

## Bloco 5.2

# JavaScript - Trabalhando com Elementos

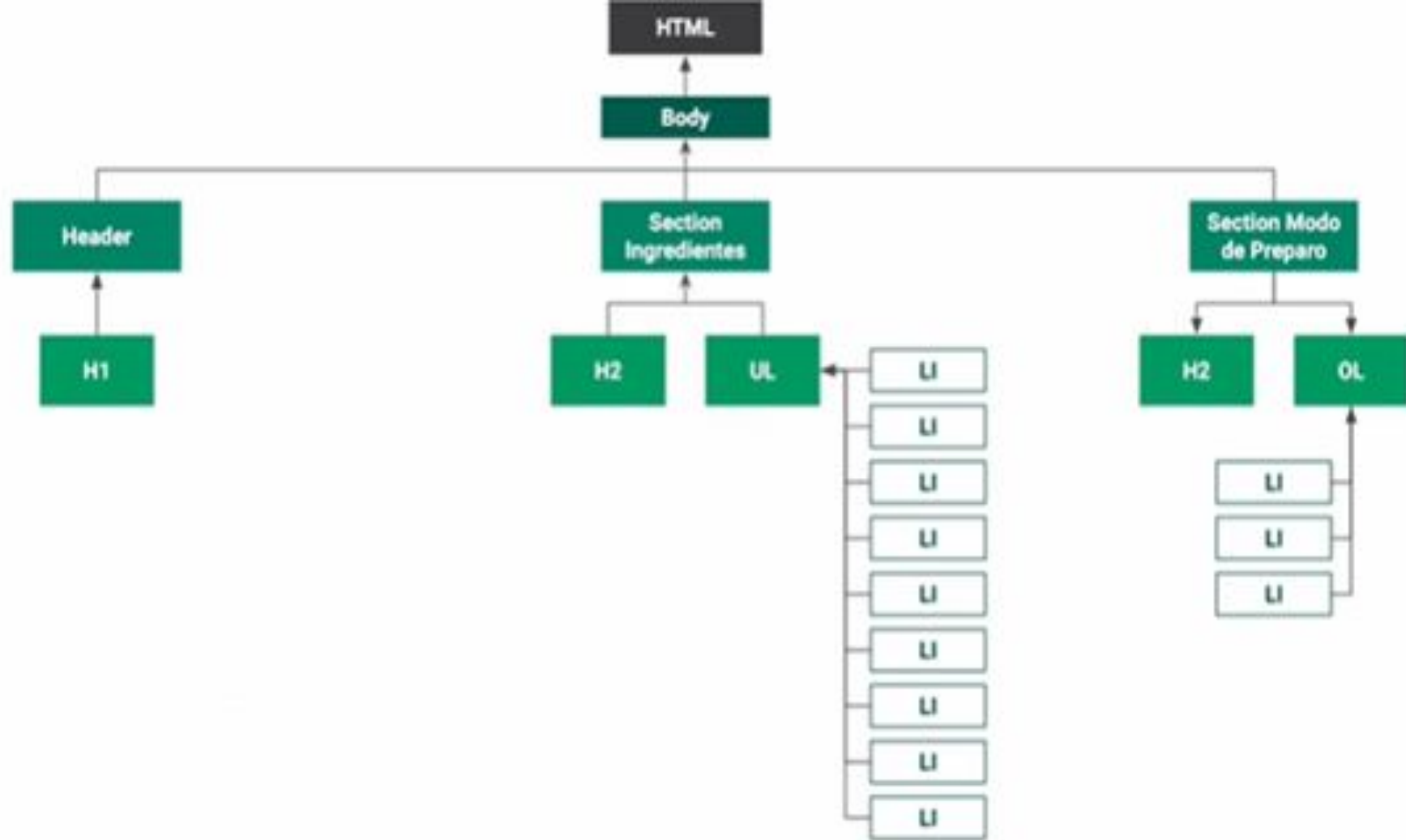
Elaborado por Ruâni Filipe (aluno - Turma 8), com base em anotações pessoais do assunto, via Course e Aulas ao Vivo

