

## Come e quando utilizzare il metodo `serialize` di jQuery?



Nei numerosi articoli che abbiamo scritto su jQuery, è sempre stato evidente come questo framework contribuisca a semplificare lo sviluppo del codice. In questo articolo vedremo come la conoscenza di tutti gli strumenti che ci vengono messi a disposizione possa **ulteriormente renderci la vita più comoda ed il codice più elegante**. Oggi analizzeremo il metodo `serialize()`; si tratta di uno strumento molto interessante il quale, applicato ad un form, **preleva il contenuto dei campi e prepara una querystring** pronta per essere inviata tramite una richiesta ajax. Se ad esempio consideriamo un form che richiede il nome ed il cognome e lo completo con i miei dati, applicando `serialize()` a questo form (all'evento `submit`) otterrò questa stringa:

```
nome=Maurizio&cognome=Tarchini
```

Utile vero?

Ma passiamo alla pratica. Consideriamo un documento html contenente un form

Ora poniamo di voler inviare questi dati tramite una chiamata asincrona (se sei un po' a digiuno di ajax, in [questo](#) articolo trovi tutto quello che c'è da sapere).

Quello che dovremo fare è prelevare il contenuto dei campi del form all'evento `submit` ed inviarli alla pagina di elaborazione. Il codice per fare questo potrebbe essere così (non ci starò sopra molto in quanto è tutto spiegato nei dettagli nell'articolo che ho segnalato precedentemente):

Come vedi, all'evento `submit`, vado a prelevare **uno alla volta** i dati contenuti nei campi del form. In seguito creo la variabile `querystring` nel formato richiesto dal metodo ajax (`nome=valore&nome=valore...`).

Ora non ci interessa cosa farà il file *elaborazione.php* ne quello che dovrà svolgere la funzione passata nel parametro *success*. Quello che ci interessa invece è: Cosa succede se il nostro form ha 400 campi? Dovremo prelevare i valori dei 400 campi uno per uno e costruire una lunghissima e complicata querystring da passare al parametro *data*? No. Utilizzando il metodo *serialize()* **potremo fare tutto con un'unica e semplice riga di codice.**

## Come utilizzare il metodo *serialize()*?

Semplice. Invece di questo

Basterà fare questo

Il metodo *serialize()* si occuperà di leggere tutti i campi del form con id *input\_form* (con *this* facciamo appunto riferimento a questo) e di creare la stringa con gli accoppiamenti *nome campo – contenuto campo* nel formato corretto e pronta per essere passata al parametro *data* del metodo *ajax()*.

Puoi provare con questo esempio che non fa altro che stampare a video (all'evento *submit*) la querystring risultante dal contenuto dei campi

## Attento, la querystring viene codificata

Una piccola nota per concludere. Se hai provato quest'ultimo script, ti sarai accorto che *serialize()* codifica (giustamente) i caratteri speciali. ad esempio l'email risulterà così:

*nome%40sito.com*

Per decodificare la stringa, PHP mette a disposizione la funzione [urldecode\(\)](#).

Mentre nel codice mostrato precedentemente potremo agire in questo modo

## Conclusione

In questo articolo abbiamo visto un esempio di come semplificare ulteriormente lo sviluppo di codice. Se pensiamo a form molto lunghi, abbiamo visto come *serialize()* risolve con una sola riga di codice il problema di prelevarne il contenuto. Utilizzando questo metodo possiamo godere anche di un secondo vantaggio: L'eventuale aggiunta o rimozione di campi dal form non dovrà

comportare **nessuna modifica al codice jQuery**, con effetti positivi sulla mantenibilità.