

Il debugger: trovare gli errori

<http://javascript.html.it/guide/lezione/873/il-debugger-trovare-gli-errori/>

Vi capiterà di scrivere delle porzioni di codice più o meno complesse, e vi capiterà senz'altro di fare degli errori. Se dimenticate un punto e virgola, se non chiudete le virgolette, o dimenticate una parentesi... in tutti questi casi state commettendo un errore e non vi funzionerà più nulla. Anzi: Internet Explorer visualizzerà un messaggio di errore in basso a sinistra (una specie di triangolino giallo).

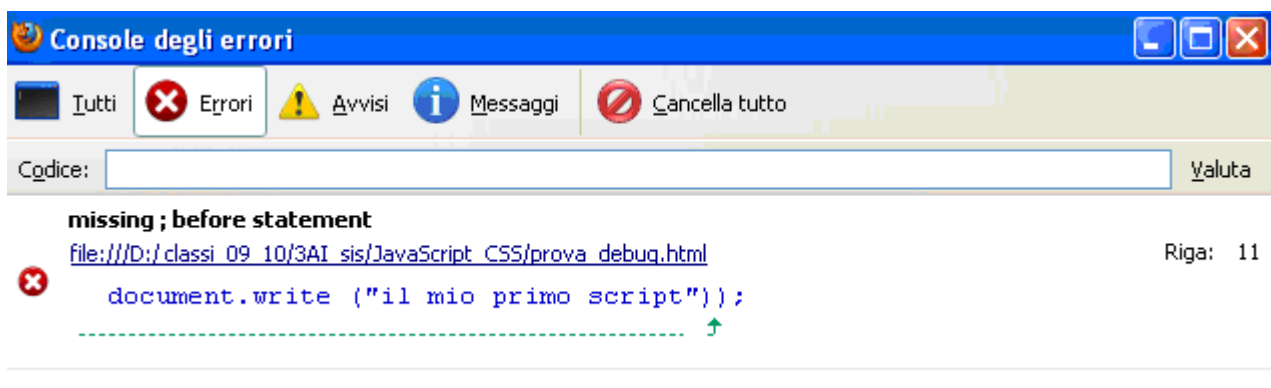
Come fare quindi? Niente paura: tutti i principali browser hanno incorporato un **debugger**, uno strumento cioè che vi permette di individuare i "bugs" (letteralmente gli insetti): i vostri errori nel programma.

Proviamo ad esempio a scrivere **white** al posto di **write** oppure ad inserire una parentesi di troppo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>JavaScript con errore</title>
</head>
<body>
<script language="JavaScript">
//Visualizza la scritta "il mio primo script"
    document.write("il mio primo script");
</script>
</body>
</html>
```

Riconoscendo gli errori di sintassi, il browser ve li segnalerà. Vediamo ora come attivare il debugger e individuare gli errori.

Se visualizziamo la pagina di prima in Mozilla - Firefox, attivando il debugger integrato, otteniamo il seguente messaggio di errore:



A questo punto, se usiamo il Blocco note di Windows siamo costretti a contare le righe a mano, ma se usiamo qualche altro software un pò più evoluto (HomeSite, oppure HTML Kit - quest'ultimo è gratis come Notepad++) basterà selezionare l'opzione che visualizza il numero delle righe per visualizzare in tutta comodità la riga contenente l'errore (per lo più l'opzione viene azionata o disabilitata da un tasto con il simbolo convenzionale "#").

Grazie alle indicazioni del debugger possiamo individuare e correggere gli errori anche in codici molto più complessi di quello utilizzato nell'esempio.

Un'avvertenza: poiché browser diversi hanno caratteristiche differenti, ogni browser visualizzerà differenti messaggi di errore a seconda del codice.

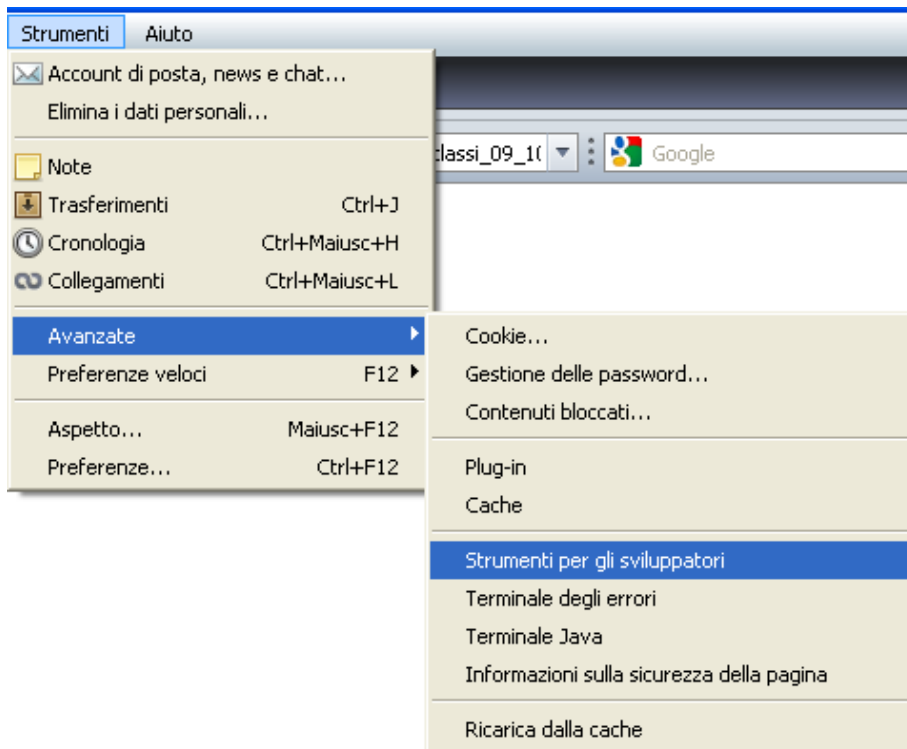
Se nella stragrande maggioranza dei casi i browser si comporteranno nello stesso modo, dovete prestare particolare attenzione in quei casi in cui elaborate codice JavaScript che riguarda i *livelli*, o in tutti quei casi in cui il DOM dei vari browser non coincida. Per ora abituatevi ad esaminare la stessa pagina con diversi browser: se non compaiono messaggi di errore strani tutto va bene, in caso contrario preoccupatevi di individuare e correggere l'errore.

