



UNIVERSITÀ DI BRESCIA
FACOLTÀ DI INGEGNERIA
Dipartimento di Elettronica per l'Automazione

Laboratorio di Robotica Avanzata **Advanced Robotics Laboratory**

Corso di Robotica Mobile
(Prof. Riccardo Cassinis)

**Miglioramento dell'accessibilità
di un sito per il sistema MyTobii
P10**

Elaborato di esame di:

Algo Carè

Consegnato il:

4 Agosto 2008

Sommario

Il lavoro di cui si riporta la sintesi è consistito nell'identificazione, finalizzata alla più rapida risoluzione, di alcuni gravissimi problemi di usabilità nell'utilizzo del sito per l'aggiornamento della base di dati del C.S.V. (Centro Servizi Volontariato) di Brescia riscontrati dall'utilizzatrice di un terminale con sistema di tracciamento oculare di tipo MyTobii P10, operante presso la sede del C.S.V. stesso.

I problemi diagnosticati e le proposte di soluzione sono stati comunicati a Metarete S.r.l., azienda manuttrice del sito in oggetto.

Le soluzioni effettivamente impiegate sono state oggetto di verifica e si sono rivelate valide. Il lavoro è inquadrato in un più ampio contesto orientato al futuro e alcuni miglioramenti sono prospettati.

1. Introduzione

1.1. Il contesto del lavoro

Il C.S.V. (Centro Servizi e Volontariato) di Brescia [1] possiede un sistema MyTobii P10 prodotto dalla svedese TOBII Technology A.B. [2] e distribuito in Italia da SR Labs S.r.l. [3]. Il sistema comprende un calcolatore dotato di un sistema di tracciamento oculare e da un ambiente software proprietario noto come iAble, comprensivo di browser.

Il sistema è utilizzato sperimentalmente per un tirocinio lavorativo. Al momento, l'unico operatore è Cinzia Rossetti (nel seguito Cinzia), che lo utilizza in qualità di terminalista per l'aggiornamento della base di dati del C.S.V. bresciano, servendosi di un sito [4] creato e gestito dall'azienda Metarete S.r.l. [5] (d'ora in poi Metarete), che serve anche gli altri C.S.V. nazionali. Dal punto di vista del terminale di Cinzia tale sito soffre di alcune deficienze, catalogabili a tutti gli effetti come catastrofi di usabilità, in quanto impedivano lo svolgimento del lavoro e la corretta fruizione di alcune funzioni.

1.2. Struttura del lavoro

In seguito ad una prima fase di analisi sono stati individuati due gravi problemi, di cui si rende conto nel capitolo 2, che sono stati comunicati a Metarete, insieme ad alcune ipotesi di soluzione. Nel paragrafo Ø si fa già menzione di una possibile soluzione ad un sottoproblema: essa, emersa in fase di analisi, non ha trovato sviluppi nell'implementazione finale. In una fase conclusiva si è provveduto a verificare la validità delle soluzioni adottate, confrontandole con le soluzioni proposte: di ciò si scrive nel capitolo 3.

Nel capitolo 4 si danno utili nozioni per la verifica del lavoro svolto; in particolare, nel paragrafo 4.1 si informa il lettore circa la raggiungibilità della sezione del sito più interessante nell'ottica della presente relazione.

Nel capitolo 5 si cerca di inquadrare il lavoro in una prospettiva più ampia che abbia al centro il sistema MyTobii P10 ed il suo software piuttosto che i singoli siti oggetto di navigazione; si aggiungono alcune considerazioni circa la perfezionabilità del lavoro sin qui svolto.

In APPENDICE, nei paragrafi 6.1 e 6.2, sono riportate due delle svariate pagine HTML di test utilizzate, le uniche che potrebbero forse rivelarsi utili anche nel futuro. Un terzo documento, nel paragrafo 6.3, elenca il risultato di alcuni test effettuati esplicitamente su singoli elementi HTML.

