

# Linguaggi

*Corso di Laurea in Ingegneria delle Telecomunicazioni  
A.A. 2010-2011*

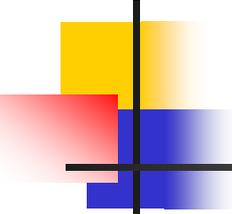
Alessandro Longheu

*<http://www.diit.unict.it/users/alongheu>  
[alessandro.longheu@diit.unict.it](mailto:alessandro.longheu@diit.unict.it)*

*- lezione 15 -*

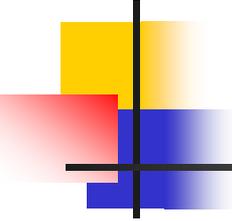
## **Javascript**

# Schema delle lezioni



---

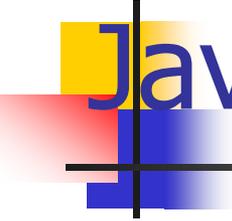
- Introduzione a Javascript: esempi, scopo
- Script interno ed esterno
- Esempi
  - scrittura di HTML da codice
  - Finestre modali, Finestre personalizzabili
  - Variabili ed oggetti
  - Funzioni
  - Array
  - Istruzioni
- Risorse



# Javascript

---

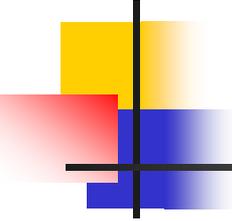
- Javascript è un linguaggio
  - di scripting
  - lato client
- ideato da Netscape (non da SUN), non ha niente in comune con Java se non il nome
- In pratica, è un linguaggio interpretato, object-based piuttosto che object-oriented, nel senso che non permette una completa definizione di nuovi tipi di dato (classi, ereditarietà ecc.), piuttosto opera sull'insieme limitato di oggetti predefiniti che rappresentano le diverse componenti della pagina



# Javascript

---

- Per collocare correttamente Javascript nel contesto dei linguaggi di scripting, è bene dire che ne esistono altri, nati da Microsoft in risposta alla nascita di Javascript:
  - **VBScript** all'interno di Internet Explorer 3
  - **Jscript**, una versione proprietaria di JavaScript
- A causa di alcune differenze presenti in Internet Explorer 3 Netscape e Sun decisero di standardizzare JavaScript e si affidano all'European Computer Manufacturers Association (**ECMA**), così nacque **ECMAScript**
- In definitiva, quando si dice JavaScript, JScript ed ECMAScript sostanzialmente si indicano tre varietà dello stesso linguaggio.



# Oggetti in Javascript

---

- Javascript permette la definizione di oggetti, ma nella stragrande maggioranza di casi, sfrutta oggetti esistenti, in particolare del DOM
- DOM = Document Object Model (modello ad oggetti della pagina WEB)
- Il DOM definisce gli oggetti di una pagina web
- Ogni oggetto ha delle proprietà che lo descrivono
- Ogni oggetto ha dei metodi che permettono di inviare dei messaggi allo stesso per comandargli di fare qualcosa

# DOM – Oggetti predefiniti

---

- Window (parent, top, self, opener)
  - frames[i]
    - document
      - images[i]
      - links[i]
      - forms[i]
        - elements[i]
        - button
        - checkbox
        - image
        - radio
        - reset
        - select
          - options[i]
        - submit
        - text/textarea
    - location

# Generalità sugli eventi

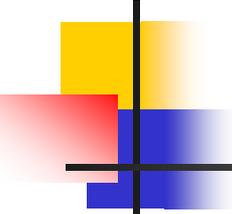
- Oltre essere organizzato a oggetti e metodi, JavaScript sfrutta moltissimo la presenza degli **eventi**.
- Un evento è qualcosa che accade nel documento. L'evento è utile perché permette di rendere interattiva la pagina (ad esempio, reagire con del codice quando l'utente clicca qualcosa)
- Esiste un elenco completo di eventi intercettabili dall'HTML e gestiti spesso con Javascript