Curiosità: Sulla parola "bug" ("insetto") c'è un aneddoto divertente. Il 9 settembre del 1945 Grace Murray Hopper (ufficiale e matematica di gran valore) che prestava servizio in Virginia presso la marina militare degli Stati Uniti stava cercando di trovare l'errore che inceppava il computer basato su un sistema Harvard Mark II, quando trovò un insetto che girovagava allegramente in mezzo ai circuiti e che era la causa del malfunzionamento. Da allora il termine "bug" entrò nell'informatica per indicare un errore di programmazione. Potete leggere tutta la vicenda cercando la Storia di Grace Murray Hopper.

In Firefox da Strumenti → Console degli errori

Mozilla Firefox	
Strumenti	Aiuto
Cerca sul web	Ctrl+K
Download	Ctrl+J
Componenti aggiuntivi	
Java Console	
Console degli errori	Ctrl+Maiusc+J
Informazioni sulla pagina	
Avvia Navigazione anonima	Ctrl+Maiusc+P
Cancella cronologia recente...	Ctrl+Maiusc+Canc
Opzioni...	

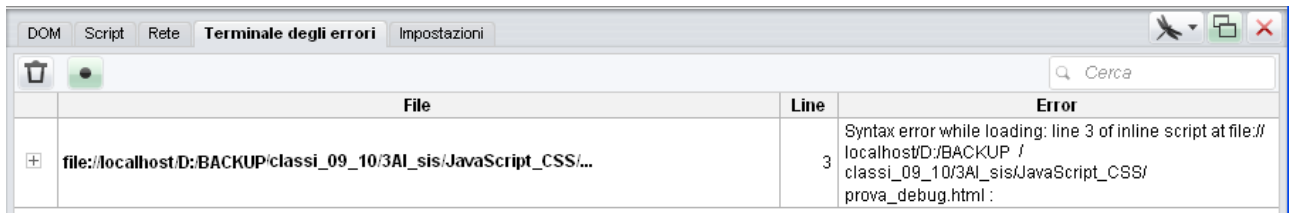
Le nuove versioni di Opera integrano Opera Dragonfly



<http://dev.opera.com/tools/>

Opera Dragonfly è un'applicazione JavaScript che consente di effettuare il debug di pagine Web locali o remote.

Selezionando Terminale degli errori



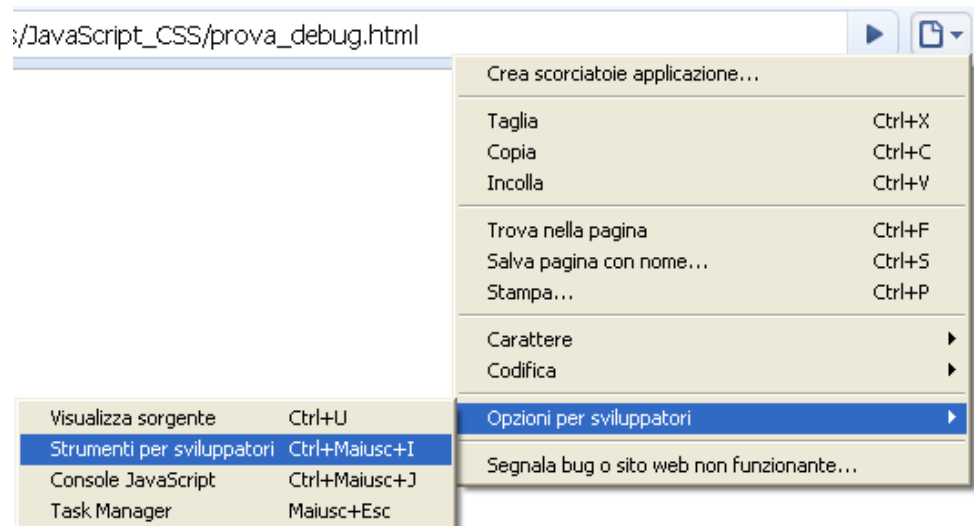
e con click sull'icona di espansione 

si possono avere maggiori dettagli sull'errore

```
Syntax error while loading: line 3 of inline script at file://localhost/D:/BACKUP/classi_09_10/3AI_sis/  
Expected token: ';'  
  document.write ("il mio primo script");  
-----^
```

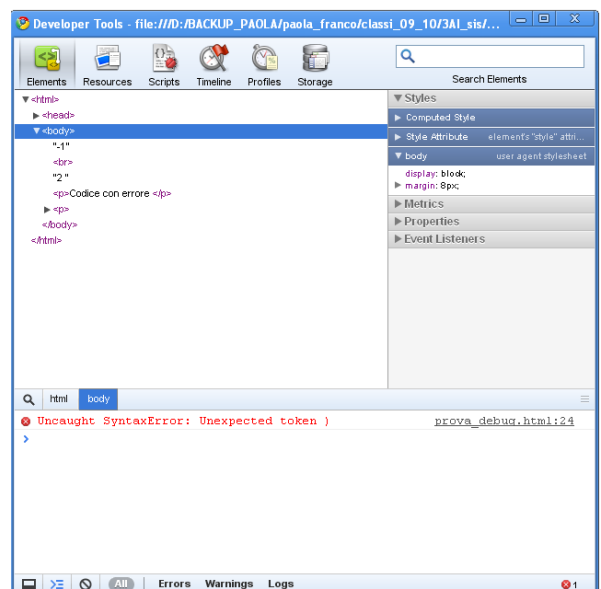
Nota: per utilizzare Opera Dragonfly è necessario che JavaScript sia abilitato.

