

Maujor

O DINOSSAURO DAS CSS

Seletores CSS3

Autor: [Roger Johansson](#)

URL do original: http://www.456bereastreet.../css_3_selectors_explained/

Título original: CSS3 selectors explained

Traduzido com autorização expressa do autor

Introdução

Nos meses de setembro e outubro de 2005 eu publiquei uma série de artigos explicando os seletores previstos nas CSS 2.1. Em geral podemos afirmar que a maioria dos seletores descritos naqueles artigos podem ser usados *atualmente* nos navegadores mais modernos como Mozilla/Firefox, Safari, e Ópera. Temos que esperar ainda mais um pouco pelo Internet Explorer antes de podermos usar os seletores CSS 2.1 em toda sua plenitude. A boa notícia é que o Internet Explorer versão 7 *suportará* ainda que em parte tais seletores.

Dando uma olhada adiante podemos constatar que existem seletores bem mais poderosos previstos nas CSS 3 esperando para serem implementados e usados. Muitos dos seletores CSS 3 já foram implementados e são suportados em navegadores modernos, mas em geral o suporte ainda é muito precário para que os desenvolvedores possam usar tais seletores. Contudo, há casos onde eles podem ser empregados para adicionar futuras melhorias em funcionalidades e assim sendo eu acredito que poderá ser útil dar uma olhada no funcionamento dos seletores CSS 3.

Este artigo baseia-se nas especificações [Selectors W3C Working Draft 15 December 2005](#) (NT: *Rascunho de Trabalho do W3C para Seletores de 15 de dezembro de 2005*) . Os novos seletores descritos naquele documento integram as CSS nível 3 mas, poderão ser usados também por outras linguagens. Se você estiver lendo este artigo meses ou mesmo anos depois da data da publicação daquele documento é uma boa idéia conferir se existe uma versão mais recente.

Eu não vou explicar neste artigo os fundamentos e o funcionamento dos seletores CSS em geral. Se você precisar fazer uma revisão dos conceitos e funcionamento dos seletores consulte [Seletores CSS 2.1 - Parte 1](#).

Tabela de seletores

Para começar vamos a uma visão geral dos novos seletores CSS 3:

Visão geral da sintaxe para seletores CSS 3

Selector type	Pattern	Description
Seletor para sub string de atributo	E[att^="val"]	Casa com qualquer elemento E cujo valor do atributo att começa com "val".
Seletor para sub string de atributo	E[att\$="val"]	Casa com qualquer elemento E cujo valor do atributo att termina com "val".
Seletor para sub string de atributo	E[att*="val"]	Casa com qualquer elemento E cujo valor do atributo att contenha a sub string "val".
Pseudo classe estrutural	E:root	Casa com o elemento raiz do documento. Em HTML o elemento raiz é sempre o elemento HTML.
Pseudo classe estrutural	E:nth-child(n)	Casa com qualquer elemento E que seja o n-th (enésimo) filho do elemento pai.
Pseudo classe estrutural	E:nth-last-child(n)	Casa com qualquer elemento E que seja o n-th (enésimo) filho do elemento pai, a contar do último filho.
Pseudo classe estrutural	E:nth-of-type(n)	Casa com qualquer elemento E que seja o n-th (enésimo) elemento irmão do seu tipo.
Pseudo classe estrutural	E:nth-last-of-type(n)	Casa com qualquer elemento E que seja o n-th (enésimo) elemento irmão do seu tipo, a contar do último filho
Pseudo classe estrutural	E:last-child	Casa com qualquer elemento E que seja o último filho do elemento pai.
Pseudo classe estrutural	E:first-of-type	Casa com qualquer elemento E que seja o primeiro elemento irmão do seu tipo.
Pseudo classe estrutural	E:last-of-type	Casa com qualquer elemento E que seja o último elemento irmão do seu tipo.
Pseudo classe	E:only-child	Casa com qualquer elemento E que

