

Filippo Geraci



AJAX ED APPLICAZIONI WEB 2.0

Il web come piattaforma per le applicazioni

Caratteristica	Web app.	Desktop app.
Grafica	buona	Illimitata
Interattività	buona	Illimitata
Uso della rete	Molto elevato	Dipende da app.
Accessibile da	Ogni computer	Dove installato
Aggiornamenti	Server	Desktop

Molti programmi sono orientati agli eventi

Programma batch

- Struttura del programma
 - Leggi input
 - Effettua computazione
 - Produci output

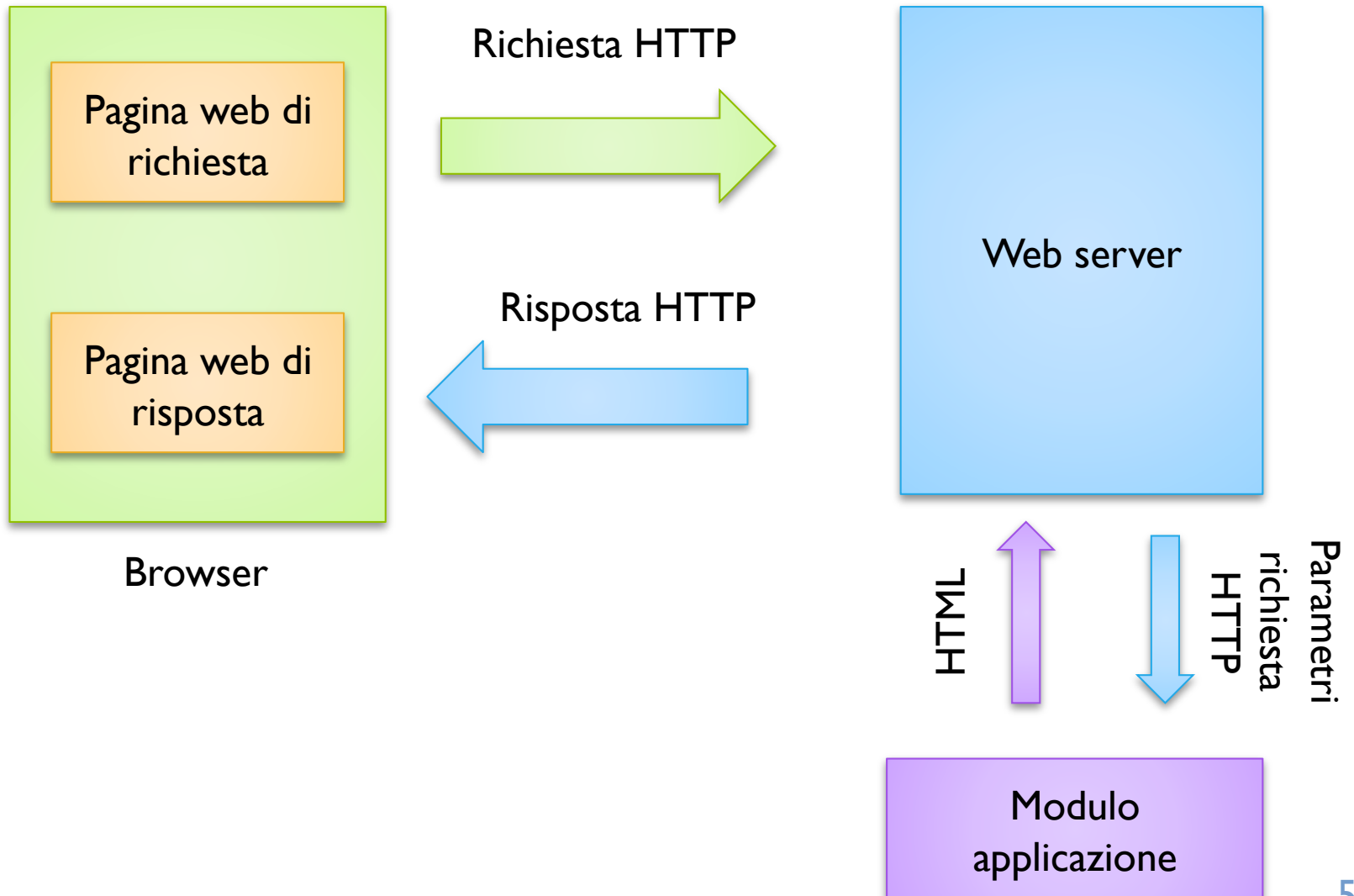
Programma ad eventi

- Struttura del programma
 - Aspetta un evento
 - Trova il metodo associato all'evento
 - Gestisci l'evento
 - ripeti
- L'aggiornamento del software consiste spesso nel catturare e gestire nuovi eventi

