

Come creare animazioni con le CSS Transitions?

Nell'[articolo precedente](#) abbiamo parlato dei CSS3 Transforms, come modo per modificare gli elementi della pagina. Oggi facciamo il primo passo verso la realizzazione di animazioni con CSS3 studiando il primo dei moduli che consentono di animare gli elementi: le CSS Transitions

Nella specifica CSS2 nei passaggi di stato non sono previsti stadi intermedi: una volta applicato l'*hover*, ad esempio, un elemento passa da uno stato all'altro immediatamente. I CSS3 forniscono un mezzo per modificare questo comportamento tramite il modulo *Transitions*

Stando alla definizione del W3C: *Le CSS Transitions consentono una variazione regolare dei valori delle proprietà CSS durante un intervallo di tempo definito*. Questo significa che è possibile passare da un valore all'altro non più tramite un salto istantaneo, ma definendo **una transizione dallo stato iniziale a quello finale**. Proviamo a fare un esempio e poi analizzeremo tutte le proprietà delle Transitions.

Un esempio: animare un link

Un effetto molto utilizzato nello sviluppo dei siti web è quello dell'*hover* sui link per segnalare all'utente che si tratta di un *elemento attivo*. Ad esempio si potrebbe avere qualcosa del genere:

In questo esempio abbiamo creato un link con sfondo azzurro, che all'*hover* diventa immediatamente blu scuro, come puoi vedere:

[Hover me!](#)

Proviamo ad aggiungere una transizione per rendere l'effetto più *dolce*:

Grazie a queste semplici dichiarazioni abbiamo specificato che la proprietà `background-color`, con valore iniziale `#69C` deve essere animata e che l'animazione deve durare 2 secondi. L'evento scatenante l'animazione è l'*hover*, mentre il valore finale della proprietà animata è `#036`. Il risultato è il seguente e sicuramente più gradevole rispetto al semplice *hover*.

[Transition me!](#)

Vediamo ora in dettaglio quali sono le varie proprietà delle Transition.

transition-property

Questa proprietà specifica quale proprietà (scusa il gioco di parole) dell'elemento deve essere animata. La sintassi è la seguente:

dove valore può essere *all* (indicando che tutte le possibili proprietà devono essere animate), *none* (l'opposto del precedente) oppure un elenco di proprietà CSS da animare, ad esempio:

In questo codice abbiamo impostato la dimensione del div a 200px e specificato che dovrà essere animata. Muovendo il mouse sul div vedrai che questo si comporta come se non fosse stata specificata nessuna transizione: questo perché la durata di default di un'animazione è 0s.

Hover me!

transition-duration

Questa proprietà stabilisce quanto deve durare l'animazione dal momento in cui viene scatenata. La sintassi è:

La durata può essere espressa in secondi (*s*) o millisecondi (*ms*) per un controllo più fine. Specificando 0 si ottiene un effetto identico all'hover.

Hover me!

transition-delay

Puoi ritardare la partenza di una transizione specificando questa proprietà. In questo modo è possibile avere delle transizioni posticipate di un tempo definito dopo l'attivazione:

Se ad esempio volessimo far partire la nostra animazione un quarto di secondo dopo l'evento scatenante, potremmo scrivere:

Hover me!

transition-timing-function

Questa proprietà permette di controllare il modo in cui l'elemento soggetto a transizione viene animato, modificando la velocità in vari punti dell'animazione. Le specifiche forniscono cinque modalità predefinite (ease, linear, ease-in, ease-out, ease-in-out), più la possibilità di creare delle funzioni personalizzate (cubic-bezier).

Ogni funzione interviene sulla velocità dell'animazione, simulando un comportamento più o meno reale: ad esempio la funzione linear fissa una velocità costante per tutta la durata dell'animazione, mentre ease-in-out parte lentamente, accelera nel mezzo e rallenta alla fine. Il mio consiglio è quello di provare i vari tipi di animazione (magari aumentando la durata, così da rendere l'effetto più evidente) e decidere quello che meglio si adatta al tuo lavoro.

Hover me!

CSS Transition: shorthand notation

Come per tutte le proprietà CSS, anche le Transition permettono di utilizzare la *shorthand notation*, che ci permette di abbreviare i nostri fogli di stile. La sintassi è la seguente:

Quindi le regole dell'esempio precedente verrebbero compattate tutte in un'unica dichiarazione:

Putroppo però non tutti i browser riconoscono la regola standard per le CSS Transition, quindi per avere il maggior supporto possibile è necessario inserire anche le **regole con i prefissi proprietari**:

Nota che la regola standard è l'ultima nell'elenco: questo, associato al fatto che i CSS tengono conto solo dell'ultima regola scritta, consente ai browser che supportano lo standard di utilizzare quest'ultima.

Animazioni multiple

Nell'esempio creato finora abbiamo cambiato la larghezza del nostro #box sfruttando le CSS Transition per rendere l'animazione fluida. E se volessimo animare più proprietà contemporaneamente? Ovviamente la specifica ci viene in aiuto: è possibile animare più proprietà semplicemente elencandole una dopo l'altra nella regola transition, separandole con delle virgole.

Ad esempio, se volessimo animare la dimensione del `#box` e contemporaneamente modificare il colore di sfondo avremmo qualcosa del genere:

Hover me!

Come puoi notare, le due *transition* sono completamente indipendenti, possiamo specificare la durata, la *timing function*, il ritardo senza che le caratteristiche dell'una influenzino l'altra. Se per caso volessimo che tutte le animazioni avessero le stesse proprietà potremmo utilizzare la parola chiave `all`, ottenendo:

Eventi scatenanti: come far attivare le Transition?

Finora abbiamo utilizzato solo lo stato *hover* per attivare le Transition, ma ovviamente non è l'unico modo.

Oltre ad `:hover`, puoi utilizzare ad esempio `:focus`, per attivare le Transition sui link oppure sugli elementi dei form. Ma il concetto fondamentale da capire è che **qualsiasi modifica alle proprietà specificate in transition, attiverà l'animazione**. Questo include anche modifiche realizzate tramite JavaScript: quindi invece di realizzare l'animazione utilizzando JavaScript possiamo semplicemente modificare una proprietà CSS o aggiungere una classe e **il browser si prenderà carico di animare l'elemento interessato**.

Conclusioni

Le CSS Transition sono uno strumento potente e flessibile per realizzare delle piccole animazioni da utilizzare nei nostri lavori. L'importante, come per tutte gli strumenti, è sapere quando utilizzarle, ma soprattutto quando non utilizzarle. Infatti è facile lasciarsi prendere la mano e applicare transizioni ad ogni elemento, facendo perdere il senso di quello che dovrebbe essere un modo per esaltare solo alcune parti del contenuto.

Tu hai già utilizzato le CSS Transition? Hai in mente qualche possibile utilizzo?