Informática

CURSO TÉCNICO

Programação WEB

Aula 02: DOM

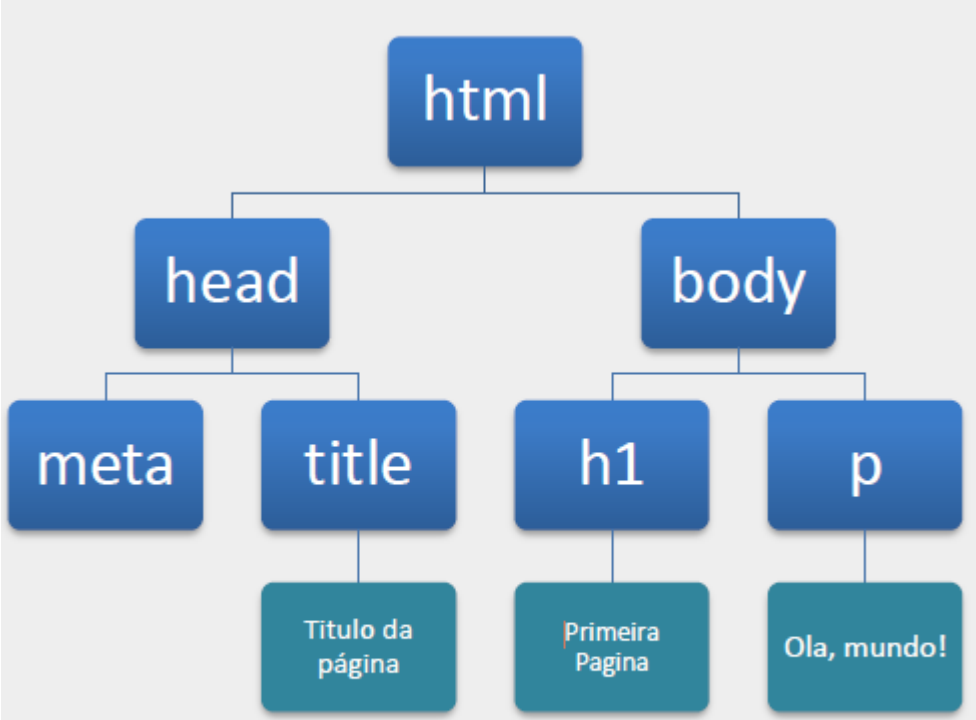
ESCOLA FUCAPI

Document Object Model

- Os navegadores criam árvore DOM da página
- Cada elemento é um nó
- Há uma relação de hierarquia entre os elementos. Eles podem ser filhos, pais, irmãos...
- Scripts acessam esses elementos através