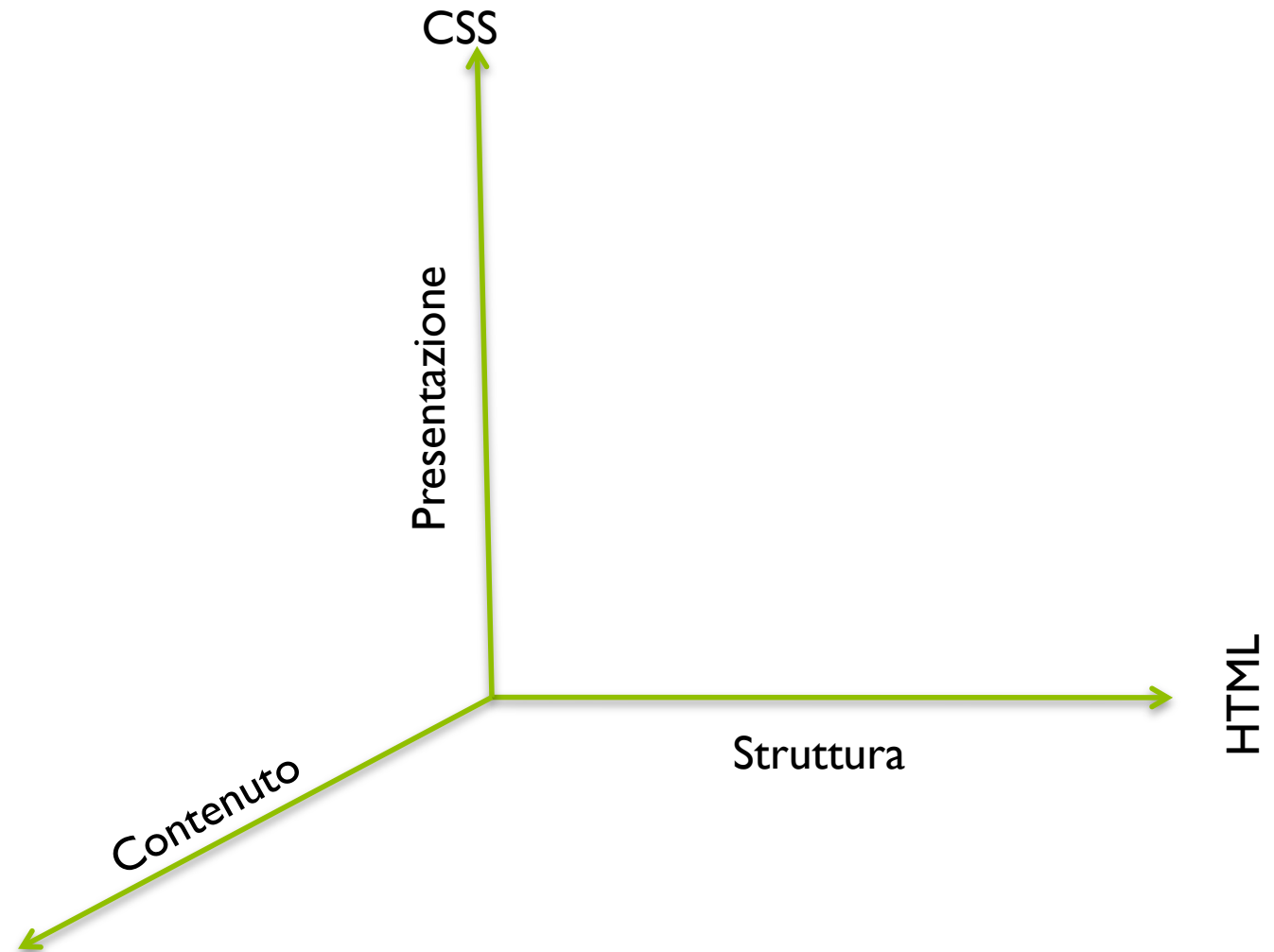
Qualche definizione

- **Server Web:** sono dei software che gestiscono delle richieste effettuate dai client
 - software che fornisce l'infrastruttura e le funzionalità di supporto, sviluppo ed esecuzione di applicazioni e componenti server.
- **Client Web (browser):** software che effettua le richieste ai server web
- **Protocollo di rete:** regole per la comunicazione su internet. Ogni servizio su internet ha un proprio protocollo.
 - **Http:** il protocollo di comunicazione usato per il web
 - **Https:** versione crittografata del protocollo http
- **Html:** è un linguaggio di markup con cui sono fatte le pagine web
 - E' un linguaggio di markup (contrassegno basato su tag) per la formattazione di testo su web (definito e gestito dal consorzio W3C)

