PUC Minas

Janelas modais

Prof. Marcos André S. Kutova

Janela modal

Uma janela modal é uma janela subordinada à janela da aplicação, que se mantém visível até ser dispensada. Até que isso aconteça, a janela principal é desativada.

```
alert("Olá mundo!");
```

JavaScript em uma página *web*

```
<!DOCTYPE html>
<html lang="pt-br">
    <head>
        <title>Olá Mundo</title>
        <meta charset="utf-8" />
        <script>
            function ola() {
                alert( "Olá mundo!" );
            }
        </script>
    </head>
    <body onload="ola()">
            <p>Olá mundo em JavaScript</p>
    </body>
</html>
```

Interação com o usuário

Enquanto o método **`alert()`** apresenta uma informação ao usuário, o método **`prompt()`**, por outro lado, permite que o usuário digite uma informação.

```
let nome = prompt("Qual é o seu nome?");
alert(`Olá ${nome}.`);
```

Valor inicial

É possível que a resposta já venha com um valor inicial, que poderá ser confirmado ou alterado pelo usuário.

```
let nome = prompt("Qual é o seu nome?",
                  "Fulano de Tal");
alert(`Olá ${nome}.`);
```

Confirmação pelo usuário

Por fim, para operações que buscam apenas uma respostas afirmativas ou negativas, o método `confirm()` oferece uma caixa de diálogo baseada em botões.

```
let confirma = confirm("Confirma a operação?");
if (confirma)
   alert("Confirmado!");
else
   alert("Cancelado.");
```

PUC Minas

Eventos da página web

Prof. Marcos André S. Kutova

Eventos

```
<!DOCTYPE html>
<html lang="pt-br">
    <head>
        <title>Olá Mundo</title>
        <meta charset="utf-8" />
        <script>
            function ola() {
                alert( "Olá mundo!" );
            }
        </script>
    </head>
    <body onload="ola()">
            <p>Olá mundo em JavaScript</p>
    </body>
</html>
```

Método associado ao evento

Evento do navegador

Eventos de mouse

```
<!DOCTYPE html>
<html lang="pt-br">
    <head>
        <title>Olá Mundo</title>
        <meta charset="utf-8" />
        <script>
            function ola() {
                alert( "Olá mundo!" );
            }
        </script>
    </head>
    <body>
        <p><button onclick="ola()">Olá</button></p>
    </body>
</html>
```

Evento de interação com o usuário

Principais eventos

Mouse

- `click` – quando o mouse é clicado em um elemento da página
- `mouseover` – quando o mouse entre sobre a área de um elemento
- `mouseout` – quando o mouse deixa a área de um elemento
- `mousemove` – quando o mouse é movimentado
- `contextmenu` – quando o botão direito do mouse é clicado
- `mousedown` – quando o botão do mouse é pressionado
- `mouseup` – quando o botão do mouse é solto

Principais eventos

Formulário

- `submit` – quando um formulário é enviado
- `focus` – quando um campo do formulário é ativado
- `blur` – quando o usuário deixa um campo do formulário
- `change` – quando o valor de um campo é alterado
- `select` – quando é feita uma escolha em uma caixa de seleção

Principais eventos

Teclado

- `keydown` – quando uma tecla é pressionada
- `keyup` – quando a tecla é liberada

Janela

- `load` – quando a página termina de ser carregada
- `unload` – quando o usuário deixa a página ou fecha o navegador
- `rezise` – quando a janela do navegador é redimensionada

Eventos criados com JS

Ao invés de atribuir os eventos no meio do código HTML, podemos especificá-los por meio da própria JavaScript:

```
window.onload = () => {
  bt.onclick = () => alert('Olá mundo!');
}

...

<button id="bt">Olá</button>
```

Uso adequado

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <title>Olá Mundo</title>
    <meta charset="utf-8" />
    <script>
       window.onload = () => {
         bt.onclick = () => alert( "Olá mundo!" );
       }
    </script>
  </head>
  <body>
    <p><button id="bt">Olá</button></p>
  </body>
</html>
```

Mais uma alternativa

Ainda já mais uma forma de associação de uma função a um evento de algum elemento.

```
window.load = () => {
  bt.addEventListener(
      'click',
      () => alert('Olá mundo!')
  );
}

...
```

```
<button id="bt">Olá</button>
```

PUC Minas

Window

Prof. Marcos André S. Kutova

Objetos do navegador

Os navegadores oferecem alguns objetos internos que nos ajudam a melhorar a interatividade com o usuário. Os principais são:

- `window`
    - Esse objeto oferece acesso às informações e recursos da janela do navegador em que a página *web* está sendo apresentada;
- `navigator`
    - Esse objeto nos oferece acesso às informações e recursos do próprio navegador;
- `document`
    - Esse objeto oferece acesso às informações e recursos do da própria página, incluindo seu conteúdo e formatação.

O objeto `window`

O objeto **`window`** nos permite obter informações sobre a janela, mas também nos deixa manipulá-la. Algumas dessas informações são:

- **`window.screen`** – Objeto com várias propriedades da tela
- **`window.history`** – Objeto com referências às últimas páginas acessadas
- **`window.location`** – Objetos com dados do URL da página

O objeto `window`

O objeto **`window`** ainda possui métodos para interação com o usuário:

- **`window.alert()`** – Mostra uma janela modal com uma mensagem com botão OK
- **`window.confirm()`** – Mostra uma janela modal com botões OK e de cancelamento
- **`window.prompt()`** – Mostra uma janela modal com uma pergunta para o usuário
- **`window.print()`** – Abre a janela de impressão do documento
- **`window.setTimeout()`** – Executa uma função depois de um tempo pré-definido
- **`window.setInterval()`** – Executa uma função repetidas vezes em intervalos de tempo pré-definidos

O objeto `window`

O objeto **`window`** é o objeto de mais alto nível. Os outros são parte dele.

