

08

Mão na massa: Adicionando jogos

Chegou a hora de colocar a mão na massa!

Clicando no link a seguir, você consegue fazer o [download dos arquivos da Aula 02 \(https://caelum-online-public.s3.amazonaws.com/694-flask-rotas-templates-autenticacao/02/aula2.zip\)](https://caelum-online-public.s3.amazonaws.com/694-flask-rotas-templates-autenticacao/02/aula2.zip).

1) Crie um novo arquivo **HTML**, no curso utilizamos o nome **novo.html** e cole o código abaixo:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Jogoteca</title>
</head>
<body>
    <div class="container">
        <div class="page-header">
            <h1>{{ titulo }}</h1>
        </div>
        <form>
            <fieldset>
                <div class="form-group">
                    <label for="nome">Nome</label>
                    <input type="text" id="nome" name="nome" class="form-control">
                </div>
                <div class="form-group">
                    <label for="categoria">Categoria</label>
                    <input type="text" id="categoria" name="categoria" class="form-control">
                </div>
                <div class="form-group">
                    <label for="console">Console</label>
                    <input type="text" id="console" name="console" class="form-control">
                </div>
                <button type="submit" class="btn btn-primary btn-salvar">Salvar</button>
            </fieldset>
        </form>
    </div>
</body>
</html>
```

2) Crie uma nova rota para exibir esse **HTML**

```
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.route('/novo')
def novo():
    return render_template('novo.html', titulo='Novo Jogo')
app.run()

```

3) Reinicie sua aplicação e teste a rota /novo

Aqui já dá pra ver o formulário!

4) Modifique o **HTML** da página **novo.html**, mais especificamente a tag **form**, mudando suas propriedades **action** e **method** olhe abaixo:

```

inicio do html...

<form action="/criar" method="post">
    <fieldset>
        <div class="form-group">
            <label for="nome">Nome</label>
            <input type="text" id="nome" name="nome" class="form-control">
        </div>
        <div class="form-group">
            <label for="categoria">Categoria</label>
            <input type="text" id="categoria" name="categoria" class="form-control">
        </div>
        <div class="form-group">
            <label for="console">Console</label>
            <input type="text" id="console" name="console" class="form-control">
        </div>
        <button type="submit" class="btn btn-primary btn-salvar">Salvar</button>
    </fieldset>
</form>

final do html...

```

5) Adicione uma rota no código e importe o helper do flask **request** (veja primeira linha), conforme abaixo:

```

from flask import Flask, render_template, request

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.route('/novo')
def novo():
    return render_template('novo.html', titulo='Novo Jogo')

@app.route('/criar')
def criar():
    nome = request.form['nome']
    categoria = request.form['categoria']
    console = request.form['console']
    jogo = Jogo(nome, categoria, console)

app.run()

```

6) Retire a criação da lista de dentro do escopo da função `ola`, veja o código abaixo:

```

from flask import Flask, render_template, request

app = Flask(__name__)

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

jogo1 = Jogo('Super Mario', 'Ação', 'SNES')
jogo2 = Jogo('Pokemon Gold', 'RPG', 'GBA')
lista = [jogo1, jogo2]

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

@app.route('/novo')
def novo():
    return render_template('novo.html', titulo='Novo Jogo')

@app.route('/criar')
def criar():
    nome = request.form['nome']
    categoria = request.form['categoria']
    console = request.form['console']
    jogo = Jogo(nome, categoria, console)

```

```
app.run()
```

7) Utilize essa lista global para adicionar o jogo que veio através do `request`, olhe abaixo a função `criar`

```
from flask import Flask, render_template, request

app = Flask(__name__)

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

jogo1 = Jogo('Super Mario', 'Ação', 'SNES')
jogo2 = Jogo('Pokemon Gold', 'RPG', 'GBA')
lista = [jogo1, jogo2]

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

@app.route('/novo')
def novo():
    return render_template('novo.html', titulo='Novo Jogo')

@app.route('/criar')
def criar():
    nome = request.form['nome']
    categoria = request.form['categoria']
    console = request.form['console']
    jogo = Jogo(nome, categoria, console)
    lista.append(jogo)
    return render_template('lista.html', titulo='Jogo', jogos=lista)

app.run()
```

8) Reinicie e teste sua aplicação, repare que teremos um erro, que vamos resolver logo em seguida.

9) Modifique a rota `/criar` para receber o método `post` (padrão do flask é `get`), veja o código abaixo:

```
from flask import Flask, render_template, request

app = Flask(__name__)

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

jogo1 = Jogo('Super Mario', 'Ação', 'SNES')
```

```
jogo2 = Jogo('Pokemon Gold', 'RPG', 'GBA')
lista = [jogo1, jogo2]

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

@app.route('/novo')
def novo():
    return render_template('novo.html', titulo='Novo Jogo')

@app.route('/criar', methods=['POST'])
def criar():
    nome = request.form['nome']
    categoria = request.form['categoria']
    console = request.form['console']
    jogo = Jogo(nome, categoria, console)
    lista.append(jogo)
    return render_template('lista.html', titulo='Jogo', jogos=lista)

app.run()
```

10) Reinicie sua app e veja que agora está tudo ok!

11) Resolva o problema de reiniciar a aplicação toda hora modificando a última linha do código, renomeie a função **ola** para **index** e altere sua rota para **/**, veja abaixo:

```
from flask import Flask, render_template, request

app = Flask(__name__)

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

jogo1 = Jogo('Super Mario', 'Ação', 'SNES')
jogo2 = Jogo('Pokemon Gold', 'RPG', 'GBA')
lista = [jogo1, jogo2]

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

@app.route('/novo')
def novo():
    return render_template('novo.html', titulo='Novo Jogo')

@app.route('/criar', methods=['POST'])
def criar():
    nome = request.form['nome']
    categoria = request.form['categoria']
    console = request.form['console']
    jogo = Jogo(nome, categoria, console)
    lista.append(jogo)
```

```
return render_template('lista.html', titulo='Jogo', jogos=list)  
app.run(debug=True)
```

Observa o: Repare que se voc  fez essas altera es (fun o index e rota /) ap s mudar a \'ltima linha, e reiniciar a aplicac o, agora o flask ir  restartar sua app automaticamente toda vez que voc  salvar alguma altera o utilizando CMD + S ou CTRL + S no Pycharm.