Comunicazione client server web

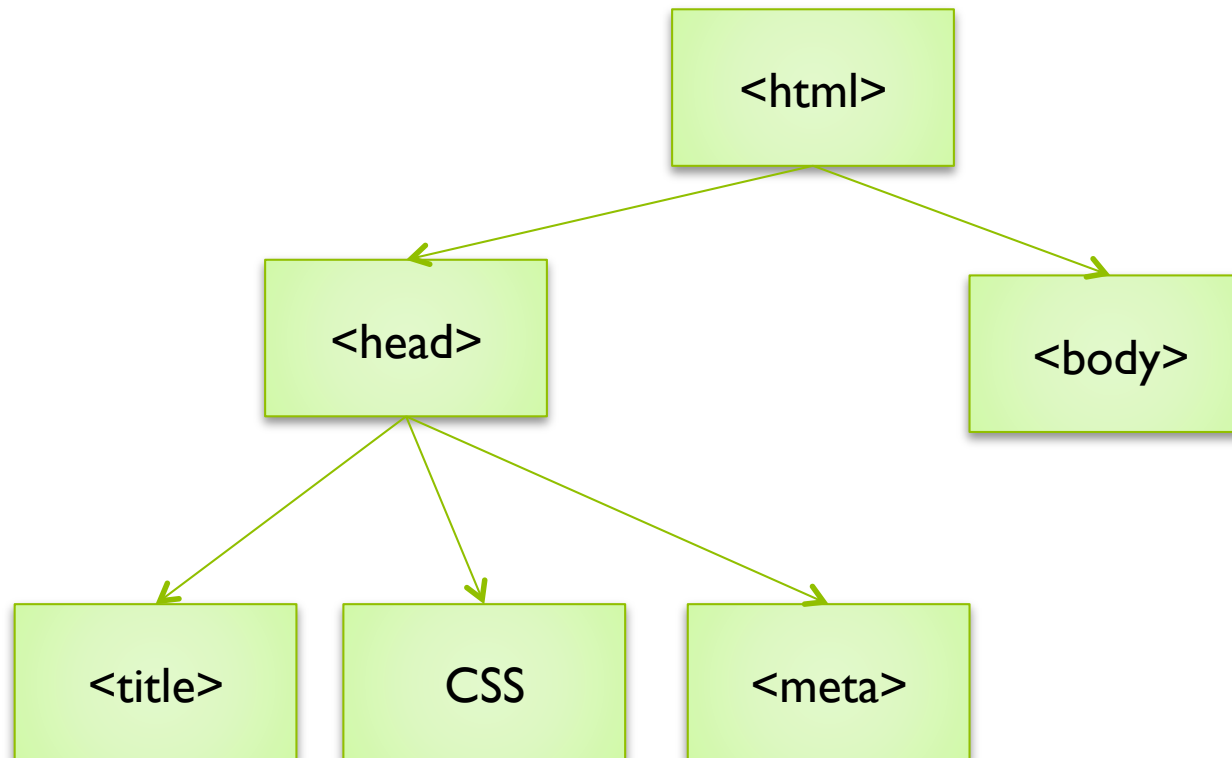


Le tre dimensioni delle pagine web



Struttura di un documento HTML

- Struttura dei tag annidati definisce un albero





I tag HTML

- **HTML:** indica l'inizio e la fine della pagina html.
- **HEAD:** contiene tutte le intestazioni della pagina (tipicamente non visualizzate)
 - Titolo, inclusione di javascript o CSS, meta tags
- **META:** contiene informazioni sulla pagina
 - Autore, keywords, data di creazione
- **BODY:** contiene tutta la parte visibile della pagina