Manipulação da Árvore DOM

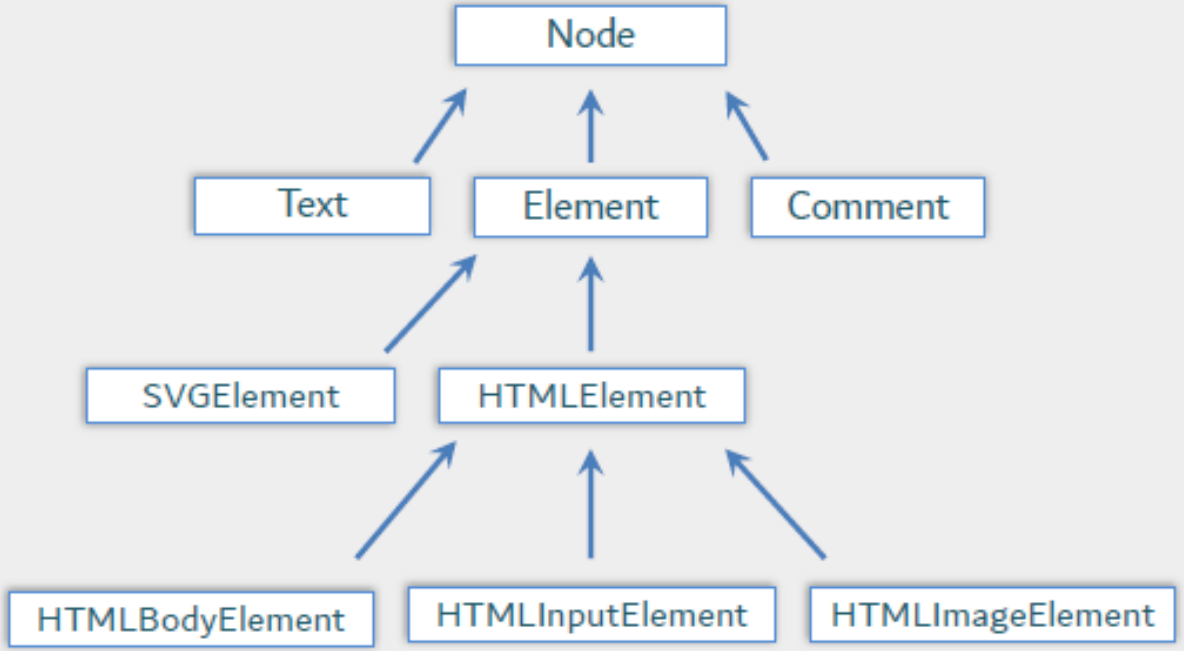
```html
<!DOCTYPE html>
<html lang="pt-BR">

  <head>
    <meta charset="UTF-8">
    <title>Titulo da Pagina</title>
  </head>

  <body>
    <h1>Primeira Pagina</h1>
    <p>Ola, mundo!</p>
  </body>

</html>
```



Abstração da Árvore DOM correspondente

Nota: Ao carregar uma página, o navegador percorre o respectivo código HTML e monta uma estrutura de dados internamente denominada **árvore DOM**, que é uma representação em memória de toda a estrutura do documento HTML. Nessa estrutura, cada elemento, comentário ou texto do documento HTML é representado como um objeto, denominado **nó**. A estrutura DOM é utilizada para manipular o documento HTML dinamicamente, utilizando programação, com a **DOM API** e a JavaScript.

Tipos de objetos na Árvore DOM



***Node**: classe abstrata com propriedades e métodos utilizados por todos os tipos de nós. Ex.: no.parentNode, no.nextSibling, no.childNodes*

***Element**: classe base para os elementos HTML da página. Fornece funcionalidades básicas a nível de elemento. Ex.: nextElementSibling, children, getElementById, getElementsByTagName, querySelector*

***HTMLInputElement**: fornece propriedade e métodos adicionais para elementos 'input', como value, patern, required, etc.*

Hierarquia de nós na estrutura DOM

Nó Root: nó representando o elemento raiz **<html>**

Nó Filho:nó representando um elemento diretamente dentro de outro

Nó Pai: nó representando o elemento que contém o nó filho

Nós Irmãos: nós representando elementos filhos do mesmo pai

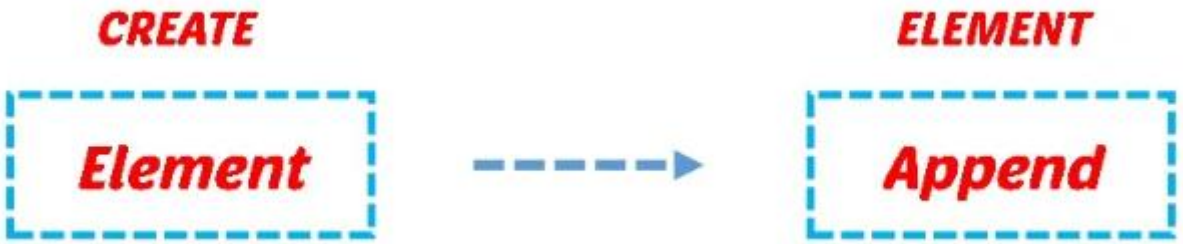
Manipulação Árvore DOM

- Criar novos elementos
- Selecionar elementos
- Atualizar elementos
- Remover Elementos

A manipulação da árvore DOM correspondente ao documento HTML é possível graças a uma Web API denominada DOM API, que pode ser utilizada pelo desenvolvedor por meio da linguagem JavaScript e do navegador de Internet.

