

1. Proje Yapısı Oluşturun

Klasör yapısı şu şekilde olmalı;

```
projeadı/  
├── projekodları/  
│   ├── projeadı.py  
│   └── __init__.py  
├── tests  
│   └── test_projeadı.py  
├── setup.py  
├── requirements.txt  
├── README.md  
├── .github/  
│   └── workflows/  
│       └── python-test.yml
```

- projeadı.py : Dosya adını kendinize göre düzenleyin. Ödevde yazdığınız(shortestpath) kodunuzu içermeli.
- test_projeadı.py : Kod kontrolü için gerekli kısıtları içeren pytest test dosyası.
- requirements.txt : Kod betiğinin çalışabilmesi için gerekli paketler. Alt alta sadece paket isimlerini yazabilirsiniz. (numpy, pytest vs)
- README.md : Proje açıklamasını yazabilir, dokümantasyona erişmek için sphinx dokümantasyonunun github linkini bu readme içinde paylaşabilirsiniz.
(Sphinx dökümantasyonu proje kodlarının ayrıntılı kullanım kılavuzudur. Readme veya testpypi'daki Project description ise kodun ne işe yaradığının, ne amaçla kullanılabileceğinin genel açıklamasıdır.)
- .github/workflows/python-test.yml : Github actions'ı aktifleştirmek için gerekli.
- __init__.py , setup.py : testpypi paket yüklemesi için dosyalar. İlgili adımda bahsedilecek.

2. Sphinx Dökümantasyonu Oluşturun

Terminalde;

pip install sphinx sphinx_rtd_theme (başka bir tema da kullanabilirsiniz)

Terminalde projenin olduğu klasöre cd komutu ile gelin;

sphinx-quickstart docs

> Separate source and build directories (y/n) [n]: y

> Project name: *Projeadı*

> Author name(s): *Adınız*

> Project release: *0.1*

Bu işlemten sonra docs klasörü altında source/, build/, Makefile, make.bat gibi dosyalar oluşur.

Ardından terminalde projeklasörünün içindeki docs klasörüne gidin;

```
cd docs
```

```
sphinx-apidoc -o source ../          (Bu, source/ klasöründe modules.rst ve calculator.rst gibi dosyalar oluşturur.)
```

windows'ta;

```
make.bat html veya .\make.bat html komutlarıyla html oluşturun.
```

Çıktılar docs\build\html\ klasöründe oluşur. index.html dosyasını çift tıklayıp tarayıcıda açabilirsiniz.

Güncellemeler yaptınız ve dökümanı güncellemek istediğinizde(terminalde proje klasöründe);

```
sphinx-apidoc -f -o docs/source ../
```

```
cd docs
```

```
.\make.bat html
```

3. Projenizi Github Reposuna Yükleyin

Projeadi klasöründe (lütfen projeadı yazan kısımları kendinize göre düzenleyin) git projesi başlatın (git init) ve reponuza yükleyin. (<https://github.com/mufide/GMT211-Tutorial>)

Main branch'e yüklemede sıkıntı yaşarsanız şu adımları takip edin;

```
git branch -M main
```

```
git remote add origin https://github.com/adınız/repoadı.git
```

```
git push -u origin main
```

4. Test PyPi için Gerekli Dosyalar

(TestPyPI, PyPI sisteminin birebir kopyasıdır. PyPI veritabanına aynı paketleri defalarca yükleyip gereksiz ağırlık oluşturmamak için ödev veya deneme paketlerini buraya yükleyeceğiz. Ancak başka paket denemelerini tamamladığınızda PyPI kullanabilirsiniz, herhangi bir yetki kısıtlaması yok.)

Terminalde gerekli kütüphaneleri indirmek için;

```
pip install setuptools wheel twine
```

setup.py içeriği;

```
from setuptools import setup, find_packages
```

```
with open("README.md", "r") as f:
```

```
    description = f.read()
```

```
setup(
```

```
    name='shortestpath000000000, (00000 olan bölüme öğrenci numaranızı yazmalısınız)
```

```
    version='0.1.0',
```

```
    author='adınız',
```

```

author_email='mail@mail.com',          (e-mail adresiniz)
description='birkaç kelimelik açıklama',
long_description=open('README.md').read(),
long_description_content_type='text/markdown'
url='https://github.com/kullaniciadi/....', (projeyi yüklediğiniz repo linki)
packages=find_packages(),
install_requires=[                      (projenizde hangi paketler gerekliyse onları yazmalısınız)
    'numpy>=1.21.0',
    'pandas>=1.3.0',
],
classifiers=[
    'Programming Language :: Python :: 3',
    'Operating System :: OS Independent',
],
python_requires='>=3.6',
)

```

Kurulum dosyalarını oluşturmak için terminalde proje klasöründe;

```
python setup.py sdist bdist_wheel
```

veya

```

pip install --upgrade build
python -m build

```

Böylece proje klasöründe dist/ klasörü ve altında 2 kurulum dosyası oluşacak.

5. Test PyPI Paket Yükleme

<https://test.pypi.org/account/register/> adresinden hesap açın.
<https://test.pypi.org/manage/account/#api-tokens> adresinden (entire account) API Token oluşturun.

Terminalde proje klasöründe;

```
Python -m twine upload --repository testpypi dist/*
```

Bu satırı çalıştırdığınızda API token isteyecek, oluşturduğunuz API Token'ı buraya yapıştırıp enter'a basın.

Paketin dosyaları testpypi'a yüklendi ve bir link verdi. Bu link artık paketinizin sisteme yüklendiğini ve indirilebilir olduğunu gösterir. Bu linke tıklayarak paketinize ulaşabilirsiniz.

Paketleme işlemini tamamladınız. Paketinizi yüklemek için terminalde;

```
pip install -i https://test.pypi.org/xxx/proje adi
```

(PyPI'a yüklemiş olsaydık `pip install shortestpath00000` yazmamız yeterli olacaktı.)