

GABRIELE ROMANATO

jQuery: contatori sugli elementi HTML

La soluzione qui proposta per aggiungere dei contatori agli elementi HTML non è da considerarsi come la soluzione definitiva al problema del mancato supporto ai contatori CSS in Internet Explorer 7 ed inferiori. Infatti manca di quell'automazione necessaria per non dover specificare ogni volta la procedura per ottenere il livello di annidamento adeguato. Gli esempi che seguono vogliono essere uno stimolo per ulteriori test e ricerche.

Intestazioni

Partiamo dalle intestazioni HTML (h1, h2 eccetera). Questi elementi sono abbastanza semplici da gestire perchè hanno già nel nome dell'elemento un'indicazione del livello a cui appartengono, fermo restando il fatto che questi elementi dovrebbero essere usati in modo semantico:

```
var level2 = 0;
var level3 = 0;
var level4 = 0;

$(' :header').each(function() {

    var $header = $(this);
    var text = $header.text();
    var tag = $header[0].tagName.toLowerCase();
    var levelTag = tag.charAt(1);

    switch(levelTag) {

        case '2':

            level2 += 1;
            $header.text(level2 + '. ' + text);
```

```

        break;
    case '3':

        level3 += 1;
        $header.text(level2 + '. ' + level3 + ' ' + text);

        break;
    case '4':
        level4 += 1;
        $header.text(level2 + '. ' + level3 + '. ' + level4 +
' ' + text);

        break;
    default:
        break;

}

});

```

Abbiamo di volta in volta incrementato tre contatori per tenere traccia del livello di annidamento delle sezioni e sottosezioni di una pagina, così da avere 1, 1.1, 1.1.1 e così via. Tramite il metodo JavaScript `charAt()` estraiamo il secondo carattere che contiene il numero indicante il livello dell'intestazione prendendolo direttamente dal nome del tag.

Liste

Il caso delle liste è un pò più complicato data l'assenza di riferimenti. L'unica cosa certa è che una voce di lista può contenere un'altra lista, ossia

può avere una struttura di questo tipo:

```
<ul id="test">
  <li>a</li>
  <li>b</li>
  <li>c
    <ul>
      <li>d</li>
      <li>e</li>
      <li>f
        <ul>
          <li>g</li>
          <li>h</li>
        </ul>
      </li>
    </ul>
  </li>
</ul>
```

Su ogni voce di lista dobbiamo incrementare due tipi di contatori:

1. un contatore sull'elemento più esterno (qui `test`) che chiameremo `counter`
2. contatori interni sugli elementi annidati usando gli indici dei cicli `each()`

Ovviamente questi contatori vanno concatenati per formare una struttura del tipo 1, 1.1, 1.1.1 e così via:

```
var counter = 0;

$('#test > li').each(function(i) {

  var level = i + 1;
  counter++;
  var $li = $(this);
```

```

    $('<span/>').text(level + '. ').
    prependTo($li);

    if($('ul:first', $li).length) {

        $('ul:first > li', $li).each(function(j) {

            $('<span/>').text(counter + '.' +
(j+1)).
            prependTo($(this));

            if($('ul:first', $(this)).length) {

                $('ul:first > li',
$(this)).each(function(k) {

                    $('<span/>').text(counter + '.' + (j+1) + '.' + (k+1)).
                    prependTo($(this));

                });

            }

        });

    }

});

```

Gli indici dei cicli `each()` partono da 0, quindi dobbiamo sempre aggiungere 1. La soluzione funziona ma è prolissa e poco efficace. Meglio sarebbe utilizzare una funzione ricorsiva navigando il DOM. Se avete una soluzione più efficace, non esitate a commentare questo post in modo da aumentarne la qualità e l'utilità.

Potete visionare l'esempio finale in [questa pagina](#).