2. Il problema affrontato

Nel momento in cui il lavoro è iniziato, alcuni problemi di semplice individuazione erano già stati segnalati a Metarete e risolti. Tali problemi riguardavano elementi HTML di tipo `<textarea>` e non sono disponibili altre informazioni in merito. Si segnala solo che elementi di questo tipo risultano ora correttamente funzionanti (si veda anche il paragrafo 6.3). L'analisi delle difficoltà incontrate nello svolgimento del lavoro di Cinzia ha portato ad individuare due tipi di problemi gravi, che verranno di seguito approfonditi.

1. Difficoltà con l'elemento HTML `<select>`.
2. Difficoltà con i "tabs" nelle pagine relative alla sezione "Modifica Organizzazione"¹.

2.1. Problema con `<select>`

Normalmente, gli elementi attivi (cliccabili) di una pagina sono utilizzabili dal terminale nel seguente modo:

1. CINZIA guarda l'elemento cliccabile
2. IL SISTEMA evidenzia l'elemento cliccabile o un gruppo di elementi nell'intorno
3. IL SISTEMA visualizza strumenti per effettuare il "clic" o altre operazioni quali lo spostamento da un elemento all'altro nel caso ne siano evidenziati molteplici (in sostanza, la pagina web viene "aumentata" con strumenti atti a compiere delle operazioni sugli elementi attivi nel fuoco dell'attenzione)
4. CINZIA decide quale operazione effettuare controllando con lo sguardo gli strumenti aggiuntivi messi a disposizione dal sistema

Con gli elementi creati per mezzo del marcatore `<select>` le fasi 3 e 4 della sequenza non si attuano. In altri termini, le voci presenti negli elenchi non sono riconosciute come "cliccabili" dal sistema: ogni volta ne restano evidenziate diverse, senza che a ciò possano seguire risultati positivi.

Nel sito, spesso, i selettori sono usati accompagnati da funzioni javascript, per creare widget come quello in Fig. 1 (ove il nome "Luca Rossi" è fittizio e ripetuto solo per oscurare i nomi reali).

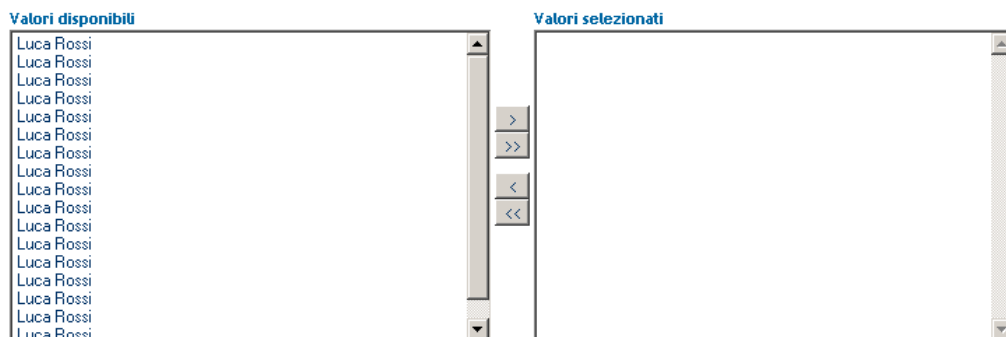


Fig. 1 - Esempio d'uso di `<select>` nel sito.

¹ Si veda il capitolo 4.1 per la descrizione di come raggiungere questa schermata.

➤ Non è comunque l'utilizzo di javascript a creare problemi, né i fogli di stile utilizzati, ma il selettore stesso, come emerso inequivocabilmente dai test.

2.2. Problema con i tabs

Per “tabs” si intende comunemente un oggetto grafico in grado di realizzare la metafora dello schedario, per cui un utente può cambiare il contenuto della pagina visualizzata cliccando su linguette (“tabs”). Tale meccanismo è stato implementato nel sito mediante javascript, in particolare attraverso la libreria *fabtabulous* [6]. Essa, grazie alla combinazione di fogli di stile e javascript, consente di interpretare i link interni associati alle linguette, cioè riferimenti verso blocchi di codice HTML univocamente identificati all'interno del medesimo documento, in modo che i contenuti di tali blocchi si alternino all'interno pagina (escludendosi quindi vicendevolmente) sulla base della linguetta che viene cliccata.