estrutural		seja o único filho do elemento pai.
Pseudo classe estrutural	E:only-of-type	Casa com qualquer elemento E que seja o único elemento irmão do seu tipo.
Pseudo classe estrutural	E:empty	Casa com qualquer elemento E que não tenha filhos (incluindo os nós de texto).
Pseudo classe :target	E:target	Casa com um elemento E que seja o destino da URL.
Pseudo classe estado de elemento de UI	E:enabled	Casa com qualquer elemento E de interface de usuário (um controle de formulário) e que esteja habilitado (enabled).
Pseudo classe estado de elemento de UI	E:disabled	Casa com qualquer elemento E de interface de usuário (um controle de formulário) que esteja desabilitado (disabled).
Pseudo classe estado de elemento de UI	E:checked	Casa com qualquer elemento E de interface de usuário (um controle de formulário) que esteja marcado (checked).
Pseudo-elemento fragmentos de elemento de UI	E::selection	Casa com a parte de qualquer elemento E que tenha sido selecionado ou destacado pelo usuário.
Pseudo classe negação	E:not(s)	Casa com qualquer elemento E que não case com o seletor simples s.
Elemento de combinação irmão em geral	E ~ F	Casa com qualquer elemento F que seja precedido pelo elemento E.

Não se preocupe se a tabela acima parecer confusa. Cada um dos seletores listados será descrito com detalhes e exemplos de uso serão mostrados neste artigo.

Seletores para sub string de atributo

Este é um novo grupo de seletores que possibilita aos desenvolvedores usar sub strings de um atributo para casar o seletor com um elemento.

Suponha a seguinte estrutura de marcação HTML em um documento:

```
1. <div id="nav-primary"></div>
2. <div id="content-primary"></div>
3. <div id="content-secondary"></div>
4. <div id="tertiary-content"></div>
5. <div id="nav-secondary"></div>
```

Com o uso do seletor para sub strings de atributo você pode casar combinações das partes estruturais do documento.

A regra a seguir define uma cor de fundo para todos os elementos DIV que tenha seu nome de ID iniciado por "nav":

```
1. div[id^="nav"] { background:#ff0; }
```

Para este nosso exemplo o seletor casa com `div#nav-primary` e `div#nav-secondary`.

Para casar com os elementos DIV que tenham seu nome de ID terminado por "primary" você poderia usar a seguinte regra:

```
1. div[id$="primary"] { background:#ff0; }
```

Agora o seletor casa com `div#nav-primary` e `div#content-primary`.

A regra a seguir define uma cor de fundo para todos os elementos DIV que contenha no seu nome de ID a sub string "content":

```
1. div[id*="content"] { background:#ff0; }
```

Os elementos casados por esta regra são `div#content-primary`, `div#content-secondary`, and `div#tertiary-content`.

Os seletores para sub string de atributo são atualmente suportados pelas últimas versões dos navegadores Mozilla, Firefox, Flock, Camino, Safari, OmniWeb, e Opera, assim se eles não se destinarem ao Internet Explorer poderemos desde já usá-los.

A pseudo classe :target

Considere URLs com link para um fragmento identificador (um sinal tralha "#", seguido por um nome de âncora ou uma ID de um elemento) apontando para um determinado elemento dentro do próprio documento. O elemento para o qual a URL aponta é o destino (target) e a pseudo classe :target possibilita casar aquele elemento. Se a URL não contém um fragmento identificador a pseudo classe

:target não casa com qualquer elemento (*NT: não funciona*).

Considere a estrutura HTML mostrada no exemplo anterior. A regra a seguir coloca uma linha de destaque (outline) em volta da `div#content-primary` quando a URL contiver este fragmento identificador:

```
1. div#content-primary:target { outline:1px solid #300; }
```

Um exemplo de URL contendo o fragmento identificador mencionado é:

`http://www.example.com/index.html#content-primary`.

A pseudo classe :target é atualmente suportada por navegadores baseados em Mozilla e Safari.

