



# Universidade Federal do Rio Grande do Norte - UFRN

Instituto Metr pole Digital – IMD  
Document Object Model - DOM

Nelson Ion de Oliveira



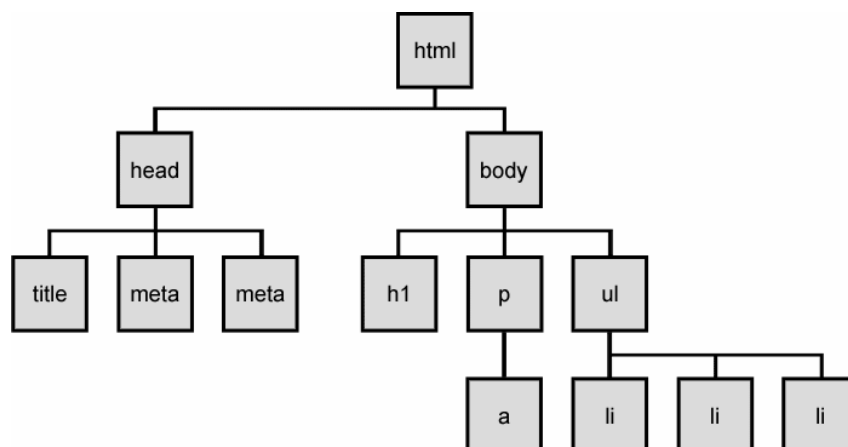
## Document Object Model

- Definido pela W3C como:
  -   uma plataforma e interface de linguagem neutra que permite que programas e scripts acessem e atualizem dinamicamente o conte do, estrutura e estilo de documentos.
-   uma multi-plataforma que representa como as marca  es (tags) e m HTML/xHTML s o organizadas e lidas pelo navegador.

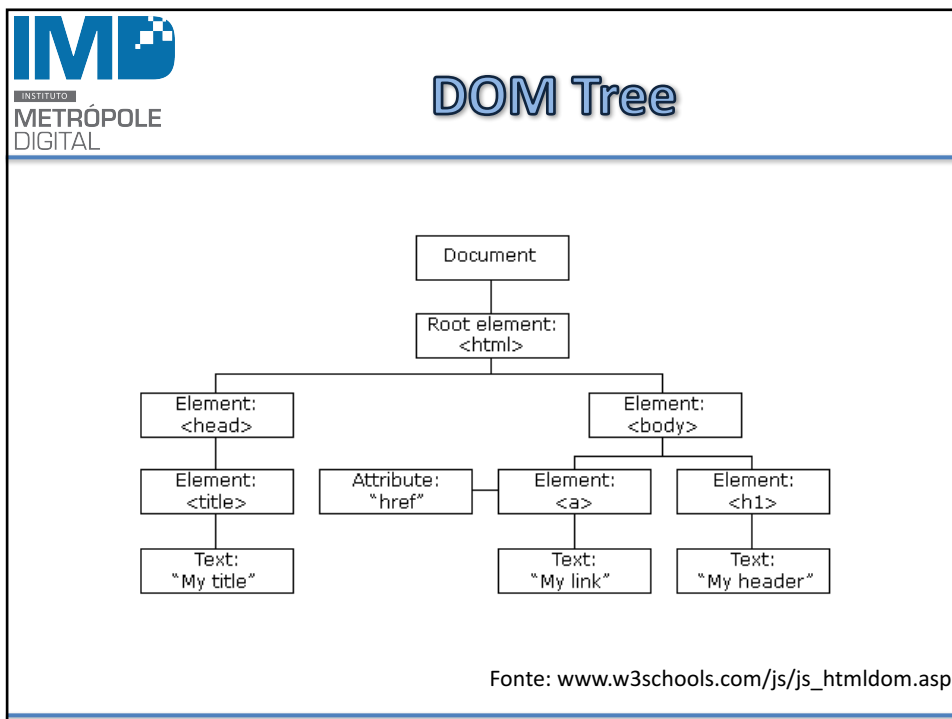
## Why DOM?

- Entender realmente como um navegador funciona é importante para qualquer desenvolvedor web.
- Manipular páginas através do DOM é extremamente fácil.

## DOM Tree



Fonte: [tableless.com.br/tenha-o-dom/](http://tableless.com.br/tenha-o-dom/)



**IMD**  
INSTITUTO  
METRÓPOLE  
DIGITAL

## DOM

- Agora que você conhece a estrutura do DOM, podemos dizer que:
  - manipular elementos através do DOM é muito semelhante ao de aplicar formatações em documentos HTML por meio de CSS.
  - As duas tarefas, manipular elementos e formatar, assumem o seguintes padrão geral:
    1. Especificar o elemento ou grupo de elementos que você deseja afetar;
    2. Especificar o efeito/ação que você deseja aplicar neles.

## JavaScript + DOM

- Você poderá:
  - Mudar todos os elementos HTML na página;
  - Alterar todos os atributos do HTML na página;
  - Modificar todos os estilos CSS na página;
  - Remover ou adiciona elementos HTML e atributos existente.
  - JavaScript pode reagir a todos os eventos HTML existentes na página.

## DOM


- Vamos ver um exemplo rápido de manipulação de DOM?
- Acesse:  
[www.anra.com.br/ufrn/aula18/aula18a.html](http://www.anra.com.br/ufrn/aula18/aula18a.html)
- Entendeu o comportamento?

## DOM

- Em HTML, todos os elementos são definidos como **objetos**.
- Esses objetos possuem métodos e propriedades.
  - O que é uma propriedade?
  - O que é um método?