In Google Chrome

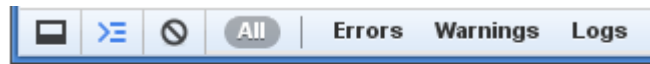


si apre
un ambiente di debugger:

Developer Tools



Potendo selezionare o deselegnare la visualizzazione della console che mostra gli errori



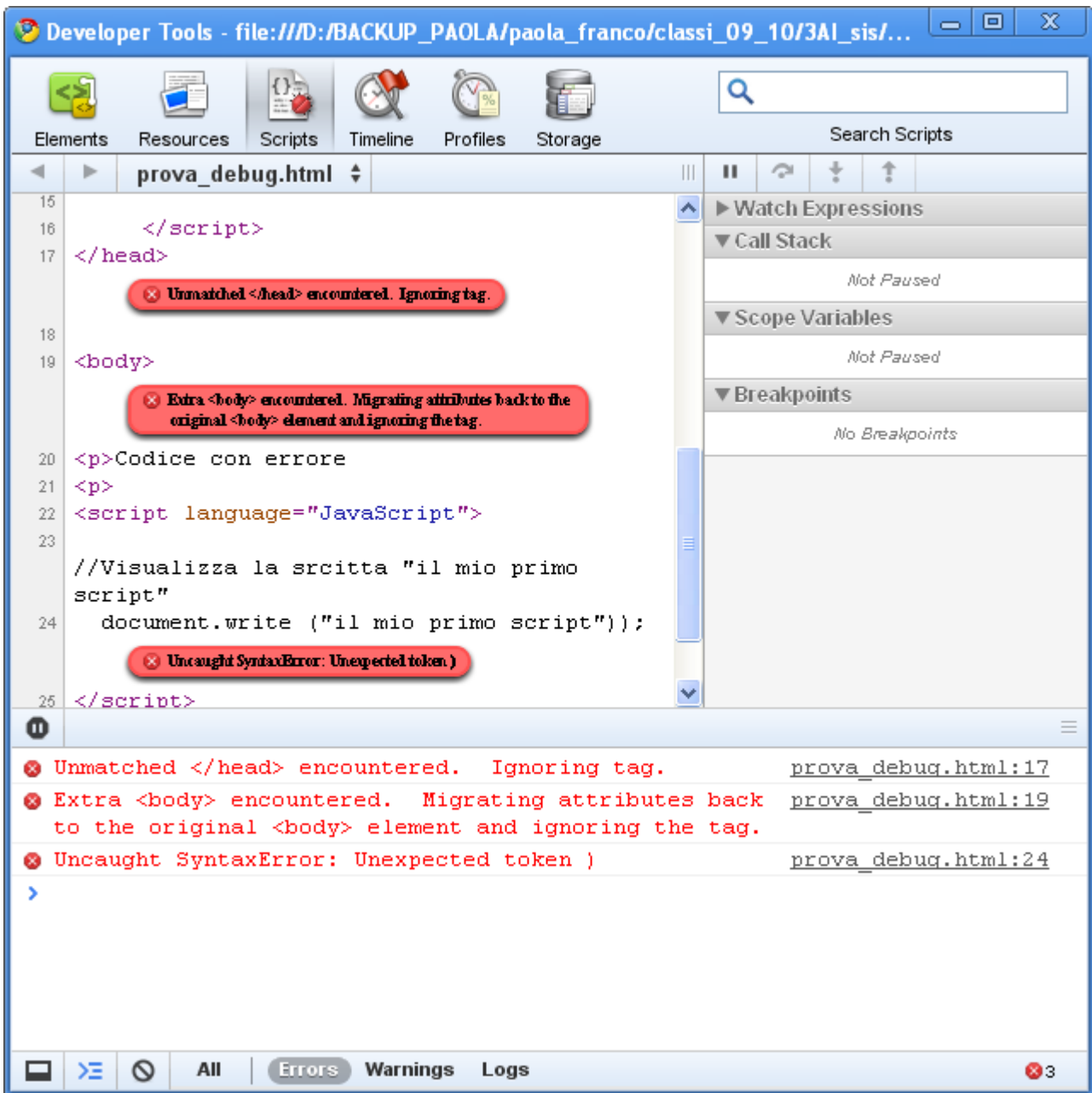
Hide console.