ESCOLA FUCAPI

Criar Elementos



A criação de novos elementos HTML no DOM geralmente ocorre em dois passos:

- Criação do elemento
- Adicionar o elemento criado no DOM (em outro elemento)

Exemplo criação de elemento

```
<body></body>
<script>
    var el = document.createElement('h1')
    el.innerHTML = 'Criação de elemento'
    document.body.appendChild(el)
    console.log(el)
</script>
```

- O exemplo ao lado, cria um elemento <h1> pelo JS e adiciona no <body>

Sintaxe para criar elementos

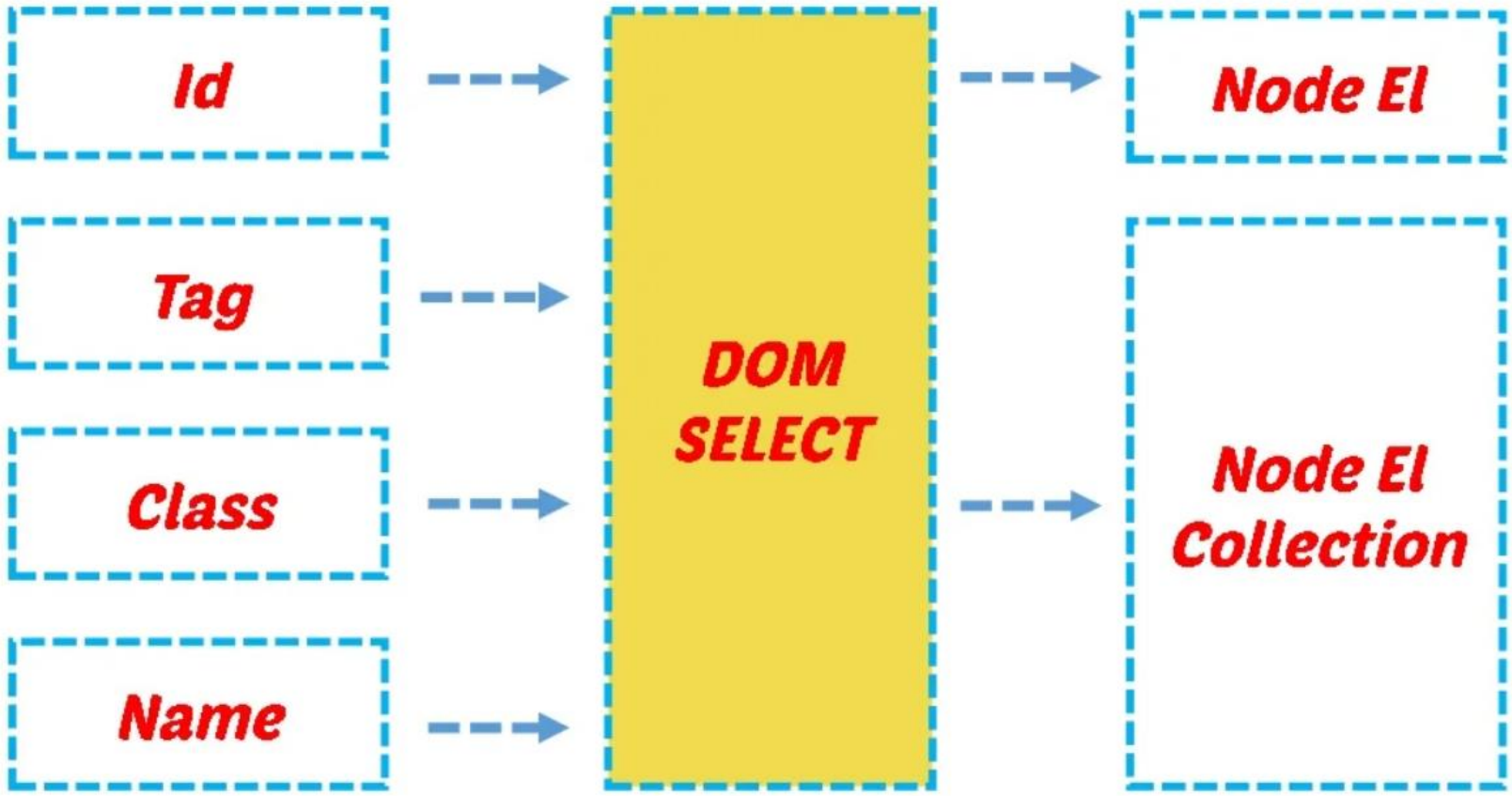
```
var el = document.createElement('new_el');
```