Nel sito, il meccanismo dei tabs è controllato dalle linguette visibili in Fig. 2. Tali linguette non risultano però cliccabili dal terminale: anche in tale contesto, con riferimento alla sequenza descritta nel paragrafo 2.1, il sistema non risponde coi passi 3 e 4 alle azioni di Cinzia.

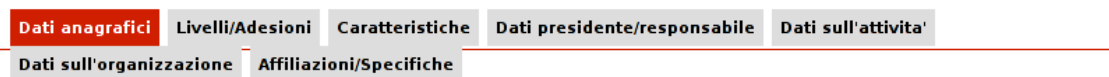


Fig. 2 – I tabs.

Dai test effettuati è emerso che i bottoni vengono percepiti come cliccabili dal sistema solo in seguito all'eliminazione di una riga presente nel foglio di stile situato in `modules/Associazioni/pnstyle/style.css` (percorso relativo alla directory della pagina principale: *index.html*). Tale riga è quella commentata nel seguente brano del CSS:

```
#tabs a {
/*      float: left;      -----RIGA DA ELIMINARE*/
padding: 6px 6px;
margin-left: 6px;
background-color: #DCDCDC;
text-decoration: none;

color: #000;
}
```

Anche risolto questo problema, però, il meccanismo dei tabs non funziona e un clic sulle linguette (sebbene formalmente fattibile) non sortisce alcun effetto positivo se non quello di portare ad un'attesa indeterminatamente lunga prima del ricaricamento della stessa pagina. Sebbene quello in oggetto sia il primo caso riscontrato in cui l'uso di javascript è legato ad un problema nell'uso del browser, è comunque il caso di notare che *fabtabulous* non è “unobtrusive”([7]): se javascript è disabilitato, il meccanismo rende sempre inaccessibili i contenuti.

3. La soluzione adottata

3.1. Problema con <select>

La funzione del selettore (soprattutto quando usato come in Fig. 1), è di definire l'appartenenza di una o più voci ad un insieme. La medesima funzione può essere svolta da una lista di voci con checkbox: le

voci segnate appartengono ad un insieme, da cui quelle non segnate sono escluse². Dunque, la proposta, accolta e implementata, era di sostituire i selettori con delle liste di checkbox.

La soluzione adottata consente inoltre di nascondere all'interno della pagina le diverse opzioni, richiamabili attraverso un apposito bottone, così da ridurre lo spazio occupato quando non è necessario effettuare una selezione (Fig. 3, ove le voci sono state cancellate a mano per ragioni di riservatezza).

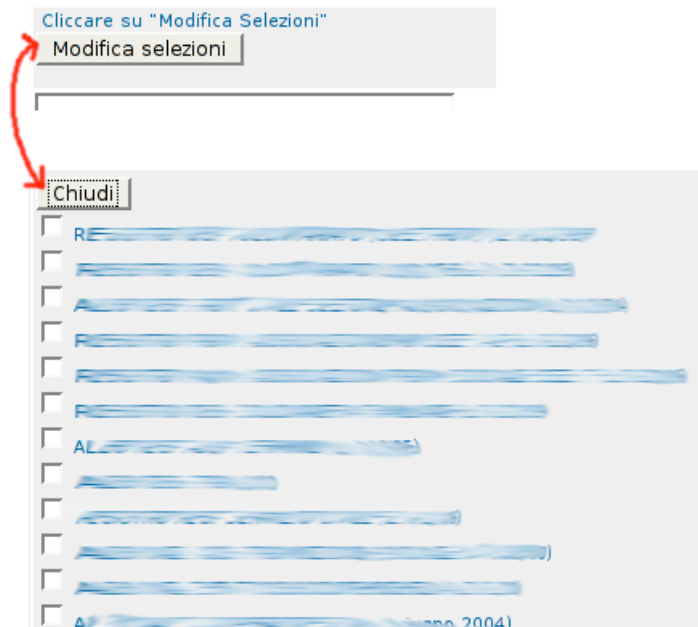


Fig. 3 – Sostituto del selettore. Il clic sopra un bottone porta nella situazione indicata dalla freccia rossa

3.2. Problema con i tabs

La soluzione proposta era di eliminare la riga del foglio di stile causa della mancata funzionalità delle linguette e di sostituire la libreria *fabtabulous* con un'altra, che garantisse il funzionamento del meccanismo anche nel caso di problemi con javascript. Si era effettivamente testata con successo una libreria alternativa [8]; nell'appendice (paragrafo 6.1) è riportato il codice di una pagina di test elementare, il cui meccanismo a schedario funziona anche sul terminale di Cinzia.

