

CSS: come i browser calcolano la specificità dei selettori

In questo articolo affronteremo un aspetto fondamentale dello standard CSS, ossia il calcolo della specificità dei selettori.

Si tratta di un argomento da cui dipende il successo o il fallimento nell'applicazione delle nostre regole di stile.

La *specificità* di un selettore si calcola come segue:

- si conti 1 se il selettore è un attributo `style`, altrimenti 0 (**=a**).
Nell'HTML, i valori dell'attributo `style` sono regole di stile, così avremo: `a=1`, `b=0`, `c=0`, `d=0`.
- si conti il numero di attributi `id` nel selettore (**=b**)
- si conti il numero di altri attributi e pseudo-classi nel selettore (**=c**)
- si conti il numero di nomi di elemento e pseudo-elementi nel selettore (**=d**)

Concatenando i quattro numeri (a, b, c, d) in serie, in un ordine di precedenza decrescente che va da sinistra a destra e confrontando quindi i numeri da sinistra a destra, otteniamo la specificità di un selettore. Esempi:

```
* { } /* a=0 b=0 c=0 d=0
      specificità = 0, 0, 0, 0 */
```

```
p { } /* a=0 b=0 c=0 d=1
      specificità = 0, 0, 0, 1 */
```

```
#id { } /* a=0 b=1 c=0 d=0
        specificità = 0, 1, 0, 0 */

style=" " /* a=1, b=0 c=0 d=0
           specificità = 1, 0, 0, 0 */
```

Si tenga presente che i valori posti più a sinistra vincono su quelli posti più a destra, a prescindere dal numero di valori. Quindi se anche l'elemento p avesse avuto quattro valori su d (per esempio con una concatenazione di selettori del discendente come `body div div p { }`) non sarebbe riuscito a vincere né su `#id` né su `style`, i quali hanno i loro valori posti più a sinistra superiori ai corrispondenti valori dell'elemento p ($a=1$ e $b=1$ per i primi, contro $a=0$ e $b=0$ per il secondo). L'attributo `style` ha una specificità superiore agli altri perché è l'unico ad avere il valore posto più a sinistra ($a=1$) superiore agli altri ($a=0$).