



Introducción a la programación con Python



Caso: Mi Red Social
Elían Laura

Entornos de Programación Python

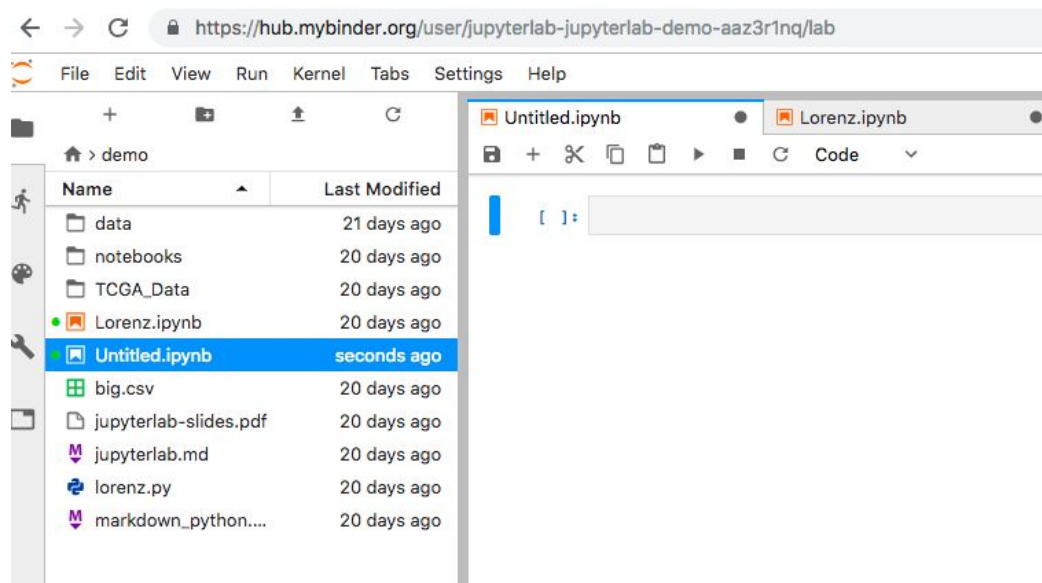
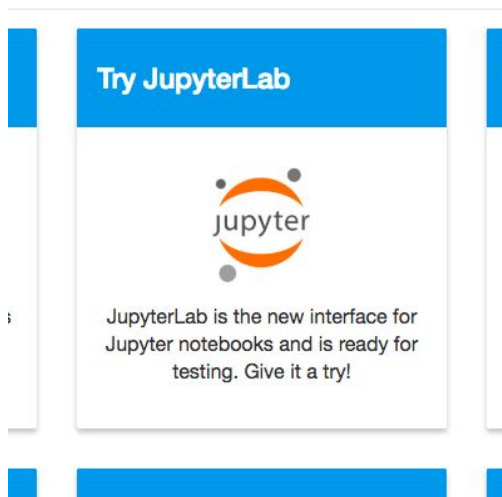


¿Cómo usar Jupyter Notebook Online?



Try it in your browser

Install the Notebook



Comencemos!

<https://goo.gl/Qct1xr>



Mi Red
Social

Etapas del proyecto práctico:

- **Módulo 1:** Recolección de datos de usuario y publicación de mensajes de estado.
- **Módulo 2:** Integración de un menú que permita efectuar distintas acciones sobre la red.
- **Módulo 3:** Separación del código en módulos que agregan distintas funciones y hacen el código más fácil de entender y modificar.



El proyecto tendrá las siguientes etapas:

- **Módulo 1:** Recolección de datos de usuario y publicación de mensajes de estado.
- **Módulo 2:** Integración de un menú que permita efectuar distintas acciones sobre la red.
- **Módulo 3:** Separación del código en módulos que agregan distintas funciones y hacen el código más fácil de entender y modificar.
- **Módulo 4:** Integración de uso de archivos para recordar y almacenar los datos de un usuario.
- **Módulo 5:** Incorporación de listas de amigos y transmisión de mensajes entre ellos.



