Fabio Burroni

Dipartimento di Ingegneria dell'Informazione

Università degli Studi di Siena

burronif@unisi.it

Sistemi e tecnologie per la multimedialità e telematica

Linguaggi per lo sviluppo di applicazioni e servizi.

La presentazione è scaricabile da http://www.ltt.dii.unisi.it/benelli.htm

I servizi delle reti

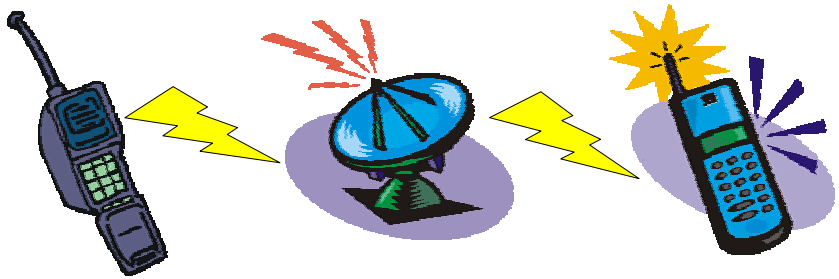
Un servizio è messo a disposizione dell'utente della rete affinché possa trarne un beneficio.

Alcuni esempi:

- Short Messages Services (SMS) sulla rete GSM
- Web browsing
- e-mail
- …

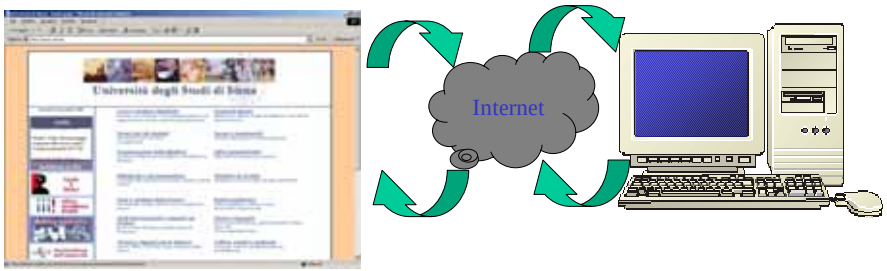
I servizi delle reti

Un servizio può essere fornito dal gestore della rete o da un qualunque altro ente in grado di realizzare una struttura capace di rispondere alle richieste di ogni utente.



I servizi delle reti

La maggior parte dei servizi sono dei **programmi software** realizzati ad hoc per soddisfare le esigenze degli utenti.

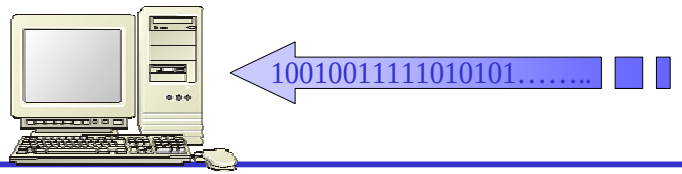


Cos'è un programma?

Un programma è un insieme di codici interpretabili dal processore di un computer che lo sta ospitando.

Tali codici prendono il nome di linguaggio macchina e sono una sequenza di bit.

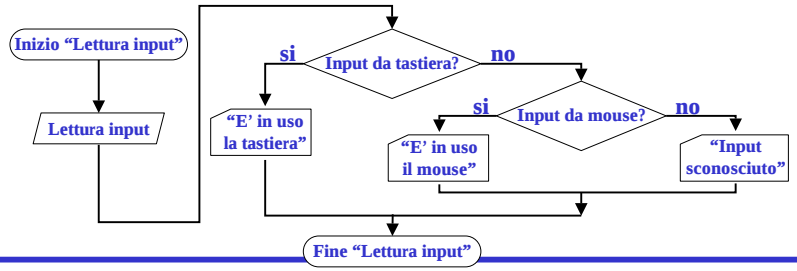
Il linguaggio macchina impartisce una sequenza di comandi ed azioni seguendo un flusso logico.



I diagrammi di flusso

Una volta studiato il problema, l'approccio alla progettazione avviene tramite la composizione del diagramma di flusso.

Il diagramma di flusso esplica tutte le azioni che il programma svolgerà in risposta a determinati eventi ricevuti in input.



Dal diagramma di flusso al programma

Dopo la redazione e la verifica del diagramma di flusso si passa alla fase di programmazione.

E' necessario scegliere tra vari **linguaggi** di programmazione, linguaggi di script e meta linguaggi.

Dal diagramma di flusso al programma

Dopo la redazione e la verifica del diagramma di flusso si passa alla fase di programmazione.

E' necessario scegliere tra vari **linguaggi** di programmazione, linguaggi di script e meta linguaggi.

Tutte queste tipologie offrono un approccio di alto livello astraendosi dal linguaggio macchina e mettendo a disposizione del programmatore una sintassi ed una semantica più vicina al linguaggio umano.

Alcuni Linguaggi

Java

Visual Basic

WML

C#

Java Script

HTML

ASP

PHP

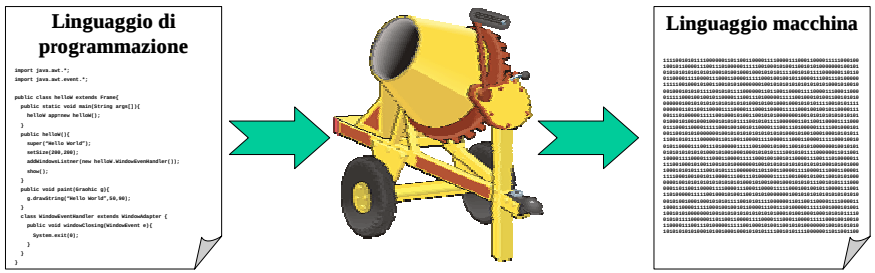
Delphi

XML

C++

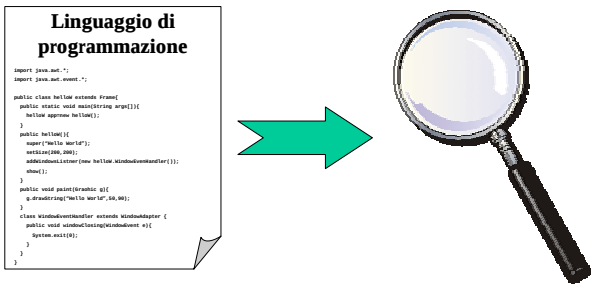
Il compilatore

Il compilatore è lo strumento in grado di trasformare il codice del linguaggio di programmazione in codice in linguaggio macchina.



