**Python**

Python es un lenguaje de programacion interpretado

Interpretado: no se ejecuta directamente en el procesador.

Una herramienta llamada interprete ejecuta el codigo paso a paso.

L. Compilado es mas rapido q L. Interpretado.

Las variables en python son una referencia a una zona de memoria donde hay un valor exacto.

Cuando yo asigno un valor a una variable, le digo q hay una zona de memoria con un contenido exacto y si cambio el valor cambio la zona de la memoria.

Esto quiere decir que los valores numericos son inmutables.

Python es dinamico, capaz de disernir segun mi valor lo que hay en una zona.

3 tipos de datos inmutables, que no se pueden cambiar:

- Numericos

- Cadenas de Texto

- Tuplas

Tipos de datos mutables:

- Listas

- Diccionario (clave: valor)

- Set

Listas:

.append("Agregar valor")

.remove("Remover valor")

Diccionario:

.pop('claveName') //Nos devuelve el valor q tuviera la clase q estamos borrando.

Set:

No puede tener valores duplicados.

a | b --> todos los valores de las dos 'listas' sin duplicados.

a & b --> todos los valores que se repiten en las dos 'listas'.

a - b --> todos los valores de 'a' que no estan en b.

a ^ b --> todos los valores donde en 'a' y 'b' no se repiten.

Texto:

.capitalize() --> Pone la primera letra en may, y el resto en min.

.title() --> Pone cada inicial de cada palabra en may.

.lower() --> Pone todo en min.

.upper() --> Pone todo en may.

.replace('parte a sustituir', 'parte sustituyente') --> Sustituye todas las partes que contengan la 'parte a sustituir' por la 'parte sustituyente'.

.find('parte a encontrar') --> Si aparece, me dice en que posicion aparece esa parte.

.split() --> Convierte en una lista, cuando encuentre un espacio entre cada palabra, cada palabra se convertira en un valor de una lista.

.split('dondeRomperElString')

' '.join(listaTexto) --> Une una lista a un texto, dejando un espacio o lo que se ponga entre cada elemento de la lista.

Operadores:

Aritmeticos:

+

-

/

// --> Resultado

% --> Resto

\*

\*\*

Asignacion:

=

+=

-=

\*=

/=

\*\*=

&=

|=

^=

>>=

<<=

Comparacion/Condicionales:

==

!=

>

>=

<

<=

Identidad:

is

is not

Membresia:

in

not in

Bit:

&

|

^ "xor" uno tiene que ser falso si o si.

<<

>>

~

round(numero,2) --> coloca el numero con solo dos decimales

sorted(lista) --> Lista ordenada

sorted(lista, reverse=True) --Lista alrevez

-- Solo python 3.10.0 en adelante !!!!!!

match valor:

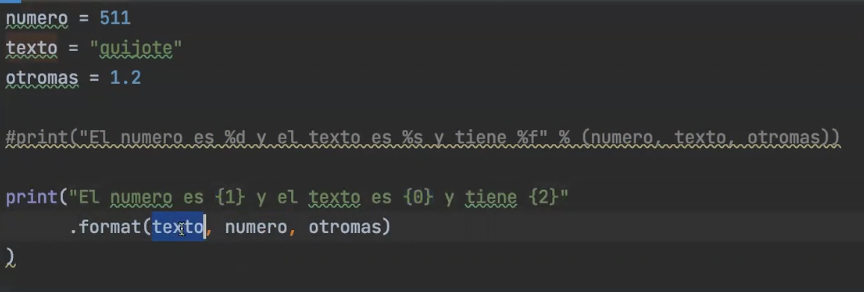
case 1:

print("valor 1")

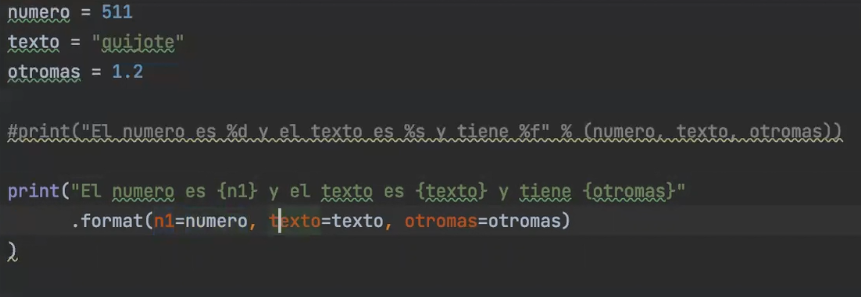
case 2:

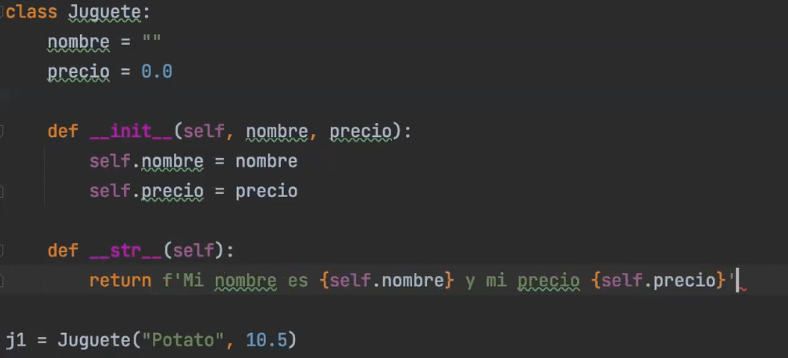
print("valor 2")

- 8.1



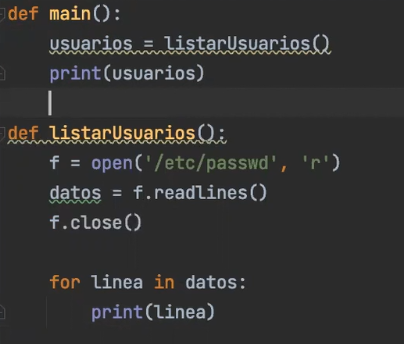
Los numeros entre {} son las posiciones del format al q iran.

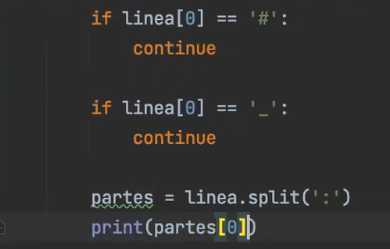




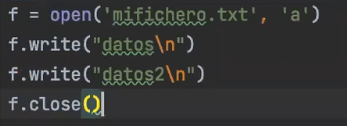
\_\_str\_\_ para salidas informales.

\_\_repr\_\_ para salidas formales / tecnicas.

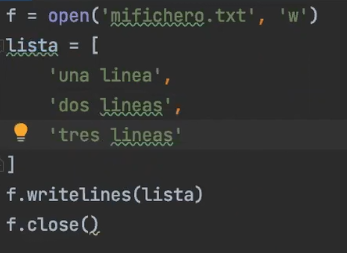


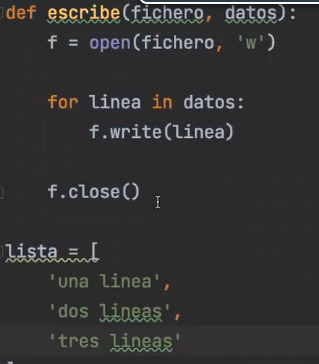


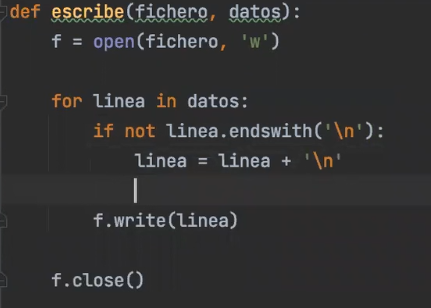
Escribir en un fichero. ‘a’ para escribir dentro del fichero despues de lo que ya estaba escrito.



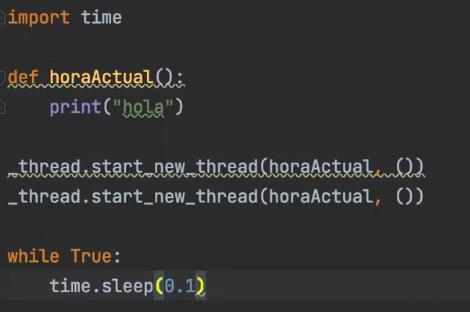
‘w’ para escribir dentro del fichero, pero borrando todo lo escrito anteriormente.