Pseudo classe estado de elemento de UI

As pseudo classes :enabled e :disabled

As pseudo classes :enabled e :disabled permite aos desenvolvedores controlar a aparência de elementos da interface do usuário (controle de formulários) que estejam habilitados ou desabilitados (enabled ou disabled) em navegadores que permitem estilizar formulários. As seguintes regras definem cores de fundo distintas para inputs de textos dependendo de estarem habilitados ou desabilitados:

```
1. input[type="text"]:enabled { background:#ffc; }
2. input[type="text"]:disabled { background:#ddd; }
```

A pseudo classe :checked

A pseudo classe :checked permite aos desenvolvedores controlar a aparência dos elementos radio e elementos checkbox. Novamente, em navegadores que permitem estilizar formulários. A regra CSS a seguir define uma borda verde para os elementos radio e checkbox que estiverem marcados (checked):

```
1. input:checked { border:1px solid #090; }
```

A pseudo classe estado de elemento UI é atualmente suportada por navegadores Opera e navegadores baseados em Mozilla.

Covém notar que muitos navegadores limitam ou restringem aos desenvolvedores as possibilidades de introduzir modificações na aparência dos controles de formulários. Para saber mais sobre estas restrições leia meu artigo [Styling form controls](#) e [Styling even more form controls](#).

Pseudo classes estruturais

As pseudo classes estruturais permitem aos desenvolvedores casar elementos baseados em informações disponíveis na árvore do documento e que não possam ser casados por seletores simples ou seletores combinados. As pseudo classes estruturais são poderosas mas, infelizmente os navegadores atuais suportam algumas poucas destas pseudo classes.

A pseudo classe :root

A pseudo classe :root casa com o elemento raiz do documento. Em HTML o elemento raiz é sempre o elemento HTML. As regras de estilo mostradas a seguir são idênticas (bem, quase idênticas- :root tem uma especificidade maior que html):

```
1. :root { background:#ff0; }
2. html { background:#ff0; }
```

A pseudo classe :root é atualmente suportada por navegadores baseados em Mozilla e Safari.

A pseudo classe :nth-child()

A pseudo classe :nth-child() casa um elemento que tenha um certo número de elemento irmão anteriores na árvore do documento. O argumento a ser colocado dentro do parênteses, no seletor, pode ser um número, uma palavra-chave (keyword) ou uma fórmula.

Um **number** n casa o enésimo filho. A regra a seguir aplica-se a todos os parágrafos que sejam o terceiro filho de seu elemento pai:

```
1. p:nth-child(3) { color:#f00; }
```

As palavras-chave (**keywords**) *odd* e *even* podem ser usadas para casar elementos filhos cuja posição índice seja ímpar ou par. A posição índice do primeiro filho é 1. A regra a seguir casa qualquer elemento p que seja o primeiro, terceiro, quinto, etc. filho do seu elemento pai:

```
1. p:nth-child(odd) { color:#f00; }
```

A regra a seguir casa qualquer elemento p que seja o segundo, quarto, sexto, etc. filho do seu elemento pai:

```
1. p:nth-child(even) { color:#f00; }
```

A **fórmula** $an + b$ pode ser usada para criar repetições mais complexas. Na fórmula a representa o tamanho de um ciclo, n é um contador que começa em 0 (zero) e b representa um valor a ser somado para determinar a primeira interação. Todos os valores são inteiros. Fica mais fácil entender como

funciona este seletor quando examinamos alguns exemplos. Então, vamos aos exemplos.

A regra a seguir casa qualquer elemento p cuja posição índice seja um múltiplo de 3. Na primeira regra b é igual a (zero) e pode ser omitido, como foi feito na segunda regra:

```
1. p:nth-child(3n+0) { color:#f00; }
2. p:nth-child(3n) { color:#f00; }
```

O valor a ser somado para determinar a primeira interação pode ser usado para definir em qual filho a regra deve começar a ser aplicada. Se você tem uma tabela com 20 linhas e deseja que cada linha ímpar a partir da décima linha tenha uma cor de fundo diferente, você pode usar a seguinte regra:

```
1. tr:nth-child(2n+11) { background:#ff0; }
```

Uma vez que n começa em 0 (zero), o primeiro elemento tr a ser casado é o 11°. O seguinte é o 13°, a seguir o 15°. e assim por diante.

Para maiores detalhes ver a seção [:nth-child\(\) pseudo-class](#) das especificações para CSS 3.

