

Italian Perl Conference 2006

Riempire dinamicamente un <SELECT> usando AJAX *Un esempio concreto*

Emiliano Bruni (<http://www.ebruni.it/>)



L'esempio dal vivo

- Usato in un sito di e-commerce B2B di una catena in franchising di videoteche
- E' stato sviluppato senza praticamente nessuna conoscenza di AJAX e della libreria prototype.js in 90 minuti
-

L'esempio dal vivo

Negozio: Amministratore del sistema ▼

Anno del listino: ▼

Mese del listino: ▼

Casa Cinematografica: Selezionare il mese/anno ▲

Invia Reimposta

L'esempio dal vivo

Negozio:

Anno del listino:

Mese del listino:

Casa Cinematografica:

L'esempio dal vivo

Negozio: Amministratore del sistema ▼

Anno del listino: 2006 ▼

Mese del listino: 04 - Aprile ▼

Casa Cinematografica: Caricamento in corso. Attendere... 

L'esempio dal vivo

Negozio:	Amministratore del sistema ▾
Anno del listino:	2006 ▾
Mese del listino:	04 - Aprile ▾
Casa Cinematografica:	<div style="border: 1px solid #ccc; padding: 5px;"><ul style="list-style-type: none">USA (Scaduto il 2006-02-25 00:00:00)Millennium (Scaduto il 2006-02-27 00:00:00)O1 (Scaduto il 2006-02-27 00:00:00)Paramount (Scaduto il 2006-02-27 00:00:00)Athena (Scaduto il 2006-03-01 00:00:00)DNC (Scaduto il 2006-03-05 00:00:00)Buena Vista (Scaduto il 2006-03-06 00:00:00)Eagle Pictures (Scaduto il 2006-03-06 00:00:00)PassWorld (Scaduto il 2006-03-06 00:00:00)Cecchi Gori (Scaduto il 2006-03-15 00:00:00)</div>
<input type="button" value="Invia"/> <input type="button" value="Reimposta"/>	

Analisi del codice sorgente

```
<head>
```

```
  <script src="/js/prototype.js"></script>
```

```
</head>
```

Analisi del codice sorgente

```
<form method="POST" action="..." id="frmSelezioneListino">  
...  
</form>
```

Analisi del codice sorgente

Negozio:

Amministratore del sistema 

```
<td>Negozio:</td>
```

```
<td>
```

```
  <select name="anagr.cod_user">
```

```
    <option value=""></option>
```

```
    <option value="info@ebruni.it">Amministratore del  
      sistema</option>
```

```
    ....
```

```
  </select>
```

```
</td>
```

Analisi del codice sorgente

Anno del listino:

2006 

```
<td>Anno del listino:</td>
```

```
<td>
```

```
  <select name="listino.num_anno" id="listino.num_anno">
```

```
    <option value=""></option>
```

```
    <option value="2005">2005</option>
```

```
    <option value="2006">2006</option>
```

```
  </select>
```

```
</td>
```

Analisi del codice sorgente

Mese del listino:

 

```
<td>Mese del listino:</td>
```

```
<td>
```

```
  <select name="listino.num_mese" id="listino.num_mese">
```

```
    <option value=""></option>
```

```
    <option value="06">06 - Giugno</option>
```

```
    ...
```

```
  </select>
```

```
</td>
```

Analisi del codice sorgente

Casa Cinematografica:

```
<td valign="top">Casa Cinematografica:</td>
```

```
<td valign="top">
```

```
  <select size="10" id="listino.id_listino">
```

```
    <option value="">Selezionare il mese/anno</option>
```

```
</select>
```

```
<span id='ajaxRequestInProgress' style="display: none;">
```

```
  <img src='/images/spinner.gif'>
```

```
</span>
```

```
</td>
```

Analisi del codice sorgente

Monitor di elementi del form. Controlla il variare del valore di un elemento del form

```
<script type="text/javascript">
<!--
  new Form.Element.EventObserver
  (
    'listino.num_anno',
    function( element, value )
    {
      new Ajax.Request
        (
          '/shared/ajax_elenco_cc_per_mese_anno.mpl',
          {
            parameters: Form.serialize('frmSelezioneListino'),
            asynchronous: 1
          }
        )
    }
  );
//-->
</script>
```

