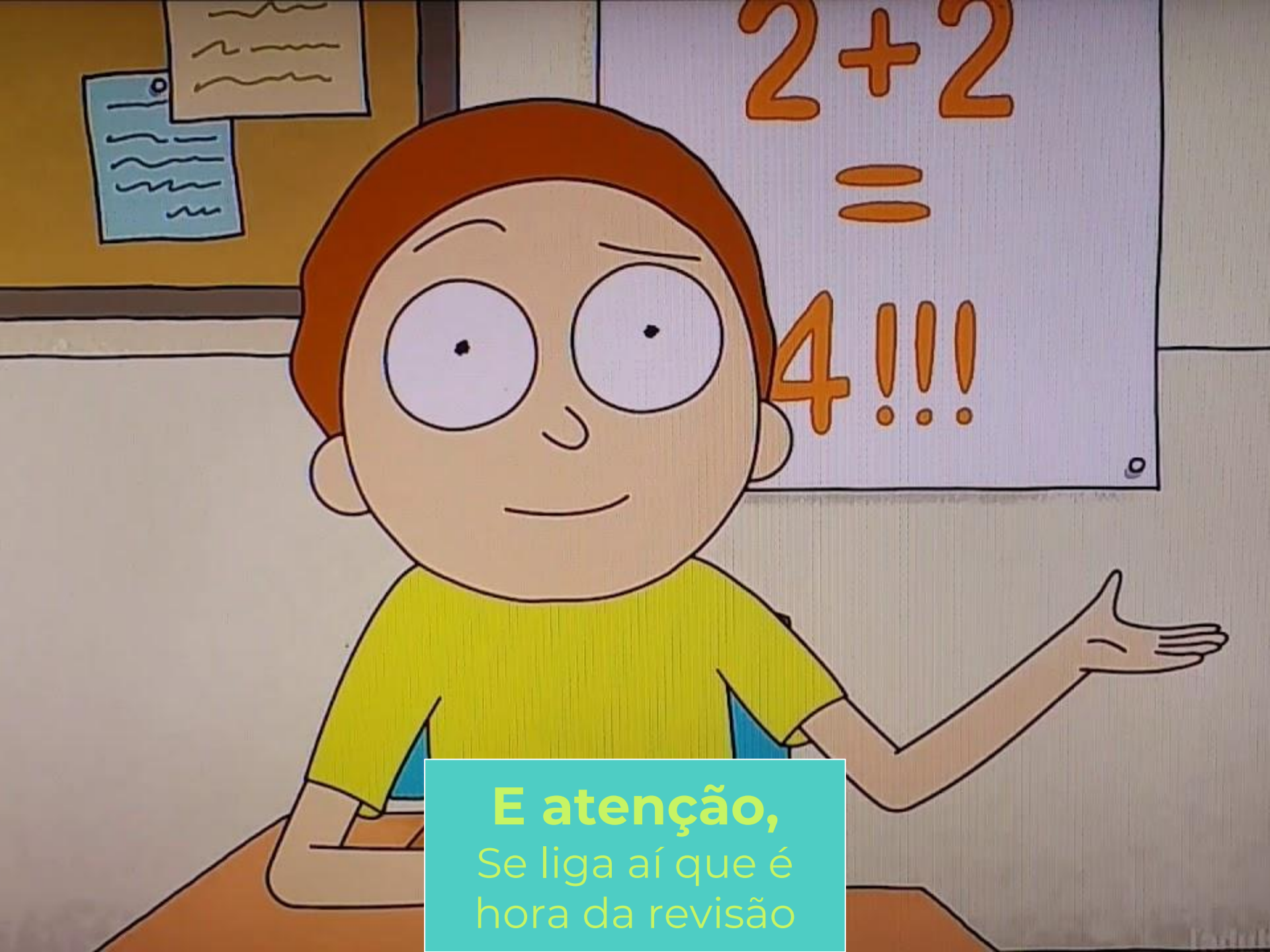


Olá!



