

In genere il metodo `alert()` è usato per verificare il valore delle variabili durante la fase di correzione degli errori (*debugging*), fase che può essere svolta anche con l'ausilio di appositi programmi per la verifica del codice. Il metodo `alert()` è associato all'oggetto `window` che all'interno del Document Object Model (*DOM*) del browser rappresenta l'oggetto di livello massimo¹.

Se la variabile non è valorizzata² appare comunque la finestra di dialogo, ma al posto del messaggio previsto dal metodo `alert` appare la scritta *undefined*.

Si noti, inoltre, la modalità con la quale si nasconde il codice JavaScript ai browser che non sono in grado di interpretarlo: dopo il tag iniziale `<script>` è inserita la sequenza HTML di apertura commento "`<!--`" e prima del tag di chiusura `</script>` è inserito il commento Javascript "`//`", seguito dalla sequenza HTML di chiusura del commento "`-->`". L'effetto sarà che i browser che supportano i tag Javascript eseguiranno le istruzioni previste, mentre quelli meno recenti ignoreranno tutto ciò che è compreso all'interno dei commenti HTML e non eseguiranno le istruzioni. Poiché Javascript non riconosce la sequenza HTML di chiusura dei commenti occorre inserire i caratteri "`//`" di commento Javascript prima della sequenza HTML.

Ovviamente, nel caso si usi Javascript, la sequenza di chiusura del commento è quella mostrata mentre sarà diversa nel caso in cui lo script è codificato con un altro linguaggio (ad es. nel caso di VBScript è un apice singolo in quanto questo è il simbolo usato da questo linguaggio per inserire i commenti).

Per quanto riguarda i commenti all'interno del tag *script* va ricordato che quelli costituiti da un'unica riga devono essere preceduti dai caratteri "`//`", mentre quelli composti da più righe sono racchiusi tra i caratteri "`/*`" e "`*/`", come riportato nel codice esemplificativo.

Al fine di fornire un **contenuto alternativo** ai browser che non supportano Javascript possono essere usati i tag `<noscript>`, da includere subito dopo il tag di chiusura `</script>`.

Tale contenuto alternativo sarà visualizzato sia se il browser non supporta Javascript sia se è stato configurato in modo da ignorarne il codice.

Il contenuto alternativo può comprendere qualsiasi parte di codice HTML, ed in tal modo potrebbero essere sostituite le funzionalità offerte da Javascript.

¹ In quanto oggetto di livello massimo del DOM `window` è implicitamente assunto. Scrivere `alert()` è equivalente allo scrivere `window.alert()`.

² Nell'esempio sarebbe scritto solo `var hello`, senza pertanto attribuire alla variabile *hello* alcun valore

Esercizio

```
<html>

<head>
<title>Alert Javascript</title>
</head>

<body>
<h1 align="center">Un esempio originale</h1>
<script language= "JavaScript" >
<!--

// Commento: una finestra di dialogo (commento su una sola riga)

var hello = "Hello world!";
alert (hello);

//-->

</script>

<noscript>
Il browser non supporta o non è abilitato per JavaScript
</noscript>

</body>

</html>
```

Disabilitando JavaScript nel browser si ottiene a video il contenuto alternativo previsto dal tag `<noscript>`, come mostrato di seguito.



Per consentire all'utente di confermare o meno un'azione il metodo `alert()` non è sufficiente, in quanto prevede l'unica opzione *OK*. A tal fine occorre usare il metodo *Confirm()* che consente di fornire due possibili risposte, *OK* e *Annulla*.

Si noti che le finestre di dialogo associate ai metodi `alert()`, `prompt()`, e `confirm()` sono di tipo **modale**, vale a dire che bloccano l'esecuzione dello script fino a quando l'utente non fornisce una risposta.

