

Javascript no intrusiu amb Prototype

Manel Zaera, manelzaera@gmail.com, Juliol 2008

Llicència del document: GNU GFDL i CC BY-SA 3.0, 2.5, 2.0, 1.0

Llicència del codi dels exemples: GNU GPL

Introducció

Aquest és un document que explica com es pot utilitzar Prototype per aplicar la tècnica de Javascript no intrusiu (*non obtrusive Javascript*, en anglès) en el desenvolupament d'aplicacions web. No és ni un tutorial de Prototype ni un document metodològic sobre la tècnica esmentada, sinó una aportació a la difusió de la utilització d'aquesta biblioteca Javascript de programari lliure.

Javascript no intrusiu

Sovint el desenvolupament d'aplicacions web en llenguatges de seqüència *-script-* com PHP acaba generant un grapat de fitxers de codi font que barregen etiquetes HTML, elements del llenguatge generador *-com PHP-* i altres, com l'especificació del comportament amb Javascript. Aquesta barreja fa menys intel·ligible el codi font i també en complica el manteniment i el reaprofitament.

La tècnica de Javascript no intrusiu [1] consisteix en la separació del comportament d'una pàgina HTML especificada en un conjunt d'objectes, variables i funcions de Javascript, de forma que quedi separat, diferenciat, aquest comportament de l'aspecte de la pàgina, definit en el codi HTML i els fulls d'estil. De forma senzilla ve a ser l'aïllament de tots els elements de Javascript que tradicionalment s'han inserit dins del codi de les etiquetes HTML *-com els gestors d'esdeveniments-* en un bloc de codi dins de la capçalera del document HTML, ja sigui en forma de codi inserit o bé referenciat cap a un fitxer independent.

En [2], [3] i [4] es pot trobar informació més completa sobre la tècnica de Javascript no intrusiu, així com algunes recomanacions útils sobre com aplicar-la. Com es veurà aquest document no aplica aquestes recomanacions, simplement per una qüestió pràctica, perquè l'objectiu es veure la utilitat de Prototype aplicat a la tècnica i no tant la tècnica en qüestió.

Prototype

Prototype [5] és un marc de desenvolupament Javascript, una biblioteca de classes desenvolupada amb Javascript que facilita el treball amb els elements de l'arbre DOM d'un document HTML, i també l'ús de crides AJAX. No és l'única biblioteca existent però, és una de les més populars, a més de ser programari lliure. Per això, per mostrar com utilitzar Javascript de forma no intrusiva, es farà amb Prototype i no directament amb la sintaxi de Javascript o amb alguna altra biblioteca.

Una de les biblioteques de Javascript interessant que existeixen a més de Prototype és

Yahoo! UI [6], també programari lliure, tot i que requereix l'ús de diversos fitxers font en funció d'allò que es vulgui fer. Prototype ofereix una via més senzilla de fer el mateix.

Exemples pràctics

A continuació s'il·lustra l'ús de Javascript no intrusiu amb alguns exemples, comparant-ho amb la versió intrusiva de la solució als mateixos.

Clic sobre un botó

Quan es vol detectar el clic sobre un botó d'un formulari s'ha d'observar l'esdeveniment **click**. L'exemple consisteix en la detecció del clic per mostrar a continuació el text "Hola" en un diàleg mitjançant la funció **alert**. El primer fragment de codi mostra com es pot fer inserint el codi de gestió de l'esdeveniment dins del codi HTML:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" lang="ca_AD" xml:lang="ca_AD">
  <head>
    <title>Javascript intrusiu</title>
    <meta http-equiv="content-type" content="text/html; charset=ISO-8859-1"/>
  </head>
  <body>
    <form name="form1" method="post" action="#">
      <input type="button" name="botohola" onclick="alert('Hola')" value="Ok"/>
    </form>
  </body>
</html>
```

Es pot fer el mateix de la següent forma:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" lang="ca_AD" xml:lang="ca_AD">
  <head>
    <title>Javascript no intrusiu</title>
    <meta http-equiv="content-type" content="text/html; charset=ISO-8859-1"/>
    <script type="text/javascript" src="prototype.js"></script>
    <script type="text/javascript">
      function onBotoHola(e) {
        alert("Hola");
      }

      function onLoad(e) {
        Event.observe('botohola', 'click', onBotoHola);
      }

      Event.observe(window, 'load', onLoad);
    </script>
  </head>
  <body>
```

```
<form name="form1" method="post" action="#">
  <input type="button" id="botohola" name="botohola" value="ok"/>
</form>
</body>
</html>
```

