

## Capitolo 9

### Applicazioni Internet

Copyright © 2004 – The McGraw-Hill Companies srl

## Introduzione

- **Internet: Concetti di base**
- Formati di dati per il Web
  - HTML, XML, DTD
- Introduzione alle architetture a tre livelli
- Il livello di presentazione
  - Moduli HTML: GET e POST HTTP, codifica di URL; Javascript; fogli di stile. XSLT
- Il livello intermedio
  - CGI, application server, servlets, JavaServerPages, passaggio di argomenti, mantenimento dello stato (cookie)

Copyright © 2004 – The McGraw-Hill Companies srl

## Uniform Resource Identifiers

- Uniformano lo schema dei nomi per identificare le risorse su Internet
- Una risorsa può essere qualunque cosa:
  - Index.html
  - Canzone.mp3
  - Immagine.jpg
- URI di esempio:
  - <http://www.cs.wisc.edu/~dbbokk/index.html>
  - <mailto:webmaster@bookstore.com>

Copyright © 2004 – The McGraw-Hill Companies srl

## Struttura degli URI

<http://www.cs.wisc.edu/~dbbokk/index.html>

- Un URI ha tre parti:
  - Schema del nome (http)
  - Nome del computer host ([www.cs.wisc.edu](http://www.cs.wisc.edu))
  - Nome della risorsa ([~dbbokk/index.html](http://www.cs.wisc.edu/~dbbokk/index.html))
- Le URL sono un sottoinsieme degli URI

Copyright © 2004 – The McGraw-Hill Companies srl

## Hypertext Transfer Protocol

- Cos'è un protocollo di comunicazione?
  - Insieme di standard che definisce la struttura dei messaggi
  - Esempi: TCP, IP, HTTP
- Che succede se fate click su <http://www.cs.wisc.edu/~dbbokk/index.html?>
  - Il client (browser web) manda una richiesta HTTP al server
  - Il server riceve la richiesta e risponde
  - Il client riceve la risposta; invia altre richieste

Copyright © 2004 – The McGraw-Hill Companies srl

## HTTP (segue)

dal client al server :

```
GET ~/index.html HTTP/1.1
User-agent: Mozilla/4.0
Accept: text/html, image/gif,
image/jpeg
```

il server risponde :

```
HTTP/1.1 200 OK
Date: Mon, 04 Mar 2002 12:00:00
GMT
Server: Apache/1.3.0 (Linux)
Last-Modified: Mon, 01 Mar 2002
09:23:24 GMT
Content-Length: 1024
Content-Type: text/html
<HTML> <HEAD></HEAD>
<BODY>
<h1>Libreria Internet di Barns e
Noble </h1>
Il nostro catalogo :
<h3>Scienza</h3>
<b>Natura della legge fisica </b>
...
```

Copyright © 2004 – The McGraw-Hill Companies srl

## Struttura del protocollo HTTP

- Richieste HTTP
- Linea di richiesta: GET ~/index.html HTTP/1.1
  - GET: campo del metodo HTTP (valori possibili sono GET e POST, più avanti)
  - ~/index.html: campo URI
  - HTTP/1.1: campo della versione HTML
- Tipo di client: User-agent: Mozilla/4.0
- Che tipi di documenti verranno accettati dal client:  
Accept: text/html, image/gif, image/jpeg

Copyright © 2004 – The McGraw-Hill Companies srl

## Struttura del protocollo HTTP (segue)

### Risposte HTTP

- Linea di stato: HTTP/1.1 200 OK
  - Versione HTTP: HTTP/1.1
  - Codice di stato: 200
  - Messaggio del server: OK
  - Combinazioni comuni di codice di stato/messaggio del server:
    - 200 OK: la richiesta ha avuto successo
    - 400 Bad Request: il server non ha potuto soddisfare la richiesta
    - 404 Not Found: l'oggetto richiesto non esiste sul server
    - 505 HTTP Version not Supported
- Data di creazione dell'oggetto:
- Last-Modified: Mon, 01 Mar 2002 09:23:24 GMT
- Numero di bytes spediti: Content-Length: 1024
- Tipo di oggetto che viene spedito: Content-Type: text/html
- Altre informazioni quali il tipo di server, l'ora del server, etc.