or Random DOM element

```
document.body.appendChild('new_el');
```

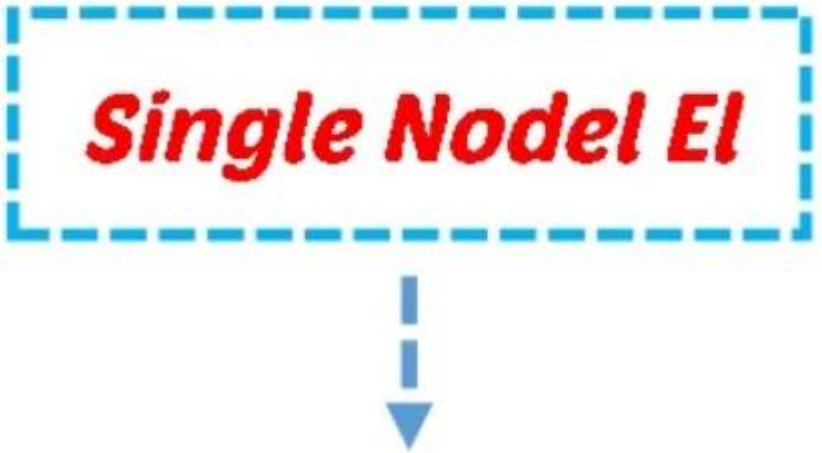
Selecionar Elementos

- Podemos selecionar elementos de várias maneiras:
  - ID
  - TAG
  - Classe
  - Nome



Dependendo do caso, cada forma de seleção retorna um nó, ou vários nós em forma de arrays(Node element Collection)

Selecionar por ID




```
var el = document.getElementById('el');
```

A seleção por ID retorna um único elemento, que tenha o id passado por parâmetro.

