jQuery – Uma biblioteca JavaScript para facilitar o desenvolvimento de interfaces web, escrevendo menos para obter resultados melhores

Alexandre Albano, Edson Chen, Filipe Salgado, Nilo Teixeira, Sérgio Lopes e Thadeu Russo

1. Introdução

Com a grande demanda para o desenvolvimento de aplicações web, a exigência de recursos javascript para se conseguir interfaces interativas e com boa usabilidade, alinhado com o problema de portabilidade entre *browsers*, tornam extremamente complicado o desenvolvimento de código portável em javascript.

Para resolver este problema, algumas bibliotecas de código javascript, que implementam funções comuns de uso no dia-a-dia de um desenvolvedor, encapsulando o problema da portabilidade, foram criadas. Destaque para a pioneira Prototype JS, YUI! (Yahoo), Dojo, Mootools e jQuery.

Neste texto iremos dar foco para o jQuery.

2. jQuery

Criada em 2006, por John Resig (Mozilla/Google), sob a licença MIT/GPL, o jQuery é a biblioteca js mais usada nos tempos atuais, pelo fato de ser elegante, prazerosa de se trabalhar e bem simples de se utilizar.

O gráfico a seguir mostra uma distribuição do uso de bibliotecas de javascript.

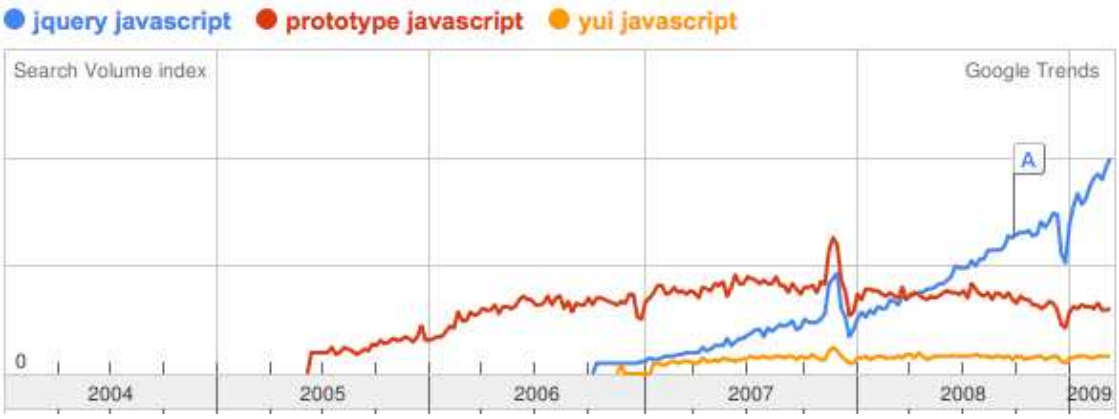

Figura 1 - Volume de buscas por biblioteca js

Pode-se notar pela figura 1 que até o início de 2007, prototype javascript reinava absoluto no mundo de bibliotecas javascript, tendo sido afetada em 2008.

A instalação é bem simples. Basta baixar o arquivo .js do site do jQuery (http://jquery.com), colocar o arquivo no mesmo diretório da página (ou em algum outro diretório desejado) e referenciar o arquivo na página como o exemplo a seguir:

```
<script type="text/javascript" src="jquery-1.3.2.min.js">
</script>
```

Atualmente, é recomendado que se use o arquivo javascript disponível nos servidores do google, no endereço http://ajax.googleapis.com/ajax/libs/jquery/1.3/jquery.min.js.

Uma das principais características do jQuery é a função **$**, esta que permite buscar os elementos na página html para que se possa aplicar funções neles. Por exemplo, imagine o seguinte trecho de página:

```
<h1>A função $ e a busca de elementos</h1>
```

Com o uso da função $, podemos esconder o trecho da seguinte maneira:

```
$('h1').hide()
```

O uso de chamadas encadeadas dão o tom da elegância que foi comentada anteriormente:

```
$('h1').slideUp('slow',
           function() {
              $(this).css('color','#FF0').slideDown('slow');
           }
        );
```

O código acima, escorrega o elemento h1 para cima e para baixo, mudando a cor do texto dentro do elemento para amarelo.

Um fator importante a se destacar, é a situação onde se deseja executar algo no momento em que a página é carregada pelo browser. O modo que isso é feito, é através de um método chamado após o evento onload do elemento body ser disparado. Este tipo de abordagem espera que a página inteira fique pronta, o que implica em aguardar o carregamento de imagens, anuncios, etc. Com o jQuery, pode-se “economizar” este tempo, pois é possível performar as ações necessárias assim que a estrutura DOM da página fique pronta. O exemplo abaixo mostra como:

```
$(document).ready(function() {
      $('h1').prepend('<span/>');
});
```

3. Referências adicionais

Mais referências, como tutoriais e documentação, além de exemplos, podem ser encontrados no site do jQuery:

http://docs.jquery.com/Main_Page

http://docs.jquery.com/How_jQuery_Works

http://docs.jquery.com/Core

http://docs.jquery.com/Downloading_jQuery