

## Vuoi imparare jQuery ma non sai da dove iniziare? Gli stili inline, gli effetti predefiniti



Nell'ultima lezione abbiamo modificato ulteriormente il nostro interruttore, dotandolo di un unico link per accendere e spegnere la lampadina, e abbiamo visto come applicare un semplice effetto, attraverso l'opportuna aggiunta di una classe all'elemento interessato. Ma è davvero necessario prevedere una classe CSS per ogni tipo di animazione che intendiamo realizzare?

### Come modificare gli stili "al volo"?

In tutti gli esempi realizzati finora, per applicare gli effetti abbiamo sfruttato unicamente le funzioni `addClass` e `removeClass`: jQuery si prendeva cura delle azioni, e tramite le classi CSS applicavamo le decorazioni all'elemento interessato. Da un certo punto di vista, è giusto che sia così, per la separazione fra azione e presentazione. Un uso intensivo di questa tecnica, tuttavia, potrebbe **appesantire notevolmente il foglio di stile**: ad esempio, se volessi cambiare il colore di un elemento, dovrei prevedere una classe base, con il colore predefinito, e una classe aggiuntiva con il colore desiderato.

Naturalmente, esiste un modo per gestire gli stili degli elementi attraverso jQuery, attraverso la funzione `css`. Questa funzione si applica a qualunque elemento selezionato, e può modificare qualsiasi proprietà CSS. La sintassi è la seguente:

Possiamo quindi modificare il codice JavaScript [dell'ultimo esempio](#) che passa dall'utilizzo della coppia `addClass/removeClass`:

all'utilizzo della funzione `css`:

Abbiamo ottenuto [lo stesso risultato](#) esclusivamente tramite jQuery, e in questo caso credo che sia più corretto così. Infatti l'accensione della lampadina è possibile solo con JavaScript attivato, quindi è giusto che tutta la gestione delle azioni relative a questo elemento siano delegate a JavaScript.

La funzione `css` così come è stata mostrata presenta però un limite. Infatti con questa sintassi è possibile specificare una sola proprietà alla volta. Se volessi aggiungere un bordo alla lampadina dovrei scrivere qualcosa del genere:

Evidentemente questo è un modo davvero poco elegante, dato che presenta delle ripetizioni (*piccola nota: jQuery non tiene conto degli spazi nel codice, quindi puoi organizzare il codice come preferisci*). Per specificare proprietà multiple possiamo includerle in una variabile, e poi passare quest'ultima come parametro della funzione `css`, in questo modo:

È importante notare le piccole differenze che ci sono fra i due metodi:

- nel primo abbiamo due stringhe, delimitate da virgolette e separate da una virgola: "proprietà","valore"
- nel secondo, abbiamo le parentesi graffe che delimitano tutti i valori, le stringhe sono separate da due punti: "proprietà" : "valore", e le varie coppie sono separate da virgole.

## Quali sono gli effetti predefiniti in jQuery?

Come puoi leggere nella [documentazione di jQuery](#), ci sono alcuni effetti predefiniti che possiamo applicare ai nostri elementi. Nella scorsa lezione abbiamo esaminato l'effetto `fadeIn`; in jQuery questi effetti sono presenti in coppia, infatti abbiamo:

- la coppia `show/hide`: l'elemento cresce/decresce in altezza, larghezza e opacità
- la coppia `fadeIn/fadeOut`: l'elemento appare/scompare in dissolvenza
- la coppia `slideUp/slideDown`: l'elemento appare/scompare a tendina, verso l'alto

Tutte queste funzioni accettano come parametri le stringhe "slow", "normal", "fast", oppure un numero, che indica quanti millisecondi deve durare l'animazione. L'utilizzo di questi effetti è davvero semplice: basta selezionare l'elemento e poi applicare, di volta in volta, l'effetto desiderato. Facciamo alcuni esempi con la nostra "amica" lampadina:

- [Primo esempio: fadeIn e fadeOut](#)

- [Secondo esempio: show e hide](#)
- [Terzo esempio: slideUp e slideDown](#)

## Conclusioni

Conoscere come modificare gli stili inline, senza dover per forza aggiungere classi ai fogli di stile apre il mondo ad un ampio ventaglio di possibilità. Questo, insieme all'argomento della prossima lezione, ovvero come creare animazioni personalizzate, ti permetterà di esprimere tutta la tua creatività utilizzando [jQuery](#).

- [Lezione 1: introduzione e primi concetti fondamentali](#)
- [Lezione 2: gli eventi \(concetti basilari\)](#)
- [Lezione 3: alternare le funzioni, le variabili, primi effetti](#)
- Lezione 4: gli stili inline, gli effetti predefiniti
- [Lezione 5: le animazioni personalizzate](#)