La soluzione adottata consiste invece semplicemente nella possibilità di disabilitare o abilitare il meccanismo dei tabs, attraverso lo strumento in Fig. 4. Quando i tabs vengono disabilitati, i contenuti delle diverse sezioni vengono contemporaneamente visualizzati. I bottoni, ancora una volta, non sono percepiti come cliccabili, ma almeno le informazioni, che non sono in quantità scoraggiante, possono essere integralmente accessibili scorrendo la pagina.



Fig. 4 – Soluzione al problema dei tabs

² Il ragionamento regge anche nel caso di una selezione singola ammissibile (uso classico di `<select>` privo dell'attributo "multiple"), sebbene si possa osservare che, in tal caso, un *radio button* catturerebbe di per sé il vincolo di unicità, senza richiedere ulteriori verifiche sul numero di scelte espresse.

4. Modalità operative

➤ Per verificare le soluzioni proposte occorre accedere al sito [4] con le credenziali di Cinzia.

Gli altri utenti, infatti, mantengono le vecchie modalità di interazione.

La sostituzione del selettore ha riguardato molte parti del sito, mentre la questione dei tabs investiva solo la sezione “Modifica Organizzazione”.

4.1. Come raggiungere “Modifica Organizzazione”

Dalla pagina di benvenuto, subito dopo aver fatto il login, si clicca su “Elenco Organizzazioni” (Fig. 5).

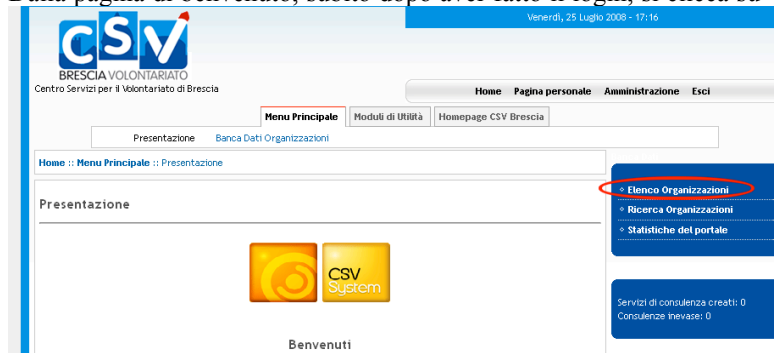


Fig. 5 – Il link da seguire è cerchiato in rosso

Appare a questo punto una lista di enti (Fig. 6).

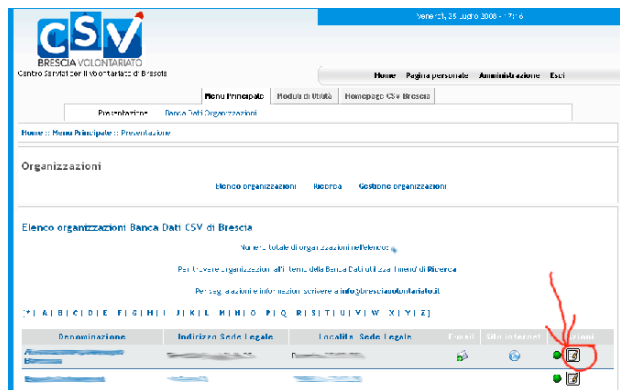



Fig. 6 – L’elemento da cliccare è evidenziato dalla freccia rossa

Nella colonna all’estrema destra ci sono elementi cliccabili a forma di taccuino  attraverso i quali si accede alla sezione di interesse.

