



# Selenium Intermediário

Usando JavaScript.

*Autor: Reinaldo M. R. Junior*

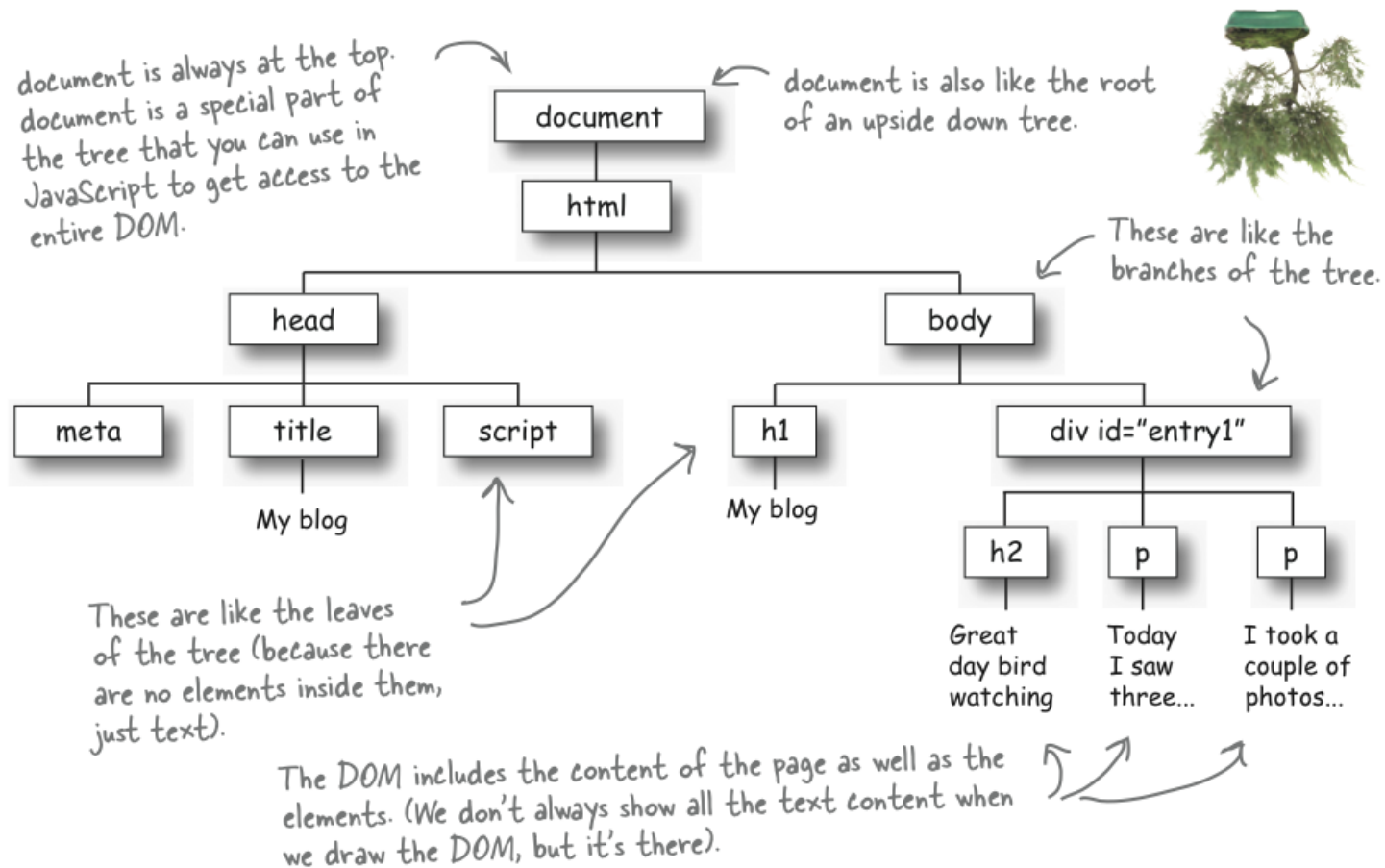
# JavaScript

- **JavaScript<sup>®</sup>** (abreviando **JS**) é uma linguagem leve, interpretada, mais conhecida como a linguagem de script para páginas Web, mas usada também em vários outros ambientes sem browser como node.js ou Apache CouchDB.
- JavaScript é uma linguagem de script multi-paradigma, baseada em protótipo que é dinâmica, e suporta estilos de programação orientado a objetos, imperativo e funcional.

# Document Object Model (DOM)

- O Document Object Model (DOM) é uma interface de programação para documentos HTML, XML e SVG. Ele fornece uma representação estruturada do documento como uma árvore. O DOM define métodos que permitem o acesso à árvore, para que eles possam alterar a estrutura do documento, estilo e conteúdo.
- DOM seja frequentemente acessado usando JavaScript, ele não faz parte da linguagem JavaScript. Ele também pode ser acessado por outras linguagens.