Analisi del codice sorgente

```
<script type="text/javascript">
<!--
  new Form.Element.EventObserver
  (
    'listino.num_anno',
    function( element, value )
    {
      new Ajax.Request
        (
          '/shared/ajax_elenco_cc_per_mese_anno.mpl',
          {
            parameters: Form.serialize('frmSelezioneListino'),
            asynchronous: 1
          }
        )
    }
  );
//-->
</script>
```

Form.EventObserver non va bene in quanto "negozio" e "casacinematografica" fanno parte dello stesso form ma non mi interessano quando cambiano

Analisi del codice sorgente

```
<script type="text/javascript">
<!--
  new Form.Element.EventObserver
  (
    'listino.num_anno',
    function( element, value )
    {
      new Ajax.Request
        (
          '/shared/ajax_elenco_cc_per_mese_anno.mpl',
          {
            parameters: Form.serialize('frmSelezioneListino'),
            asynchronous: 1
          }
        )
    }
  );
//-->
</script>
```

Identificativo dell'elemento da monitorare per eventuali cambiamenti nel valore

Analisi del codice sorgente

```
<script type="text/javascript">
<!--
  new Form.Element.EventObserver
  (
    'listino.num_anno',
    function( element, value )
    {
      new Ajax.Request
        (
          '/shared/ajax_elenco_cc_per_mese_anno.mpl',
          {
            parameters: Form.serialize('frmSelezioneListino'),
            asynchronous: 1
          }
        )
    }
  );
//-->
</script>
```

Questa sarà la funzione da richiamare sull'onChange del valore dell'elemento sotto "osservazione"

Analisi del codice sorgente

```
<script type="text/javascript">
<!--
  new Form.Element.EventObserver
  (
    'listino.num_anno',
    function( element, value )
    {
      new Ajax.Request
        (
          '/shared/ajax_elenco_cc_per_mese_anno.mpl',
          {
            parameters: Form.serialize('frmSelezioneListino'),
            asynchronous: 1
          }
        )
    }
  );
//-->
</script>
```

Oggetto che incapsula tutta la complessità di XMLHttpRequest. Chiama la pagina remota in modo asincrono, estrae la risposta e la processa

Analisi del codice sorgente

```
<script type="text/javascript">
<!--
  new Form.Element.EventObserver
  (
    'listino.num_anno',
    function( element, value )
    {
      new Ajax.Request
        (
          ' /shared/ajax_elenco_cc_per_mese_anno.mpl ',
          {
            parameters: Form.serialize('frmSelezioneListino'),
            asynchronous: 1
          }
        )
    }
  );
//-->
</script>
```

Url della pagina chiamata dal componente AJAX

Analisi del codice sorgente

```
<script type="text/javascript">
<!--
  new Form.Element.EventObserver
  (
    'listino.num_anno',
    function( element, value )
    {
      new Ajax.Request
        (
          '/shared/ajax_elenco_cc_per_mese_anno.mpl',
          {
            parameters: Form.serialize('frmSelezioneListino'),
            asynchronous: 1
          }
        )
    }
  );
//-->
</script>
```

I parametri da passare via GET/POST alla pagina

Analisi del codice sorgente

```
<script type="text/javascript">
<!--
  new Form.Element.EventObserver
  (
    'listino.num_anno',
    function( element, value )
    {
      new Ajax.Request
        (
          '/shared/ajax_elenco_cc_per_mese_anno.mpl',
          {
            parameters: Form.serialize('frmSelezioneListino'),
            asynchronous: 1
          }
        )
    }
  );
//-->
</script>
```

Genera automaticamente la stringa
key1=val1&key2=val2&...
dove key1...keyn sono i "name" di tutti i campi di
frmSelezioneListino e val1..valn sono i valori
contenuti in tali campi

Analisi del codice sorgente

```
<script type="text/javascript">
<!--
  new Form.Element.EventObserver
  (
    'listino.num_anno',
    function( element, value )
    {
      new Ajax.Request
        (
          '/shared/ajax_elenco_cc_per_mese_anno.mpl',
          {
            parameters: Form.serialize('frmSelezioneListino'),
            asynchronous: 1
          }
        )
    }
  );
//-->
</script>
```

Si può continuare a lavorare nell'attesa che il server risponda. Questo valore è comunque il default e può essere omissso

Analisi del codice sorgente

```
<script type="text/javascript">
<!--
  new Form.Element.EventObserver
  (
    'listino.num_mese',
    function( element, value )
    {
      new Ajax.Request
      (
        '/shared/ajax_elenco_cc_per_mese_anno.mpl',
        {
          parameters: Form.serialize('frmSelezioneListino'),
          asynchronous: 1
        }
      )
    }
  );
//-->
</script>
```