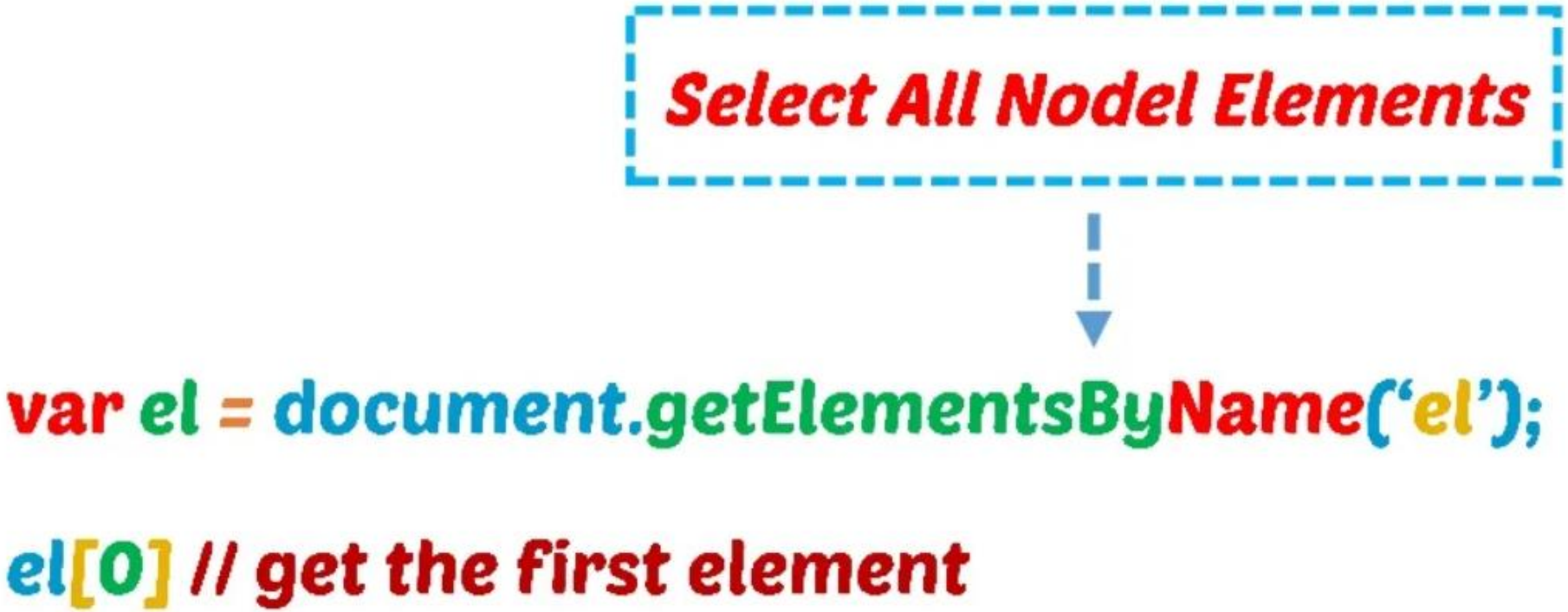
Selecionar por Classe

A seleção por classe retorna todos os elementos de uma determinada classe. Retorna um array onde é possível pegar cada elemento pelo índice.

Selecionar por Tag

A seleção por Tag retorna todos os elementos de uma determinada Tag. Retorna um array onde é possível pegar cada elemento pelo índice.