Árbol de Navidad

Actividad 1:

Genera un árbol de
Navidad con la librería
TURTLE



1 - navidad.py

Actividad 1:

```
1 print("Bienvenido a ... ")
```

```
2 print('hi')
```

3. _____ () _____ / /

4 $\int \frac{1}{x^2} dx = -\frac{1}{x} + C$

5 / / / / / / / / / / / / / / / /

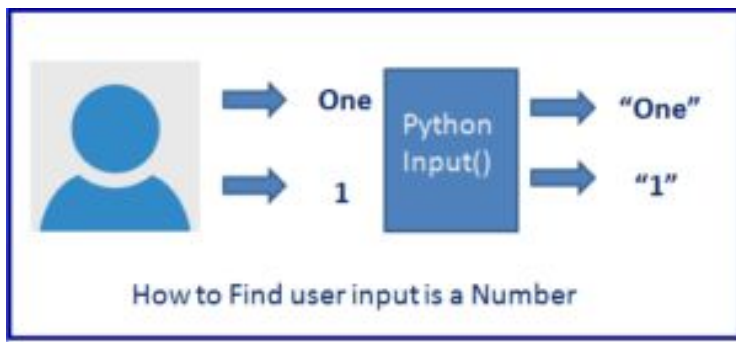
6 /_/_/_/_/_/_/_/_/_/_

7

8 11 11)

9

<http://www.network-science.de/ascii/>



Actividad 2

2 - input.py

```
In[1]: message = "Welcome to Python"  
print(message)
```

Escribe 3 **solicitudes** de datos al usuario, por ejemplo:

- género
- número de teléfono
- ciudad donde vive
- país de nacimiento
- dirección

Guarda estos datos en **variables**

Escríbelos por pantalla utilizando la función **"print"**.



Actividad 3

while
if/elif/else

3 - ciclos y condiciones.py

1. Este programa termina cada vez que el valor de la variable 'escribir_mensaje' es distinto a "S" ó a "s".

Modifique el programa para que termine ÚNICAMENTE cuando se ingresa "N" o "n".

En caso que se ingrese algo distinto, debe **volver a solicitar** una opción al usuario.



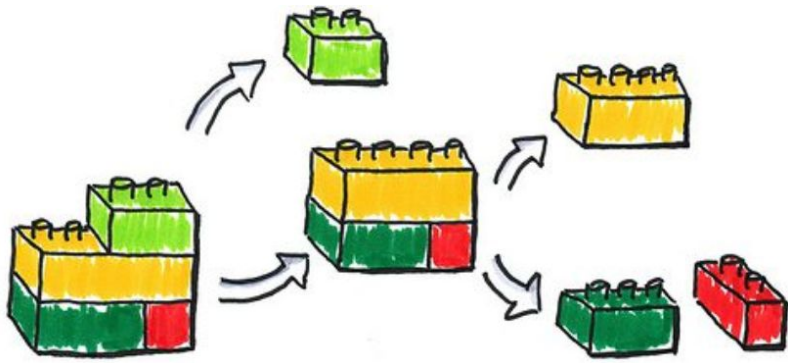
Actividad 3

while
if/elif/else

3 - ciclos y condiciones.py

2. Modifica este menú para que le permita el usuario realizar más de una acción.

Por ejemplo, puedes agregar una acción que permita al usuario modificar su nombre.



Actividad 4

funciones I

4 - funciones.py

4.1 Cuando tenemos **instrucciones que se repiten tantas veces** en distintas partes del programa, es una indicación de que tal vez necesitamos agregar funciones.

Te invitamos a pensar en al menos 3 alternativas o funcionalidades de este código que podrían convertirse en una **función**.



Actividad 5

funciones II

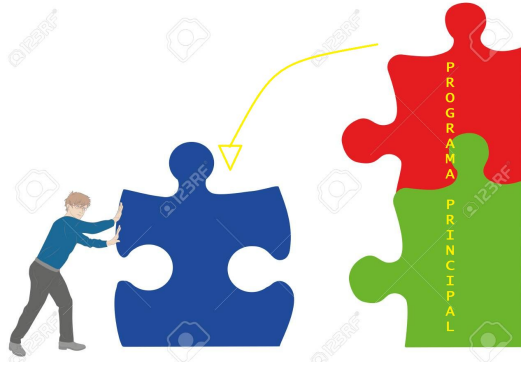
5 - funcionesII.py

5.1

Agrega los atributos "**género**" y "**pais de nacimiento**" (no pongas tilde en "país" en tu código) a los datos que se le piden al usuario.

Realiza un código que solicite datos y los lea utilizando funciones.

Identifica qué partes del código que te facilitamos en "**FuncionesII.py**" debes variar para hacerlo.



Actividad 5

funciones III

6 - funcionesIII.py

6 - main - funcionesIII.py

Ahora ya cuentas con un programa que interactúa con el usuario con un menú, que tiene código encapsulado en funciones y que tiene un programa principal.



