



Dynamic (x)HTML

DHTML

Pagine dinamiche

- xhtml e css vengono “letti” dal browser
 - È possibile aggiungere script
 - vengono ***eseguiti*** dal browser
 - attenzione alla sicurezza!!
 - possono gestire l'***interazione*** (complessa) con l'utente
 - possono ***modificare*** la pagina
 - attenzione alla validazione!!
-

Come aggiungere script

- Gli script si aggiungono in un documento xhtml tra i tag
 - `<script type="text/javascript">`
 - `</script>`
- All'interno dei tag `head`
- All'interno dei tag `body`
- In un file esterno con estensione `.js` e poi nel file xhtml con

```
<script type="text/javascript"
  src="nomefile.js"></script>
```

Esempio 1

Questo è un commento html e serve a nascondere il codice javascript ai vecchi browser che non lo capirebbero

```
<h1>Primo esempio</h1>
<p>Testo scritto nel sorgente.</p>
<script type="text/javascript">
<!--
    document.writeln( '<p>Ciao</p>' );
//-->
</script>
```

Esempio 2

```
<h1>Secondo esempio</h1>
<script type="text/javascript">
<!--
    document.write('<p>Modificato il ');
    document.writeln(document.lastModified
        + '</p>');
//-->
</script>
```

Esempio 3

```
<h1>Terzo esempio</h1>
<script type="text/javascript">
<!--
    window.alert('Allegria!!');
//-->
</script>
```

Eventi

- È possibile associare l'esecuzione di uno script in seguito ad un evento
 - Gli eventi possibili in xhtml sono:
 - `onclick`, `ondblclick`, `onmousedown`, `onmouseup`, `onmouseover`, `onmousemove`, `onmouseout`, `onkeypress`, `onkeydown`, `onkeyup`
 - `onblur`, `onfocus`, `onload`, `onunload`, `onreset`, `onsubmit`, `onchange`, `onselect`
-

Eventi

- `onload` e `onunload` sono associati a `body`
 - `<body onunload="window.alert('ciao ciao');">`
 - `onchange` e `onselect` sono associati a `input` e `textarea`, `onchange` anche a `select`
 - `onblur` e `onfocus` sono associati ad `a`, `label`, `input`, `select`, `textarea` e `button`
 - `<button onfocus="window.alert('focus');">`
 prova
 `</button>`
 - `onreset` e `onsubmit` sono associati a `form`
 - Gli altri eventi sono associati a (quasi) tutti i tag xhtml
-

Cambiare lo stile

- Possiamo cambiare lo stile dei tag in seguito a determinati eventi

- Es:

```
<p onclick="this.style.color='red';">
  prova a cliccare
</p>
<h2
  ondblclick="this.style.backgroundColor='yellow';">
  Prova a fare doppio click
</h2>
```

Le funzioni

- Un altro modo per eseguire delle istruzioni in seguito ad un evento è scrivere una funzione

- Es

```
<script type="text/javascript">
<!--
  function cambia(elemento) {
    elemento.style.color='red';
    elemento.style.fontSize='200%';
  }
  //-->
</script>
<p onclick="cambia(this);">
  Prova a cliccare e vedrai!!</p>
```

Le variabili

```
<head>...
<script type="text/javascript">
<!--
  var dimensione = 10;
  function aumenta(elemento) {
    elemento.style.fontSize=dimensione+'pt';
    dimensione = dimensione + 2;
  }
  //-->
</script>
</head><body>...
<p onclick="aumenta(this);">Ad ogni click il testo
  aumenta!</p>
```

Oggetto window

- Il metodo `window.alert('messaggio')` mostra una finestra di dialogo con un bottone OK. Viene utilizzato per dare messaggi all'utente.
 - Il metodo `window.confirm('messaggio')` mostra una finestra con un bottone OK ed uno CANCEL. Ritorna il valore `true` se l'utente schiaccia OK, `false` altrimenti. Serve per dare messaggi e sapere le scelte dell'utente.
 - Es: `document.writeln(window.confirm('Clicca'));`
 - Il metodo `window.prompt('messaggio','default')` mostra una finestra con una casella di testo ed un bottone OK. Ritorna il testo inserito nella casella. Può essere usato per far inserire dati all'utente.
 - Es: `document.writeln(window.prompt('Scrivi','a'));`
-

Aggiungere tag

```
function aggiungiTag() {  
  var p1 = document.createElement('p');  
  var t1 = document.createTextNode('Sopra la  
  panca la capra campa.');
```

`p1.appendChild(t1);`

```
  var b1 =  
  document.getElementsByTagName('body').item(0);  
  b1.appendChild(p1);  
}
```

Oggetto document

- `document.createElement('tag')` serve per creare un elemento nell'albero del documento
 - `document.createTextNode('testo')` serve per creare del testo
 - `appendChild` serve ad appendere il nodo nell'albero del documento
-