| Evento       | Gestore       | Descrizione   |
|--------------|---------------|---|
| Load         | onLoad        | Caricamento della pagina                                    |
| Unload       | onUnLoad      | Uscita dalla pagina   |
| Click        | onClick       | Click col tasto sinistro del mouse su di un elemento        |
| Double click | onDbClick     | Doppio click col tasto sinistro del mouse su di un elemento |
| Mouse over   | onMouseOver   | Passaggio del mouse su di un elemento                       |
| Mouse move   | onMouseMove   | Movimento del mouse su di un elemento                       |
| Mouse out    | onMouseOut    | Uscita del mouse dall'area di un elemento                   |
| Mouse down   | onMouseDown   | Click del mouse su di un elemento                           |
| Context menu | onContextMenu | Click col tasto destro del mouse su di un elemento          |
| Change       | onChange      | Cambio di elemento  |
| Blur         | onBlur        | Perdita del focus (focalizzazione) da parte di un elemento  |
| Focus        | onFocus       | Raggiungimento del focus da parte di un elemento            |
| Submit       | onSubmit      | Pressione del tasto "Invio" della tastiera                  |

[http://space.cinet.it/cinetclub/ingar/elenco\\_eventi\\_javascript.htm](http://space.cinet.it/cinetclub/ingar/elenco_eventi_javascript.htm)

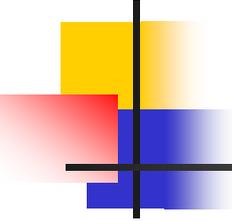
contiene semplici esempi di applicazione di tutti gli eventi sopra citati

# Javascript



---

- Il comportamento dinamico degli oggetti nella pagina HTML che incorpora Javascript non e' altro che una lista di istruzioni del tipo **proprietà=valore** (la proprietà spesso è un **attributo** di un qualche **oggetto** della pagina) e/o messaggi del tipo **oggetto.metodo()** e puo' essere predefinito da chi ha scritto la pagina in modo che avvenga quando la pagina e' caricata, oppure iniziato da qualcosa che l'utente compie.
- In questo caso si dice che si e' avuto un certo **evento** e chi scrive la pagina deve indicare cosa deve succedere quando l'evento accade



# Javascript

---

- Un primo programma:

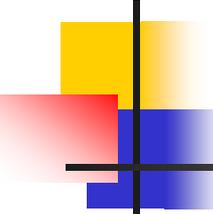
```
<title> Benvenuti in Javascript </title>
```

```
<SCRIPT language=JavaScript>
```

```
document.write(Benvenuti in Javascript );
```

```
</SCRIPT>
```

- document è un oggetto, write un metodo dell'oggetto



# Javascript

---

- Un altro esempio con form e oggetti di tipo text per leggere i dati e per scrivere il risultato:

```
<title>Somma di due numeri</title>
```

```
<form name="somma">
```

```
<input name="operando1" > +
```

```
<input name="operando2" > =
```

```
<input name="risultato" value="0"> <p>
```

```
<input type="button" value="Fai la somma!" onClick=  
"somma.risultato.value=Number(somma.operando1.value)  
+ Number(somma.operando2.value)"> </form>
```

- **onclick** è un **evento** intercettato dall'HTML, in corrispondenza del quale viene eseguito il codice Javascript

# Javascript

- **Esempio di funzione (metodo):**

```
<head>
```

```
<title>Calcola la media di una serie di numeri</title>
```

```
<script language=javascript>
```

```
    function calcolamedia(lista){
```

```
        dati = lista.split(" ")
```

```
        if (dati.length > 0) {
```

```
            var somma = 0;
```

```
            for (var i = 0; i < dati.length; i++){
```

```
                somma = somma+Number(dati[i]) }
```

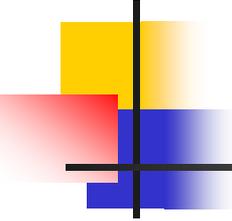
```
            return (somma / dati.length);
```

```
        }
```

```
    }
```

```
</script>
```

```
</head>
```