E atenção,
Se liga aí que é
hora da revisão

Lembrando...

1. Tabelas
 - a. Table, tr, td, th, thead, tbody, tfoot
2. Iframes
3. Âncoras (marcadores)
4. Hipermissãoia
 - a. audio, video, plug-ins (object e embed)

HTML

Aula 03

1. Elementos

Hoje iremos
aprender mais
sobre
formulários
HTML

Formulário

O elemento formulário, representado pela tag `<form>`, é responsável por coletar dados de entrada do usuário.



Setup your account

Make sure to use a valid email address, you'll need to verify it before you can send any campaigns.

Name

Email Address

Username

Password

Company

Country

Timezone

Elementos de formulário



Elementos de formulário são tags específicas para captar diferentes tipos de entradas do usuário, como datas, senhas, email, clique de botão, escolha de dados em uma lista, etc.

Elementos de um formulário são encapsulados pela tag `<form>`.

Input



O elemento `<input>` é o mais importante de um formulário, pois com ele diversos tipos de dados podem ser coletados, a depender do que for colocado em seu atributo "type".

Input - tipo "text"



O atributo "type" de valor "text" indica que o input receberá um tipo textual simples do usuário.

Input - tipo "radio"



O atributo "type" de valor "radio" indica que o input receberá dados a partir da escolha de um valor de uma lista de itens pré-determinados.

Input - tipo "submit"



O atributo "type" de valor "submit" criará um botão, que será responsável pelo envio dos dados preenchidos no formulário para o manipulador do form.

O manipulador do form é definido pelo atributo "action", na tag de abertura do form. Ele receberá os dados e poderá os processar devidamente.

Method

O formulário possui um atributo "method", que especifica o método HTTP (GET ou POST) para ser utilizado na apresentação de dados do formulário, ao ser enviado.

```
<form action="handler.html" method="get">  
<form action="handler.html" method="post">
```

GET x POST



GET: quando não há sigilo nos dados, pois os mesmos serão enviados via URL (ficarão visíveis ao usuário).

POST: quando há dados sigilosos, pois serão enviados no corpo da requisição HTTP (não visível ao usuário).

Atributo "name"



O atributo "name" está presente em todos os elementos input.

Eles servem para nomear os campos em uma requisição HTTP (GET ou POST).

Todos os inputs deverão ter este atributo, pois apenas serão enviados na submissão se houver este atributo.

Atributo "value"



O atributo "value" indica o valor pré-determinado de um campo (como por exemplo, elementos "radio").

Caso seja usado em um elemento textual, esse atributo fará com que tal input seja carregado com o valor do "value" como padrão.



Exercício 1

1. Em <https://github.com/drcabral/Design-Culture-HTML-CSS-JS> baixe o arquivo "formHandler.html" que se encontra em Aula 03/Exercicio 1 e o coloque na mesma pasta do arquivo deste exercício.
2. Crie um novo documento HTML que conterá um formulário com:
 - a. Um campo para nome (name = "firstName")
 - b. Um campo para sobrenome (name = "lastName")
 - c. Um radio para escolha de sexo (name = "gender")
 - d. Um botão para submissão
3. O formulário deverá ter uma action, que apontará para o arquivo baixado do github no passo (1).
4. Utilize o método GET

Resultado Esperado

Formulário

Nome

Sobrenome

Sexo

- ☒ Masculino
☐ Feminino
☐ Outro

Enviar

Resultado Esperado

Olá Diogo Cabral do gênero masculino!

Formulário



Existem containers para organização de conteúdo de um formulário. Podemos utilizar o elemento `<fieldset>` para agrupar dados relacionados em um formulário, e o elemento `<legend>` para dar uma legenda ao que esse conjunto de dados representa.

```
<form action="/handler.html">
  <fieldset>
    <legend>Informação Pessoal</legend>
    <span>First name</span>
    <br>
    <input type="text" name="firstname">
    <br>
    <span>Last name:</span>
    <br>
    <input type="text" name="lastname">
  </fieldset>
  <br>
  <input type="submit" value="Submit">
</form>
```

Input - tipo "password"



O atributo "type" de valor "password" indica que o input receberá dados de senha, e por isso ele esconderá os caracteres digitados.