L'interprete

L'interprete è lo strumento in grado di riconoscere la sintassi e la semantica del linguaggio di programmazione agendo direttamente sul codice sorgente.



La filosofia client-server

La filosofia client-server è basata su due figure principali: il **server** ed il **client**.

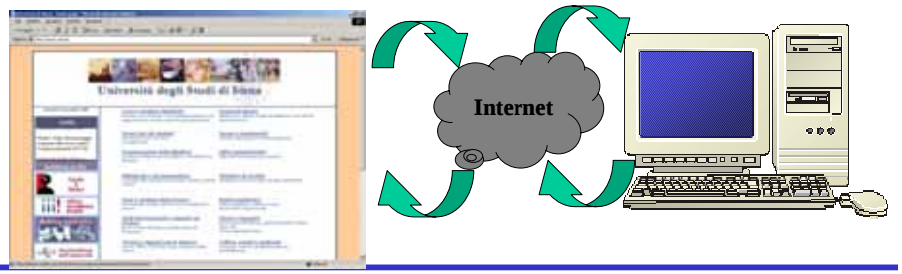
Il server offre l'intelligenza necessaria al servizio e di conseguenza risponde alle domande pervenute dai vari client.

Il client è un'entità nata per ricevere il servizio offerto dal server. Il servizio viene ricevuto dopo aver interpellato il server. Più client possono rivolgersi allo stesso server.



Il client-server HTTP

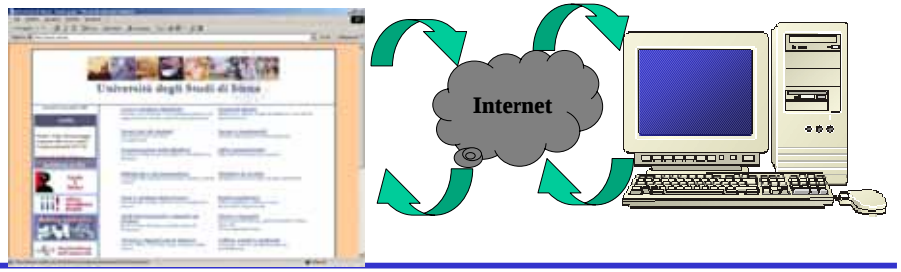
Il protocollo HTTP è basato sul modello client-server.



Il client-server HTTP

Il protocollo HTTP è basato sul modello client-server.

Il client (browser) invia la richiesta al server ed attende la risposta HTML.

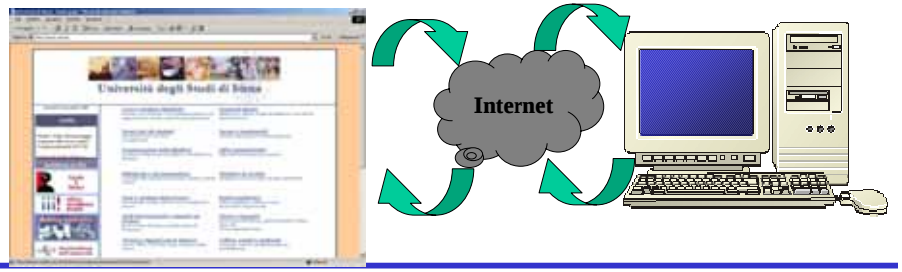


Il client-server HTTP

Il protocollo HTTP è basato sul modello client-server.

Il client (browser) invia la richiesta al server ed attende la risposta HTML.

Il web server risponde alle richieste pervenute dai client inviando informazioni HTML.



HTML

HTML è un meta linguaggio basato su TAG.

Un TAG è un elemento racchiuso tra parentesi angolari (“<”,”>”), con eventuali attributi ed ha un’apertura ed una chiusura.

Ogni TAG ha un significato particolare stabilito dall’ente W3C **http://www.w3c.org**.

Esistono varie tipologie di TAG che si occupano della formattazione del testo, delle immagini, delle tabelle, dei collegamenti ipertestuali, ecc.

La struttura delle pagine HTML

```
<HTML>
 <HEAD>
  <TITLE>Titolo pagina</TITLE>
 </HEAD>
 <BODY>
  <IMG SRC=“image.jpg”>
  <P ALIGN=“CENTER”>
   Hello World
  </P>
 </BODY>
</HTML>
```

La struttura delle pagine HTML

```
<HTML>
 <HEAD>
  <TITLE>Titolo pagina</TITLE>
 </HEAD>
 <BODY>
  <IMG SRC=“image.jpg”>
  <P ALIGN=“CENTER”>
   Hello World
  </P>
 </BODY>
</HTML>
```

La pagina HTML è racchiusa tra i TAG di apertura e di chiusura

La struttura delle pagine HTML

```
<HTML>
 <HEAD>
  <TITLE>Titolo pagina</TITLE>
 </HEAD>
 <BODY>
  <IMG SRC=“image.jpg”>
  <P ALIGN=“CENTER”>
   Hello World
  </P>
 </BODY>
</HTML>
```

Il TAG HEAD contiene il titolo della pagina

La struttura delle pagine HTML

```
<HTML>
 <HEAD>
  <TITLE>Titolo pagina</TITLE>
 </HEAD>
 <BODY>
  <IMG SRC="image.jpg">
  <P ALIGN="CENTER">
   Hello World
  </P>
 </BODY>
</HTML>
```

Il TAG BODY contiene l'effettiva informazione della pagina HTML

La struttura delle pagine HTML

```
<HTML>
 <HEAD>
  <TITLE>Titolo pagina</TITLE>
 </HEAD>
 <BODY>
  <IMG SRC="image.jpg">
  <P ALIGN="CENTER">
   Hello World
  </P>
 </BODY>
</HTML>
```

Per inserire le immagini occorre il TAG IMG

La struttura delle pagine HTML

```
<HTML>
 <HEAD>
  <TITLE>Titolo pagina</TITLE>
 </HEAD>
 <BODY>
  <IMG SRC="image.jpg">
  <P ALIGN="CENTER">
   Hello World
  </P>
 </BODY>
</HTML>
```

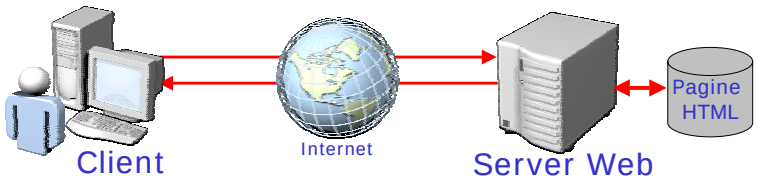
Per inserire un nuovo paragrafo occorre il TAG P