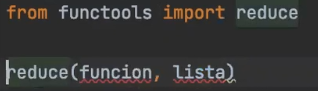
- 9.1

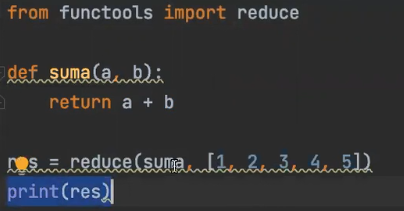


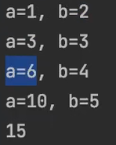


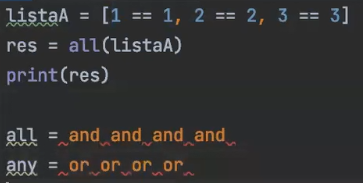






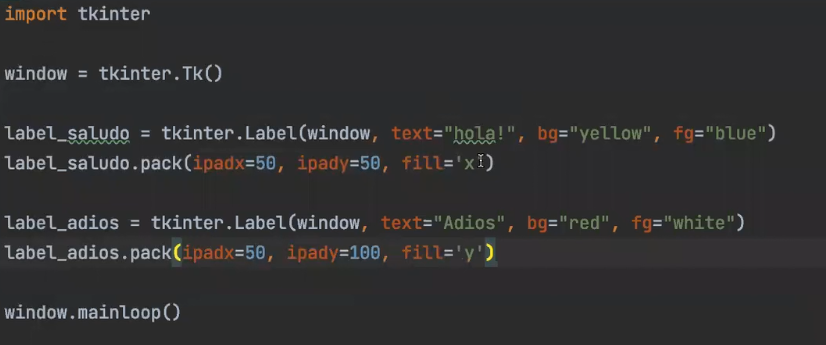


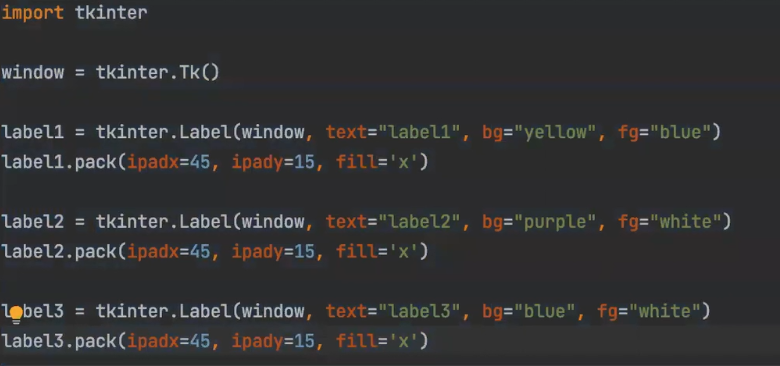


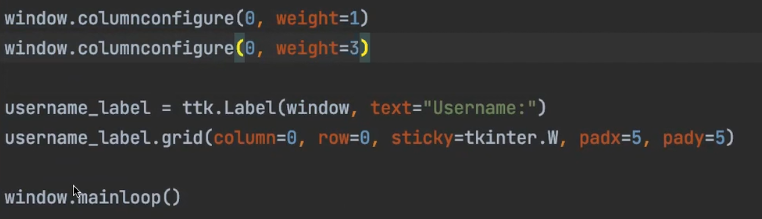


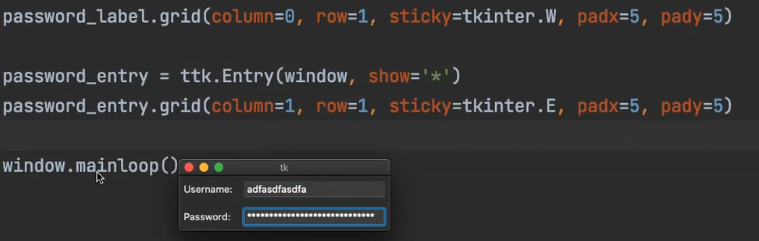
- 10.1

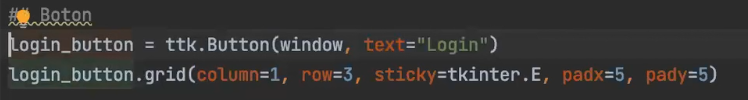
Tkinter





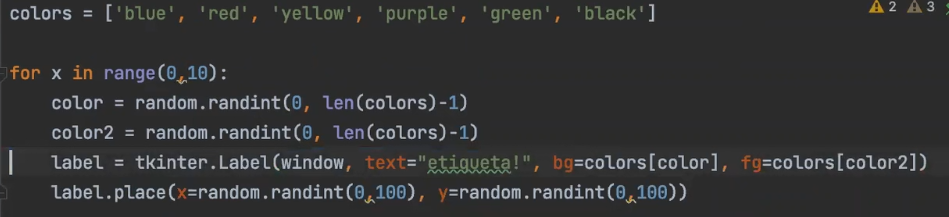