Alcuni tag utili

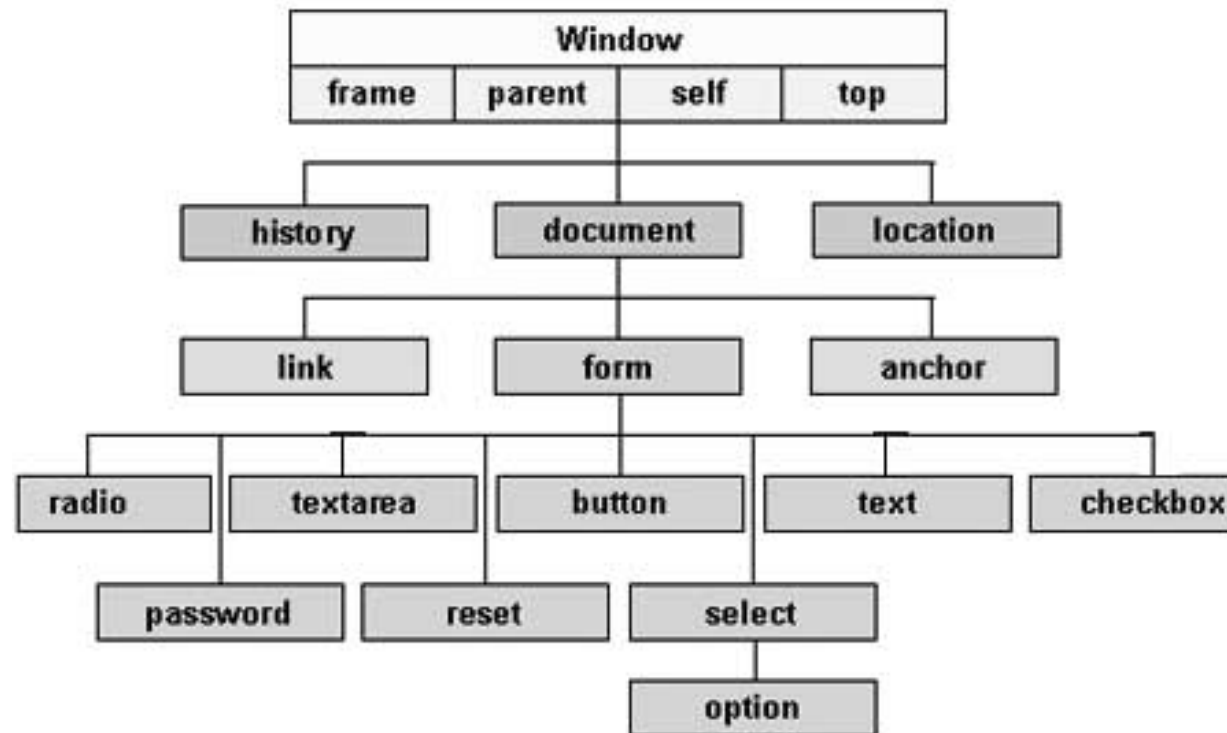
- **DIV:** viene usato come contenitore di parti della pagina che devono avere caratteristiche comuni
 - Esempio: visualizzazione tramite fogli di stile CSS
- **FORM:** contenitore di una maschera per l'inserimento dati e l'invio al web server
 - Il web server risponde inviando una nuova pagina web visualizzata dal browser al posto di quella che ha fatto la richiesta
 - Sulla modifica di questo meccanismo di comunicazione si basa il web 2.0

Passaggio di parametri

- Specificato tramite il parametro method del tag form
- Si passano in due modi: get, post.
 - **Get:** i dati concatenati nella URL della pagina di risposta
 - `http://host/page?param1=value1¶m2=value2`
 - **Post:** l'invio dei dati avviene dopo aver contattato la pagina sul web server
 - Usato per passare molti dati o non farli vedere/modificare all'utente

Document object model (DOM)

- Rappresentazione ad oggetti della pagina web
- Rappresentazione ad albero
- Si possono aggiungere/rimuovere elementi o cambiarne gli attributi



JavaScript lato client

- Eseguito dal browser
- Si evolve quindi ci sono problemi di compatibilità
 - Microsoft non rispetta gli standard
 - Esistono librerie che attenuano le difficoltà
- La programmazione lato client consente di
 - Validare l'input prima di mandarlo al server
 - Catturare eventi e reagire ad azioni dell'utente
 - Aggiornare parti della pagina senza richiederla al server

Alcuni eventi javascript

Nome funzione	Quando viene catturata
onclick	Click del mouse su un elemento
onchange	L'oggetto cambia stato
onfocus	L'elemento ottiene il focus
onsubmit	Sottomissione del form
onmouseover	Il mouse si muove sopra l'elemento
onmouseout	Il mouse "esce" dall'elemento
onmousedown	Pressione bottone sinistro del mouse
onmouseup	Rilascio bottone sinistro del mouse
onkeydown	Pressione di un tasto
onkeypress	Pressione e rilascio di un tasto
onkeyup	Rilascio di un tasto
onload	Fine caricamento della pagina

Modificare l'HTML con javascript

- `document.getElementById(id)`
- `document.getElementsByTagName(tag)`
- `<tag>Testo</tag>`
 - `innerHTML` → Testo
 - `outerHTML` → `<tag>Testo</tag>`