El botó continua tenint un esdeveniment associat però, queda separat de la seva presentació. L'esdeveniment es defineix mitjançant el mètode observe de la classe Event. Com es pot veure, l'esdeveniment es declara dins de la funció **onLoad**. Aquesta és una funció que s'executa quan s'ha carregat el document. Fins que no s'ha carregat no es pot monitoritzar cap esdeveniment dels objectes del formulari, perquè no existeixen. Una alternativa seria posar la declaració de l'esdeveniment dins del codi HTML, després de posar l'objecte però, significaria tornar enrere, al Javascript intrusiu.

Es pot veure en el fragment de la figura següent:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" lang="ca_AD" xml:lang="ca_AD">
  <head>
    <title>Javascript no intrusiu</title>
    <meta http-equiv="content-type" content="text/html; charset=ISO-8859-1"/>
    <script type="text/javascript" src="prototype.js"></script>
    <script type="text/javascript">
      function onBotoHola(e) {
        alert("Hola");
      }
    </script>
  </head>
  <body>
    <form name="form1" method="post" action="#">
      <input type="button" id="botohola" name="botohola" value="ok"/>
      <script type="text/javascript">
        Event.observe('botohola', 'click', onBotoHola);
      </script>
    </form>
  </body>
</html>
```

Com es pot veure, es passa un paràmetre a les funcions onLoad i onBotoHola, el paràmetre **e**. El paràmetre no sempre s'utilitza però, permet accedir a la informació de l'objecte que ha desencadenat l'esdeveniment. Per fer-ho, només cal fer la següent crida:

```
var el = Event.element(e);
```

L'atribut **id** de la variable **el**, conté l'identificador de l'element. En el cas d'onBotoHola, **el.id** conté el valor **botohola**. Es pot veure en el següent tros de codi, el qual mostra un text diferent en funció del botó que s'ha premut.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="ca_AD" xml:lang="ca_AD">
<head>
  <title>Javascript no intrusiu</title>
  <meta http-equiv="content-type" content="text/html; charset=ISO-8859-1"/>
  <script type="text/javascript" src="prototype.js"></script>
  <script type="text/javascript">
function onBoto(e) {
  var el = Event.element(e);
  if (el.id == 'botohola') {
    $('text').innerHTML = 'Hola';
  }
  if (el.id == 'botoadeu') {
    $('text').innerHTML = 'Adéu';
  }
}

function onLoad(e) {
  Event.observe('botohola', 'click', onBoto);
  Event.observe('botoadeu', 'click', onBoto);
}

Event.observe(window, 'load', onLoad);
</script>
</head>
<body>
<form name="form1" method="post" action="#">
  <input type="button" id="botohola" name="botohola" value="ok"/>
  &nbsp;
  <input type="button" id="botoadeu" name="botoadeu" value="ko"/>
  <br/><br/>
  <div id="text"></div>
</form>
</body>
</html>
```

Amb una funció es gestiona el clic sobre els dos botons.

Selecció d'un element d'una llista

De vegades es vol conèixer quin element, en una llista desplegable, s'ha seleccionat, per tal d'activar altres objectes d'un formulari, com ara el cas en què entre una llista de llenguatges de programació es pot optar per triar l'opció corresponent a "Un altre", activant-se un camp de text per especificar quin és aquest altre llenguatge. La primera figura mostra un formulari amb una llista de selecció i un camp de text, inicialment no actiu. El camp de text s'habilita per escriure-hi quan se selecciona l'element "Un altre" de la llista, utilitzant la funció **onSeleccioLlenguatge**, indicada dins de l'element select, en resposta a l'esdeveniment **change**.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ca_AD" xml:lang="ca_AD">
<head>
```

