

# Informatica A e B

## Antonio Lieto

### Parte V

### **Tecnologie Web - Linguaggi**

# ARGOMENTI DI QUESTO GRUPPO DI LUCIDI

- **LINGUAGGI DI MARKUP**
- **HTML (HyperText Markup Language)**
- **CSS (Cascading Style Sheets)**
- **Principali tecnologie per pagine Web dinamiche**
  - Tecnologie client-side
    - JavaScript
    - Applet Java
    - HTML 5
  - Tecnologie server-side
    - Scripting server-side
    - Servlet Java
    - Java Server Pages
- **CMS (Content Management Systems)**

# SITI WEB: Principali tecnologie di base

- Iniziamo con il presentare alcune tecnologie base:
  - **Linguaggi di markup** per specificare la struttura e annotare le pagine Web: **HTML** (HyperText Markup Language), **XHTML** (eXtensible HyperText Markup Language)... attualmente, è sempre più utilizzato **HTML5**...
  - **Linguaggi per specificare lo stile di presentazione** delle pagine Web: principalmente **CSS** (Cascading Style Sheets)...attualmente, il riferimento è a **CSS3**

# Linguaggi di markup (I)

- I linguaggi di markup sono linguaggi che consentono di annotare dei documenti (spesso testuali, ma non solo)
- Le annotazioni sono meta-informazioni (cioè informazioni sul contenuto vero e proprio dei documenti) di vario tipo
- Relativamente alle pagine Web, i linguaggi di markup come HTML o XHTML consentono di specificare meta-informazioni sul contenuto delle pagine Web: ad esempio, indicazioni per la visualizzazione del contenuto (come formato, dimensione e colore dei caratteri di paragrafi, frasi o singole parole, ...) o indicazioni sulla struttura del contenuto (come strutturazione in paragrafi, sottoparagrafi, titoli di primo livello, titoli di secondo livello...)

# Linguaggi di markup (II)

- I principali elementi dei linguaggi di markup usati nelle pagine Web sono i **TAG** e gli **ATTRIBUTI**

- **TAG**

- sono le etichette per mezzo delle quali è possibile annotare porzioni di contenuto e suddividerlo in *elementi*
- in HTML e XHTML ogni elemento è costituito da un contenuto (eventualmente vuoto), delimitato da un tag di inizio e da un tag di fine (HTML è meno rigoroso di XHTML sulla necessità che ad ogni tag di inizio corrisponda sempre un tag di fine)
- I tag di inizio hanno la seguente sintassi `<nomeTag>`, quelli di fine `</nomeTag>`
- Es. `<B>` e `</B>` sono, rispettivamente, il tag HTML di inizio e di fine di una porzione di contenuto testuale che dovrà essere visualizzata in grassetto (bold); `<h1>` e `</h1>` sono, rispettivamente, i tag XHTML di inizio e fine di un titolo di livello 1

# Linguaggi di markup (III)

- **ATTRIBUTI**

- servono a fornire caratterizzazioni aggiuntive per gli elementi
- hanno la forma 'nomeAttributo = "valoreAttributo" '
- sono inseriti nei tag di inizio (dopo il nome del tag)
- Es.

`<a href="http://www.di.unito.it/index.html" >Dipartimento di Informatica</a>`

- specifica che l'elemento il cui contenuto è "Dipartimento di Informatica" è un collegamento ipertestuale (tag "`<a>`", che sta per "**anchor**") che conduce alla risorsa il cui URL è "http://www.di.unito.it/index.html" ed è specificato come valore dell'attributo href

# HTML (I)

HTML (HyperText Markup Language) è:

- un linguaggio di markup per scrivere pagine Web ipertestuali
- definito e mantenuto dal W3C

Rispetto alle tre distinzioni viste prima:

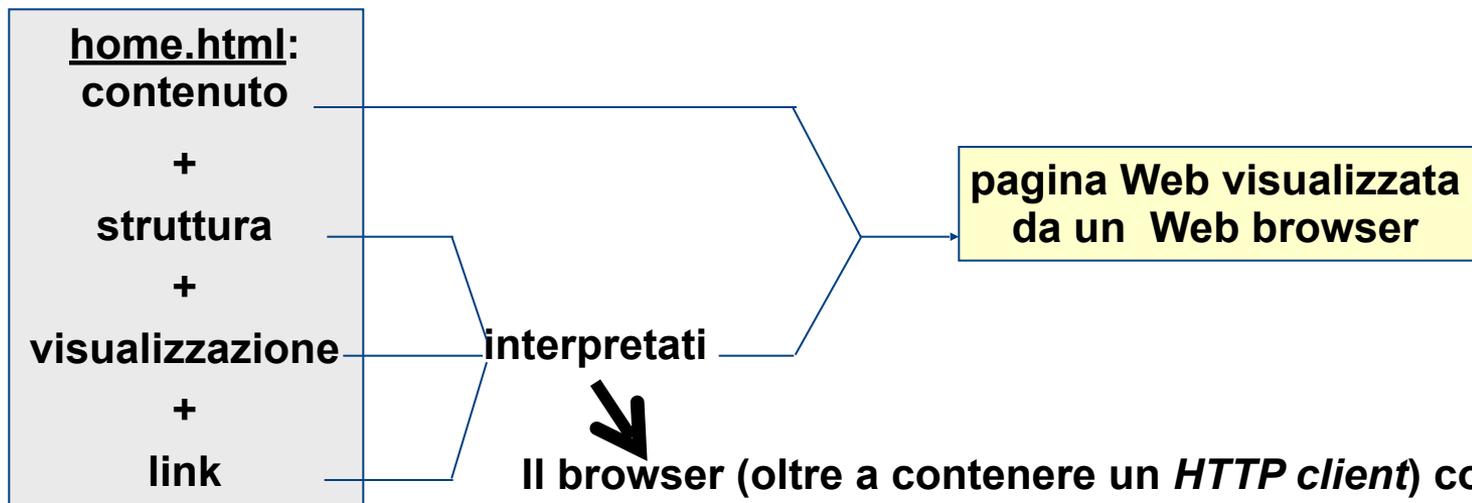
- è un linguaggio di markup e non di scripting/programmazione
- l'uso del solo HTML nella scrittura di pagine Web produce pagine Web statiche
- è una tecnologia client-side (l'HTML è “interpretato” sul client, dal browser)

# HTML (II)

HTML (HyperText Markup Language) è definito e mantenuto dal W3C

Una pagina Web HTML è un documento “annotato” con HTML, cioè un file di testo (per es. home.html) che, oltre al contenuto, contiene meta-informazioni su:

- (a) la **struttura del contenuto**
- (b) la **visualizzazione** (che, però, come vedremo, è bene mantenere separata dalla pagina HTML vera e propria)
- (c) i **link** (connessioni ipertestuali)

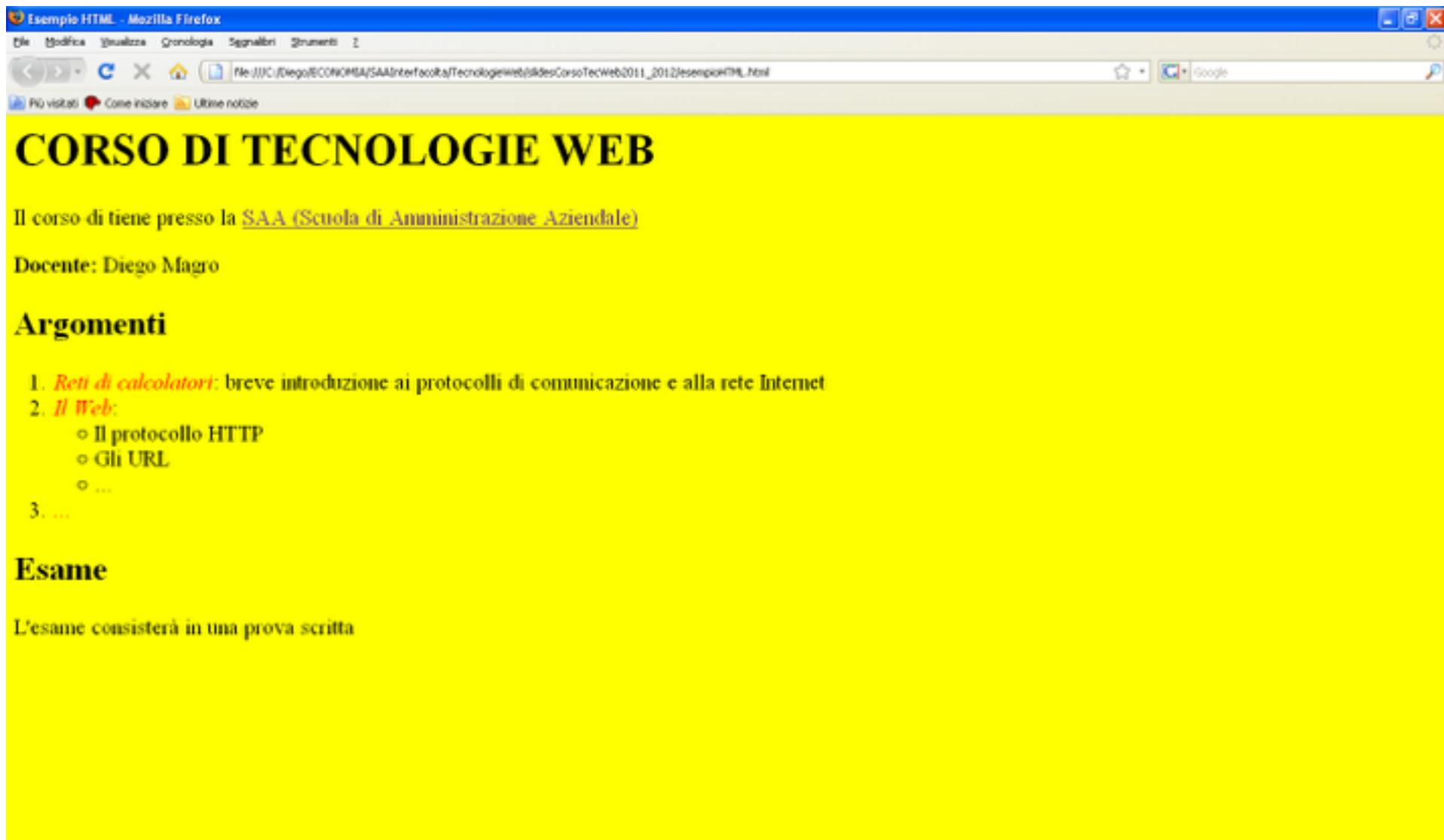


Il browser (oltre a contenere un *HTTP client*) contiene anche un interprete HTML che, interpretando le indicazioni HTML relative alla struttura, quelle relative alla visualizzazione e quelle relative ai link, visualizza la pagina Web

# HTML (III)

```
<html>
<head>
  <title>Esempio HTML</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="yellow">
  <h1>CORSO DI TECNOLOGIE WEB</h1>
  <p>Il corso di tiene presso la <a href="http://www.saa.unito.it">SAA (Scuola di Amministrazione Aziendale) </a><p>
  <p><b>Docente: </b>Diego Magro</p>
  <h2>Argomenti</h2>
  <ol>
    <li><i><font color="red">Reti di calcolatori</font></i>: breve introduzione ai protocolli di comunicazione e alla rete Internet </li>
    <li><i><font color="red">Il Web</font></i>:
      <ul>
        <li>Il protocollo HTTP</li>
        <li>Gli URL</li>
        <li>...</li>
      </ul>
    </li>
    <li><i><font color="red">...</font></i></li>
  </ol>
  <h2>Esame</h2>
  <p>L'esame consistere&agrave; in una prova scritta</p>
</body>
</html>
```

# HTML (IV)



Esempio HTML - Mozilla Firefox

File Modifica Visualizza Cronologia Segnalibri Strumenti ?

file:///C:/Diego/ECONOMIA/SAA/interfacoltà/TecnologieWeb/slides/CorsoTecWeb2011\_2012/EsempioHTML.html

PGI visitati Come iniziare Ultime notizie

## CORSO DI TECNOLOGIE WEB

Il corso di tiene presso la [SAA \(Scuola di Amministrazione Aziendale\)](#)

Docente: Diego Magro

### Argomenti

1. *Reti di calcolatori*: breve introduzione ai protocolli di comunicazione e alla rete Internet
2. *Il Web*:
  - o Il protocollo HTTP
  - o Gli URL
  - o ...
3. ...