Accedere agli elementi

```
<script type="text/javascript">
<!--
  function col_elem() {
    var intest2 = document.getElementsByTagName('h2');
    intest2.item(0).style.color = 'red';
    intest2.item(1).style.color = 'green';
    intest2.item(2).style.color = 'blue';
  }
  //-->
</script></head><body>
<p onclick="col_elem();">Clicca per colorare le
  intestazioni di secondo livello</p>
<h2>Intestazione uno</h2><h2>Intestazione
  due</h2><h2>Intestazione tre</h2>
<h2>Intestazione quattro</h2>
```

Accedere agli elementi 2

```
<p onclick =
  "document.getElementById('pippo').style.color =
  'blue';">
Clicca per colorare il tag con
id="&quot;pippo&quot;";</p>
<h2>Intestazione uno</h2>
<h2>Intestazione due</h2>
<h2 id="pippo">Intestazione tre</h2>
```

Cancellare elementi

```
<script type="text/javascript">
<!--
function cancellElem(){
    var p1 = document.getElementById('ex1');
    var b1 = document.getElementById('pluto');
    p1.removeChild(b1);
}
//-->
</script></head><body>
<p id="ex1" onmouseover="cancellElem();">Se passi
sopra con il mouse <span id="pluto">questo
testo scompare</span></p>
```

Modificare elementi

```
<script type="text/javascript">
<!--
function modifElem(){
    var p1 = document.getElementById('ex1');
    var b1 = document.getElementById('pluto');
    var c1 = document.createTextNode('... sei passato!');
    var b2 = document.createElement('span');
    b2.appendChild(c1);
    p1.replaceChild(b2,b1);
}
//-->
</script></head><body>
<p id="ex1" onmouseover="modifElem();">Se passi sopra con
il mouse <span id="pluto">questo testo viene
modificato</span></p>
```

Inserire elementi

```
<script type="text/javascript">
<!--
function inserElem(){
    var p1 = document.getElementById('ex1');
    var b1 = document.getElementById('pluto');
    var c1 = document.createTextNode('questo è stato aggiunto!');
    var b2 = document.createElement('span');
    b2.appendChild(c1);
    p1.insertBefore(b2,b1);
}
//-->
</script></head><body>
<p id="ex1" onmouseover="inserElem();">Se passi sopra con il
  mouse <span id="pluto">viene aggiunto del testo</span></p>
```

Modificare gli attributi dei tag

```
<script type="text/javascript">
<!--
function cambiaLink() {
    var l = document.getElementById('abc');
    l.setAttribute('href', 'http://www.google.com/');
}
//-->
</script></head><body>
<p onclick="cambiaLink();">Clicca per cambiare il
  link del prossimo paragrafo</p>
<p>Questo <a href="http://web.uniud.it/"
  id="abc">link</a> sar&agrave; modificato</p>
```

Operazioni aritmetiche

- Javascript è un linguaggio di programmazione che ci permette di fare calcoli

```
<script type="text/javascript">
document.writeln('la somma di 2 e 3 &egrave; ');
document.writeln(2+3);
document.writeln('<br />');
document.writeln('Se divido 10 in 3 parti ottengo ');
document.writeln(10/3);
</script>
```

Operazioni aritmetiche

- Somma +
- Sottrazione -
- Divisione /
- Moltiplicazione *
- Resto della divisione %
- Attenzione agli errori di arrotondamento!!

```
document.writeln((1936.27*3)/3);
```

Controlli

- Con il comando `if (condizione) {comandi1} else {comandi2}` possiamo far eseguire istruzioni diverse in occasioni diverse. Se la condizione è vera verranno eseguite le istruzioni `comandi1` altrimenti `comandi2`
 - ```
<script type="text/javascript">
a = prompt('indovina un numero','');
if (a == 10) {
 document.writeln('Bravo!');
} else {
 document.writeln('No! Era 10. Tu hai detto '+a);
}
</script>
```
- 

## Condizioni

---

- |                     |    |                                  |
|---------------------|----|----------------------------------|
| ■ Uguale            | == | ■ Valori booleani                |
| ■ Diverso           | != | ■ true                           |
| ■ Maggiore          | >  | ■ false                          |
| ■ Minore            | <  | ■ Operatori booleani             |
| ■ Maggiore o uguale | >= | ■ E &&                           |
| ■ Minore o uguale   | <= | ■ a maggiore di 5 e b uguale a 3 |
|                     |    | ■ (a > 5) && (b == 3)            |
|                     |    | ■ O                              |
|                     |    | ■ Non !                          |
-

## Cicli for

---

- Con il comando

`for (var i=0; i<x; i++) {comandi}`  
facciamo ripetere le istruzioni *comandi* un numero *x* volte

- ```
<script type="text/javascript">
  for (var i=0; i<10; i++) {
    document.writeln(i+' ciao');
    document.writeln('<br />');
  }
</script>
```

Cicli while

- Con il comando

`while (condizione) {comandi}`
facciamo ripetere le istruzioni *comandi* finché la condizione diventa falsa

- ```
<script type="text/javascript">
while(!window.confirm('Andiamo avanti?')) {
 window.alert('Non ho capito bene...');
}
document.writeln('Bravo!');
</script>
```