```
window.document === document

window.navigator === navigator

window.alert === alert
```

E é por isso que ele não precisa ser escrito de forma explícita o tempo todo.

O objeto `window`

Os elementos com ID no documento também são declarados no objeto `window`.

```
window.onload = () => {
  window.bt.onclick =
                  () => alert('Olá mundo!');
}

...

<button id="bt">Olá</button>
```

O objeto `window`

Mas não precisa ser tudo explícito.

```
window.onload = () => {
  window.bt.onclick =
            () => window.alert('Olá mundo!');
}
```

equivale a

```
onload = () => {
  bt.onclick = () => alert('Olá mundo!');
}
```

PUC Minas

Navigator & Document

Prof. Marcos André S. Kutova

O objeto `navigator`

O objeto **`navigator`** nos ajuda a entender qual é o contexto de navegação do usuário. Por exemplo, podemos obter as seguintes informações:

- **`navigator.userAgent`** – Assinatura exclusiva do navegador
- **`navigator.language`** – Idioma padrão do navegador
- **`navigator.geolocation`** – Coordenadas geográficas do dispositivo
- **`navigator.online`** – Conectividade do dispositivo
- **`navigator.platform`** – Plataforma do usuário

O objeto `document`

O objeto **`document`** nos permite obter informações sobre a página que está sendo apresentada no navegador. A lista dessas informações é bem extensa. Algumas delas são:

- **`document.documentURI`** – URL da página
- **`document.documentElement`** – Referência ao elemento **`<html>`** da página
- **`document.head`** e **`document.body`** – Referências ao cabeçalho e corpo da página
- **`document.links`** e **`document.images`** – Listas de links e de imagens da página

O objeto document

É possível se obter uma referência para um elemento qualquer da página, por meio do seu ID:

```
window.onload = () => {
  let botao = document.getElementById('bt');
  ...
}

...

<button id="bt">Olá</button>
```

O objeto document

A partir do acesso ao elemento, podemos fazer qualquer alteração na sua estrutura, conteúdo, formatação ou comportamento.

```
window.onload = () => {
  let botao = document.getElementById("bt");
  botao.innerHTML =
              "<strong>Olá mundo!</strong>";
  botao.style.backgroundColor = "#D0D0D0";
  botao.onclick = () => alert("Olá mundo!");
}

...

<button id="bt">Olá</button>
```

O objeto `document`

O conjunto de métodos e propriedades do objeto `document` é chamado de:

DOCUMENT OBJECT MODEL (DOM)

PUC Minas

DOM
Document Object Model

Prof. Marcos André S. Kutova

Código JS

Código CSS

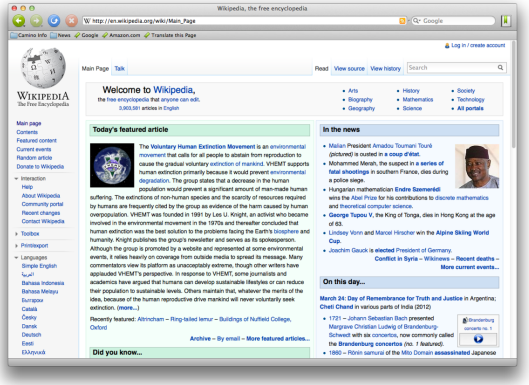
Código HTML

PARSER

document

Objeto que representa a página

RENDER



Página no navegador

Objeto que representa a página

- Alterações no objeto *document*, feitas por meio de JavaScript, resultam em alterações na página (mas não no código fonte)
- A especificação HTML diz quais são os métodos cada elemento do documento (Essa API é chamada de DOM – *Document Object Model*)
- *Por meio desses métodos é possível se alterar:*
  - A estrutura (hierarquia dos elementos)
  - Os conteúdos
  - Os estilos

Métodos para acesso aos elementos do documento

- `document.getElementById( 'id' )`
  retorna um único elemento
- `document.getElementsByTagName( 'tag' )`
  retorna uma lista de elementos
- `document.querySelector( 'seletor' )`
  retorna um único (primeiro) elemento usando a seleção no formato da CSS
- `document.querySelectorAll( 'seletor' )`
  retorna uma lista de elementos

Alteração do conteúdo dos elementos

A alteração do conteúdo dos elementos pode ser feita através da propriedade `innerHTML`.

O conteúdo inclui os seus elementos descendentes

```
document
    .querySelector('#introdução')
    .innerHTML = '<strong>Introdução</strong>';
```

Alteração do conteúdo dos elementos

Se a alteração for apenas do texto, pode-se usar a propriedade `innerText`.