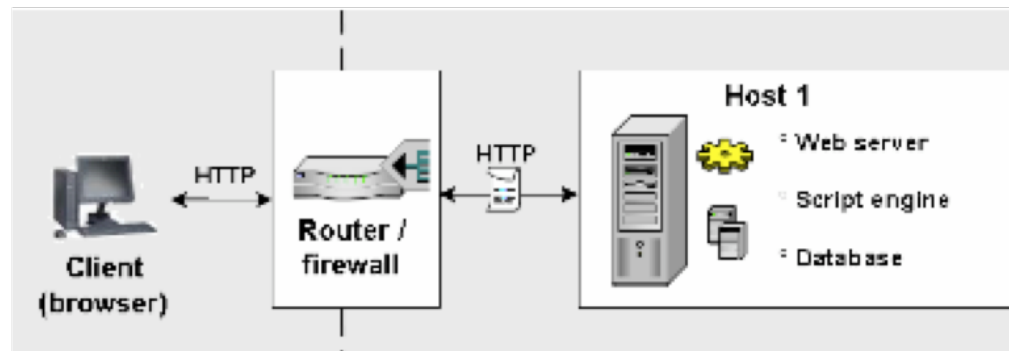


Programmazione lato server

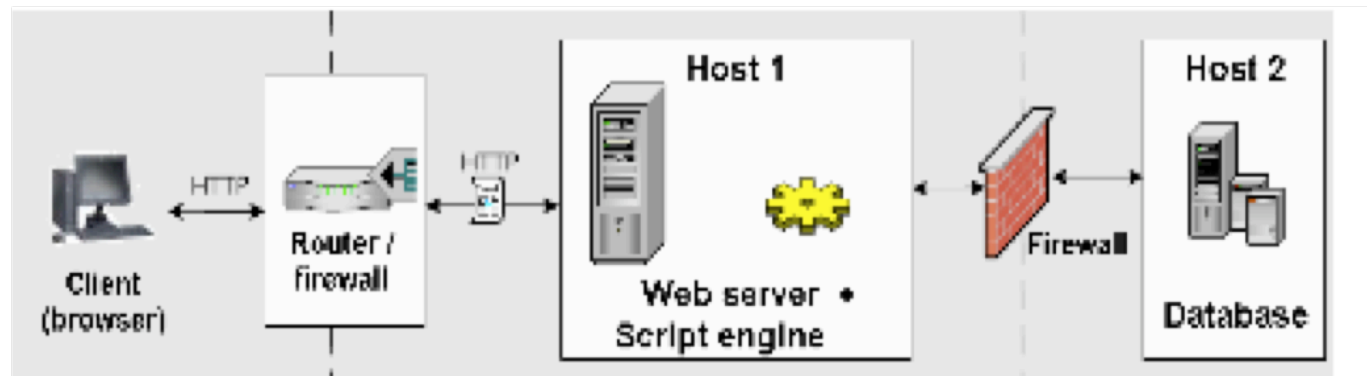
- CGI – programmi stand-alone lanciati dal web server
 - Producono un documento HTML in output
 - Ricevono parametri in input tramite HTTP
- Programmi inseriti dentro la pagina web (php, ASP, JSP)
 - Eseguiti dentro il processo del web server
 - Errori di programmazione possono influenzare le prestazioni del web server
- Client e server usano linguaggi diversi
- A cosa servono le pagine dinamiche?
 - Personalizzazione dei risultati
 - Interazione con database
 - Applicazioni web

Sistemi Web-based a tier multipli (MEMO)

- **2 tier:** un solo server per script engine: logica applicativa e database



- **3 tier dual host:** server separati per database e script engine



Usare PHP per creare pagine dinamiche

- Il codice viene inserito all'interno della pagina
- Interpretato da un modulo eseguito dal web server
- “stampa” l'html che verrà visualizzato nella pagina

```
<html>
...
<body>
<?php
echo 'Hello world' ;
?>
</body>
</html>
```

Usare PHP per interagire con il database

```
<?php
// Connessione al database
mysql_connect('host', 'user', 'password');
mysql_select_db('dbname');
// seleziona dato da una tabella
$nome = 'filippo';
$res = mysql_query("SELECT * FROM 'people' WHERE nome='$nome'");
$filippo = mysql_fetch_assoc ($res);
Print_r ($filippo);
?>
```

Nome	Cognome	Age
Filippo	Geraci	33
Mario	Rossi	40

people

```
array (
  ['nome'] => 'Filippo'
  ['cognome'] => 'Geraci'
  ['age'] => 33
)
```

Il passaggio dei parametri - POST

Scrivi il tuo nome

```
Sorgente d
<html>
<head></head>
<body>
<form action="prova.php" method="post">
Scrivi il tuo nome
<input type="text" name="nome">
<input type="submit" value="Invia">
</form>
</html>
```



`$_POST["nome"]`

Il passaggio dei parametri - GET

Scrivi il tuo nome

```
Sorgen  
<html>  
<head></head>  
<body>  
<form action="prova.php" method="get">  
Scrivi il tuo nome  
<input type="text" name="nome">  
<input type="submit" value="Invia">  
</form>  
</html>
```



`$_GET["nome"]`



Cosa so fare a questo punto?