# Document Object Model (DOM)



# Procurando os elementos HTML:

Método	Descrição	Selenium
<code>document.getElementById('h3')</code>	Procura um elemento pelo atributo ID da Tag.	<i><code>find_element(By.ID, "h3")</code> <code>find_element_by_id('foo')</code></i>
<code>document.getElementsByTagName('input')</code>	Procura os elementos pela nome da tag.	<i><code>find_element_by_tag_name('input')</code></i>
<code>document.getElementsByClassName('h3')</code>	Procura os elementos pelo nome do atributo class.	<i><code>find_elements(By.CLASS_NAME, "h3")</code> <code>find_elements_by_class_name('foo')</code></i>
<code>Document.getElementsByName("animal")</code>	Procura os elementos pelo atributo name da tag.	<i><code>find_element_by_name('foo')</code></i>

# Procurando os elementos HTML via JavaScript:

Método	Descrição	Selenium
<code>document.querySelector("#id_password");</code> <code>document.querySelectorAll("input");</code>	Procura um elemento por CSS selector.	<i>find_element_by_css_selector(".example")</i> <i>find_elements_by_css_selector(".example")</i>
<code>document.elementFromPoint(x, y)</code> <code>document.elementsFromPoint(x, y)</code>	Procura o elemento pelas coordenadas x e y.	
Jquery Busca: <code>\$("CSS_Selector").next().children().find("CSS_Selector")</code>	Procura o elemento filho apartir do segundo nó do elemento Pai.	
Jquery Busca: <code>\$("CSS_Selector").children().find("CSS_Selector")</code>	Procura o elemento filho, você passa o Pai e o Filho.	
<code>\$("button").position()</code> <code>\$("button").offset()</code>	Pega a posição do elemento.	

Encontra um elemento pelo atributo ID da Tag e muda a cor do elemento x.

Faça via  
Selenium

```
<script>  
function myFunction() {  
    var x = document.getElementById("demo");  
    x.style.color = "red";  
}  
</script>
```

[Exemplo getElementById](#)

Procura os elementos pelo atributo Name da Tag  
e inseri em outra tag.

Faça via  
Selenium

```
<script>  
function myFunction() {  
    var x =  
        document.getElementsByName("animal").length;  
        document.getElementById("demo").innerHTML = x;  
}  
</script>
```

[Exemplo getElementByName](#)



# Procura os elementos pelo atributo Class da Tag e altera a cor para vermelho.



Faça via  
Selenium

```
<script>  
function myFunction() {  
    var x = document.getElementsByClassName("example");  
    var i;  
    for (i = 0; i < x.length; i++) {  
        x[i].style.backgroundColor = "red";  
    }  
}  
</script>
```

[Exemplo getElementByClassName](#)

# Procura os elementos pelo Nome da Tag e altera seu texto.

Faça via  
Selenium

```
<script>
function myFunction() {
    document.getElementsByTagName("P")[0].innerHTML = "Hello
    World!";
}
</script>
```

[Exemplo getElementsByTagName](#)

# Selenium função `execute_script()`

- **`execute_script`** : De forma *Sincrona* Executa o JavaScript na janela / frame atual.

- **Como usar:**

```
driver.execute_script("""  
                        Conteúdo via javascript;  
                        """)
```

- **Como fazer o Click via JavaScript DOM e JQuery:**

```
driver.execute_script("""  
    document.getElementById("datetimepicker").click();  
    document.querySelector("#datetimepicker").click()  
    $("#datetimepicker").click();  
    """)
```

# Selenium função `execute_script()`

- **`driver.execute_script ()`**: De forma *Sincrona Executa o JavaScript na janela / frame atual.*

- ***Exemplo:***

```
driver.execute_script('return document.title')  
driver.execute_script('return document.URL')  
driver.execute_script("""  
    return document.getElementsByTagName("P")[0].innerHTML  
""")
```