```
document
    .querySelector('#introdução')
    .innerText = 'Introdução';
```

Alteração do estilo dos elementos

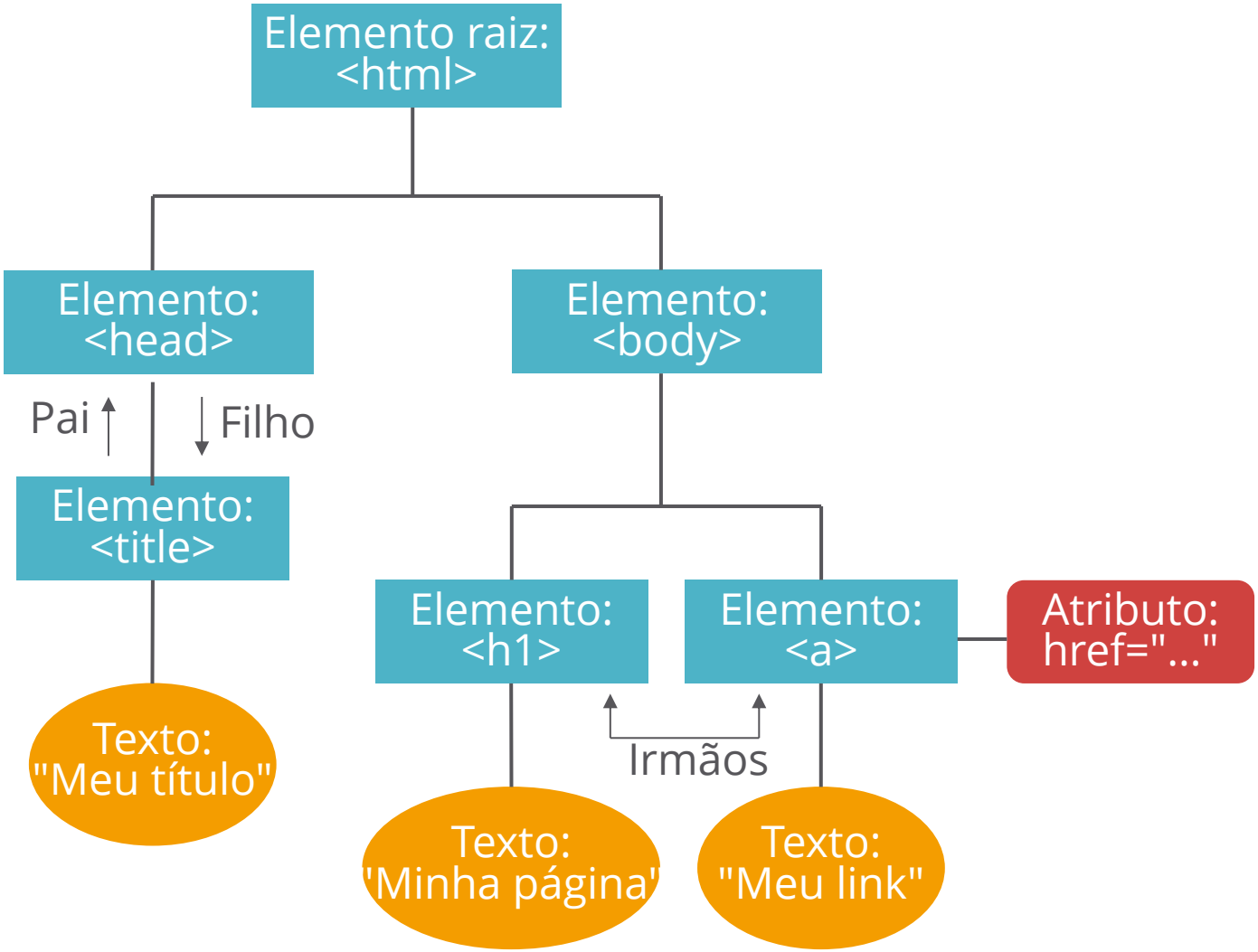
A alteração dos estilos de qualquer elemento nas páginas é feita através do objeto `style` presente na maioria dos elementos do DOM.

A sintaxe geral para modificação dos estilos é:

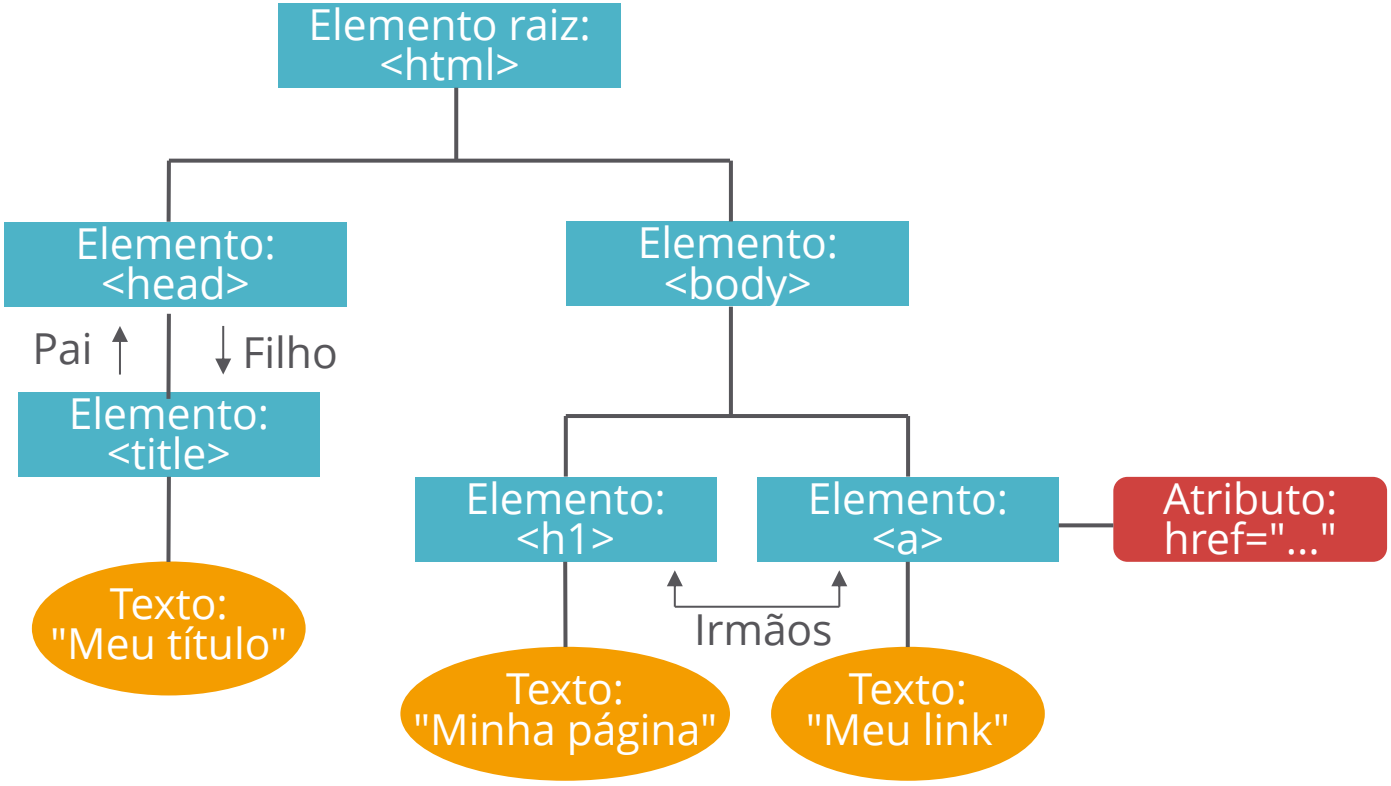
```
document
    .getElementById('id')
    .style.propriedade='valor'
```