```

<title>Javascript intrusiu</title>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1"/>
<script type="text/javascript">
function onSeleccioLlenguatge() {
    var llenguatge = document.form1.llenguatge;
    document.form1.altre.disabled = (llenguatge.value!=5);
    document.form1.altre.focus();
}
</script>
</head>
<body>
<form name="form1" method="post" action="#">
<b>Llenguatge de programaci-:</b>
<select name="llenguatge" onchange="onSeleccioLlenguatge()">
<option value="-1">---</option>
<option value="0">C/C++</option>
<option value="1">Java</option>
<option value="2">C#</option>
<option value="3">Python</option>
<option value="4">PHP</option>
<option value="5">Un altre</option>
</select>
<input type="text" name="altre" size="15" maxlength="15" disabled="false"/>
</b>
</form>
</body>
</html>

```

Es pot fer el mateix de la següent forma:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ca_AD" xml:lang="ca_AD">
<head>
<title>Javascript no intrusiu</title>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1"/>
<script type="text/javascript" src="prototype.js"></script>
<script type="text/javascript">
function onSeleccioLlenguatge(e) {
    var llenguatge = $('llenguatge');
    $('altre').disabled = (llenguatge.value!=5);
    $('altre').focus();
}

function onLoad(e) {
    $('altre').disabled = true;
    Event.observe('llenguatge', 'change', onSeleccioLlenguatge);
}

Event.observe(window, 'load', onLoad);
</script>
</head>
<body>
<form name="form1" method="post" action="#">
<b>Llenguatge de programaci-:</b>

```

```
<select name="llenguatge" id="llenguatge">
<option value="-1">---</option>
<option value="0">C/C++</option>
<option value="1">Java</option>
<option value="2">C#</option>
<option value="3">Python</option>
<option value="4">PHP</option>
<option value="5">Un altre</option>
</select>
<input type="text" name="altre" id="altre" size="15" maxlength="15"/>
</form>
</body>
</html>
```

Com es pot veure, la funció `onSeleccioLlenguatge` s'encarrega de gestionar el canvi a la llista **llenguatge** però, el control d'aquest esdeveniment passa d'estar en el corresponent paràmetre de l'element `select`, a estar declarat dins de la funció `onLoad`.

Creació i eliminació dinàmica d'objectes

A continuació es proposa el problema de tenir una taula amb una capçalera i un botó per afegir elements a la taula. Cada nova fila de la taula té un botó que serveix per eliminar-la. Una forma de fer-ho és la següent:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" lang="ca_AD" xml:lang="ca_AD">
<head>
<title>Javascript no intrusiu</title>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1"/>
<script type="text/javascript" src="prototype.js"></script>
<script type="text/javascript">
var numfiles = 0;

function onAfegeix() {
var fila = document.createElement('tr');
fila.id = 'fila_'+numfiles;
var nom = document.createElement('td');
nom.innerHTML = '<input type="text" id="nom_'+numfiles+' "
name="nom_'+numfiles+'"/>';
var cg1 = document.createElement('td');
cg1.innerHTML = '<input type="text" id="cg1_'+numfiles+' "
name="cg1_'+numfiles+'"/>';
var cg2 = document.createElement('td');
cg2.innerHTML = '<input type="text" id="cg2_'+numfiles+' "
name="cg2_'+numfiles+'"/>';
var boto = document.createElement('td');
boto.innerHTML = '<input type="button" id="elimina_'+numfiles+' "
name="elimina_'+numfiles+' " value="- "
onclick="onElimina(\'fila_'+numfiles+'\')"/>';
fila.appendChild(nom);
fila.appendChild(cg1);
fila.appendChild(cg2);
```

```

        fila.appendChild(boto);
        var cos = document.getElementById('cos');
        cos.appendChild(fila);

        numfiles++;
    }

    function onElimina(idfila) {
        var fila = document.getElementById(idfila);
        var cos = document.getElementById('cos');
        cos.removeChild(fila);
    }