- So scrivere l'html
- So come modificarlo dinamicamente tramite javascript
- So costruire form per passare parametri al server chiamando altre pagine
- Conosco i meccanismi di comunicazione get e post tra client e server
- So leggere con php i parametri che mi sono stati passati per fare elaborazioni
- So interfacciarmi con il database MySQL

AJAX



- Asynchronous
- JavaScript
- ~~● And~~
- ~~● XML~~

AJAX



- Insieme di tecnologie correlate per lo sviluppo di applicazioni web.
- Permette di contattare in maniera asincrona senza interferire con la visualizzazione ed il comportamento di una pagina

Effettuare la richiesta remota

```
function loadXMLDoc(url) {  
    // codice per Mozilla, etc.  
    if (window.XMLHttpRequest) {  
        xmlhttp=new XMLHttpRequest()  
        xmlhttp.onreadystatechange=xmlhttpChange  
        xmlhttp.open("GET" ,url,true)  
        xmlhttp.send(null)  
    }  
    // codice per IExplore  
    else if (window.ActiveXObject) {  
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP")  
        if (xmlhttp) {  
            xmlhttp.onreadystatechange=xmlhttpChange  
            xmlhttp.open("GET" ,url,true)  
            xmlhttp.send()  
        }  
    }  
}
```


Prototype



- Prototype e' un framework javascript che semplifica lo sviluppo di applicazioni web
- Libreria per accedere al DOM con sintassi ad oggetti semplificata
- Si fa carico delle differenze tra browser diversi
- Completamente open source
- Disponibile a: <http://www.prototypejs.org/>

Prototype – Come funziona

- Includere la libreria

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

Comunicazione con ajax

Pagina html

```
<html>
<head>
<script
  type="text/javascript"
  src="prototype.js">
</script>
</head>
<body>
<div id="container">
  Scrivi il tuo nome
  <input type="text" id="nome">
  <input type="button"
    value="Invia"
    onClick="call (url);">
</div>
</body>
</html>
```

Codice javascript

```
function call () {
  Ajax.Request(url, {
    method: 'post', onSuccess: runOk,
    onFailure: runFail,
    parameters: {name: $("name").value}
  });
}
function runOk (transport) {
  $('container').innerHTML =
  transport.responseText;
}
function runFail (transport) {
  alert ("Errore in AJAX");
}
```

