



Django İle Python Arka Yüz Yazılım Geliştirme

Teknik Bilimler
Meslek Yüksekokulu

Öğr. Gör. Furkan DURMUŞ



Bu Haftanın Ders Kazanımları



Sanal Ortam

startproject

venv Oluşturma

runserver

Django Kurulumu



```
require 'capybara/reails'
```

```
1 Capybara.javascript_driver = :webkit
```

```
2 Category.delete_all; Category.create
```

```
13 Shoulda::Matchers.configure do |config|
```

```
14 config.integrate do |with|
```

```
15 with.test_framework :rspec
```

```
16 with.library :rails
```

```
17 end
```

```
18 end
```

```
19 end
```

```
20 # Add additional requires below this line
```

```
21 # Requires supporting ruby files with support
```

```
22 # spec/support/ and its subdirectories. Files
```

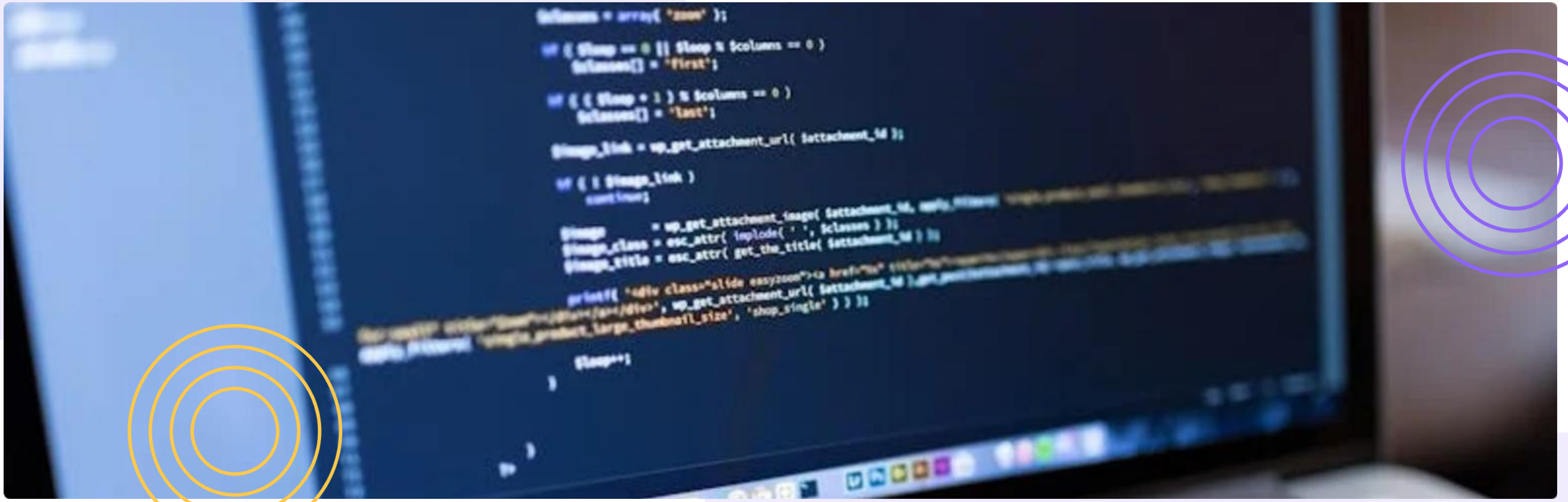
```
23 # run as spec files by default. This will
```

```
24 # in _spec.rb will both be required and
```

```
25 It is recommended that you configure
```

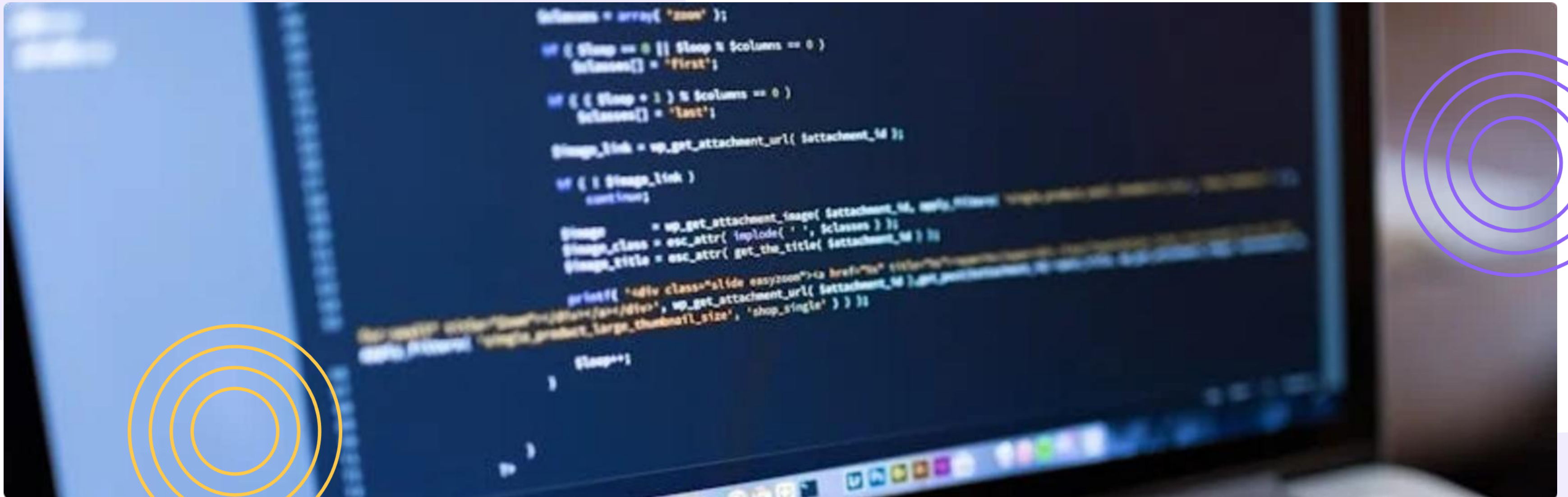
SANAL ORTAM

Sanal Ortam nedir?



Sanal Ortam nedir?

Deep Freeze ve sanal ortam benzerliği



Sanal Ortam

Bilgisayarınızda bir Python projesi yaptınız, ardından başka bir proje yaptınız.

İkisinde de farklı paket sürümleri gerekiyorsa ne olur?



Sanal Ortam

Python Global Ortamı

Bilgisayarınızda Python kurduğunuzda:

- Tek bir Python vardır.
- Tüm projeler aynı Python'u kullanır.
- Tüm pip paketleri aynı yere yüklenir.





Sanal Ortam

```
pip install django
```



django



Sanal Ortam

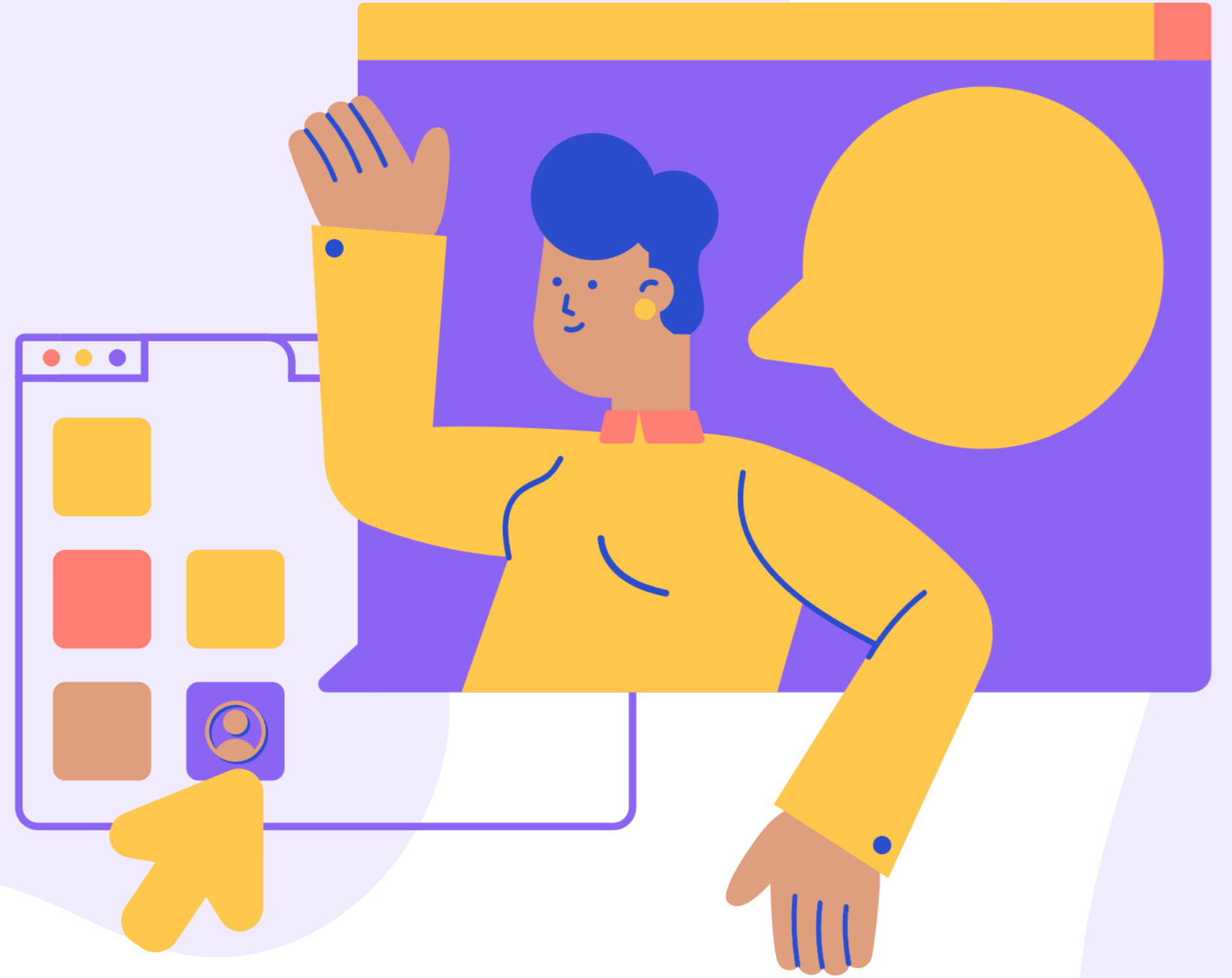
```
pip install django
```

Django'yu tüm sistem için kurar.
Her proje aynı Django sürümünü kullanır.



django

Çözüm : Sanal Ortam





Normalde:

```
Bilgisayar  
├── Python  
│   ├── Django (global)
```



Normalde:

```
Bilgisayar
├── Python
│   └── Django (global)
```

Sanal ortamlarla:

```
Bilgisayar
├── Proje1
│   ├── venv
│   │   └── Django 3.2
│   └── Proje2
│       ├── venv
│       │   └── Django 4.2
```



Sanal ortam (venv)

Nedir?

venv (virtual environment) = Python projeleri için oluşturulan izole (bağımsız) çalışma ortamı.

- Python'ın kendi içinde gelen bir modüldür.
- Ekstra bir program değildir.
- Python 3 ile birlikte gelir.
- Ortam izolasyonu sağlar.





**venv ile her proje
kendi Python
paketlerini ayrı
kullanır.**



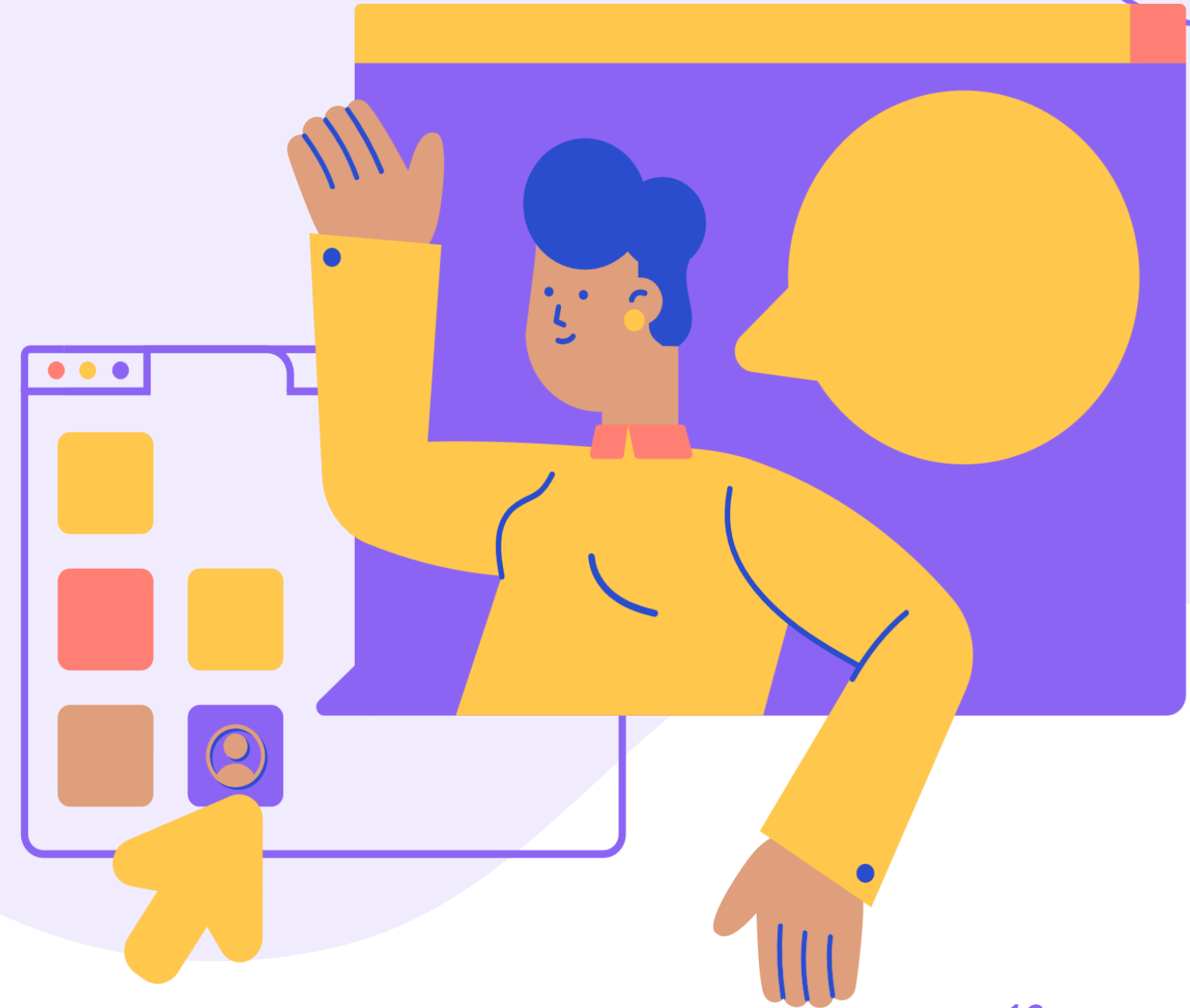
Sanal Ortam Olmazsa Ne Olur?



Sanal Ortam Olmazsa

Ne Olur?