Input - tipo "reset"



O atributo "type" de valor "reset" fará com que o input se transforme em um botão que, ao clicado, resetará todos os valores dos campos do formulário.

Input - tipo "button"



O atributo "type" de valor "button" fará com que o input se transforme em um botão que poderá receber eventos de click (onclick).

```
<input type="button" onclick="alert('Hello World!')" value="Click Me!">
```

Input - tipo "checkbox"

O atributo "type" de valor "checkbox" indica que o input será uma caixa de seleção. Caso mais de uma seja criada, elas serão independentes, podendo ter zero ou mais de itens selecionados.

```
<form>
  <input type="checkbox" name="vehicle1" value="Bike"> I have a bike
  <br>
  <input type="checkbox" name="vehicle2" value="Car"> I have a car
</form>
```


Select



O elemento <select> define um seletor que criará uma lista drop-down.

Os itens desse elemento devem ser definidos com a tag <option>.

Por padrão, o primeiro elemento é o selecionado ao abrir a página. Caso queira definir um diferente do primeiro, tal opção deverá vir com o atributo "selected".

```
<select name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat" selected>Fiat</option>
  <option value="audi">Audi</option>
</select>
```

Textarea



O elemento <textarea> define uma área multi-linhas para a inserção de texto do usuário.

Ele pode receber os atributos "rows" (para indicar a quantidade de linhas padrão) e "cols" (para indicar a quantidade de colunas padrão).

```
<textarea name="message" rows="10" cols="30">
```

Button



O elemento `<button>` define um botão que aceita a ação de click (chamado "onclick").

```
<button type="button" onclick="alert('Oi!')">Clique aqui!</button>
```

Datalist



O elemento `<datalist>` define uma lista pré-definida de opções para um input do tipo texto.

Para relacionar o input ao datalist, deverá ser adicionado ao input o atributo "list", que será de valor igual ao atributo "id" do datalist.

```
<form action="handler.html">
  <input list="browsers">
    <datalist id="browsers">
      <option value="Internet Explorer">
      <option value="Firefox">
      <option value="Chrome">
      <option value="Opera">
      <option value="Safari">
    </datalist>
  </form>
```

Entradas HTML5



O HTML5 possui diversos outros novos tipos de entrada pré-definidas, dentre eles:

- color
- date
- datetime-local
- email
- month
- number
- range
- search
- time
- url
- week

Input - tipo "color"



O atributo "type" de valor "color" serve para tipos de entrada que esperam uma cor. A depender do browser usado, poderá aparecer uma paleta de seleção de cores.

Input - tipo "color"



O atributo "type" de valor "color" serve para tipos de entrada que esperam uma cor. A depender do browser usado, poderá aparecer uma paleta de seleção de cores.

Input - tipo "date"



O atributo "type" de valor "date" serve para tipos de entrada que esperam uma data. A depender do browser usado, poderá aparecer um campo de seleção de datas.

Podem ser adicionados ao elemento os atributos "max" e "min", que permitirá que uma data máxima possível de escolha seja adicionada (exemplo: "2000-01-02")

Input - tipo "datetime-local"



O atributo "type" de valor "datetime-local" serve para tipos de entrada que esperam uma data, mas sem fuso horário. A depender do browser usado, poderá aparecer um campo de seleção de datas.

Input - tipo "email"



O atributo "type" de valor "email" serve para tipos de entrada que esperam uma email.