Actividad 6

funciones III

solo ejecutar y analizar

6 - funcionesIII.py

6 - main - funcionesIII.py

*Ahora vamos a separar
nuestro código en dos
archivos:*

- 1. El archivo que contiene
solo funciones.*
- 2. El archivo principal que
invoca las funciones*



Actividad 6

funciones III

solo ejecutar y analizar

6 - funcionesIII.py

6 - main - funcionesIII.py

*Ahora gran parte de la
funcionalidad está
encapsulada.*

*Podemos concentrarnos en
funciones más específicas.*

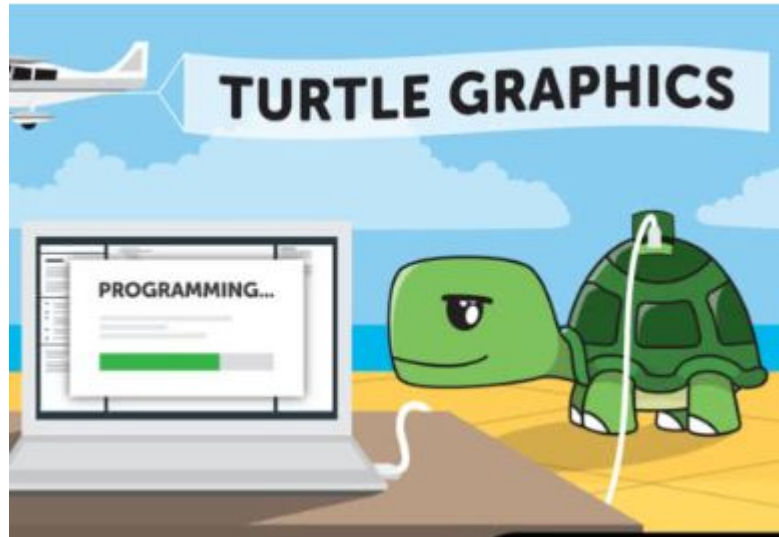
*Esto nos ayuda a razonar a
un nivel más alto.*

Tu Red Social!



Aprende con Talia

<https://www.codeavengers.com/python/100#1.1>



Actividad 7

Listas

7 - listas funciones.py

7 - listas main.py

Fíjate en los archivos "7 - listas funciones.py" y el "7 - listas main.py" para entender los cambios que hemos realizado en el código para esto.

Ejecuta el código "MiREdS6-Listas.py" y prueba las nuevas funciones de lista de amigos y muro para entender bien cómo funcionan. Revisa y examina el código con atención para que no se te escape nada y trata de extenderlo con los ejercicios 1 y 2

Actividad 7

Listas

7 - listas funciones.py

7 - listas main.py

7. 1. Agrega una opción que permita agregar un nuevo amigo a tu lista.

Esta funcionalidad solamente agregará al usuario, sin pedir autorización y aceptación por parte del destinatario como hace Facebook.

Es decir, que la relación de amistad solamente existe en un sentido.

Actividad 7

Listas

7 - listas funciones.py

7 - listas main.py

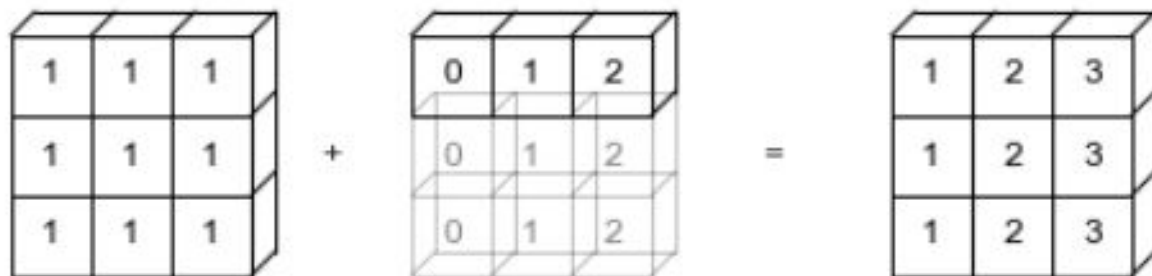
7. 2. Agrega una opción que permita mostrar los últimos estados de todos los amigos de un usuario.

Ten en cuenta que esto no es equivalente a publicar los mensajes de su muro, sino que necesitarás leer una línea particular de los archivos de cada usuario en su lista de amigos.