- ✗ Paket sürümleri çakışır
- ✗ Proje taşınamaz
- ✗ Başka bilgisayarda çalışmaz
- ✗ Hata ayıklamak zorlaşır



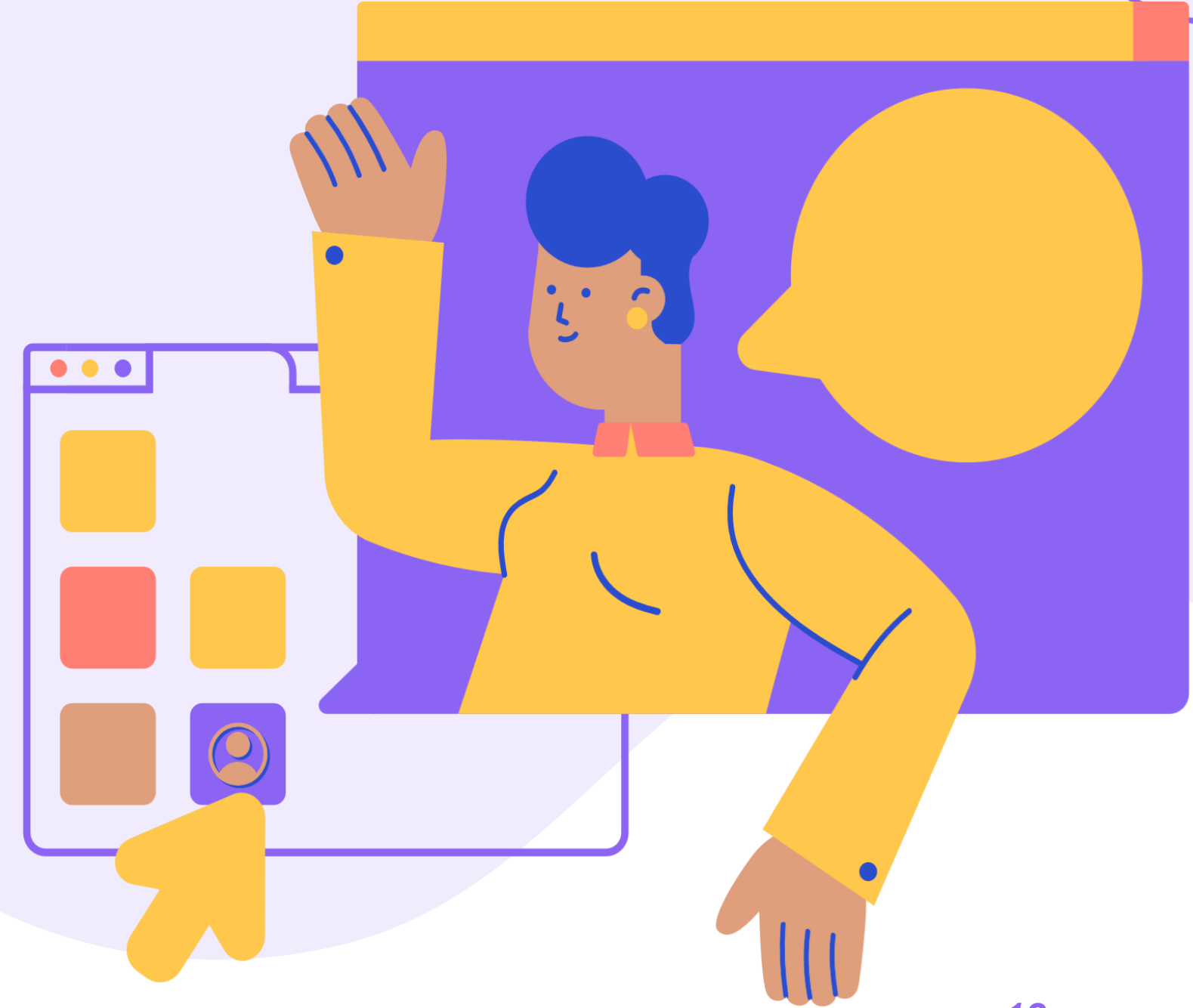


Sanal Ortamın Avantajları



Sanal Ortamın Avantajları

- ✓ Sürüm kontrolü
- ✓ Proje taşınabilirliği
- ✓ Güvenli geliştirme ortamı
- ✓ Sistem Python'unu bozmaz
- ✓ Ekip çalışmasına uygundur



Sanal Ortam Özetle:



Projeyi izole eder.



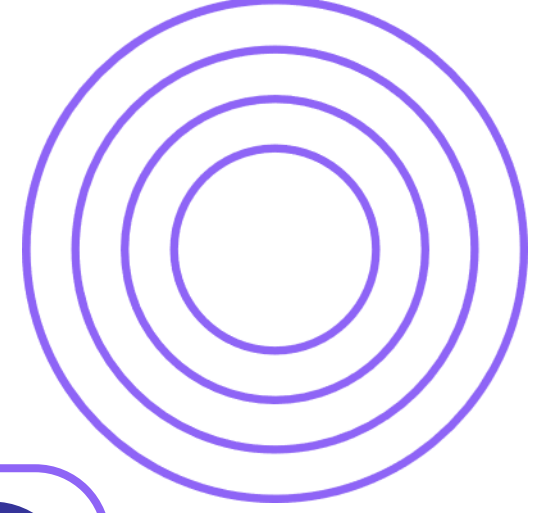
Paket çakışmasını engeller.



Profesyonel yazılım standardıdır.



Her Django projesi için gereklidir.





venv oluşturma

```
require 'capybara/reails'
```

```
1 Capybara.javascript_driver = :webkit
```

```
2 Category.delete_all; Category.create
```

```
13 Shoulda::Matchers.configure do |config|
```

```
14 config.integrate do |with|
```

```
15 with.test_framework :rspec
```

```
16 with.library :rails
```

```
17 end
```

```
18 end
```

```
19 # Add additional requires below this
```

```
20 # Requires supporting ruby files with support/ and its subdirectories. These files
```

```
21 # spec/support/ and its subdirectories. These files will be loaded by default.
```

```
22 # run as spec files by default. This will be required unless you specify
```

```
23 # in _spec.rb will both be required. It is recommended that you specify
```

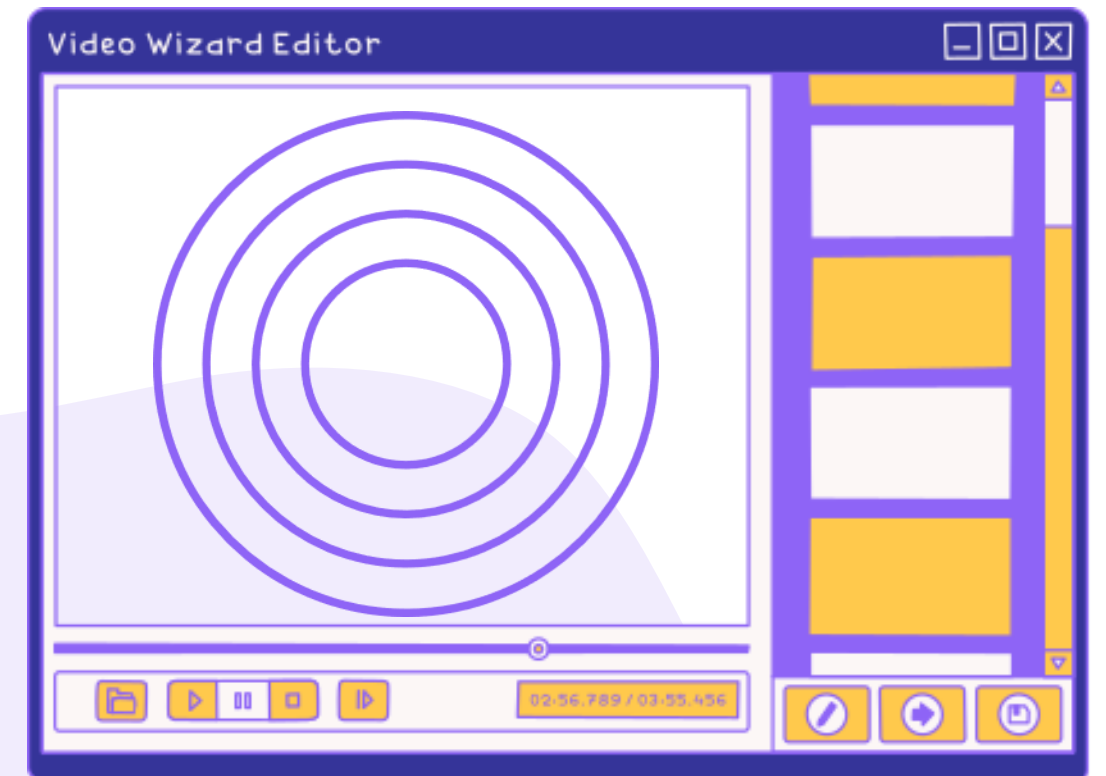
```
24 # in _spec.rb will both be required. It is recommended that you specify
```

```
25 # in _spec.rb will both be required. It is recommended that you specify
```

venv Oluşturma

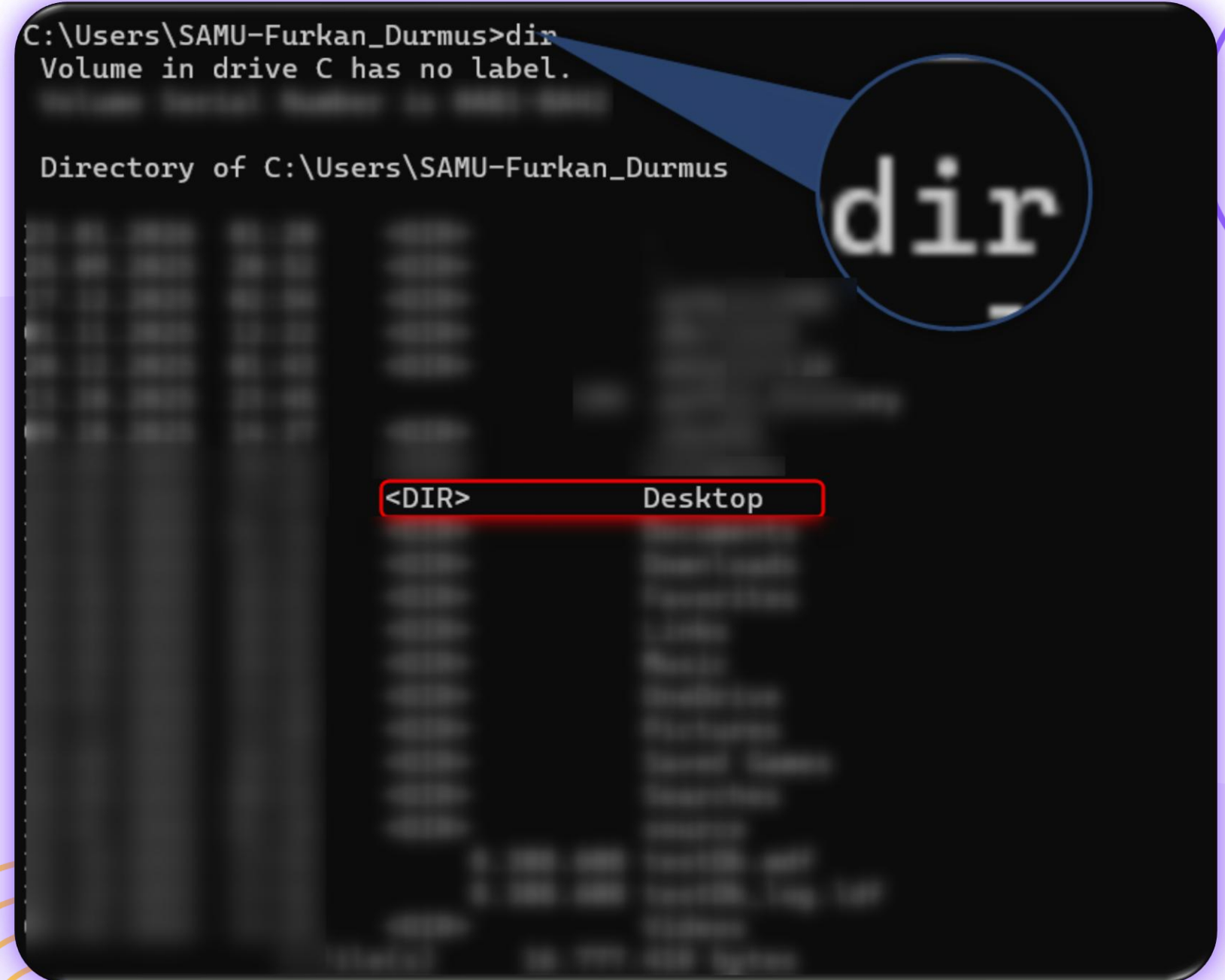
- Windows + R
- cmd

```
python --version  
python3 --version
```



venv Oluşturma

- Windows + R
- cmd
- dir



```
C:\Users\SAMU-Furkan_Durmus>dir
Volume in drive C has no label.

Directory of C:\Users\SAMU-Furkan_Durmus

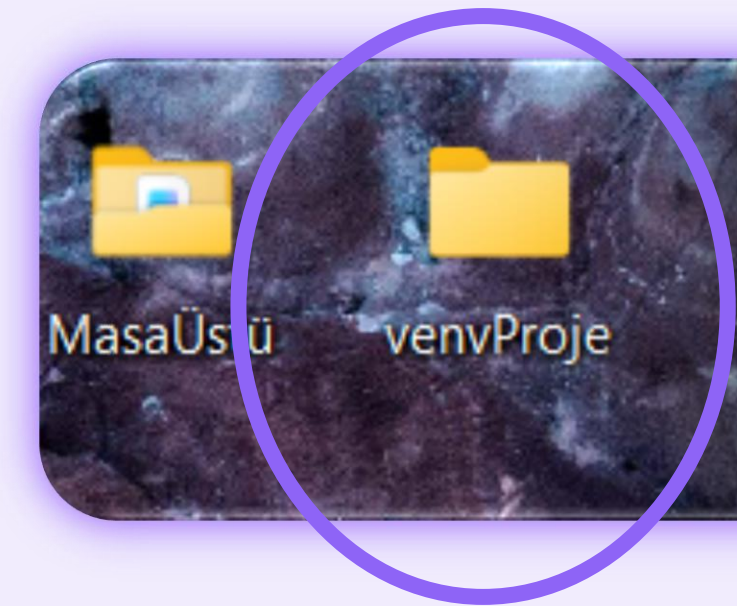
<DIR> Desktop
```

venv Oluşturma

```
C:\Users\SAMU-Furkan_Durmus>cd desktop  
C:\Users\SAMU-Furkan_Durmus\Desktop>
```

cd desktop

venv Oluşturma



```
C:\Users\SAMU-Furkan_Durmus>cd desktop
```

```
C:\Users\SAMU-Furkan_Durmus\Desktop>mkdir venvProje
```

```
C:\Users\SAMU-Furkan_Durmus\Desktop>
```

mkdir venvProje

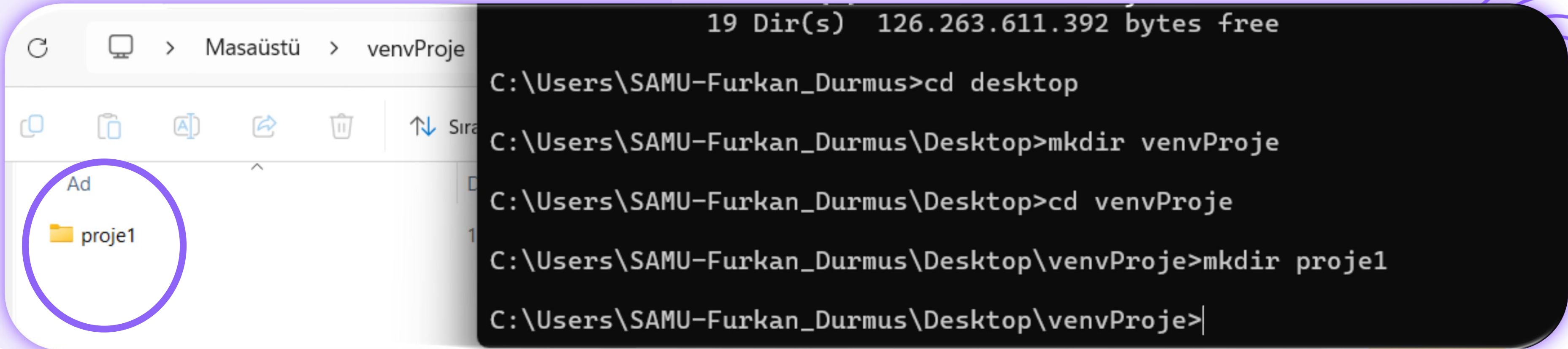
venv Oluşturma

```
C:\Users\SAMU-Furkan_Durmus\Desktop>cd venvProje
```

```
C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje>
```