E o suporte dos navegadores para estes utilíssimos seletores? Nada bom. Tudo que eu sei é que nenhum navegador suporta este ou qualquer outro seletor "nth". Por favor, corrija-me se eu estiver errado.

A pseudo classe :nth-last-child()

A pseudo classe :nth-last-child() funciona de maneira semelhante a pseudo classe :nth-child() exceto pelo fato de que ela casa um elemento que tenha um certo número de elemento irmão **posteriores** na árvore do documento. Em outras palavras, a contagem inicia-se no último filho e de trás para frente. A regra a seguir casa o penúltimo (segundo de trás para frente) elemento tr de uma tabela:

```
1. tr:nth-last-child(2) { background:#ff0; }
```

A pseudo classe :nth-last-child() não é suportada atualmente por qualquer navegador.

A pseudo classe :nth-of-type()

A pseudo classe :nth-of-type() funciona de maneira semelhante a pseudo classe :nth-child(), mas somente leva em consideração elementos do mesmo tipo do elemento ao qual a regra se aplica. A regra a seguir casa qualquer elemento p que seja o terceiro filho do seu elemento pai:

```
1. p:nth-of-type(3) { background:#ff0; }
```

Este seletor é útil quando você quer ter certeza que está casando o terceiro elemento p. À primeira vista você poderá pensar que não há diferença com o seletor nth-child, contudo :nth-child(3) considera todos os elementos irmãos na sua contagem e assim sendo, o resultado será diferente a menos que todos os elementos irmãos de p sejam também elementos p.

A pseudo classe :nth-of-type() não é suportada atualmente por qualquer navegador.

A pseudo classe :nth-last-of-type()

A pseudo classe :nth-last-of-type() casa um elemento que tem um certo número de elementos irmãos do mesmo tipo **após** na árvore do documento. Assim como a pseudo classe :nth-last-child() o começo da contagem é a partir do último filho e a contagem faz-se de trás para frente. A regra a seguir casa cada penúltimo elemento irmão do tipo p:

```
1. p:nth-last-of-type(2) { background:#ff0; }
```

A pseudo classe :nth-last-of-type() não é suportada atualmente por qualquer navegador.

A pseudo classe :last-child

A pseudo classe :last-child casa um elemento que é o último filho do seu elemento pai.

É o mesmo que :nth-last-child(1). A regra a seguir casa todos os elementos p que são o último filho do seu elemento pai:

```
1. p:last-child { background:#ff0; }
```

A pseudo classe :last-child funciona em navegadores baseados em Mozilla. Não é suportada pelo Ópera e apresenta bug no Safari (a regra acima casa todos os elementos p contidos no documento).

Surpreendentemente funciona no navegador OmniWeb (testado na versão 5.1.1), ainda que este navegador seja baseado no Safari. Isto pode ter sido causado por uma regressão na última versão do WebKit da Apple, uma vez que OmniWeb é usualmente lançado como uma versão antiga do WebKit que é usado pelo Safari.

A pseudo classe :first-of-type

A pseudo classe :first-of-type casa um elemento que é o primeiro irmão do seu tipo. É o mesmo que :nth-of-type(1).

```
1. p:first-of-type { background:#ff0; }
```


A pseudo classe `:first-of-type` não é suportada atualmente por qualquer navegador.

A pseudo classe `:last-of-type`

A pseudo classe `:last-of-type` casa um elemento que é o último irmão do seu tipo. É o mesmo que `:nth-last-of-type(1)`.

```
1. p:last-of-type { background:#ff0; }
```

A pseudo classe `:last-of-type` não é suportada atualmente por qualquer navegador.

A pseudo classe `:only-child`

A pseudo classe `:only-child` casa um elemento que é filho único de seu elemento pai. É o mesmo que (mas com um especificidade menor) `:first-child:last-child` ou `:nth-child(1):nth-last-child(1)`.

```
1. p:only-child { background:#ff0; }
```

A pseudo classe `:only-child` funciona em navegadores baseados em Mozilla. Safari interpreta este seletor de modo semelhante a `:first-child` (No Safari a regra acima casa todos os elementos p contidos no documento que sejam o primeiro filho de seu elemento pai).