Codice PHP

```
<?php
echo $_POST["name"];
?>
```

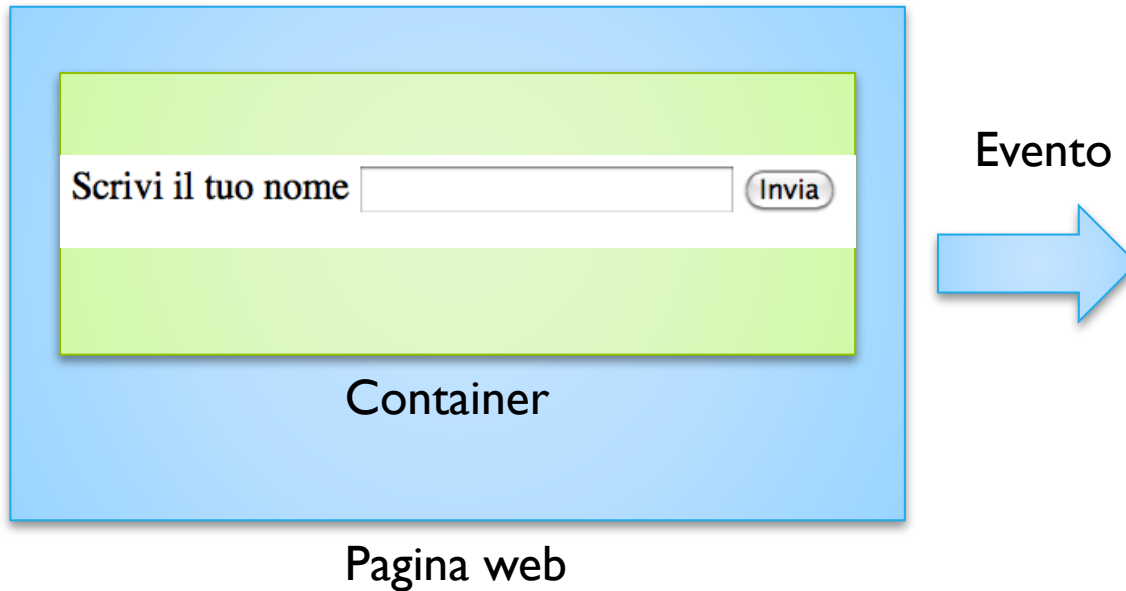
Scrivi il tuo nome

Invia



Filippo

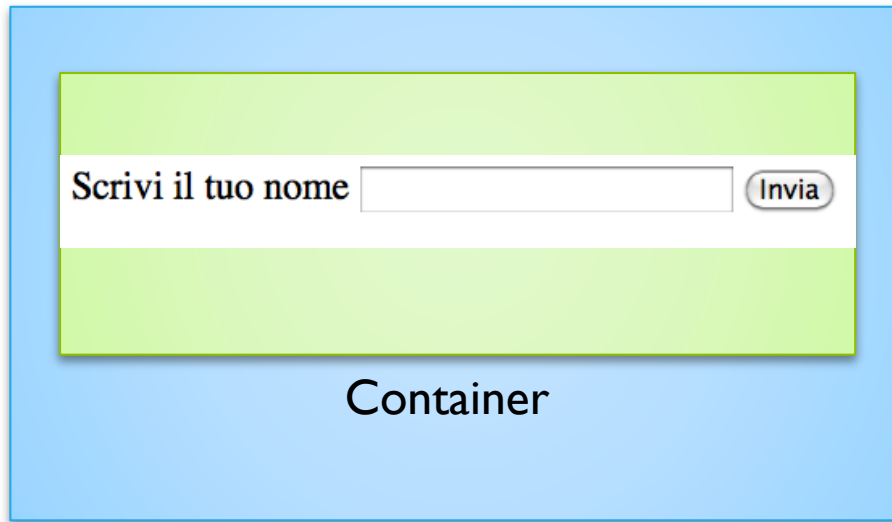
Come funziona



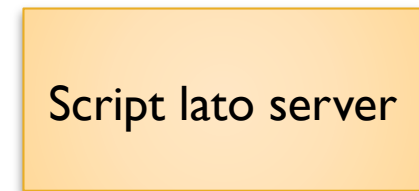
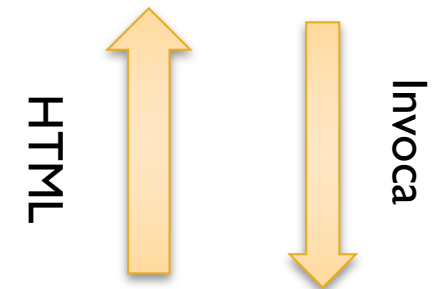
  Lato client