Copyright © 2004 – The McGraw-Hill Companies srl

## Argomenti del capitolo

- Internet: Concetti di base
- **Formati di dati per il Web**
  - HTML, XML, DTD
- Introduzione alle architetture a tre livelli
- Il livello di presentazione
  - Moduli HTML: GET e POST HTTP, codifica di URL; Javascript; fogli di stile. XSLT
- Il livello intermedio
  - CGI, application server, servlets, JavaServerPages, passaggio di argomenti, mantenimento dello stato (cookie)

Copyright © 2004 – The McGraw-Hill Companies srl

## Formati di dati per Web

- HTML
  - Il linguaggio di presentazione per Internet
- XML
  - Un modello di dati gerarchico auto-descrittivo
- DTD
  - Schemi standardizzati per XML
- XSLT (non trattato nel libro)

Copyright © 2004 – The McGraw-Hill Companies srl

## HTML: un esempio

```
<HTML>
<HEAD></HEAD>
<BODY>
<h1>Libreria Internet Barns &
Nobble </h1>
Il nostro inventario :
<h3>Scienza</h3>
<b>Natura della legge fisica
</b>
<UL>
<LI>Autore: Richard
Feynman</LI>
<LI>Pubblicato nel 1980</LI>
<LI>Copertina dura</LI>
</UL>
<h3>Fiction</h3>
<b>Aspettando il
Mahatma</b>
<UL>
<LI>Autore: R.K.
Narayan</LI>
<LI>Pubblicato nel 1981</LI>
</UL>
<b>L'insegnante di
Inglese</b>
<UL>
<LI>Autore: R.K.
Narayan</LI>
<LI>Pubblicato nel 1980</LI>
<LI>Tascabile</LI>
</UL>
</BODY>
</HTML>
```

Copyright © 2004 – The McGraw-Hill Companies srl

## HTML: breve introduzione

- L'HTML è un linguaggio di marcatura
- I comandi sono tag
  - Tag di inizio e di fine
  - Esempi
    - <HTML>...</HTML>
    - <UL>...</UL>
- Molti editor generano automaticamente l'HTML direttamente dal documento (ad esempio Microsoft Word ha una funzione "Salva come HTML")

Copyright © 2004 – The McGraw-Hill Companies srl

## HTML: esempio di comandi

- <HTML>
- <UL>: lista non ordinata
- <LI>: elemento di una lista
- <h1>: intestazione più grande
- <h2>: intestazione di secondo livello, analogamente <h3>, <h4>
- <B>Title</B>: grassetto

Copyright © 2004 – The McGraw-Hill Companies srl

## XML: un esempio

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<LISTALIBRI>
  <LIBRO GENERE="Scienza" FORMATO="Copertina dura">
    <AUTORE>
      <NOME>Richard</NOME>
      <COGNOME>Feynman</COGNOME>
    </AUTORE>
    <TITOLO>Natura della legge fisica</TITOLO>
    <PUBBLICATO>1980</PUBBLICATO>
  </LIBRO>
  <LIBRO GENERE="Fiction">
    <AUTORE>
      <NOME>R. K.</NOME>
      <COGNOME>Narayan</COGNOME>
    </AUTORE>
    <TITOLO>Aspettando il Mahatma</TITOLO>
    <PUBBLICATO>1981</PUBBLICATO>
  </LIBRO>
  <LIBRO GENERE="Fiction">
    <AUTORE>
      <NOME>R. K.</NOME>
      <COGNOME>Narayan</COGNOME>
    </AUTORE>
    <TITOLO>Insegnante di Inglese</TITOLO>
    <PUBBLICATO>1980</PUBBLICATO>
  </LIBRO>
</LISTALIBRI>
```

Copyright © 2004 – The McGraw-Hill Companies srl

## XML: eXtensible Markup Language

- Language
  - Un modo di comunicare informazione
- Markup
  - Note o meta-dati che descrivono i dati o il linguaggio
- Extensible
  - Capacità illimitata di definire nuovi linguaggi o insiemi di dati

Copyright © 2004 – The McGraw-Hill Companies srl

## XML – Qual è il punto?

- Si possono includere i propri dati e una descrizione di ciò che tali dati rappresentano
  - Utile per definire il proprio linguaggio o protocollo personale
- Esempio: Chemical Markup Language