### Esame

L'esame consisterà in una prova scritta

# HTML (V)

# TAG

```
<html>
<head>
  <title>Esempio HTML</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="yellow">
  <h1>CORSO DI TECNOLOGIE WEB</h1>
  <p>Il corso di tiene presso la <a href="http://www.saa.unito.it">SAA (Scuola di Amministrazione Aziendale) </a><p>
  <p><b>Docente: </b>Diego Magro</p>
  <h2>Argomenti</h2>
  <ol>
    <li><i><font color="red">Reti di calcolatori</font></i>: breve introduzione ai protocolli di comunicazione e alla rete
Internet </li>
    <li><i><font color="red">Il Web</font></i>:
      <ul>
        <li>Il protocollo HTTP</li>
        <li>Gli URL</li>
        <li>...</li>
      </ul>
    </li>
    <li><i><font color="red">...</font></i></li>
  </ol>
  <h2>Esame</h2>
  <p>L'esame consistere&grave; in una prova scritta</p>
</body>
</html>
```

# HTML (V)

# ATTRIBUTI

```
<html>
<head>
  <title>Esempio HTML</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="yellow">
  <h1>CORSO DI TECNOLOGIE WEB</h1>
  <p>Il corso di tiene presso la <a href="http://www.saa.unito.it">SAA (Scuola di Amministrazione Aziendale) </a><p>
  <p><b>Docente: </b>Diego Magro</p>
  <h2>Argomenti</h2>
  <ol>
    <li><i><font color="red">Reti di calcolatori</font></i>: breve introduzione ai protocolli di comunicazione e alla rete
Internet </li>
    <li><i><font color="red">Il Web</font></i>:
  <ul>
    <li>Il protocollo HTTP</li>
    <li>Gli URL</li>
    <li>...</li>
  </ul>
  </li>
  <li><i><font color="red">...</font></i></li>
  </ol>
  <h2>Esame</h2>
  <p>L'esame consistere&grave; in una prova scritta</p>
</body>
</html>
```



# HTML (V)

# CONTENUTO

```
<html>
<head>
  <title>Esempio HTML</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="yellow">
  <h1>CORSO DI TECNOLOGIE WEB</h1>
  <p>Il corso di tiene presso la <a href="http://www.saa.unito.it">SAA (Scuola di Amministrazione Aziendale) </a><p>
  <p><b>Docente: </b>Diego Magro</p>
  <h2>Argomenti</h2>
  <ol>
    <li><i><font color="red">Reti di calcolatori</font></i>: breve introduzione ai protocolli di comunicazione e
    alla rete Internet </li>
    <li><i><font color="red">Il Web</font></i>:
    <ul>
      <li>Il protocollo HTTP</li>
      <li>Gli URL</li>
      <li>...</li>
    </ul>
    </li>
    <li><i><font color="red">...</font></i></li>
  </ol>
  <h2>Esame</h2>
  <p>L'esame consistere&agrave; in una prova scritta</p>
</body>
</html>
```



# HTML (V)

# LINK

```
<html>
<head>
  <title>Esempio HTML</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="yellow">
  <h1>CORSO DI TECNOLOGIE WEB</h1>

  <p >Il corso di tiene presso la <a href="http://www.saa.unito.it/">SAA (Scuola di Amministrazione Aziendale) </a><p>
  <p><b>Docente: </b>Diego Magro</p>
  <h2>Argomenti</h2>
  <ol>
    <li><i><font color="red">Reti di calcolatori</font></i>: breve introduzione ai protocolli di comunicazione e alla rete
Internet </li>
    <li><i><font color="red">Il Web</font></i>:
    <ul>
      <li>Il protocollo HTTP</li>
      <li>Gli URL</li>
      <li>...</li>
    </ul>
    </li>
  </ol>
  <h2>Esame</h2>
  <p>L'esame consistere in una prova scritta</p>
</body>
</html>
```



# HTML (V)

## ALCUNI TAG SPECIFICANO LA STRUTTURA

```
<html>
<head>
  <title>Esempio HTML</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="yellow">
  <h1>CORSO DI TECNOLOGIE WEB</h1>
  <p>Il corso di tiene presso la <a href="http://www.saa.unito.it">SAA (Scuola di Amministrazione Aziendale) </a><p>
  <p><b>Docente: </b>Diego Magro</p>
  <h2>Argomenti</h2>
  <ol>
    <li><i><font color="red">Reti di calcolatori</font></i>: breve introduzione ai protocolli di comunicazione e alla rete Internet </li>
    <li><i><font color="red">Il Web</font></i>:
      <ul>
        <li>Il protocollo HTTP</li>
        <li>Gli URL</li>
        <li>... </li>
      </ul>
    </li>
  </ol>
  <h2>Esame</h2>
  <p>L'esame consistere&agrave; in una prova scritta</p>
</body>
</html>
```

# HTML (V)

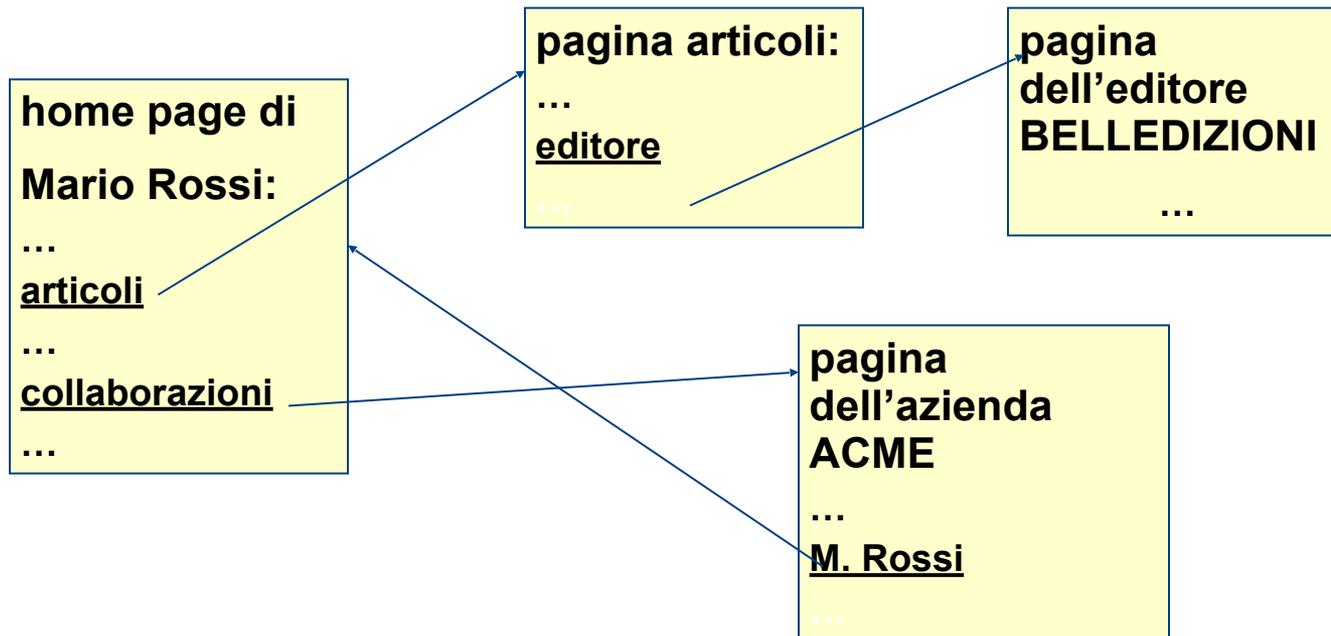
## ALTRI TAG E ALCUNI ATTRIBUTI SPECIFICANO L'ASPETTO

```
<html>
<head>
  <title>Esempio HTML</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="yellow">
  <h1>CORSO DI TECNOLOGIE WEB</h1>
  <p>Il corso di tiene presso la <a href="http://www.saa.unito.it">SAA (Scuola di Amministrazione Aziendale) </a><p>
  <p><b>Docente: </b>Diego Magro</p>
  <h2>Argomenti</h2>
  <ol>
    <li><i><font color="red">Reti di calcolatori</font></i>: breve introduzione ai protocolli di comunicazione e
    alla rete Internet </li>
    <li><i><font color="red">Il Web</font></i>:
    <ul>
      <li>Il protocollo HTTP</li>
      <li>Gli URL</li>
      <li>...</li>
    </ul>
    </li>
    <li><i><font color="red">...</font></i></li>
  </ol>
  <h2>Esame</h2>
  <p>L'esame consistere&agrave; in una prova scritta</p>
</body>
</html>
```



# HTML (VI)

- N.B. I link fanno sì che la pagina diventi (si inserisca all'interno di) un ipertesto



# HTML (VII)

## NB:

- cliccare su un link corrisponde a spostarsi da un nodo ad un'altro nell'ipertesto Web, percorrendo un arco (collegamento ipertestuale)
- dal momento che in un link è specificato un URL, ciò che accade è esattamente la stessa cosa che accade quando scriviamo un URL nella barra degli indirizzi del browser:
  - il browser analizza l'URL
  - chiede al DNS Server l'indirizzo IP corrispondente al nome dell'host contenuto nell'URL
  - effettua una **connessione TCP** all'host corrispondente all'indirizzo IP ricevuto e gli invia una **HTTP request** contenente la risorsa indicata nell'URL e gli eventuali parametri
  - il server **risponde** inviando una HTTP response contenente la risorsa richiesta
  - il browser riceve l'**HTTP response** e **interpreta** la risorsa in essa contenuta (come vedremo...)

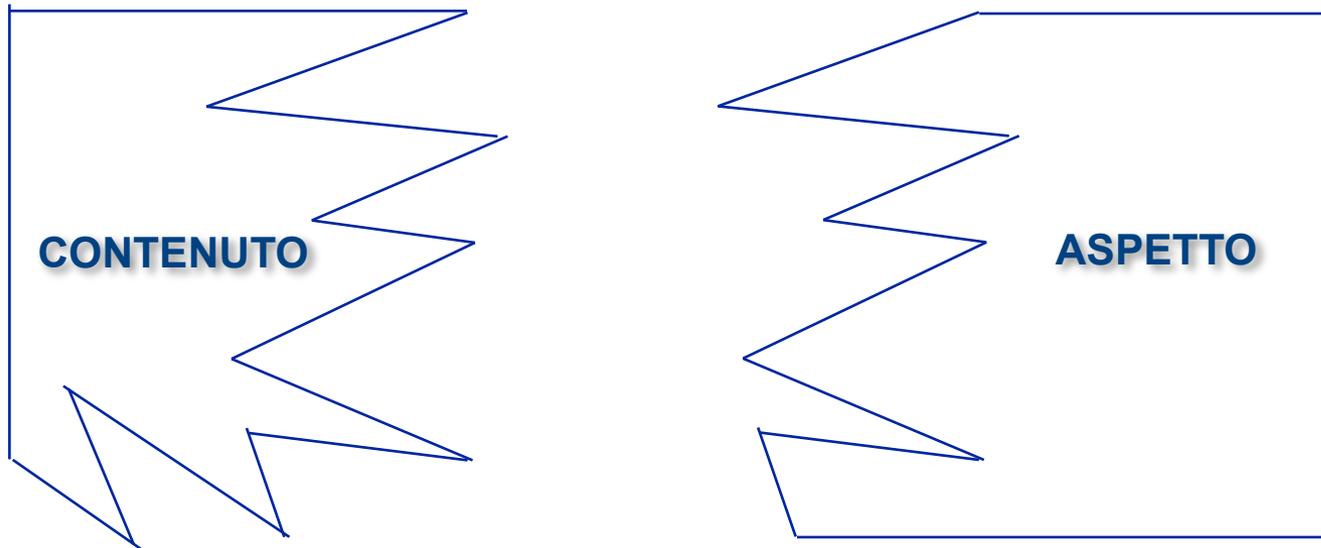
# HTML (VIII)

IN UNA PAGINA HTML E' POSSIBILE MESCOLORE CONTENUTO E ASPETTO...