Alguns browsers salvam o e-mail a partir deste tipo para dar como sugestão em futuras inserções. Além disso, alguns dispositivos móveis alteram seu teclado para facilitar o acesso ao "@" e ao ".com"

Input - tipo "month"



O atributo "type" de valor "month" serve para tipos de entrada que esperam uma data, mas especificamente selecionando apenas o mês do ano. Dependendo do browser utilizado, um campo de seleção de mês poderá aparecer.

Input - tipo "number"



O atributo "type" de valor "number" serve para tipos de entrada que esperam um valor numérico.

Também podem ser definidos atributos "max" e "min" para este elemento com este tipo, dando um range de números aceitos.

Input - tipo "range"



O atributo "type" de valor "range" serve para tipos de entrada que esperam um valor numérico, que será selecionado a partir de um controle deslizante.

Também podem ser definidos atributos "max" e "min" para este elemento com este tipo, dando um range de números aceitos.

Input - tipo "search"



O atributo "type" de valor "search" funciona como um campo textual comum, mas é definido como um campo de pesquisa.

Input - tipo "time"



O atributo "type" de valor "time" funciona como um campo que permite ao usuário a escolha de tempo (sem fuso horário).

Dependendo no browser escolhido para uso, poderá aparecer um campo de seleção de tempo.

Input - tipo "url"



O atributo "type" de valor "url" funciona como um campo que espera receber uma URL.

Dependendo no browser escolhido para uso, a URL inserida pode ser automaticamente validada quando submetida.

Input - tipo "week"



O atributo "type" de valor "week" funciona como um campo que espera receber uma semana e um ano

Dependendo no browser escolhido para uso, aparecerá um campo para escolha da semana e do ano.



Exercício 2

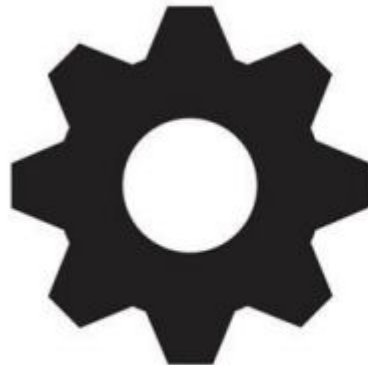
1. Utilizando o documento gerado no Exercício 1, complemente seu formulário utilizando:
 - a. Containers de organização do form
 - b. Diversos dos novos elementos apresentados (livre escolha)
2. Altere o método de envio do form para POST
3. Utilize o mesmo form action utilizado no Exercício 1

Resultado Esperado

Não enxergo mais os parâmetros na URL :(

SVG

SVG (Scalable Vector Graphics) é usado para definir gráficos para a web, e é uma recomendação da W3C.



SVG



PNG

SVG



O elemento <svg> é um recipiente para este tipo de gráfico e possui vários elementos filhos para diversos tipos de desenhos.

```
<svg height="150" width="500">  
  <ellipse cx="240" cy="100" rx="220" ry="30" style="fill:purple" />  
  <ellipse cx="220" cy="70" rx="190" ry="20" style="fill:lime" />  
  <ellipse cx="210" cy="45" rx="170" ry="15" style="fill:yellow" />  
  Sorry, your browser does not support inline SVG.  
</svg>
```

SVG



Lembrando que o elemento `<object>` visto na aula passada também pode ser usado para inserir conteúdo do tipo SVG!



Desafio

1. Utilizando os conhecimentos adquiridos em HTML, iremos construir um site mais complexo
2. O tema é livre, mas deve conter mais de uma página, mesmo que ainda sem estilo, que represente a elaboração de alguma ideia sua
3. Se baseiem em ideias de construção de outros sites para montar sua própria ideia
4. Utilizaremos o site criado agora futuramente, incrementando com CSS e JS
5. Enviem, por e-mail, dúvidas e o resultado do site, para que haja um acompanhamento da atividade
6. Envios até 23/11

Sugestão:

<https://github.com/estevanmaito/apenas-crie-websites>



Dúvidas?



github.com/drcabral



twitter.com/DrCabrales



diogo.cabral.dev@gmail.com