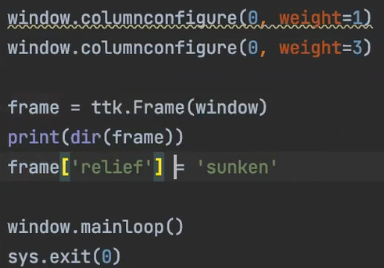


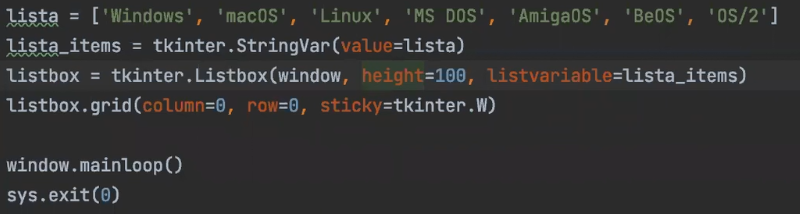


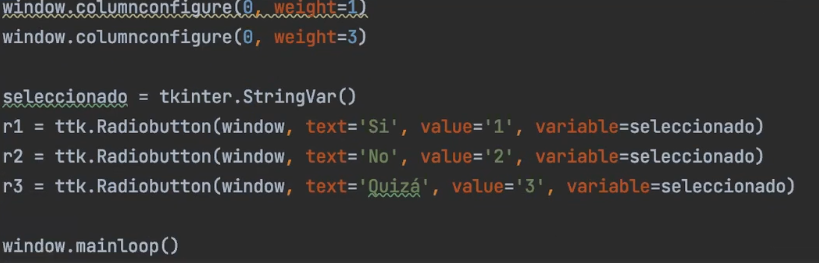


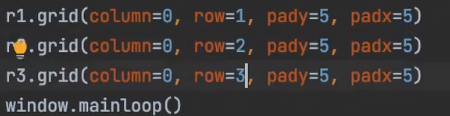


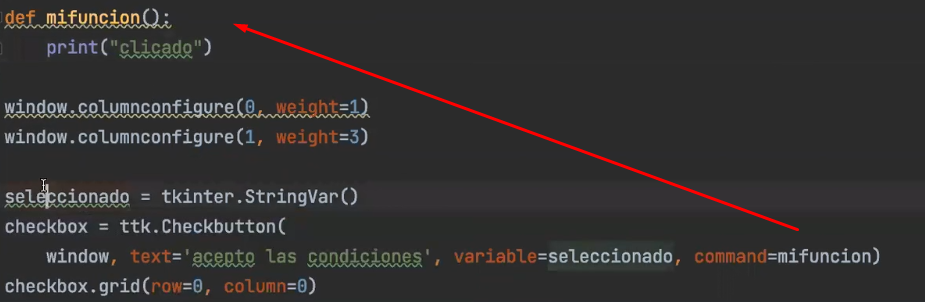


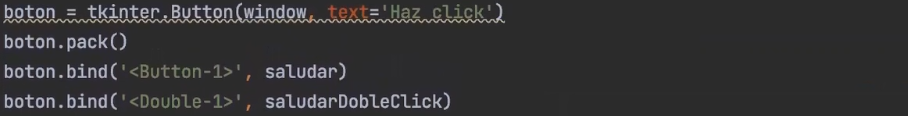


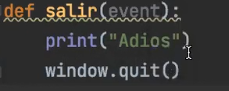




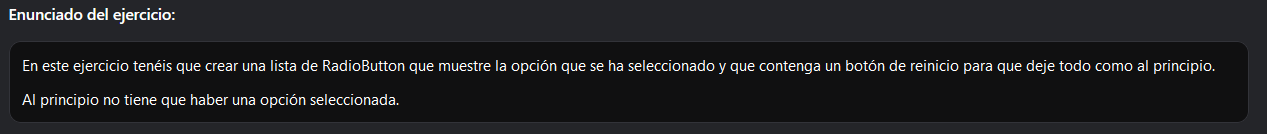


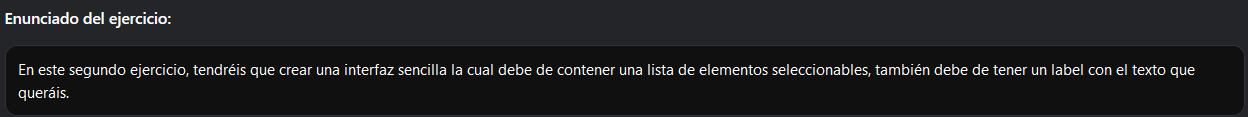






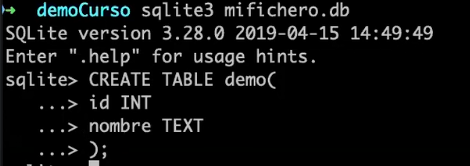
Ejercicios:

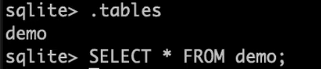