La struttura delle pagine HTML

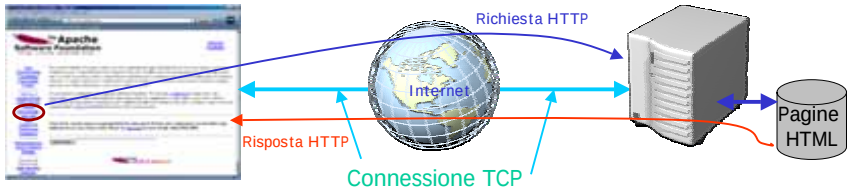


Funzionamento di un server web (statico)

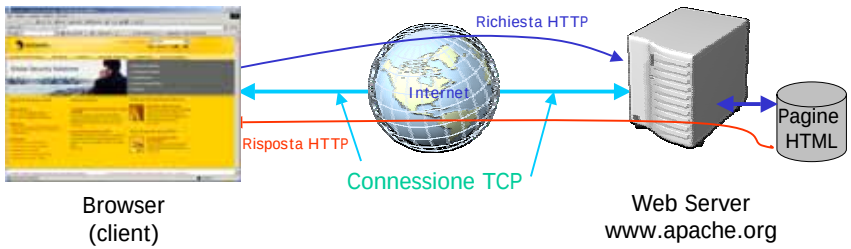
I primi servizi Web offrivano informazioni Statiche: uguali per tutti gli utenti.
La navigazione web, da un punto di vista tecnologico si basa su dei principi molto semplici.
L'utente usa un programma, denominato **Client**, per richiedere delle risorse (pagine), ad un altro programma, denominato **Server**, in esecuzione su una macchina remota.



Funzionamento di un server web (statico)



Funzionamento di un server web (statico)



WAP e il WML

Wireless Application Protocol (**WAP**) è un protocollo di comunicazione che rende possibile la navigazione su terminali mobili.

I dispositivi in grado di analizzare WAP sono dei modelli particolari che contengono al loro interno dei microbrowser per visualizzare le informazioni presenti in rete.

Le pagine visualizzabili sono scritte in WML, un metalinguaggio simile al HTML, ma adattato alle limitate capacità hardware del terminale.

WML

```
<WML>
 <CARD NAME="Card1">
  <DO TYPE="ACCEPT">
   <GO URL="#Card2"/>
  </DO
  Welcome!
 </CARD>
 <CARD NAME="Card2">
  <DO TYPE="ACCEPT">
    <GO URL="/submit?N=$(N)&S=$(S)"/>
  </DO>
  Enter name: <INPUT KEY="N"/>
  Choose speed:
  <SELECT KEY="S">
   <OPTION VALUE="0">Fast</OPTION>
   <OPTION VALUE="1">Slow</OPTION>
  <SELECT>
 </CARD>
</WML>
```

Deck

WML

```
<WML>
 <CARD NAME="Card1">
  <DO TYPE="ACCEPT">
   <GO URL="#Card2"/>
  </DO
  Welcome!
 </CARD>
 <CARD NAME="Card2">
  <DO TYPE="ACCEPT">
    <GO URL="/submit?N=$(N)&S=$(S)"/>
  </DO>
  Enter name: <INPUT KEY="N"/>
  Choose speed:
  <SELECT KEY="S">
   <OPTION VALUE="0">Fast</OPTION>
   <OPTION VALUE="1">Slow</OPTION>
  <SELECT>
 </CARD>
</WML>
```

Card

Deck

WML

Navigazione

```
<WML>
 <CARD NAME="Card1">
  <DO TYPE="ACCEPT">
   <GO URL="#Card2"/>
  </DO
  Welcome!
 </CARD>
 <CARD NAME="Card2">
  <DO TYPE="ACCEPT">
    <GO URL="/submit?N=$(N)&S=$(S)"/>
  </DO>
  Enter name: <INPUT KEY="N"/>
  Choose speed:
  <SELECT KEY="S">
   <OPTION VALUE="0">Fast</OPTION>
   <OPTION VALUE="1">Slow</OPTION>
  <SELECT>
 </CARD>
</WML>
```

Card

Deck

WML

Navigazione

Variabili

```
<WML>
 <CARD NAME="Card1">
  <DO TYPE="ACCEPT">
   <GO URL="#Card2"/>
  </DO
  Welcome!
 </CARD>
 <CARD NAME="Card2">
  <DO TYPE="ACCEPT">
    <GO URL="/submit?N=$(N)&S=$(S)"/>
  </DO>
  Enter name: <INPUT KEY="N"/>
  Choose speed:
  <SELECT KEY="S">
   <OPTION VALUE="0">Fast</OPTION>
   <OPTION VALUE="1">Slow</OPTION>
  <SELECT>
 </CARD>
</WML>
```

Card

Deck

WML

Navigazione

Variabili

Elementi di input

```
<WML>
 <CARD NAME="Card1">
  <DO TYPE="ACCEPT">
   <GO URL="#Card2"/>
  </DO
  Welcome!
 </CARD>
 <CARD NAME="Card2">
  <DO TYPE="ACCEPT">
   <GO URL="/submit?N=$(N)&S=$(S)"/>
  </DO>
  Enter name: <INPUT KEY="N"/>
  Choose speed:
  <SELECT KEY="S">
   <OPTION VALUE="0">Fast</OPTION>
   <OPTION VALUE="1">Slow</OPTION>
  <SELECT>
 </CARD>
</WML>
```

Card

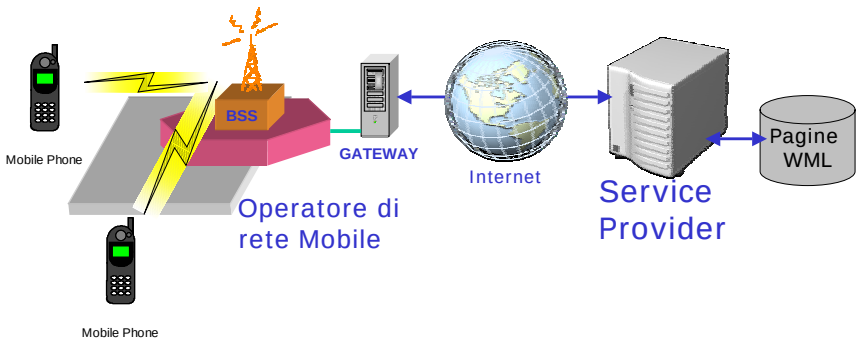
Deck

Un esempio

Il browser del terminale mobile:



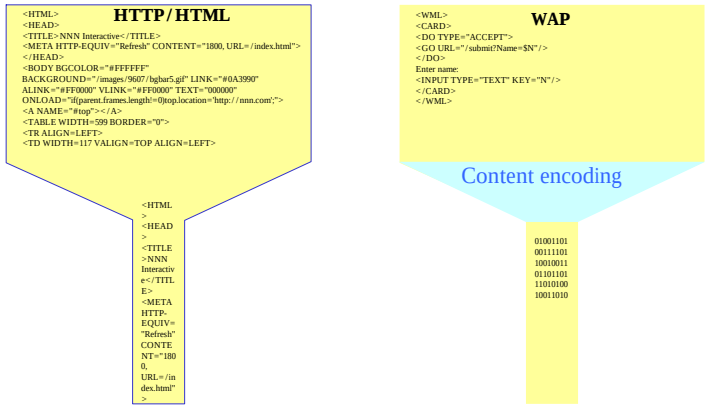
Architettura WAP



Il WAP Gateway effettua una codifica delle pagine WML per minimizzare l'occupazione di banda.

HTML vs WML

Differente utilizzo del canale:



WML, Elementi del linguaggio



In pratica, il Wml supporta molti elementi presenti nell'HTML.

La dinamicità delle pagine HTML: JavaScript

Per sua natura il codice HTML è statico. Non è possibile controllare le azioni dell'utente all'interno della pagina web.

Per ovviare a questo inconveniente è stato introdotto un particolare linguaggio di script lato client capace di controllare gli elementi e gli eventi ad essi associati all'interno della pagina HTML: JavaScript.

JavaScript

Si inserisce all'interno della pagina HTML con il tag SCRIPT oppure si connette alla pagina tramite l'attributo SRC, che specifica l'URL dal quale prelevare il codice.

<script language="JavaScript" src=http://www.unisi.it/file1.js**>**

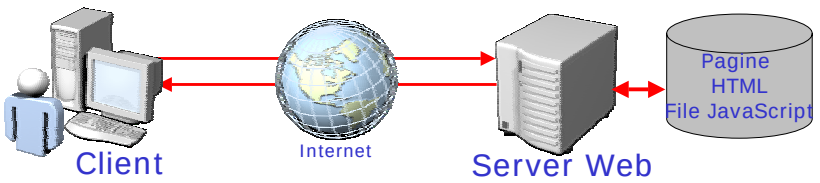
E' processato direttamente dal browser.

E' possibile definire delle funzioni e richiamarle in seguito al manifestarsi di determinati eventi.

Interagisce con tutti gli elementi attivi all'interno della pagina: form, input, button, ecc.

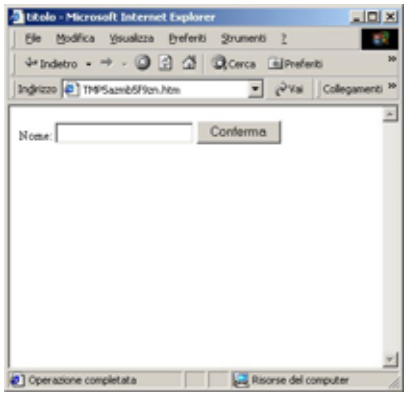
JavaScript

Il server web dovrà ospitare anche i file Java Script.



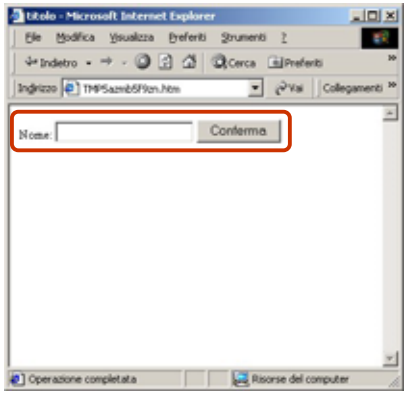
JavaScript nella pagina HTML

```
<html>
 <head>
  <title>titolo</title>
 </head>
 <body>
  <script language="JavaScript">
   function controlla(){
    if (frmDati.txtNome.value==""){
     alert("Inserire almeno un valore.");
     return false;
    }
    else{
     return true;
    }
   }
  </script>
  <form action="invio.asp" method="post“
   name="frmDati“ onSubmit=“return controlla();”>
   Nome:
   <input name="txtNome" type="text“
    size="20" maxlength="20">
   <input name="cmdConf" type=“submit“
    value="Conferma“>
  </form>
 </body>
</html>
```



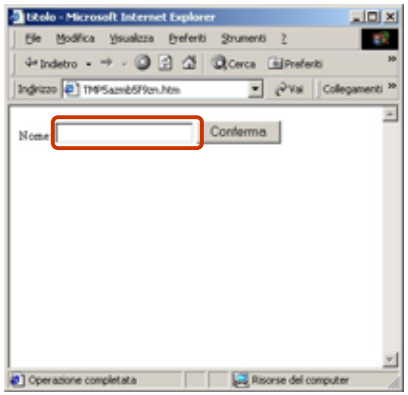
JavaScript nella pagina HTML

```
<html>
 <head>
  <title>titolo</title>
 </head>
 <body>
  <script language="JavaScript">
   function controlla(){
    if (frmDati.txtNome.value==""){
     alert("Inserire almeno un valore.");
     return false;
    }
    else{
     return true;
    }
   }
  </script>
  <form action="invio.asp" method="post“
   name="frmDati“ onSubmit=“return controlla();”>
   Nome:
   <input name="txtNome" type="text“
    size="20" maxlength="20">
   <input name="cmdConf" type=“submit“
    value="Conferma“>
  </form>
 </body>
</html>
```



JavaScript nella pagina HTML

```
<html>
 <head>
  <title>titolo</title>
 </head>
 <body>
  <script language="JavaScript">
   function controlla(){
    if (frmDati.txtNome.value==""){
     alert("Inserire almeno un valore.");
     return false;
    }
    else{
     return true;
    }
   }
  </script>
  <form action="invio.asp" method="post“
   name="frmDati“ onSubmit=“return controlla();”>
   Nome:
   <input name="txtNome" type="text“
    size="20" maxlength="20">
   <input name="cmdConf" type=“submit“
    value="Conferma“>
  </form>
 </body>
</html>
```