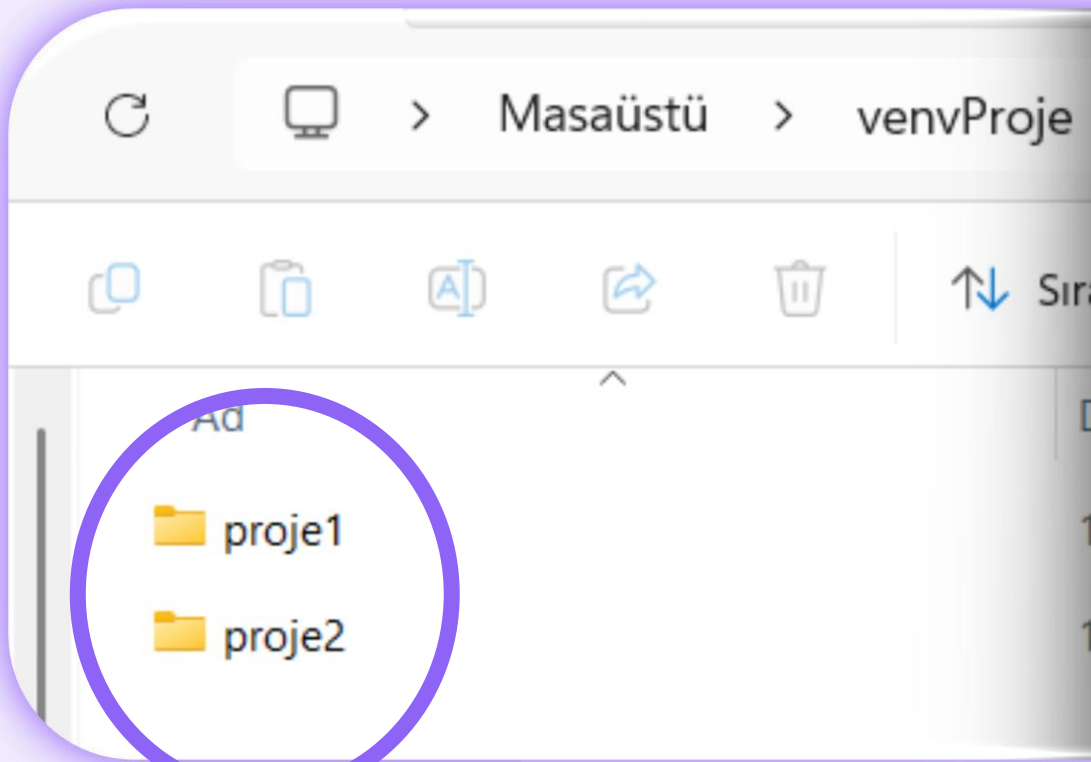
cd venvProje

venv Oluşturma



mkdir proje1

venv Oluşturma



```
C:\Users\SAMU-Furkan_Durmus>cd desktop
C:\Users\SAMU-Furkan_Durmus\Desktop>mkdir venvProje
C:\Users\SAMU-Furkan_Durmus\Desktop>cd venvProje
C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje>mkdir proje1
C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje>mkdir proje2
C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje>
```

mkdir proje2

venv Oluşturma

```
C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje>cd proje1  
C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje\proje1>
```

cd proje1

venv Oluşturma

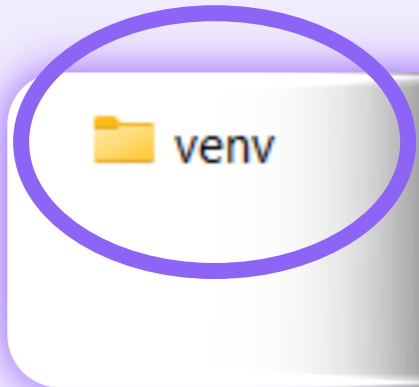
venv

```
C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje>cd proje1
```

```
C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje\proje1>python -m venv venv
```

python -m venv venv

venv Oluşturma



```
C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje>cd proje1
```

```
C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje\proje1>python -m venv venv
```

python -m venv venv

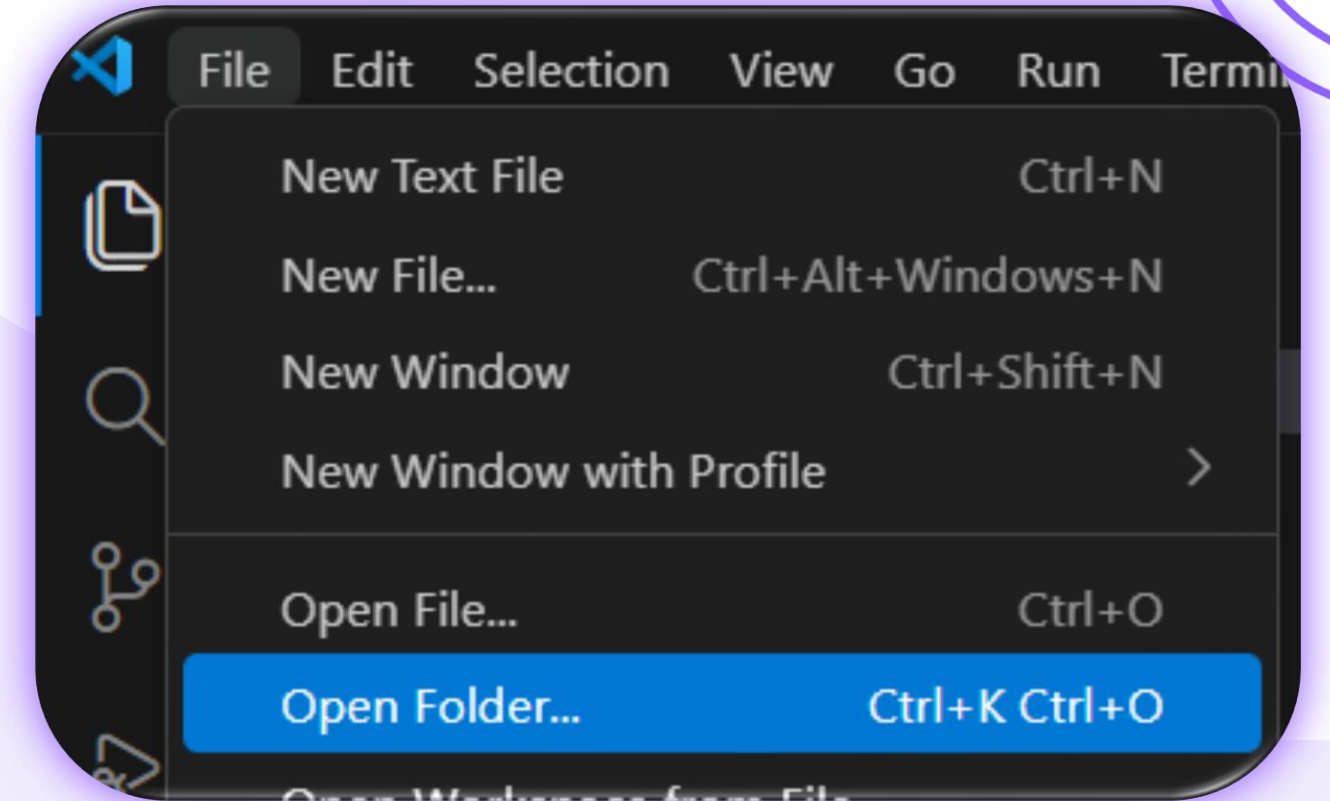
python -m → Python modül çalıştırır

venv → virtual environment modülü

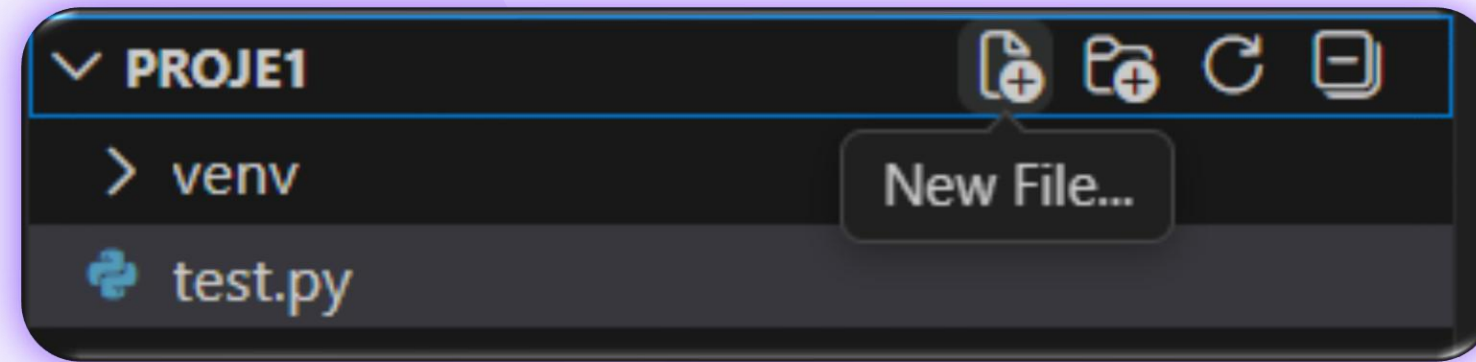
venv → oluşturulacak klasör adı

venv Oluşturma

Vs code > open folder diyerek proje1'i
açalım.

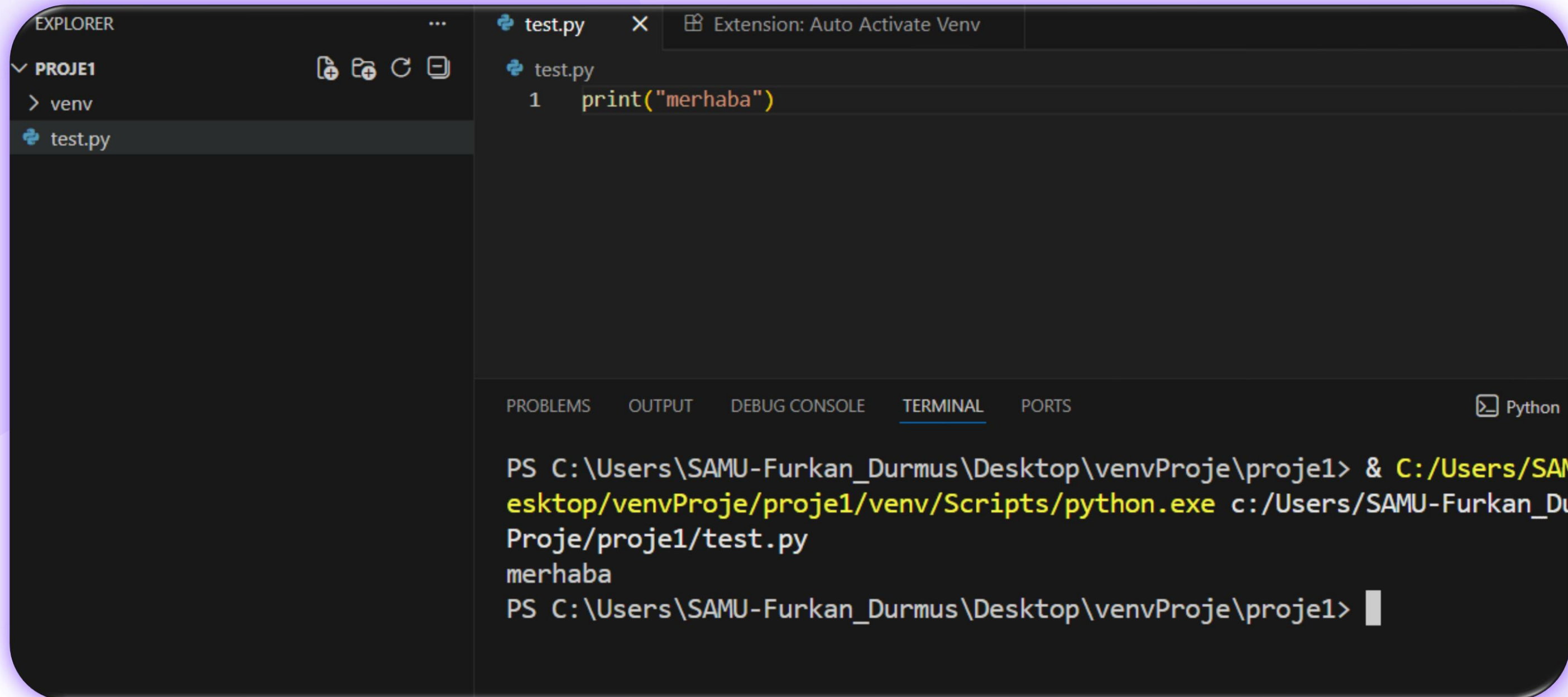


venv Oluşturma



Proje1'in altına test.py adında bir dosya oluşturalım.