# HTML (IX)

**...QUESTA NON E' UNA BUONA STRATEGIA DI COSTRUZIONE DELLE PAGINE WEB: CONTENUTO ED ASPETTO DEVONO ESSERE MANTENUTI IL PIU' POSSIBILE SEPARATI!**



# CSS- *Cascading Style Sheets* (I)

Separazione (rispetto al semplice uso di HTML) tra contenuto e aspetto (indicazioni che vengono interpretate dal programma che visualizza il documento)



creazione di un documento separato (foglio di stile) dalla pagina Web (file .html) che contiene le informazioni relative all'aspetto (alla "visualizzazione") della pagina



Principale vantaggio: risparmio di tempo e maggior praticità nella gestione del sito: se si vuole modificare l'aspetto dell'intero sito (magari costituito da molte pagine) non è necessario modificare tutte le pagine, una per una, ma è sufficiente modificare i fogli di stile che contengono gli aspetti che si vogliono cambiare...ma vi sono altri vantaggi, come vedremo

# CSS- Cascading Style Sheets (II)

I fogli di stile più comunemente usati con HTML sono i **CSS (Cascading Style Sheets)**

Le specifiche per i CSS sono prodotte e mantenute dal W3C

Un foglio di stile CSS è costituito da un insieme di regole del tipo:

```
selettore {proprietà1:valore1;  
          proprietà2:valore2;}
```

- è frequente che il selettore sia il nome di un **tag** (ma possono esservi anche selettori più “complessi”)

Per esempio:

```
p {font-family:"Times New Roman";  
  font-size:10pt;  
  color:blue;}
```

dice che i paragrafi di testo (identificati dal tag `<p></p>`) vanno scritti col font Times New Roman, grandezza 10 punti, colore blu



# CSS- Cascading Style Sheets (III)

In realtà le regole per gli stili possono essere posizionate:

1. All'interno di un **singolo tag**
2. All'interno della **singola pagina** (nella sez. *head*)
- 3. In un **file separato**, con estensione **.css**

Il caso 3 è in realtà l'unico che garantisce i vantaggi menzionati (netta separazione contenuto/aspetto) + risparmio di banda (infatti, il file **.css** viene salvato nella cache del browser e utilizzato per più pagine senza dover essere riscaricato)

All'interno delle pagine del sito occorre inserire un riferimento al file contenente il foglio di stile:

```
<head>
  ...
  <link rel="stylesheet" type="text/css"
        href="miostile.css">
</head>
```

# CSS- Cascading Style Sheets (IV)

```
<html>
<head>
  <title>Esempio HTML con CSS</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <link rel="stylesheet" type="text/css" href="miostile.css">
</head>
<body>
  <h1>CORSO DI TECNOLOGIE WEB</h1>
  <p>Il corso di tiene presso la <a href="http://www.saa.unito.it">SAA (Scuola di Amministrazione Aziendale) </a><p>
  <p> <strong> Docente: </strong> Diego Magro</p>
  <h2>Argomenti</h2>
  <ol>
    <li> <strong> Reti di calcolatori </strong> : breve introduzione ai protocolli di comunicazione e alla rete Internet </li>
    <li> <strong> Il Web </strong> :
      <ul>
        <li>Il protocollo HTTP</li>
        <li>Gli URL</li>
        <li> <strong>... </strong> </li>
      </ul>
    </li>
  </ol>
  <h2>Esame</h2>
  <p>L'esame consistere&grave; in una prova scritta</p>
</body>
</html>
```



# CSS- Cascading Style Sheets (V)

**NEL FILE HTML SOLO MARKUP PER LA STRUTTURA (NO TAG E ATTRIBUTI CHE SPECIFICANO L'ASPETTO)**

```
<html>
<head>
  <title>Esempio HTML con CSS</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <link rel="stylesheet" type="text/css" href="miostile.css">
</head>
<body>
  <h1>CORSO DI TECNOLOGIE WEB</h1>
  <p>Il corso di tiene presso la <a href="http://www.saa.unito.it">SAA (Scuola di Amministrazione Aziendale) </a><p>
  <p> <strong> Docente: </strong> Diego Magro</p>
  <h2>Argomenti</h2>
  <ol>
    <li> <strong> Reti di calcolatori </strong> : breve introduzione ai protocolli di comunicazione e alla rete
    Internet </li>
    <li> <strong> Il Web </strong> :
    <ul>
      <li>Il protocollo HTTP</li>
      <li>Gli URL</li>
      <li> <strong>... <strong> </li>
    </ul>
  </li>
  <li>...</li>
</ol>
  <h2>Esame</h2>
  <p>L'esame consistere&grave; in una prova scritta</p>
</body>
</html>
```



# CSS- *Cascading Style Sheets* (VI)

## L'ASPETTO SPECIFICATO IN UN FILE SEPARATO: MIOSTILE.CSS

```
body {  
  background-color: yellow;  
}
```

```
strong {  
  font-weight: bold;  
}
```

```
li strong {  
  font-weight: normal;  
  font-style: italic;  
  color: red;  
}
```

# CSS- Cascading Style Sheets (VII)

(altro esempio: *miostile1.css*)

`body { background-color: white; }` **proprietà generali della pagina**

`a:link { color: red; }`

`a:visited { color: maroon; }`

`a:active { color: purple; }`

**proprietà dei link**

`p { font-family: Verdana, Arial, Helvetica, sans-serif;`

`font-size: 10pt;`

`color: black; }`

**proprietà dei paragrafi**

`ul { font-family: Verdana, Arial, Helvetica, sans-serif;`

`font-size: 10pt;`

`color: black;`

`font-style: italic; }`

**proprietà delle liste**

`h1 { font-family: "Comic Sans MS";`

`font-size: 18pt;`

`color: green;`

`font-weight: bold;`

`text-align: center; }`

**proprietà dei titoli (di primo livello)**

`h2 { font-family: "Comic Sans MS";`

`font-size: 14pt;`

`color: olive;`

`font-weight: bold;`

`text-align: center; }`

**proprietà dei titoli (di secondo livello)**

# CSS- Cascading Style Sheets(VIII)

(altro esempio: *miostile1.css*)

body . . .  
p . . .  
ul . . .

h1  
h2 . . .

a:link . . .  
a:visited . . .

The screenshot shows a web browser window with the following content:

- Header: **La pubblicazione on-line** (h1)
- Section: **Progettazione e costruzione di siti Web** (h2)
- Left Column: **Materiale e riferimenti bibliografici:**
  - *Principi di Interazione Uomo-Macchina*
  - *Progettazione di siti Web*
  - *Architettura dell'informazione*
  - *Realizzazione di siti Web*
  - *Usabilità*
- Right Column: **Informazioni generali:**
  - *Durata: 30 ore*
  - *Crediti (CFU): 5*
  - [Orario](#)
  - [Programma](#)
  - [Verifica finale](#)
  - *Docenti:*
    - [Giovanna Petrone](#)
    - [Cristina Gena](#)
    - [Anna Goy](#)

Blue arrows indicate the mapping of CSS selectors to page elements: 'body' points to the page container; 'p' points to the header and section titles; 'ul' points to the list of bibliographic references; 'h1' points to the main title; 'h2' points to the section title; 'a:link' points to the 'Orario' link; and 'a:visited' points to the 'Programma' link.

```
body {background-color:yellow}
a:link    {color:blue}
a:visited {color:navy}
a:active  {color:teal}
p {font-family:"Times New Roman";
   font-size:12pt;
   color:black}
ul {font-family:"Times New Roman";
    font-size:10pt;
    color:black;
    font-style:normal}
h1 {font-family:"Courier New";
     font-size:18pt;
     color:#660099;
     font-weight:bold;
     text-align:left}
h2 {font-family:"Courier New";
     font-size:14pt;
     color:#9933CC;
     font-weight:bold;
     text-align:left}
```

**body...**  
**p...**  
**ul...**

**h1...**  
**h2...**

**a:link...**  
**a:visited...**

La pubblicazione on-line  
Progettazione e costruzione di siti Web

Materiale e riferimenti bibliografici

- Principi di Interazione Uomo-Macchina
- Progettazione di siti Web
- Architettura dell'informazione
- Realizzazione di siti Web
- Usabilità

Informazioni generali:

- Durata: 30 ore
- Crediti (CFU): 5
- [Orario](#)
- [Programma](#)
- [Verifica finale](#)
- Docenti:  
[Giovanna Petrone](#)  
[Cristina Gena](#)  
[Anna Goy](#)

Operazione completata

Risorse del computer

# CSS- Cascading Style Sheets(XVI) :

## PIU' PRESENTAZIONI PER UNO STESSO CONTENUTO

- Associando più CSS alternativi ad uno stesso contenuto, è possibile prevedere diverse modalità di presentazione per quello stesso contenuto
- Questo consente di:
  - dare ad un utente la possibilità di scegliere fra varie opzioni per la **presentazione di uno stesso contenuto** (ad esempio, è possibile prevedere una presentazione alternativa a quella “di default”, con caratteri ingranditi, font semplici, migliore contrasto contenuto/sfondo, ecc. per utenti ipovedenti)
  - **fornire presentazioni diverse in base al dispositivo di presentazione** ...almeno in linea di principio...anche se non tutte le potenzialità si sono già trasformate in specifiche tecniche, tecnologie e pratiche consolidate... (es., presentazioni alternative per browser visuali “classici”, browser visuali su schermi di piccole dimensione [es. smartphone,...], stampe, browser vocali, dispositivi braille, ecc.): il browser carica e applica i fogli di stile in base al dispositivo

# ESEMPIO CSS ALTERNATIVI PER STESSO CONTENUTO (I)

```
<html>
<head>
  <title>Esempio HTML con CSS</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <link rel="alternate stylesheet" title = "Caratteri molto grandi" type="text/css" href="miostileAlt1.css">
  <link rel="alternate stylesheet" title = "Caratteri molto piccoli" type="text/css" href="miostileAlt2.css">
  <link rel="stylesheet" title = "Stile di default" type="text/css" href="miostile.css" >
</head>
<body>
  <h1>CORSO DI TECNOLOGIE WEB</h1>
  <p>Il corso di tiene presso la <a href="http://www.saa.unito.it/">SAA (Scuola di Amministrazione Aziendale) </a><p>
  <p><strong>Docente: </strong>Diego Magro</p>
  <h2>Argomenti</h2>
  <ol>
    <li><strong>Reti di calcolatori</strong>: breve introduzione ai protocolli di comunicazione e alla rete Internet </li>
    <li><strong>Il Web</strong>:
      <ul>
        <li>Il protocollo HTTP</li>
        <li>Gli URL</li>
        <li>...</li>
      </ul>
    </li>
    <li><strong>...</strong></li>
  </ol>
  <h2>Esame</h2>
  <p>L'esame consistere&agrave; in una prova scritta</p>
</body>
</html>
```

# ESEMPIO CSS ALTERNATIVI PER STESSO CONTENUTO (II)



Esempio HTML con CSS - Mozilla Firefox

File | Strumenti | Cronologia | Segnalibri | Strumenti | ?

PC:\Diego\ECONOMIA\SAA\Interfacoltà\tecnologieWeb\slides\CorsoTecWeb2011\_2012\esempioHTMLConCSSalternativi.html

Google

## COI TECNOLOGIE WEB

Il corso c... (Amministrazione Aziendale)

Docente: Diego Magro

### Argomenti

1. *Reti di calcolatori*: breve introduzione ai protocolli di comunicazione e alla rete Internet
2. *Il Web*:
  - o Il protocollo HTTP
  - o Gli URL
  - o ...
3. ....

