

Tutorial Básico de Jinja2

1. Introdução ao Jinja2

Jinja2 é um motor de template para Python que permite separar a lógica da aplicação da apresentação. Ele é frequentemente usado com frameworks web como Flask para gerar HTML dinamicamente.

2. Instalação

Para usar Jinja2 com Flask, você precisa instalar o Flask. Você pode instalar Flask (e Jinja2, que vem junto) usando pip:

```
pip install Flask
```

3. Configuração Básica do Flask com Jinja2

Estrutura de Pastas

Crie uma estrutura de pastas simples para o seu projeto:

```
flask_app/  
  app.py  
  templates/  
    index.html
```

Código do app.py

```
from flask import Flask, render_template  
  
app = Flask(__name__)  
  
@app.route('/')  
def index():  
    return render_template('index.html', titulo='Página Inicial', usuario='Alice')  
  
if __name__ == '__main__':  
    app.run(debug=True)
```

4. Sintaxe Básica do Jinja2

Template index.html

Crie um arquivo index.html dentro da pasta templates:

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <title>{{ titulo }}</title>  
</head>  
<body>  
  <h1>Olá, {{ usuario }}!</h1>  
  <p>Bem-vindo à sua primeira página renderizada com Jinja2 e Flask.</p>  
</body>  
</html>
```

Executando a Aplicação

No terminal, navegue até a pasta do projeto e execute:

```
python app.py
```

Acesse <http://127.0.0.1:5000/> em seu navegador para ver a página renderizada.

5. Estruturas de Controle no Jinja2

Condicionais

Você pode usar estruturas de controle como if e else nos templates Jinja2.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>{{ titulo }}</title>
</head>
<body>
  <h1>Olá, {{ usuario }}!</h1>
  {% if usuario == 'Alice' %}
    <p>Bem-vinda de volta, Alice!</p>
  {% else %}
    <p>Olá, visitante!</p>
  {% endif %}
</body>
</html>
```

Laços de Repetição

Você pode iterar sobre listas ou outros iteráveis.

Atualize app.py

```
@app.route('/usuarios')
def usuarios():
    usuarios = ['Alice', 'João', 'Carlos']
    return render_template('usuarios.html', titulo='Usuários', usuarios=usuarios)
```

Crie usuarios.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>{{ titulo }}</title>
</head>
<body>
  <h1>Lista de Usuários</h1>
  <ul>
    {% for usuario in usuarios %}
      <li>{{ usuario }}</li>
    {% endfor %}
  </ul>
</body>
</html>
```

6. Inclusão de Templates

Você pode incluir um template dentro de outro para reutilizar código.

Atualize index.html para incluir um cabeçalho comum

Crie cabecalho.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>{{ titulo }}</title>
</head>
<body>
  <header>
    <h1>Meu Website</h1>
    <nav>
      <a href="/">Home</a> | <a href="/usuarios">Usuários</a>
    </nav>
  </header>
```

Atualize index.html:

```
{% include 'cabecalho.html' %}
<p>Bem-vindo à sua primeira página renderizada com Jinja2 e Flask.</p>
</body>
</html>
```

7. Extensão de Templates

Você pode criar um layout base e estender esse layout em outros templates.

Crie base.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>{% block title %}Meu Website{% endblock %}</title>
</head>
<body>
  <header>
    <h1>Meu Website</h1>
    <nav>
      <a href="/">Home</a> | <a href="/usuarios">Usuários</a>
    </nav>
  </header>
  <main>
    {% block content %}{% endblock %}
  </main>
</body>
</html>
```

Atualize index.html para estender base.html:

```
{% extends 'base.html' %}

{% block title %}Página Inicial{% endblock %}

{% block content %}
<p>Bem-vindo à sua primeira página renderizada com Jinja2 e Flask.</p>
{% endblock %}
```

Atualize usuarios.html para estender base.html:

```
{% extends 'base.html' %}

{% block title %}Usuários{% endblock %}

{% block content %}
<h1>Lista de Usuários</h1>
<ul>
    {% for usuario in usuarios %}
        <li>{{ usuario }}</li>
    {% endfor %}
</ul>
{% endblock %}
```

8. Filtros

Filtros permitem modificar a saída das variáveis.

Atualize app.py

```
@app.route('/saudacao/<nome>')
def saudacao(nome):
    return render_template('saudacao.html', nome=nome)
```

Crie saudacao.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Saudação</title>
</head>
<body>
    <h1>Olá, {{ nome | capitalize }}!</h1>
</body>
</html>
```

9. Variáveis Globais

Você pode passar variáveis globais para todos os templates.

Atualize app.py

```
@app.context_processor
def inject_global_vars():
    return dict(empresa="Minha Empresa")

@app.route('/sobre')
def sobre():
    return render_template('sobre.html')
```

Crie sobre.html

```
{% extends 'base.html' %}

{% block title %}Sobre{% endblock %}

{% block content %}
<h1>Sobre Nós</h1>
<p>Bem-vindo à página sobre a {{ empresa }}.</p>
{% endblock %}
```