



## ML Days Study Jams #1 – Showcasing Skills(Git, Pip, Jupyter Lab)

### VIRTUAL ENVIRONMENTS



İki tip Virtual Environments vardır.

- Python Pip
- Conda

#### PIP BAŞLANGIÇ

- ⇒ **pip kontrol**  
| pip --version
- ⇒ **pip kullanımı için virtualenv kurulumu**  
| pip install virtualenv
- ⇒ **pip virtual environment oluştur.**  
| python -m venv my\_env\_name
- ⇒ **oluşturulan env'yi aktive etme**  
| source my\_env\_name/bin/activate
- ⇒ **aktive edilmiş env'yi deaktive etme**  
| deactivate

#### PIP PAKET YONETİMİ

- ⇒ **pip paket yükleme**  
| pip install paket\_ismi
- ⇒ **pip gerekli paketlerin birlikte yüklenmesi**  
| pip install -r requirements.txt
- ⇒ **pip env'ye yüklenmiş paketlerin listesi**  
| pip freeze
- ⇒ **pip env'ye yüklenmiş paketlerin 'requirements.txt' olarak listelenip dosya haline getirilmesi.**  
| pip freeze > requirements.txt
- ⇒ **pip paketin env'den kaldırılması**  
| pip uninstall paket\_ismi

## ANACONDA & CONDA

- ⇒ **conda versiyon kontrolü**  
| conda -version
- ⇒ **conda virtual enviroment oluştur.**  
| conda create -n env\_ismi
- ⇒ **spesifik bir python versiyonu ile çalışmak için**  
| conda create -n env\_ismi python==3(daha spesifik : python == 3.3)
- ⇒ **oluşturulan env'yi activate/deactivate etme**  
| conda activate env\_ismi (veya) source activate env\_ismi  
| conda deactivate
- ⇒ **conda paket listeleme**  
| conda list
- ⇒ **conda paket yükleme & update**  
| conda install paket\_ismi & conda update paket\_ismi

## JUPYTER LAB



- ⇒ **pip ile jupyter lab yükleme**  
| pip install jupyterlab
- ⇒ **conda ile jupyter lab yükleme**  
| conda install -c conda-forge jupyterlab
- ⇒ **conda env'yi jupyter lab'e kernel olarak ekleme**  
| conda activate env\_ismi  
| conda install ipykernel  
| ipython kernel install --user --name == kernel\_ismi
- ⇒ **terminalde jupyter lab başlatma**  
| jupyter lab



## GİT \_VERSION CONTROL

Kodumun eski versiyonuna dönebilirim.

## GİT CONFIGURATION

|git config --global user.name "saaydmr"

|git config --global user.email "[saaydmr@gmail.com](mailto:saaydmr@gmail.com)"

⇒ **Git yapılandırma ayarlarını listele**

|git config --list

⇒ **Git local repository başlat**

|git init

⇒ **Projedeki değişim durumunu göster**

|git status

## GİT LOCAL OPERATIONS

Working Directory(Kendi bilgisayarımız)

+git add+

Staging Area(Hazırlık alanı)

+git commit+

Local Repository

+git checkout+(kullanarak eski haline dönebiliyoruz.)

|git add <file\_name> or <.>

|git commit -m <commit\_message>

## GİT REMOTE OPERATIONS

⇒ **GitHub repository için ilk remote url ekleme**

|git remote add origin <url(https)>

⇒ **Local repository'den master branch'e aktarma**

|git push -u origin master

## **GİTIGNORE**

Oluşturduğumuz GitHub depomuza ya da local depomuza projemizdeki Göndermek istemediğimiz dosya ve klasörleri .gitignore uzantılı bir dosya oluşturarak içine Ekleyebiliriz.

## **BRANCHING, MERGING**

- Branch İngilizcede dal demektir.  
ana projeye yeni bir şey eklemek istiyorsunuz.Ayrıca bir branch oluşturarak masterı etkilemeden ayrı bir yol açarak çalışmaya devam ederiz.
- Merging İngilizcede birleştirmek demektir.  
Branch'i productiona eklemek istersek merging process'i kullanırız.

### ⇒ **Yeni branch ekle**

|git branch <branch\_name>

### ⇒ **Eklenen branchleri göster**

|git branch

### ⇒ **Başka bir branch'e geçiş yap**

|git checkout <branch\_name>

### ⇒ **Farklı bir branch'i birleştir**

|git merge <branch\_name> (on master branch)