```
document.querySelector('#introdução')
        .style.fontWeight = 'bold';
document.querySelector('#total’)
        .style.backgroundColor = '#F00';
```

Árvore de um documento



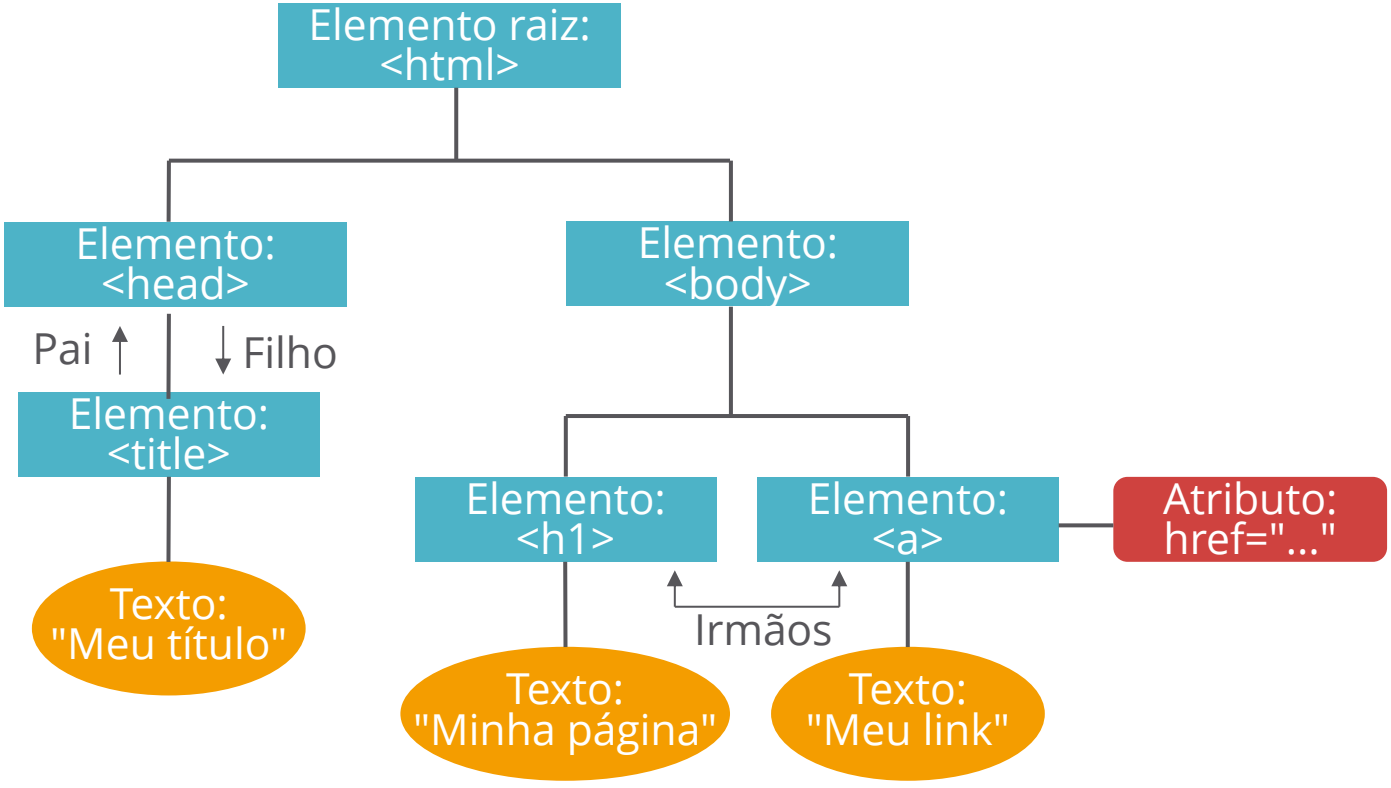
Árvore de um documento



No DOM, tudo é um nodo:

- O documento (inteiro) é um nodo (*document node*)
- Todo elemento é um nodo (*element node*)
- O texto dos elementos são nodos (*text node*)