# Javascript

---

```
...  
<body>  
  <form name="media">  
    Lista di valori <input name="valori"> <br>  
    Risultato <input name="risultato"> <p>  
    <input type="button" value="Calcola la media!"  
      onClick = "media.risultato.value =  
        calcolamedia(media.valori.value)">  
  </form>  
</body>
```

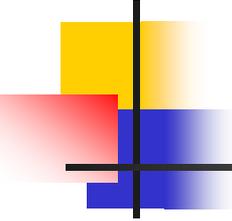
# Javascript

- **Perché viene usato Javascript?** Con questo semplice linguaggio di scripting che viene "letto" dal browser, si possono eseguire un'infinità di operazioni che il semplice HTML non permette. E' possibile:
  - "aggiustare" l'HTML in modo da correggere le imperfezioni che si vedono in un browser, ma che in un altro non ci sono
  - scambiare delle immagini (effetto di roll-over)
  - aprire nuove finestre secondarie
  - effettuare controlli sui moduli, per verificare l'input dell'utente
  - eseguire azioni "a tempo" ("dopo 5 secondi vai a quella pagina")
  - aggiungere effetti "dinamici" (oggetti che si muovono in giro per la pagina, ma anche menu a discesa, a comparsa", ecc)
  - modificare il contenuto di un frameset

# Javascript

- Si può inserire il codice JavaScript in qualsiasi parte del documento (nella head oppure nel body) a seconda delle nostre esigenze:
  - Solitamente, **è buona norma inserire il tag SCRIPT nella HEAD** per separare il codice dello scripting dal codice HTML della pagina vera e propria, in modo da mantenere la pagina più leggibile
  - Si inserisce il tag SCRIPT nel BODY della pagina ogni volta che la sintassi JavaScript sia necessaria per visualizzare il layout della pagina stessa. Può capitare infatti che il codice HTML venga scritto attraverso JavaScript: in questo caso è bene che la sintassi che completa la visualizzazione di un punto della pagina venga scritta in quel determinato punto
- Per **scrivere la sintassi** è sufficiente aprire il tag <SCRIPT>. Il codice JavaScript va inserito tra l'apertura e la chiusura del tag, esempio:

```
<script type="text/javascript">  
    alert("ciao");  
</script>
```
- si può anche scrivere <script language="javascript"> , ma **l'uso di language è deprecato**, per cui è opportuno abituarsi ad utilizzare type <sup>14</sup>

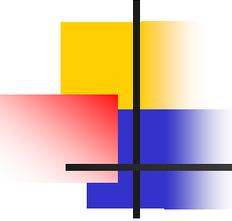


# Javascript

---

- Per prevenire l'eventualità che Javascript sia disattivato dall'utente nel proprio browser e la pagina diventi inutilizzabile, è il caso di predisporla in modo che sia visualizzata in ogni caso correttamente. Si usa a tale scopo il tag **noscript**, racchiudendo all'interno il contenuto alternativo da visualizzare, ad esempio:

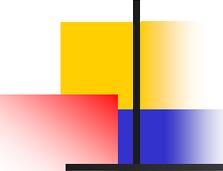
```
<noscript>  
<div align="center">  
<h3><font face="Verdana,sans-serif">  
  Per visualizzare correttamente <br>  
  il contenuto della pagina<br>  
  occorre avere JavaScript abilitato.  
</font></h3></div>  
</noscript>
```



# Javascript

---

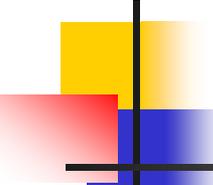
- Nel caso in cui lo stesso JavaScript venga ripetuto in più pagine, può essere utile includere il codice in un **file esterno**.
- Si tratta di prender tutto il codice Javascript elaborato e inserirlo in un file di testo vuoto. È sufficiente mettere il codice così com'è senza nessun preambolo e salvarlo con estensione .js
- Una volta creato il file "js", per includerlo nelle pagine la sintassi è:
  - `<script src="miojavascript.js" type="text/javascript"></script>`
- L'attributo src deve contenere l'indicazione del percorso (assoluto o relativo) del file, come per le immagini.