```
<molecola>
  <peso>234.5</peso>
  <Spettro>...</Spettro>
  <Numeri>...</Numeri>
</molecola>
```
- Obiettivi del progetto XML:
  - L'XML dovrebbe essere compatibile con SGML
  - La scrittura di programmi che elaborano documenti XML dovrebbe essere un compito semplice
  - Il progetto dovrebbe essere formale e preciso

Copyright © 2004 – The McGraw-Hill Companies srl

## XML – Struttura

- XML: punto di incontro di SGML e HTML
- L'XML somiglia all'HTML
- L'XML è una gerarchia di tag definiti dall'utente chiamati elementi con attributi e dati
- I dati sono descritti dagli elementi, gli elementi sono descritti dagli attributi



Copyright © 2004 – The McGraw-Hill Companies srl

## XML – Attributi



- L'XML è sensibile alle maiuscole e agli spazi
- I nomi dei tag di apertura e chiusura devono essere identici
- Tag di apertura: "<" + nome elemento + ">"
- Tag di chiusura: "</" + nome elemento + ">"
- Elementi vuoti non hanno dati e non hanno tag di chiusura:
  - cominciano con un "<" e finiscono con un "/>"
- <LIBRO/>

Copyright © 2004 – The McGraw-Hill Companies srl

## XML – Attributi

`<LIBRO genere="Scienza" formato="Copertina dura">...</LIBRO>`

↑ tag di apertura nome di elemento    ↑ attributo    ↑ valore dell'attributo    ↑ dati    ↑ tag di chiusura

- Gli attributi forniscono informazioni aggiuntive sui tag elementi
- Ci possono essere zero o più attributi in ogni elemento; ciascuno ha la forma
  - Nome\_attributo='valore\_attributo'
  - Non ci sono spazi tra il nome e "="
  - I valori degli attributi devono essere racchiusi dai caratteri ' oppure "
- Attributi multipli sono separati da spazi bianchi (uno o più spazi o tabulazioni)

Copyright © 2004 – The McGraw-Hill Companies srl

## Dati e commenti

`<LIBRO genere="Scienza" formato="Copertina dura">...</LIBRO>`

↑ tag di apertura nome di elemento    ↑ attributo    ↑ valore dell'attributo    ↑ dati    ↑ tag di chiusura

- I dati XML sono qualunque informazione tra un tag di apertura e un tag di chiusura
- I dati XML non devono contenere i caratteri '<' oppure '>'
- Commenti:
  - <!-- commento -->

Copyright © 2004 – The McGraw-Hill Companies srl

## Annidamento e gerarchia

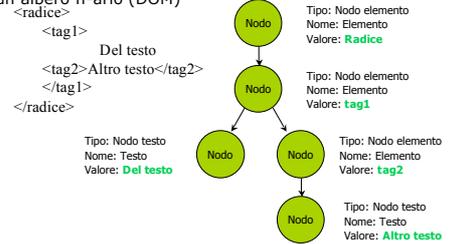
- I tag XML possono essere annidati in una gerarchia ad albero
  - I documenti XML possono avere un solo tag radice
  - Tra un tag di apertura e un tag di chiusura si possono inserire:
    - Dati
    - altri elementi
    - una combinazione di dati ed elementi
- ```

<radice>
  <tag1>
    Del testo
  </tag1>
  <tag2>Dell'altro</tag2>
</radice>
    
```

Copyright © 2004 – The McGraw-Hill Companies srl

## Memorizzazione

- La memorizzazione viene effettuata proprio come in un albero n-ario (DOM)



Copyright © 2004 – The McGraw-Hill Companies srl

## DTD - Document Type Definition

- Un DTD è uno schema per i dati XML
- I protocolli e i linguaggi XMP possono essere standardizzati con file DTD
- Un DTD dice quali elementi e attributi sono obbligatori e quali opzionali
  - Definisce la struttura formale del linguaggio

Copyright © 2004 – The McGraw-Hill Companies srl

## DTD – Un esempio