- Cada atributo é um nodo (*attribute node*)
- Os comentários são nodos (*comment node*)

Árvore de um documento

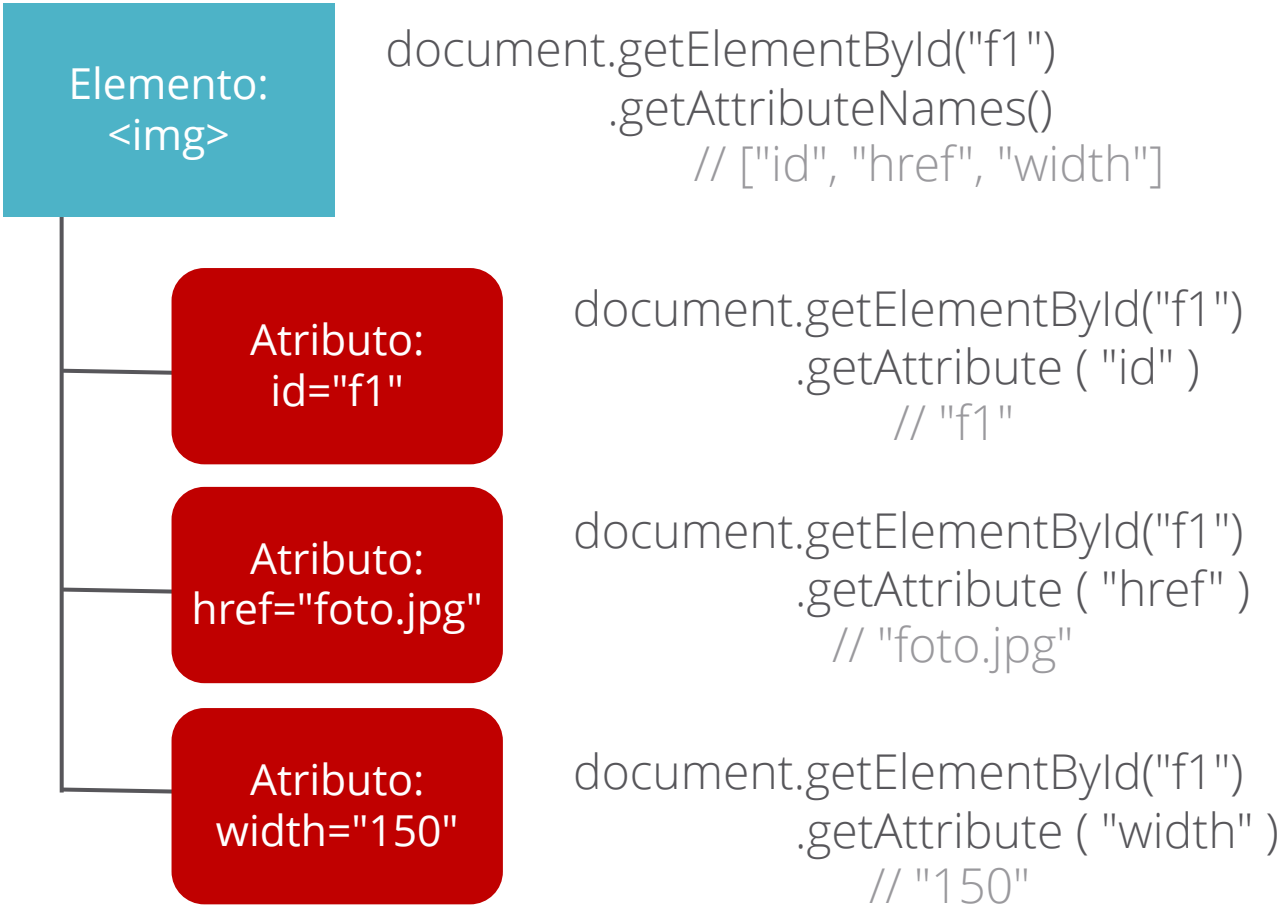


A relação entre os nodos é semelhante a uma árvore

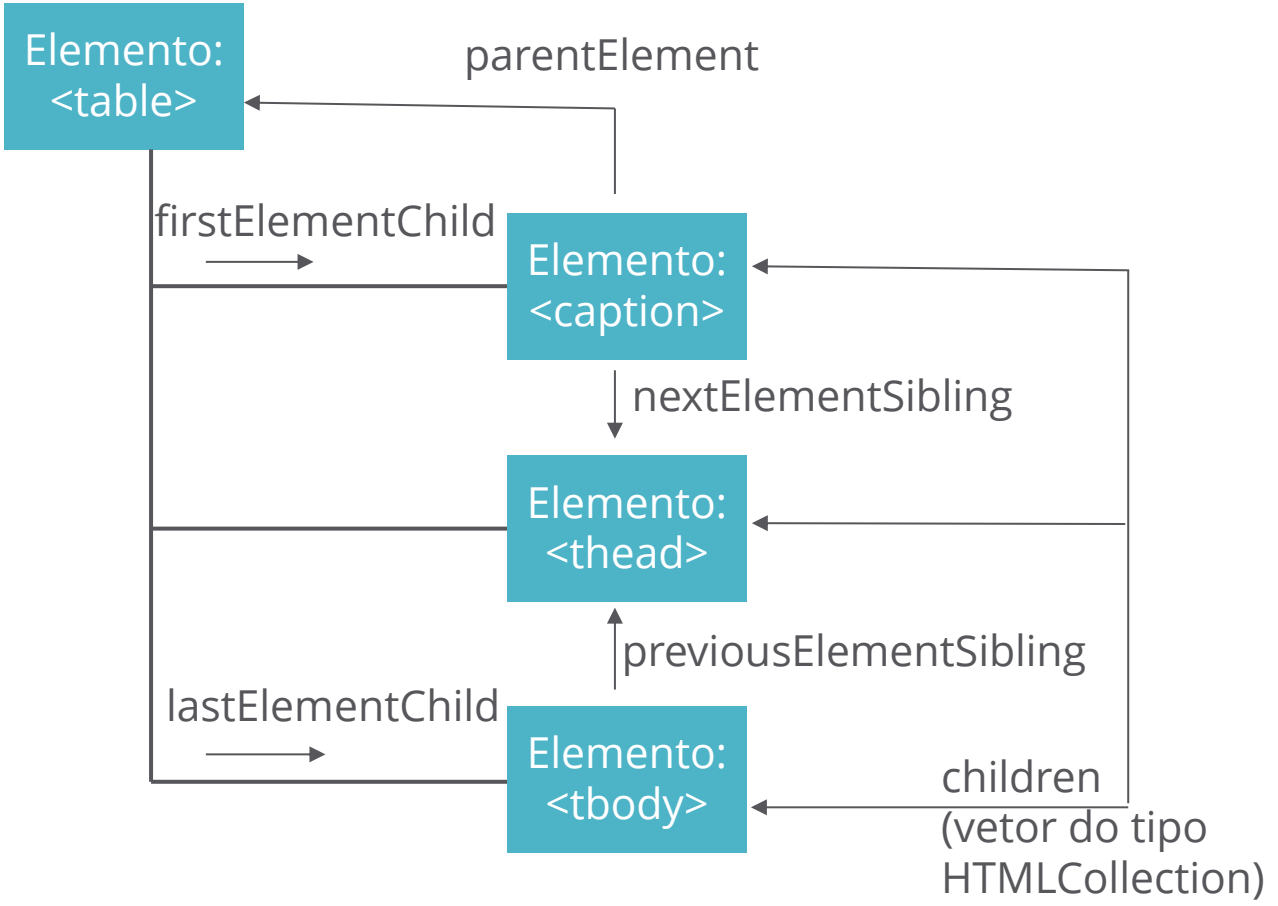
- Eles possuem elementos pai (*parent*) e filho (*child*)
- O nodo documento(*document node*) não possui pai, nem as folhas possuem filhos

Atributos

