

Inserire un video con codice validato senza Javascript

Autore: **Lorenzo Forti** (lorenzo.forti@gmail.com)

Introduzione

L'**obiettivo** di questo articolo è inserire un video con codice valido XHTML Strict, accessibile, senza utilizzare Javascript e che suggerisca all'utente l'eventuale download del plugin adatto.

[Esempio finale](#)

Tra le tante novità apportate dall'introduzione di **HTML5** c'è da segnalare la possibilità di inserire video nella pagine senza l'utilizzo di componenti esterni come **Flash** o **QuickTime**. In un futuro prossimo il tag **<video>** rappresenterà il metodo validato XHTML più veloce per presentare filmati agli utenti.

Oggi, in attesa del passaggio definitivo a questo nuovo standard, esistono diverse alternative per inserire file multimediali in un sito:

- uso di librerie Javascript basate su SWFObject
- utilizzo congiunto del tag object ed embed
- inserimento del solo tag object

Lo svantaggio principale di queste tecniche è dato dal fatto che alcune non siano accessibili o non validate XHTML, altre non compatibili con tutti browser.

In questo momento di transazione tra standard ci possono venire in aiuto i **commenti condizionali negativi** di Internet Explorer.

Raggiungeremo il risultato finale passando attraverso varie soluzioni altrettanto valide (e validate!). Ognuno è libero di scegliere quella che preferisce in base alle proprie esigenze.

Per gli esempi mi sono affidato all'ottimo [JWPlayer](#), prodotto in Flash, gratuito ed opensource.

Il tag object standard

Il tag object, utilizzato in maniera standard per inserire un video, è interpretato correttamente da tutti i browser moderni compreso Explorer (test effettuati a partire dalla versione 6).

Un codice simile a questo inserito in una pagina mostrerà il filmato su tutte le piattaforme:

Codice n° 1

```
<object id="myvideo" type="application/x-shockwave-flash" data="player/player.swf" width="400" height="315">
  <param name="movie" value="player/player.swf" />
  <param name="allowScriptAccess" value="always" />
  <param name="flashvars" value="file=video.flv&image=img/preview.jpg" />
</object>
```

[Guarda l'esempio pratico n°1](#)

Attenzione! E' fondamentale, per IE, inserire l'id nel tag object altrimenti verrà prodotto un errore nella pagina.

Come mai allora tutta questa (apparente) difficoltà per avere un file multimediale nel proprio sito? Per capire il motivo bisogna analizzare la situazione in maniera più specifica.

Se avete visto il video con il codice appena analizzato è perchè, molto probabilmente, il plugin Flash è installato sulla vostra macchina. Ma cosa succede se un utente non ha il componente adatto?

Ecco due schermate che mettono a confronto il risultato del codice appena visto sia su **Explorer** che su **Firefox**:

Explorer



Firefox



La differenza è chiara: **in assenza** del plugin il browser di casa Microsoft non visualizza nulla mentre Firefox suggerisce automaticamente il file da scaricare.

Molti staranno obiettando che, al giorno d'oggi, è quasi impossibile trovare un client senza Flash. Personalmente mi trovo d'accordo ma se pensiamo, ad esempio, all'accessibilità in ambito Pubblica Amministrazione è evidente come non si possa dare nulla per scontato.

Ecco, quindi, un'altra possibile soluzione:

Codice n° 2

```
<object id="myvideo" type="application/x-shockwave-flash" data="player/player.swf" width="400" height="315">
  <param name="movie" value="player/player.swf" />
  <param name="allowScriptAccess" value="always" />
  <param name="flashvars" value="file=video.flv&image=img/preview.jpg" />
  <!-- codice alternativo solo per chi non ha Flash installato -->
  <p>Contenuto alternativo: <a href="#">scarica il plugin se non vedi il video</a></p>
</object>
```

[Guarda l'esempio pratico n°2](#)

La novità è rappresentata dal codice visibile solo all'utente che non ha installato il plugin Flash. Inserendo questo markup aggiuntivo Firefox non sarà più in grado di suggerire automaticamente, come in precedenza, il download del componente adatto ma mostrerà, al pari di Explorer, ciò che è stato inserito come contenuto alternativo: il link al sito di Adobe, un collegamento diretto al plugin o ad una pagina specifica, etc etc.

