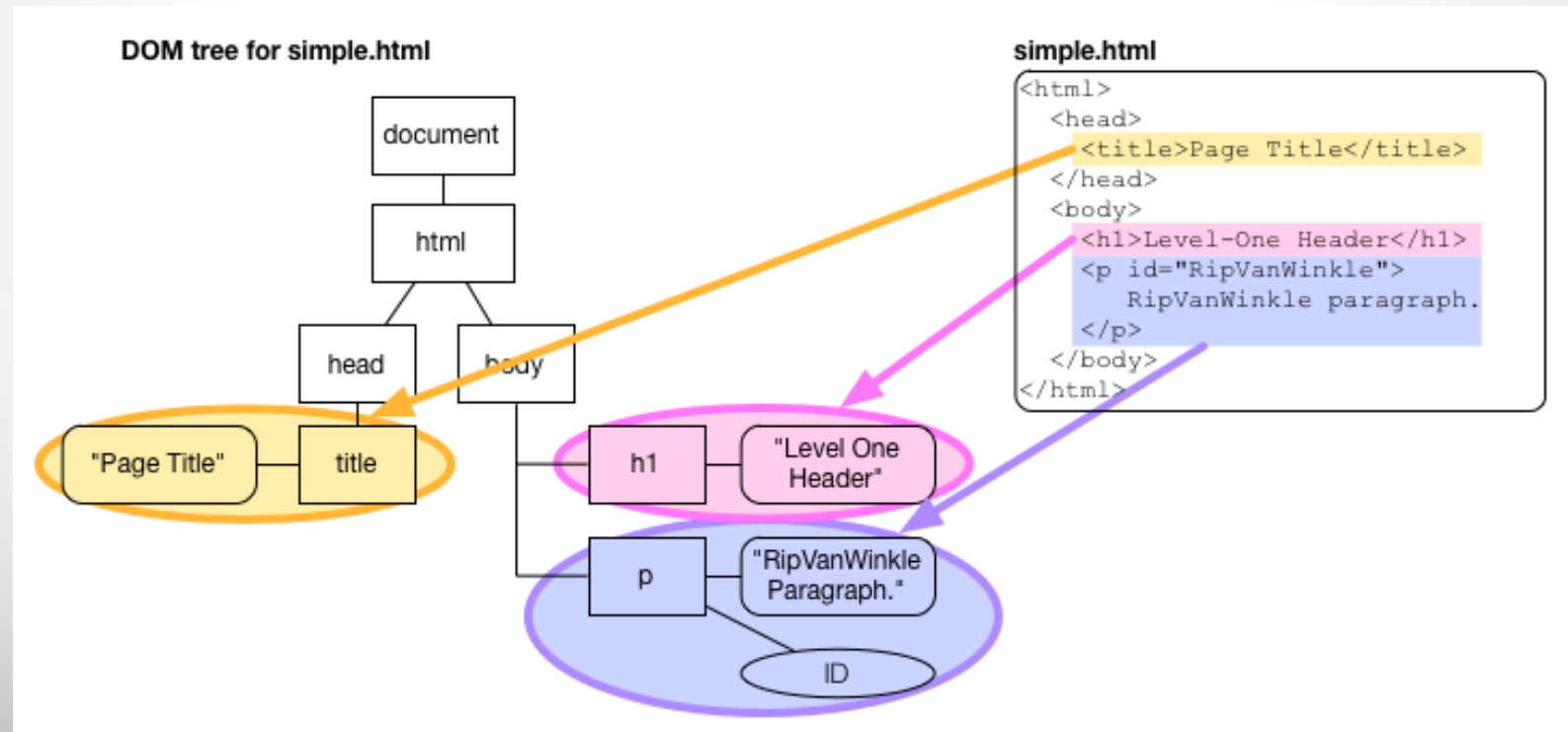


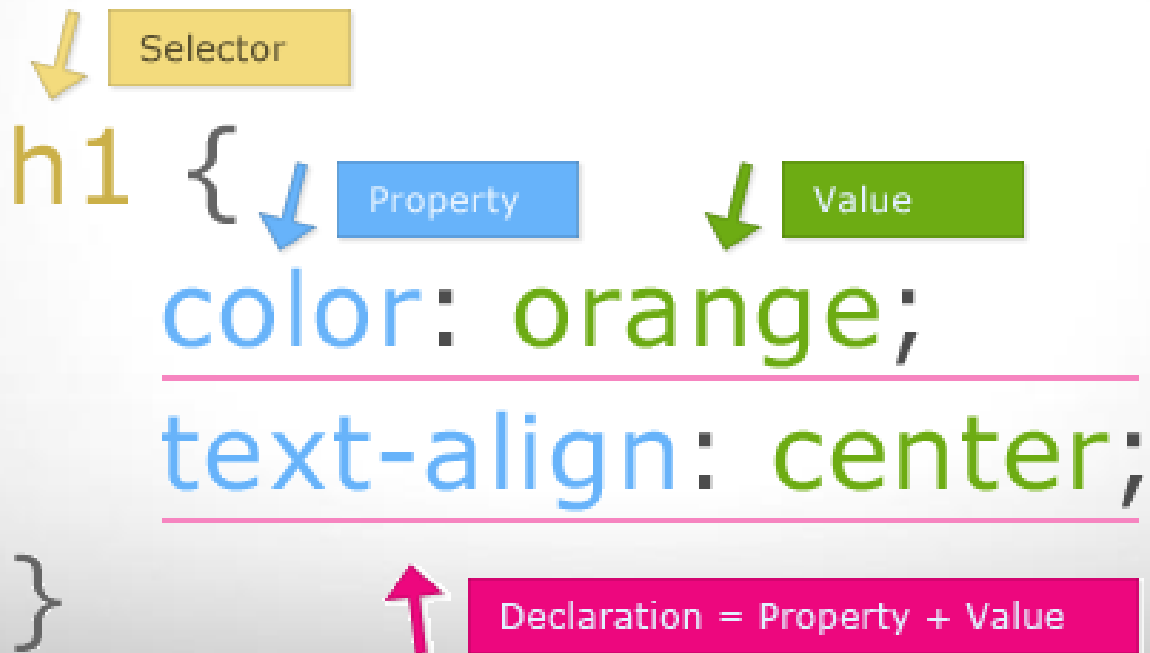
*Autor: Reinaldo M. R. Junior*

# HTML Page:



# Anatomia de um CSS.

## Anatomy of a CSS Rule



# **CSS ou Cascading Style Sheets**

Cascading Style Sheets (**CSS**) é uma linguagem de folhas de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML ou XML.

# Exemplo:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body {
      background-color: lightblue;
    }

    h1 {
      color: white;
      text-align: center;
    }

    p {
      font-family: verdana;
      font-size: 20px;
    }
  </style>
</head>
<body>

<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

# **CSS Selector**

**Em CSS, os selectors são padrões usados para selecionar os elementos para você aplicar o estilo. No entanto podemos usar o CSS no Selenium ou na DOM para selecionar um determinado elemento. Muitos autores do Selenium aconselham usar ele ao invés do xpath.**

# HTML <a>=tag / rel=Attribute

- <a rel="external" href="http://www.functravel.com/">Cheap Flights</a>

The screenshot shows a web browser window with a search page. The page has a dark blue header with the text "Vida de Testador..." and "Profissional da Área de Teste". Below the header is a search bar with a blue input field and a "Pesquisar" button. The browser's developer console is open, showing the HTML structure of the page. The HTML tree on the left highlights the search bar's structure, including a form, a table, and an input field. The CSS tree on the right shows the styles applied to the input field, including border, padding, and width. The input field's value is "q".

