

JavaScript para seres humanos

Everson Santos Araujo

everson@vuse.com.br

<http://everson.com.br>



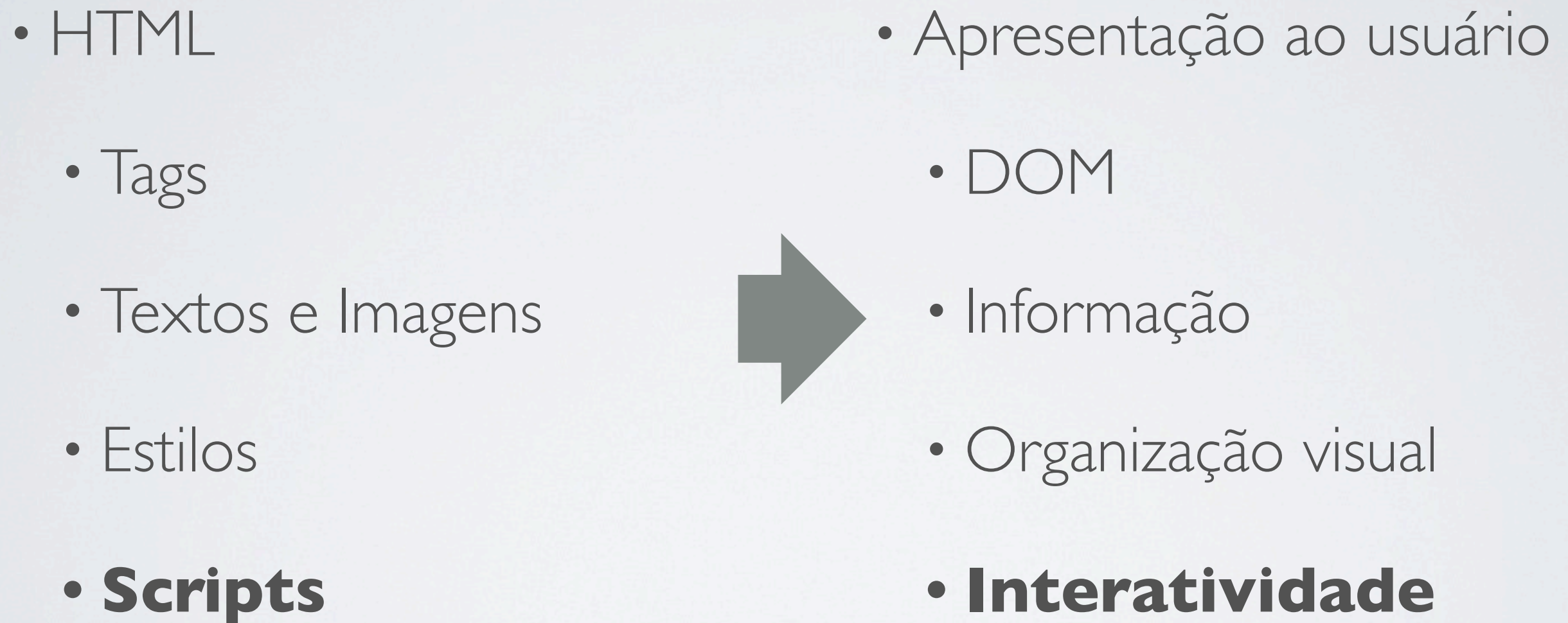
<http://vuse.com.br>

JAVASCRIPT

- Linguagem de alto nível criada para ser executada pelo navegador de internet e oferecer interatividade à documentos HTML
- Hoje existe em aplicações desktop, móveis e servidores