Esercizio

```
<html>
<title>Prompt Javascript</title>
<body>
<h1 align="center">Prompt, write, if, else</h1>
<script language= "JavaScript" >
<!--

var nome = prompt ("Digitare il proprio nome.", "");
if (nome) {
    document.write ("Salve, " + nome);
}
else {
    document.write ('<font color="red">');
    document.write ("Salve, anonimo!");
}
//-->

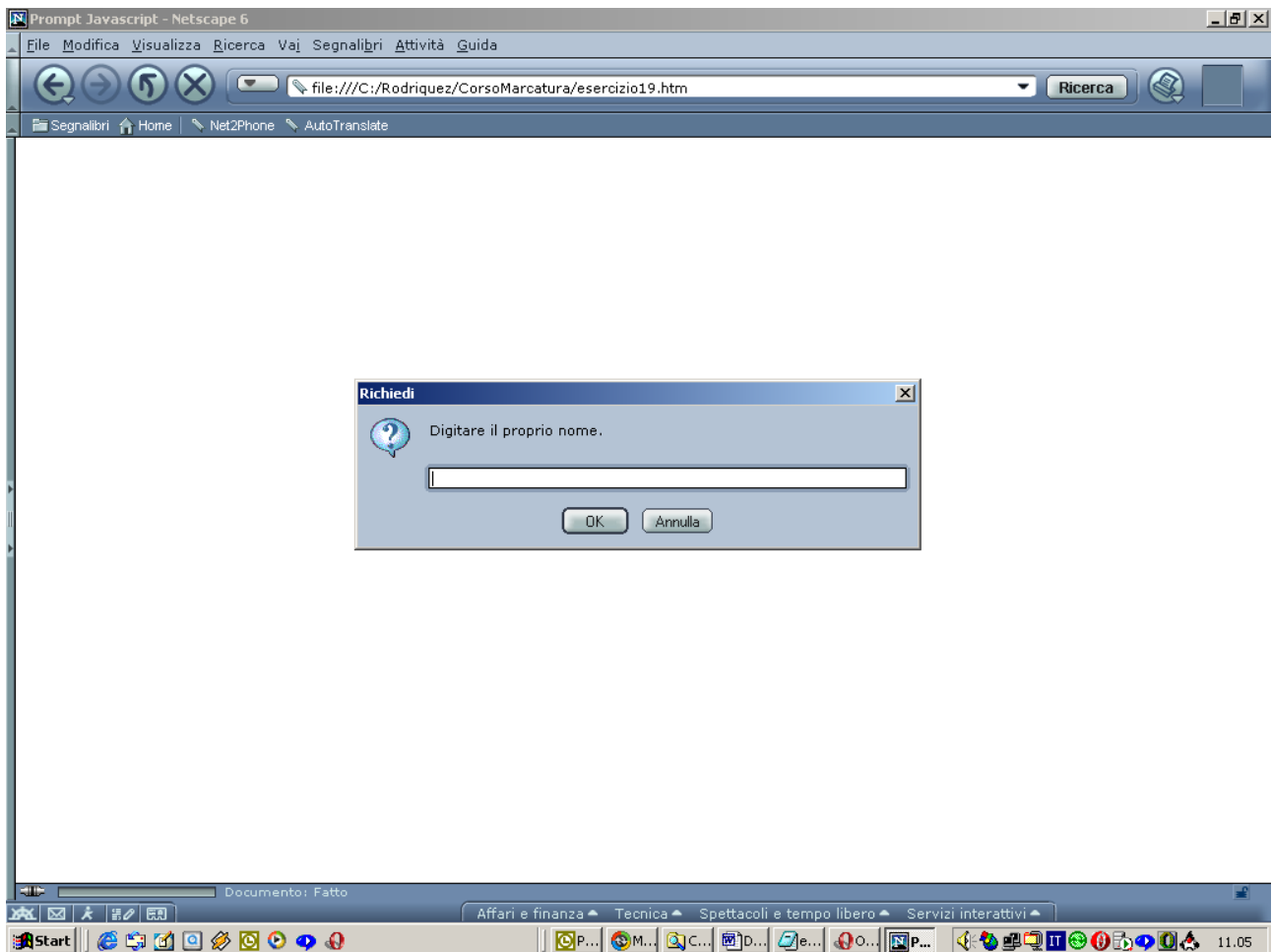
</script>
</body>
</html>
```

Nell'istruzione *document.write*, *document* è un oggetto predefinito e si riferisce al documento HTML in esecuzione. Gli oggetti hanno metodi ad essi associati.³

Nell'esempio *write* è un metodo dell'oggetto *document*. I metodi possono accedere ad argomenti e eseguire azioni. Nell'esempio l'argomento del metodo è costituito dalla stringa "hello world!" e il metodo *write* inserisce la stringa nel documento HTML inviato al client quando questi lo richiede.

Il codice produce il seguente risultato:

³ Un metodo è paragonabile ad una funzione che esegue determinate azioni sia sull'oggetto sia su sue proprietà



Si consideri che quando sono possibili più di due alternative è possibile usare l'istruzione *else if*.

Nell'esempio alla variabile *var* è associata una stringa. L'operatore "+" unisce le due stringhe ed il risultato è nuovamente assegnato alla variabile *var*. Con il metodo *write* dell'oggetto *document* (il documento corrente in esecuzione) il risultato finale è poi scritto a video.