5. Conclusioni e sviluppi futuri

Cinzia può ora portare a termini i compiti che le erano preclusi dai problemi oggetto di questa relazione.

Permangono problemi di usabilità, alcuni imputabili al software del sistema in uso, altri risolvibili o raggiungibili esaminando pazientemente il lavoro di Cinzia e individuando possibili soluzioni ad hoc, pertinenti esclusivamente al sito [4]. Un esempio del primo tipo riguarda l’impossibilità di esplorare

l'albero dei file, per es. in seguito al clic su bottoni generati attraverso il marcatore `<input>` con attributo `type="file"`. Un esempio del secondo genere riguarda il layout del sito che, nel complesso, costringe a un uso non proprio occasionale delle barre di scorrimento. Chiaramente, queste tipologie di problemi non sono disgiunte: il problema delle dimensioni riguarda molti siti, in quanto è il sistema che costringe la navigazione entro una finestra di dimensioni ridotte.

Ribadito che attualmente i problemi esclusivamente imputabili al sito non sono mai (fino a prova contraria) tali da impedire lo svolgimento del lavoro di Cinzia, è chiaro che, nella prospettiva di uno studio più radicale del sistema in uso, alcune modifiche che comportino dispendio di energie da parte degli sviluppatori del sito potrebbero essere evitate agendo a monte, e con migliori risultati, sul software del terminale.

Per ciò che concerne in maggior dettaglio i problemi fin qui risolti, si propongono due osservazioni.

- Come già scritto, il meccanismo di tabs potrebbe essere sostituito con uno *veramente* funzionante, evitando di “srotolare” in una sola pagina tutti i contenuti, o mantenendo almeno la funzione dei link interni.
- Potrebbe essere introdotto un meccanismo per selezionare contemporaneamente tutte le checkbox, in modo da rendere meno dispendioso per Cinzia il “caso medio³” di selezione delle voci. Si è già testato con successo sul terminale un semplice meccanismo che fa uso di javascript per l'opzione “seleziona tutti” (la pagina è riportata tale e quale, senza alcuna pretesa di generalità ed esemplarità della codifica, al paragrafo 6.2, in appendice).

³ Tale “caso medio” è inteso in senso puramente matematico e non deriva da una vera analisi del lavoro svolto in media da Cinzia con lo strumento in questione. Vale la pena di osservare però che nella Fig. 1 il caso di selezione contemporanea di tutte le voci è previsto (bottone con la doppia freccia). L'estensore della presente relazione è convinto di aver fatto menzione di detto meccanismo in un colloquio con un dipendente di Metarete, ma ha poi omesso di tradurlo in scritto durante lo scambio di e-mail che a tale colloquio ha fatto seguito, e questa è l'unica cosa certa. Va anche detto che Cinzia non ha al momento (oltre venti giorni dall'ultimo test effettuato insieme) segnalato casi in cui si presentino difficoltà sotto questo aspetto.

6. APPENDICE

Le pagine riportate nei prossimi due paragrafi sono esattamente quelle provate sul browser del My Tobii. Si trovano raccolte nei file *par6.1.zip* (insieme alle librerie necessarie) e *par6.2.html*.

6.1. Pagina di test con libreria alternativa a fabtabulous

Una qualità della libreria di seguito utilizzata è quello di rendere javascript utile ma non necessario per la completa visualizzazione delle informazioni. Infatti, nel caso in cui javascript fosse disabilitato, i contenuti corrispondenti alle diverse linguette verrebbero semplicemente “srotolati” di seguito e in questo caso le linguette rimanderebbero (in quanto link interni) a sottoparti del documento. Ciò, comunque, col terminale di Cinzia non avviene perché non necessario, in quanto il meccanismo fornito dalla libreria funziona pienamente con javascript.