### Esame

L'esame consisterà in una prova scritta

# ESEMPIO CSS ALTERNATIVI PER STESSO CONTENUTO (III)



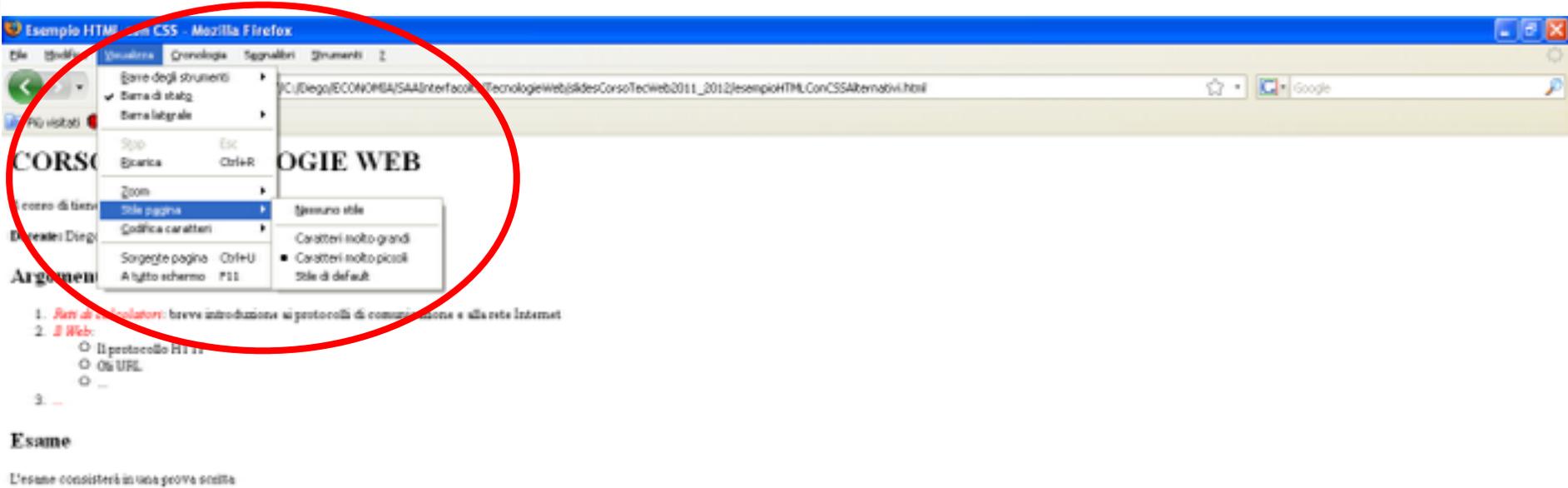
Il corso di tiene presso la [SAA \(Scuola di Amministrazione Aziendale\)](#)

**Docente:** Diego Magro

## Argomenti

1. *Reti di calcolatori*: breve introduzione ai protocolli di comunicazione e alla rete Internet
2. *Il Web*:
  - Il protocollo HTTP

# ESEMPIO CSS ALTERNATIVI PER STESSO CONTENUTO (IV)



# RIASSUNTO DEI PRINCIPALI VANTAGGI DELL'USO DEI CSS

(E DELLA SEPARAZIONE CONTENUTO/ASPETTO CHE ESSI CONSENTONO)

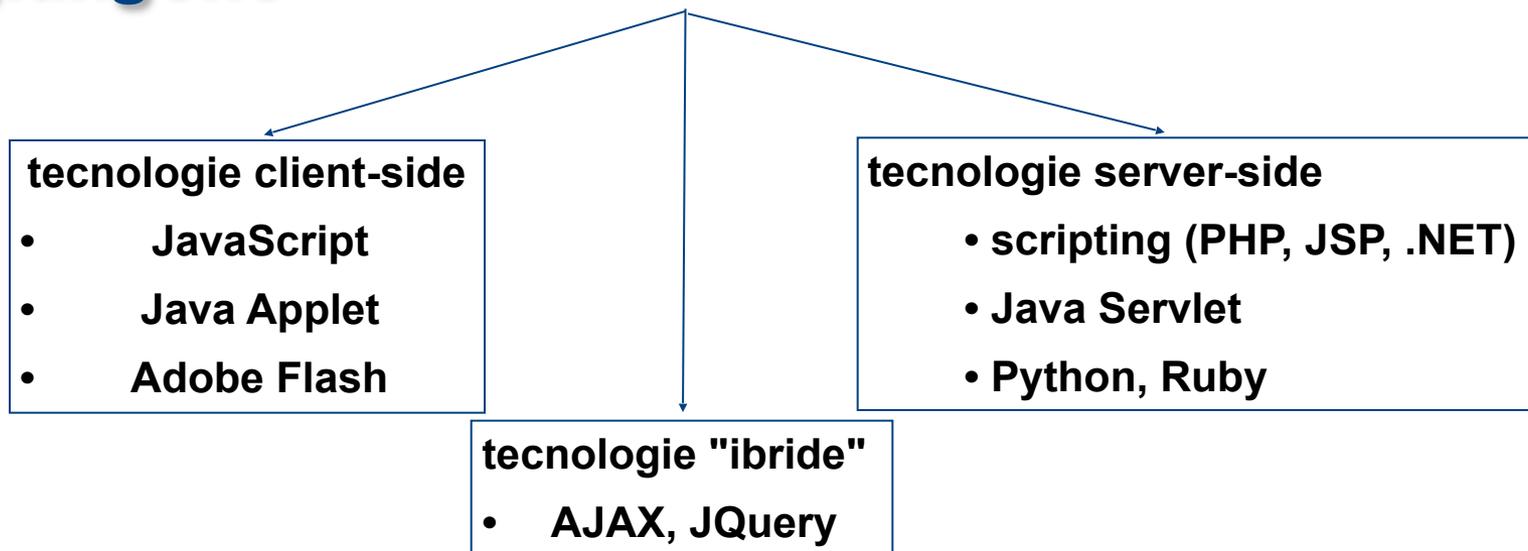
- **Risparmio di tempo nella costruzione di un sito:** gli aspetti comuni nella presentazione delle pagine vengono “estrapolati” e specificati in fogli di stile, anziché ripetuti in ogni pagina
- **Facilitazione nel mantenimento di un sito:** per cambiare l’aspetto di un sito è sufficiente modificare i (presumibilmente pochi) fogli di stile che specificano le caratteristiche di presentazione che si vogliono modificare; se queste caratteristiche fossero ripetute su ogni singola pagina, bisognerebbe modificare ogni singola pagina!
- **Riduzione del consumo di banda:** i browser che visitano un sito, scaricano i fogli di stile collegati alle pagine Web che ricevono e li salvano nella loro cache: in questo modo, uno stesso foglio di stile può essere usato per più pagine senza essere richiesto ogni volta al server; inoltre, le singole pagine non sono appesantite dall’informazione di presentazione
- **Possibilità di realizzare più presentazioni diverse per uno stesso contenuto, senza duplicare i file del contenuto**
- **Facilitazione nell’uso di tecnologie assistive (es. browser vocali, ecc.)**

# ...PERO' ATTENZIONE!

- ...non tutti i browser supportano i CSS nello stesso modo!
- → nell'uso dei CSS (soprattutto delle loro caratteristiche più avanzate o meno consolidate), occorre preoccuparsi di capire se le funzionalità CSS che intendiamo usare sono effettivamente supportate dai browser che ci interessano (perché sono i più diffusi, perché sono i più usati dagli utenti a cui ci rivolgiamo...)

# SITI WEB CON PAGINE DINAMICHE: Principali tecnologie (I)

- Alle tecnologie citate sopra, nella realizzazione di pagine Web dinamiche si aggiungono:



# SITI WEB CON PAGINE DINAMICHE: Principali tecnologie (II)

rispetto alle tre distinzioni viste prima, queste soluzioni:

- sono linguaggi di **programmazione** o di **scripting** (e non di mark-up)
- servono a scrivere pagine "**dinamiche**"
- alcune sono tecnologie **client-side** (il codice è elaborato sul client, dal browser), altre sono **server-side** (il codice è elaborato sul server)

# SITI WEB CON PAGINE DINAMICHE:

## Principali tecnologie (III)

- Non abbiamo qui lo scopo di esaminare nei dettagli nessuna di queste tecnologie, né la pretesa di elencarle tutte
- ...ciò che ci interessa è capire le principali caratteristiche dei diversi approcci
- A questo fine, dobbiamo introdurre il concetto di **scripting** e, per far questo, dobbiamo prima capire cos'è un **programma**

# ALGORITMO

- **Algoritmo**: impossibile darne una precisa definizione formale. Informalmente, è la specifica di un procedimento per la risoluzione di un qualche tipo di problema.
- Un algoritmo, solitamente, specifica la sequenza di passi che deve essere eseguita per risolvere una certa tipologia di problemi, con gli eventuali punti di scelta e le eventuali ripetizioni di sequenze di istruzioni

# PROGRAMMA

- Programma: è la specifica di un algoritmo in un linguaggio “comprensibile” ad un calcolatore
- Il processore è in grado di eseguire direttamente solo istruzioni macchina, pertanto, per poter essere eseguito direttamente dall’hardware, un algoritmo deve essere espresso in linguaggio macchina
- Le istruzioni di un linguaggio macchina sono espresse in formato digitale (linguaggio macchina è “a basso livello”, cioè “lontano” dall’utente umano e “vicino” alla macchina)
- → il linguaggio macchina è poco adatto agli utenti umani → esistono linguaggi di programmazione “ad alto livello” (cioè più “vicini” all’utente umano) in cui il programmatore può specificare gli algoritmi

# LINGUAGGI DI PROGRAMMAZIONE AD ALTO LIVELLO

Fra i linguaggi di programmazione "ad alto livello" troviamo: *Fortran, Cobol, Pascal, C, Lisp, Prolog, Smalltalk, C++, Java, PHP, Python, Ruby...e molti altri!*

**Esempio:  
istruzioni  
scritte in  
Java**

```
public String stampaIscritti() {  
    String risultato = "";  
    int n=0;  
    while (n<listaStudenti.size()) {  
        Studente iscritto = lista.get(n);  
        risultato += n+1 + iscritto.getNome();  
        n++;  
    }  
    return ris;  
}
```

# LINGUAGGI MACCHINA

**Non esiste un unico linguaggio macchina: ogni tipo di “macchina” (caratterizzato approssimativamente dal tipo di processore e dal sistema operativo) ha il proprio**

**Esempio:  
istruzioni  
scritte in  
linguaggio  
macchina**

```
0100010 0011 0101011111001 0101 0000111
01010 00001 01 010 10110010 1010000111
00010 0011 0101011111001 010100 0000111
10 0011 010100000001 0101111 000011101
00010 0011 010101111100111 01010 000111
0 0011 0101011111000001 0101 0000111 00
010 011 0000111001 01010111 00111 0001
01010 0011 10001011111001 0101 00001110
0100 0011 011101011 0000 1100 00001100
010010 0111011 01100 010101001 0000111
```