```

<?xml version="1.0"?>
<!ELEMENT Cesto (Ciliegia+, (Mela | Arancia)*)>
<!ELEMENT Ciliegia EMPTY>
  <!ATTLIST Ciliegia sapore CDATA #REQUIRED>
<!ELEMENT Mela EMPTY>
  <!ATTLIST Mela colore CDATA #REQUIRED>
<!ELEMENT Arancia EMPTY>
  <!ATTLIST Arancia provenienza 'Florida'>
    
```

```

☺<Cesto>
  <Ciliegia sapore="buono"/>
  <Mela colore="rosso"/>
  <Mela colore="verde"/>
</Cesto>

☹<Cesto>
  <Mela/>
  <Ciliegia sapore="buono"/>
  <Arancia/>
</Cesto>
    
```

Copyright © 2004 – The McGraw-Hill Companies srl

## DTD – Ben formato e valido

```
<?xml version='1.0'?>
<!ELEMENT Cesto (Ciliegia+)>
  <!ELEMENT Ciliegia EMPTY>
  <!ATTLIST Ciliegia sapore CDATA #REQUIRED>
```

### Non ben formato

```
<Cesto>
  <Ciliegia
sapore=buono>
</Cesto>
```

### Ben formato ma non valido

```
<Lavoro>
  <Luogo>Casa</Luogo>
</Lavoro>
```

### Ben formato e valido

```
<Cesto>
  <Ciliegia sapore='buono' />
</Cesto>
```

Copyright © 2004 – The McGraw-Hill Companies srl

## XML e DTD

- Un numero sempre maggiore di DTD verrà sviluppato
  - MathML
  - Chemical Markup Language
- Permette rapidi scambi di dati con la stessa semantica
- Sono disponibili sofisticati linguaggi di interrogazione:
  - Xquery
  - Xpath

Copyright © 2004 – The McGraw-Hill Companies srl

## Argomenti del capitolo

- Internet: Concetti di base
- Formati di dati per il Web
  - HTML, XML, DTD
- **Introduzione alle architetture a tre livelli**
- Il livello di presentazione
  - Moduli HTML: GET e POST HTTP, codifica di URL; Javascript; fogli di stile. XSLT
- Il livello intermedio
  - CGI, application server, servlets, JavaServerPages, passaggio di argomenti, mantenimento dello stato (cookie)

Copyright © 2004 – The McGraw-Hill Companies srl

## Componenti dei sistemi "data-intensive"

Tre tipi separati di funzionalità:

- gestione dei dati
  - logica di applicazione
  - presentazione
- L'architettura del sistema determina se queste tre componenti risiedono su un singolo sistema (tier) oppure se sono distribuite su diversi tier

Copyright © 2004 – The McGraw-Hill Companies srl

## Architettura a livello singolo

Tutte le funzionalità sono combinate in un singolo tier, generalmente un mainframe

- Accesso utente tramite terminali non intelligenti

### Vantaggi :

- facilità di manutenzione e amministrazione

### Svantaggi :

- Oggi gli utenti si aspettano interfacce utente di tipo grafico
- Il calcolo centralizzato di tutte le interfacce grafiche è troppo costoso per un singolo sistema



Copyright © 2004 – The McGraw-Hill Companies srl

## Architetture client-server

Due possibili suddivisioni del lavoro:

- **thin client**
  - Il client implementa solo l'interfaccia utente grafica
  - Il server implementa la logica dell'applicazione e la gestione dei dati
- **thick client**
  - Il client implementa sia l'interfaccia grafica che la logica dell'applicazione
  - Il server implementa la gestione dei dati



Copyright © 2004 – The McGraw-Hill Companies srl

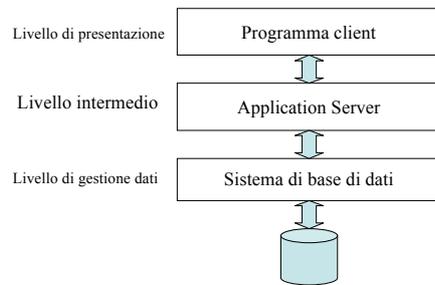
## Architetture client-server (segue)

### Svantaggi dei thick client

- Nessun luogo centralizzato per aggiornare la logica dell'applicazione
- Problemi di sicurezza: il server deve fidarsi dei client
  - Il controllo di accesso e l'autenticazione devono essere gestiti dal server
  - I client devono lasciare la base di dati del server in uno stato consistente
  - Una possibilità: incapsulare tutti gli accessi alla base di dati in stored procedure
- Non scalabile a più di un centinaio di client