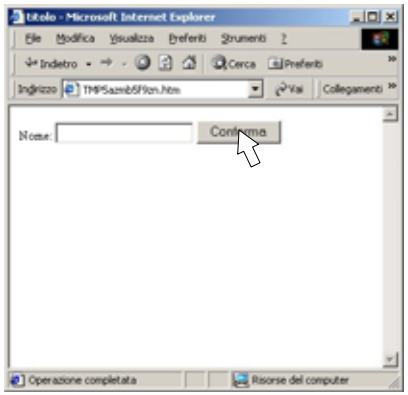
JavaScript nella pagina HTML

```
<html>
 <head>
  <title>titolo</title>
 </head>
 <body>
  <script language="JavaScript">
   function controlla(){
    if (frmDati.txtNome.value==""){
     alert("Inserire almeno un valore.");
     return false;
    }
    else{
     return true;
    }
   }
  </script>
  <form action="invio.asp" method="post“
   name="frmDati“ onSubmit=“return controlla();”>
   Nome:
   <input name="txtNome" type="text“
    size="20" maxlength="20">
   <input name="cmdConf" type=“submit“
    value="Conferma“>
  </form>
 </body>
</html>
```



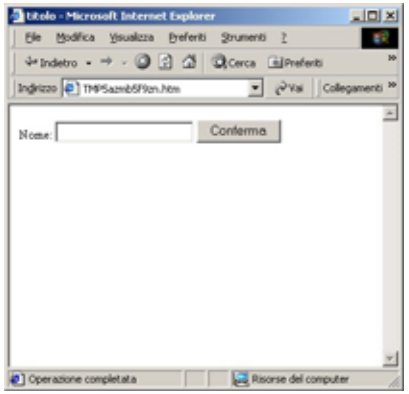
JavaScript nella pagina HTML

```
<html>
 <head>
  <title>titolo</title>
 </head>
 <body>
  <script language="JavaScript">
   function controlla(){
    if (frmDati.txtNome.value==""){
     alert("Inserire almeno un valore.");
     return false;
    }
    else{
     return true;
    }
   }
  </script>
  <form action="invio.asp" method="post“
   name="frmDati“ onSubmit=“return controlla();”>
   Nome:
   <input name="txtNome" type="text“
    size="20" maxlength="20">
   <input name="cmdConf" type=“submit“
    value="Conferma“>
  </form>
 </body>
</html>
```



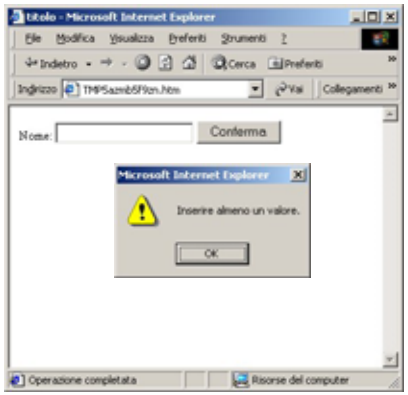
JavaScript nella pagina HTML

```
<html>
 <head>
  <title>titolo</title>
 </head>
 <body>
  <script language="JavaScript">
   function controlla(){
    if (frmDati.txtNome.value==""){
     alert("Inserire almeno un valore.");
     return false;
    }
    else{
     return true;
    }
   }
  </script>
  <form action="invio.asp" method="post“
   name="frmDati“ onSubmit=“return controlla();”>
   Nome:
   <input name="txtNome" type="text“
    size="20" maxlength="20">
   <input name="cmdConf" type=“submit“
    value="Conferma“>
  </form>
 </body>
</html>
```



JavaScript nella pagina HTML

```
<html>
 <head>
  <title>titolo</title>
 </head>
 <body>
  <script language="JavaScript">
   function controlla(){
    if (frmDati.txtNome.value==""){
     alert("Inserire almeno un valore.");
     return false;
    }
    else{
     return true;
    }
   }
  </script>
  <form action="invio.asp" method="post“
   name="frmDati“ onSubmit=“return controlla();”>
   Nome:
   <input name="txtNome" type="text“
    size="20" maxlength="20">
   <input name="cmdConf" type=“submit“
    value="Conferma“>
  </form>
 </body>
</html>
```



JavaScript

Il tipico impiego del javascript è il controllo dei dati inseriti in un form da un utente.

Altri utilizzi riguardano la grafica delle pagine, ad esempio cambiare l'aspetto di una immagini quando ci si passa sopra con il mouse (roll-over).

Il Java Script può interagire con altri "Oggetti" presenti nella pagina HTML, quali: Applet Java, ActiveX Control, etc…

eXstensible Markup Language: XML

E' un linguaggio di mark-up ideale per **applicazioni distribuite**;

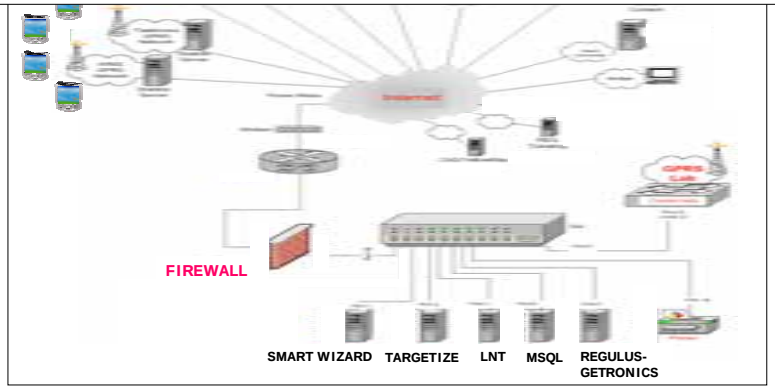
Mantiene la descrizione e la presentazione dei contenuti su due piani distinti;

Fornisce un linguaggio ad hoc, XSL, per la trasformazione da un formato ad un altro.