# SCRIPT E LINGUAGGI DI SCRIPTING

**Script** = piccoli programmi contenuti nelle pagine Web

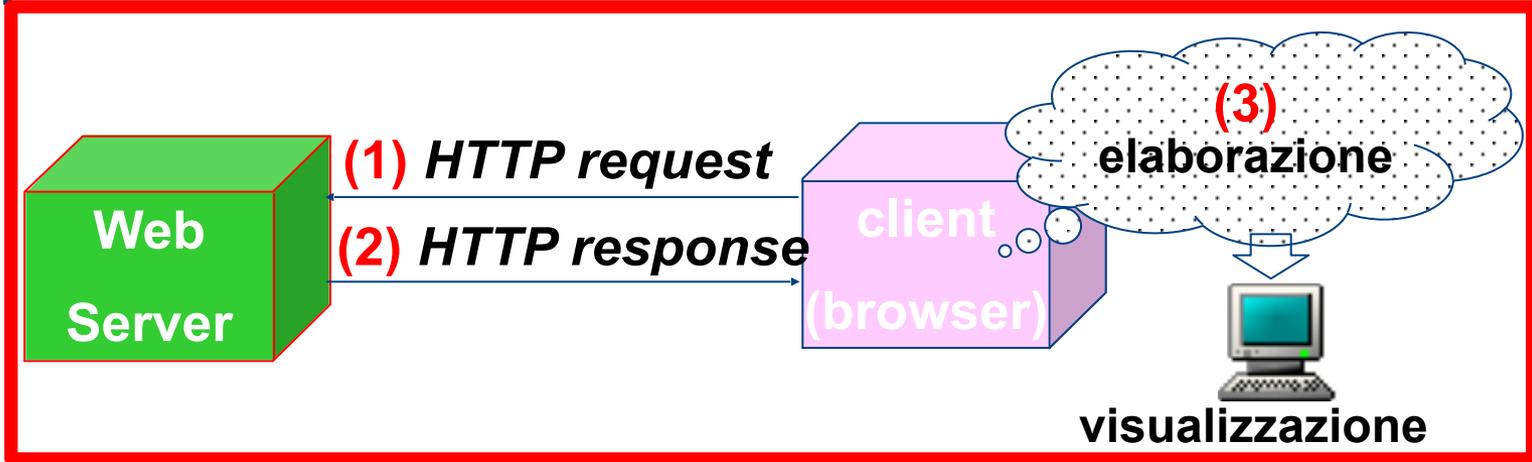
**Linguaggi di scripting** = linguaggi di programmazione "ad alto livello" utilizzati per scrivere script; esempi: *Javascript, PHP, Python, Perl, Visual Basic, C# (C sharp),...*

**Esempio:**  
**pagina**  
**HTML**  
**con script**  
**PHP**

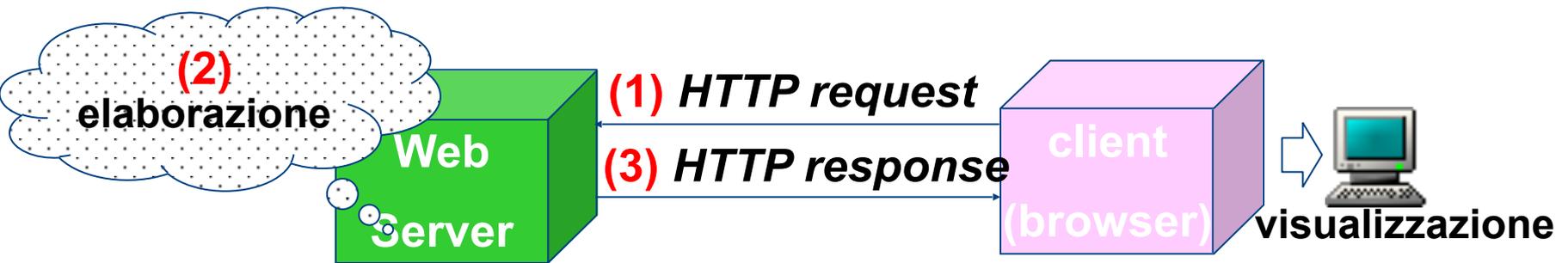
```
<p class="standard">  
  <strong>Es. di script (php)</strong>  
</p>  
<?php  
  $id = $_GET["user"];  
  if ($id == "admin")  
    echo "Buongiorno " . $id;  
  else  
    echo "Accesso negato";  
?>
```

# PRINCIPALI TECNOLOGIE CLIENT-SIDE

Client-side:



Server-side:



# JAVASCRIPT (I)

**JavaScript è un linguaggio di scripting = linguaggio di programmazione "ad alto livello" che serve per scrivere degli script (piccoli programmi contenuti nelle pagine Web)**

**Come tutti i linguaggi di scripting, è interpretato**

**L'interprete degli script JavaScript è contenuto nel browser ⇒ tecnologia client-side: è il client (il browser) che interpreta (ed esegue) gli script Javascript**

**Gli script JavaScript vengono utilizzati soprattutto per aggiungere interattività alle pagine Web**

Nota: JavaScript (nonostante la somiglianza nel nome) è un linguaggio completamente separato da Java

# JAVASCRIPT (II)

⇒ rispetto alle tre distinzioni viste prima, JavaScript:

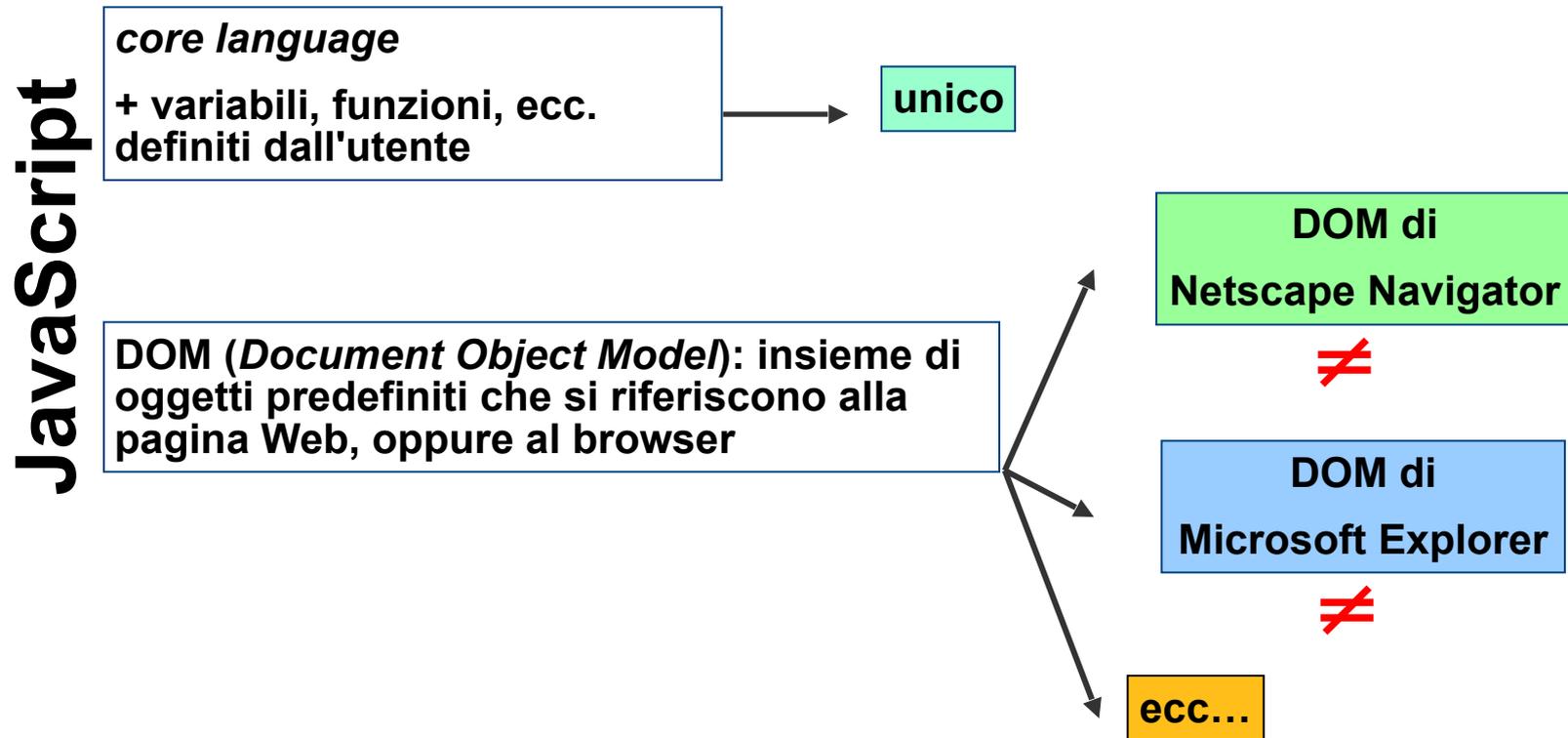
- è un linguaggio di **scripting** (quindi un particolare tipo di linguaggio di programmazione e NON di markup)
- serve a scrivere pagine "**dinamiche**"
- è una tecnologia **client-side** (il codice è elaborato sul client, dal browser)... anche se esistono interessanti applicazioni di JavaScript lato server (che però noi non vedremo)

Mediante JavaScript è possibile creare pagine Web in grado di intercettare alcuni eventi prodotti dall'utente e di interagire con vari elementi del browser e della pagina stessa, per esempio:

- fa sì che, **al passaggio del mouse**, un'immagine cambi (*roll-over*)
- far sì che, **al click del mouse** su un link, venga eseguita una **funzione** (per es. di stampa)
- fare aprire **nuove finestre**, con proprietà controllate (dimensioni, menu, ecc.) e gestire oggetti (per es. immagini) contenuti in tali finestre
- ...molte altre cose...

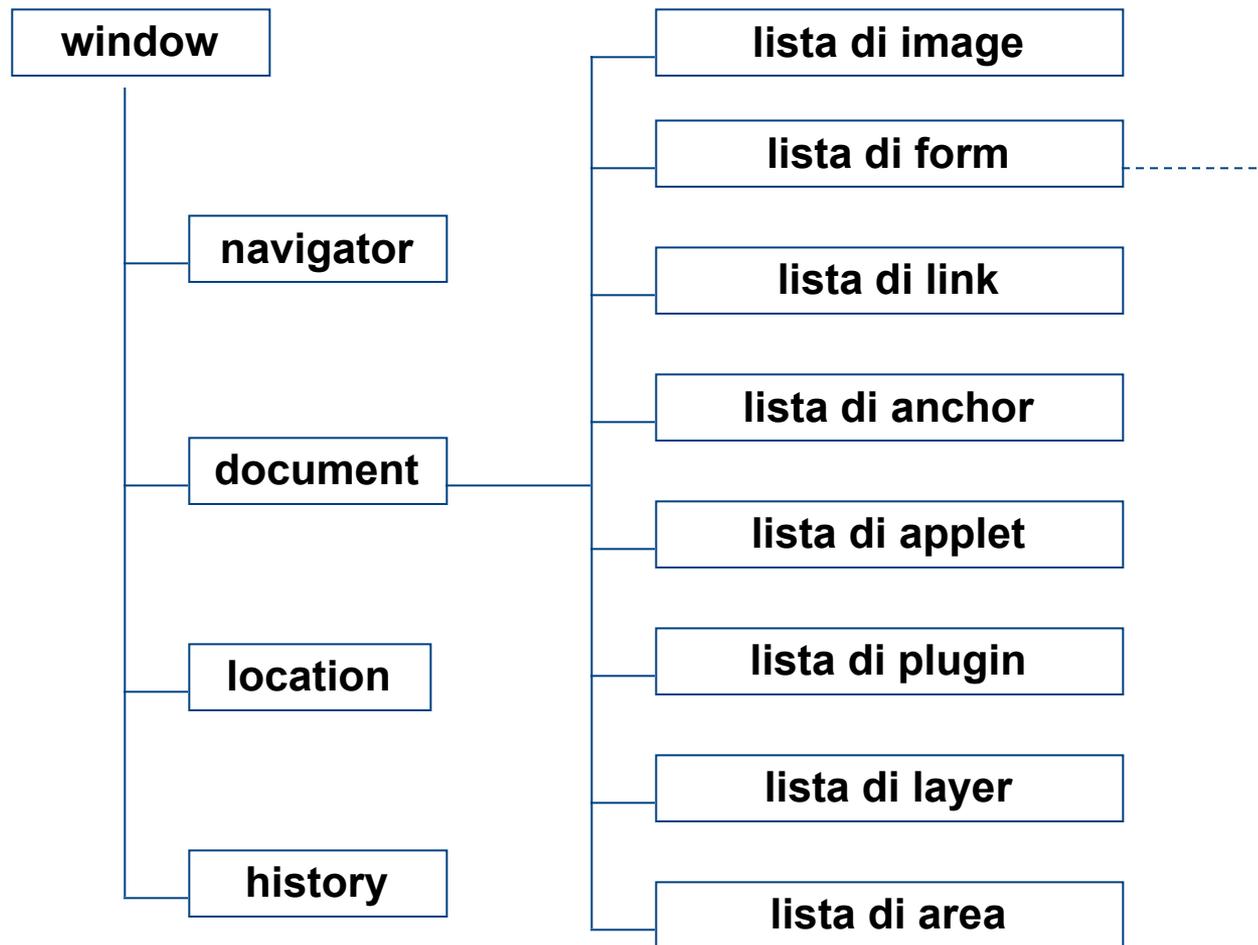
# JAVASCRIPT (IV)

- JavaScript si basa su due concetti principali:
  1. il **DOM** (*Document Object Model*)
  2. gli **eventi** (per questo si dice che gli script JavaScript sono *event-driven*)



**NB: Il DOM è definito dal browser! E la definizione è fatta separatamente da per Firefox, MS Explorer, Mozilla, ecc!**  
⇒ possono nascere incompatibilità...