  - Grossi trasferimenti di dati tra server e client
  - Più di un server crea un problema: x client, y server: x\*y connessioni

Copyright © 2004 – The McGraw-Hill Companies srl

## L'architettura a tre livelli



Copyright © 2004 – The McGraw-Hill Companies srl

## I tre livelli

### Livello di presentazione

- Interfaccia primaria con l'utente
- Deve adattarsi a diversi dispositivi di visualizzazione (PC, PDA, telefoni cellulari, accesso vocale?)

### Livello intermedio

- Implementa la logica dell'applicazione (implementa azioni complesse, mantiene lo stato tra diversi passi di un flusso di lavoro)
- Accede a diversi sistemi di gestione dei dati

### Livello di gestione dei dati

- Uno o più sistemi standard per la gestione di basi di dati

Copyright © 2004 – The McGraw-Hill Companies srl

## Esempio 1: prenotazioni aeree

- Costruire un sistema per prenotazioni aeree
- Cosa viene fatto dai vari livelli?
  - Sistema di basi di dati
    - Informazioni sulle aerolinee, posti disponibili, informazioni sui clienti, etc.
  - Application server
    - Logica per fare le prenotazioni, cancellare le prenotazioni, aggiungere nuove aerolinee, etc.
  - Programma client
    - Log in dei vari utenti, visualizzazione di moduli e output in forma leggibile

Copyright © 2004 – The McGraw-Hill Companies srl

## Esempio 2: iscrizione a corsi

- Costruire un sistema usando il quale degli studenti possono iscriversi a dei corsi
- Sistema di base di dati
  - Informazioni sugli studenti, informazioni sui corsi, informazioni sui docenti, disponibilità dei corsi, pre-requisiti, etc.
- Application server
  - Logica per modificare un corso, cancellare un corso, creare un nuovo corso, etc.
- Programma client
  - Login dei vari utenti (studenti, personale, professori), visualizzazione di moduli e output in forma leggibile

Copyright © 2004 – The McGraw-Hill Companies srl

## Tecnologie

Programma client (Browser web)	HTML Javascript XML/XSLT
Application Server (Tomcat, Apache)	JSP, ASP, PHP Servlets Cookies CGI
Sistema di basi di dati (DB2, MySQL, Oracle)	XML Stored Procedures

Copyright © 2004 – The McGraw-Hill Companies srl

## Vantaggi dell'architettura a tre livelli

- Sistemi eterogenei
- Thin client
- Accesso integrato ai dati
- Scalabilità
- Sviluppo software

Copyright © 2004 – The McGraw-Hill Companies srl

## Argomenti del capitolo

- Internet: Concetti di base
- Formati di dati per il Web
  - HTML, XML, DTD
- Introduzione alle architetture a tre livelli
- **Il livello di presentazione**
  - Moduli HTML: GET e POST HTTP, codifica di URL; Javascript; fogli di stile. XSLT
- Il livello intermedio
  - CGI, application server, servlets, JavaServerPages, passaggio di argomenti, mantenimento dello stato (cookie)

Copyright © 2004 – The McGraw-Hill Companies srl

## Introduzione al livello di presentazione

- Richiamo: funzionalità del livello di presentazione
  - Interfaccia primaria per l'utente
  - Deve adattarsi ai diversi dispositivi di visualizzazione (PC, PDA, telefono cellulare, accesso vocale?)
  - Funzionalità semplice, come il controllo della validità dei campi
- Ci occuperemo di:
  - moduli HTML: come passare dati al livello intermedio
  - JavaScript: funzionalità semplice al livello di presentazione
  - Fogli di stile: separare i dati dalla visualizzazione

Copyright © 2004 – The McGraw-Hill Companies srl

## Moduli HTML

- Modo diffuso per comunicare dati dal client al livello intermedio
- Formato generale di un modulo :