**Objetivos:**

**Adicionar e remover elementos HTML,  
usando o JavaScript**

**Buscar elementos da página, usando a  
propriedade parentNode**

# Buscando Elementos

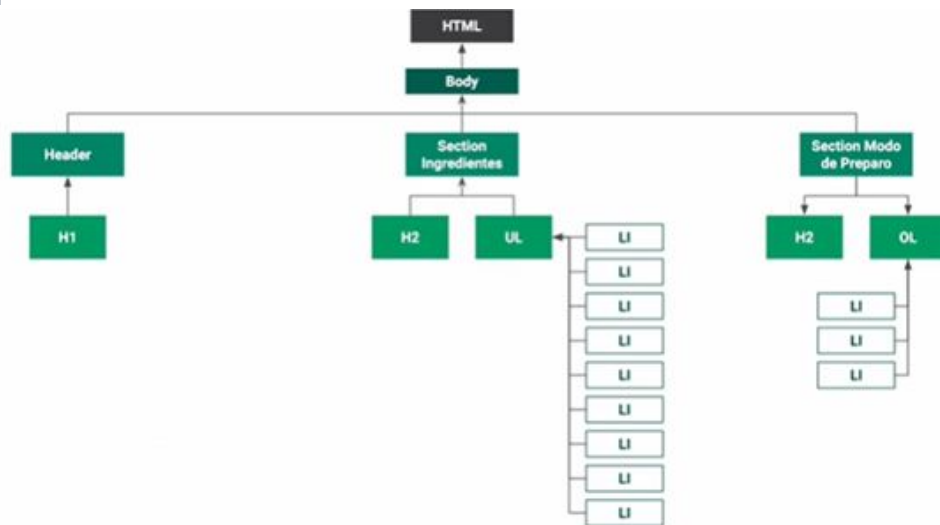


# Buscando Elementos

Para melhor entender o assunto, vale olhar a árvore DOM e fazer relações de parentesco, tal qual uma árvore genealógica

Observando a imagem ao lado (igual à imagem anterior), percebemos que:

Body seria o pai de todos os elementos de sua linha abaixo, assim como o Header seria o pai de relação ao seu H1, assim como e que os `<li>` são filhos de `<ul>` e assim por diante



# Buscando Elementos (.parentNode)

Estruturação: `document.querySelector (.nomeClasse).parentNode`

Significa -> qual o elemento pai do .nomeClasse?

Obs.: Vale também para (`#nomeld`) e (`tag`)

Também é possível subir as “gerações” e buscar pelo avô, bisavô...

Basta concatenar outros `.parentNode`

Ex.: `document.querySelector (.nomeClasse).parentNode.parentNode`

Atenção: A propriedade `.parentNode` é somente leitura, isto é, não se pode reatribuir valores por meio dela

# Buscando Elementos (outras propriedades de “filiação”)

Estruturação: `document.querySelector (.nomeClasse).childNodes`

Significa -> quais são os elementos filhos? (retorna na forma de Array)

Obs.: ele retorna as linhas (que atendam à filiação) por completo

Estruturação: `document.querySelector (.nomeClasse).children`

Retorna apenas os elementos HTML (e não a linha por inteiro) que sejam filhos

# Buscando Elementos (outras propriedades de “filiação”)

## firstChild x firstElementChild

Estruturação: `document.querySelector (.nomeClasse).firstChild`

Significa -> qual o primeiro nó-filho?

Estruturação: `document.querySelector (.nomeClasse).firstElementChild`

Significa -> qual o primeiro elemento-HTML-filho?

## nextSibling x nextElementSibling

Segue o mesmo raciocínio acima, `.nextSibling` -> traz o próximo nó; enquanto

`.nextElementSibling` traz o próximo elemento



# **Criando Elementos e Adicionando e Removendo Filhos**

# Criando elementos (.createElement)

Estruturação: `document.createElement("h1").innerHTML`

Entre parênteses é o parâmetro da função `.createElement`

Vale também para `(#nomeld)` e `(.nomeClasse)`

Exemplo prático ao lado:

uma variável para melhor organizar, criando `div`;

`div.className` é para atribuir o nome da `div` criada;

`div.id` atribui o nome ao `id` da `div` criada

estrutura criada ao final `<div class...>`

```
> let div = document.createElement('div');  
< undefined  
> div.className = 'month-container';  
< "month-container"  
> div.id = 'algumaCoisa';  
< "algumaCoisa"  
> console.log(div);
```

VM758:1

```
<div class="month-container" id=  
  "algumaCoisa"></div>
```

```
< undefined
```

# Criando filhos (.appendChild)

Estruturação: `document.nomePai.appendChild(nomeFilho)`

Ex.: `document.body.appendChild(div)` -> para criar um filho `<div>` ligado direto ao `<body>`

Exemplo ao lado mostra:

- 1) a criação do (desgarrado) elemento filho;
- 2) a criação do “cordão umbilical” com o body;
- 3) estilizações diversas

`.innerHTML` -> acrescentando texto;

`.style.backgroundColor` -> acrescentando cor de fundo

```
> let div = document.createElement('div');  
< undefined  
> document.body.appendChild(div);  
< <div></div>  
> div.innerHTML = "OLÁ PESSOAL!!";  
< "OLÁ PESSOAL!!"  
> div.style.backgroundColor = 'yellow';  
< "yellow"  
>
```

# Removendo filhos (.removeChild)

Estruturação: `document.nomePai.removeChild(nomeFilho)`

Ex.: `document.body.removeChild(div)` -> para remover um filho `<div>` ligado direto ao `<body>`

```
> let divs = document.querySelectorAll('div');  
< undefined  
  
> for (let div of divs){  
    document.body.removeChild(div);  
}  
< ▶ <div class="days-container">_</div>
```

Caso queira remover mais de um filho, pode-se fazer um `for... ..of`

O `for... ..of` vai contar `div` por `div`, dentre todas as `divs` (nesse exemplo) e a cada elemento `div` encontrado (que seja filho de `<body>`), ele a remove

**ATENÇÃO:** Usa-se o `for`, pois o retorno do `.querySelectorAll` é um `Array`