**Een Website Maken met Flask in Python: Een Stap-voor-Stap Tutorial**

**Inleiding**

In deze tutorial ga je een simpele website maken met behulp van **Flask**, een populaire webframework in Python. Flask is een goede beginners omdat het heel envoudig is en je snel kunt leren hoe je ermee een website bouwt. Of je nu een hobbyist bent die nieuwsgierig is naar webontwikkeling of iemand die een webproject wil starten, deze tutorial is voor jou.

We gaan stap voor stap door het proces, zodat je begrijpt wat je doet en waarom. Aan het einde van deze tutorial kun je je eigen website draaien op je computer! Dus, open je teksteditor en laten we beginnen!

**Wat heb je nodig?**

Voordat we starten, moet je ervoor zorgen dat je een paar dingen hebt geïnstalleerd:

1. **Python**: Dit is de programmeertaal die je gaat gebruiken. Je kunt Python downloaden van de officiële [Python website](https://www.python.org/downloads/).
2. **Flask**: Dit is het framework waarmee we de website gaan bouwen. We installeren het later via de terminal.

Zodra je Python hebt geïnstalleerd, ben je klaar om aan de slag te gaan. Laten we nu beginnen!

**Stap 1: Installeren van Flask**

Flask is een extensie voor Python die we moeten installeren. Om dat te doen, openen we de terminal of command prompt op je computer.

Typ het volgende commando om Flask te installeren:

pip install Flask

Druk op Enter. Dit zal Flask downloaden en installeren. Wanneer het installeren klaar is, kun je verdergaan naar de volgende stap.

**Stap 2: Maak een nieuwe map voor je project**

Het is altijd een goed idee om je werk netjes te organiseren. Maak een nieuwe map op je computer voor je project. Noem de map bijvoorbeeld mijn\_flask\_website. Dit maakt het makkelijker om al je bestanden te vinden.

Binnen de map maken we een Python bestand waar we onze code gaan schrijven. Open een teksteditor (zoals Visual Studio Code, Notepad++ of een andere editor naar keuze) en maak een nieuw bestand met de naam app.py.

**Stap 3: Schrijf je eerste Flask-app**

In het nieuwe bestand app.py gaan we de basis van de website schrijven. Dit doen we door Flask te importeren en een eenvoudige route in te stellen.

**Wat is een route?**  
In webontwikkeling is een route een pad of URL die een gebruiker bezoekt op je website. Wanneer een gebruiker bijvoorbeeld naar http://localhost:5000 gaat, zal de website iets teruggeven, zoals een pagina met tekst.

Laten we nu de volgende code schrijven in app.py:

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

return 'Welkom op mijn eerste Flask-website!'

app.run(debug=True)

**Uitleg van de code:**

* **from flask import Flask**: Hiermee importeren we het Flask-framework in ons project.
* **app = Flask(\_\_name\_\_)**: Dit maakt een nieuwe Flask-applicatie aan zodat je alle commandos ervan kan gebruiken.
* **@app.route('/')**: Dit is de route die wordt geactiveerd wanneer iemand naar de hoofdpagina van je website navigeert. Het pad '/' betekent de startpagina.
* **def home():**: Dit is de functie die wordt uitgevoerd wanneer iemand naar de hoofdpagina navigeert. In dit geval retourneert het een simpel bericht: "Welkom op mijn eerste Flask-website!".
* **app.run(debug=True)**: Dit start de Flask-server. Hij wordt ook gestart in debug mode, zodat je snel fouten kunt oplossen (Als die er zijn).

**Stap 4: De website staren**

Nu we de basis van de website hebben geschreven, kunnen we de app uitvoeren. Ga terug naar de terminal, navigeer naar de map waar je app.py hebt opgeslagen, en typ het volgende commando:

python app.py

Je zou een bericht moeten zien waarin staat dat de server opgestart is, bijvoorbeeld:

\* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

Dit betekent dat je website nu actief is op http://127.0.0.1:5000/. Open een browser en ga naar dat adres. Je zou de tekst "Welkom op mijn eerste Flask-website!" moeten zien.

**Stap 5: Een HTML-pagina toevoegen**

Nu willen we de simpele tekst vervangen door een echte webpagina. Maak een nieuwe map genaamd templates in je projectmap. Flask zoekt automatisch naar HTML-bestanden in deze map en hierdoor kan je deze gebruiken voor je website.

Maak een bestand genaamd index.html in de templates map met de volgende code:

<!DOCTYPE html>

<html lang="nl">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Mijn Flask Website</title>

</head>

<body>

<h1>Welkom op mijn Flask Website!</h1>

<p>Dit is de startpagina van mijn eerste Flask-webapplicatie.</p>

</body>

</html>

Nu moeten we de Python code aanpassen om deze HTML pagina te laten zien in plaats van de tekst.

Wijzig de home functie in je app.py bestand als volgt:

from flask import Flask, render\_template

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

return render\_template('index.html')

app.run(debug=True)

De verandering is klein, maar belangrijk: we gebruiken nu de functie render\_template() om de HTML pagina te tonen. We geven ook aan welk bestand we willen gebruiken voor deze route, in dit geval is dat het bestand index.html.

**Stap 6: De website vernieuwen**

Sla je bestanden op en ga terug naar je terminal. Stop de server door CTRL+C in te drukken en start de server opnieuw:

python app.py

Als je nu je website vernieuwt, zou je de HTML pagina moeten zien die je hebt gemaakt. De tekst "Welkom op mijn Flask Website!" en de paragraaf "Dit is de startpagina van mijn eerste Flask-webapplicatie." zouden zichtbaar moeten zijn.

**Stap 7: Meer pagina's toevoegen**

Nu je een werkende website hebt, kun je gemakkelijk meer pagina's toevoegen. Maak bijvoorbeeld een nieuwe HTML pagina genaamd over.html en voeg een nieuwe route toe in app.py:

@app.route('/over')

def over():

return render\_template('over.html')

De code van over.html zou iets kunnen zijn zoals:

<!DOCTYPE html>

<html lang="nl">

<head>

<meta charset="UTF-8">

<title>Over ons</title>

</head>

<body>

<h1>Over deze website</h1>

<p>Dit is een demowebsite gemaakt met Flask.</p>

</body>

</html>

Door naar http://127.0.0.1:5000/over te gaan in je browser, kun je de nieuwe pagina bekijken.

**Slot: Wat nu?**

Gefeliciteerd! Je hebt nu je eerste website gemaakt met Flask. Je hebt geleerd hoe je een eenvoudige website maakt, hoe je HTML pagina's maakt en hoe je verschillende routes instelt. Dat is nog maar het begin!

Flask biedt nog veel meer mogelijkheden, zoals het verwerken van formulieren, het gebruiken van een database en het toevoegen van interactieve elementen. Als je verder wilt leren, kun je de [officiële Flask-documentatie](https://flask.palletsprojects.com/en/2.3.x/) bekijken voor meer geavanceerde technieken.

Nu je de basis onder de knie hebt, kun je experimenteren met het toevoegen van meer functies en het verbeteren van je website. Veel succes met je verdere reis in de wereld van webontwikkeling!