venv Oluşturma



```
EXPLORER
PROJE1
  venv
  test.py

test.py
1 print("merhaba")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python
PS C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje\proje1> & C:/Users/SAMU-Furkan_Durmus/Desktop/venvProje/proje1/venv/Scripts/python.exe c:/Users/SAMU-Furkan_Durmus/Desktop/venvProje/proje1/test.py
merhaba
PS C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje\proje1>
```

venv Başlatma

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - [ ] [ ] ... | [ ] [ ] X
PS C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje\proje1> & C:/Users/SAMU-Furkan_Durmus/Desktop/venvProje/proje1/venv/Scripts/python.exe c:/Users/SAMU-Furkan_Durmus/Desktop/venvProje/proje1/test.py
merhaba
PS C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje\proje1> venv\Scripts\activate
(venv) PS C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje\proje1> |
```

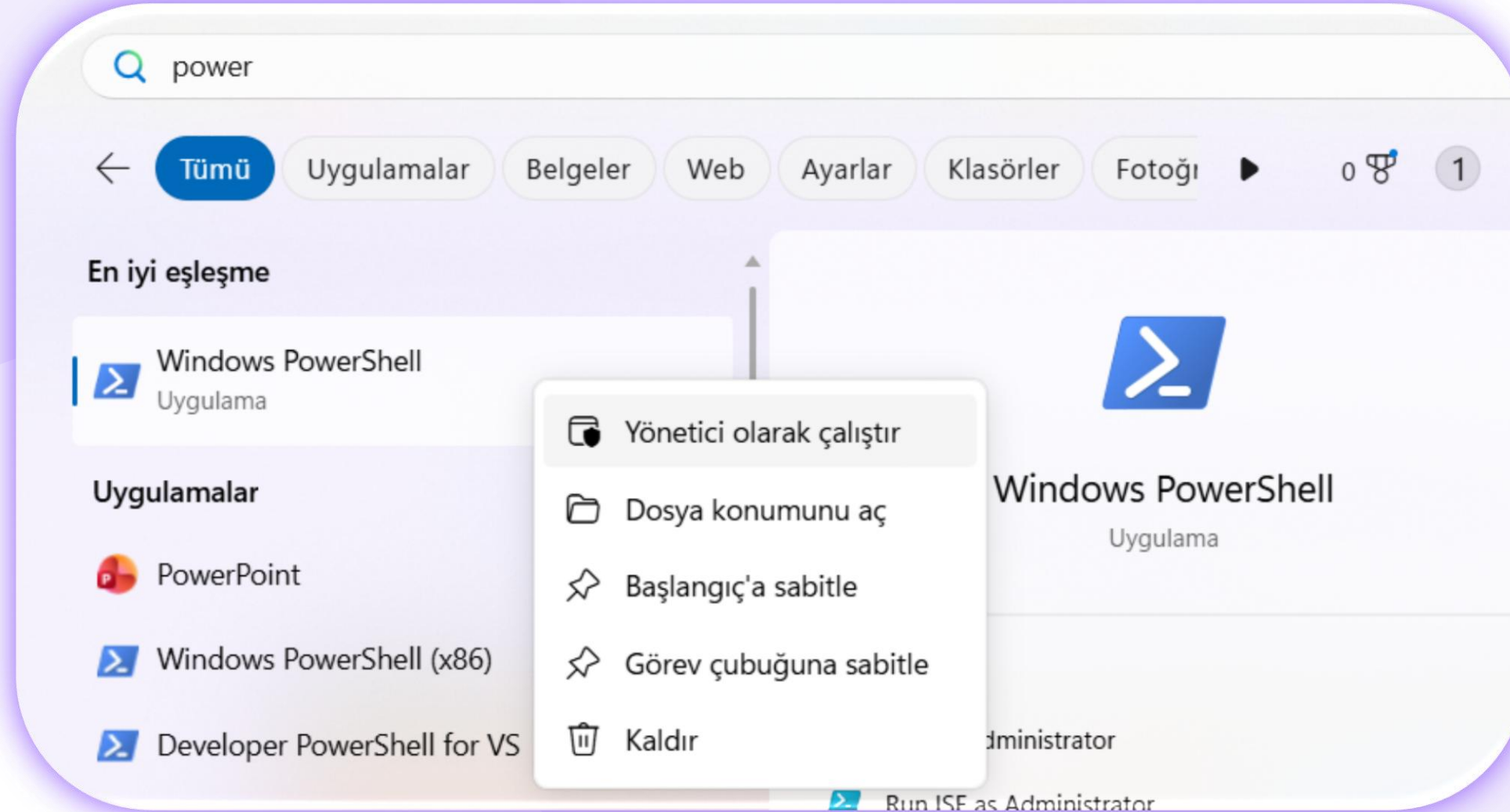
venv\Scripts\activate

venv Başlatma Hatası

```
PS C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje\proje1> & C:/Users/SAMU-Furkan_Durmus/Desktop/venvProje/proje1/venv/Scripts/Activate.ps1
& : File C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje\proje1\venv\Scripts\Activate.ps1 cannot be loaded because running scripts is disabled on this system. For more information, see about_Execution_Policies at https:/go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:3
+ & C:/Users/SAMU-Furkan_Durmus/Desktop/venvProje/proje1/venv/Scripts/A ...
+ ~~~~~
+ CategoryInfo          : SecurityError: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
```

Bu hatayı alırsanız PowerShell güvenlik ayarları güncellenmelidir.

venv Başlatma Hatası



PowerShell Yönetici olarak çalıştırılır.

venv Başlatma Hatası

Anasayfa / Dersler / Django İle Python Arka Yüz Yazılım Geliştirme

Django İle Python Arka Yüz Yazılım Geliştirme

- Ders Bilgisi
- Ders İzlenesi
- Ders Uygulamaları ^
- Sanal Ortam Hatası
- Ders Notları v

Sanal Ortam (venv) Aktivasyon Hatası ve Çözümü

Windows sistemlerde PowerShell üzerinde sanal ortamı aktif etmeye çalışırken (.venv\Scripts\activate) "running scripts is disabled on this system" uyarısı alıyorsanız, ilgili hata Windows'un komut çalıştırma politikasından kaynaklanmaktadır.

Hata Sebebi: İlgili sistemde scriptlerin çalıştırılması devre dışı bırakıldığından venv dosyası yüklenemiyor.

PowerShell Çözüm Komutu

```
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
```

Kopyala

<https://furkandurmus.com> - Dersler - Django - Ders Uygulamaları - Sanal Ortam Hatası sayfasındaki kodu kopyalayınız.

venv Başlatma Hatası

Administrator: Windows PowerShell

```
PS C:\WINDOWS\system32> Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
```

Kopyalanan kod PowerShell yapıştırılır ve enter'a tıklanır.

venv Başlatma Hatası

```
C:\WINDOWS\system32> Set-ExecutionPolicy RemoteSigned -Scope CurrentUser

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose
you to the security risks described in the about_Execution_Policies help topic at
https://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): A
```

Onay için A yazılıp tekrar enter'a tıklanır.

venv Başlatma

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + v [ ] [ ] ... | [ ] [ ] x  
PS C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje\proje1> & C:/Users/SAMU-Furkan_Durmus/Desktop/venvProje/proje1/venv/Scripts/python.exe c:/Users/SAMU-Furkan_Durmus/Desktop/venvProje/proje1/test.py  
merhaba  
PS C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje\proje1> venv\Scripts\activate  
(venv) PS C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje\proje1> |
```

Vs code yeniden başlatıldığında sanal ortam başarılı çalışacaktır.

venv\Scripts\activate

venv Test

```
(venv) PS C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje\proje1> pip install TRSehirler
```

pip install TRSehirler

venv Test

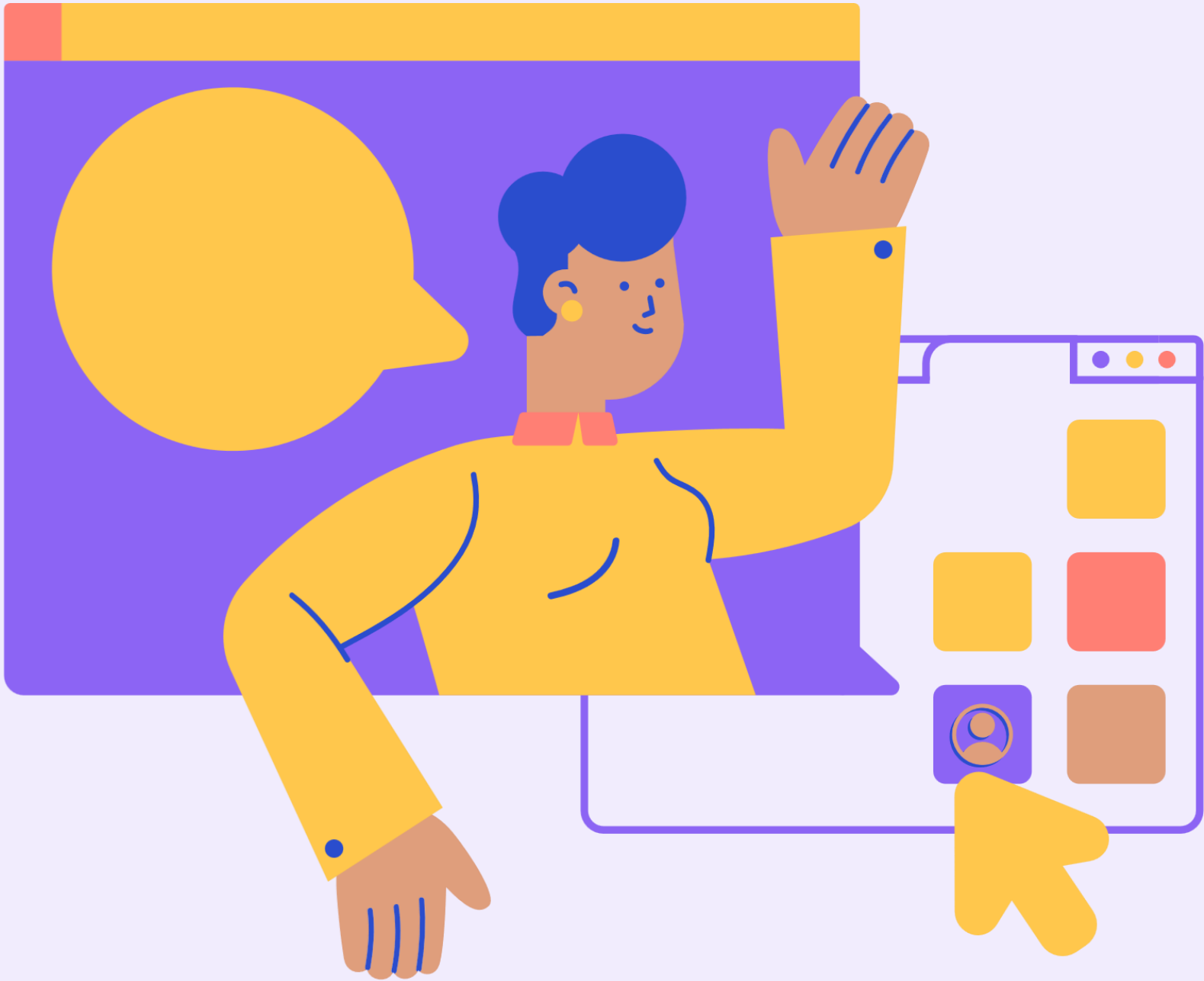
test.py dosyasına
yandaki kodu yazıp
çalıştırabilirsiniz.

```
test.py > ...
1  from TRSehirler import Sehir
2
3  sehir = Sehir()
4
5  print(sehir.il('Bursa'))
6
7  # print(sehir.tablo(sehir.plaka(16)))
8  # print(sehir.il('Bursa'))
9  # print(sehir.plaka(16))
10 # print(sehir.telefon(224))
11 # print(sehir.ilce('Atakum'))
12 # print(sehir.gorsel(sehir.plaka(16)))
13
```

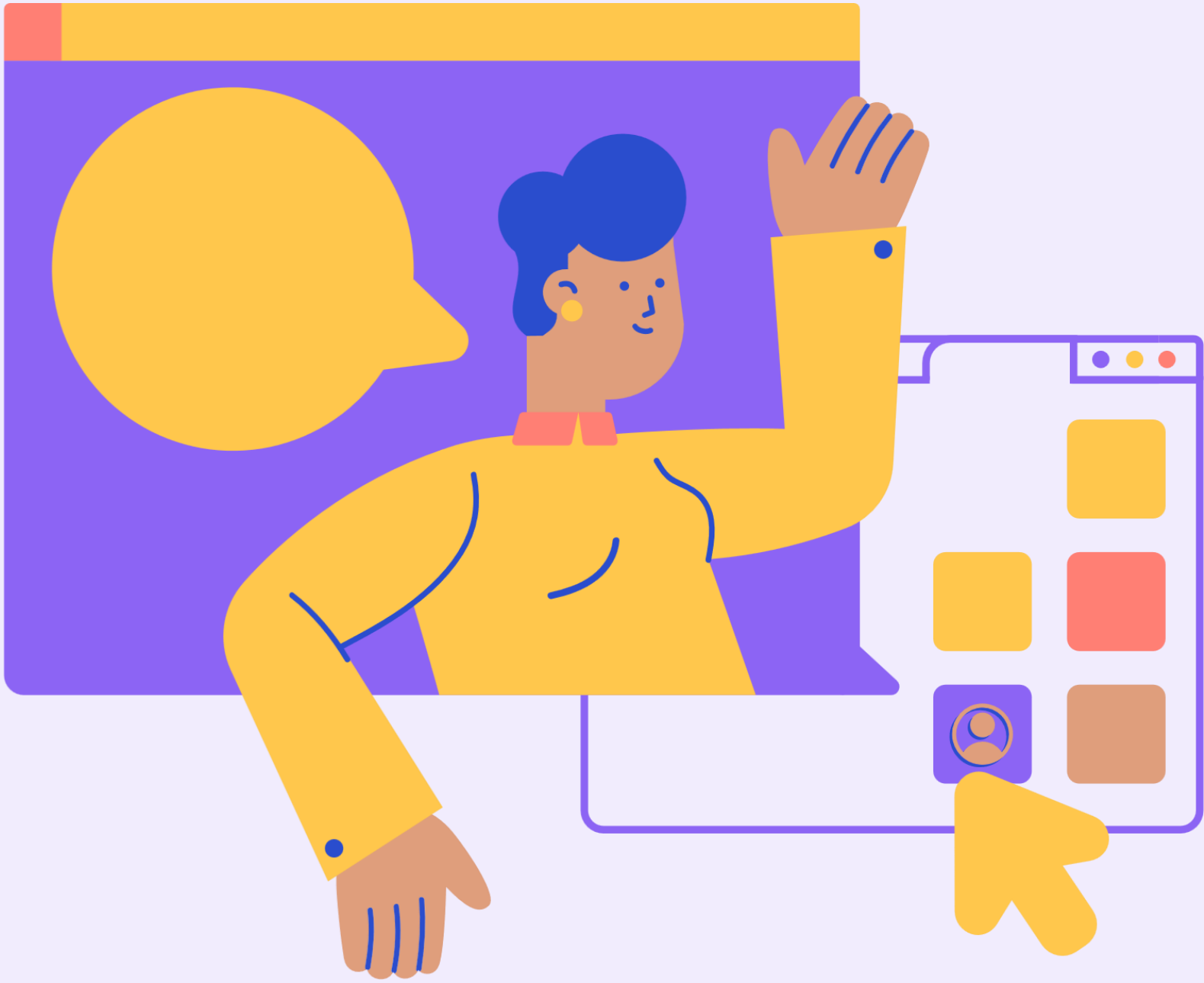
venv Test