HTML structure (selected element):

```
<div class="region-inner tabs-inner">
  <div id="crosscol" class="tabs section" name="Entre colunas">
    <div id="BlogSearch1" class="widget BlogSearch" data-version="1">
      <h2 class="title">Pesquisar este blog</h2>
      <div class="widget-content">
        <div id="BlogSearch1_form">
          <form class="gsc-search-box" action="http://reinaldorossetti.blogspot.com.br/search">
            <table class="gsc-search-box" cellspacing="0" cellpadding="0">
              <tbody>
                <tr>
                  <td class="gsc-input">
                    <input class="gsc-input" autocomplete="off" name="q" size="10" title="search" value="" type="text">
                  <td class="gsc-search-button">
                    <input class="gsc-search-button" title="search" value="Pesquisar" type="submit">
                </tr>
              </tbody>
            </table>
          </form>
        </div>
      </div>
    </div>
  </div>
</div>
```

CSS styles (selected element):

```
input.gsc-input {
  border: 1px solid #ddd;
  padding: 1px 6px;
  width: 99%;
}
```

Herdeiro de table.gsc-search-box:

```
table.gsc-search-box {
  border-spacing: 0;
}
```

Herdeiro de form.gsc-search-box search:

```
form.gsc-search-box {
  font-size: 13px;
}
```

Herdeiro de div#BlogSearch1.widget.BlogSearch:

```
.widget {
  line-height: 1.4;
}
```

Herdeiro de div.content-outer:

```
.content-outer {
  font-size: 90%;
}
```

Selector	Exemplo de Uso	Descrição	Selenium Exemplo
Class = <b>Ponto</b>	.intro	Seleciona todos os elementos com class="intro".	find_element(By.CSS_SELECTOR, ".intro")
Id = <b>Hashtag</b>	#firstname	Seleciona todos os elementos com id="firstname".	find_element(By.CSS_SELECTOR, "#firstname")
Tudo = <b>Asterisco</b>	*	Seleciona todos os elementos.	find_element(By.CSS_SELECTOR, "*")
Elemento	p	Seleciona todos os elementos <p>.	find_element(By.CSS_SELECTOR, "p")
Virgula ,	div, p	Seleciona todas as <div> e todos os elementos <p>.	find_element(By.CSS_SELECTOR, "div, p")
Espaço entre elementos	div p	Seleciona todos elementos <p> dentro da <div>.	find_element(By.CSS_SELECTOR, "div p")
Sinal de Maior >	div > p	Seleciona todos elementos <p> que são parentes da <div>.	find_element(By.CSS_SELECTOR, "div > p")
Sinal de Soma +	div + p	Seleciona todos os elementos <p> que estão após o elemento <div>.	find_element(By.CSS_SELECTOR, "div + p")
<b>Til</b> ou <b>Tilde</b> ~	p ~ ul	Seleciona cada elemento <ul> que é precedido por um elemento <p>.	find_element(By.CSS_SELECTOR, "p ~ ul")



Selector	Exemplo de Uso	Descrição	Selenium Exemplo
<a href="#">[attribute]</a>	[class]	Seleciona todos os elementos com o atributo class.	find_element(By.CSS_SELECTOR, "input[class]")
<a href="#">[attribute=value]</a>	[class="red"]	Seleciona todos os elementos com o atributo class cujo o valor é igual a red.	find_element(By.CSS_SELECTOR, "input[class='red']")
<a href="#">[attribute~=value]</a>	[title~=flower]	Seleciona todos os elementos com o atributo title que contém a palavra "flower".	find_element(By.CSS_SELECTOR, "input[class~='flower']")
<a href="#">[attribute^=value]</a>	a[href^="https"]	Seleciona cada elemento <a> cujo valor de atributo href começa com "https".	find_element(By.CSS_SELECTOR, "a[href^='https']")
<a href="#">[attribute\$=value]</a>	a[href\$=".pdf"]	Seleciona cada elemento <a> cujo valor de atributo href finaliza com ".pdf".	find_element(By.CSS_SELECTOR, "a[href\$='pdf']")
<a href="#">[attribute*=value]</a>	a[href*="w3schools"]	Seleciona cada elemento <a> cujo valor de atributo href contém a substring "w3schools".	find_element(By.CSS_SELECTOR, "a[href*='w3schools']")

