra gran aliada

NOSOTROS USAREMOS ESTO

Las librerías serán nuestro mayor aliado

import solucion



[Help](#)[Sponsors](#)[Log in](#)[Register](#)

Find, install and publish Python packages with the Python Package Index



Or [browse projects](#)

352,453 projects

3,180,044 releases

5,487,650 files

566,399 users



Contacto:

luis@lubay.es





red.es



UNIÓN EUROPEA

"El FSE invierte en tu futuro"

Fondo Social Europeo