A pseudo classe `:only-of-type`

A pseudo classe `:only-of-type` casa um elemento cujo elemento pai não tenha outro filho do mesmo tipo. *whose parent element has no other children of the same element type*. É o mesmo que (mas com um especificidade menor) `:first-of-type:last-of-type` ou `:nth-of-type(1):nth-last-of-type(1)`.

```
1. p:only-of-type { background:#ff0; }
```

A pseudo classe `:only-of-type` não é suportada atualmente por qualquer navegador.

A pseudo classe `:empty`

A pseudo classe `:empty` casa um elemento que não tenha filhos. Estão incluídos nós de textos e assim sendo, nos elementos a seguir somente o primeiro não tem filhos:

```
1. <p></p>  
2. <p>Text</p>  
3. <p><em></em></p>
```

A seguinte regra casa o primeiro elemento mostrado nos exemplos acima:

```
1. p:empty { background:#ff0; }
```

A pseudo classe :empty é atualmente suportada por navegadores baseados em Mozilla. Safari erroneamente aplica a regra para todos os elementos do tipo do elemento a casar.

A pseudo classe negação

A pseudo classe negação, escreve-se :not(s), usa um elemento simples como argumento. Casa todos os elementos que não sejam o elemento definido pelo simples seletor. Por exemplo: a regra a seguir casa todos os elementos que não sejam o elemento p:

```
1. :not(p) { border:1px solid #ccc; }
```

A pseudo classe negação é atualmente suportada por navegadores baseados em Mozilla e Safari.

O pseudo-elemento ::selection

O pseudo-elemento ::selection casa com a parte de qualquer elemento E que tenha sido selecionado ou destacado pelo usuário. Um possível uso para este seletor poderia ser o controle da aparência de um texto que tenha sido selecionado.

Algumas poucas propriedades CSS aplicam-se ao pseudo-elemento ::selection: color, background, cursor e outline.

A regra a seguir define a cor vermelha para uma seleção no documento:

```
1. ::selection { color:#f00; }
```

O pseudo-elemento ::selection funciona atualmente em navegadores baseados em Safari. O comportamento do Safari para este seletor é inconsistente e precisa de correções. Os navegadores baseados em Mozilla suportam este seletor se for usado o prefixo -moz- assim: ::-moz-selection. O prefixo será eventualmente removido para uso no Mozilla.

Elemento de combinação irmão em geral

O elemento de combinação irmão em geral consiste de dois seletores simples separados por um sinal de "til" (~). Este seletor casa ocorrências do segundo elemento seletor simples que sejam precedidos pelo

primeiro elemento seletor simples. Ambos os elementos devem ter o mesmo elemento pai, mas o segundo elemento não precisa seguir-se imediatamente após o primeiro. A regra a seguir casa elementos ul que são precedidos por um elemento p e que tenham o mesmo elemento pai:

```
1. p ~ ul { background:#ff0; }
```

O elemento de combinação irmão em geral é atualmente suportado pelo Ópera e por navegadores baseados em Mozilla.

Queremos melhor suporte pelos navegadores

Alguns seletores CSS3 são amplamente suportados. Infelizmente alguns dos mais úteis seletores ou não são bem suportados ou são suportados com limitações pelos navegadores atuais. Isto faz com que muitos dos seletores descritos neste artigo sejam mais ou menos inúteis para a web atual. Entretanto, não tenha medo de experimentá-los. Você poderá ainda usar os seletores que *são* suportados para acrescentar um realce extra para navegadores avançados.

Então, na *sua* opinião qual navegador deverá ser o primeiro a suportar a maioria ou mesmo todos os seletores CSS3? Safari, Firefox, ou Opera? Ou talvez, até mesmo o Internet Explorer?

Meu nome é Maurício Samy Silva e eu sou conhecido na Web como Maujor.

Mantenho o Site do Maujor hospedado em: maujor.com e meu e-mail de contato é maujorcss[ARROBA]maujor [PT]com.



Minha residência é em:

Copacabana - Rio de Janeiro,, RJ - - Brasil