eXstensible Markup Language: XML

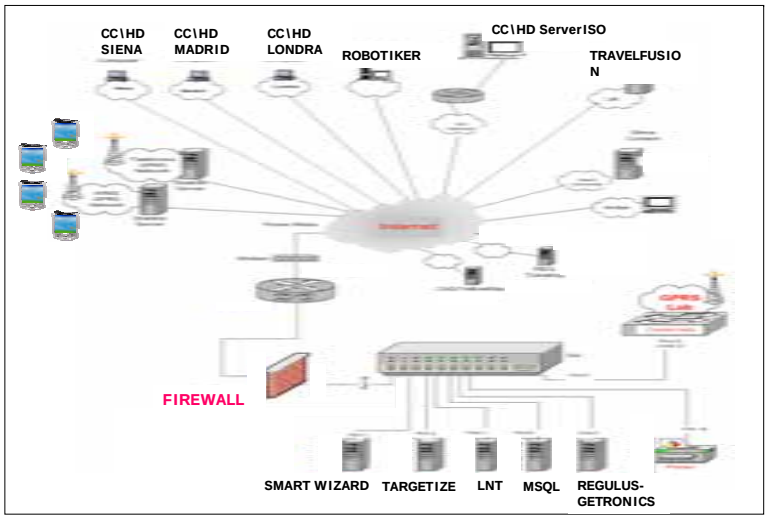
Sistema Distribuito

Un sistema distribuito è costituito da un insieme di elementi (hardware, software) fisicamente separati tra loro e che comunicano tramite una rete di comunicazione.



eXstensible Markup Language: XML

Sistema Distribuito

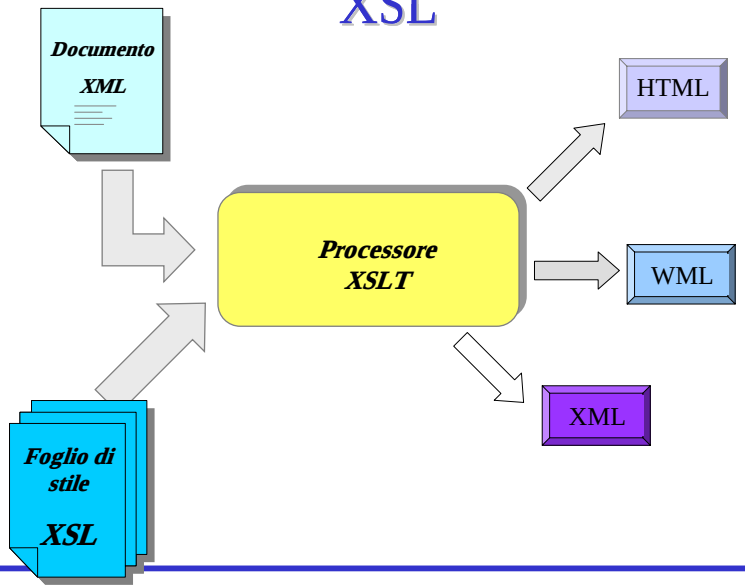


Un documento XML

```
<?xml version="1.0"?>


<CATALOGO>
 <PIANTA BESTSELLER="SI">
  <NOME>
   <COMUNE>Viola</COMUNE>
   <BOTANICO>Sanguinaria canadensis</BOTANICO>
  </NOME>
  <CRESCITA >
   <ZONA>Toscana</ZONA>
   <LUCE>Poca Luce</LUCE>
  </CRESCITA>
  <INFOVENDITA>
   <PREZZO>€ 3.00</PREZZO>
   <DISPONIBILITA>15/07/02</DISPONIBILITA>
  </INFOVENDITA>
 </PIANTA>
</CATALOGO>
```

XML e la presentazione dei contenuti: XSL



XML e la presentazione dei contenuti: XSL

```
<xsl:for-each select="CATALOGO/PIANTA">
       <TR>
        <xsl:apply-templates/>
       </TR>
      </xsl:for-each>
    </TABLE>
   </BODY>
 </HTML>
</xsl:template>

<xsl:template match="NOME">
 <TD><xsl:value-of select="COMUNE"/></TD>
 <TD><xsl:value-of select="BOTANICO"/></TD>
</xsl:template>

<xsl:template match="CRESCITA">
 <TD><xsl:value-of select="ZONA"/></TD>
 <TD><xsl:value-of select="LUCE"/></TD>
</xsl:template>
```

XML e la presentazione dei contenuti: XSL

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl"
href="mystyle.xsl"?>
<CATALOGO>
 <PIANTA BESTSELLER="SI">
  <NOME>
   <COMUNE>Viola</COMUNE>
   <BOTANICO>Sanguinaria canadensis</BOT
  </NOME>
  <CRESCITA >
   <ZONA>Toscana</ZONA>
   <LUCE>Poca Luce</LUCE>
  </CRESCITA>
  <INFOVENDITA>
   <PREZZO>€ 3.00</PREZZO>
   <DISPONIBILITA>15/07/02</DISPONIBILITA>
  </INFOVENDITA>
 </PIANTA>
```

```
<xsl:for-each select="CATALOGO/PIANTA">
       <TR>
        <xsl:apply-templates/>
       </TR>
      </xsl:for-each>
    </TABLE>
   </BODY>
 </HTML>
</xsl:template>

<xsl:template match="NOME">
 <TD><xsl:value-of select="COMUNE"/></TD>
 <TD><xsl:value-of select="BOTANICO"/></TD>
</xsl:template>

<xsl:template match="CRESCITA">
 <TD><xsl:value-of select="ZONA"/></TD>
 <TD><xsl:value-of select="LUCE"/></TD>
</xsl:template>
```

XML e la presentazione dei contenuti: XSL

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl"
href="mystyle.xsl"?>
<CATALOGO>
 <PIANTA BESTSELLER="SI">
  <NOME>
```

C:\Documents and Settings\aloha\Desktop\Maui.

File Modifica Visualizza Preferiti Strumenti ?

Indietro Cerca Prefe

Indirizzo C:\Documents and Settings\aloha\Desktop\Maui

Nome Comune	Nome Botanico	Zona di crescita	Luce	Prezzo	Disponibilità
Viola	Sanguinaria canadensis	Toscana	Poca Luce	€ 3.00	15/07/02
Colombina	Aquilegia canadensis	Liguria	Luce diffusa	€ 9.00	13/07/02

```
<xsl:for-each select="CATALOGO/PIANTA">
       <TR>
        <xsl:apply-templates/>
       </TR>
      </xsl:for-each>
    </TABLE>
   </BODY>
 </HTML>
</xsl:template>

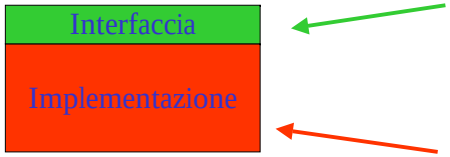
<xsl:template match="NOME">
 <TD><xsl:value-of select="COMUNE"/></TD>
 <TD><xsl:value-of select="BOTANICO"/></TD>
</xsl:template>

<xsl:template match="CRESCITA">
 <TD><xsl:value-of select="ZONA"/></TD>
 <TD><xsl:value-of select="LUCE"/></TD>
</xsl:template>
```

Il linguaggio Java: gli oggetti

Java è un linguaggio di programmazione orientato agli "oggetti". L'oggetto è l'unità minima di programmazione. La funzione di un oggetto è quella di rappresentare un "concetto" del problema che il programma deve risolvere.