NAVEGADORES

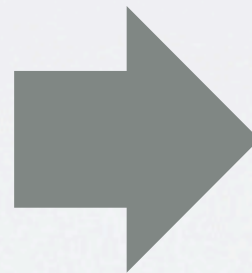


JQUERY

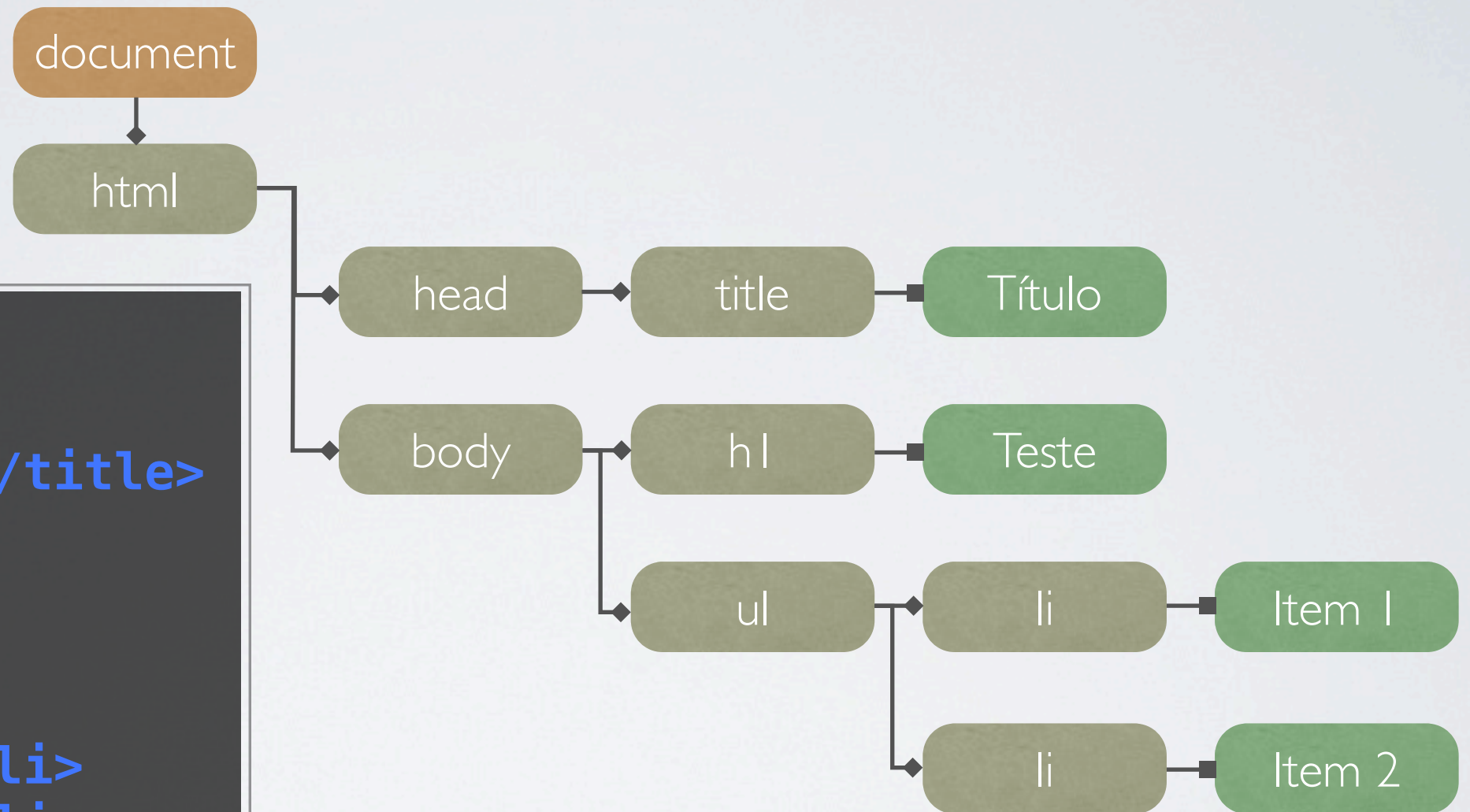
- Biblioteca JavaScript rápida, pequena e com vastos recursos
- Possui uma interface fácil de usar e que funciona igualmente na grande maioria dos navegadores
- **CDN** Google
 - `//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js`

HTML - DOM

```
<html>
<head>
  <title>Título</title>
</head>
<body>
  <h1>Teste</h1>
  <ul>
    <li>Item 1</li>
    <li>Item 2</li>
  </ul>
</body>
</html>
```