```
<img id="f1" href="foto.jpg" width="150" />
```



Hierarquia de nodos



- Raiz do documento (elemento <html>): **document.documentElement**
- Quantidade de filhos: **childElementCount**

PUC Minas

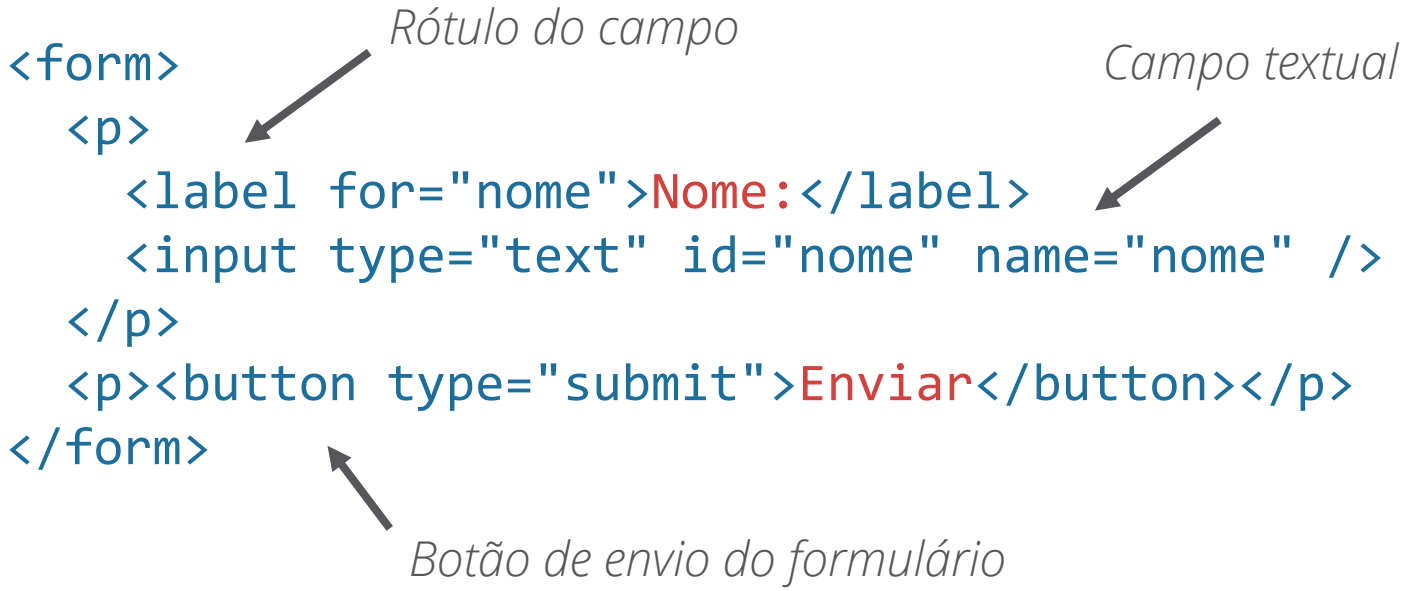
Formulários

Prof. Marcos André S. Kutova

Formulários

Um formulário web é um conjunto de campos por meio dos quais o usuário pode enviar informações para a página ou para um servidor web.

Geralmente esses campos são criados por meio de elementos **<input>** da HTML.



O elemento principal

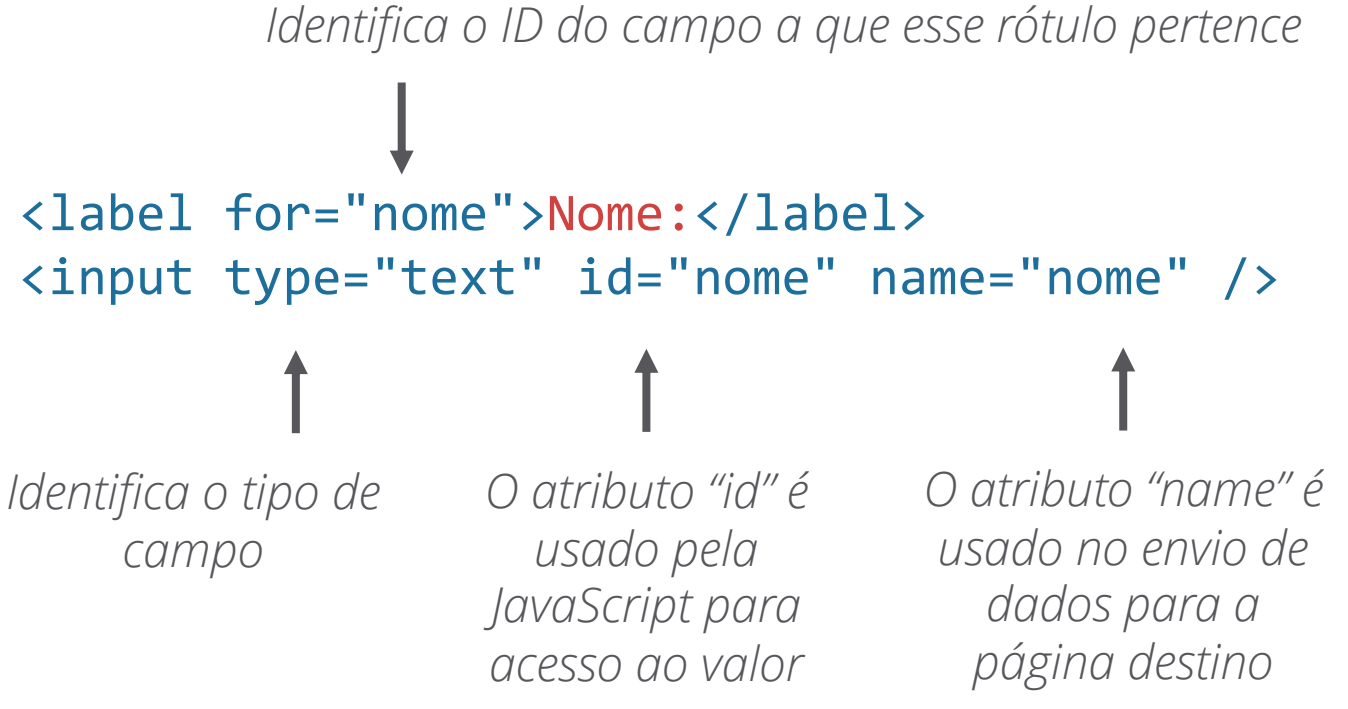
O elemento `<form>` engloba todos os campos e define o que deve ser feito com o fformulário. Geralmente possui dois atributos:

- `action` – define o URL para o qual os dados do formulário devem ser enviados
- `method` – define o método HTTP que será usado no envio dos dados (geralmente `get` ou `post`)

