

Bloco 5.1

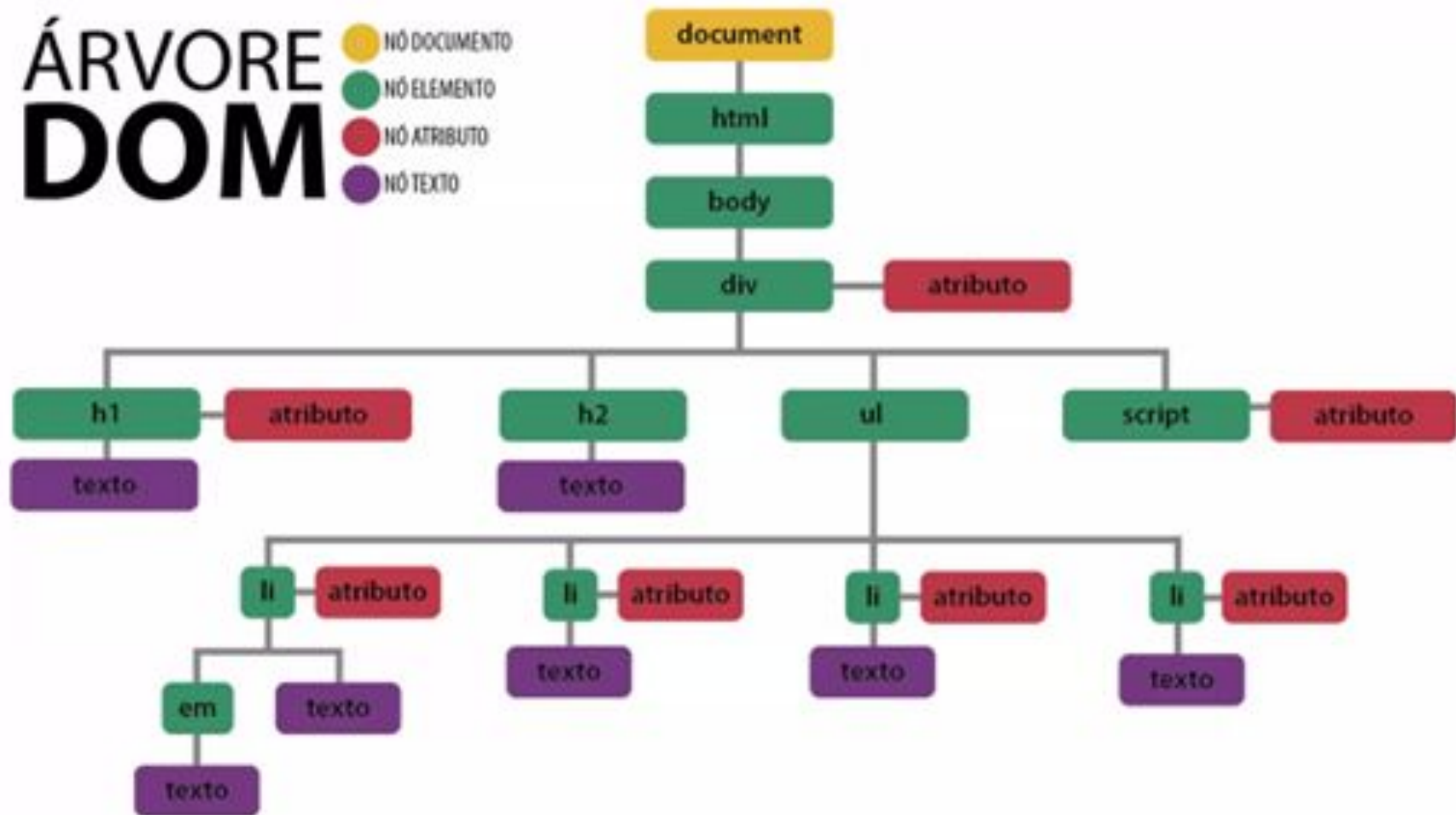
JavaScript - DOM e Seletores

Elaborado por Ruâni Filipe (aluno - Turma 8), com base em anotações pessoais do assunto, via Course e Aulas ao Vivo