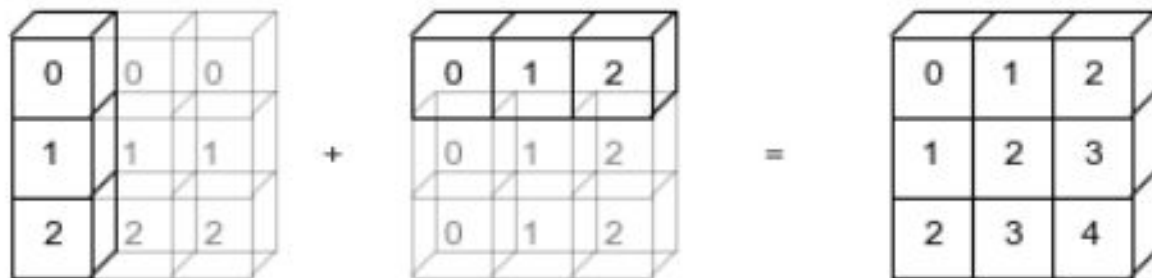
`np.arange(3)+5`



`np.ones((3,3))+np.arange(3)`



`np.arange(3).reshape((3,1))+np.arange(3)`



Broadcasting in Python

Numpy Arrays

```
1 # scalar and one-dimensional
2 from numpy import array
3 a = array([1, 2, 3])
4 print(a)
5 b = 2
6 print(b)
7 c = a + b
8 print(c)
```

Running the example first prints the defined one-dimensional array, then the scalar, followed by the result where the scalar is added to each value in the array.

```
1 [1 2 3]
2
3 2
4
5 [3 4 5]
```

Broadcasting in Python

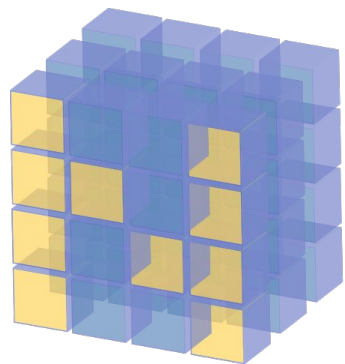
Numpy Arrays

```
1 # scalar and two-dimensional
2 from numpy import array
3 A = array([[1, 2, 3], [1, 2, 3]])
4 print(A)
5 b = 2
6 print(b)
7 C = A + b
8 print(C)
```

Running the example first prints the defined two-dimensional array, then the scalar, then the result of the addition with the value "2" added to each value in the array.

```
1 [[1 2 3]
2  [1 2 3]]
3
4 2
5
6 [[3 4 5]
7  [3 4 5]]
```