```
<form action="/login.php" method="post">
  ...
</form>
```

O campo textual

O elemento **<input>** é usado para entrada de informações textuais e pode ser acompanhado de um rótulo definido por meio de um elemento **<label>**.



Tipos de campos textuais

O atributo `type` do elemento `<input>` ajuda a definir seu conteúdo ou seu uso. Alguns dos principais tipos textuais são:

`text`	Valor padrão, usado para qualquer tipo de texto
`email`	Campo usado para entrada de e-mails (o teclado poderá se ajustar automaticamente
`date`	Campo para data no formato de dia, mês e ano
`datetime-local`	Campo para data e hora
`number`	Campo para entrada de números
`password`	Campo para entrada de senhas (os caracteres são substituídos por *)
`hidden`	Campo escondido (usado para passagem de valores à página destino sem conhecimento do usuário)

O botão de envio

Um elemento **<button>** pode ser usado para o envio dos dados à página de destino. Esse botão precisa apenas ser do tipo **submit**.

```
<form id="login">
  <p>
    <label for="nome">Nome:</label>
    <input type="text" id="nome" name="nome" />
  </p>
  <p>
    <label for="senha">Senha:</label>
    <input type="password" id="senha" name="senha" />
  </p>
  <p><button type="submit">Enviar</button></p>
</form>
```

Recuperando os valores

O envio do formulário gera um evento **`submit`** que pode ser capturado pela JavaScript.

