Vuoi imparare jQuery ma non sai da dove iniziare? Alternare le funzioni, le variabili, primi effetti

Nell'articolo della settimana scorsa avevamo realizzato un semplice interruttore per lampadina. In questa lezione procederemo all'ottimizzazione ed al miglioramento di questo esempio, illustrando **nuove funzionalità del framework jQuery**.

Procediamo quindi senza indugi!

Alternare più azioni: la funzione toggle

Chiaramente, un normale interruttore non ha un pulsante per accendere e uno per spegnere la lampadina: tutto è incluso in un unico dispositivo. Proviamo allora ad ottimizzare anche il nostro jButton! Quello che vogliamo realizzare è un unico link che spenga e accenda la lampadina, a seconda del suo stato.

Teoricamente parlando, ci servirebbe un costrutto del tipo *"Al mio click **se** fra gli attributi del link non è presente la classe 'accesa' aggiungila, **altrimenti** rimuovila.*

Se ti intendi un po' di programmazione, avrai già capito che quello di cui potremmo aver bisogno è la coppia if-else, tuttavia jQuery fornisce una funzione più elegante adatta allo scopo: la funzione toggle

Toggle ci consente di alternare fra due o più funzioni al click del mouse, avendo cura di "ricordarsi" lo stato dell'oggetto che cambia. Potremmo quindi modificare il codice, eliminando un link nell'HTML:

ed inserendo nel file lampadina.js

In questo modo, cliccando sul link viene aggiunta e rimossa la classe accesa al div che contiene la lampadina, permettendoci di accenderla e spegnerla. C'è però **un problema: il testo del link rimane sempre "Accendi Lampadina"**, indipendentemente dal fatto che questa sia accesa o spenta. Dovremmo poter cambiare durante l'esecuzione il testo del link, in modo tale che cambi a seconda dello stato della lampadina.

Questa volta ci viene in soccorso **la funzione html di jQuery**. Questa sostituisce il testo all'interno di un elemento con quello specificato all'interno del file JavaScript, e si utilizza in questo modo:

Ad esempio, se avessimo un paragrafo:

con queste righe di codice potremmo cambiare il testo:

Vediamo come poter applicare tutto ciò al nostro jButton. Noi vogliamo che il testo all'interno del link con ID "interruttore" diventi "Spegni Lampadina" quando questa è accesa, altrimenti diventi "Accendi Lampadina". Possiamo quindi modificare il codice JavaScript:

Il risultato, direi, è più che soddisfacente.

Ottimizzare il codice: le variabili

Come puoi notare, **il nostro codice contiene alcune ripetizioni**. E le ripetizioni, quando si parla di programmazione, non sono mai un bene. Cosa accadrebbe se dovessimo, per un qualsiasi motivo, cambiare l'ID da "interruttore" a "button"? **Dovremmo modificare il codice JavaScript in tre diversi punti** e questo, per un progetto di centinaia di righe di codice, **condurrebbe facilmente a errori** e inconsistenze.

Possiamo ovviare facilmente a questa situazione con le **variabili**. In pratica le variabili sono dei contenitori: possono contenere testi, numeri oppure oggetti più complessi. L'aspetto interessante è che una variabile, una volta definita, può essere riusata più volte. Come sempre, un esempio vale più di mille parole:

Nel codice precedente abbiamo salvato il valore $("a#interruttore") nella variabile interruttore

utilizzando la sintassi:

Abbiamo poi utilizzato la variabile al posto del selettore. Potresti obiettare che la ripetizione c'è sempre. In fondo abbiamo aggiunto una riga di codice e ripetuto la parola "interruttore" per tre volte. **La differenza sta nel fatto che, in caso di modifiche all'HTML, dovremo cambiare il codice in un solo punto**, precisamente alla riga 2, invece che in tre punti diversi. Infatti, il nome della variabile resta fisso (scusa il gioco di parole), espandendo le modifiche a tutto il codice. Riesci a trovare un altro elemento che potrebbe essere salvato in una variabile?

Ma dove sono gli effetti di jQuery?

Proviamo ora a rendere un po' più interessante il nostro jButton aggiungendo una semplice animazione all'accensione ed allo spegnimento per emulare una lampadina a risparmio energetico (qui su YIW siamo molto sensibili ai temi ambientali). La caratteristica peculiare di questo tipo di lampadine è che non raggiungono subito la loro massima luminosità, ma hanno bisogno di qualche secondo per riscaldarsi. Per abbozzare questo effetto, utilizzeremo **la funzione fadeIn di jQuery**. La sintassi è la seguente:

L'effetto prodotto da fadeIn è la comparsa in dissolvenza di un **elemento nascosto**. La durata può essere espressa in millisecondi, oppure con una delle tre parole chiave: "slow", "normal", "fast". È importante **utilizzare le virgolette quando si passano dei parametri testuali** ad una funzione, mentre le tralasciamo per i valori numerici.

Ho sottolineato il fatto che l'elemento deve essere nascosto perché fadeIn operi. Al momento, però, il nostro div "lampadina" non è nascosto. Possiamo ovviare aggiungendo la riga:

al nostro CSS, oppure possiamo utilizzare la funzione hide di jQuery che, come può suggerire il nome, nasconde l'elemento selezionato. Combiniamo dunque queste nozioni nel nostro codice JavaScript:

Con queste modifiche, abbiamo nascosto il div "lampadina" (riga 4) e poi, sfruttando il concatenamento, abbiamo imposto che compaia in dissolvenza, con una durata di 5 secondi (5000 millisecondi). Allo spegnimento, nascondiamo nuovamente il div preparando così per una nuova accensione.

È importante per la buona riuscita del codice **ritornare ad uno stato consistente**: se non nascondessimo il div nuovamente, il fadeIn funzionerebbe solo la prima volta, mentre le successive no, perché non avrebbe un elemento nascosto su cui operare. Questo dovrebbe essere il risultato

finale.

Conclusioni

Fiuu! Abbiamo percorso un sacco di strada in questa lezione: siamo partiti da un interruttore grezzo, fino ad ottenere un interruttore variabile per una lampadina a risparmio energetico! E tutto questo con poche righe di codice grazie a jQuery. Nella prossima lezione vedremo come modificare gli stili CSS di un elemento senza dover per forza aggiungere o rimuovere una classe e scopriremo qualche altro effetto standard di jQuery. Alla prossima settimana!

- 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