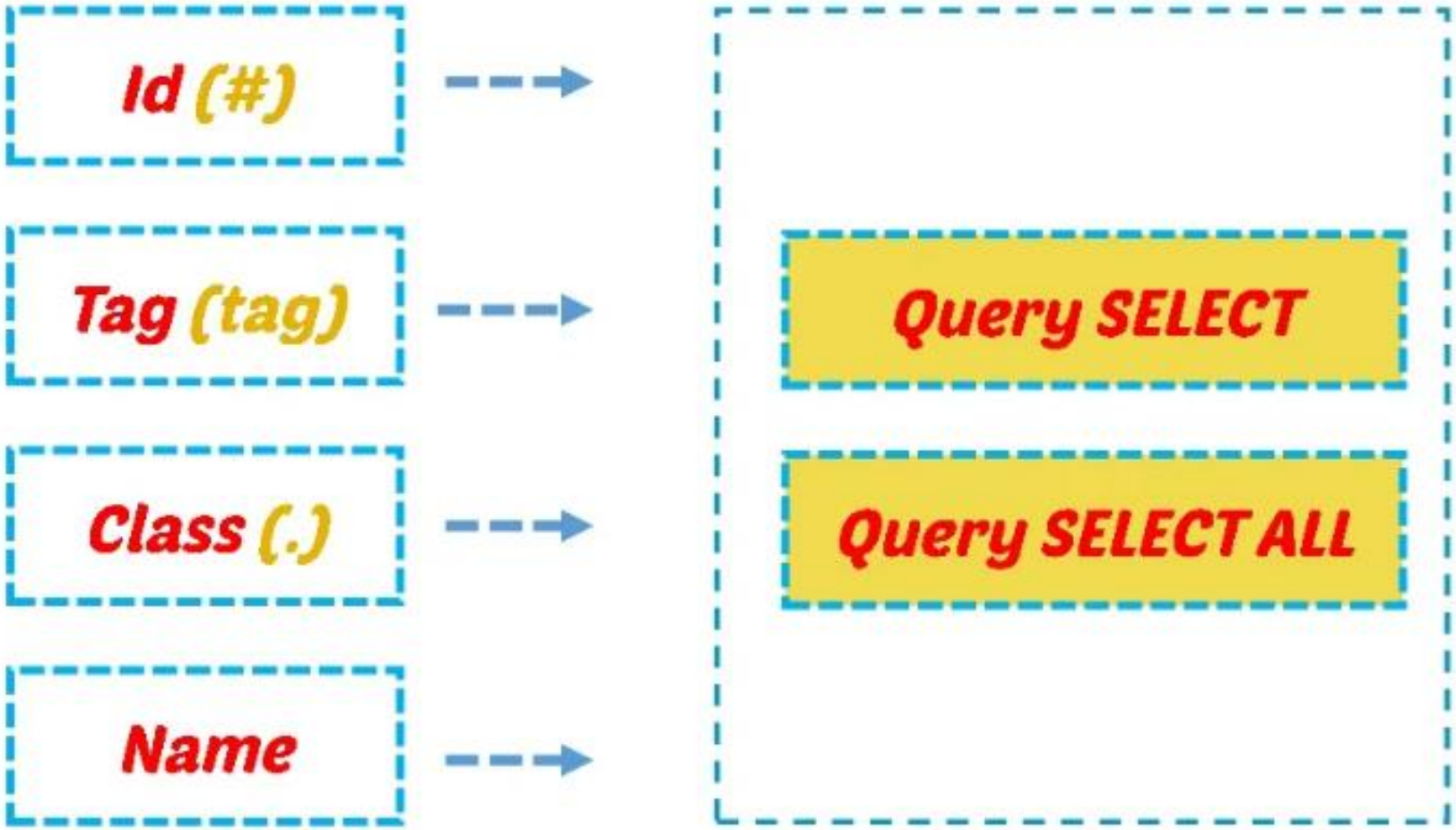
Selecionar por Name



A seleção pelo atributo name retorna todos os elementos que tem um determinado valor name. Retorna um array onde é possível pegar cada elemento pelo índice.

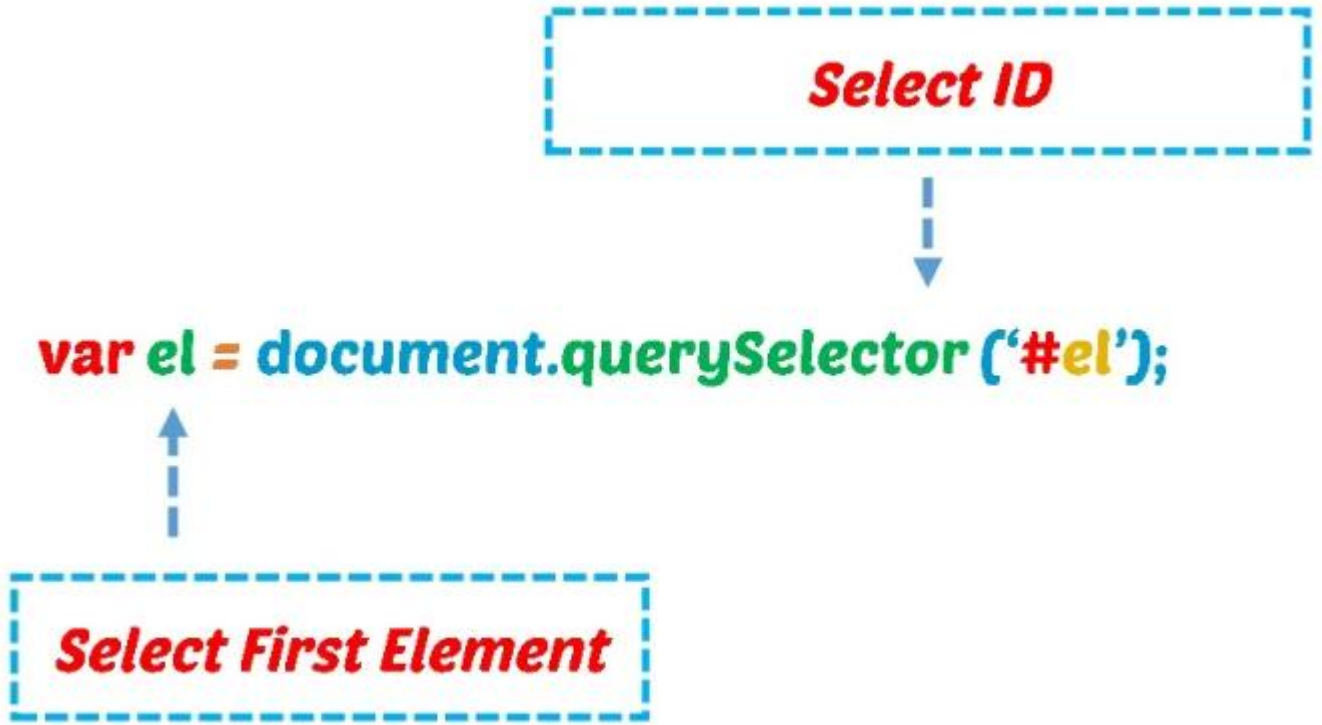
Selecionando por query

Também é possível selecionar elementos por query's CSS.