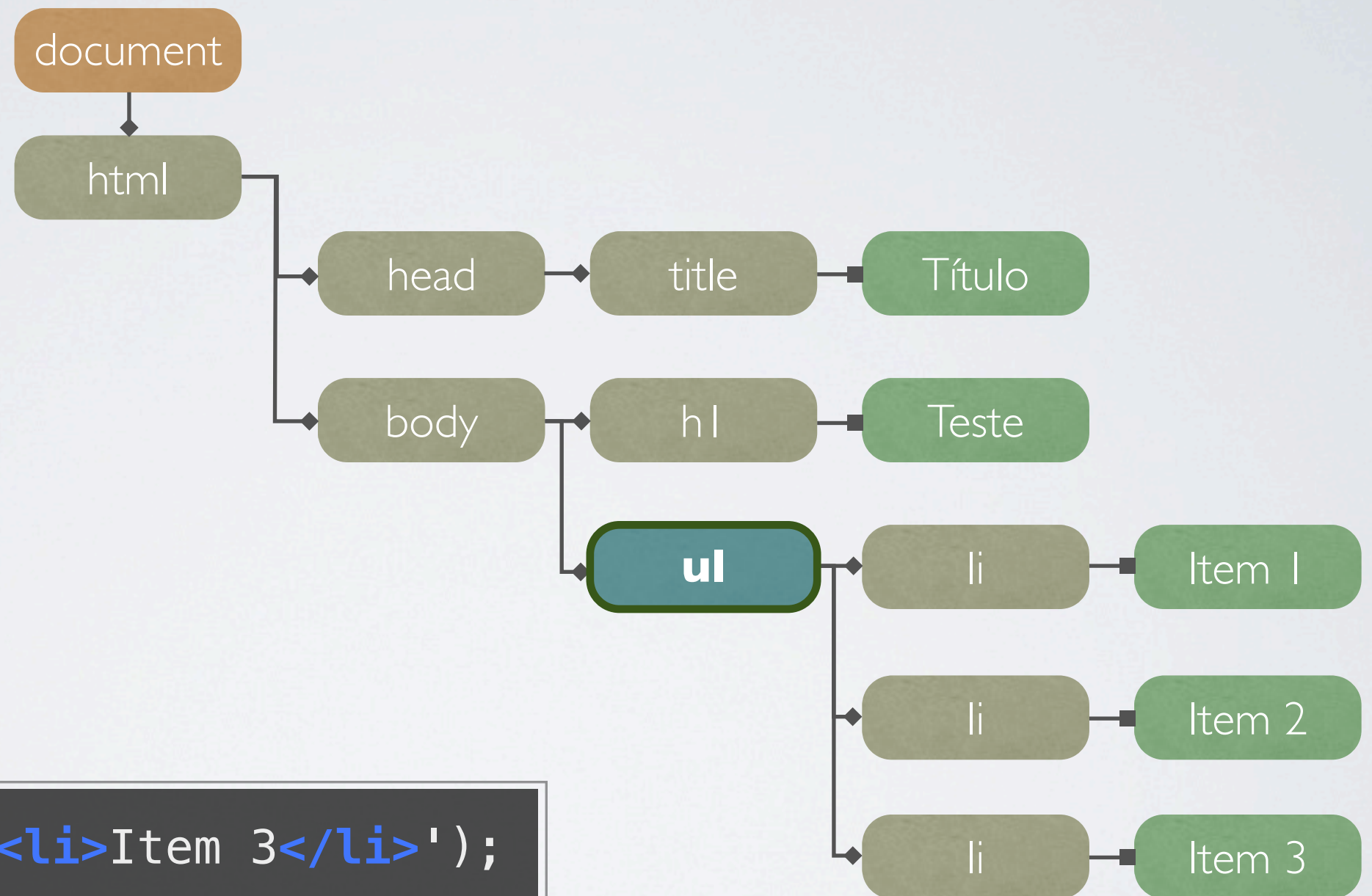


HTML - DOM



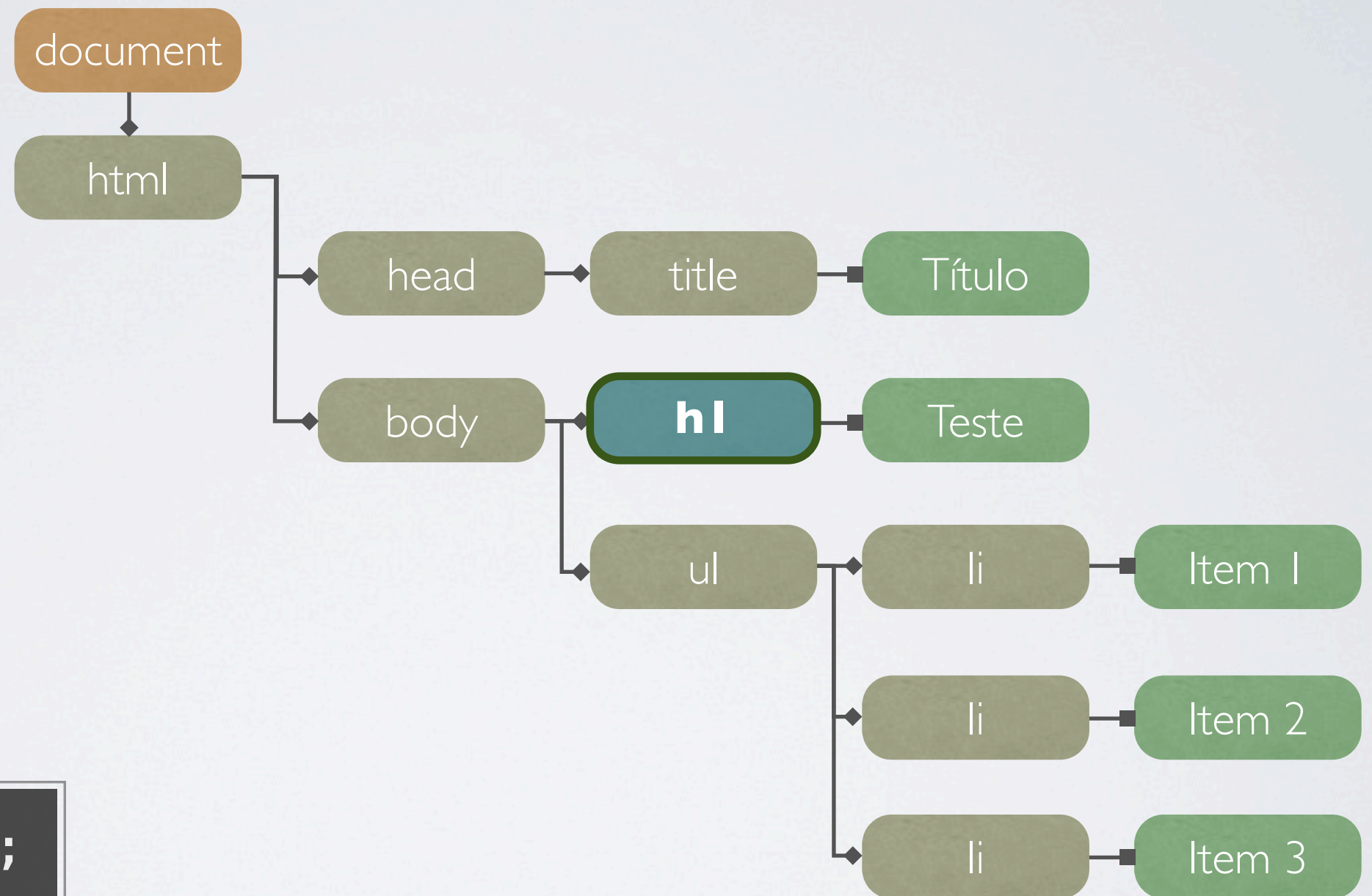
```
<html>
<head>
  <title>Título</title>
</head>
<body>
  <h1>Teste</h1>
  <ul>
    <li>Item 1</li>
    <li>Item 2</li>
  </ul>
</body>
</html>
```

HTML - DOM



```
$( 'ul' ).append( '<li>Item 3</li>' );
```

HTML - DOM






```
$('h1').remove();
```


ANTES DE CONTINUAR...