Librerías Python



NumPy

matplotlib

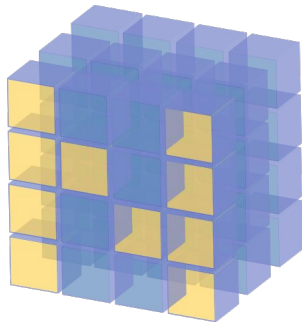


Pandas

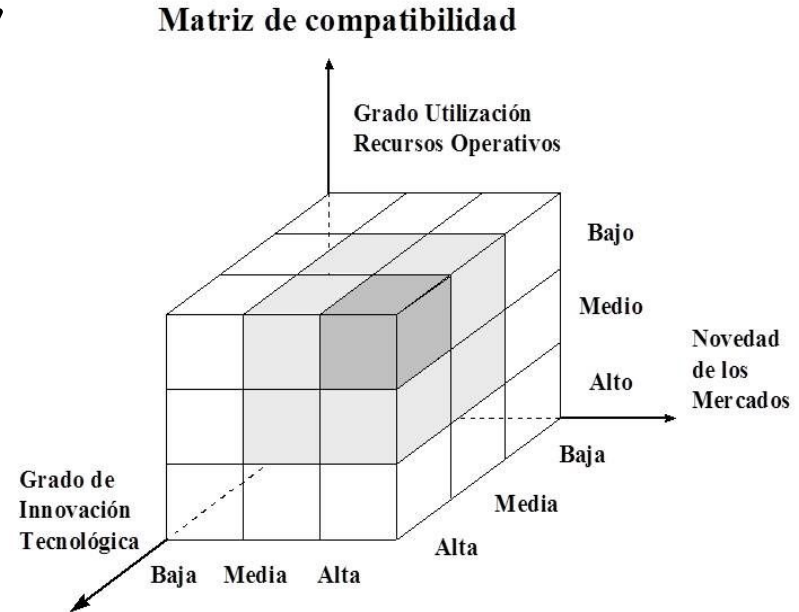


SQLAlchemy

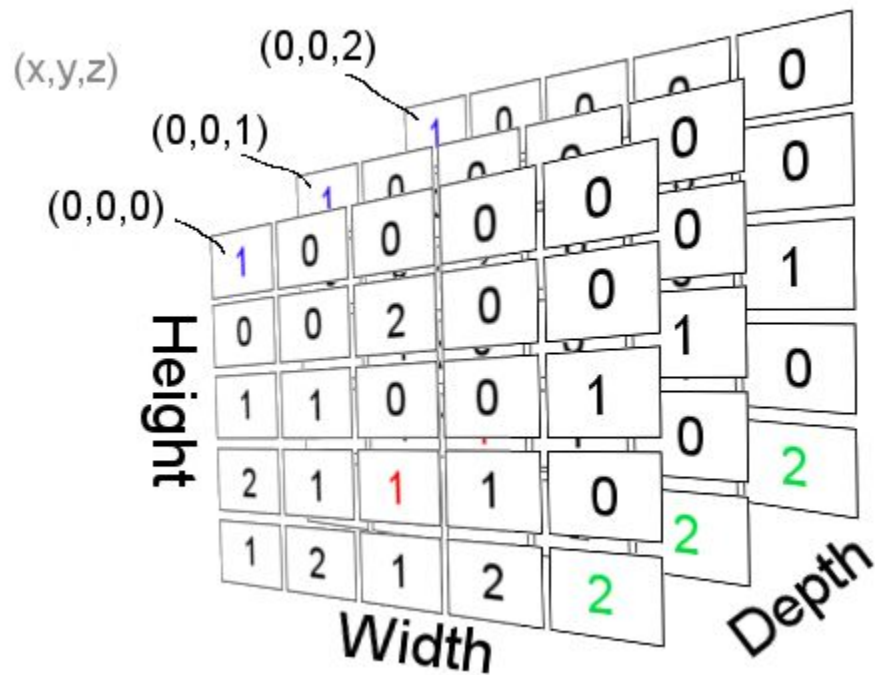
Librerías Python: Numpy



NumPy



Numpy: Arrays



Numpy: Arrays

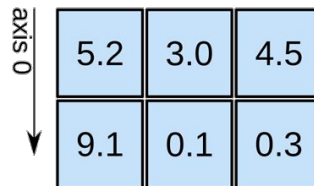
1D array



axis 0 →

shape: (4,)

2D array

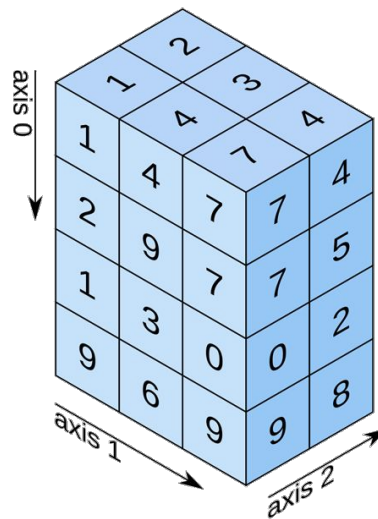


axis 0 ↓

axis 1 →

shape: (2, 3)

3D array



shape: (4, 3, 2)

Numpy: Arrays

addition

a $\square \square$

.70	.20	.10
.30	.60	.10
.50	.10	.40

a[1][2]

b $\square \square$

.80	.30	.50
.10	.40	.10
.10	.30	.40

b[1][2]

c $\square \square$

1.5	.50	.60
.40	1.0	.20
.60	.40	.80

c[1][2] = .1 + .1

multiplication

a $\square \square$

.70	.20	.10
.30	.60	.10
.50	.10	.40

← row 1

b $\square \square$

.80	.30	.50
.10	.40	.10
.10	.30	.40

column 2

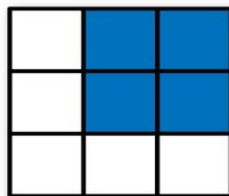
c $\square \square$

.59	.32	.41
.31	.36	.25
.45	.31	.42

c[1][2] = .3 * .5
+ .6 * .1
+ .1 * .4
= .25

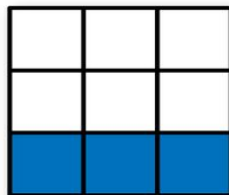
Typical matrix operations

Array Indexing



Expression
`arr[:2, 1:]`

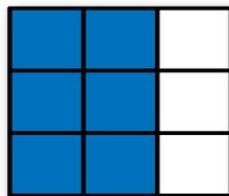
Shape
(2, 2)



`arr[2]` (3,)

`arr[2, :]` (3,)

`arr[2:, :]` (1, 3)



`arr[:, :2]` (3, 2)



`arr[1, :2]` (2,)

`arr[1:2, :2]` (1, 2)