    function onLoad() {
        numfiles = 0;
    }
</script>
</head>
<body onload="onLoad()">
<form id="form1" name="form1" method="post" action="#">
<input type="button" id="afegeix" name="afegeix" value="+"
onclick="onAfegeix()"/>
<table id="taula1">
<thead><tr><th>Nom</th><th>Cognom 1</th><th>Cognom 2</th></tr></thead>
<tbody id="cos"></tbody>
</table>
</form>
</body>
</html>

```

La versió no intrusiva seria la següent:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ca_AD" xml:lang="ca_AD">
<head>
<title>Javascript no intrusiu</title>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1"/>
<script type="text/javascript" src="prototype.js"></script>
<script type="text/javascript">
    var numfiles = 0;

    function onAfegeix(e) {
        var fila = document.createElement('tr');
        fila.id = 'fila_'+numfiles;
        var nom = document.createElement('td');
        nom.innerHTML = '<input type="text" id="nom_'+numfiles+'
name="nom_'+numfiles+'"/>';
        var cg1 = document.createElement('td');
        cg1.innerHTML = '<input type="text" id="cg1_'+numfiles+'
name="cg1_'+numfiles+'"/>';
        var cg2 = document.createElement('td');
        cg2.innerHTML = '<input type="text" id="cg2_'+numfiles+'

```

```
name="cg2_'+numfiles+'"/>';
    var boto = document.createElement('td');
    boto.innerHTML = '<input type="button" id="elimina_'+numfiles+'
name="elimina_'+numfiles+'" value="-"/>';
    fila.appendChild(nom);
    fila.appendChild(cg1);
    fila.appendChild(cg2);
    fila.appendChild(boto);
    $('cos').appendChild(fila);
    Event.observe('elimina_'+numfiles,'click',onElimina);

    numfiles++;
}

function onElimina(e) {
    var id = Event.element(e).id;
    var parts = id.split('_');
    var fila = $('fila_'+parts[1]);
    $('cos').removeChild(fila);
}

function onLoad(e) {
    numfiles = 0;
    Event.observe('afegeix','click',onAfegeix);
}

Event.observe(window,'load',onLoad);
</script>
</head>
<body>
<form id="form1" name="form1" method="post" action="#">
<input type="button" id="afegeix" name="afegeix" value="+"/>
<table id="taula1">
<thead><tr><th>Nom</th><th>Cognom 1</th><th>Cognom 2</th></tr></thead>
<tbody id="cos"></tbody>
</table>
</form>
</body>
</html>
```

Com es pot veure, en la generació dinàmica de les files de la taula s'insereix la crida a `Event.observe` per capturar el clic sobre el botó d'eliminació de la fila, mentre que a la primera versió, s'havia de generar el codi HTML de la fila amb el corresponent paràmetre `onclick` per gestionar aquest esdeveniment.

Conclusions

La implementació de comportament de les pàgines web en el desenvolupament de llocs web dinàmics es pot fer de moltes maneres però, utilitzant la tècnica de Javascript no intrusiu es creen aplicacions que es poden mantenir, reutilitzar i provar de forma més senzilla i eficient. Prototype és un exemple de les diferents biblioteques existents de programari lliure que faciliten la implementació de comportament en pàgines HTML mitjançant aquesta tècnica. L'ús de Prototype, com s'ha vist, ha d'anar acompanyat del

marcatge adequat dels elements que componen el document HTML que compon la pàgina web. Així, l'ús d'identificadors en tots els elements que cal monitoritzar és una característica essencial.

Referències

- [1] Javascript no intrusiu (Wikipedia, anglès): http://en.wikipedia.org/wiki/Unobtrusive_JavaScript
- [2] Guia ràpida de Prototype (anglès): <http://particletree.com/features/quick-guide-to-prototype>
- [3] Javascript no intrusiu, CSS y PHP: <http://www.cristalab.com/tips/31507/javascript-no-intrusivo-css-y-php.html>
- [4] The Seven Rules of Unobtrusive Javascript: <http://www.cristalab.com/tips/31507/javascript-no-intrusivo-css-y-php.html>
- [5] Prototype: <http://www.prototypejs.org>
- [6] Yahoo! UI: <http://developer.yahoo.com/yui>

Llicència

Aquest document es publica amb les llicències GNU Free Documentation License (<http://www.gnu.org/copyleft/fdl.html>) i amb Creative Commons Reconeixement-Compartir Igual 3.0 (<http://creativecommons.org/licenses/by-sa/3.0>), 2.5 (<http://creativecommons.org/licenses/by-sa/2.5>), 2.0 (<http://creativecommons.org/licenses/by-sa/2.0>) i 1.0 (<http://creativecommons.org/licenses/by-sa/1.0>).

A més, el codi font dels exemples del document es publica amb llicència GNU GPL (<http://www.gnu.org/copyleft/gpl.html>).