icona di colore nero se deselegnata



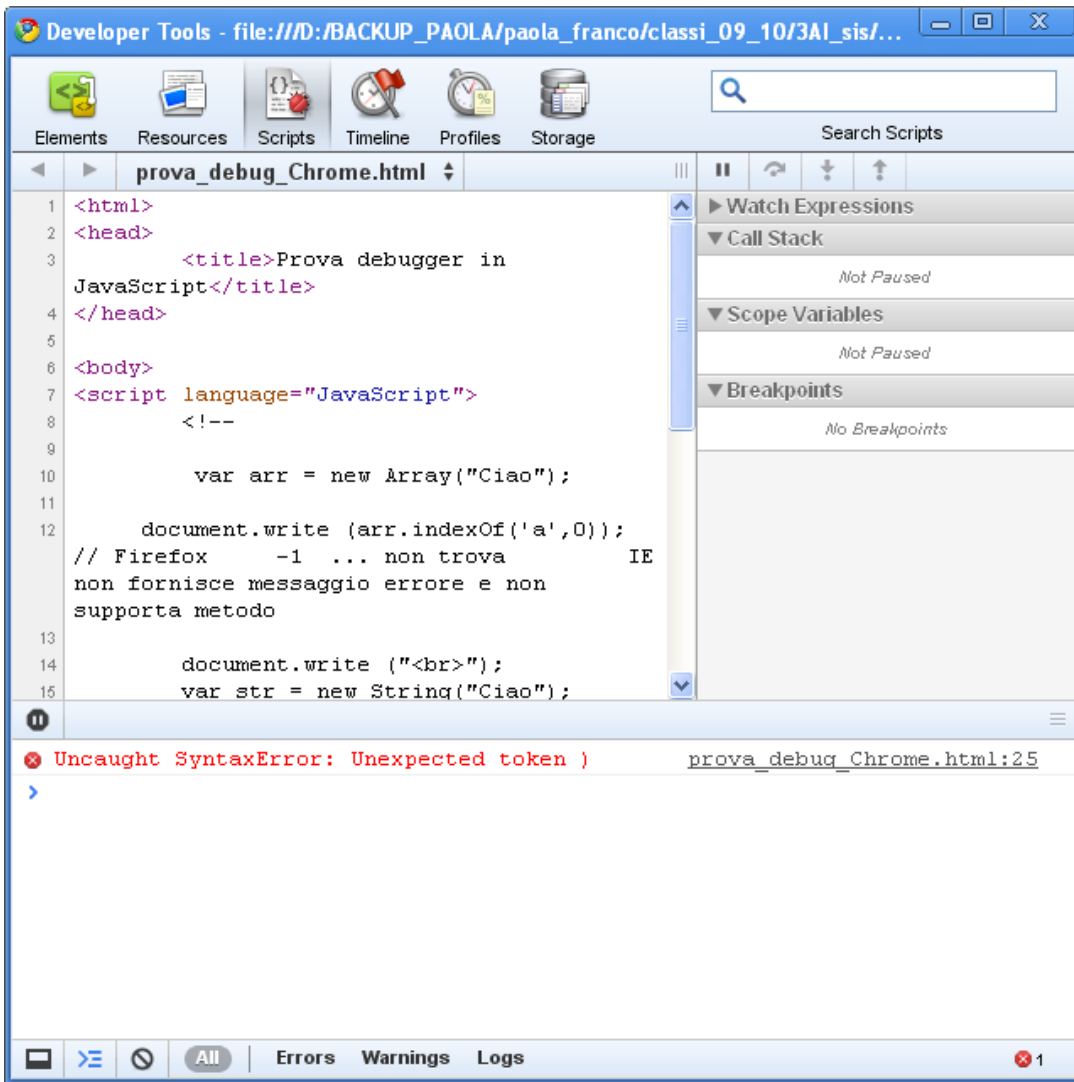
Show console.

Attivata l'icona **Scripts**, lanciare l'esecuzione della pagina



si noti che gli script nella sezione head generano messaggi di errore ignorati

Fasi: attivare l'icona **Scripts** - lanciare l'esecuzione della pagina (**script solo nel body**)



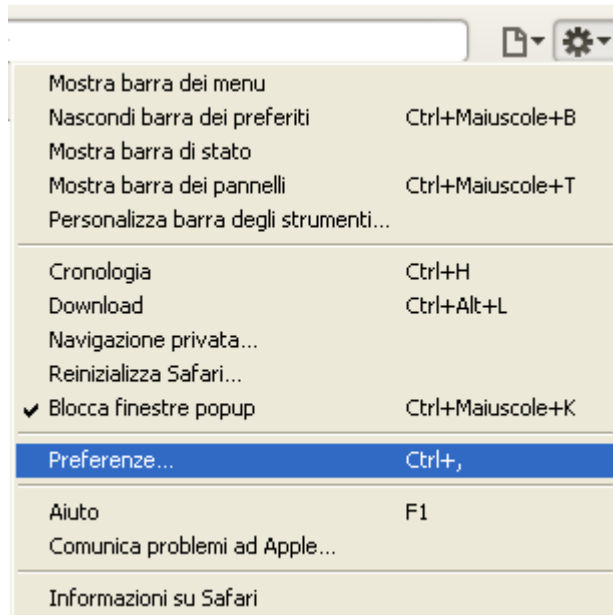
In IE

La versione 8 offre dei Developer Tools integrati nel browser (alcuni test hanno evidenziato che a volte ignora i breakpoint); per le versioni precedenti, se si dispone di Visual Studio si dispone anche di uno script debugger, oppure, in caso contrario, esiste Microsoft Script Debugger o altri che si possono installare e collegare a Internet Explorer.

