

Mãos na massa: Mostrando jogos do servidor

Chegou o momento de implementar o que vimos nos vídeos!

Clicando no link a seguir, você consegue fazer o [download dos arquivos da Aula 01 \(https://caelum-online-public.s3.amazonaws.com/694-flask-rotas-templates-autenticacao/01/aula1.zip\)](https://caelum-online-public.s3.amazonaws.com/694-flask-rotas-templates-autenticacao/01/aula1.zip)!

1) Abra seu arquivo **lista.html** no PyCharm.

2) Utilize a diretiva do flask para usar um conteúdo dinâmico e modifique o título **Jogoteca** conforme abaixo:

```
<h1>{{ titulo }}</h1>
```

3) Abra o arquivo **jogoteca** e passe mais um parâmetro para o **render_template**, que será o conteúdo que será utilizado no **HTML**, veja abaixo:

```
from flask import Flask, render_template

app = Flask(__name__)

@app.route('/inicio')
def ola():
    return render_template('lista.html', titulo='Jogos')

app.run()
```

4) Rode a aplicação e veja que agora o título é dinâmico!

Agora vamos deixar mais conteúdo dinâmico...

5) Dentro de **ola** crie uma lista de jogos, veja abaixo:

```
from flask import Flask, render_template

app = Flask(__name__)

@app.route('/inicio')
def ola():
    lista = ['Tetris', 'Super Mario', 'Pokemon Gold']
    return render_template('lista.html', titulo='Jogos')

app.run()
```

6) Agora passe essa lista como parâmetro do **render_template**

```
from flask import Flask, render_template

app = Flask(__name__)

@app.route('/inicio')
def ola():
    lista = ['Tetris', 'Super Mario', 'Pokemon Gold']
    return render_template('lista.html', titulo='Jogos', jogos=lista)

app.run()
```

7) Volte para seu **HTML** e delete as tags `<tr>`, deixando apenas uma e modifique para pegar o nome do jogo dinamicamente, utilizando o jinja para escrever Python em HTML:

```
<tbody>
{% for jogo in jogos %}
    <tr>
        <td>{{ jogo }}</td>
    </tr>
{% endfor %}
</tbody>
```

Obs: Repare que a diretiva utilizada é diferente!

8) Reinicie sua aplicação e verifique se tudo correu como esperado.

Neste momento temos uma lista exibida. Abaixo vamos fazer aparecer mais informações do Jogo com um objeto.

9) Crie uma classe Jogo no arquivo **jogoteca**, veja abaixo:

```
from flask import Flask, render_template

app = Flask(__name__)

class Jogo:
    def __init__(self, nome, categoria, console):
        self.nome = nome
        self.categoria = categoria
        self.console = console

@app.route('/inicio')
def ola():
    lista = ['Tetris', 'Super Mario', 'Pokemon Gold']
    return render_template('lista.html', titulo='Jogos', jogos=lista)

app.run()
```

10) Crie os objetos dentro de **ola** e modifique a variável lista para armazenar os dois objetos criados:

```
from flask import Flask, render_template
```

```
app = Flask(__name__)

class Jogo:
    def __init__(self, nome, categoria, console):
        self.nome = nome
        self.categoria = categoria
        self.console = console

@app.route('/inicio')
def ola():
    jogo1 = Jogo('Super Mario', 'Ação', 'SNES')
    jogo2 = Jogo('Pokemon Gold', 'RPG', 'GBA')
    lista = [jogo1, jogo2]
    return render_template('lista.html', titulo='Jogos', jogos=lista)

app.run()
```

11) Teste sua aplicação e repare que será exibida uma mensagem de erro.

Bom, faltou editar o HTML que mostra a lista. Vamos fazer isso!

12) Abra seu arquivo **HTML** e modifique conforme abaixo para obter a saída desejada.

```
<thead>
  <tr>
    <th>Nome</th>
    <th>Categoria</th>
    <th>Console</th>
  </tr>
</thead>
<tbody>
{% for jogo in jogos %}
  <tr>
    <td>{{ jogo.nome }}</td>
    <td>{{ jogo.categoria }}</td>
    <td>{{ jogo.console }}</td>
  </tr>
{% endfor %}
</tbody>
```

Observação: Repare que, dessa vez, acessamos o conteúdo de nome, categoria e console do objeto em questão.

13) Dê um refresh na sua aplicação e veja a mudança, teste adicionando mais jogos.