DOM (Document Object Model)

ÁRVORE DOM

- NÓ DOCUMENTO
- NÓ ELEMENTO
- NÓ ATRIBUTO
- NÓ TEXTO



DOM (Document Object Model)

Window -> Página completa

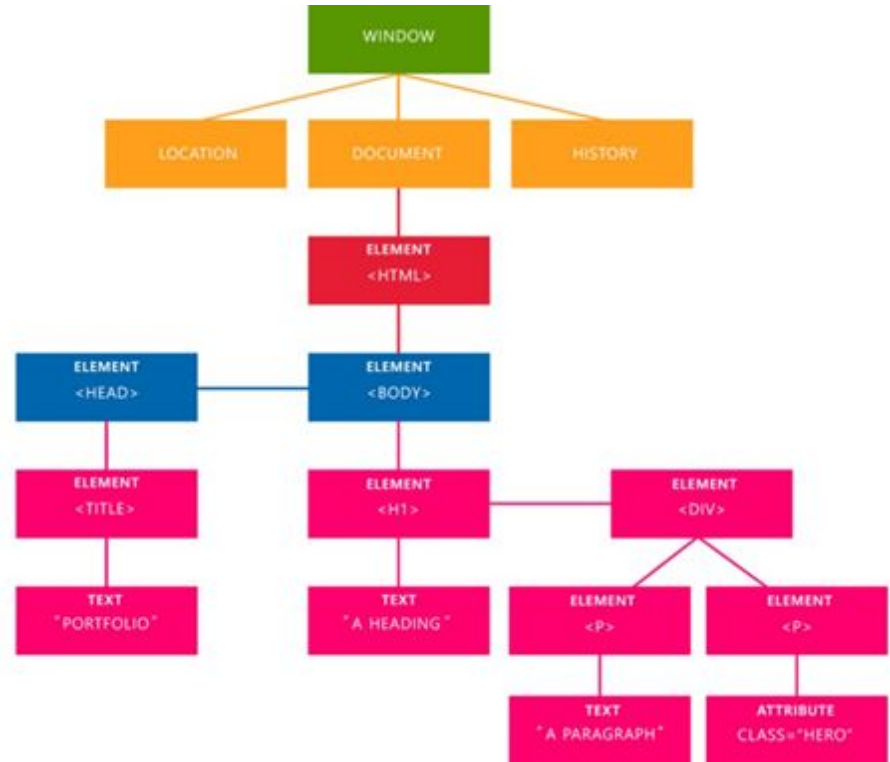
Location -> Localização do objeto

Document -> A página em si

History -> Histórico da Sessão

Element -> Elementos HTML

Nó -> Cada caixa do fluxograma



Seletores

Como a página sabe como acionar os seletores?

Por meio da tag `<script> </script>`, podendo ser de duas formas:

“Inline” -> fazendo o código JS diretamente no arquivo HTML
(não é boa prática, por dar uma aparência “suja” ao HTML)

Externo -> abrir um arquivo externo (normalmente chamado `script.js`, mas pode ser qualquer nome.js) e referenciar, tal qual fazemos com o CSS e pode ser feito de duas formas:

- 1) Ao final do body, incluindo a linha `<script src="script.js"></script>`
- 2) Final Head ou Início do Body, DESDE QUE tenha o atributo `defer` `<script src="script.js" defer></script>` (veremos em blocos mais à frente sobre esse atributo)

Como usar os os seletores? Quais são?

O objeto `#document` nos dá acesso para manipular/alterar o DOM

Ex.: se fizermos `document.body` ele irá retornar tudo o que está entre as tags `<body>` e `</body>`

Seletor `getElement` -> por meio dele você conseguirá acesso para manipular ID, Classes e Tags, por meio de suas funções:

Função *`getElementById`*

Função *`getElementsByClassName`*

Função *`getElementsByTagName`*

Função `querySelector` e `querySelectorAll`

Função getElementById

Inicialmente, atenção:

Escrever getElementByID causa erro;

Verificar a referência ao Id, pois é Case-Sensitive
(diferencia MAIÚSCULAS de minúsculas)

Estrutura: `document.getElementById("nomeDoSeuld")`

Boa prática fazer `let nomeDescritivo = document.getElementById("nomeDoSeuld")`, pois fica mais clean referenciar ao `document.getElementById` apenas pelo nome da variável

Função getElementById - Manipulando o DOM

Para manipular por meio das funções, há duas principais formas:

`.innerHTML` -> preferível usar desse modo, que tem maior aplicabilidade, não se recomenda quando houver <tags> no meio do texto (, , , <i>...)