Identico codice anche per monitorare le variazioni sul <SELECT> del mese

Analisi del codice sorgente

```
<script>
  var myGlobalHandlers =
  {
    onCreate: function()
    {
      Element.show('ajaxRequestInProgress');
      Element.update
      (
        'listino.id_listino',
        '<option>Caricamento in corso. Attendere...</option>'
      );
    },
    onComplete: function(json,request)
    {
      Element.hide('ajaxRequestInProgress');
      Element.update('listino.id_listino',request.responseText);
    }
  };
  Ajax.Responders.register(myGlobalHandlers);
</script>
```

Lista di cose da fare quando, globalmente, accadono degli eventi AJAX (onLoading, onSuccess, onFailure, onException, onCreate, onComplete)

Analisi del codice sorgente

```
<script>
  var myGlobalHandlers =
  {
    onCreate: function()
    {
      Element.show('ajaxRequestInProgress');
      Element.update
      (
        'listino.id_listino',
        '<option>Caricamento in corso. Attendere...</option>'
      );
    },
    onComplete: function(json,request)
    {
      Element.hide('ajaxRequestInProgress');
      Element.update('listino.id_listino',request.responseText);
    }
  };
  Ajax.Responders.register(myGlobalHandlers);
</script>
```

Deve essere un oggetto con dei metodi che onXxxxxx che ritornano le funzioni di callback che si vogliono eseguire in presenza dell'evento

Analisi del codice sorgente

```
<script>
  var myGlobalHandlers =
  {
    onCreate: function()
    {
      Element.show('ajaxRequestInProgress');
      Element.update
      (
        'listino.id_listino',
        '<option>Caricamento in corso. Attendere...</option>'
      );
    },
    onComplete: function(json,request)
    {
      Element.hide('ajaxRequestInProgress');
      Element.update('listino.id_listino',request.responseText);
    }
  };
  Ajax.Responders.register(myGlobalHandlers);
</script>
```

Andiamo quindi a definire l'oggetto che serve alla funzione Ajax.Responders.register

Analisi del codice sorgente

```
<script>
  var myGlobalHandlers =
  {
    onCreate: function()
    {
      Element.show('ajaxRequestInProgress');
      Element.update
      (
        'listino.id_listino',
        '<option>Caricamento in corso. Attendere...</option>'
      );
    },
    onComplete: function(json,request)
    {
      Element.hide('ajaxRequestInProgress');
      Element.update('listino.id_listino',request.responseText);
    }
  };
  Ajax.Responders.register(myGlobalHandlers);
</script>
```

Alla creazione di una nuova richiesta AJAX, vogliamo che venga eseguita una funzione che...

Analisi del codice sorgente

```
<script>
  var myGlobalHandlers =
  {
    onCreate: function()      Mostra l'elemento con l'immagine dello spinner
    {
      Element.show('ajaxRequestInProgress');
      Element.update
      (
        'listino.id_listino',
        '<option>Caricamento in corso. Attendere...</option>'
      );
    },
    onComplete: function(json,request)
    {
      Element.hide('ajaxRequestInProgress');
      Element.update('listino.id_listino',request.responseText);
    }
  };
  Ajax.Responders.register(myGlobalHandlers);
</script>
```

Analisi del codice sorgente

```
<script>
  var myGlobalHandlers =
  {
    onCreate: function()
    {
      Element.show('ajaxRequestInProgress');
      Element.update      Aggiorna il contenuto (innerHTML) della
      (                    <SELECT> inserendo la richiesta di attesa
        'listino.id_listino',
        '<option>Caricamento in corso. Attendere...</option>'
      );
    },
    onComplete: function(json,request)
    {
      Element.hide('ajaxRequestInProgress');
      Element.update('listino.id_listino',request.responseText);
    }
  };
  Ajax.Responders.register(myGlobalHandlers);
</script>
```

Analisi del codice sorgente

```
<script>
  var myGlobalHandlers =
  {
    onCreate: function()
    {
      Element.show('ajaxRequestInProgress');
      Element.update
      (
        'listino.id_listino',
        '<option>Caricamento in corso. Attendere...</option>'
      );
    },
    onComplete: function(json,request)
    {
      Element.hide('ajaxRequestInProgress');
      Element.update('listino.id_listino',request.responseText);
    }
  };
  Ajax.Responders.register(myGlobalHandlers);
</script>
```

Al completamento della risposta del server...