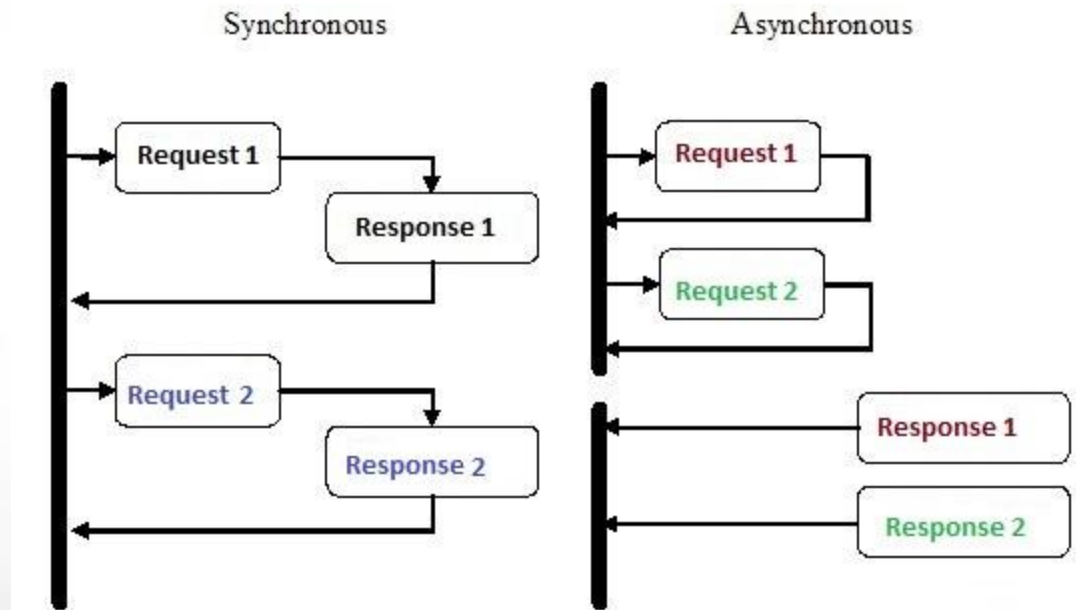
- ***Traz o título da página atual.***
- ***Traz o URL da página atual.***
- ***Traz o conteúdo da primeira tag <p>.***

# Selenium função `execute_async_script()`

- **`driver.execute_async_script()`**: De forma *assíncrona* Executa o JavaScript na janela / frame atual.
- **Exemplo:**

```
driver.execute_async_script('return document.title')
driver.execute_async_script('return document.URL')
driver.execute_async_script("""
    return document.getElementsByTagName("P")[0].innerHTML
""")
```
- **Traz o título da página atual.**
- **Traz o URL da página atual.**
- **Traz o conteúdo da primeira tag `<p>`.**

# Synchronous e Asynchronous

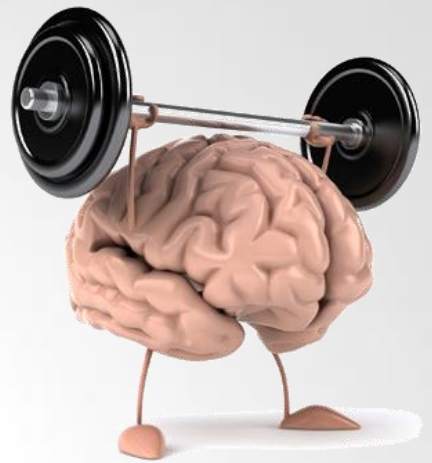


# Exemplos Avançados

Quando não é possível realizar uma tarefa via Selenium, como no exemplo abaixo temos um teclado digital, o Selenium não consegue move para o elemento, neste caso temos que mapear a posição das teclas e enviar via JavaScript ou via Windows.

- [https://github.com/reinaldorossetti/tests\\_solution/blob/master/teclado\\_digital\\_via\\_javascript](https://github.com/reinaldorossetti/tests_solution/blob/master/teclado_digital_via_javascript)
- [https://github.com/reinaldorossetti/tests\\_solution/blob/master/teclado\\_digital\\_via\\_windows.py](https://github.com/reinaldorossetti/tests_solution/blob/master/teclado_digital_via_windows.py)

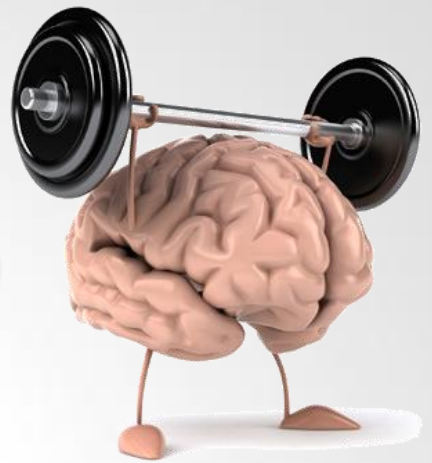
# Desafio - Fácil



- Dado o link abaixo, selecione as datas do componente Linked Pickers, são a data inicial e data final, selecione datas diferentes usando javascript.
- <http://eonasdan.github.io/bootstrap-datetimepicker/>



# Desafio - Intermediário



- Dado o link abaixo, selecione as datas do componente Linked Pickers, faça os campos aparecer via function show do próprio componente e depois der o click em cada data tudo via javascript.
- <http://eonasdan.github.io/bootstrap-datetimepicker/>
- <http://eonasdan.github.io/bootstrap-datetimepicker/Functions/#enable>

# Referências:

- [0] [http://www.w3schools.com/js/js\\_htmlDOM\\_document.asp](http://www.w3schools.com/js/js_htmlDOM_document.asp)
- [1] <https://cs.wellesley.edu/~cs110/lectures/L11/dom.html>
- [2] <https://dom.spec.whatwg.org/#interface-document>
- [3] <http://www.w3schools.com/js/>
- [4] <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>
- [5] [http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery\\_hide\\_class](http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_hide_class)