

1 O selector document ready

Cando falamos do JavaScript non intrusivo, estamos dicindo que o comportamento está separado da estrutura do documento. Polo tanto realizaremos operacións sobre os elementos da páxina dende fora da estrutura da mesma.

Para poder facer isto, necesitamos esperar a que todos os elementos DOM (Document Object Model - Modelo de Obxectos do Documento) na páxina estén completamente cargados, antes de comenzo a operar con eles.

Tradicionalmente facíase con evento onload do obxecto window de JavaScript. A sintaxis típica era:

```
window.onload = function() {
  $("table tr:nth-child(even)").addClass("colorido");
};
```

Este código facía que o JavaScript non se execute ata que a táboa esté completamente cargada.

Desafortunadamente co evento onload o navegador ralentiza a execución deste código JavaScript ata que o DOM esté completamente cargado, e ademais ata que tódalas imaxes e recursos externos tamén estén cargados.

Como resultado os visitantes notarán un certo retraso dende que ven a páxina ata que se executa o script. Incluso pode ser peor, se unha imaxe necesita moito tempo para cargar, teríamos que esperar ata que estivera completamente cargada para que o Javascript se execute, e isto faría que non fora moi enriquecedor o emprego de certas funcións visuais en Javascript.

Unha mellora desta técnica sería o ter que esperar simplemente a que o arbol DOM esté construído. Programar ésto para que funcione en varios navegadores é unha tarefa complexa que con jQuery pódese facer relativamente sinxelo, da seguinte forma:

```
$(document).ready(function()
{
  $("table tr:nth-child(even)").addClass("colorido");
}
);
```

O primeiro que facemos é encapsular o documento coa función jQuery(), e a continuación aplicámoslle o método ready() ó cal se lle pasa unha función que se executará cando o documento xa esté preparado para a súa manipulación.

A forma reducida do exemplo anterior sería:

```
$(function() {
  $("table tr:nth-child(even)").addClass("colorido");
});
```

Pasando unha función a \$(), estamos indicándolle ó navegador que espere ata que o DOM esté completamente cargado (pero soamente o DOM) antes de comenzo a executa-lo código. Incluso mellor, podemos usar esta técnica moitas veces dentro do documento HTML, e o navegador executará tódalas funcións indicadas no orden no que foron declaradas na páxina.

Polo contrario a función onLoad do Javascript soamente permite o uso dunha soa función.