

Javascript e temporizzazioni - timing

Ultimo aggiornamento giovedì 29 novembre 2007

Con Javascript è possibile eseguire alcune funzioni non in esecuzione ma dopo un determinato intervallo di tempo. I due metodi utilizzati (appartenenti al DOM HTML Window) sono:

- setTimeout() - esegue del codice dopo un determinato intervallo di tempo
- clearTimeout() - cancella il setTimeout() settato

setTimeout()

Sintassi: `var t=setTimeout(funzione, tempo*)` *(espresso in millisecondi: 2 secondi = 2 * 1000)

setTimeout() ritorna un valore che nel nostro esempio è immesso nella variabile 't', servirà nel caso si debba cancellare l'evento con clearTimeout().

Il primo parametro è una stringa che contiene uno statement o una chiamata a funzione.

Il secondo parametro è invece il tempo di attesa espresso in millisecondi.

Ecco un esempio di un messaggio dopo 5 secondi di attesa `var t=setTimeout("alert('5 secondi')",5000);`
`clearTimeout()`

Sintassi: `clearTimeout(setTimeout_variable)` esempio: `<html>`

`<head>`

`<script type="text/javascript">`

`var t`

```
function time(){
  setTimeout("alert()",5000);
}
```

```
function stoptime(){
  clearTimeout(t);
}
```

`</script>`

`</head>`

`<form>`

`<input type="button" value="conta" onClick="time()">
`

`<input type="button" value="stop" onClick="stoptime()">`

`</form>`

`</html>` L'esempio sopra mostra le due funzioni di timing: con il primo bottone è possibile avviare i 5 secondi prima della finestra di alert, conteggio che è possibile bloccare con il secondo bottone.