

Flask Dökümantasyonu

Flask ,python dilinden yazılmış web alanında kullanılan Django'ya kıyasla daha basit bir kütüphanedir.Bu kütüphane HTML şablonlarını dinamik içeriklerle birleştirmek için Jinja2 şablon motorunu kullanır ve eklentilerle genişletilebilir bir kütüphanedir

Terminalde “ pip install Flask” yazarak Flask'ı kurabilirsiniz

Bir Flask uygulaması oluşturmak için:

```
from flask import Flask
```

```
app = Flask(__name__)
```

```
@app.route('/')
```

```
def hello():
```

```
    return 'Merhaba, Dünya!'
```

```
if __name__ == '__main__':
```

```
    app.run(debug=True)
```

Bu kütüphanede “@app.route” dekoratörü ile URL rotaları tanımlayıp bu decaratörün altına yazılan fonksiyonlarla da o sayfada ne yapılmasını istiyorsanız ona göre kod yazabilirsiniz

Üstte yazılan kodda en altındaki kod satırında app.run(debug=True) bu şu an Flaskın debug modunda çalıştığını gösterir Web siteyi yayımlarken bu debug=True ifadesini kaldırmanız gerek.

Dinamik bir URL yapısı istiyorsanız:

“@app.route(“/sayfaadi/<string:id>”)”

şeklinde dekoratörü ayarlayabilirsiniz .

#render_template nedir ve nasıl kullanılır?

render_template jinja2 motorunu kullanarak html şablonlarını dinamik hale getirmeyi sağlayan bir araçtır kullanmak için önce flask projenizin olduğu klasöre “templates” adında yeni klasör oluşturun ama yazımına dikkat edin çünkü flask direk bu isimde klasör aramaya ayarlanmıştır.Bu klasöre html dosyalarınızı atıp:

@app.route("/")

def index():

return render_template("index.html")

şeklinde kullanabilirsiniz.

#@app.route() dekoratörünün içine methods diye liste tanımlayıp “GET” ,”POST” gibi istekleri kabul edecek hale getirebilirsiniz.

Sitenizin CSS işlemlerini proje klasörünüzde “static” isminde bir klasör oluşturup içine CSS dosyanızı ekleyebilirsiniz