## Il DOM è organizzato secondo una gerarchia:



# JAVASCRIPT (VII)

## Il DOM - III

↑  
**window**

↑  
**document**

Scienze della Comunicazione  
Università di Torino

Home Corsi di laurea Didattica Per le aziende

### Iscrizione agli appelli

Per accedere al servizio bisogna essere utenti registrati.  
Se sei già registrato, digita il tuo numero di matricola e la password.  
Altrimenti puoi [registrarti ora](#)

Matricola  
  
Password

[Per ulteriori informazioni](#)

Torino Segreteria Scidecom  
Via Verdi 25  
Tel 011 883040

Ivrea (TO) Segreteria Scidecom Ivrea  
Ex Officina II - Via Monte navale 11  
Tel 0125 627665

### Gli script Javascript sono tipicamente "guidati dagli eventi" (*event-driven*):

- **Eventi** = azioni dell'utente sulla pagina Web: ogni volta che l'utente fa click su un link, scrive qualcosa nella casella di un modulo online, preme un pulsante, ridimensiona una finestra, ecc... genera un "evento"
  - ⇒ uno script JavaScript contiene un **gestore di eventi** (*event handler*), in grado di intercettare le azioni dell'utente (eventi) ed eseguire, di conseguenza, delle istruzioni
- Il **DOM** fornisce una serie di **gestori di eventi predefiniti** (cioè capaci di intercettare diversi tipi di azioni dell'utente): l'accadere di un evento nella pagina attiva il corrispondente gestore di eventi, che eseguirà le istruzioni assegnategli

# JAVASCRIPT (X)

⇒ grazie agli oggetti definiti nel DOM e ai gestori di eventi, negli script Javascript è possibile reagire agli eventi (azioni dell'utente) ed interagire con vari elementi del browser e della pagina Web ⇒ rendere interattive le pagine Web

**Per esempio:**

- fa sì che, **al passaggio del mouse**, un'immagine cambi
- far sì che, **al click del mouse** su un link, venga eseguita una **azione** (per es. di stampa)
- far sì che, **al click del mouse** su un link, venga visualizzata un'immagine
- fare aprire **nuove finestre**, con proprietà controllate (dimensioni, menu, ecc.)
- ecc...

# JAVASCRIPT (XI)

Per ottenere questi effetti occorre:

1. Intercettare l'azione ("evento") dell'utente, per es. il passaggio o il click del mouse
2. In corrispondenza di tale azione ("evento") eseguire la funzione desiderata

Per esempio:

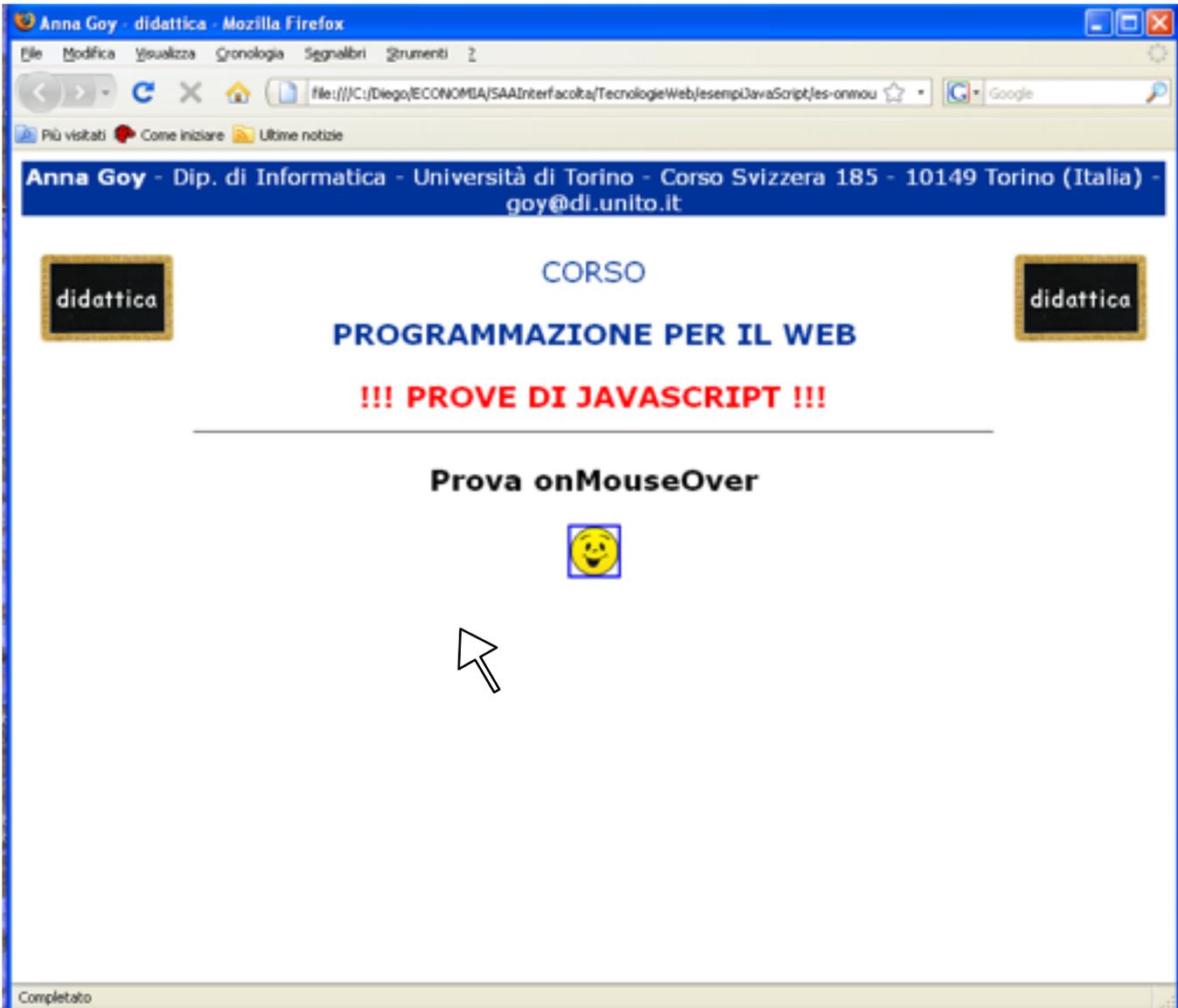
```
... onMouseOver = "document.il_sole.src='soleRosso.gif';"  
   onMouseOut  = "document.il_sole.src='sole.gif';"...
```

```
... onClick = "window.print()"
```

intercetto l'azione (evento)

eseguo la funzione

# JAVASCRIPT (XII)



# JAVASCRIPT (XIII)

Anna Goy - didattica - Mozilla Firefox

file:///C:/Diego/ECONOMIA/SAAInterfacolta/TecnologieWeb/lesemplJavaScript/es-onmou

Anna Goy - Dip. di Informatica - Università di Torino - Corso Svizzera 185 - 10149 Torino (Italia) - goy@di.unito.it

didattica CORSO didattica

**PROGRAMMAZIONE PER IL WEB**

**!!! PROVE DI JAVASCRIPT !!!**

---

**Prova onmouseover**



← Faccina gialla

Completato



# JAVASCRIPT (XIV)

The screenshot shows a Mozilla Firefox browser window with the following content:

- Address bar: file:///C:/Diego/ECONOMIA/SAAInterfaccolta/TecnologieWeb/lesemplJavaScript/es-onmou
- Page title: Anna Goy - Dip. di Informatica - Università di Torino - Corso Svizzera 185 - 10149 Torino (Italia) - goy@di.unito.it
- Page content:
  - Two "didattica" logos on the left and right.
  - Center text: CORSO PROGRAMMAZIONE PER IL WEB
  - Center text: **!!! PROVE DI JAVASCRIPT !!!**
  - Section title: Prova onMouseOver
  - A yellow smiley face emoji with a red mouse cursor icon pointing at it.
- Status bar: Completato

**Evento: L'utente sposta il mouse fino a sovrapporlo alla faccina**

# JAVASCRIPT (XV)

Anna Goy - didattica - Mozilla Firefox

file:///C:/Diego/ECONOMIA/SAAInterfacolta/TecnologieWeb/esempiJavaScript/es-onmou

Anna Goy - Dip. di Informatica - Università di Torino - Corso Svizzera 185 - 10149 Torino (Italia) - goy@di.unito.it

didattica

CORSO

PROGRAMMAZIONE PER IL WEB

!!! PROVE DI JAVASCRIPT !!!

Prova onMouseOver

Reazione: La faccina cambia colore

file:///C:/Diego/ECONOMIA/SAAInterfacolta/TecnologieWeb/esempiJavaScript/es-onmouseover.html#

# JAVASCRIPT (XVI)



The screenshot shows a Mozilla Firefox browser window with the following content:

- Address bar: file:///C:/Diego/ECONOMIA/SAAInterfacolta/TecnologieWeb/esempiJavaScript/es-onmou
- Page title: Anna Goy - Dip. di Informatica - Università di Torino - Corso Svizzera 185 - 10149 Torino (Italia) - goy@di.unito.it
- Page content:
  - Two "didattica" logos in black boxes with yellow borders.
  - Text: CORSO
  - Text: PROGRAMMAZIONE PER IL WEB
  - Text: **!!! PROVE DI JAVASCRIPT !!!**
  - Text: Prova onMouseOver
  - A red circle highlights a mouse cursor icon pointing at a small red square, with a red arrow indicating the mouse movement.

Event: L'utente sposta il mouse fuori dall'area della faccina

# JAVASCRIPT (XVII)



**Reazione: La faccina cambia nuovamente colore**

# JAVASCRIPT (XVIII)

- **Questo comportamento dinamico della pagina è il risultato dell'esecuzione sul client (da parte dell'interprete Javascript) di uno script Javascript contenuto nella pagina Web**
- ...naturalmente, questo è un esempio di comportamento dinamico di una pagina Web piuttosto banale: Javascript (e, in generale, le tecnologie dinamiche client-side) consentono di realizzare comportamenti più sofisticati...