Selector	Exemplo de Uso	Descrição	Selenium Exemplo
<a href="#"><u>:first-child</u></a>	p:first-child	Seleciona cada <p> elemento que é o primeiro filho do seu pai.	find_element(By.CSS_SELECTOR, "p:first-child")
<a href="#"><u>::first-line</u></a>	p::first-line	Seleciona a primeira linha de cada elemento <p>.	find_element(By.CSS_SELECTOR, "p::first-line")
<a href="#"><u>:first-of-type</u></a>	p:first-of-type	Seleciona cada <p> elemento que é o primeiro <p> elemento de seu pai.	find_element(By.CSS_SELECTOR, "a")[0]
<a href="#"><u>:last-child</u></a>	p:last-child	Seleciona cada <p> elemento que é o último filho do seu pai.	find_element(By.CSS_SELECTOR, "a")[-1]
<a href="#"><u>:not(selector)</u></a>	:not(p)	Seleciona cada elemento que não é um elemento <p>.	find_element(By.CSS_SELECTOR, "a :not(p)")
<a href="#"><u>:nth-child(n)</u></a>	p:nth-child(2)	Seleciona cada <p> elemento que é o segundo filho do seu pai. Esse geralmente é o mais usado pra selecionar elementos filhos.	find_element(By.CSS_SELECTOR, "p:nth-child(2)")
<a href="#"><u>:nth-of-type(n)</u></a>	p:nth-of-type(2)	Seleciona cada <p> elemento que é o segundo <p> elemento de seu pai	find_element(By.CSS_SELECTOR, "p:nth-of-type(2)")

Podemos combinar multiples atributos de uma Tag.

# [title=one][rel ^=external]

## Multiple Attribute Matches

Vital to note is that you can use **multiple attribute selectors** in the same selector, which requires all of them to match for the selector itself to match.

HTML

```
<h1 rel="handsome" title="Important note">Multiple Attributes</h1>
```

CSS

```
h1[rel="handsome"][title^="Important"] { color: red; }
```

<https://css-tricks.com/attribute-selectors/>

# Exemplo

Link da Vídeo Aula:

[https://jsfiddle.net/reinaldo\\_junior/8tmt61ye/5/](https://jsfiddle.net/reinaldo_junior/8tmt61ye/5/)

# Prática 01



Como você estão no Intermediário, meus parabéns por terem chegado até aqui.

No exemplo que mostra a figura ao lado, temos vários parágrafos com mesmo nome. Quero mude a cor do parágrafo One nas duas colunas para azul e mude somente o segundo two para vermelho, depois diminua o tamanho da imagem altura e largura para 10, vocês podem fazer pela DOM ou pelo Selenium com `execute_script` usando CSS Selector.

<https://jsfiddle.net/80e4tec0/>



# Desafio



**Podemos combinar vários atributos e tags ,  
queria que vocês fizessem a combinação  
para pegar a img que contém a substring  
“Stop”, mas eu desejo que dar o click no  
elemento href w3schools, ou seja combine  
estes dois elementos.**

```
<span class="caption"><a  
href="http://www.google.com">Google</a></span>  
<span class="caption"><a  
href="https://www.w3schools.com">Google</a></span>
```

- One
- Two
- Three

- One
- Two
- Three



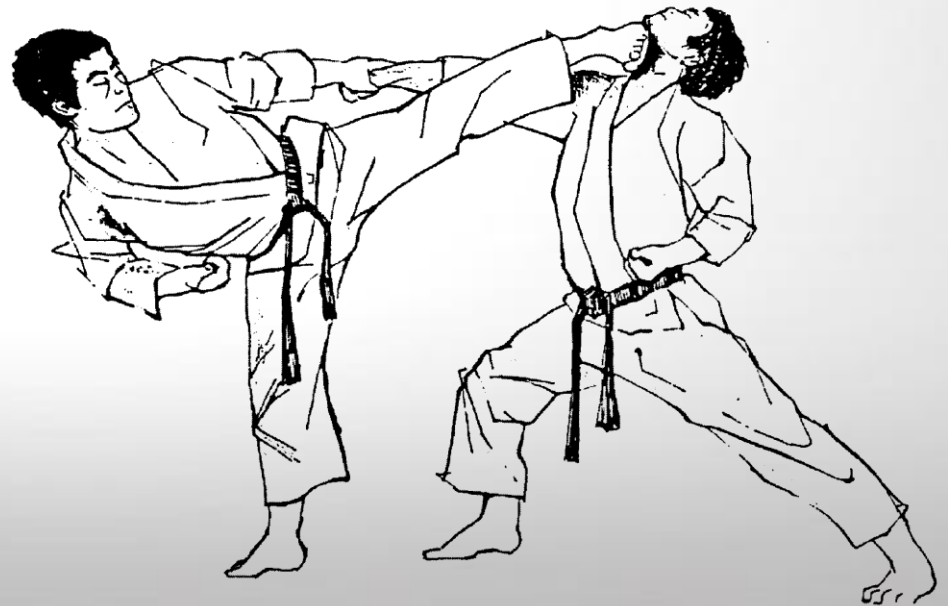
[Google](http://www.google.com)