# Javascript

- Uno dei metodi più importanti di JavaScript è il metodo **write** che si riferisce all'oggetto **document** (la pagina). Questo metodo ci consente di scrivere all'interno di una pagina HTML usando JavaScript
- Con questo metodo possiamo scrivere anche del codice HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0  
Transitional//EN">  
<html><head></head>  
<body>  
<script type="text/javascript">  
document.write("<h1><font face=|\"Verdana,Arial,Helvetica,sans-  
serif|\">prova di script</font></h1>");  
</script>  
</body>  
</html>
```

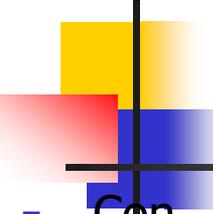


# Javascript

- JavaScript è in grado di generare **tre differenti tipologie** di "finestre implicite" (dette anche "finestre modali" o "finestre di dialogo"):
- **alert** visualizza un avvertimento, un messaggio di errore che blocca qualsiasi azione dell'utente finché egli non dà il suo benestare cliccando sul bottone. L'aspetto grafico della finestra è gestito dal browser e dal sistema operativo. La sintassi è *alert("messaggio");*
- **Confirm** è una finestra che pone una domanda e chiede conferma dando la possibilità di scegliere tra due opzioni ("ok" e "annulla"). La sintassi è: *confirm("domanda");*, ad esempio  

```
<a href="http://www.altrosito.com" onClick="confirm('Sei sicuro di voler uscire dalla pagina?');" TARGET="_blank">Altro sito</a>
```
- **Prompt** è una finestra che pone una domanda e consente all'utente di dare la risposta che vuole. Può avere anche un valore di risposta predefinito. La sintassi è: *prompt("domanda","risposta predefinita");*, esempio:  

```
<body onLoad="prompt('Come ti chiami?');">
<body onLoad="prompt('Come ti chiami?','inserisci il tuo nome');">
```



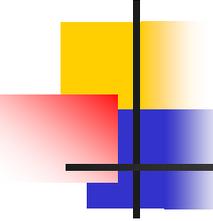
# Javascript

- Con JavaScript è possibile creare delle finestre personalizzate. Sintassi: `window.open('percorso','nome finestra','caratteristiche separate da virgola');`
- **percorso** di un file html da inserire all'interno della finestra
- **nome finestra** indica il nome della finestra che abbiamo creato
- **caratteristiche separate da virgola** specifica la forma, la dimensione e la posizione della finestra, possiamo anche far sì che la finestra non sia ridimensionabile, o che non ci siano barre degli strumenti e di scorrimento.
- Esempio:

```

        <html><head><title>esempio</title></head>
        <body bgcolor="#ffffff" text="#000000">
<a      href="#"      onClick="window.open('http://sito1.it','finestra','');">finestra
                1</a><br/>
<a      href="#"      onClick="window.open('http://sito2.it','finestra','');">finestra  2
                </a><br/>
<a href="http://sito3" target="finestra"> clicca qui per cambiare un'altra finestra
                da sito1 a sito3</a><br/>
</body></html>

```

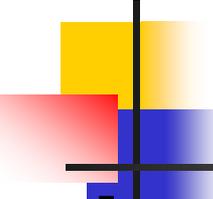


# Javascript

- Le proprietà delle finestre personalizzabili sono diverse:

| caratteristica | valore   | spiegazione   | esempio        |
|----------------|----------|---|----------------|
| width          | numerico | la larghezza della finestra in pixel  | width=400      |
| height         | numerico | l'altezza della finestra in pixel   | height=200     |
| left           | numerico | la distanza dalla sinistra del monitor                                      | left=300       |
| top            | numerico | la distanza dal lato superiore del monitor                                  | top=350        |
| resizable      | yes / no | indica se la finestra può essere ridimensionata o no                        | resizable=no   |
| fullscreen     | yes / no | indica se la finestra va aperta a tutto schermo                             | fullscreen=no  |
| channelmode    | yes / no | indica se la finestra deve essere aperta "in modalità canale" (solo per ie) | channelmode=no |
| menubar        | yes / no | la barra del menu (quella con scritto "file", "modifica", ecc.)             | menubar=no     |
| toolbar        | yes / no | la barra degli strumenti del browser (con i pulsanti "indietro", "avanti")  | toolbar=no     |
| location       | yes / no | la barra degli indirizzi del browser  | location=no    |
| scrollbars     | yes / no | le barre di scorrimento laterali  | scrollbars=no  |
| status         | yes / no | la barra di stato (quella in basso)   | status=no      |

■ `window.open('prova.html', "", 'width=220,height=220,left=0,top=0,resizable=no,menubar=yes,toolbar=yes,scrollbars=no,locations=no,status=no');`



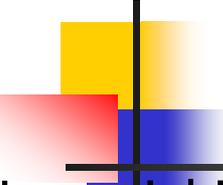
# Javascript

---

- Javascript permette l'uso di **variabili** con la sintassi del C, senza bisogno di dichiarazione preventiva. Esempio:

```
                                <html>
<head> <title>Esempio</title> </head>
<body>
                                <SCRIPT type="text/javascript">
    nomeUtente=prompt("Scrivi il tuo nome","il tuo nome");
    document.write("Benvenuto ");
    document.write(nomeUtente);
</SCRIPT> </body> </html>
```

- Le variabili hanno validità all'interno della singola pagina, dopodiché vengono distrutte, (è stata una scelta degli ideatori di JavaScript, dovuta a motivi di sicurezza). Per memorizzare i dati dell'utente si devono quindi utilizzare altre strategie (ad esempio cookies).



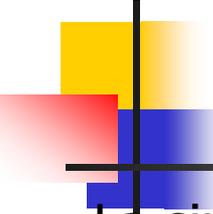
# Javascript

Le variabili possono anche contenere **oggetti**:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
    Transitional//EN">
<html><head><title>Esempio</title></head><body>
    <SCRIPT type="text/javascript">
/*variabili contenenti larghezza e altezza desiderata della finestra*/
        larghFinestra=300;
        altezFinestra=300;
/*variabili per ricavare la posizione della finestra rispetto allo schermo*/
        sinistra=screen.width-larghFinestra+20;
        alto=(screen.height-larghFinestra)/2;
        /*creazione di un oggetto*/
miaFinestra=window.open("sponsor.htm", "", "left="+sinistra+",
top="+alto+", width="+larghFinestra+", height="+altezFinestra+", menubar,
        toolbar");
    </script>
    <a href="javaScript:miaFinestra.close()">chiudi</a>
</body></html>

```



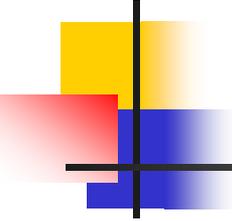
# Javascript

- La sintassi per creare una **funzione** è:

```
function nomeFunzione() { // codice }
```

- La funzione creata si richiama poi con *nomeFunzione()*; da inserire nella head, nel body, oppure da richiamare tramite un evento. Esempio:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html><head><title>Esempio</title>
<SCRIPT type="text/javascript">
function saluta() { alert("ciao"); }
function saluta1() { alert("Benvenuto!!!"); }
function scriviRes(){
  document.write("Risoluz: "+screen.width+" x "+screen.height); }
</script></head>
<body onLoad="saluta1();">
<a href="#" onClick="saluta();">clicca per ricevere un saluto</a>
<br/>
<script type="text/javascript">
  scriviRisoluzione();
</script>
</body></html>
```



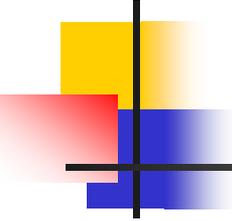
# Javascript

---

- Nell'esempio precedente sono presenti tre funzioni, due delle quali vengono richiamate attraverso l'utilizzo di eventi, mentre la terza viene richiamata all'interno della pagina.
- Da notare inoltre che l'evento "onLoad" viene eseguito subito dopo il completamento della pagina, dunque una funzione richiamata all'interno del BODY della pagina verrà "lanciata" prima della funzione richiamata dall'evento onLoad. Nell'esempio la funzione "scriviRes()" viene lanciata prima di "saluta1()".
- E' possibile anche specificare che la funzione deve essere richiamata da un evento direttamente nel codice JavaScript anziché tramite HTML:
- Il codice è questo (da inserire nell' <head>):