Se neanche questa soluzione fa al caso vostro ma volete proporre un video con codice validato XHTML Strict, che non utilizzi Javascript e che suggerisca all'utente, in ogni caso, il componente giusto da scaricare allora continuate la lettura.

La soluzione in attesa di HTML5

Questo approccio prevede, oltre ai già citati commenti condizionali negativi, l'utilizzo del solo **tag object** per mostrare i video su tutti i browser.

Tempo fa si utilizzava una soluzione con il tag **embed**: crossbrowser ma non standard (e quindi non validata XHTML).

Come già detto Internet Explorer interpreta correttamente l'object standard solo se il plugin Flash è già installato sul client. Per

questo motivo, solo per il prodotto di casa Microsoft, è necessario utilizzare un tag con una sintassi specifica che però, in caso di mancanza del plugin, è in grado di attivare la procedura per l'installazione automatica dei componenti.

Questo codice atipico non è interpretato da altre piattaforme.

Per la piena compatibilità, quindi, si rende necessario "innestare" - all'interno del primo - un secondo tag object con sintassi standard. Attraverso i **commenti condizionali negativi** faremo in modo che Explorer interpreti il blocco di codice adatto.

Attenzione! In teoria potremmo innestare un tag dentro l'altro senza commenti condizionali ma a causa di un **bug**, risolto con la versione 7, IE farebbe vedere un rettangolo vuoto sul secondo elemento.

L'object specifico per IE deve contenere gli attributi **classid** e **codebase** (utile per suggerire l'URL del componente e l'eventuale versione specifica).

Ecco come impostare correttamente il tag object per Explorer:

```
<!-- IE only -->
<object id="myvideo" classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=9,0,0,0"
title="Video di prova" width="400" height="315">
  <param name="movie" value="player/player.swf" />
  <param name="allowfullscreen" value="true" />
  <param name="allowscriptaccess" value="always" />
  <param name="flashvars" value="file=video.flv&image=img/preview.jpg" />
  <param name="wmode" value="transparent" />
</object>
```

Il codice appena visto **non viene interpretato** dai browser diversi da IE. Quello che segue invece sì:

```
<object id="myvideo" type="application/x-shockwave-flash" data="player/player.swf"
title="Video di prova" width="400" height="315">
  <param name="allowfullscreen" value="true" />
  <param name="allowscriptaccess" value="always" />
  <param name="flashvars" value="file=video.flv&image=img/preview.jpg" />
  <param name="wmode" value="transparent" />
</object>
```

Come innestare un tag dentro l'altro in maniera che IE interpreti solo il primo mentre gli altri browser solo il secondo? Con i commenti condizionali negativi, appunto.

Nello specifico i navigatori diversi da IE leggono tutti e due i tag object ma interpretano e mostrano solo il secondo. Primo problema risolto.

Purtroppo anche Explorer potrebbe sovrascrivere il primo con il secondo, con tutte le difficoltà già viste in caso di mancanza del plugin adatto.

Per questo motivo, attraverso i tag condizionali, facciamo in modo che IE ignori del tutto l'object valido per gli altri browser.

Il fulcro di questa tecnica, quindi, è rappresentato dal particolare modo in cui vengono scritti i tag condizionali:

```
<!--[if !IE]><!-->
... CODICE NON INTERPRETATO DA IE MA SOLO DAGLI ALTRI BROWSER...
<!--<![endif]-->
```

Per comprendere il motivo per cui si utilizza questa sintassi è necessaria una breve analisi dei tag condizionali negativi. Si vedano gli esempi successivi.

Commento condizionale negativo standard ma non validato

```
<![if !IE]>
BLABLA
<![endif]>
```

Explorer, correttamente, **ignora** il contenuto. Gli altri browser, correttamente, **leggono** il contenuto. Il codice non viene validato.

Commento condizionale negativo validato ma non funzionante su browser diversi da ie

```
<!--[if !IE]>
BLABLA
<![endif]-->
```

Explorer, correttamente, **ignora** il contenuto. Gli altri browser **vedono** il contenuto come un commento html e non lo visualizzano. Il codice viene validato.

