

1 Conceptos crave de CSS

Para familiarizarse co comportamento de CSS é importante comprender os seus conceptos crave. Unhas ideas fundamentais serían as seguintes:

- **Estrutura e herdanza documentais.**

Un documento XHTML ten unha estrutura determinada, esta estrutura convértese nunha árbore xenealóxica cando se fai referencia á relación entre elementos. Así un elemento que está contido directamente noutro dise que é o "fillo" dese.

Por exemplo: *p* é fillo de *body* e *body* é pai de *p*. Os elementos que teñen o mesmo pai son "irmáns". A relación pai-fillo é fundamental para o funcionamento de CSS.

Un fillo pode "herdar" valores de propiedade do seu pai. Cunha boa planificación, a herdanza pode empregarse para facer mais eficaz a especificación de estilos. Este principio polo que algunhas regras ignoran outras achéganos un concepto moi importante: **a cascada**.

- **Regras de estilo en conflito: a "cascada".**

A "cascada" das follas de estilo en cascada refírese ó que ocorre si varias fontes de información de estilo coinciden polo control dos elementos dunha páxina. Cando un navegador atopa un elemento mira todas as declaracións de estilo que pode ter aplicadas e as ordena de acordo á orixe da folla de estilo, á especificidade dos selectores e á orde da regra para determinar cal aplicar.

- **Orixe da folla de estilo.**

Os navegadores dan distinto peso ás follas de estilo que listadas de menor a maior peso son:

- ◊ Follas de estilo do navegador.
- ◊ Follas de estilo do lector.
- ◊ Follas de estilo do autor da páxina web.
- ◊ Declaracións de estilo *!important* do lector.

Tras considerar a fonte da folla de estilo existe outra xerarquía de pesos que se aplica ás follas de estilo creadas polo autor da páxina web. A seguinte lista é importante e hai que ter en conta que as regras de estilo que están ó final da lista ignoran ás primeiras:

- ◊ Follas de estilo externas vinculadas (empregando o elemento *link*).
- ◊ Follas de estilo externas importadas (empregando o elemento *@import*).
- ◊ Follas de estilo incrustadas (có elemento *style*).
- ◊ Estilos en liña (empregando o atributo *style* nunha etiqueta de elemento).
- ◊ Declaracións de estilo marcadas como *!important*.

- **Especificidade do selector.**

Ata agora tivéronse en conta as distintas fontes da información do estilo. Cando se elixiu esa parte pode seguir existindo conflitos, por esa razón a "cascada" continúa a nivel de regras.

No seguinte código de exemplo hai dúas regras que fan referencia ó elemento *strong*.

```
strong {color: red;}
h1 strong {color: blue;}
```

Canto máis específico sexa o selector máis peso se lle dará para ignorar as declaracións en conflito. No exemplo anterior, todo o texto *strong* porase en cor bermello. Sen embargo, se o texto *strong* aparece nunha cabeceira de primeiro nivel (*h1*), será azul porque un elemento dun contexto determinado é máis específico e ten máis peso.

- **Orden das regras.**

Por último, cando unha folla de estilo contén varias regras en conflito de igual peso, a que estea en último lugar ten máis peso e ignora ó resto. Por exemplo, no seguinte exemplo, todas as cabeceiras de primeiro nivel do documento serían vermellas porque se impón sempre a última regra:

```
h1 {color: green;}
h1 {color: blue;}
h1 {color: red;}
```