```
(venv) PS C:\Users\SAMU-Furkan_Durmus\Desktop\venvProje\proje1> & C:/Users/SAMU-Furkan_Durmus/Desktop/venvProje/proje1/venv/Scripts/python.exe c:/Users/SAMU-Furkan_Durmus/Desktop/venvProje/proje1/test.py  
{'plaka': 16, 'il': 'Bursa', 'telefon': 224, 'buyuksehir_den_beri': 1987, 'bolge': 'Marmara', 'ilceler': ['Büyükorhan', 'Gemlik', 'Gürsu', 'Harmancık', 'İnegöl', 'İzmit', 'Karacabey', 'Keles', 'Kestel', 'Mudanya', 'Mustafakemalpaşa', 'Nilüfer', 'Orhaneli', 'Orhangazi', 'Osmangazi', 'Yenişehir', 'Yıldırım']}
```

Sanal ortamda başarılı bir şekilde bir kütüphane kuruldu ve çalıştırıldı.



Vs Code'u kapatın.
Proje1'in içindeki test.py' dosyasını
Proje2'nin içine kopyalayın ve
proje2'deki test.py dosyasını vs code
ile açın ve çalıştırın.



Vs Code'u kapatın.
Proje1'in içindeki test.py' dosyasını
Proje2'nin içine kopyalayın ve
proje2'deki test.py dosyasını vs code
ile açın ve çalıştırın.

Sonuç ?

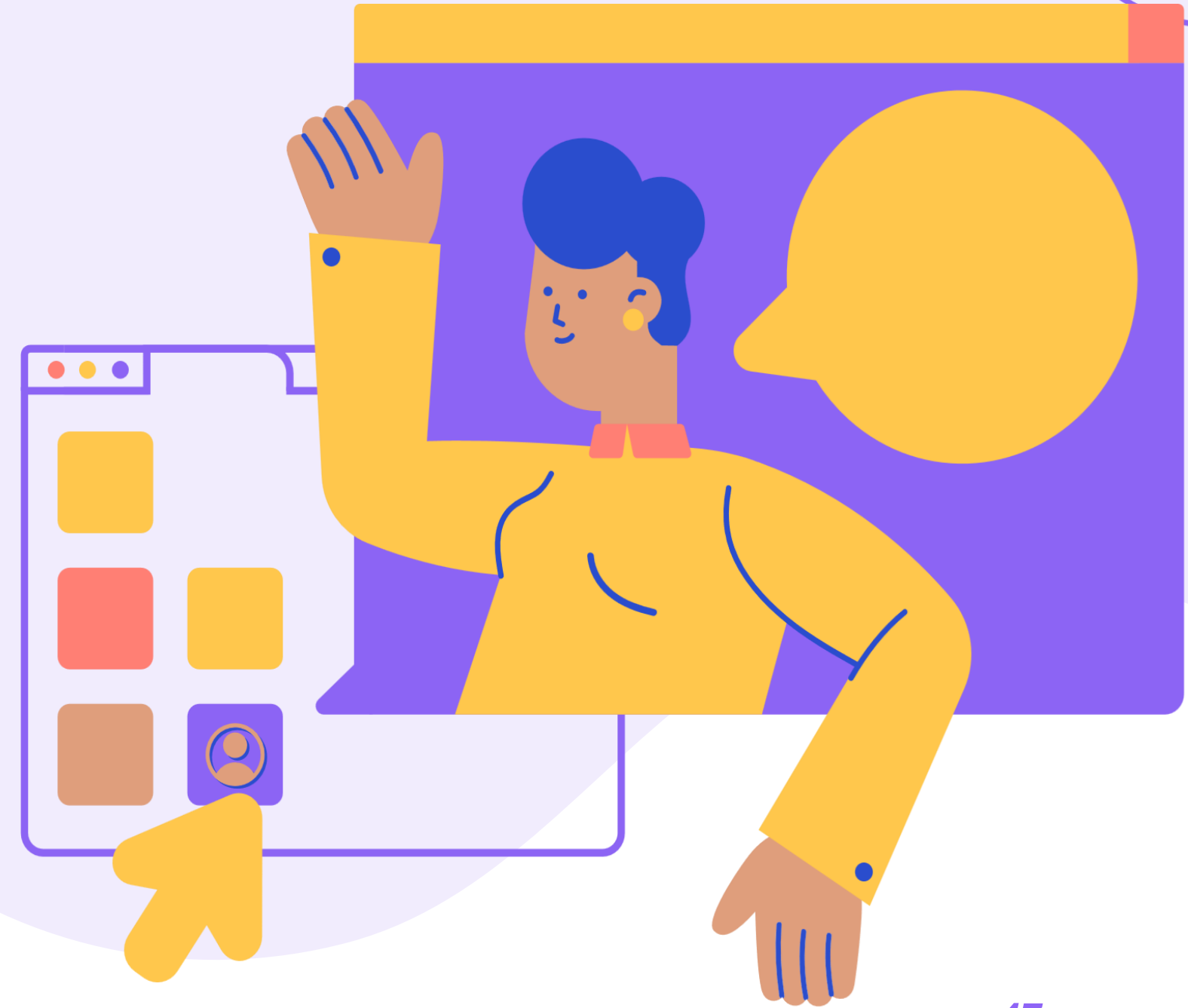
venv

```
PROJE2
test.py 1
test.py > ...
1 from TRSehirler import Sehir Import "TRSehirler" could not be resolved
2
3 sehir = Sehir()
4
5 print(sehir.il('Bursa'))
6
```

Proje 1 içerisinde kurulan kütüphane venv sayesinde global Python'a kurulmadığı için Proje 2'de tanınmadı.

En Çok Yaptığı Hatalar

- ✗ venv oluşturmadan pip install yapmak
- ✗ activate etmeyi unutmak
- ✗ Yanlış klasörde çalışmak





```
require 'capybara/reails'
```

```
1 Capybara.javascript_driver = :webkit
```

```
2 Category.delete_all; Category.create
```

```
13 Shoulda::Matchers.configure do |config|
```

```
14 config.integrate do |with|
```

```
15 with.test_framework :rspec
```

```
16 with.library :rails
```

```
17 end
```

```
18 end
```

```
19 # Add additional requires below this line. 20
```

```
21 # Requires supporting ruby files with support/
```

```
22 # spec/support/ and its subdirectories. 23
```

```
24 # run as spec files by default. 25
```

```
25 # in _spec.rb will both be required and run.
```

Django Kurulumu

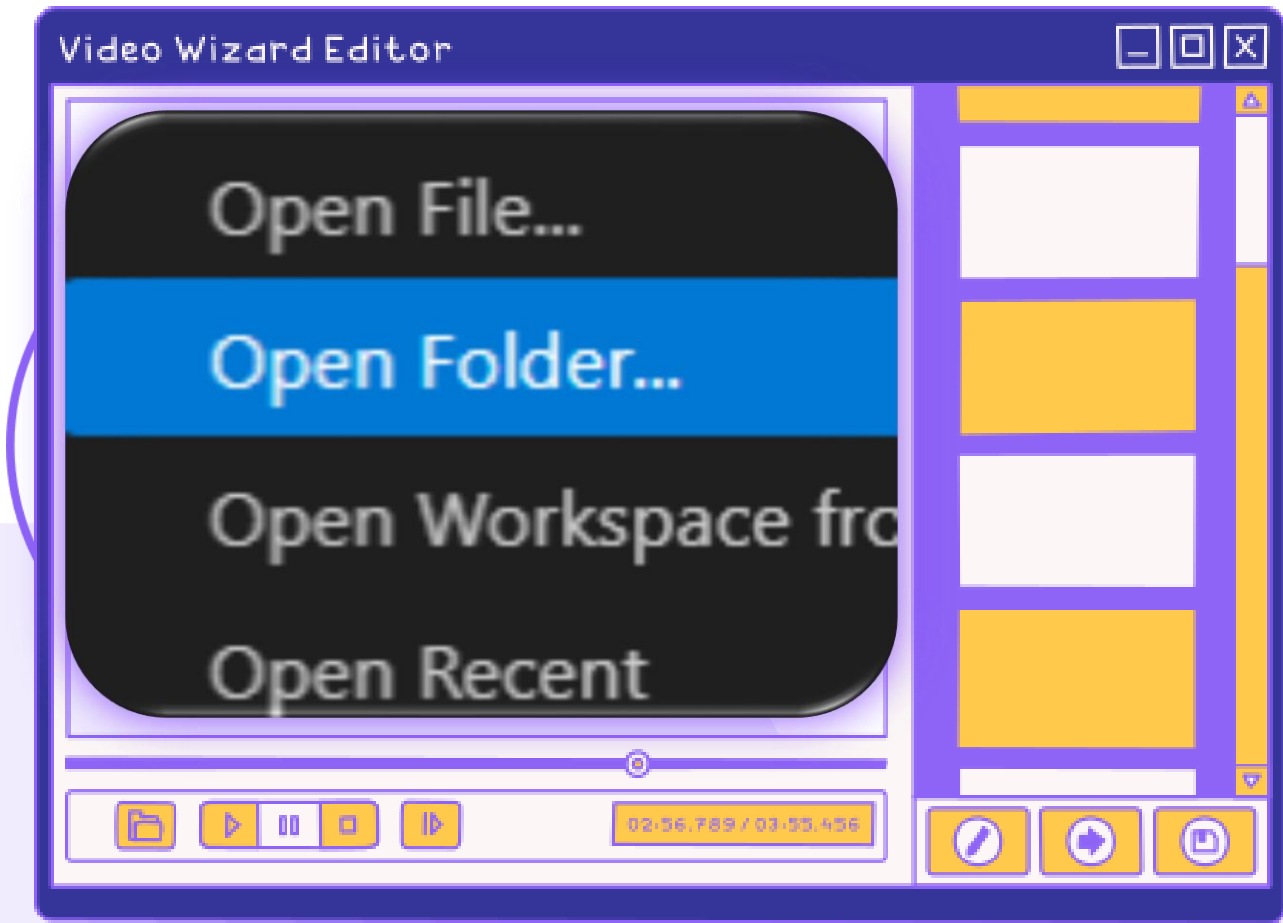
Django Kurulumu

- Masaüstü'nde **kampus_blog** adında bir klasör oluşturun.



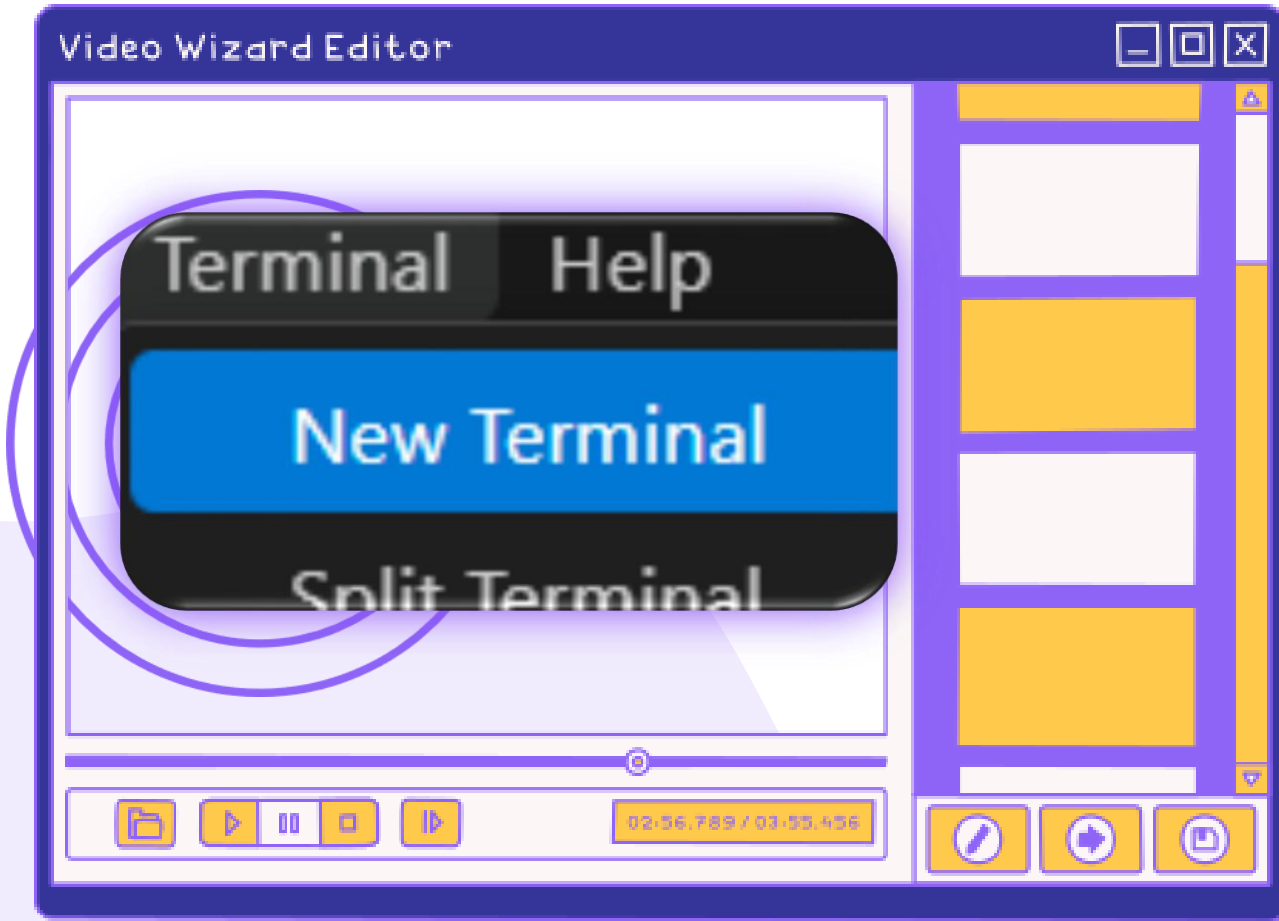
Django Kurulumu

- Masaüstü'nde kampus_blog adında bir klasör oluşturun.
- Vs code'da Klasör Aç'a tıklayarak ilgili klasörü seçin.



Django Kurulumu

- Masaüstü'nde kampus_blog adında bir klasör oluşturun.
- Vs code'da Klasör Aç'a tıklayarak ilgili klasörü seçin.
- Üst menüden terminal > yeni terminal ile terminali açın.



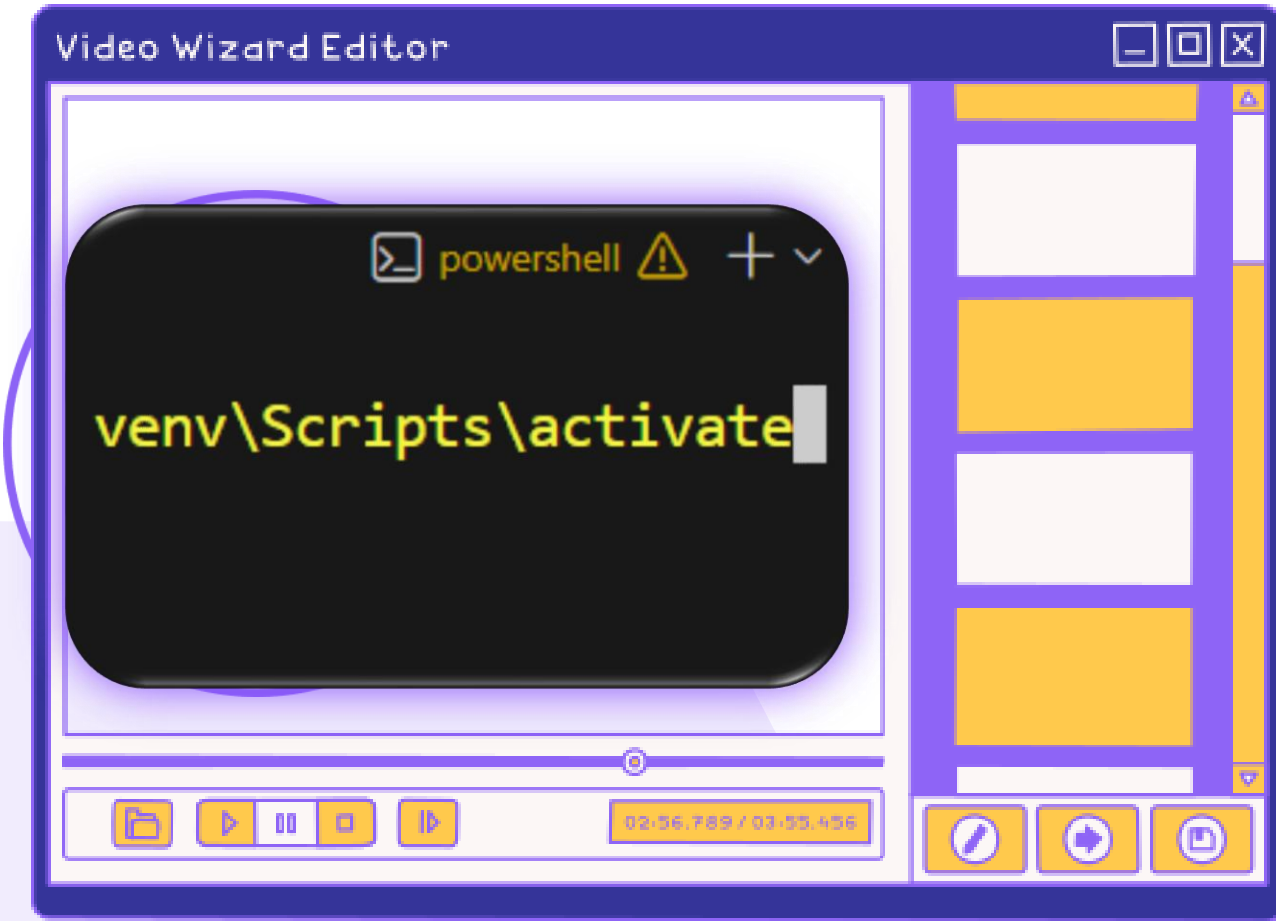
Django Kurulumu

- Masaüstü'nde kampus_blog adında bir klasör oluşturun.
- Vs code'da Klasör Aç'a tıklayarak ilgili klasörü seçin.
- Üst menüden terminal > yeni terminal ile terminali açın.
- İçerisine venv adında bir sanal ortam kurun.



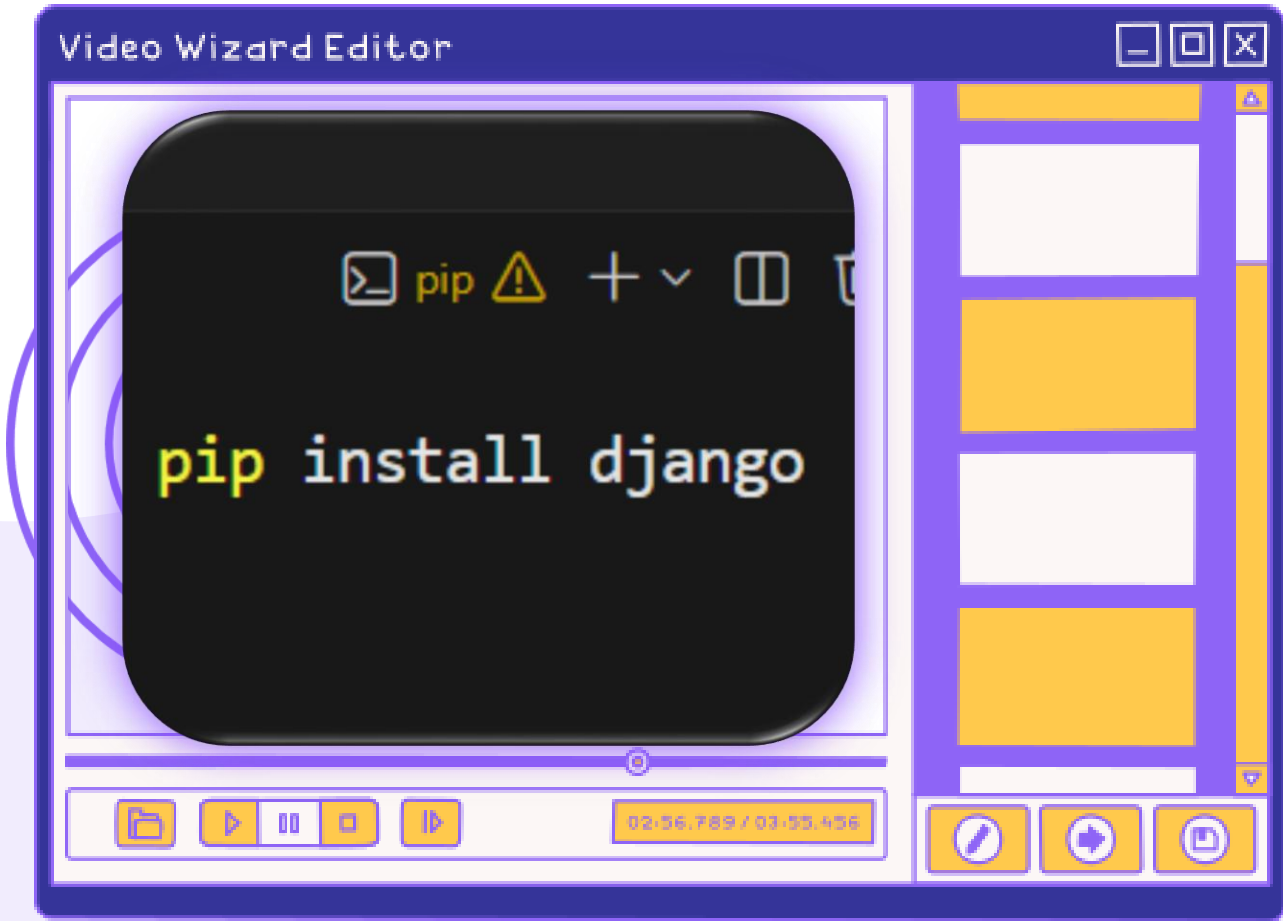
Django Kurulumu

- Masaüstü'nde kampus_blog adında bir klasör oluşturun.
- Vs code'da Klasör Aç'a tıklayarak ilgili klasörü seçin.
- Üst menüden terminal > yeni terminal ile terminali açın.
- İçerisine venv adında bir sanal ortam kurun.
- Sanal ortamı başlatın.



Django Kurulumu

- Masaüstü'nde kampus_blog adında bir klasör oluşturun.
- Vs code'da Klasör Aç'a tıklayarak ilgili klasörü seçin.
- Üst menüden terminal > yeni terminal ile terminali açın.
- İçerisine venv adında bir sanal ortam kurun.
- Sanal ortamı başlatın.
- **pip install django** yazarak django kurulumunu tamamlayın.



Django Kurulumu

pip list ile Python içerisindeki mevcut paketleri görüntüleyebilirsiniz. Django'nun kurulduğunu **pip list** ile teyit ediniz.