```
<FORM ACTION="pagina.jsp" METHOD="GET"
NAME="ModuloLogin">
...
</FORM>
```
- Componenti di un tag FORM HTML:
  - ACTION: specifica l'URI che gestisce il contenuto
  - METHOD: specifica il metodo HTML GET o POST
  - NAME: nome del modulo; può essere usato in script sul lato client per far riferimento al modulo

Copyright © 2004 – The McGraw-Hill Companies srl

## Dentro i moduli HTML

- Tag INPUT:
  - Attributi:
    - TYPE: text (campo per l'inserimento di testo), password (campo per l'inserimento di testo dove il testo immesso è visualizzato in maniera protetta, reset (ripristina tutti i campi del modulo))
    - NAME: nome simbolico, usato per identificare il valore del campo al livello intermedio
    - VALUE: valore predefinito
  - Esempio: `<INPUT TYPE="text" Name="titolo">`
- Modulo di esempio:

```
<form method="POST" action="Sommario.jsp">
<input type="text" name="userid">
<input type="password" name="password">
<input type="submit" value="Login" name="Invia">
<input type="reset" value="Reimposta">
</form>
```

Copyright © 2004 – The McGraw-Hill Companies srl

## Passaggio di argomenti

Due metodi: GET e POST

- GET
  - I contenuti del modulo vanno nell'URI specificato
  - Struttura:
    - azione?nome1=valore1&nome2=valore2&nome3=valore3
      - azione: nome dell'URI specificato nel modulo
      - le coppie (nome, valore) provengono dai campi INPUT del modulo; campi vuoti hanno valori vuoti ("nome=")
- esempio dal precedente modulo per l'immissione di una password:
  - `Sommario.jsp?userid=john&password=johnpw`
- Notate che la pagina chiamata azione deve essere un programma, uno script o una pagina che dovrà elaborare i dati inseriti dall'utente

Copyright © 2004 – The McGraw-Hill Companies srl

## Codifica dei campi del modulo

- I campi del modulo possono contenere caratteri ASCII generici che possono non apparire in un URI
- Una speciale convenzione di codifica converte tali valori in caratteri "compatibili con gli URI":
  - Converti tutti i caratteri "speciali" in %xyz, dove xyz è il codice ASCII del carattere. I caratteri speciali includono &, =, +, %, etc.
  - Converti tutti gli spazi nel carattere "+"
  - Incolla le coppie (nome, valore) dai tag INPUT del modulo tramite "&" per formare l'URI

Copyright © 2004 - The McGraw-Hill Companies srl

## Moduli HTML: un esempio completo

```
<form method="POST" action="Sommaro.jsp" >
  <table align = "center" border="0" width="300">
    <tr>
      <td>Userid</td>
      <td><input type="text" name="userid" size="20"></td>
    </tr>
    <tr>
      <td>Password</td>
      <td><input type="password" name="password"
        size="20"></td>
    </tr>
    <tr>
      <td align = "center"><input type="submit" value="Login"
        name="submit"></td>
    </tr>
  </table>
</form>
```

Copyright © 2004 - The McGraw-Hill Companies srl

## JavaScript

- Scopo: aggiungere funzionalità al livello di presentazione
- Applicazioni di esempio:
  - rilevare il tipo di browser e caricare una pagina specifica per quel browser
  - validazione di moduli: validare i campi di immissione testo del modulo
  - controllo del browser: aprire nuove finestre, chiudere finestre esistenti (esempio: finestre pop-up di pubblicità)
- Di solito incapsulato direttamente nell'HTML tramite il tag <SCRIPT>...</SCRIPT>
- <SCRIPT> ha diversi attributi:
  - LANGUAGE: specifica il linguaggio dello script (ad esempio javascript)
  - SRC: file esterno con il codice di script
  - Esempio:
- <SCRIPT LANGUAGE="JavaScript" SRC="validazione.js"></SCRIPT>

Copyright © 2004 - The McGraw-Hill Companies srl

## JavaScript (segue)

- Se il tag <SCRIPT> non ha un attributo SRC, allora il JavaScript è direttamente nel file HTML
- Esempio:

```
<SCRIPT LANGUAGE="JavaScript">
  <!--alert("Benvenuto nella nostra libreria")
  //-->
</SCRIPT>
```
- Due diversi stili di commento:
  - <!--commento per HTML, poiché il codice JavaScript che segue dovrebbe essere ignorato dall'elaboratore HTML
  - // commento per JavaScript allo scopo di chiudere il commento HTML

Copyright © 2004 - The McGraw-Hill Companies srl

## JavaScript (segue)

- JavaScript è un linguaggio di scripting completo
  - Variabili
  - Assegnazioni (=, +=, ...)
  - Operatori di confronto (<, >, ...) operatori booleani (&&, ||, !)
  - Comandi
    - If (condizione) {comandi;} else {comandi;}
    - Cicli for, cicli do-while e cicli while
  - Funzioni con restituzione di valori
    - Si creano funzioni usando la parola chiave function
    - F(arg1, ..., argk) {comandi;}

Copyright © 2004 - The McGraw-Hill Companies srl

## JavaScript: un esempio completo

### Modulo HTML :

```
<form method="POST"
  action="Sommaro.jsp"
  NAME="LoginForm">
  <input type="text"
    name="userid">
  <input type="password"
    name="password">
  <input type="submit"
    value="Login"
    name="submit">
  <input type="Reimposta"
    value="Clear">
</form>
```

### JavaScript associato :

```
<script language="javascript">
function Function
  controllaLoginVuoto ()
{
  loginForm = document.LoginForm
  if ((loginForm.userid.value == "" )
  ||
  (loginForm.password.value == ""))
  {
    Alert("Immettere un valore per
    userid e password") ;
    return false;
  }
  else return true;
}
</script>
```

Copyright © 2004 - The McGraw-Hill Companies srl

## Argomenti del capitolo

- Internet: Concetti di base
- Formati di dati per il Web
  - HTML, XML, DTD
- Introduzione alle architetture a tre livelli
- Il livello di presentazione
  - Moduli HTML: GET e POST HTTP, codifica di URL; Javascript; fogli di stile. XSLT
- **Il livello intermedio**
  - **CGI, application server, servlets, JavaServerPages, passaggio di argomenti**

Copyright © 2004 – The McGraw-Hill Companies srl

## Introduzione al livello intermedio

- Richiamo: funzionalità del livello intermedio
  - Codifica la logica dell'applicazione
  - Effettua le connessioni al/ai sistema/i di basi di dati
  - Riceve il testo immesso nei moduli al livello di presentazione
  - Genera i risultati per il livello di presentazione
- Ci occuperemo di
  - CGI: protocollo per il passaggio di argomenti ai programmi in esecuzione al livello intermedio
  - Application server: ambienti di esecuzione al livello intermedio
  - Servlet: programmi Java al livello intermedio
  - JavaServerPages: script Java al livello intermedio
  - Mantenimento dello stato: come mantenere lo stato al livello intermedio. Argomento principale: cookie

Copyright © 2004 – The McGraw-Hill Companies srl

## CGI: Common Gateway Interface

- Scopo: trasmettere argomenti dai moduli HTML ai programmi applicativi che vengono eseguiti al livello intermedio
- I dettagli del reale protocollo CGI non sono importanti - > le librerie implementano le interfacce ad alto livello
- Svantaggi:
  - Il programma dell'applicazione viene eseguito come un nuovo processo ad ogni invocazione (rimedio: FastCGI)
  - Non c'è condivisione di risorse tra i programmi applicativi (ad esempio connessioni a basi di dati)
  - Rimedio: application server

Copyright © 2004 – The McGraw-Hill Companies srl

## CGI: esempio

- Modulo HTML:

```
<FORM ACTION="trovaLibri.cgi" METHOD="POST">
Inserisci il nome di un autore:
<INPUT TYPE="text" NAME="nomeAutore">
<INPUT TYPE="submit" VALUE="Invia">
<INPUT TYPE="reset" VALUE="Cancella">
</FORM>
```
- Codice Perl:

```
use CGI;
$dataIn=new CGI;
$dataIn->header();
$authorName=$dataIn->param('nomeAutore');
print("<HTML><TITLE>Prova di passaggio di argomenti</TITLE>");
print("Il nome dell'autore è " + $authorName);
print("</HTML>");
exit;
```

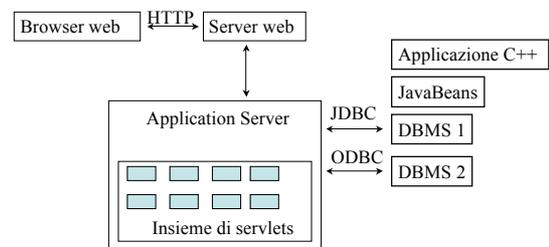
Copyright © 2004 – The McGraw-Hill Companies srl

## Application Server

- Idea: evitare il sovraccarico delle CGI
  - Insieme principale dei thread dei processi
  - Gestisce le connessioni
  - Consente l'accesso a sorgenti di dati eterogenee
  - Altre funzionalità quali API per la gestione delle sessioni

Copyright © 2004 – The McGraw-Hill Companies srl

## Application server: struttura dei processi



Copyright © 2004 – The McGraw-Hill Companies srl

## Servlet

- Java Servlets: codice Java che viene eseguito al livello intermedio
  - Indipendente dalla piattaforma
  - API Java completamente disponibile, incluso JDBC

Esempio :

```
import java.io.*;
import java.servlet.*;
import java.servlet.http.*;
public class ScheletroDiServlet extends HttpServlet {
    public void doGet(HttpServletRequest richiesta,
        HttpServletResponse risposta)
        throws ServletException, IOException {
        PrintWriter out=risposta.getWriter();
        out.println("Ciao mondo");
    }
}
```

Copyright © 2004 – The McGraw-Hill Companies srl

## Servlet (segue)

- Vita di un servlet?
  - Il server web inoltra la richiesta al contenitore del servlet
  - Il contenitore crea una istanza del servlet (chiama il metodo `init()`; al momento della deallocazione: chiama il metodo `destroy()`)
  - Il contenitore chiama il metodo `service()`
    - `Service()` chiama `doGet()` per il GET HTTP o il `doPost()` per il POST HTTP
    - Di solito `service()` non viene sovrascritto, ma vengono sovrascritti `doGet()` e `doPost()`

Copyright © 2004 – The McGraw-Hill Companies srl

## Servlet: un esempio completo

```
public class LeggiNomeUtente extends HttpServlet {
    public void doGet(
        HttpServletRequest richiesta,
        HttpServletResponse risposta)
        throws ServletException, IOException {
        risposta.setContentType("text/html");
        PrintWriter out=risposta.getWriter();
        out.println("<HTML><BODY>\n <UL> \n" +
            "<LI>" + richiesta.getParameter("userid") + "\n" +
            "<LI>" + richiesta.getParameter("password") + "\n" +
            "<UL>\n<BODY></HTML>");
    }
    public void doPost(
        HttpServletRequest richiesta,
        HttpServletResponse risposta)
        throws ServletException, IOException {
        doGet(richiesta, risposta);
    }
}
```

Copyright © 2004 – The McGraw-Hill Companies srl

## Java Server Pages

- Servlet
  - Generano HTML scrivendolo sull'oggetto "PrintWriter"
  - Prima il codice, poi la pagina web
- JavaServerPages
  - Codice scritto in HTML, simile al codice dei servlet, incapsulato nell'HTML
  - Prima la pagina web, poi il codice
  - Di solito sono compilate in un servlet

Copyright © 2004 – The McGraw-Hill Companies srl

## JavaServerPages: esempio

```
<html>
<head><title>Benvenuto alla B&N</title></head>
<body>
  <h1>Bentornato!</h1>
  <% String name="NuovoUtente";
     if (request.getParameter("nomeUtente") != null) {
       nome=request.getParameter("nomeUtente");
     }
  %>
  Sei connesso come <%=nome%>
<p>
</body>
</html>
```

Copyright © 2004 – The McGraw-Hill Companies srl