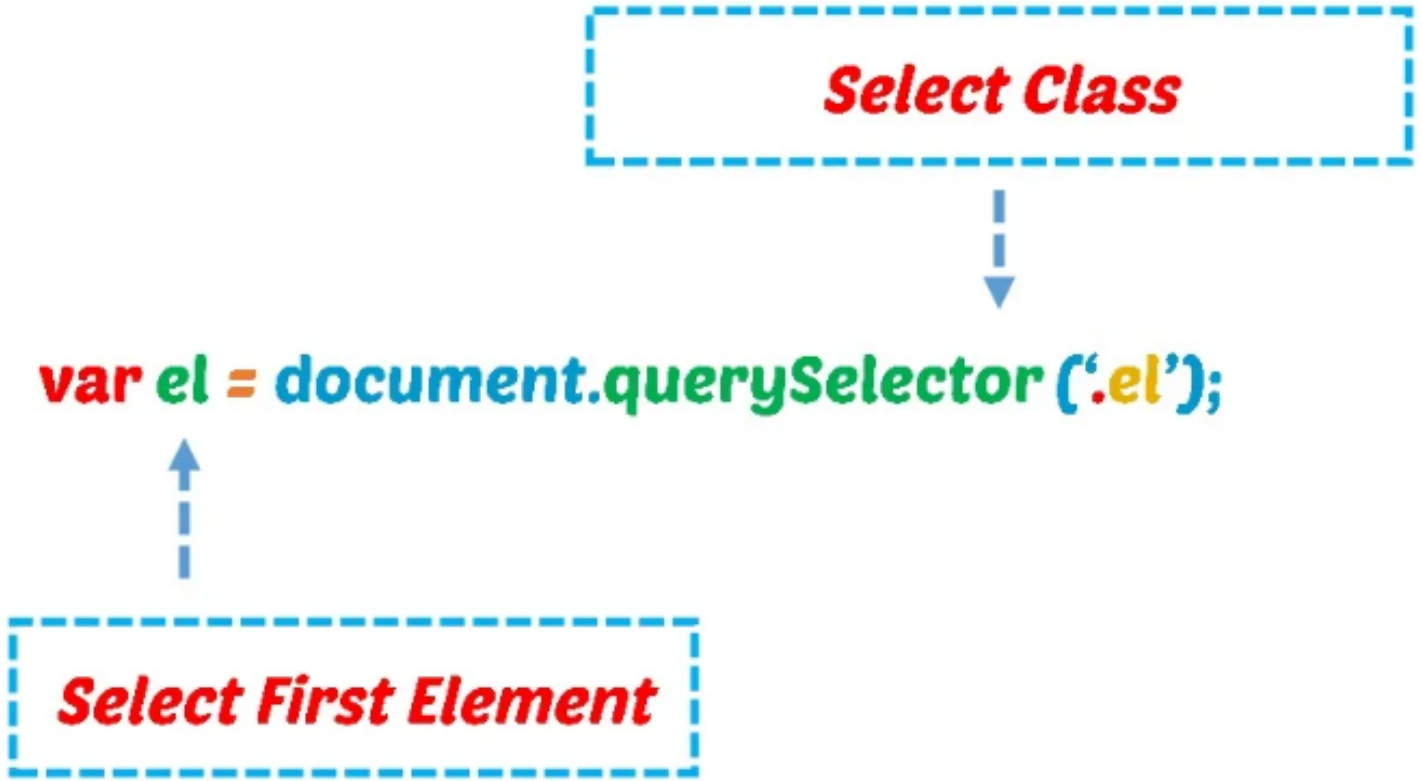


Selecionando por query::id

- O exemplo ao lado seleciona o elemento pelo id. Observe que esse é o mesmo seletor utilizado no CSS.
- A função querySelector() sempre retorna o primeiro elemento



Selecionando por query::class



Seleciona o primeiro elemento com determinada classe.