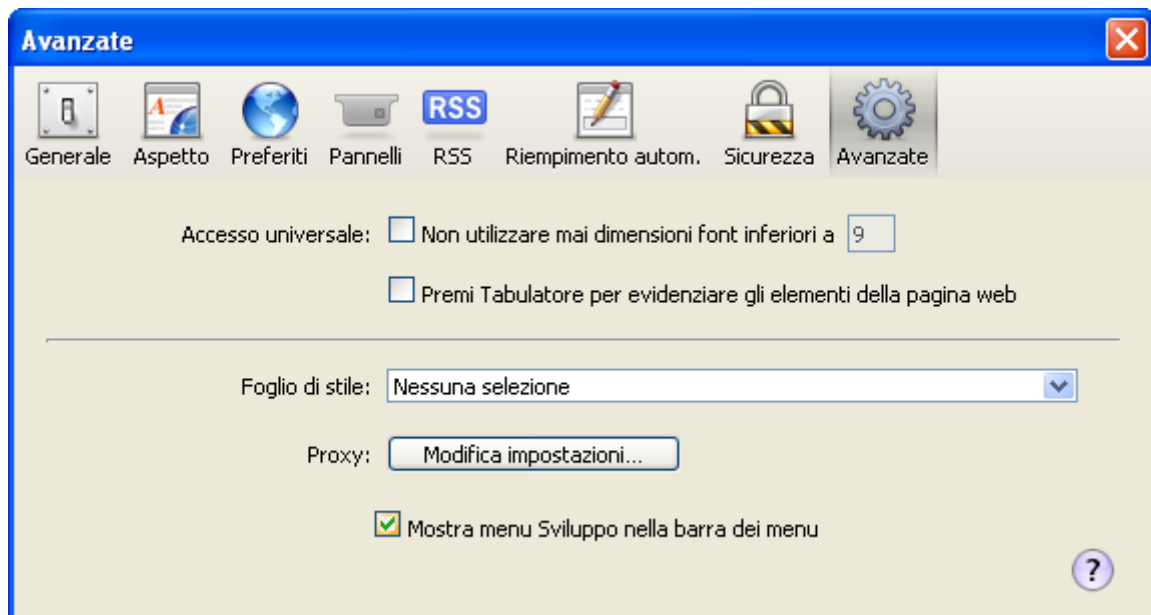
In Safari

Si può sfruttare sia il [debugger Drosera JS](#), sia un tool per l'esplorazione del DOM chiamato [WebInspector](#) (nelle versioni più recenti del browser il debugger è stato integrato proprio nel Web Inspector)

Fasi: da Preferenze