Un oggetto è costituito da due parti: Interfaccia ed Implementazione.



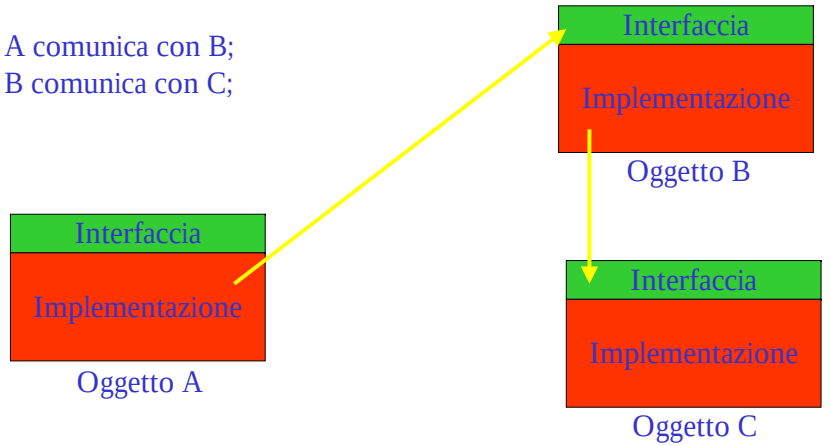
L'interfaccia rappresenta tutto ciò che è visibile ed utilizzabile esternamente: funzionalità e dati di un oggetto. Le funzionalità sono chiamate **metodi** ed i dati **attributi**.

L'implementazione è costituita dagli algoritmi che realizzano le funzionalità esposte nell'interfaccia, e da tutto ciò che non deve essere visibile esternamente

Comunicazione tra oggetti

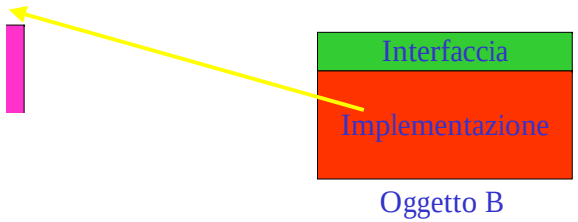
Gli oggetti appartenenti ad uno stesso programma possono comunicare tra di loro tramite un meccanismo detto a "scambio di messaggi":

A comunica con B;
B comunica con C;



Vantaggi della programmazione ad oggetti

E' possibile cambiare l'implementazione senza modificare l'interfaccia di un oggetto e senza influenzare la comunicazione con gli altri oggetti.



Vantaggi della programmazione ad oggetti

E' possibile cambiare l'implementazione senza modificare l'interfaccia di un oggetto e senza influenzare la comunicazione con gli altri oggetti.



Vantaggi della programmazione ad oggetti

E' possibile cambiare l'implementazione senza modificare l'interfaccia di un oggetto e senza influenzare la comunicazione con gli altri oggetti.



La portabilità del codice

La caratteristica fondamentale che ha decretato il successo di Java è la "portabilità".

Per portabilità si intende la possibilità di eseguire un programma su qualsiasi piattaforma (Sistema Operativo + CPU). Questo è possibile perché il codice compilato di un programma Java, ha un formato detto byte-code e non è un codice macchina.

Linguaggi tradizionali (C, Pascal, ecc):

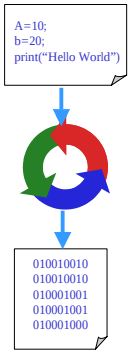


Codice Sorgente (scritto dal programmatore)

Compilatore (dipende dal Sistema Operativo)

Codice Macchina (eseguito dalla CPU)

La portabilità del codice Java

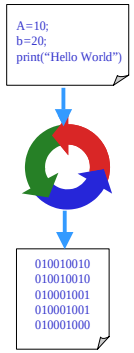


Codice Sorgente (scritto dal programmatore)

Compilatore (Non dipendente dal Sistema Operativo)

Byte-Code (Codice binario intermedio)

La portabilità del codice Java



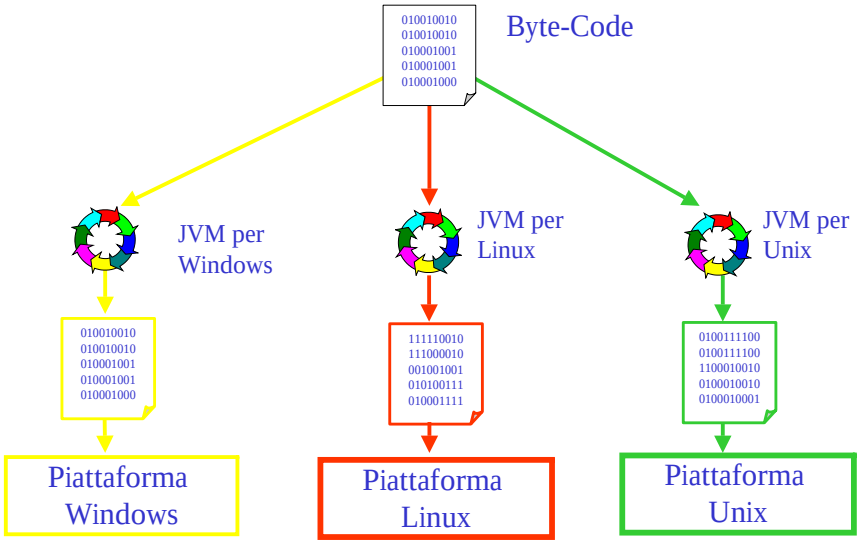
Codice Sorgente (scritto dal programmatore)

Compilatore (Non dipendente dal Sistema Operativo)

Byte-Code (Codice binario intermedio)

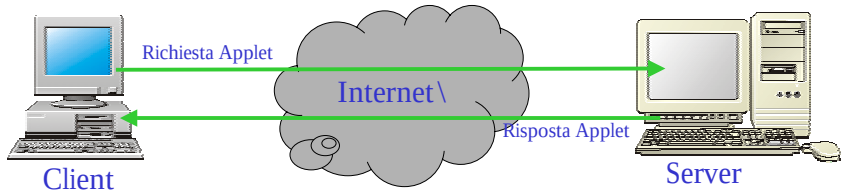
Il Byte-Code non è un codice eseguibile, ma deve essere interpretato da un apposito modulo detto "Java Virtual Machine"

Java byte-code e Virtual Machine



Le Applet Java

Sono particolari applicazioni Java incorporate nelle pagine Web. Vengono scaricate dai siti internet come le pagine Web, ed il fatto di essere scritte in Java garantisce la loro compatibilità con qualsiasi piattaforma client.