```
(venv) PS C:\Users\SAMU-Furkan_Durmus\Desktop\kapmus_blog> pip list
Package Version
-----
asgiref  3.11.1
Django   6.0.2
pip      25.2
sqlparse 0.5.5
tzdata   2025.3
```





```
require 'capybara/re Rails'
```

```
1 Capybara.javascript_driver = :webkit
```

```
2 Category.delete_all; Category.create
```

```
13 Shoulda::Matchers.configure do |config|
```

```
14 config.integrate do |with|
```

```
15 with.test_framework :rspec
```

```
16 with.library :rails
```

```
17 end
```

```
18 end
```

```
19 # Add additional requires below this line. The
```

```
20 # Requires supporting ruby files with support
```

```
21 # spec/support/ and its subdirectories. These
```

```
22 # run as spec files by default. This will
```

```
23 # in _spec.rb will both be required and
```

```
24 It is recommended that you configure the
```

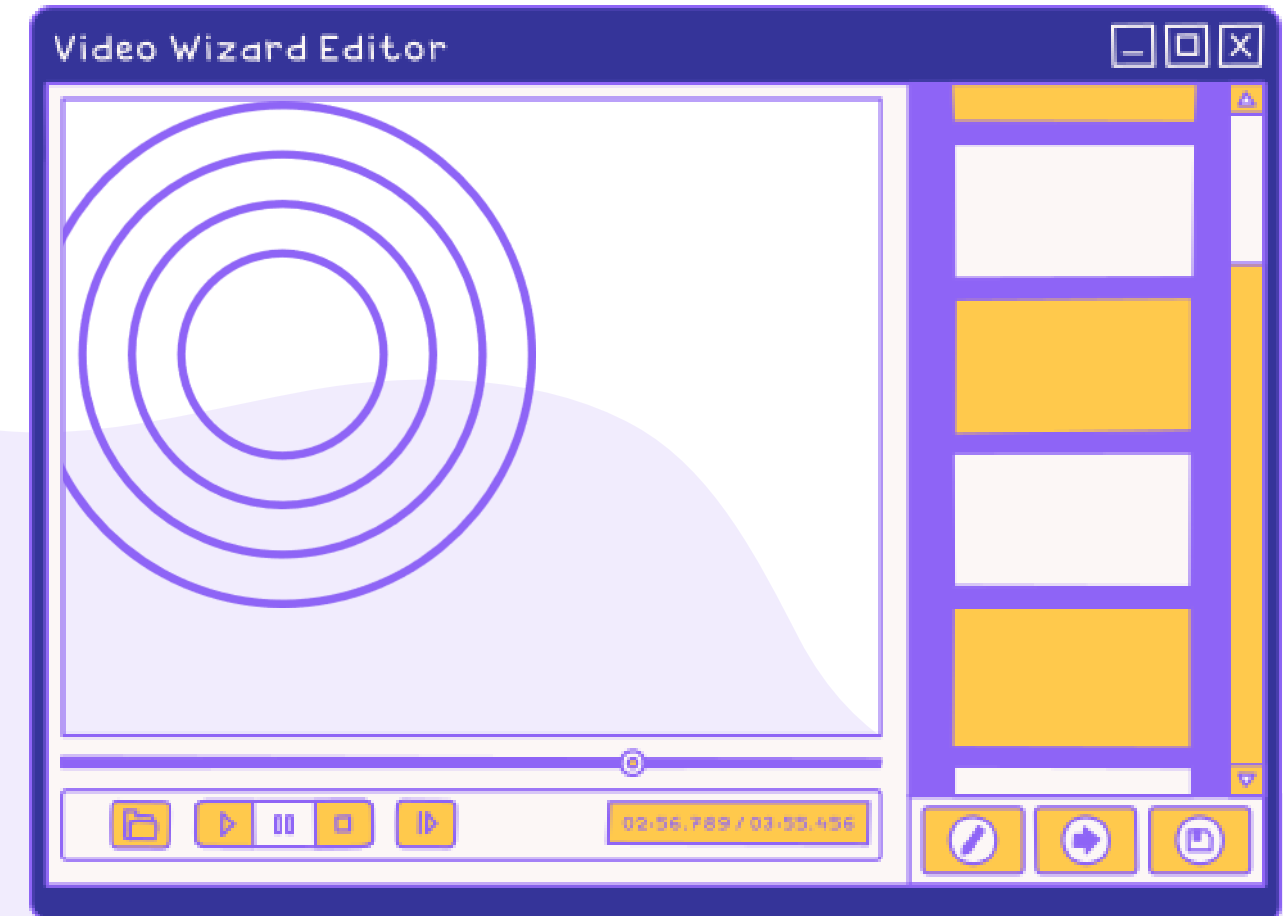
startproject

startproject

startproject şunu yapar:

- Django'nun çalışması için gerekli çekirdek dosyaları üretir.
- Ayar dosyasını (settings) oluşturur.
- URL yönlendirme dosyasını (urls) oluşturur.
- Sunucu çalıştırma altyapısını hazırlar.

“Boş ama çalışan bir Django projesi iskeleti” oluşturur.





startproject

kampus_blog içerisinde oluşturduğunuz
venv'i başatın.



startproject

```
(venv) PS C:\Users\SAMU-Furkan_Durmus\Desktop\kapmus_blog> django-admin startproject config .
```

django-admin startproject config .

startproject

```
(venv) PS C:\Users\SAMU-Furkan_Durmus\Desktop\kapmus_blog> django-admin startproject config .
```

django-admin startproject config .

Bu komut ne demek?

config → Django projesinin iç ayar paketinin adı (settings/urls burada olur)

. → “bulduğum klasörün içine kur” demektir

startproject