→Avanzate

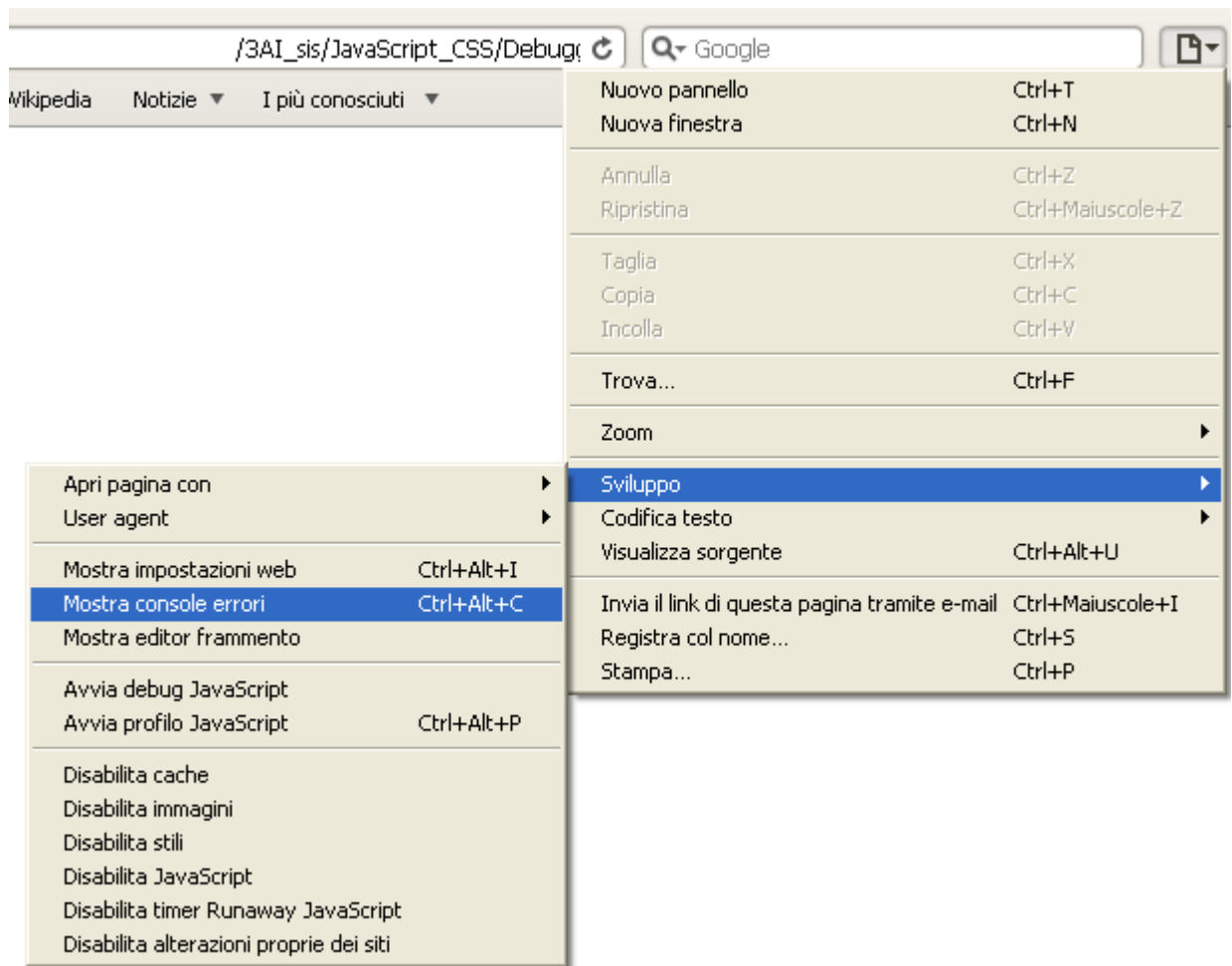


smarcare Mostra menu Sviluppo

Ora nell'elenco dei menù

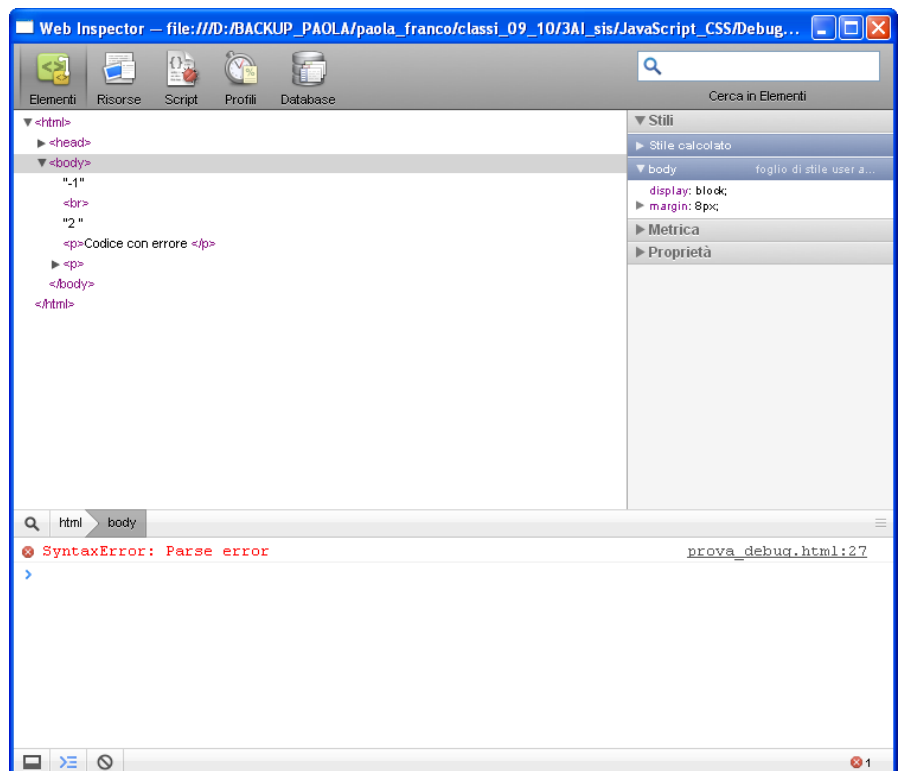


appare Sviluppo → Mostra console errori

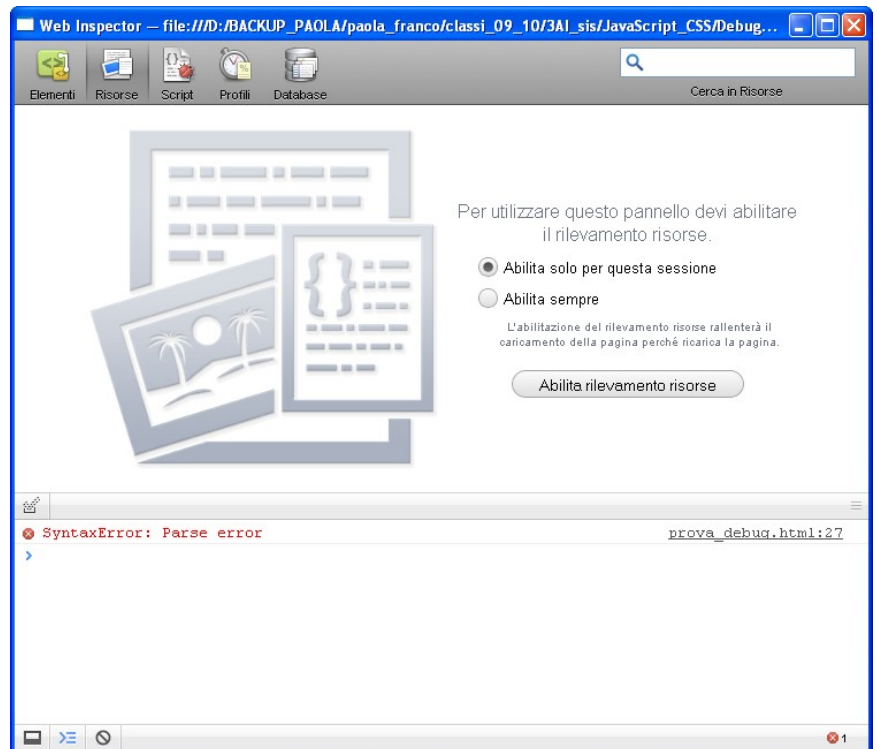


interfaccia molto simile

all'analogo Tool in Google

