



Alex Silva de Sousa <alex.ssousa@aluno.faculdadeimpacta.com.br>

AC3 - Desenvolvimento Web

1 mensagem

Formulários Google <forms-receipts-noreply@google.com>
Para: alex.ssousa@aluno.faculdadeimpacta.com.br

19 de maio de 2024 às 11:00

Agradecemos o preenchimento de [AC3 - Desenvolvimento Web](#)

Veja as respostas enviadas.

AC3 - Desenvolvimento Web

As questões contidas nessa atividade estão relacionadas aos conteúdos das partes 09, 10, 11 e 12.

Seu e-mail (alex.ssousa@aluno.faculdadeimpacta.com.br) foi registrado quando você enviou este formulário.

O padrão MVC é utilizado em muitos projetos devido à arquitetura que possui, o que possibilita a divisão do projeto em camadas muito definidas: o Model, o Controller e a View. A respeito desse padrão, assinale a alternativa correta:

*

- ☐ O Model é a camada que recebe requisições dos clientes e decide o tipo de visualização mais adequado
- ☐ A View contém a lógica de negócio, e pode acessar, por exemplo, o banco de dados
- ☐ O papel do Controller é gerar a visualização dos dados que será encaminhada ao cliente
- ☒ O Controller é a ponte entre o Model e a View. Ele recebe a requisição do usuário, é responsável pelo fluxo de dados e de trabalho da aplicação, além de decidir o caminho da resposta do processamento
- ☐ A View processa a rota que irá responder a requisição do usuário. Para isso, ela deve receber essa requisição diretamente

Dada a aplicação em Flask a seguir, qual será a resposta do servidor ao acessar a rota /principal?

```
from flask import Flask
```

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

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

```
def abertura():
```

```
    return "abcdef"
```

```
def principal():
```

```
    return "123456"
```

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

```
def principal_abertura():
```

```
    return "abcdef123456"
```

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

```
    app.run()
```

*

- ☒ abcdef
- ☐ 123456
- ☐ 123456abcdef
- ☐ abcdef123456
- ☐ Não há nenhum controle configurado para responder à rota /principal

Imagine a seguinte configuração de rotas no Flask. Marque a única URL que **NÃO** redireciona para nenhum Controller:

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

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

@app.route('/cursos/<sigla>')
def curso(sigla):
    return render_template('curso.html', nome=sigla)

@app.route('/cursos/<sigla>/disc')
def disciplinas(sigla):
    return render_template('disciplinas.html', disc=sigla)

@app.route('/cursos/<sigla>/disc/<int:id>')
def cursos_com_sigla_id(sigla, id):
    return render_template('cursos2.html', nome_curso=sigla, cod_curso=id)
```

*

- ☐ <http://localhost:5000/>
- ☐ <http://localhost:5000/cursos/ads/disc>
- ☐ <http://localhost:5000/cursos>
- ☐ <http://localhost:5000/cursos/ads/disc/4>
- ☒ <http://localhost:5000/cursos/ads/disc/TecWeb>

Considere o controller do código Python a seguir. Qual das alternativas substitui de forma correta os trechos destacados em números romanos, para que o template mostre todos os nomes dos alunos enviados pelo controller, em uma lista definida pela tag ?

Trecho de código: app.py

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

def alunos():

    lista_alunos = [

        "Ana",

        "João",

        "Maria",

        "Pedro"

    ]

    return render_template('alunos.html', alunos=lista_alunos)
```

Trecho de código: template alunos.html

```
<ul class="listagem-alunos">
```

```
__ (I) __
```

```
    <li>__ (II) __</li>
```

```
__ (III) __
```

```
</ul>
```

*

- ☐ (I) { for item in alunos }; (II) { item }; (III) { end }
- ☐ (I) { for item in lista }; (II) {% aluno %}; (III) { end }
- ☐ (I) {{ for aluno in alunos }}; (II) {% aluno %}; (III) {{ end }}
- ☐ (I) {% for item in lista_alunos %}; (II) {{ aluno }}; (III) {{ endfor }}
- ☒ (I) {% for aluno in alunos %}; (II) {{ aluno }}; (III) {% endfor %}

Qual a template tag do Flask usamos para herdar a estrutura de um outro template, nesse caso, do template "base.html"?

*

- ☒ {% extends "base.html" %}
- ☐ {% inherits "base.html" %}
- ☐ {% implements "base.html" %}
- ☐ {{ base.html }}
- ☐ {% base.html %}

Considerando o fluxo de autenticação de um usuário, qual dos seguintes passos NÃO seria necessário no processo:

*

- ☒ Verificação e conversão do nome de usuário para letras maiúsculas através do evento onchange no atributo <input>
- ☐ Envio do usuário e senha ao servidor
- ☐ Verificação da existência do usuário e da senha enviados
- ☐ Criação do objeto de sessão do usuário no servidor
- ☐ Definição do cookie de sessão no navegador e redirecionamento para a área logada

Supondo que um formulário envie dados para a rota /cadastro no servidor usando o método POST, como podemos recuperar, dentro de um controle do Flask, a informação associada ao name cpf enviada pelo formulário?

*

- ☐ request.cpf
- ☐ session['cpf']
- ☐ request.get('cpf')
- ☒ request.form.get('cpf')
- ☐ request.args.get('cpf')

O AJAX é uma tecnologia usada para desenvolvimento de sites interativos. A abordagem de desenvolvimento tradicional tem semelhanças e diferenças em relação ao AJAX. Uma característica exclusiva do AJAX em relação à abordagem tradicional é que:

*

- ☐ Executa somente requisições síncronas através do protocolo HTTP
- ☐ Usa Javascript como linguagem para desenvolver código no lado do servidor
- ☐ Usa HTML/CSS como linguagem de templates para definir o aspecto visual da página
- ☒ Permite recuperação assíncrona de dados usando XMLHttpRequest ou fetch API
- ☐ Representa os objetos no lado cliente com DOM

Sobre o objeto XMLHttpRequest, assinale a alternativa verdadeira:

*

- ☐ Permite apenas requisitar URLs que retornam informações no formato XML de maneira assíncrona, não sendo aceitos outros formatos de dados
- ☒ O atributo responseText contém a resposta enviada pelo servidor, após uma requisição
- ☐ O evento readyState é disparado toda vez que ocorre uma mudança no atributo responseText
- ☐ A propriedade status será igual a 405 quando a requisição terminar de forma bem sucedida
- ☐ Permite apenas requisições com o método GET

O JSON é um formato para representação de dados leve, comumente usado para transferência de informações entre o servidor e o cliente. Sobre esse formato, assinale a alternativa correta:

*

- ☒ É um formato similar ao dicionário do Python, onde os dados são armazenados no formato chave/valor. Todas as chaves devem ser strings, e os valores podem ser de vários tipos de dados (strings, números, booleanos, null, vetores, e inclusive outros objetos JSON)
- ☐ No formato JSON, os dados devem sempre estar entre colchetes []
- ☐ No Flask, podemos criar uma string no formato JSON através da função render_template()

- ☐ É um formato para armazenar um conjunto de dados em formato binário. As chaves podem ser de qualquer tipo, e não é possível definir um JSON com apenas uma chave e um valor
- ☐ É uma linguagem similar ao XML, que utiliza marcações com os símbolos <> para definir chaves e valores

Crie seu próprio formulário do Google.

[Denunciar abuso](#)