```
kampus_blog/  
├── venv/  
├── manage.py  
├── config/  
│   ├── __init__.py  
│   ├── settings.py  
│   ├── urls.py  
│   ├── asgi.py  
│   ├── asgi.py  
│   └── wsgi.py
```

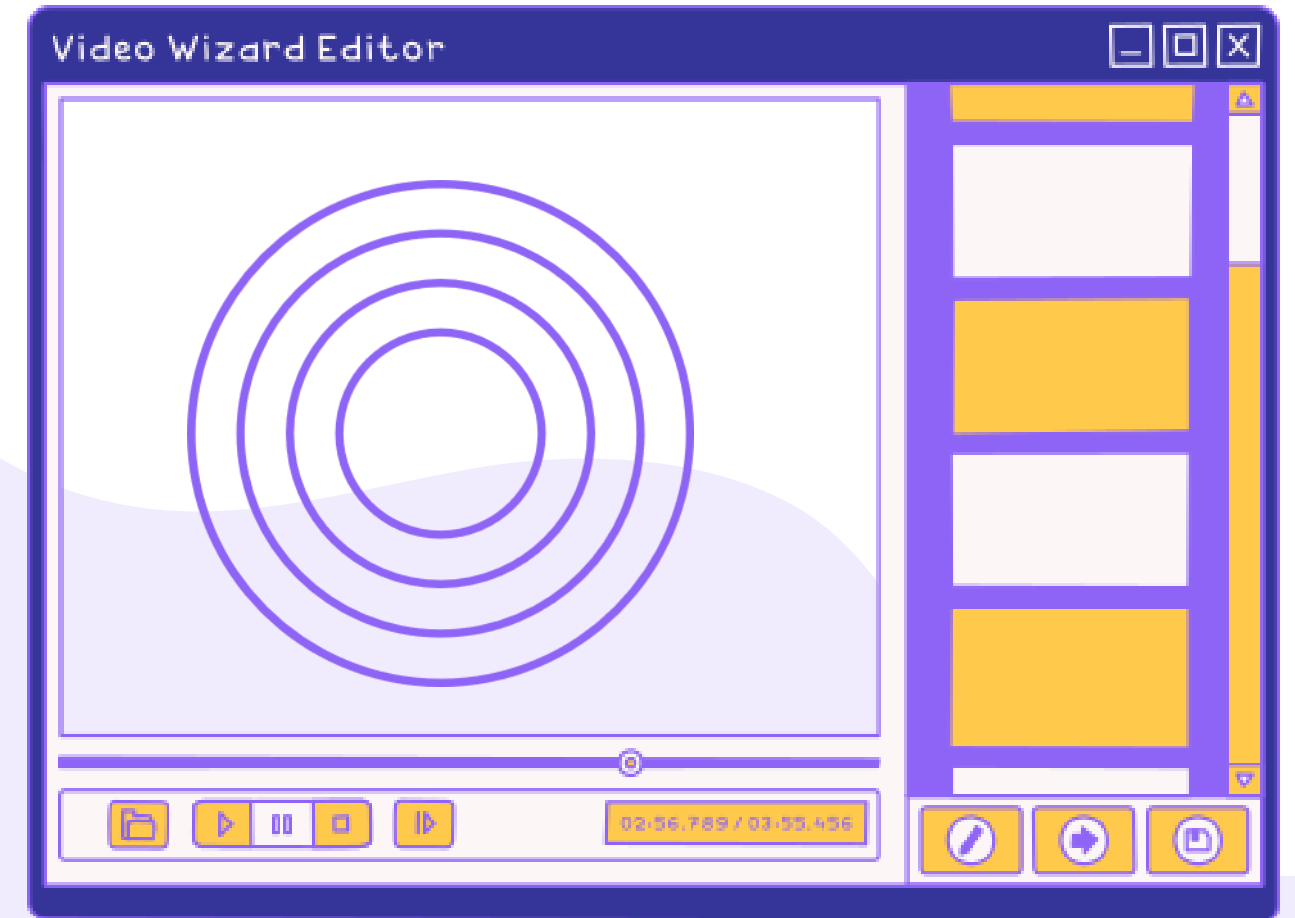
Mevcut klasör yapısı

startproject

Klasörler ne işe yarar ?

manage.py :

- Projeyi yönetmek için komut aracıdır. Örnek: sunucu çalıştırma, migration, app oluşturma.



startproject

Klasörler ne işe yarar ?

settings.py :

- Veritabanı
- Uygulamalar
- Güvenlik
- dil/saat
- static ayarları

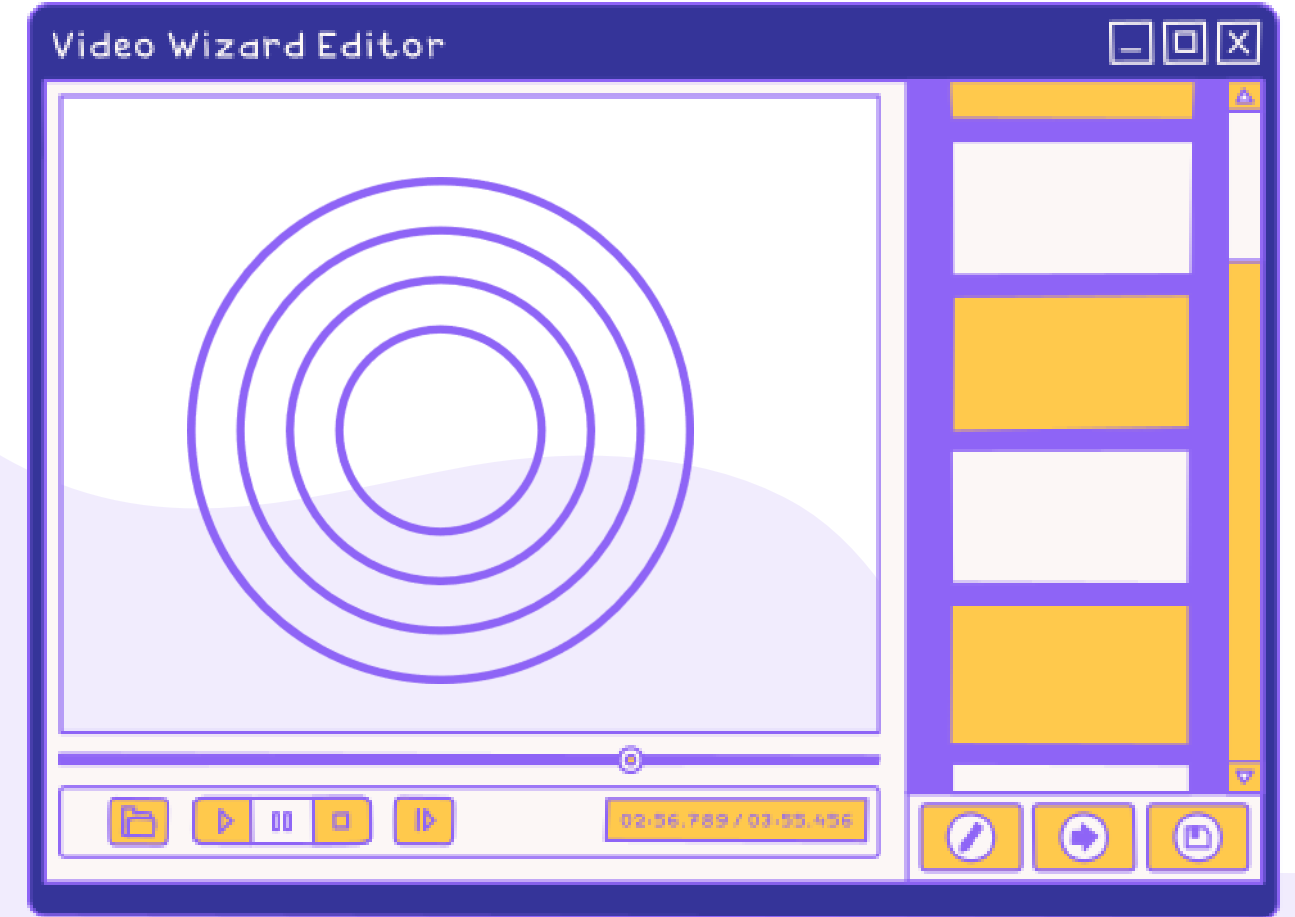


startproject

Klasörler ne işe yarar ?

urls.py :

- Siteye gelen URL isteklerinin "hangi view'a gideceğini" belirler.





```
require 'capybara/re Rails'
```

```
1 Capybara.javascript_driver = :webkit
```

```
2 Category.delete_all; Category.create
```

```
3 Shoulda::Matchers.configure do |config|
```

```
4   config.integrate do |with|
```

```
5     with.test_framework :rspec
```

```
6     with.library :rails
```

```
7   end
```

```
8 end
```

```
9 # Add additional requires below this line.  When you require a
```

```
10 # Requires supporting ruby files with support/ directory.  The
```

```
11 # spec/support/ and its subdirectories.  These files will be
```

```
12 # run as spec files by default.  This will be overridden by
```

```
13 # in _spec.rb will both be required and run as spec files.
```

```
14 It is recommended that you configure the support directory
```

runserver



Şimdi proje hazır. Test ediyoruz:

```
python manage.py runserver
```

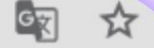
runproject

```
(venv) PS C:\Users\SAMU-Furkan_Durmus\Desktop\kapmus_blog> python manage.py runserver
Run 'python manage.py migrate' to apply them.
February 19, 2026 - 17:39:10
Django version 6.0.2, using settings 'config.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead.
```

Browserdan <http://127.0.0.1:8000> açınız
Yada ctrl + sol tık ile ekrandaki adrese tıklayınız.

127.0.0.1:8000



The install worked successfully! Congratulations!

View [release notes](#) for Django 6.0

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

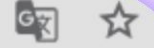
django

 **Django Documentation**
Topics, references, & how-to's

 **Tutorial: A Polling App**
Get started with Django

 **Django Community**
Connect, get help, or contribute

127.0.0.1:8000



“It worked!” sayfasını görürsen:

- Proje doğru kuruldu
- venv doğru
- Django doğru
- startproject doğru



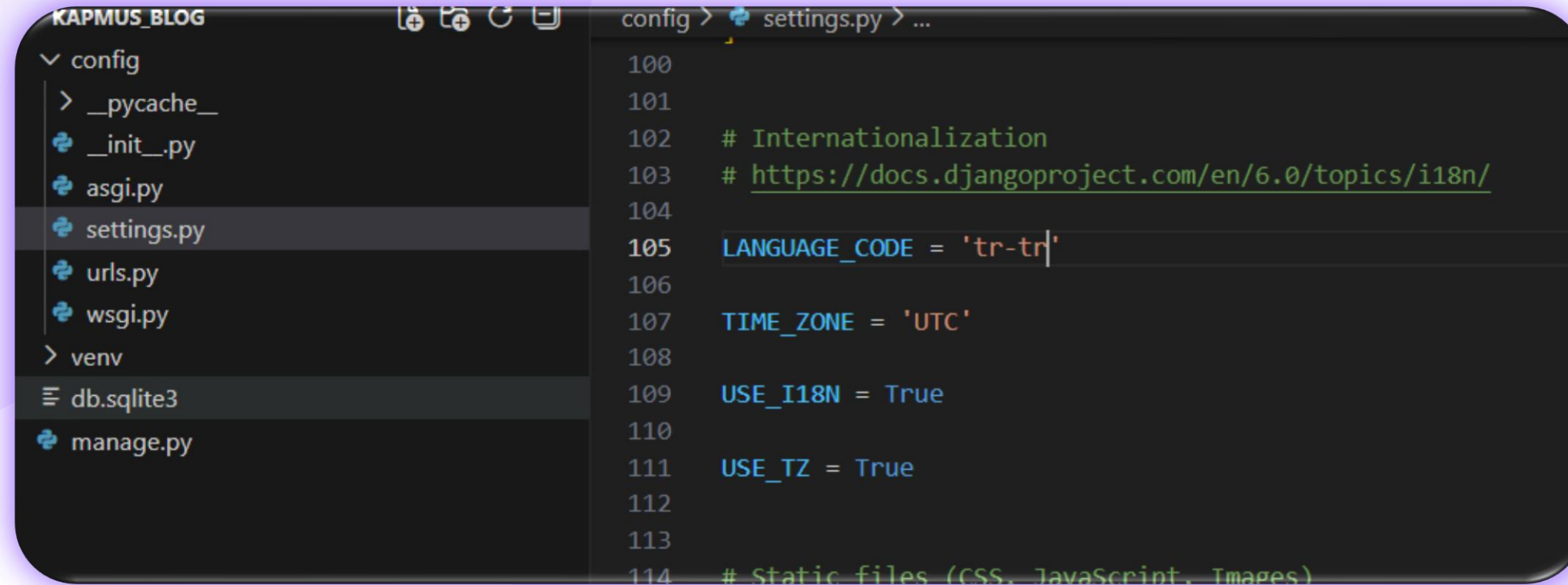
The install worked successfully! Congratulations!

View [release notes](#) for Django 6.0

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

django

runproject



```
KAPMUS_BLOG
├── config
│   ├── __pycache__
│   ├── __init__.py
│   ├── asgi.py
│   ├── settings.py
│   ├── urls.py
│   └── wsgi.py
├── venv
├── db.sqlite3
└── manage.py

config > settings.py > ...
100
101
102 # Internationalization
103 # https://docs.djangoproject.com/en/6.0/topics/i18n/
104
105 LANGUAGE_CODE = 'tr-tr'
106
107 TIME_ZONE = 'UTC'
108
109 USE_I18N = True
110
111 USE_TZ = True
112
113
114 # Static files (CSS, JavaScript, Images)
```

Setting içerisindeki LANGUAGE_CODE 'u 'tr-tr' olarak güncelleyin.

Ctrl + S ile dosyayı kayıt edin.

Web sayfası auto reload ile güncellenecek.

runproject



Yükleme başarılı olarak çalıştı! Tebrikler!

Django 6.0 sürümü için [yayım notlarını](#) göster

Bu sayfayı görüyorsunuz çünkü `DEBUG=True` parametresi ayarlar dosyanızın içindedir ve herhangi bir URL yapılandırmadınız.

django



App

Django'da App Nedir?

App = Belirli bir işi yapan modül

Yani Django projesi küçük parçalara bölünür. Her parça kendi işinden sorumludur.





App

Django'da App Nedir?

Bir blog sitesinde:

Blog yazıları → blog app

Kullanıcı işlemleri → kullanıcı app

Ana sayfa & statik sayfalar → core app

Hepsi tek proje içinde ama ayrı ayrı çalışır.



App

Neden App Var?

Çünkü Django büyük projeleri düzenli tutmak için tasarlanmıştır.

App sayesinde:

- ✓ Kod düzenli olur
- ✓ Proje büyüdükçe karışmaz
- ✓ Tekrar kullanılabilir olur
- ✓ Test etmek kolay olur.



Proje Yapısı

```
kampus_blog/      ← Proje ana klasörü
├── venv/          ← Sanal ortam (kod yazılmaz)
├── manage.py     ← Proje yönetim dosyası
├── config/       ← Proje ayarları (settings, urls)
│   ├── __init__.py
│   ├── settings.py
│   ├── urls.py
│   ├── asgi.py
│   └── wsgi.py
├── core/         ← App (Ana sayfa, hakkımızda, iletişim)
├── blog/        ← App (Blog yazıları ve kategoriler)
└── kullanıcı/   ← App (Kayıt, giriş, profil işlemleri)
```

Kampus_blog

```
(venv) PS C:\Users\SAMU-Furkan_Durmus\Desktop\kapmus_blog> python manage.py startapp core
```

Ctrl + c ile mevcut server'ı durdurun. Ardından :
python manage.py startapp core

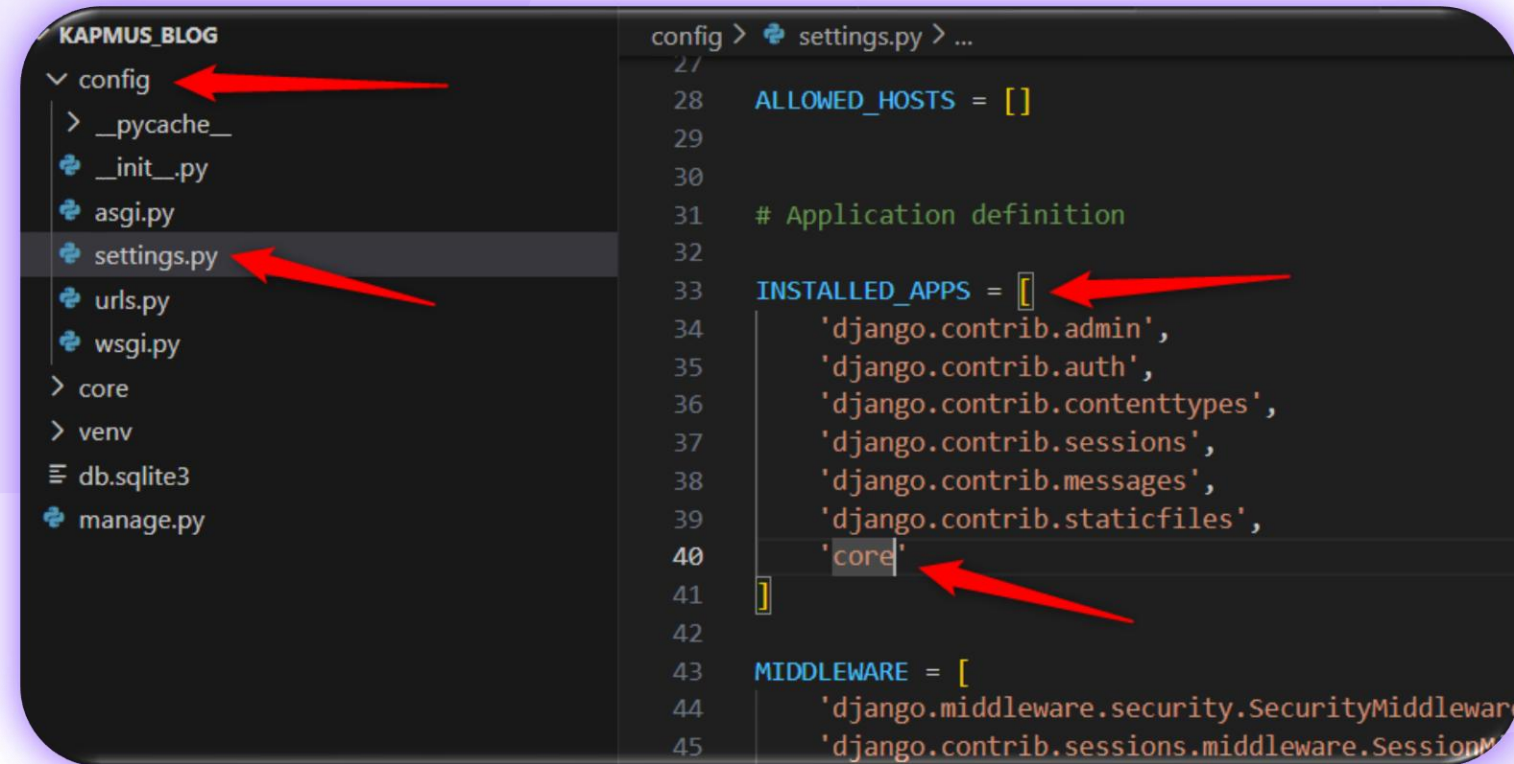


Kampus_blog

Projeye 'core' adında bir app ekledik
şimdi bu app'i aktif etmeliyiz.
Böylece django çalışırken bu app'i de
çalıştıracak.



Kampus_blog