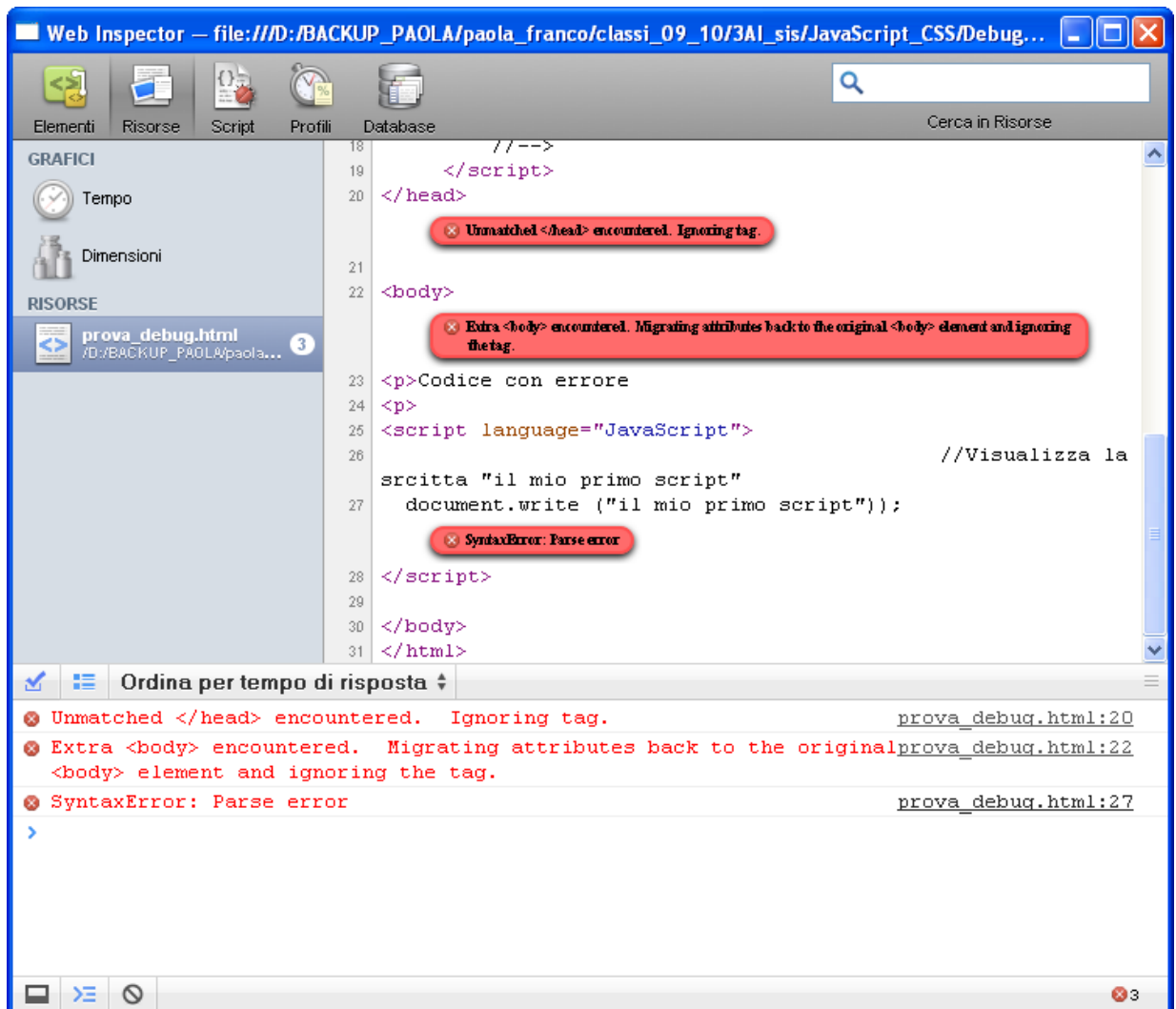


Fase: abilitare per questa sessione il rilevamento di risorse

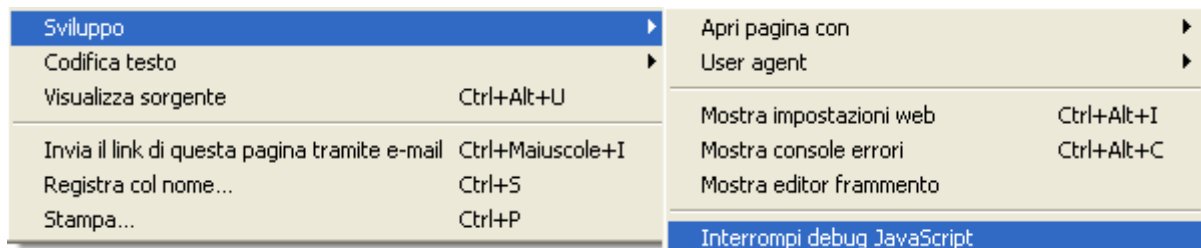


Cliccando su riga errata

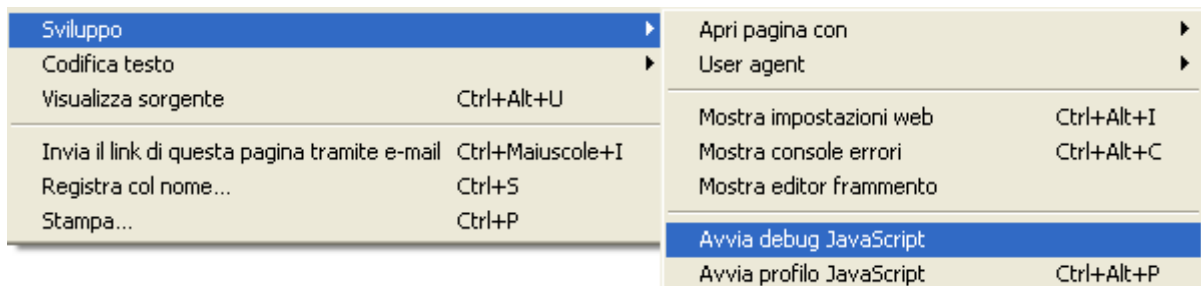
vengono evidenziati gli errori



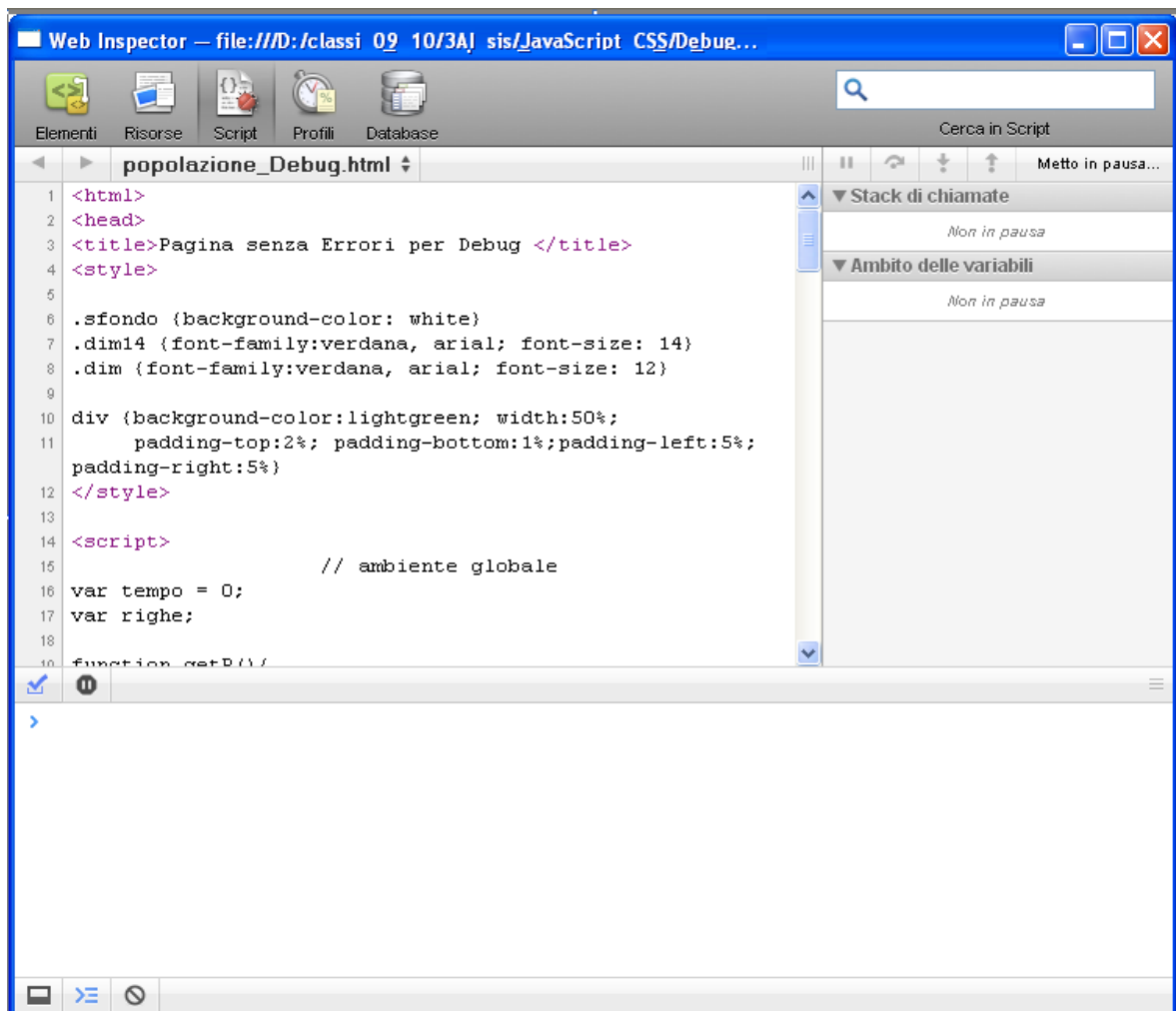
Per interrompere il debug JavaScript in Safari si segue il percorso:



Per avviare il debug JavaScript si segue il percorso:



potendo debuggare eseguendo *passo-passo*



e mettere in pausa l'esecuzione dello script per investigare valori di variabili e call-stack