# JAVASCRIPT (XIX)

```
<html>
```

```
<head>
```

Un frammento del sorgente della pagina Web dell'esempio precedente

```
...
```

```
</head>
```

```
<body>
```

```
...
```

```
<p >
```

```
<strong>Prova onMouseOver</strong>
```

```
</p>
```

```
<div id="sole">
```

```
<p> <a href="#" onMouseOver="document.il_sole.src='soleRosso.gif';"  
  onMouseOut="document.il_sole.src='sole.gif';">
```

```
</a> </p>
```

```
</div>
```

```
</body>
```

```
</html>
```

# JAVASCRIPT (XX)

```
<html>
```

```
<head>
```

```
...
```

```
</head>
```

```
<body>
```

```
...
```

```
<p >
```

```
<strong>Prova onMouseOver</strong>
```

```
</p>
```

```
<div id="sole">
```

```
<p > <a href="#" onMouseOver="document.il_sole.src='soleRosso.gif';"  
onMouseOut="document.il_sole.src='sole.gif';">
```

```
</a> </p>
```

```
</div>
```

```
</body>
```

```
</html>
```

**“Normali” tag, attributi e valori HTML**

# JAVASCRIPT (XXI)

```
<html>
```

```
<head>
```

```
...
```

```
</head>
```

```
<body>
```

```
...
```

```
<p >
```

```
<strong>Prova onMouseOver</strong>
```

```
</p>
```

```
<div id="sole">
```

Script Javascript

```
<p> <a href="#"
```

```
  onMouseOver="document.il_sole.src='soleRosso.gif';"
```

```
  onMouseOut="document.il_sole.src='sole.gif';">
```

```
</a> </p>
```

```
</div>
```

```
</body>
```

```
</html>
```

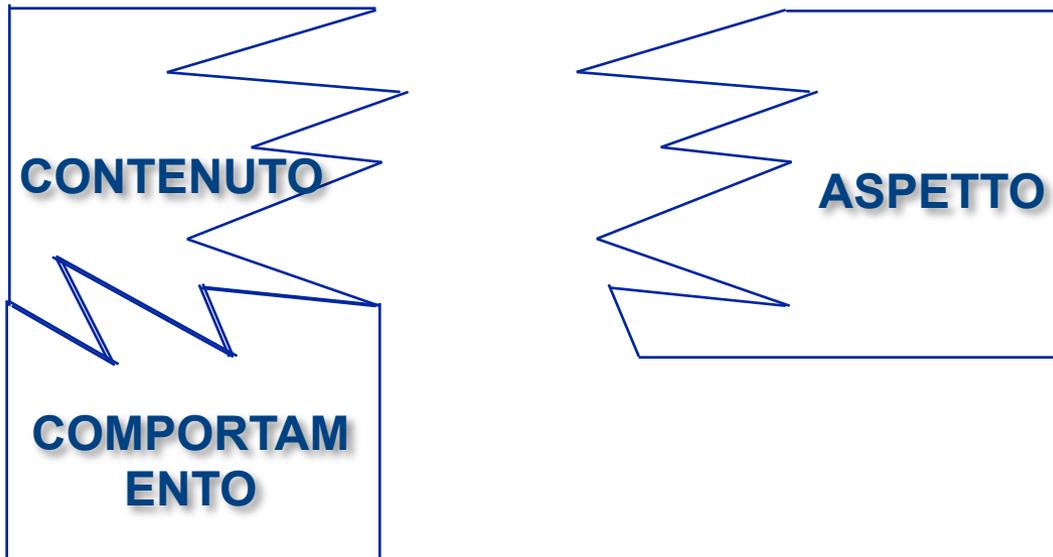
# JAVASCRIPT (XXII)

...RICORDATE L'OPPORTUNITA' DI SEPARARE ASPETTO E CONTENUTO  
NELLE PAGINE WEB?



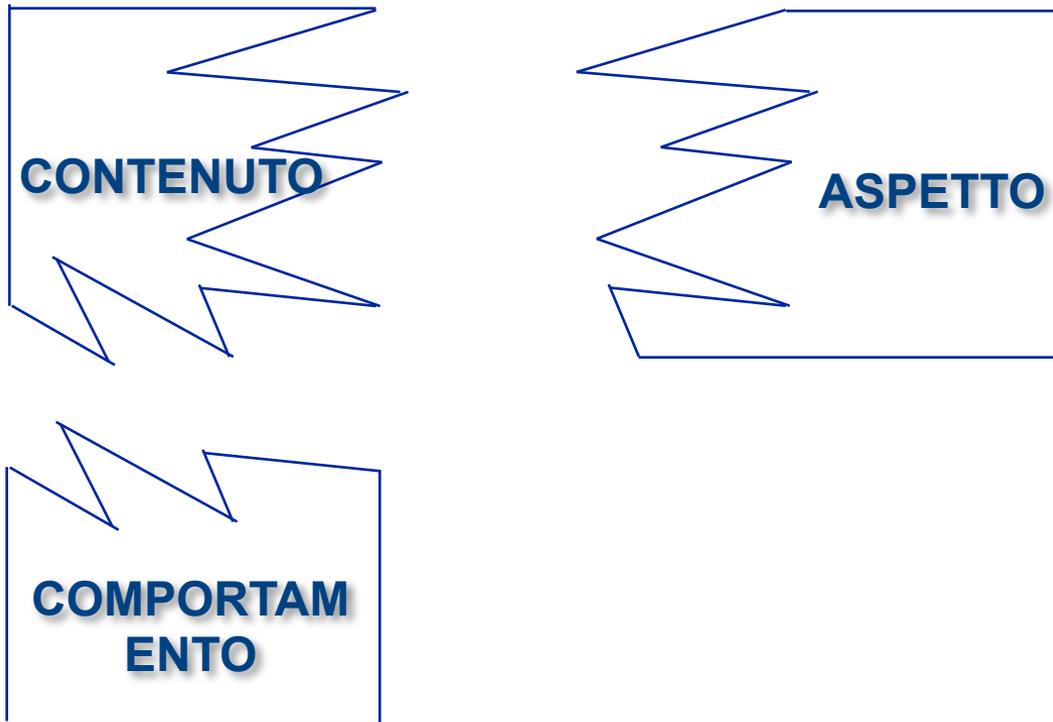
# JAVASCRIPT (XXIII)

COME TESTIMONIA L'ESEMPIO PRECEDENTE, E' POSSIBILE MESCOLORE IN UNA STESSA PAGINA (X)HTML CONTENUTO E COMPORAMENTO (QUEST'ULTIMO ESPRESSO TRAMITE SCRIPT)



# JAVASCRIPT (XXIV)

IN MOLTI CASI, CONTENUTO E COMPORTAMENTO VENGONO EFFETTIVAMENTE MESCOLATI NELLA STESSA PAGINA (X)HTML. IN ALTRI CASI (AD ESEMPIO, QUANDO UNO SCRIPT E' USATO IN PIU' PAGINE), E' OPPORTUNO SEPARARE CONTENUTO E COMPORTAMENTO



# JAVASCRIPT (XXV)

- Per separare aspetto e comportamento, occorre salvare gli script in un file separato e aggiungere in ciascuna pagina (X)HTML che usa tali script un collegamento a tale file (senza inserire gli script all'interno della pagina)
- Ad es., supponiamo che un insieme di script Javascript siano contenuti nel file “mieiScript.js”

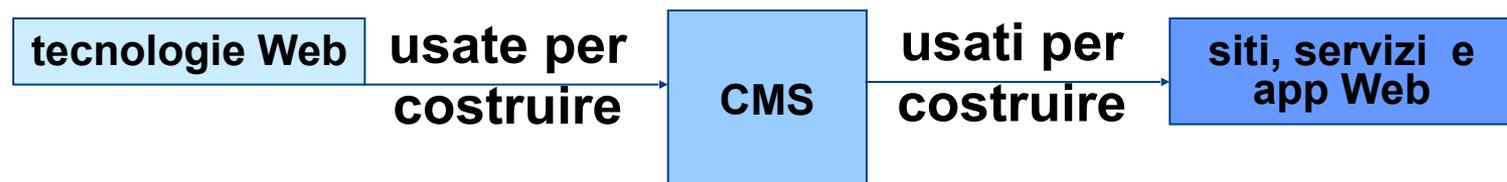
Ogni pagina che utilizza script contenuti nel file “mieiScript.js” deve contenere nella sezione head, un collegamento a questo file come il seguente:

```
<html>  
<head>  
<script type="text/javascript" src="mieiScript.js"></script>  
...  
</head>  
<body>  
...  
</body>  
</html>
```

# CONTENT MANAGEMENT SYSTEMS (I)

- Le tecnologie Web descritte sopra consentono la realizzazione di siti e, in generale, di applicazioni Web anche piuttosto complesse...
- Vi sono molte tipologie di applicazioni Web: siti classici, blog, piattaforme di social network, siti di e-commerce, ecc. ecc. ecc.
- ...fra le applicazioni Web realizzate (principalmente) con le tecnologie descritte sopra, menzioniamo qui i

## CMS (Content Management Systems)



# CONTENT MANAGEMENT SYSTEMS (II)

**Che cos'è un CMS (Content Management System)?**



**Un CMS è un software che supporta e facilita la gestione collaborativa di un sito web**

**At MS è possibile costruire e aggiornare un sito web anche di notevoli dimensioni, senza necessariamente conoscere il linguaggio HTML o altri linguaggi di programmazione web e senza dover progettare autonomamente un apposito database per la gestione delle informazioni dinamiche**

**Nota:** la nozione di CMS può avere un'accezione più ampia e riferirsi non solo a siti web, ma alla gestione di contenuti (es. documenti) in genere; qui consideriamo solo i CMS come sistemi di supporto alla gestione dei contenuti sul web

# CONTENT MANAGEMENT SYSTEMS (III)

- Quando un'azienda decide di (farsi) costruire un sito web, spesso si dimentica dell'importanza della manutenzione del sito e, in particolare, dell'importanza dell'aggiornamento dei contenuti → un CMS facilita soprattutto la manutenzione e l'aggiornamento del sito
- È molto importante separare gli eventuali interventi sugli aspetti grafici e strutturali del sito dall'aggiornamento dei contenuti: le due cose hanno cadenze temporali diverse e sono fatte da persone diverse

⇒ Un CMS consente

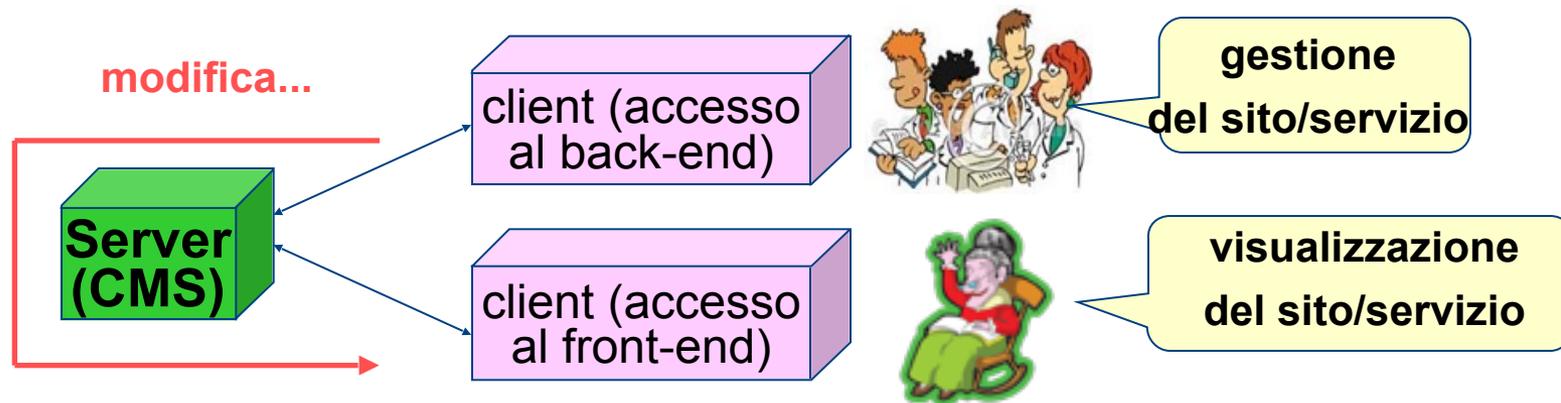
- al *web master* di costruire (ed eventualmente modificare) il *template* del sito
- agli *autori* di inserire (ed aggiornare) i **contenuti**

# CONTENT MANAGEMENT SYSTEMS (IV)

I CMS possono essere costruiti con tecnologie diverse, ma spesso utilizzano (anche) linguaggi di scripting server-side (PHP, JSP,...) e interagiscono con un database

⇒ un CMS è un'applicazione web che si installa sul server che ospita il sito web e si suddivide in due parti:

- **back-end** = sezione da cui si gestisce il sito; viene utilizzata dal *web master* per controllare e modificare l'**aspetto** e le **funzionalità** del sito e dagli *autori* per gestire i **contenuti**
- **front-end** = la parte visibile agli utenti finali (il sito web)

