

# PyTorch

## Bootcamp Udemy - Sección 7

**Programar con módulos: una forma más eficiente de organización y reutilización de código.**

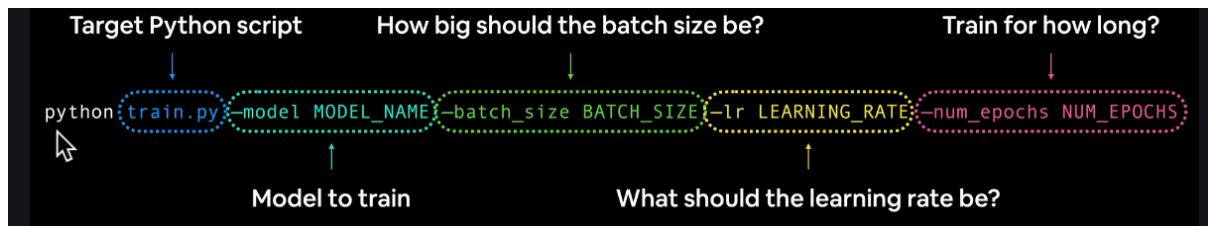
Dado que nuestro código mayormente fue escrito en laboratorios.ipynb y que otros pueden ser escritos en [scripts.py](#) es necesario saber que estos pueden ser utilizados en conjunto mediante línea de comandos, permitiendo de esta manera programar una única vez en caso de que sea necesario y ejecutar con un terminal u otro medio el mismo.

Una buena manera de valernos de la reutilización es en primer lugar conocer el directorio utilizado en nuestro sistema.

Para valernos de la linealidad de aprendizaje del curso usaré el mismo ejemplo del vídeo:



Sabiendo esto estaremos habilitados a entrenar nuestros modelos con una línea de código, ya sea en un [archivo.py](#), archivo.ipynb o una consola por comandos.

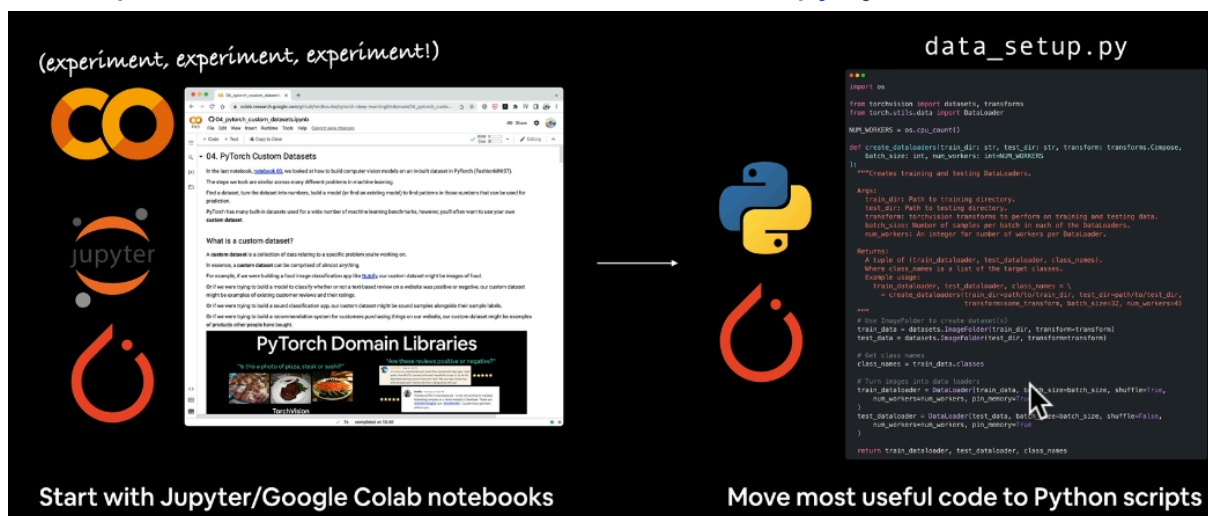


Por ejemplo:

```
python train.py -model tiniyvgg -batch_size 32 -lr 0.001 -num_epochs 10
```

Podremos agregar más hiperparametros, es un ejemplo.

Muchas veces una buena práctica es realizar pruebas en laboratorios.ipynb y luego, comprobada su eficiencia, llevar el que más se adapte a nuestras necesidades a un [archivo.py](#) fijo:



Los [archivos.py](#) serán importados como módulos genericos que cumplirán con nuestras necesidades, sin la necesidad de realizar nuevamente este código, algunos de estos usos pueden ser:

- Crear datasets y dataloaders: **data\_setup.py**
- Crear modelo: **model\_builder.py**
- Función de entrenamiento con entrenamiento y prueba: **train.py**
- Guardar el modelo: **utils.py**
- [train.py](#) y [utils.py](#) en un solo modulo.

- Funciones de ayuda: **heper\_functions.py**

Dado que el código del curso fue escrito completamente diferente omitiré importar los modelos y reescribir el código nuevamente para evitar inconsistencias, sin embargo estará disponible en el [github oficial del curso](#).