Selecionando por query::class

Neste caso, querySelectorAll(), seleciona TODOS os elementos da classe. Os elementos podem ser acessados pelos índices da coleção.

Atualizar Elemento

Para atualizar um elemento temos as seguintes formas

```
element.innerHTML = "HTML code";
```

Atualiza o conteúdo entre(inner) as tags do elemento. Podemos adicionar tags a um elemento dessa forma, ou com o appendChild()

```
element.getAttribute("attribute");

element.setAttribute("attribute");
```

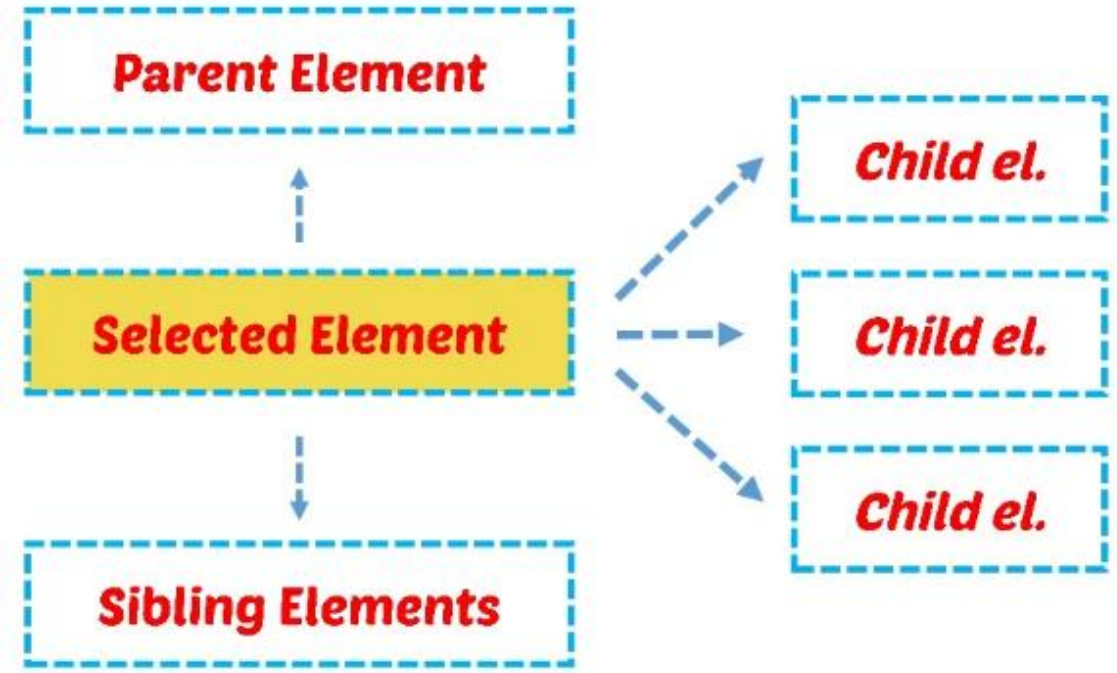
Podemos atualizar um determinado atributo do elemento HTML.

ESCOLA FUCAPI

Deletar elemento

A função removeChild() serve para remover um elemento filho. Para funcionar, devemos selecionar o pai e o filho

Pegar elementos por hierarquia



A partir de um elemento, podemos selecionar seu:

- Pai (Parent Element)
- Filhos (Child elements)
- Irmãos (Sibling elements)

Seletores por hierarquia

Selecionando o elemento PAI

```
var el = element.parentNode;
```

Selecionando os filhos

```
var el = element.children;
```

Selecionando os irmãos (próximo e anterior)

```
var el = element.nextElementSibling;
```

```
var el = element.previousElementSibling;
```

ESCOLA FUCAPI