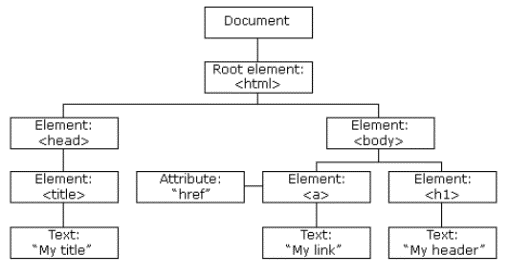
**Nodes**

É considerado um nó:

* O próprio **documento html**;
* Os **elemento HTML**;
* O **texto** fora dos elementos;
* Todo os **atributos** dos elementos;
* **Comentários**.

Todos os nodes podem acessados, alterados e criados através de JavaScript.

**Relacionamento entre os Nodes**

Todos os Nodes estão relacionados uns com outros como se fosse uma árvore genealógica, a **árvore de nó**.

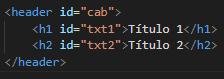
* **Node Root:** node que não derivida de nenhum outro node. É o node no top da node tree;
* **Node Parent:** node que origina outros nodes, exceto o node root;
* **Node child:** node que é derivado de outro. Um mesmo node pode ser parent e child ao mesmo tempo.;
* **Node sibling:** nodes que têm o mesmo node parent.

**Ex.**:

* **Elemento html:**
  + É o node root;
  + Não tem pai(node parent);
  + Tem dois filhos(node child): head e body;
  + Head é o primeiro filho(first node child);
  + Body é o último filho(last node child).
* **Elemento head:**
  + Tem um filho(node child): title;
* **Elemento title:**
  + Tem um filho(node child): o texto “DOM Tutorial”
* **Elemento body:**
  + Tem dois filhos(node child): h1 e p;
  + h1 e p são irmãos(node siblings).
* **Elemento h1:**
  + Tem um filho(node child): o texto “DOM Lesson one”;
* **Elemento p:**
  + Tem um filho(node child): o texto “Hello World”

**Propriedades dos Nodes**

Como cada node é um objeto, ele têm propriedades e métodos.



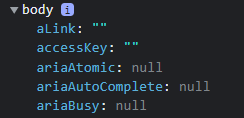
**Propriedades de navegação**

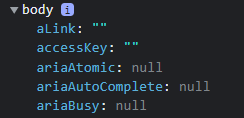


**parentNode:** pega o **node parent**.

**parentElement:** pega o **element parent node**.

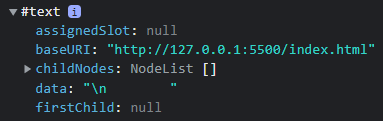
**Ex. 1:** o parentNode do element node header foi o body.



**Ex.:** 2 parentElement de header foi o elemento body.

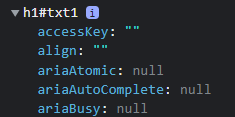
**firstChild:** retorna o primeiro child node

**firstElementChild:** retorna o primeiro elemento child de um node.

**Ex.1**:pega o primeiro o primeiro node child do elemento header

.

**Nota:** mesmo que tenha uma linha vazia entre um elemento e outro, será considerado com um text node vazio.

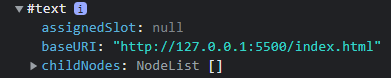
**Ex. 2:** pega o primeiro elemento child do elemento header

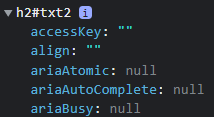


**lastChild:** retorna o último child node

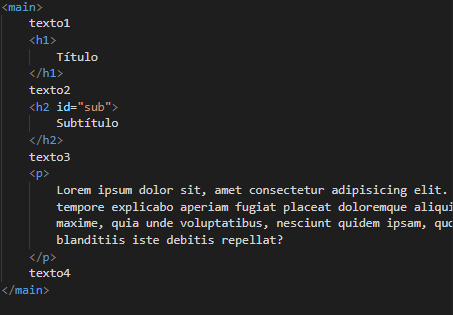
**lastElementChild:** retorna o último elemento child de um elemento.

**Ex. 1:** pega o último child node do elemento header.



**Ex. 2:** pega o último elemento child node do node header.

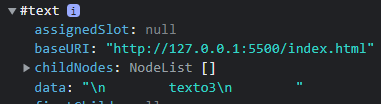


**Usando o exemplo HTML e JS abaixo:**

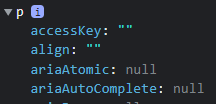


**nextSibling:** pega o próximo sibling node.

**nextSiblingElement:** pega o próximo element sibling.

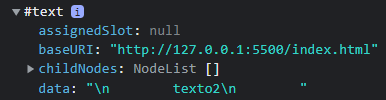
Ex. 1: pegando o próximo sibling node do elemento h2

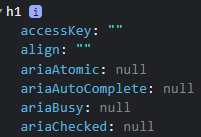


**Ex. 2:** pegando o próximo element sibling do elemento h2

**previousSibling:** retorna o sibling node anterior.

**previousElementSibling:** retorna o elemento sibling anterior.

**Ex. 1:** pega o sibling node anterior do elemento h2

**Ex.2:** pega o element sibling node anterior do node h2.

**nodeName:** retorna o nome do node. Onde:

* O nodeName de um elemento é o **nome da tag**;
* O nodeName de um texto é sempre **#text**;
* O nodeName de document é sempre **#document**;
* O nodeName de um atributo é **nome do atributo**.

Ex.:

**nodeValue:** acessa o valor de um node

* nodeValue para **element node é null**;
* nodeValue para text node é o **próprio texto**;
* nodeValue para atributos é o **valor do atributo**.

Ex.:

**nodeType:** retorna o tipo do node.