```
KAMPUS BLOG
└─ config
  └─ __pycache__
  └─ __init__.py
  └─ asgi.py
  └─ settings.py
  └─ urls.py
  └─ wsgi.py
  └─ core
  └─ venv
  └─ db.sqlite3
  └─ manage.py

config > settings.py > ...
27
28 ALLOWED_HOSTS = []
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'core'
41 ]
42
43 MIDDLEWARE = [
44     'django.middleware.security.SecurityMiddleware',
45     'django.contrib.sessions.middleware.SessionM
```

Projeye 'core' adında bir app ekledik şimdi bu app'i aktif etmeliyiz.

Böylece django çalışırken bu app'i de çalıştıracak.

Config > settings.py > INSTALLED_APPS > 'core'

Kampus_blog

Core içinde bir anasayfa oluşturacağız.
Bunun için ilk önce views katmanında
tanımlama yapmalıyız.



Kampus_blog

```
core > views.py > ...  
1 from django.shortcuts import render  
2  
3 def anasayfa(request):  
4     return render(request, "anasayfa.html")  
5
```

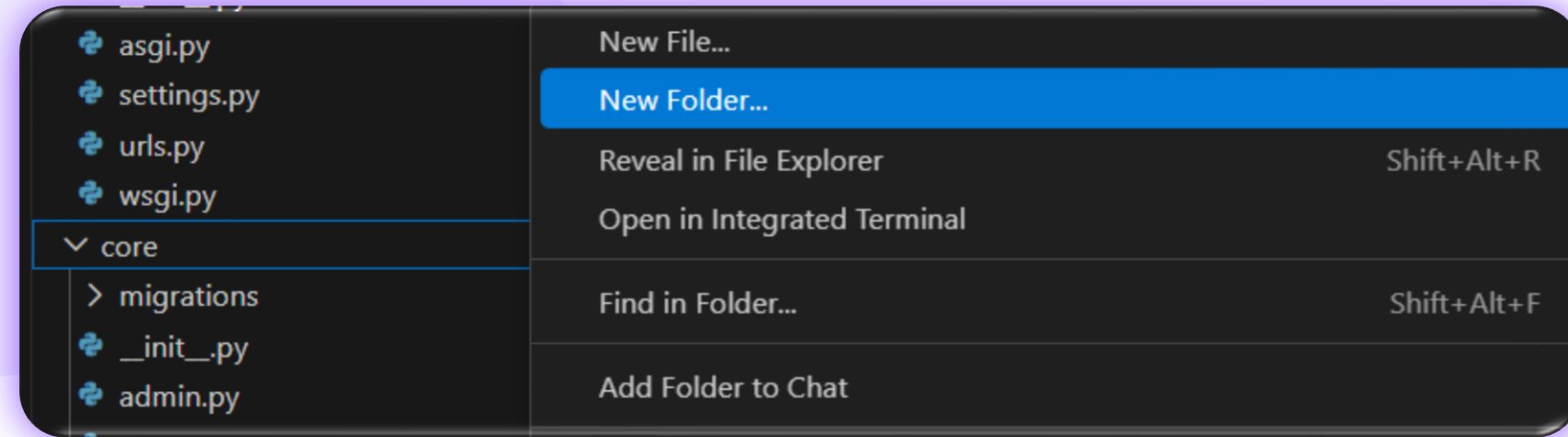
Core > views.py

Kampus_blog

Views katmanında bir anasayfa fonksiyonu belirledik ve bu fonksiyon çağrıldığında nereye gideceğini belirttik fakat henüz template yapımız yok.

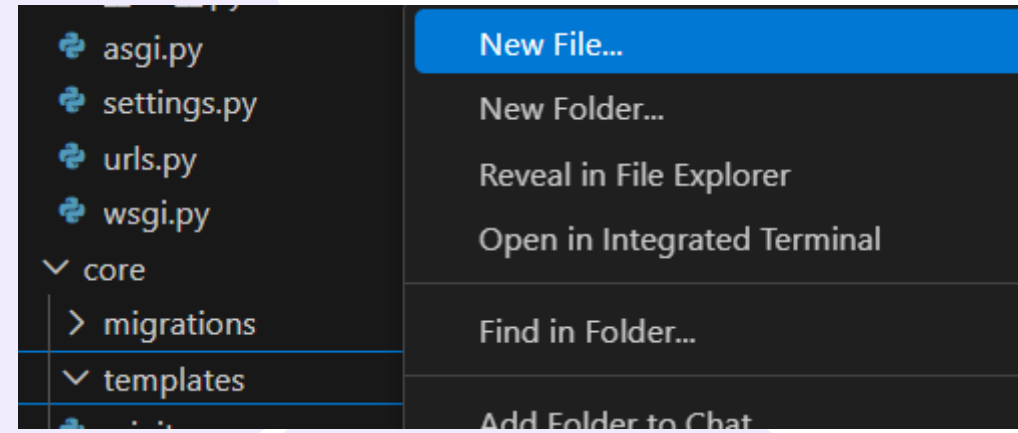


Kampus_blog



Core > sağ tık > new folder > templates

Kampus_blog



core > templates > sağ tık > new file > anasayfa.html

Kampus_blog

```
✓ core
  > migrations
  ✓ templates
    <> anasayfa.html
```

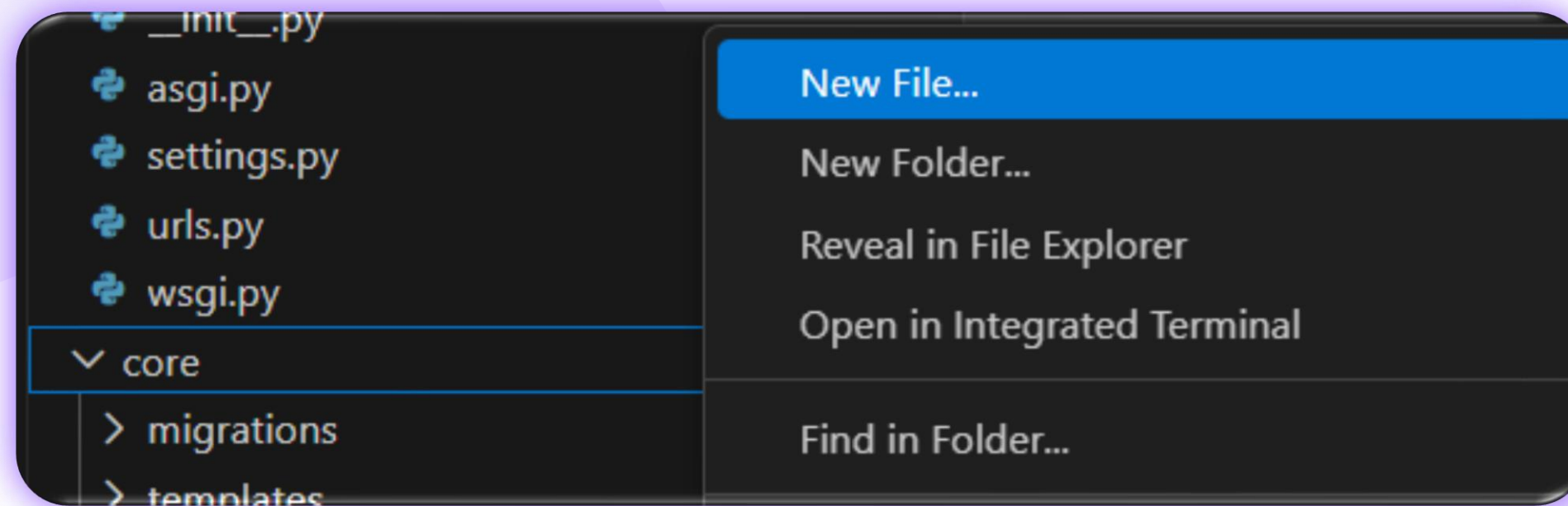
Güncel yapı

Kampus_blog

```
core > templates > <> anasayfa.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Kampüs Blog</title>
5  </head>
6  <body>
7  |   <h1>Kampüs Blog Sitesine Hoş Geldiniz</h1>
8  |   <p>Bu site Django ile geliştirilmektedir.</p>
9  </body>
10 </html>
11
```


Basit bir anasayfa html kodu yazınız.

Kampus_blog



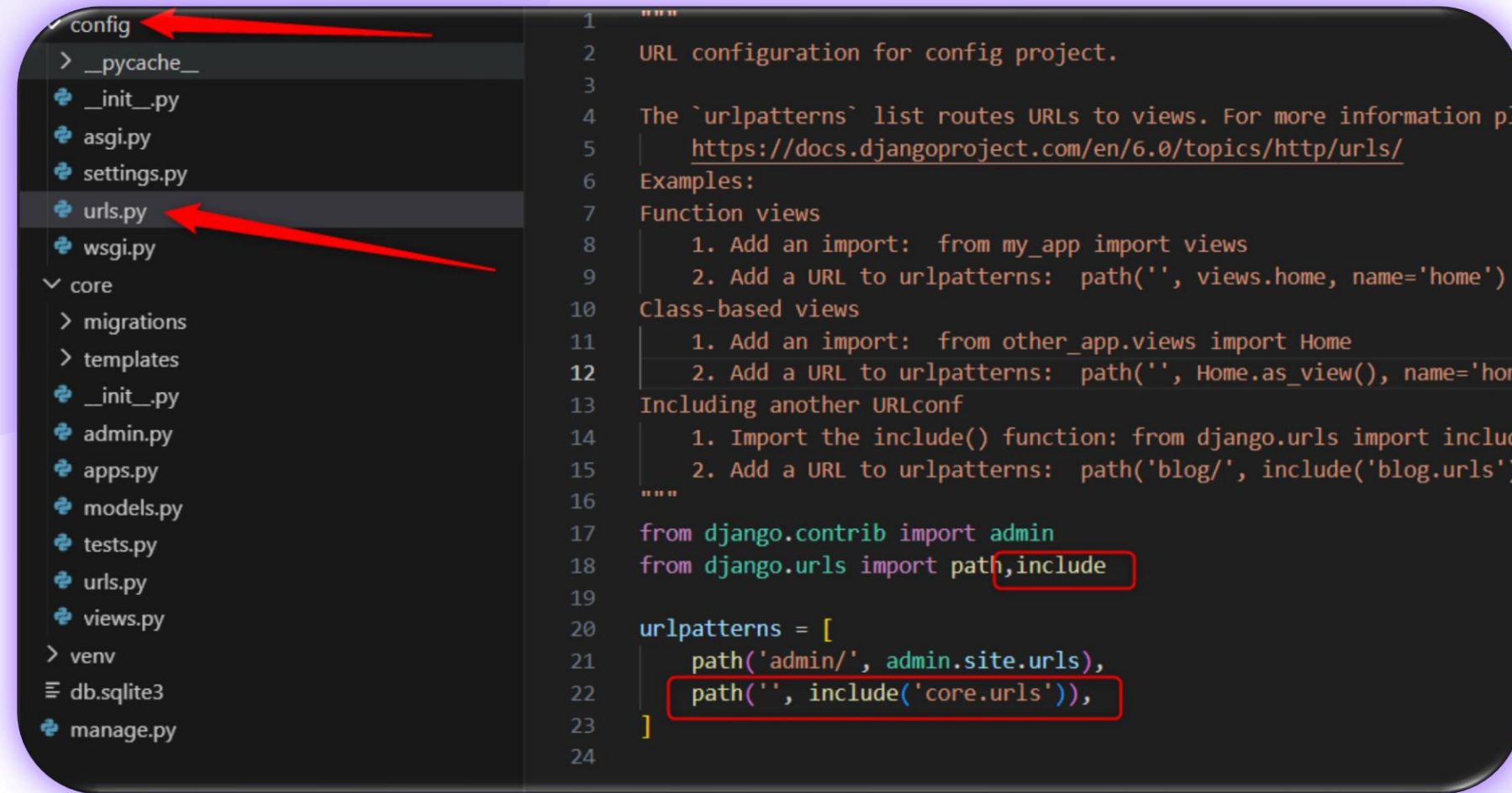
Core > new file > urls.py

Kampus_blog

```
core >  urls.py > ...  
1   from django.urls import path  
2   from . import views  
3  
4   urlpatterns = [  
5       path('', views.anasayfa, name='anasayfa'),  
6   ]  
7
```

İlgili kodu core > urls.py içine eksiksiz yazınız.

Kampus_blog



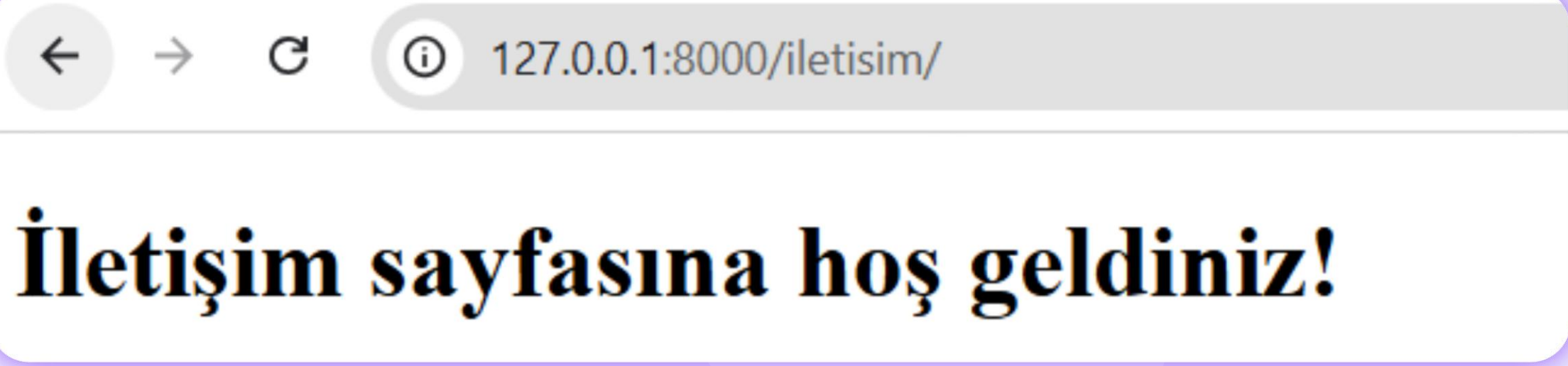
```
1 """
2 URL configuration for config project.
3
4 The `urlpatterns` list routes URLs to views. For more information pl
5 https://docs.djangoproject.com/en/6.0/topics/http/urls/
6 Examples:
7 Function views
8     1. Add an import: from my_app import views
9     2. Add a URL to urlpatterns: path('', views.home, name='home')
10 Class-based views
11     1. Add an import: from other_app.views import Home
12     2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
13 Including another URLconf
14     1. Import the include() function: from django.urls import include
15     2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
16 """
17 from django.contrib import admin
18 from django.urls import path, include
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('', include('core.urls')),
23 ]
24
```

Son olarak config urls.py içerisine config urls'lerini ekleyelim.

Kampus_blog

İletişim sayfasını siz ekleyiniz.

- İletişim sayfası core app'i içinde olacak.
- Core views içerisine bir fonksiyon tanımlayınız ve bu fonksiyon iletisim.html'i açsın.
- Core templates altında iletisim.html oluşturun ve içine basit html yazın.
- Core urls.py altında 'iletisim/' yazıldığında views'te oluşturulan fonksiyon çağrılсын.



Son