## DOM

- Veja o código fonte:  
[www.anra.com.br/ufrn/aula18/aula18b.html](http://www.anra.com.br/ufrn/aula18/aula18b.html)



# DOM

Objeto


Método

Propriedade

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Título da Página</title>
5      <meta charset="utf8" />
6      <script language="text/javascript">
7        function digaOi(nome){
8          document.getElementById("saudacao").innerHTML = 'Ola <b>' + nome + '!' </b>';
9        }
10
11       function perguntarNome(){
12         nome = prompt('Qual seu nome?');
13         digaOi( nome );
14       }
15     </script>
16   </head>
17   <body>
18     <p>O conteúdo da página aqui...</p>
19
20     <button onclick="perguntarNome();">Clique</button>
21
22     <div id="saudacao"> Olá! </div>
23
24   </body>
25 </html>
  
```

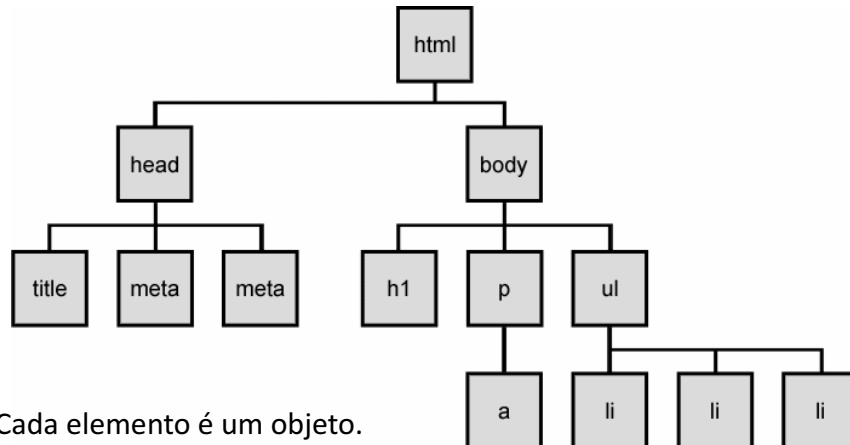
Valor



# DOM

- O método getElementById, está disponível **apenas** no **document**, recebe uma String como argumento e, então, localiza o elemento que tem essa String como ID.
  - Caso não encontre o elemento, o método retorna **null**.
- Com base no exemplo de código do slide anterior, faça:
  - Adicione uma nova saudação **sem** remover as anteriores.
  - Cada nova saudação deve ser colocada em uma nova linha.
  - Altere o que for preciso, mas a dica é de que apenas adicionado novos 7 caracteres será suficiente (mas pode existir resposta mais simples ainda!).

## Relembrando o DOM Tree



- Cada elemento é um objeto.
- Então podemos criar novos objetos...
- E associar os novos objetos!

Fonte: [tableless.com.br/tenha-o-dom/](http://tableless.com.br/tenha-o-dom/)

## DOM

- Isso que fizemos no exemplo anterior foi “artesanal”.
- Abra o exemplo do link:  
<http://www.anra.com.br/ufrn/aula18/aula18f.html>
- Abra o console do seu navegador.

- No console, faça:  
`var novoitem = document.createElement("li");`  
`novoitem.innerHTML = "Um novo elemento";`  
`document.getElementById("nomes").appendChild(novoitem);`
- E se eu fizer agora isso?  
`novoitem.innerHTML = "Será que funciona?"`;

- Vamos manipular tabelas?
- Acesso o arquivo  
– [www.anra.com.br/ufrn/aula18/aula18g.html](http://www.anra.com.br/ufrn/aula18/aula18g.html)
- Em seguida, abra o console do seu navegador e execute o comando:  
`var tabela = document.getElementById("cadastros");`



## DOM

- Para inserir uma nova linha na tabela:  
`var row = tabela.insertRow();`
- Para inserir células na linha criada acima:  
`row.insertCell();`
- Para inserir uma linha já com o conteúdo:  
`row.insertCell().innerHTML = "Conteúdo";`
- Se você quiser acessar um elemento da linha "atual":  
`row.cells[i] //Onde i é o índice da célula.`

## Exercícios

### Crie um objeto Pessoa:

Com as **propriedades**

- primeiroNome
- ultimoNome
- Nascimento
  - formato ano-mês-dia (String)
- altura
- peso
- sexo

Com os **métodos**

- nomeCompleto()
- idadeAtual()
  - //calcule com base no ano de 2015, uma simples subtração do ano de nascimento com o ano atual.
- dataNascimento()
- diaNascimento()
- mesNascimento()
- anoNascimento()
- imc()

## Exercícios

- Com base na classe do exercício da **aula anterior**, faça:
  - Um cadastro de pessoas que recebe um novo cadastro ao se clicar em **Cadastrar**.
- Ao final do cadastro:
  - Inserir em nova linha no final da tabela.
  - com o nome completo em maiúsculas e seus respectivos dados.
- Veja exemplo no próximo slide.

## Exercícios

- Observe que, com exceção do nome, as demais informações da tabela estão centralizadas.
- Pesquise sobre a propriedade `style` para aplicar formatação de centralizar o conteúdo das colunas de Idade, Sexo, Nascimento e IMD.

Nome	Idade	Sexo	Data Nascimento	IMC
LARANJA DO SENADO	35	Masculino	1980-01-08	32,56
NELSON ION DE OLIVEIRA	22	Masculino	1993-10-08	19,59
NINJA POTIGUAR	55	Masculino	1993-10-08	26,52

## Referências

- Textos e Exemplos retirados dos sites:
  - [www.w3schools.com/js/](http://www.w3schools.com/js/)
  - [www.codecademy.com](http://www.codecademy.com)
  - [www.w3.org/DOM/](http://www.w3.org/DOM/)
  - [tableless.com.br/tenha-o-dom/](http://tableless.com.br/tenha-o-dom/) (recomendo a leitura)

## Dúvidas?