```
<script type="text/javascript">  
    function saluta1() {  
        alert("Benvenuto!!!");  
    }  
    document.onload=saluta;
```

```
</script>
```



# Javascript

---

- La parola chiave var permette la definizione di variabili locali:

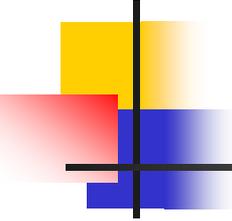
```

<script type="text/javascript">
  scritta2="ciao globale"; // variabile globale
  function saluta() {
    var scritta2="ciao locale"; // variabile locale
  }
  saluta();
  alert(scritta2);

```

```
</script>
```

- Nell'ultimo alert, tra le due variabili "vince" la variabile globale, perché la variabile locale ha il suo ciclo di vita solo all'interno della funzione.
- Se occorre restituire al codice HTML una variabile locale, occorre utilizzare il comando return alla fine della funzione. Spesso, si omette var (quindi le variabili sono globali) e si può quindi fare a meno anche di return



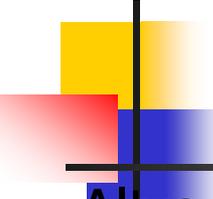
# Javascript

- Esempio di if con **funzione**:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
  <html><head>
    <script type="text/javascript">
      function valida() {
        //inserimento dei valori dei campi in variabili
        eta=dati.eta.value;
        email=dati.email.value;
        if (isNaN(eta)){
          alert ("Inserisci un'età valida");
          return false;
        }
        if (email.indexOf("@")==-1) {
          alert ("Inserisci un'email valida");
          return false;
        }
      }</script></head><body>
    <form name="dati" action="http://www.html.it" onSubmit="return valida();">
      <!--notare il return anteposto alla chiamata della funzione-->
      La tua età: <input type="text" name="eta"><br/>
      La tua e-mail: <input type="text" name="email"><br/>
      <input type="submit" value="invia">
    </form></body></html>

```



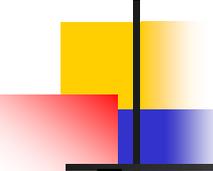
# Javascript

- Altro esempio di if con **funzione**:

```

        <html><head>
        <script type="text/javascript">
            function conferma () {
chiediConferma = confirm('Sei sicuro di voler uscire dalla
                        pagina?');
            if (chiediConferma == true){
location.href="#"; //ricarica la pagina
            }
            /* forma sintetica:
            if (confirm('Sei sicuro di voler uscire dalla pagina?')){
location.href="#"; //ricarica la pagina
            } */
            }
        </script></head><body>
        <A href="#" onClick="conferma();">testo link</a>
</body></html>

```



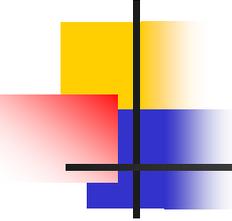
# Javascript

- Esempio di **array** e sua rappresentazione tabellare in HTML tramite Javascript:

```

<script type="text/javascript">
alunni = new Array("Aldo", "Giovanni", "Gianni", "Mario", "Gianni", "Monica");
voti= new Array(3,8,5,7,4,4);
</script>
<table border="1">
<tr>
<th><B>Voti</B></th>
<th><B>Alunni</B></th>
</tr>
<script type="text/javascript">
i=0;
while(i<alunni.length) {
document.write("<tr>");
document.write("<td>" +alunni[i]+ "</td>");
document.write("<td>" +voti[i]+ "</td>");
document.write("</tr>");
i++;
} </script> </table>

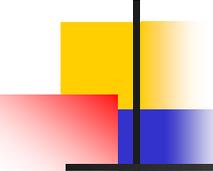
```