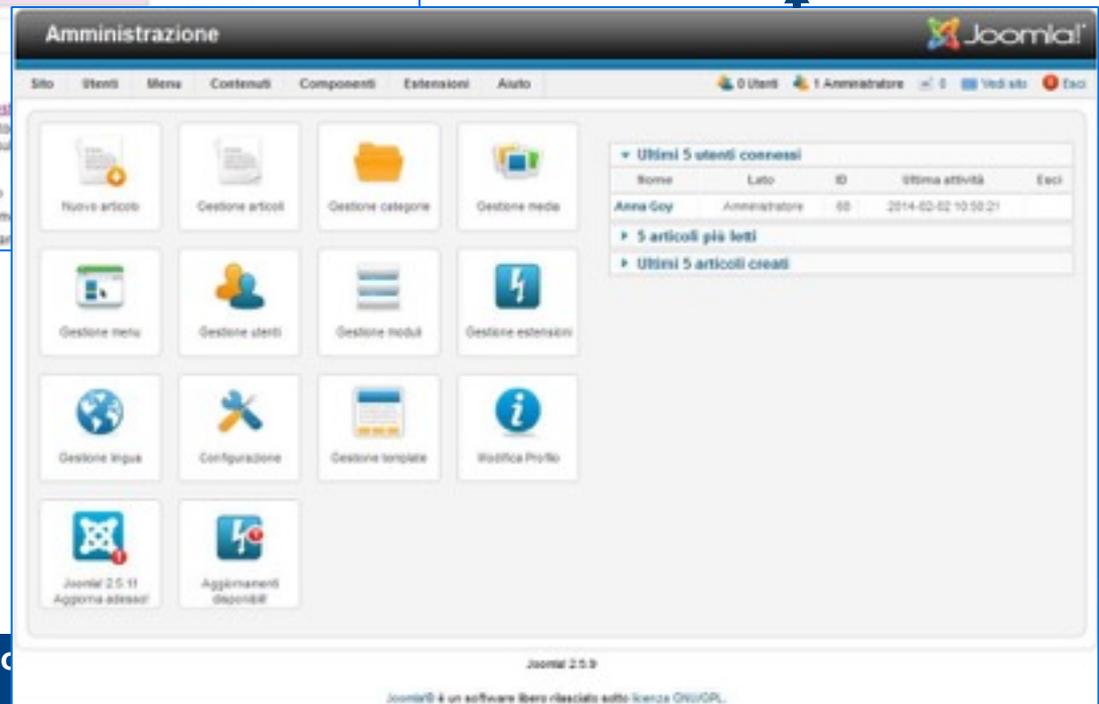


# CONTENT MANAGEMENT SYSTEMS (V)



**Frontend**  
(sito/servizio)

**Backend**  
(strumenti di  
amministrazione e  
gestione contenuti)



Tecnologia

Joomla! 2.5.9

Joomla! è un software libero rilasciato sotto licenza GNU/GPL.

Informatica

# CONTENT MANAGEMENT SYSTEMS (VI)

**Principali vantaggi dell'uso di un CMS** per la creazione e gestione di un sito/servizio web:

- possibilità di costruire e aggiornare un sito/servizio/app web **senza conoscere i linguaggi di programmazione web** (anche se questo è parzialmente falso... a volte è necessario capire ciò che un CMS ha prodotto...)
- **manutenzione e aggiornamento** del sito/servizio facilitati
- separazione di codice di programmazione, grafica/layout (*template*) e contenuti
  - ⇒ possibilità di applicare lo stesso *template* a contenuti differenti (es: il layout viene automaticamente rispettato quando si inseriscono nuovi contenuti)
  - ⇒ possibilità di utilizzare *template* differenti per un unico contenuto

# CONTENT MANAGEMENT SYSTEMS (VII)

- **disponibilità di funzionalità aggiuntive:** oltre agli strumenti per l'inserimento dei contenuti, i CMS includono spesso dei componenti (moduli, estensioni, plugin) aggiuntivi utili per inserire nel sito/servizio gallerie fotografiche, forum di discussione, sondaggi, funzionalità di e-commerce, sistemi di criptazione dei dati, ecc...
- **definizione di diversi profili di utenti, con diversi privilegi** (es: possibilità di modifica dei contenuti, del template, della configurazione del CMS stesso, ecc...)
- **gestione multi-utente e redazione distribuita** (in particolare, essendo applicazioni web-based, l'interazione con il backend può generalmente avvenire attraverso qualsiasi browser web: ciò consente agli utenti, anche se dislocati in sedi lontane tra loro, di aggiornare i contenuti e gestire il sito/servizio avendo a disposizione semplicemente un collegamento ad Internet)

# CONTENT MANAGEMENT SYSTEMS (VIII)

## Principali limiti dei CMS:

- I CMS consentono generalmente di inserire i contenuti in sezioni prestabilite dal *template*, collocate in zone fisse della pagina e con layout preimpostati: se un utente però vuole **personalizzare** queste impostazioni, posizionando i contenuti a proprio piacimento o modificando le funzionalità di alcune componenti, dovrà **modificare** direttamente il **codice sorgente del sistema** stesso
- I CMS possono essere **applicazioni molto complesse** e il loro utilizzo non è sempre così semplice!

# CONTENT MANAGEMENT SYSTEMS (IX)

## Spezziamo una lancia a favore dei CMS open source...

- offrono l'opportunità ad utenti e aziende di essere presenti online con bassi investimenti (sono generalmente scaricabili gratuitamente dal web)
- possono essere personalizzati con l'aggiunta di componenti particolari, gratuiti o a pagamento
- avendo a disposizione il codice sorgente, gli utenti più esperti possono liberamente modificare il software in base alle proprie esigenze
- in caso di problemi è possibile contare sulle community di sviluppatori, in grado di fornire supporto generalmente costante e tempestivo (tipicamente, più il sistema è diffuso, più è ampia la comunità di sviluppatori)

# CONTENT MANAGEMENT SYSTEMS (X)

## Esempi - I

### Alcuni esempi di CMS open source

#### Wordpress [it.wordpress.org]

Wordpress nasce come piattaforma per la gestione di blog.

E' scritto in PHP ed attualmente è forse il CMS più popolare ed utilizzato; ha una ricca documentazione online e la community è una delle più attive.

Per gli utenti che non hanno familiarità con HTML o altri linguaggi di programmazione web, Wordpress fornisce un editor WYSIWYG (*What You See Is What You Get*) piuttosto semplice da usare; anche il pannello di amministrazione (*backend*) è semplice ed intuitivo; supporta anche la gestione di contenuti multimediali ed offre un'ampia scelta di *template* e componenti aggiuntivi

# CONTENT MANAGEMENT SYSTEMS (X)

## Esempi - II

### Joomla! [[www.joomla.it](http://www.joomla.it)]

scritto in PHP, distribuito gratuitamente sotto licenza GNU/GPL, è un CMS molto noto e molto usato; supporta la gestione di molti aspetti di un sito web (aggiunta e modifica di contenuti, creazione di gallerie di immagini, realizzazione di sistemi di prenotazioni o vendita on-line) ed esiste una comunità mondiale molto attiva (utilizzatori, designers, traduttori e sostenitori) suddivisa in comunità nazionali, che forniscono supporto e documentazione agli utilizzatori del CMS.

Ci sono moltissimi *template* e componenti aggiuntivi (estensioni) disponibili (sia gratuiti sia a pagamento)

# CONTENT MANAGEMENT SYSTEMS (X)

## Esempi - III

### **Drupal [drupal.org]**

è un CMS caratterizzato da una comunità molto ampia e attiva; è dotato di moltissimi moduli opzionali che aggiungono molte caratteristiche interessanti come forum, blog, profili, ecc... ed è particolarmente indicato per siti con caratteristiche Web 2.0 e *social*.

Una delle caratteristiche più popolari di Drupal è il modulo *Taxonomy*, che permette di strutturare tutti i tipi di contenuti in categorie a livelli e tipologie multiple

### **Plone [plone.org]**

è un CMS molto utilizzato soprattutto nella Pubblica Amministrazione; è supportato da una community tra le più grandi al mondo

# CONTENT MANAGEMENT SYSTEMS (X)

## Esempi - IV

### OpenCMS [[www.opencms.org](http://www.opencms.org)]

è basato su tecnologie Java e XML, è compatibile sia con ambienti open source (Linux, Apache, Tomcat, MySQL o PostgreSQL) sia con ambienti proprietari (Windows, IIS, Oracle).

Comprende una sezione per la creazione di *template* e rende disponibili, oltre alle funzionalità di base, moduli ufficiali aggiuntivi sia rilasciati dagli stessi sviluppatori del prodotto sia da sviluppatori terzi (gestione articoli, funzionalità di ricerca estese, form con invio di email)

Tra le funzionalità base offre un editor WYSIWYG per la pubblicazione di pagine anche con layout complessi (senza necessità di conoscenza del linguaggio HTML), un motore di *workflow* (gestione di processi) con notifiche via e-mail e un sistema di gestione delle versioni per tenere traccia delle modifiche effettuate

# CONTENT MANAGEMENT SYSTEMS (X)

## Esempi - V

### **XOOPS** [[xoops.org](http://xoops.org)]

acronimo di *eXtensible Object Oriented Portal System*, è scritto in PHP e distribuito sotto licenza GNU/GPL; ha, tra i suoi maggiori punti di forza, oltre alla semplicità dell'interfaccia utente, la presenza di una struttura fortemente modulare (si possono installare, attivare o disattivare componenti in modo molto semplice, tramite il pannello di amministrazione); è altamente personalizzabile

### **Concrete5** [[www.concrete5.it](http://www.concrete5.it)]

è considerato un CMS ideale per chi non possiede ampie competenze informatiche: l'interfaccia di amministrazione user-friendly consente di aggiungere, modificare ed eliminare pagine dal proprio sito web in maniera molto semplice

# CONTENT MANAGEMENT SYSTEMS (XI)

## Come Scegliere? – I

Se un'organizzazione decide di utilizzare un CMS per costruire ad aggiornare il proprio sito, come fa a scegliere?

Esistono sostanzialmente 3 opzioni:

- realizzare (o commissionare) *ex-novo* un **CMS su misura**
  - vantaggi: probabilmente ben rispondente alle esigenze dell'organizzazione
  - svantaggi: alti costi di sviluppo; tempi di sviluppo lunghi
- acquistare un **CMS proprietario**
  - vantaggi: collaudato, tempi di sviluppo brevi, assistenza
  - svantaggi: poco personalizzabile, costi alti
- scegliere un **CMS open source**
  - vantaggi: costi bassi, molto personalizzabile, supporto dalla comunità di utenti e sviluppatori
  - svantaggi: necessità di competenze tecniche per la personalizzazione e la manutenzione

# CONTENT MANAGEMENT SYSTEMS (XI)

## Come Scegliere? – II

**Alcuni ulteriori parametri significativi che possono orientare la scelta del CMS:**

- **Costo**

- **Tipo di licenza**

- **Requisiti di sistema**

caratteristiche del software necessarie al suo funzionamento; dipendono, per es, dalla tecnologia su cui si basa il CMS: linguaggio (PHP, Java, ...), tipo di DBMS, ecc...

- **Sicurezza**

tra le principali problematiche dei CMS (in particolare quelli open source) c'è il rischio di una alta vulnerabilità (malintenzionati possono prendere il controllo del sito, modificandone i contenuti, creando e rimuovendo utenti, ecc...); una delle prime regole di sicurezza, è quella di aggiornare frequentemente il CMS con le nuove versioni

# CONTENT MANAGEMENT SYSTEMS (XI)

## Come Scegliere? – III

- **Supporto**

la presenza di una community attiva consente di interagire con gli sviluppatori in caso di problemi (bug o errori particolari del sistema), ma anche per esigenze particolari

- **Facilità d'uso**

dell'interfaccia utente, delle modalità di inserimento dei contenuti e della personalizzazione del template

- **Disponibilità di applicazioni integrate e funzionalità aggiuntive**

- **Flessibilità (personalizzabilità)**

della gestione dei contenuti, dell'amministrazione e della configurazione del CMS

- **Interoperabilità**

possibilità di interagire con altri sistemi software (es. supporto alla generazione di contenuti in formato RSS)