Não serão apresentadas todas as funções disponíveis na biblioteca

As funções iniciadas com \$ são chamadas diretamente sem precisar ser direcionada com um seletor

A utilização de parâmetros será identificada com cores

-  obrigatório
-  recomendado
-  opcional

SELETORES

- `$('#identificador')`

```
$('#id').append('ola!');
```

- `$('.classe')`

```
$('.clas').append('Mundo');
```

- `$('elemento[atributo]')`

```
$('input[name=nome]').val('A');
```

- `$('#id .classe elemento')`

```
$('div .aqui').append('!');
```

- `$('elemento:pseudo')`

```
$('li:last').append('X');
```

- `$('.multiplo, .classe2')`

```
$('.c1, .c2').append('?');
```


EVENTOS

- `.change(função)`
 - Ao alterar dados
- `.click(função)` e `.dblclick(fun..)`
 - Ao clicar
- `.hover(função)`
 - Passar o mouse
- `.keypress(função)`
 - Pressionar uma tecla
- `.submit(função)`
 - Enviar formulário
- `.trigger(função)`
 - Executar evento

MANIPULAÇÃO

- `.addClass()` e `.removeClass()`
 - Adiciona e remove classes
- `.append(elemento)`
 - Inserir no final do seletor
- `.attr(atributo, valor)`
 - Retorna ou altera atributo
- `.html(html)`
 - Retorna ou altera HTML
- `.text(texto)`
 - Retorna ou altera texto
- `.wrap(elemento)`
 - Inclui elemento ao redor

EXEMPLO

```
<html>
<head>
  <style type="text/css">
    .red {
      background: red;
    }
  </style>
</head>
<body>
  <h1>Teste</h1>
  <form>
    <input id="texto">
    <input type="submit"
      value="Enviar">
  </form>
</body>
</html>
```

```
$( 'h1' ).hover(function() {
  $(this).toggleClass( 'red' );
}).click(function() {
  $(this).append( '<u>!!</u>' );
});
$( 'input' ).keypress(function(e) {
  console.log( e.charCode );
}).dblclick(function() {
  $(this).addClass( 'red' );
});
$( 'form' ).submit(function(e) {
  e.preventDefault();
  console.log( 'Enviar' );
});
```

EFEITOS

- `.show()` e `.hide()`
 - Mostra e esconde elemento
- `.fadeIn(ms)` e `.fadeOut(ms)`
 - Mostrar e esconder elemento usando transparência
- `.slideDown(ms)`
 - Mostrar e esconde arrastando
- `.slideUp(ms)`
 - Mostrar e esconde arrastando
- `stop(ms)`
 - Parar animações

ms - tempo em milissegundo

EXEMPLO

```
<button id="mo">show</button>
<button id="es">hide</button>
<button id="fi">fadeIn</button>
<button id="fo">fadeOut</button>
<button id="sd">slideDown</button>
<button id="su">slideUp</button>
<div>
  <p>Lorem ipsum</p>
  <p>Lorem ipsum</p>
</div>
```

```
$( '#mo' ).click(function() {
  $( 'div' ).show();
});
$( '#es' ).click(function() {
  $( 'div' ).hide();
});
$( '#fi' ).click(function() {
  $( 'div' ).fadeIn(2000);
});
$( '#fo' ).click(function() {
  $( 'div' ).fadeOut(2000);
});
$( '#sd' ).click(function() {
  $( 'div' ).slideDown(2000);
});
$( '#su' ).click(function() {
  $( 'div' ).slideUp(2000);
});
```

AJAX

- \$.ajax(url)
 - Enviar dados
- .load(url)
 - Carregar para elemento
- \$.get(url, dados, sucesso)
 - Enviar dados por GET
- \$.post(url, dados, sucesso)
 - Enviar dados por POST
- .serialize() e .serializeArray()
 - Retorna dados dos elementos

EXEMPLO

```
<button id="lo">load</button>
<button id="po">$.post</button>
<div id="re"></div>
```

```
$('#lo').click(function() {
    $('#re').load('ex.html');
});
$('#po').click(function() {
    $.post('ex.php', {
        teste: 'oi'
    },
    function(r) {
        $('#re').html(r);
    });
});
```

ex.html

```
<h1>Exemplo</h1>
<p>Lorem</p>
```

ex.php

```
<?php
if (count($_POST)) {
    echo "<h1>Dados recebidos</h1><ul>";
    foreach ($_POST as $k=>$v) {
        echo "<li>" . $k . " = " .
        $v . "</li>";
    }
    echo "</ul>";
}
```