Le librerie per le tabs utilizzate sono reperibili da [8] (da cui deriva in sostanza il codice che segue) e [9].

```
<HTML>
<head>
  <title></title>
  <!--<link rel="stylesheet" href="screen.css"/>-->
  <script src="prototype.js"></script>
  <script src="control.tabs.2.1.1.js"></script>

  <style></style>
</head>
<body>

  <!-- begin example 2 code -->
  <ul id="tabs_example_two" class="subsection_tabs">

    <li><a class="active" href="#a">One</a></li>
    <li><a class="" href="#b">Two</a></li>
    <li><a class="" href="#c">Three</a></li>
    <li><a href="#d">Four</a></li>

  </ul>
  <div style="" id="a"><p>This example demonstrates scripting a Control.Tabs instance. Try clicking the
links below, or changing the select box.</p></div>

  <div style="display: none;" id="b"><p>I am the second tab.</p></div>
  <div style="display: none;" id="c"><p>I am the third tab.</p></div>
  <div style="display: none;" id="d"><p>I am the fourth tab.</p></div>
<a href="" id="tabs_example_two_first" style="margin-right: 10px;"><<</a> <br>
<a href="" id="tabs_example_two_previous" style="margin-right: 10px;"><<</a>
<select id="tabs_example_two_select" style="margin-right: 10px;">

  <option value="a">One</option>
  <option value="b">Two</option>
  <option value="c">Three</option>
  <option value="d">Four</option>
</select><br>
<a href="" id="tabs_example_two_next" style="margin-right: 10px;">>></a> <br>
<a href="" id="tabs_example_two_last">>></a>
  <script>
    var tabs_example_two = new Control.Tabs('tabs_example_two',{
      afterChange: function(new_container){
        $A($('tabs_example_two_select').options).each(function(option,i){
          if(option.value == new_container.id){

            $('tabs_example_two_select').options.selectedIndex = i;
                                                                    throw $break;
          }
        });
      }
    });
    $('tabs_example_two_select').observe('change',function(){
      tabs_example_two.setActiveTab($('tabs_example_two_select').options[$('tabs_example_two_select').options.
selectedIndex].value);
    });
    $('tabs_example_two_first').observe('click',function(event){
      this.first();
      Event.stop(event);
    }.bindAsEventListener(tabs_example_two));
    $('tabs_example_two_previous').observe('click',function(event){
      this.previous();
      Event.stop(event);
    }.bindAsEventListener(tabs_example_two));
    $('tabs_example_two_next').observe('click',function(event){
      this.next();
      Event.stop(event);
    }.bindAsEventListener(tabs_example_two));
    $('tabs_example_two_last').observe('click',function(event){
      this.last();
    });
  </script>
</body>
</HTML>
```

```

                Event.stop(event);
            }.bindAsEventListener(tabs_example_two));
    </script>
    <!-- end example 2 code -->
<br>
</body>
</HTML>

```

6.2. Pagina di test “seleziona tutti”

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<HTML lang="it">
<head>
<title>Tutte insieme</title>
<body>

<form name="modulo">
<input type="checkbox" name="colori" value="Rosso"> Rosso<br>
<input type="checkbox" name="colori" value="Giallo"> Giallo<br>
<input type="checkbox" name="colori" value="Verde"> Verde<br>
<input type="checkbox" name="colori" value="Blu"> Blu<br><br>
<input type="button" value="Seleziona/Deseleziona tutto" onclick="SelezTT1()">
</form>

<TABLE id="tabella">
<TR>
<td><input id="pn_csv_formation_public_flag" name="Marco" value="1" type="checkbox">Marco</td><td><input
id="pn_csv_formation_public_flag" name="Luca" value="1" type="checkbox">Luca</td>
</tr>
<td><input id="pn_csv_formation_public_flag" name="Giovanni" value="1" type="checkbox">Giovanni
</td><td><input id="pn_csv_formation_public_flag" name="Matteo" value="1" type="checkbox">Matteo</td>
</tr><tr>
<td><input id="pn_csv_formation_public_flag" name="Isacco" value="1" type="checkbox">Isacco </td>
<td><input id="pn_csv_formation_public_flag" name="Ismaele" value="1" type="checkbox">Ismaele</td>
</tr><tr>
<td><input id="pn_csv_formation_public_flag" name="Filiberto" value="1" type="checkbox">Filiberto</td>
<td><input id="pn_csv_formation_public_flag" name="Claudio" value="1" type="checkbox">Claudio</td>
</tr><tr>
<td><input id="pn_csv_formation_public_flag" name="Mario" value="1" type="checkbox">Mario</td>
<td><input id="pn_csv_formation_public_flag" name="Andrea" value="1" type="checkbox">Andrea</td>
</tr>
</table>
<input type="button" value="Seleziona/Deseleziona tutto" onclick="SelezTT2()">