Esempio validazione di una maschera

```
<html>
<head>
<title>Esempio validazione form</title>

<script language= "JavaScript" >
<!--

function verificaInformazioni() {
    var nome=document.dati.nome.value;
    var cognome=document.dati.cognome.value;
    var citta=document.dati.citta.value;
    var messaggio="Attenzione! Valorizzare i seguenti campi:\n";
    var errori = "";

    if (!nome)      {

        errori += "nome\n";
    }

    if (!cognome)   {
        errori += "cognome\n";
    }

    if (!citta)     {
        errori += "citta\n";
    }

    if (errori)     {
        alert (messaggio + errori)
    }
}

//-->

</script>

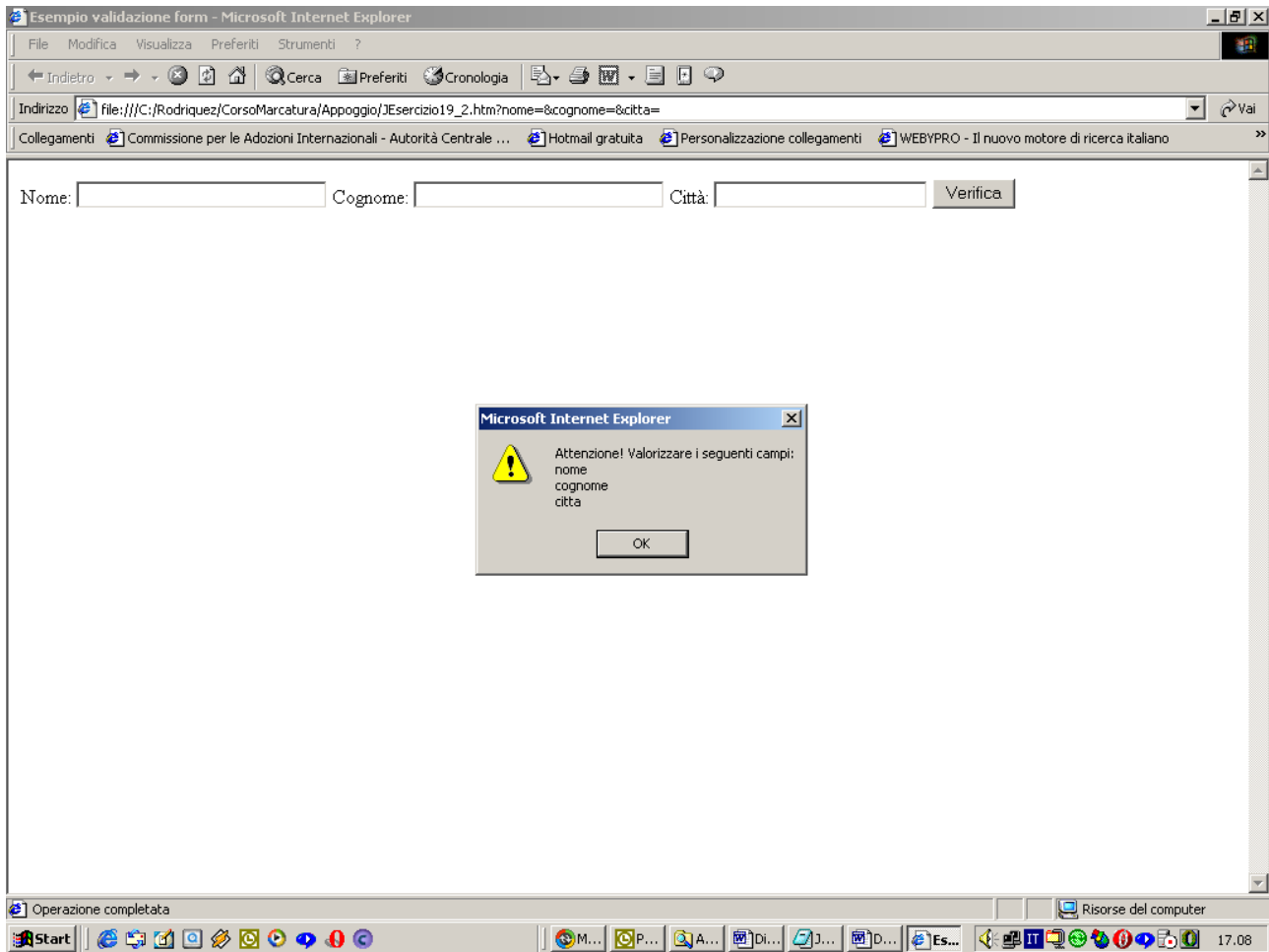
</head>

<body>

<form name= "dati" action="" method="">
Nome: <input type="text" size="30" name ="nome">
Cognome: <input type="text" size="30" name ="cognome">
Città: <input type="text" size="25" name ="citta">
<input type="submit" value= "Verifica" onClick="verificaInformazioni()" >
</form>

</body>
</html>
```

Il codice produce il seguente risultato, se nessun campo è valorizzato:



Nel codice esemplificativo si può notare che:

- è stata creata una funzione (insieme di istruzioni) Javascript tramite l'istruzione *function*
- la funzione è stata invocata tramite il gestore di eventi *onClick*
- è stata utilizzata la funzione *if* per verificare la presenza di valori nelle variabili (tramite l'operatore "!=" la funzione è eseguita se il valore è assente)
- è stata utilizzata la sequenza di escape "\n" per aggiungere un'interruzione di linea all'interno della stringa di testo del messaggio di errore
- è stato usato l'operatore "+" che, nell'esempio, concatena due stringhe di caratteri
- è stato usato l'operatore "+=" che, nell'esempio, concatena alla stringa *errori* il valore di altre variabili