---

## Far eseguire qualcosa a tempo

- Con `setTimeout('comando',x)` si fa eseguire l'istruzione `comando` dopo `x` millisecondi
- ```
<script type="text/javascript">
setTimeout('window.alert('\u0027uella!\u0027')',3000);
</script>
```

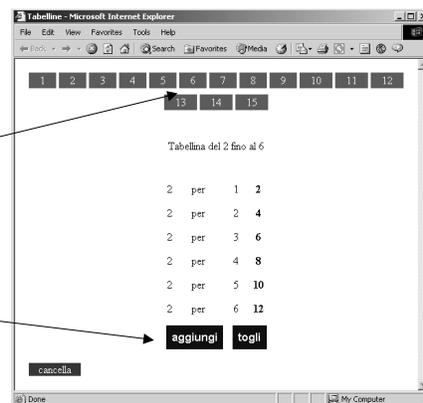
Apice iniziale

Apice in mezzo (con la \ davanti)

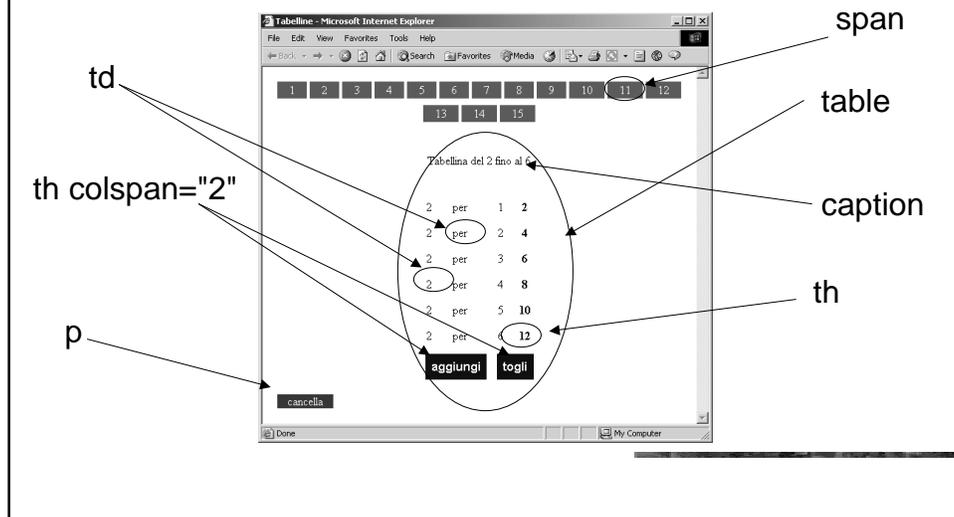
Apice finale

Esempio: le tabelline

- Una pagina xhtml che mostra le tabelline
- Possibilità di scegliere quale tabellina
- Possibilità di scegliere fino a quanto



Struttura xhtml



Cambiare tabellina

- Quando si clicca su uno degli **span** verdi la tabellina cambia
- Associamo all'evento **onclick** di ogni **span** una funzione **tabellina(n)**

```
<span onclick="tabellina(1)">1</span>
```

```
<span onclick="tabellina(2)">2</span>
```

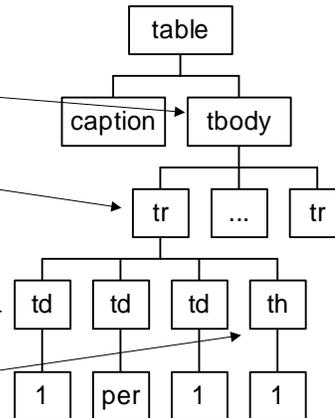
```
<span onclick="tabellina(3)">3</span>
```

...

Funzione tabellina(n)

1. Selezionare la tabella
2. Selezionare il corpo
3. Per ogni riga tranne l'ultima

1. Costruisci un nuovo nodo td con n e sostituisci
2. Costruisci un nuovo th con il risultato e sostituisci



Funzione tabellina(n)

```
function tabellina(n) {  
  var tab = document.getElementById('tabellina');  
  var corpotab = tab.firstChild.nextSibling;  
  var righe = corpotab.childNodes;  
  for (var i=0; i<righe.length-1; i++) {  
    var celle = righe.item(i).childNodes;  
    var num = creaCella('td',n);  
    var res = creaCella('th',n*(i+1));  
    righe.item(i).replaceChild(num,celle.item(0));  
    righe.item(i).replaceChild(res,celle.item(3));  
  }  
}
```

table ha id="tabellina"

collezione di tutti i figli (tr)

collezione di tutti i figli (td e th)

Funzione creaCella(tag,txt)

```
function creaCella(tag,txt) {  
  var c = document.createElement(tag);  
  var txt = document.createTextNode(txt);  
  c.appendChild(txt);  
  return c;  
}
```