<script type="text/javascript">
function debug(msg){
    debugwindow=window.open('','debug',
    'toolbar=1,resizable=1,width=500,height=500,scrollbars=yes');
    debugwindow.document.writeln(msg+'<br>');
}

function SelezTT2()
{
    var i = 0;
    var figli;
    var righe = document.getElementById('tabella').rows;
    for (i=0; i<righe.length; i++)
    {
        riga=righe[i];
        inputs=riga.getElementsByTagName("input");
        for (var j=0; j<inputs.length; j++)
        {
            if(inputs[j].type == "checkbox")
            {
                inputs.item(j).checked = !(inputs.item(j).checked);
            }
        }
    }
}

function SelezTT1()
{
    var i = 0;
    var modulo = document.modulo.elements;
    for (i=0; i<modulo.length; i++)
    {
        if(modulo[i].type == "checkbox")
        {
            modulo[i].checked = !(modulo[i].checked);
        }
    }
}

</script>
</body></HTML>

```

6.3. Elementi esplicitamente testati

Nella seguente tabella si riportano i risultati di alcuni test svolti su singoli elementi HTML, avulsi dal contesto del sito [4] ove non diversamente specificato. Questo significa che essi sono stati testati senza appoggiarsi al framework in uso nel sito, basato su javascript e CSS. I risultati di vari test intermedi svolti facendo uso di parti di fogli di stile o javascript non sono riportati perché generalmente non significativi (l'unico caso in cui si è rilevata una significativa influenza di singoli elementi del framework sulla bontà del funzionamento è quello dei tabs, trattato nel paragrafo Ø).

Elemento	Risultato
checkbox – anche con javascript per (de)selezione multipla	OK
radio-button	OK
unordered list (e)	OK
menu a tendina	OK
<input> con e senza attributo disabled="disabled"	OK
<textarea> con e senza attributi id, cols, rows, name impostati	OK
<select> (isolato e dentro al sito, con e senza CSS, con e senza javascript)	SEMPRE FALLIMENTARE
<input> con attributo type="file"	IMPOSSIBILE esplorazione del file system (la resa HTML è comunque corretta, il bottone funziona, non è un problema del browser)

Bibliografia

- [1] <http://www.bresciavolontariato.it/>
- [2] <http://www.tobii.com/>
- [3] <http://srlabs.it/contenuti/mytobii.HTML>
- [4] <http://www.csvsystem.it/bs/index.pHTML>
- [5] <http://www.metarete.it/>
- [6] <http://tetlaw.id.au/view/blog/fabtabulous-simple-tabs-using-prototype/>
- [7] http://en.wikipedia.org/wiki/Unobtrusive_JavaScript
- [8] <http://livepipe.net/control/tabs>
- [9] <http://www.prototypejs.org/> (Sito del framework JavaScript Prototype)

Indice

SOMMARIO	1
1. INTRODUZIONE	1
1.1. Il contesto del lavoro	1
1.2. Struttura del lavoro	1
2. IL PROBLEMA AFFRONTATO	2
2.1. Problema con <select>	2
2.2. Problema con i tabs	3
3. LA SOLUZIONE ADOTTATA	3
3.1. Problema con <select>	3
3.2. Problema con i tabs	4
4. MODALITÀ OPERATIVE	5
4.1. Come raggiungere “Modifica Organizzazione”	5
5. CONCLUSIONI E SVILUPPI FUTURI.....	5
6. APPENDICE.....	7
6.1. Pagina di test con libreria alternativa a fabtabulous	7
6.2. Pagina di test “seleziona tutti”	8
6.3. Elementi esplicitamente testati	9
BIBLIOGRAFIA	10
INDICE	11