

Tecniche di debug

Usare gli alert per scoprire gli errori di scripting

Può capitare che tutto lo script sia esatto per il browser (nel senso che non restituisca errore), eppure non dia il risultato voluto. Bisogna allora individuare il punto in cui siamo incappati in un **bug** che mina la validità del nostro programma (questa volta un **errore concettuale**, non errore di sintassi).

Possiamo allora utilizzare in vario modo gli **alert**, per sorvegliare il "buon andamento" dello script.

Possiamo usare istruzioni di uscita a monitor e quindi, in JavaScript, il metodo alert per:

Usiamo l>alert per	Esempio
Visualizzare il contenuto di una variabile	<pre>numero=6; numero=numero-5; alert (numero);</pre>
Scoprire in quale caso del controllo condizionale ci troviamo	<pre>if (document.all) { alert("IE"); } else { alert("non IE"); }</pre>
Sorvegliare il buon andamento di un ciclo	<pre>alunni = new Array ("Aldo", "Giovanni", "Giacomo"); for (n=0;n<alunni.length;n++) { alert(n); alert(alunni[n]); }</pre>

Tutti questi utilizzi del metodo alert ci posso venire in aiuto per sorvegliare il "buon andamento" del codice e per scoprire a che punto del programma siamo arrivati.

Tratto da capitolo conclusivo della guida:

47. [Tecniche di debug](#)

Usare gli alert per scoprire gli errori di scripting

Si veda anche:

46. [L'Indentazione](#)

Rendere più leggibile il codice JavaScript

Diversità dei DOM dei browser

La difficoltà della programmazione JavaScript sta principalmente nella **diversità dei DOM** dei browser: vengono, ad esempio, trattati in modo diverso i **livelli** descritti attraverso i fogli di stile.

Prendiamo ad esempio la pagina seguente, realizzata con **livelli** descritti attraverso i fogli di stile incorporati (sezione creata con tag **div** nella pagina):

```
<html>
<head>
<style type="text/css">

    #mioLiv {
        position:absolute;
        background-color:orange;
        width:300px; height:300px; border:1px;
        visibility:show;
    }

</style>
<script>
function nascondi(){
    // istruzioni per nascondere il livello
}
</script>
</head>
<body>
<!-- chiamata diretta a una funzione JavaScript-->
<a href="javascript: nascondi();">nascondi il livello</a><br>
<div id="mioLiv"><br><br><br></div>
</body>
</html>
```

per assegnare l'invisibilità al livello identificato come **mioLiv** dovremmo scrivere:

- Con Netscape 4: **document.layers**["mioLiv"].visibility= "hide";
- Con Internet Explorer 4: **document.all**["mioLiv "].style.visibility= "hidden";
- Con Internet Explorer 6: **document.getElementById**["mioLiv "].style.visibility= "hidden";

Per approfondire:

48. [L'istruzione "eval" - I](#)
Come aggirare le differenti interpretazioni dei Browser
49. [L'istruzione "eval" - II](#)
Come aggirare le differenti interpretazioni dei Browser