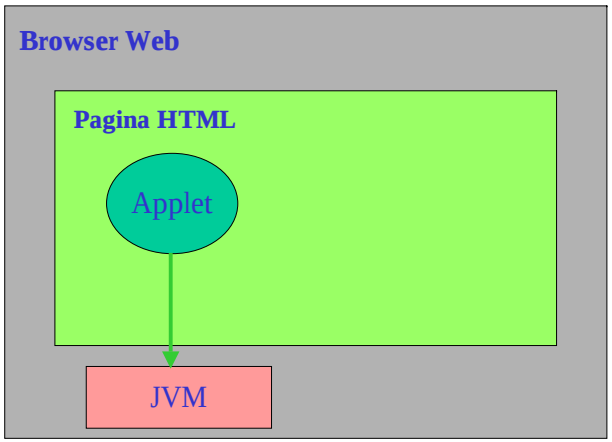


Le possibili aree d'impiego:

- Animazioni grafiche.
- Trasmissione di dati utente criptati verso Web Server.
- Accesso a particolari periferiche, ad esempio Lettori di Smart Card.
- …

Esecuzione delle Applet Java

Le applet Java vengono eseguite all'interno del browser web. Tali Browser incorporano una apposita Java Virtual Machine.



Java 2 Micro Edition

Una svolta decisiva nell'evoluzione dei terminali mobili è stata data dall'introduzione di una nuova piattaforma, univoca per tutti i dispositivi, che si avvale del linguaggio JAVA: il J2ME (Java 2 Micro Edition).

Sul sito internet della Sun è pubblicata una lista aggiornata contenente l'elenco dei terminali mobili sui quali è implementata la piattaforma J2ME. Questo elenco nel mese di Agosto 2002 conteneva circa 80 dispositivi, mentre nel Novembre 2003 ne contava più di 130, testimoniando l'integrazione pressoché totale di questa tecnologia in ogni GSM di ultima generazione.

E' evidente quindi come l'introduzione di una piattaforma per lo sviluppo di applicazioni unica per tutti i dispositivi di ogni marca e modello rappresenta una prospettiva estremamente interessante in primo luogo per i produttori.

Java 2 Micro Edition

Fin dalla sua introduzione la tecnologia J2ME è stata utilizzata quasi esclusivamente per la creazione di giochi, ed i moderni terminali con display grafico stanno decretando il pieno successo di tale settore.

In realtà una delle caratteristiche più interessanti del J2ME è quella di essere in grado di sfruttare le potenzialità di connessione alla rete offerte dal GSM: esistono precise classi JAVA che si occupano proprio della creazione e della gestione di connessioni ad internet tramite rete GPRS.

Grazie a tale peculiarità il J2ME si presta particolarmente alla realizzazione di servizi nei quali è previsto lo scambio di informazioni con un Server remoto.

Una Applicazione Java per terminali mobili si chiama **MIDLET**

Java 2 Micro Edition

HttpView



Esempio di una midlet che si collega ad un server e scarica una pagina html.

Java 2 Micro Edition

Esempio di una midlet che si collega ad un server Gis e scarica una mappa.



Generazione di pagine WEB(html/wml) Dinamica

Molto spesso è necessario generare le informazioni da trasferire al Client in maniera dinamica. Ad esempio si può generare una pagina Html/WML contenente l'elenco dei prodotti disponibili in magazzino, oppure realizzare pagine differenti in funzione dell'utente che vi accede.

Per realizzare pagine dinamicamente è necessaria una "qualche tecnologia" lato server, che permetta di inserire all'interno delle pagine stesse delle istruzioni che vengono elaborate dal server prima di restituire la pagina al Client.

Queste Tecnologie, quindi, permettono di realizzare quelle che si chiamano **Web Application**.

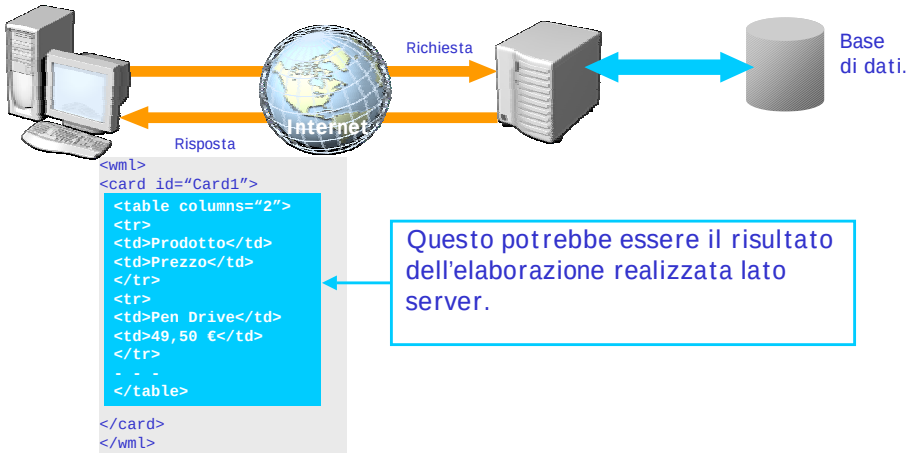
Alcuni esempi sono: ASP, Perl, PHP, JSP, etc…

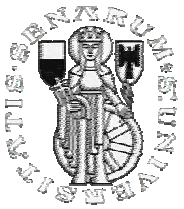
Generazione di pagine WEB(html/wml) Dinamica



Generazione di pagine WEB(html/wml) Dinamica

Per realizzare pagine dinamicamente è necessaria una "qualche tecnologia" lato server, che permetta di inserire all'interno delle pagine stesse delle istruzioni che vengono elaborate dal server prima di restituire la pagina al Client.





Fabio Burroni

Dipartimento di Ingegneria dell'Informazione

Università degli Studi di Siena

burronif@unisi.it