`.innerText` -> preferível usar quando há <tags> ao longo dos textos

Estrutura -> `document.getElementById.innerHTML(ou.innerText) = "novoTexto"`

Estilizações Estrutura -> `document.getElementById("nomeId").style.seletorCSS = valor`

ex.: `document.getElementById("nomeId").style.color = "blue"`

Função `getElementsByClassName`

Inicialmente, atenção:

Ele retorna como um `Array`, então, para ter acesso, lembre-se de usar os `[indices]`;

O nome da função é `getElements` e não `getElement`;

Estrutura: `document.getElementsByClassName("nomeDaSuaClasse")`

Mesma boa prática que vimos anteriormente se aplicará aqui, qual seja, a de criar uma variável para depois referenciar a função por meio do nome descritivo

Função `getElementsByClassName` - Manipulando o DOM

Estrutura -> Por se tratar de Array, é necessário percorrer esse Array ou acessá-lo por meio do índice, como pode ser visto no exemplo abaixo, com base no vídeo do Course

```
let pilotosDeF1 = document.getElementsByClassName("nomeClasse")

for (let i = 0; i < pilotosDeF1.length; i +=1) {

  pilotosDeF1[i].innerText="Lewis Hamilton" (poderia ser também
pilotosDeF1[i].style.color ou pilotosDeF1[i].style.background ou diversas outras opções)
```

ATENÇÃO: Mesmo que o Array contenha apenas um índice, deverá acessar via índice, no caso, por meio do índice [0]

Função `getElementsByTagName`

Inicialmente, atenção:

Ele retorna como um Array, então, para ter acesso, lembre-se de usar os [índices];

O nome da função é `getElementsByTagName` e não `getElement`;

Lembrando que Tag é o elemento HTML (`<div>`, `<p>`, `<section>`, `<header>`...)

Estrutura: `document.getElementsByTagName("nomeDaSuaTag")`

Mesma boa prática que vimos anteriormente se aplicará aqui, qual seja, a de criar uma variável para depois referenciar a função por meio do nome descritivo

Função `getElementsByTagName` - Manipulando o DOM

Estrutura -> Por se tratar de Array, é necessário percorrer esse Array ou acessá-lo por meio do índice, como vimos na Manipulação da `ClassName`

Apenas destacando que a estrutura da função é variada, a depender do que se necessita, assim, a função pode estilizar fonte, cor, background...

Função querySelector e querySelectorAll

Inicialmente, atenção:

`querySelector` retorna apenas o primeiro elemento

Essas funções trabalham mais especificamente com os seletores CSS, ao contrário do que se faz em `getElementById` (“nomeId”), por exemplo, aqui se deve fazer tal qual no CSS

(# -> para Id’s e . -> para Classes)

Se fizer `document.querySelector(“nomeId”)` retornará null

`querySelectorAll` retornará todos os elementos com a especificação dada, NA FORMA DE ARRAY

Função querySelector e querySelectorAll - Manipulando o DOM

```
function alteraCor(elementos, cor){  
  document.querySelector(elementos).style.backgroundColor = cor;  
};
```

Exemplo feito em Aula ao Vivo, usando a manipulação do querySelector dentro de uma função.

No caso acima, a função quando chamada são passados 2 parâmetros (elementos, cor); uma vez chamada, o querySelector irá pegar os elementos (lembrando que ele retornar apenas o primeiro) e irá aplicar a cor passada como 2º parâmetro da função

ATENÇÃO: é apenas um exemplo de inúmeras utilizações possíveis das funções ;)