```
window.onload = () => {
  login.onsubmit = () => {
    ...
    return true;
  };
};
...
<form id="login" action="/login.php"
method="post">
   ...
</form>
```

*Um retorno **false** impediria o envio do formulário*

Recuperando os valores

Cada campo do tipo **<input>** possui um atributo **value** que pode ser acessado ou definido via JavaScript.

```
window.onload = () => {
  login.onsubmit = () => {
    if(nome.value=="" || senha.value=="") {
       alert("Os campos nome e senha " +
             "são obrigatórios");
       return false;
    }
    return true;
  };
};
```

Os eventos do campo

Um campo **<input>** pode ter alguns eventos de teclado ou de mudança de valor.

```
window.onload = () => {
  login.onsubmit = e => {
    console.log(nome.value, senha.value);
      return false;
    };
  nome.onchange =
       () => alert("Novo nome: " + nome.value);
  nome.onkeydown =
       e => console.log(e.code);
};
```

Outros campos

Caixa de texto de várias linhas

```
<textarea id="mensagem"
          name="mensagem"
          cols="40"
          rows="5">
</textarea>
```

Outros campos

Campo de seleção

```
<p>
  <input type="checkbox"
         id="salvar"
         name="salvar"
         checked />
  <label for="salvar">
       Salvar dados do <em>login</em>
  </label>
</p>
```

Outros campos

Campo de seleção exclusiva

```
<p>
  <input type="radio"
         id="sexoM" name="sexo"
         value="masculino" checked />
  <label for="sexoM">Masculino</label>
</p>
<p>
  <input type="radio"
         id="sexoF" name="sexo"
         value="feminino" />
  <label for="sexoF">Feminino</label>
</p>
```

Outros campos

Caixa de seleção

```
<select id="estado" name="estado">
  <option value="ES">Espírito Santo</option>
  <option value="MG" selected>Minas Gerais</option>
  <option value="RJ">Rio de Janeiro</option>
  <option value="SP">São Paulo</option>
</select>
```

Outros campos

Slider

```
<input id="idade"
       name="idade"
       type="range"
       min="18"
       max="60"/ >
```

PUC Minas