# JavaScript

■ Gli  
di diversi metodi utili:

| Metodo o proprietà              | Descrizione  | Esempio   |
|---------------------------------|--|---|
| length                          | Conoscere la lunghezza di un array   | <pre>alunni=new Array("Mario", "Gianni","Monica"); alert(alunni.length);</pre>  |
| push(elemento)                  | Aggiungere un elemento in coda all'array e restituire la nuova lunghezza   | <pre>alunni=new Array("Mario", "Gianni","Monica"); aggiungi=alunni.push("Davide"); alert(aggiungi);</pre>                                 |
| concat (elementi da aggiungere) | Aggiungere elementi ad un array e restituire la nuova lunghezza. Restituisce un nuovo array formato dalla somma degli elementi | <pre>alunni=new Array("Mario", "Gianni","Monica"); aggiungi=alunni.concat("Davide","Giovanni"); alert(aggiungi);</pre>                    |
| pop()                           | Eliminare un elemento dalla fine dell'array e restituire il nome dell'elemento eliminato.                                      | <pre>alunni=new Array("Mario", "Gianni","Monica"); togli=alunni.pop(); alert(togli); alert(alunni.length);</pre>                          |
| shift()                         | Eliminare un elemento dall'inizio dell'array e restituire il nome dell'elemento eliminato                                      | <pre>alunni=new Array("Mario", "Gianni","Monica"); togli=alunni.shift(); alert(togli); alert(alunni.length);</pre>                        |
| reverse()                       | Invertire l'ordine degli elementi di un array  | <pre>alunni=new Array("Mario", "Gianni","Monica"); alunni.reverse(); alert(alunni[0]);</pre>  |
| slice(inizio,fine)              | Dividere l'array in un array più piccolo e restituire il nuovo array   | <pre>alunni=new Array("Mario", "Gianni","Monica","Davide"); alert(alunni.length); alunni2=alunni.slice(0,2); alert(alunni2.length);</pre> |

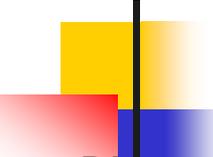


# Javascript

- Istruzioni disponibili: if, while, do-while, for, switch... analoghe a quelle disponibili nel linguaggio C
- Esiste anche il for-in, esempio:
 

```
<form name="mioForm">
  <input type="button" NAME="miotesto" value="mio campo">
</form>
<script type="text/javascript">
for (prop in document.mioForm) {
  document.write(prop+"<br/>");
}
</script>
```
- altro esempio:
 

```
<script type="text/javascript">
alunni = new
Array("Aldo", "Giovanni", "Gianni", "Mario", "Gianni", "Monica");
for (prop in alunni) {
document.write(alunni[prop]+"<br/>");}
</script>
```



# Javascript

- Risorse con esempi commentati:
  - <http://www.wowarea.com/italiano/aiuto/javascit.htm>
  - [http://www.agatello.com/v2/java\\_sc.htm](http://www.agatello.com/v2/java_sc.htm)
  - <http://www.mrwebmaster.it/tutorial/dhtml/>
  - <http://www.mrwebmaster.it/tutorial/js/>
  - <http://www.mrwebmaster.it/script/js/>
  - <http://javascript.internet.com/>
  - <http://www.webmasterpoint.org/appunti/javascript/home.asp>
- Risorse per documentazione e manualistica
  - <http://www.dmoz.org/World/Italiano/Computer/Programmazione/JavaScript/>
  - <http://www.excite.co.uk/directory/World/Italiano/Computer/Programmazione/Javascript>
    - molti link verso siti relativi a Javascript
  - <http://www.morpheusweb.it/html/manuali/javascript.asp>
    - manuale con DOM
  - <http://www.w3.org/TR/REC-DOM-Level-1/level-one-html.html>
    - guida completa al DOM (autoritativa)
  - <http://it.wikipedia.org/wiki/JavaScript>
    - elenco completo di funzioni javascript e oggetti DOM
  - <http://research.nihonsoft.org/javascript/CoreReferenceJS15/index.html>
    - manuale scaricabile
  - <http://www.ecma-international.org/publications/standards/Ecma-262.htm>
    - manuale di riferimento ECMAScript