[Google](https://www.w3schools.com)

# Dificuldades

- A maior dificuldade das pessoas é entender que na tela existe vários iframes, se você não consegue trazer o elemento de nenhuma forma, sem dúvida tem um iframe que você está esquecendo de selecionar.



# Siga sempre a Estratégia

## Melhor ordem:

- **ID;**
- **Class;**
- **CSS Selector;**
- Xpath.

## Pior Caso:

- Relacionar com Texto e Link.

## Locator Strategies

- Class
- *CSS selectors*
- ID
- Link Text
- Partial Link Text
- Tag Name
- *XPath*

Good locators are:

- unique
- descriptive
- unlikely to change

That rules a few of these out

Start with IDs and Classes

Use CSS or XPath (with care)



# O que não fazer!!!

```
<button id="defaultOverviewTable:j_id54" class="ui-button  
ui-widget ui-state-default ui-corner-all ui-button-text-  
only ui-state-hover" type="submit"  
name="defaultOverviewTable:j_id54" role="button" aria-  
disabled="false"> <span class="ui-button-text ui-  
c">New</span> </button>
```

## Pior Caso:

- Relacionar com Texto e Link.

```
driver.find_element_by_xpath('//button[text()='Ne  
w'])
```

Ele não sabe, mas qualquer hora o texto ou link muda e quebra a automação.

# Qual a sua Faixa em Automação?



**Faixa Branca:** Começando a caminha agora;

**Faixa Amarela:** Consegui fazer um ou duas aplicações com muito esforço;

**Faixa Vermelha:** Já consigo fazer automação sem problemas, mas não segue as boas práticas e não domina o Selenium;

**Laranja:** Tenho participado de vários projetos de automação Web, Mobile e Desktop, mas não domina nenhuma;

**Faixa Verde:** Domina bem automação Web, BDD com Behave, tem seguido as boas práticas, tem estudado bastante a linguagem de automação não somente Selenium;

**Faixa Roxa:** Tenho todas as anteriores, conhece todas as Plataformas e Domina uma e tem contribuído para a comunidade, ensinando outras pessoas;

**Faixa Marrom:** Tenho usado as melhores técnicas e domina duas linguagens de programação para automatizar e duas plataformas, sem nenhum sleep fixo no seu código, boas práticas está no sangue, facilmente as pessoas entender o seu código;

**Faixa Preta:** Faço todas anteriores, participo de trabalhos Internacionais de Automação, nas horas vagas se dedica a dar palestras open-source, e contribuído com novas APIs ou correções de bugs.

# Minha Faixa

- **Tenho todas as anteriores, conhece todas as Plataformas e Domina uma e tem contribuído para a comunidade, ensinando outras pessoas;**



- **Treine Bastante, Pratique, se cair não desista, ninguém cresce de faixa sem suor e muito Treino.**

# Referências:

- [0] <http://selenium-python.readthedocs.io/api.html>
- [1] [http://www.w3schools.com/tags/att\\_a\\_rel.asp](http://www.w3schools.com/tags/att_a_rel.asp)
- [2] [http://www.w3schools.com/cssref/css\\_selectors.asp](http://www.w3schools.com/cssref/css_selectors.asp)
- [3] [\[Webinar\] Mastering Test Automation: How To Use Selenium Successfully](#)