Commento condizionale validato con hack e compatibile con tutti i browser

```
<!--[if !IE]><!-->
BLABLA
<!--<![endif]-->
```

Explorer, correttamente, **ignora** il contenuto. Gli altri browser, correttamente, **leggono** il contenuto. Il codice è **validato xhtml strict**.

Si può notare come sia stato aggiunto il codice di chiusura del commento dopo l'apertura e la chiusura del tag condizionale. In questa maniera IE interpreta correttamente la condizione negativa ed ignora il codice all'interno del blocco.

Gli altri browser, invece, vedono i tag condizionali come semplici commenti ma possono interpretare il markup racchiuso tra essi.

Codici di esempio

Finalmente possiamo giungere al codice di esempio finale:

Codice n° 3

```
<object id="myvideo" classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=9,0,0,0"
title="Video di prova" width="400" height="315">
  <param name="movie" value="player/player.swf" />
  <param name="allowfullscreen" value="true" />
  <param name="allowscriptaccess" value="always" />
  <param name="flashvars" value="file=video.flv&image=img/preview.jpg" />
  <param name="wmode" value="transparent" />

  <!--[if !IE]><!-->
  <object id="myvideo-2" type="application/x-shockwave-flash" data="player/player.swf"
title="Video di prova" width="400" height="315">
    <param name="allowfullscreen" value="true" />
    <param name="allowscriptaccess" value="always" />
    <param name="flashvars" value="file=video.flv&image=img/preview.jpg" />
```

```
<param name="wmode" value="transparent" />
</object>
<!--<![endif]-->

</object>
```

[Guarda l'esempio pratico n°3](#)

Il codice appena visto ha il difetto di essere alquanto **ridondante** per quanto riguarda i parametri "param" che vengono necessariamente ripetuti.

In alternativa, cambiando leggermente la sintassi, è possibile inserirli una sola volta anche se il risultato può non piacere da un punto di vista estetico e della leggibilità (dal punto di vista funzionale non cambia nulla):

```
<!--[if IE]>
<object id="myvideo" classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=9,0,0,0" width="400" height="315">
<param name="movie" value="player/player.swf" />
<!-->

<!--[if !IE]><!-->
<object id="myvideo-2" type="application/x-shockwave-flash" data="player/player.swf"
title="Video di prova" width="400" height="315">
<!--<![endif]-->

<param name="allowfullscreen" value="true" />
<param name="allowscriptaccess" value="always" />
<param name="flashvars" value="file=video.flv&image=img/preview.jpg" />
<param name="wmode" value="transparent" />

</object>
```

[Guarda l'esempio pratico n°4](#)

In questo caso il **primo object** è inserito come un commento vero e proprio interpretato solo da IE ma ignorato dagli altri browser che leggeranno solo il secondo tag.

A voi la scelta del metodo che maggiormente si adatta alle vostre esigenze.

Compatibilità browser

Prima di concludere i dati di compatibilità e le piattaforme su cui ho effettuato dei test.

Windows

- **Explorer 6** [richiesta attivazione controlli video*]
- **Explorer 7** [richiesta attivazione controlli video*]
- **Explorer 8** [richiesta attivazione controlli video*]
- **Explorer 9** [richiesta attivazione controlli video*]
- **Chrome 6.0.x**
- **Firefox 3.6.x**
- **Safari 5.0.x**
- **Opera 10.x** [richiesta attivazione controlli video*]

* l' attivazione dei controlli (attraverso la barra spaziatrice, pulsante invio o click del mouse) è dovuta ad un aggiornamento di sicurezza introdotto a partire da IE 6. L'unico modo per disattivare questa procedura è aggiungere il sito all'area Siti attendibili.

Anche Opera ha adottato questa procedura a partire dalla versione 9.x. **Il problema non si verifica adottando la soluzione Javascript.**

Linux

- Firefox 3.6.x
- Chrome 6.0.x
- Chromium 6.0.x
- Epiphany 2.30.x

Mac OS

- Nessun browser testato... per mancanza di risorse Apple :-)