

Guida CSS3: pseudo classi e pseudo elementi

Bentrovato a questo nuovo articolo della guida CSS3! Come preannunciato nel [capitolo precedente](#), quest'oggi ci occuperemo di pseudo classi e pseudo elementi.

Ti sarà capitato spesso di voler applicare uno stile a degli elementi non proprio ben identificabili del tuo HTML, ad esempio agendo sull'aspetto dei link già visitati, sulla prima lettera di un paragrafo o ancora su tutti i titoli h1 che non hanno una specifica classe associata. Bene, le pseudo classi e gli pseudo elementi servono proprio a riferirci a queste specifiche porzioni di codice! Iniziamo il nostro percorso dalle pseudo classi associate ai link che, essendo anche presenti nella versione 2 del linguaggio CSS, risulteranno più familiari.

Pseudo classi per link — a

Le pseudo classi associate ai link, quindi al tag `a`, sono generalmente legate alle azioni che l'utente può compiere tramite esso:

- **a:visited** — modifica l'aspetto di un link già visitato dall'utente;
- **a:hover** — modifica l'aspetto di un link nel momento in cui l'utente posiziona il cursore su di esso;
- **a:active** — modifica l'aspetto di un link nel momento in cui l'utente clicca e tiene premuto il pulsante del mouse su di esso, ovvero in quella frazione di tempo in cui il link resta "schacciato";

Un semplice esempio potrebbe essere:

Pseudo elementi per paragrafi — p

I paragrafi hanno due specifici pseudo elementi che ci permettono di modificare la prima lettera e la prima linea per ricreare fedelmente effetti tipici della tipografia ai tempi in cui il web non esisteva e le parole restavano impresse su carta.

- **:first-line** — modifica l'aspetto della prima riga di un paragrafo;
- **:first-letter** — modifica l'aspetto della prima lettera di un paragrafo;

Combinando insieme questi due elementi, si possono ottenere effetti tipografici di impatto come:

Pseudo classi e pseudo elementi generali

Alcune pseudo classi e pseudo elementi sono anche più versatili e, oltre che ai link e ai paragrafi, possono essere quindi utilizzati in associazione a molteplici tag. Ad esempio:

- **:focus** — è una pseudo classe che agisce pressappoco come *:hover* modificando, non unicamente link, bensì campi di testo, input, text area e campi di password che vengono “selezionati” tramite click o tab dall’utente.
- **:before & :after** — sono due pseudo elementi che ci permettono di aggiungere del contenuto prima (*:before*) e dopo (*:after*) il tag selezionato. Ad esempio:
- **::selection** — permette di specificare uno stile per qualsiasi elemento selezionato dall’utente. Ad esempio potremmo voler indicare che del testo selezionato, normalmente segnalato in blu dai browser, venga indicato dal colore rosso.

Due cose vanno, però, specificate:

1. eh sì, *selection* è preceduto da `::`, non è un errore! Il CSS3, in realtà, differenzia le pseudo classi (singolo `:`) dagli pseudo attributi (doppio `::`), proprio grazie a questo stratagemma. Nei precedenti esempi (*:first-line* e *:first-letter*) per motivi di retro compatibilità è stata scelta la vecchia sintassi invece del `::` obbligatorio questa volta;

- alcuni browser non supportano *selection* (ad esempio Internet Explorer) mentre altri (come Mozilla Firefox) necessitano di uno specifico *vendor-prefix* come *::-moz-selection*;

Selezionare tramite attributi

Una delle novità più interessanti introdotte dal CSS3 è senz'altro la possibilità di selezionare tag specificando, all'interno di parentesi quadre, l'attributo ad esso associato. Ad esempio, potremmo voler modificare tutte le immagini che posseggono l'attributo *title* e scrivere quindi:

Oppure potremmo voler modificare tutti i tag *input* di tipo *text*:

Oppure ancora potremmo differenziare tutti i tag *a*, distinguendo tra quelli che rimandano a link esterni, e iniziano quindi per *http://*, e quelli interni. Per far ciò scriveremo:

Come puoi notare ho utilizzato il suffisso *^* per indicare di prendere in considerazione tutti gli attributi *href* che iniziano con *http://* o *https://*. Se avessi, invece, voluto modificare i link che terminavano con una particolare estensione avrei dovuto usare un altro suffisso: *\$*.

Infine potrei indicare tutti i link che, all'interno dell'attributo *href*, richiamano un determinato termine:

Tutti gli esempi citati ci permettono di ottenere questo effetto finale:

Selezionare tramite ereditarietà

Un altro utilissimo metodo per selezionare elementi “fantasma” è quello che si avvale dell’ereditarietà. Ad esempio, immaginiamo di avere un tag *body*, che contiene un tag *ul* che, come consuetudine, contiene a sua volta molteplici tag *li*:

Come possiamo applicare uno stile agli elementi *li* senza necessariamente dover ricorrere a id e classi specifiche? Attraverso questi semplici pseudo elementi:

- **:first-child** — ci permette di selezionare esclusivamente il primo elemento della lista;
- **:last-child** — ci permette di selezionare esclusivamente l’ultimo elemento della lista;
- **:nth-child** — ci permette di selezionare praticamente qualsiasi altro elemento della lista. Ad esempio con **:nth-child(odd)** selezioniamo gli elementi dispari, con **:nth-child(even)** quelli pari. Oppure, se volessimo aumentare la complessità, selezionando un elemento ogni tre partendo dal primo, useremo **:nth-child(3n+1)**, dove $3n$ rappresenta il terzo elemento della lista e $+1$ indica l’indice dal quale iniziare il conteggio;

Otterremo, quindi, facilmente un risultato di questo tipo:

Potremmo anche volerci riferire ai tag utilizzando altri “legami di parentela”, usando il concetto “fratello di” invece che “figlio di”. Ad esempio in una struttura:

i tag p non sono “figli di” $h2$ poiché non sono contenuti in esso. Possiamo utilizzare la sintassi $h2 + p$ e quindi il concetto “fratello di”, per selezionare il primo, e solo il primo, dei tag p .

Il selettore :not

L'ultimo selettore che tratteremo oggi è un selettore inverso, ovvero ci permette di riferirci ad un tag che non possiede un determinato id o classe. Ad esempio:

Conclusioni

Come consuetudine tramite [questo link](#) puoi scaricare l'intero codice sorgente contenente tutti gli esempi citati nel corso dell'articolo. Per qualsiasi dubbio, perplessità, appunto o suggerimento, ti rimando ai commenti: sarò lieto di risponderti!

Nel prossimo capitolo parleremo in modo più approfondito di ereditarietà. Non perdere l'appuntamento: ti aspetto!

GUIDA CSS3: INDICE LEZIONI

- 1) [Introduzione e selettori](#)
- 2) Pseudo classi e pseudo elementi
- 3) [Ereditarietà](#)
- 4) [I font - parte prima](#)