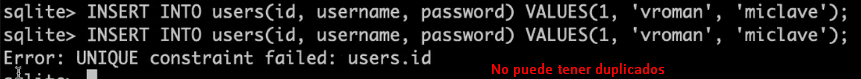
- 11.1



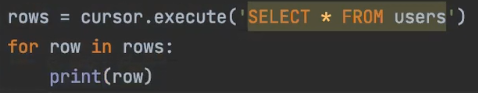






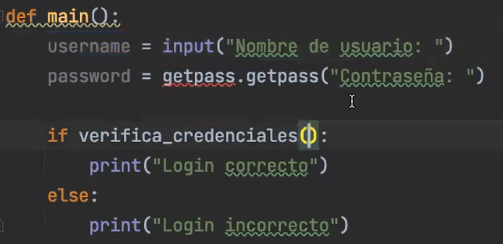


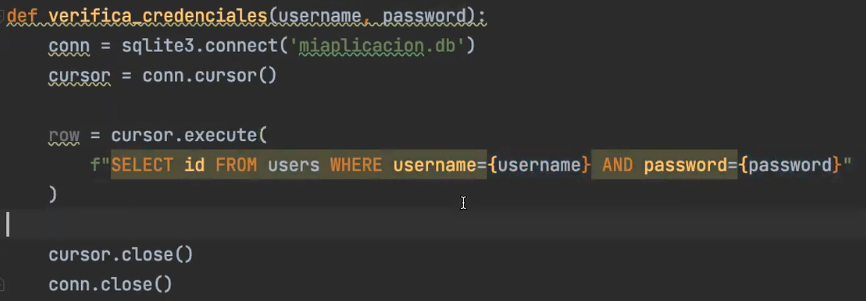


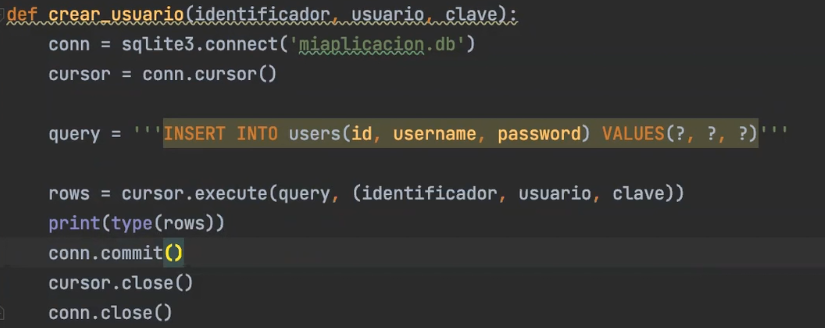


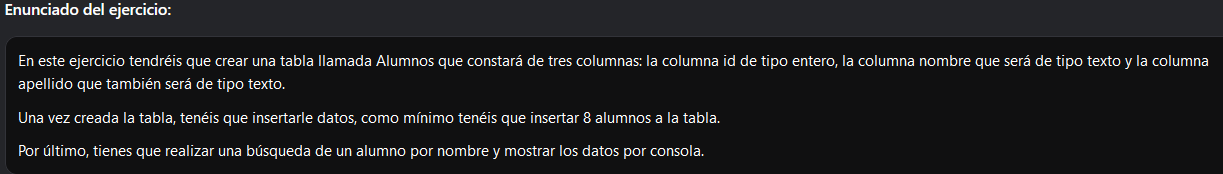


----









- 12.1

Django

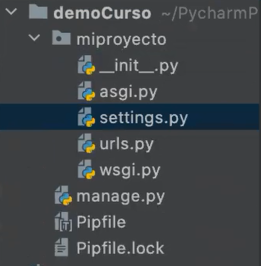
Framework que es un conjunto de herramientas que nos permite desarrollar sitios web de forma rapida.