Analisi del codice sorgente

```
<script>
  var myGlobalHandlers =
  {
    onCreate: function()
    {
      Element.show('ajaxRequestInProgress');
      Element.update
      (
        'listino.id_listino',
        '<option>Caricamento in corso. Attendere...</option>'
      );
    },
    onComplete: function(json,request)
    {
      Element.hide('ajaxRequestInProgress');
      Element.update('listino.id_listino',request.responseText);
    }
  };
  Ajax.Responders.register(myGlobalHandlers);
</script>
```

Nasconde l'elemento con l'immagine dello spinner

Analisi del codice sorgente

```
<script>
  var myGlobalHandlers =
  {
    onCreate: function()
    {
      Element.show('ajaxRequestInProgress');
      Element.update
      (
        'listino.id_listino',
        '<option>Caricamento in corso. Attendere...</option>'
      );
    },
    onComplete: function(json,request)
    {
      Element.hide('ajaxRequestInProgress');
      Element.update('listino.id_listino',request.responseText);
    }
  };
  Sostituisce l'innerHTML della <SELECT> con il risultato ottenuto dal server
  Ajax.Responders.register(myGlobalHandlers);
</script>
```

Analisi del codice sorgente

Tutti questi metodi, oltre all'intero oggetto Element non esistono nel JS standard ed sono aggiunte di prototype che, come si vede, semplificano, non di poco, la vita

```
<script>
  var myGlobalHandlers =
  {
    onCreate: function()
    {
      Element.show('ajaxRequestInProgress');
      Element.update
      (
        'listino.id_listino',
        '<option>Caricamento in corso. Attendere...</option>'
      );
    },
    onComplete: function(json,request)
    {
      Element.hide('ajaxRequestInProgress');
      Element.update('listino.id_listino',request.responseText);
    }
  };
  Ajax.Responders.register(myGlobalHandlers);
</script>
```

HTML::Prototype

- Tutto il codice di prototype.js può essere scritto ovviamente a mano direttamente nell'HTML
- Può essere generato a livello di scripting tramite il modulo HTML::Prototype e inserito poi nell'HTML

HTML::Prototype

```
my $ajax = new HTML::Prototype;
$ajax->observe_field(
    'listino.num_anno',
    {
        url => '/shared/ajax_elenco_cc_per_mese_anno.mpl',
        with => "Form.serialize('frmSelezioneListino')",
    }
);
```

Produce l'EventObserver visto [in precedenza](#)

Problemi riscontrati

- A creare da zero questo esempio, senza conoscere AJAX => 90 min
- Nella documentazione i parametri json e request della callback onComplete sono invertiti (mi ha fatto perdere un paio d'ore)
- Su Internet Explorer x.y impostare l'innerHTML di una <SELECT> NON FUNZIONA (Q276228 - <http://support.microsoft.com/default.aspx?scid=kb;en-us;276228>)
- Quest'ultimo bug mi ha fatto pensare due giorni

Problemi riscontrati

- Dato che prototype già possiede dei workaround per dei bug di IE...
- Ho modificato la funzione `Element.update` per evitare questo bug così che
`$('#select_id').innerHTML = '<option>...</option>';`
continua ovviamente a non funzionare ma almeno
`Element.update('select_id', '<option>...</option>');`
è ora browser indipendente nel comportamento.
- La patch è stata inviata allo sviluppatore di `prototype.js`

La patch

```
--- /usr/src/prototype-1.4.0/dist/prototype.js 2006-01-20 21:08:20.000000000 +0100
+++ prototype.js 2006-05-14 23:36:10.000000000 +0200
@@ -874,7 +874,14 @@
     },

    update: function(element, html) {
-     $(element).innerHTML = html.stripScripts();
+     var iHTML = html.stripScripts();
+     $(element).innerHTML = iHTML;
+     /* Avoid IE bug when setting the innerHTML Property of the Select Object */
+     /* Ref: http://support.microsoft.com/default.aspx?scid=kb;en-us;276228 */
+     if ($(element).tagName == 'SELECT' && $(element).innerHTML != iHTML) {
+         $(element).innerHTML = '!~~~~~!' + iHTML;
+         $(element).outerHTML = $(element).outerHTML.replace('!~~~~~!', '');
+     }
+     setTimeout(function() {html.evalScripts()}, 10);
    },

@@ -1778,4 +1785,4 @@

    return [valueL, valueT];
}
-}
\ No newline at end of file
+}
```