



Prezado candidato,

Para o desenvolvimento dos algoritmos propostos abaixo, não serão permitidos o uso de bibliotecas de terceiros.

As questões 1 e 2 devem ser desenvolvidas em Javascript + JQuery. É permitido usar bibliotecas de terceiros.

A questão 3 deve ser desenvolvida com HTML e CSS. É permitido usar bibliotecas de terceiros.

Ao finalizar o desafio, os códigos devem ser commitados no GitHub/BitBucket.

Favor enviar o endereço público dos mesmos para cristiane.moraes@duotecnologia.com.br

1) A função adicionaDiv deveria apenas adicionar uma child div para cada div existente em um documento. Toda nova div será decorada pela função decoraDiv.
Por exemplo:

Estado atual do documento

```
<div id="1">Div 1  
  
    <div id="2">Div 2</div>  
  
</div>
```

O resultado esperado ao executar a função adicionarDiv é:

```
<div id="a">Div A  
  
    <div id="b">Div B  
  
        <div>Nova Div</div>  
  
    </div>  
  
    <div>Nova Div</div>  
  
</div>
```

O problema é que o código atual está entrando em looping eterno.

Segue trecho do código com bug. Corrija o erro:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Teste Projuris</title>

<script>

    function adicionaDiv() {

        var divList = document.getElementsByTagName("div");

        for (var i = 0; i < divList.length; i++) {

            var novaDiv = document.createElement("div");

            decorarDiv(novaDiv);

            divList[i].appendChild(novaDiv);

        }

    }

    function decorarDiv(div) {

        div.textContent = 'Nova Div';

    }

</script>

</head>

<body>

    <div id="1">Div 1

        <div id="2">Div 2</div>

    </div>

</body>

</html>
```

2) Existe uma lista que distingue animais por região e criticidade de ameaça de extinção.

```
<div>
  <ul regioao="Norte">
    <li data-animal="condor negro">Ameaçado de Extinção</li>
    <li data-animal="sapo cururu">Abundante</li>
  </ul>
  <ul regioao="sul">
    <li data-animal="mamute">Extinto</li>
  </ul>
</div>
```

Dada esta lista, crie uma função que recebe a região e o nome do animal para retornar a criticidade de ameaça de extinção do mesmo.

Exemplo: se a função receber ('Norte', 'condor negro'), deve retornar 'Ameaçado de Extinção'.

```
function situacaoAnimal(regiao, animal) {
  // Implemente a função aqui
}
```

3) O Designer do produto desenhou uma tela de cadastro. Crie um HTML para codificar esta especificação.
Segue mockup da tela:

Adicionar Pessoa

Cancelar Cadastro

TIPO ★

Pessoa Física

TRATAMENTO

Tratamento

NOME ★

Informe o nome completo

GÊNERO

Masculino

CPF

000.000.000-00

RG

Informe o RG

NASCIMENTO

Informe a data

ADVOGADO? ★

Não

QUALIFICAÇÃO

Insira mais informações sobre a pessoa

PALAVRAS-CHAVE

Informe as palavras-chave desejadas

E-MAIL

Pessoal

Informe o endereço e-mail

+

PRINCIPAL

TELEFONE

Pessoal

(7) 98833-9999

+

PRINCIPAL

Comercial

(7) 99337-2222

+

PRINCIPAL

ENDEREÇO

Pessoal

Informe o endereço completo

+

PRINCIPAL

Ocultar campos

Salvar

Cancelar

O campo “Tipo” permite os valores “Pessoa física” e “Pessoa Jurídica”.

O campo “Gênero” permite os valores “Masculino” e “Feminino”.

O campo “Advogado” permite os valores “Sim” e “Não”.

Os campos de texto como CPF, Telefone, RG e Nascimento possuem máscaras de input.

Não é necessário implementar ações nos clicks de nenhum botão.

Não é necessário submeter o formulário e nem salvar os dados em banco de dados.

Basta criar a tela nos moldes e estilos previstos pelo desenho.