Nos da cosas hechas ya.

[pip install django] o [pipenv install django]

-> Ir a carpeta -> [django-admin startproject miproyecto .]



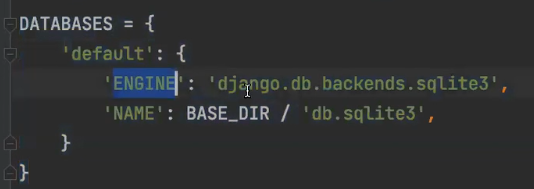


El proyecto es el todo de django, multiples aplicaciones.

Cada proyecto esta dividido en aplicaciones.



-> Para darle start / run al proyecto.

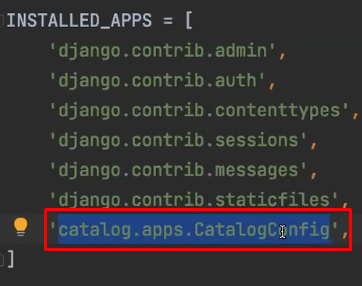


Deja donde guardar los datos configurado ya. En settings.py linea 80 aprox.

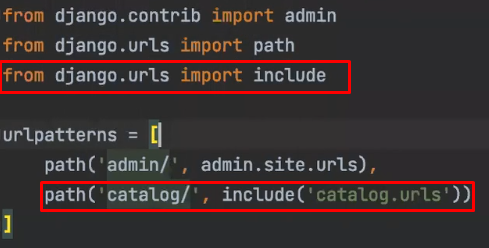
Aplicacion ->

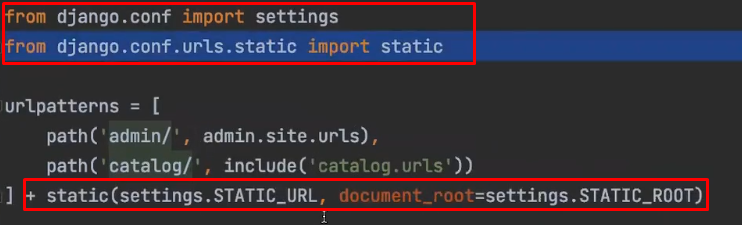


Conectar la aplicacion al proyecto -> Settings.py ->

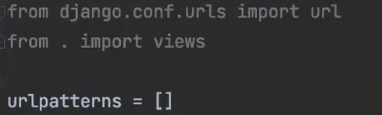


