

Come rendere trasparenti gli elementi con CSS3?

La gestione della trasparenza è stata sempre una delle operazioni più noiose del web design, dato che non c'era un modo effettivo di applicarla agli elementi di un sito se non tramite immagini preparate ad hoc (filtri proprietari esclusi). Le specifiche CSS3 ci vengono incontro con le dichiarazioni `opacity` e `color: rgba` che permettono di specificare quanto deve essere opaco un elemento. Vediamo come si utilizzano.

La dichiarazione `opacity`

La dichiarazione `opacity` accetta come argomento un numero intero senza unità di misura: un valore di 1 sta per "elemento completamente opaco" e un valore di 0 sta per "elemento completamente trasparente". Ecco [un primo esempio](#) di utilizzo:

In questo esempio l'elemento `.opaco` viene reso in rosso pieno, l'elemento trasparente viene reso al 50% trasparente e l'elemento `.invisibile` è effettivamente invisibile all'occhio umano (anche se occupa sempre spazio nel flusso della pagina, come si vede dall'esempio).

`rgba`

Come detto in precedenza, lo stesso effetto si può ottenere con la dichiarazione `rgba`, impostando in un sol colpo la il colore e la trasparenza.

Qual è la differenza tra le due dichiarazioni?

A questo punto potrai chiederti quale sia la differenza tra i due approcci, e perché sono stati introdotti due metodi a prima vista simili per gestire la trasparenza degli elementi.

La differenza è che la regola `opacity` imposta il valore dell'opacità per **l'elemento stesso e tutti quelli contenuti**, mentre `rgba` viene **applicata solo all'elemento selezionato**. Puoi notare questa differenza nell'esempio: gli outline vengono resi trasparenti nel caso `opacity` mentre sono lasciati invariati nel caso `rgba`.

Possibili utilizzi?

Un ambito in cui spesso viene utilizzata la trasparenza è la realizzazione di didascalie trasparenti per slider di immagini.

[In questo esempio](#) il paragrafo contenente la didascalia ha un colore di sfondo nero, a cui viene applicata un'opacità del 40%. Se avessi usato opacity anche il testo sarebbe diventato trasparente.

[In quest'altro esempio](#) invece la trasparenza è stata utilizzata per far risaltare la differenza tra stato attivo e stato non attivo di un'immagine. Questo effetto prima doveva essere creato utilizzando più immagini (magari [utilizzando una sprite](#)) ed era un incubo da aggiornare (fare due copie di ogni immagine, per ogni variazione).

Qual è la compatibilità con i browser?

Il supporto sia per [opacity](#) che per [rgba](#) è abbastanza diffuso: IE 9+, Firefox, Chrome, Safari e Opera supportano entrambe le dichiarazioni. Anche nel mobile il supporto è buono.

Per quanto riguarda i browser più vecchi (IE 8 in primis), purtroppo ci si deve rifare ancora alle vecchie tecniche, però bisogna considerare caso per caso se ne vale la pena. Questi effetti CSS3 dovrebbero essere un ausilio all'usabilità del sito e non una parte necessaria, quindi si potrebbe anche decidere di fornire due esperienze diverse agli utenti con browser moderni e non.