Urls.py ->

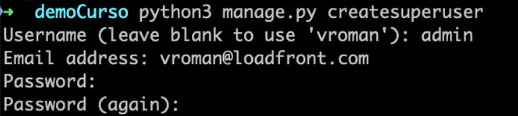




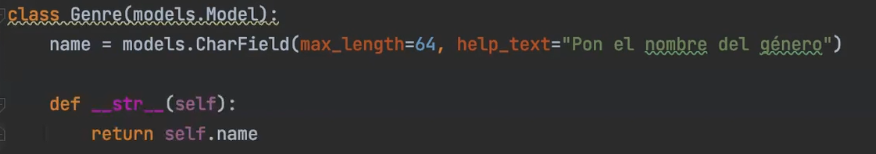
Carpeta catalog -> crear urls.py ->

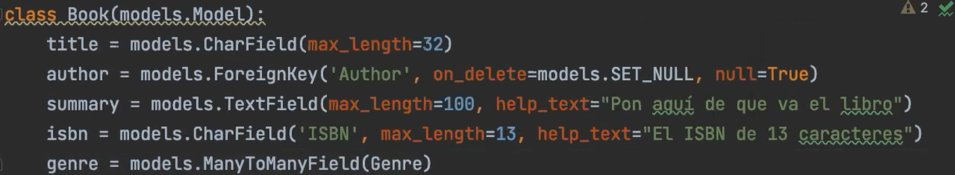


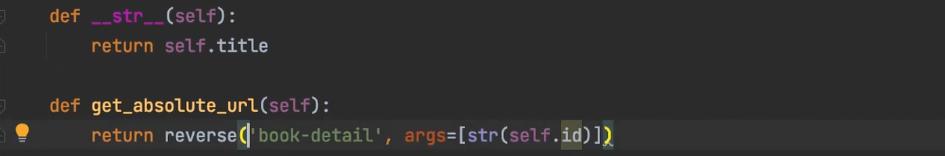
Admin ->

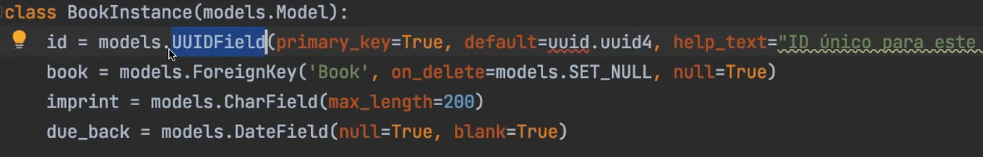


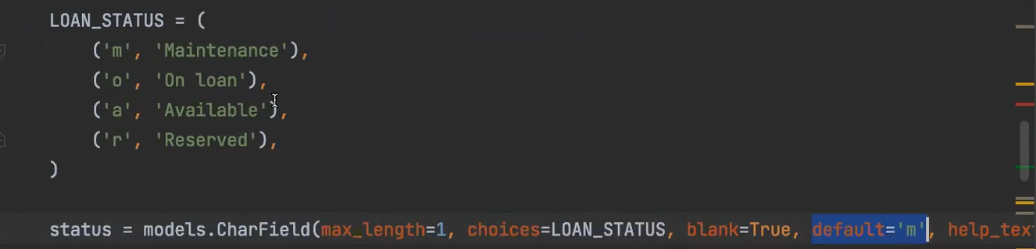
Catalog / Models.py ->







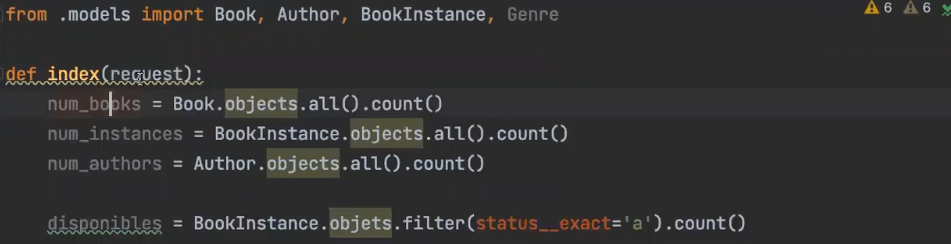




Catalog / admin.py ->

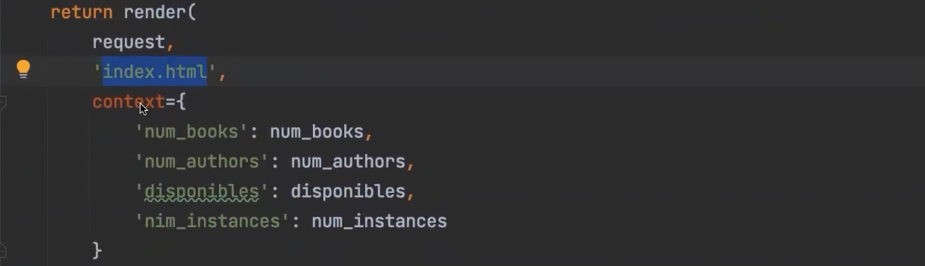


Catalog / Views.py ->

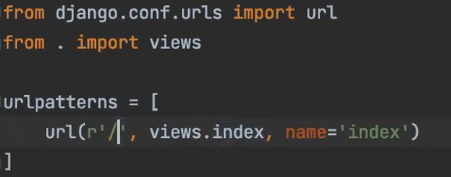


->

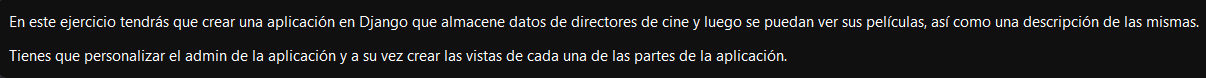




Catalog / urls.py ->



Ejercicios ->



Fin Python :)