 Lato server

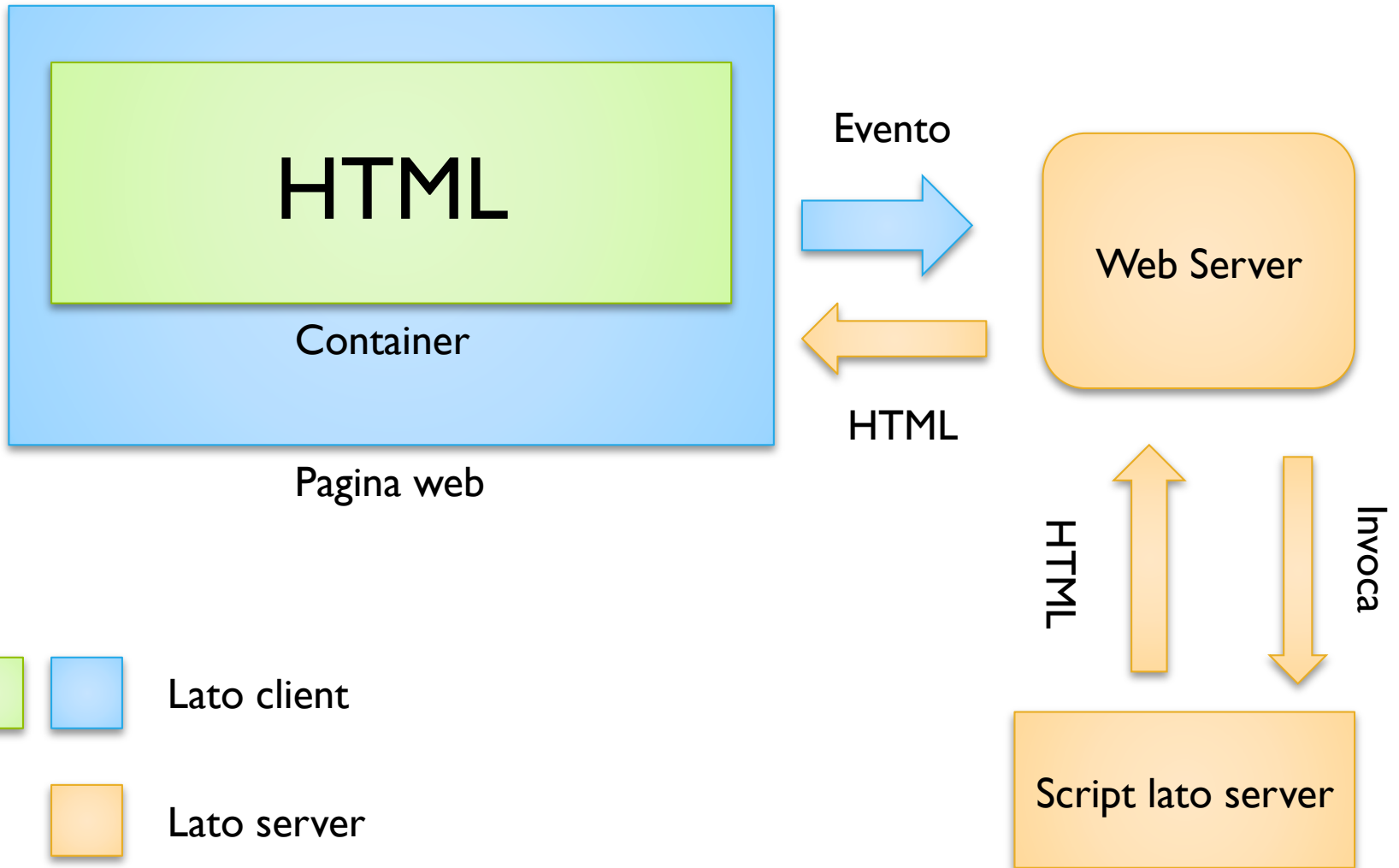
Come funziona



Pagina web

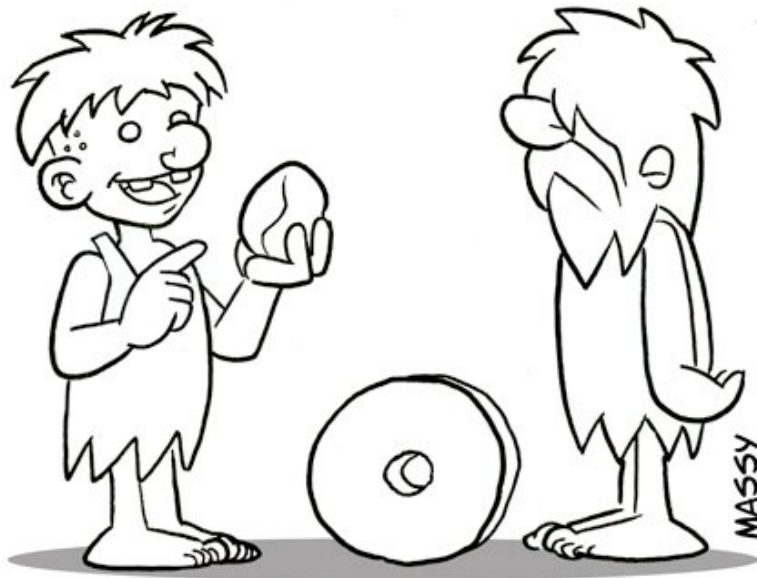


Come funziona



Il motto della programmazione Web 2.0

- HO INVENTATO IL FRENO.



- Non reinventare la ruota
- Disponibili tanti framweork per:
 - Gestire l'interfaccia grafica
 - Gestire la comunicazione con il server
 - Accedere al DOM
 - “Animare” gli oggetti



Script.aculo.us

- Libreria di effetti visuali per aggiungere animazioni
- Libreria per il drag-and-drop
- Creazione/modifica dinamica di elementi del DOM
- Controlli ajax
 - **Esempio:** Completamento automatico, In Place Editing
- Disponibile su: <http://script.aculo.us>



Qualche esempio di cosa può fare

- Effetti principali:
 - Evidenzia, sposta, opacizza, scala
- Combinazioni di effetti:
 - Fai apparire, avvolgi, svolgi, lampeggia, vibra, strizza, spegni
- Altri effetti:
 - Sposta
- Comportamento dei componenti:
 - Draggable, Droppables, Ordinabile
- Controlli:
 - Editor in place, autocompletamento
- altro:
 - Suoni

Esempio

```
new Effect.Morph('error_message', {  
  // CSS Properties  
  style: 'background:#f00; color: #fff;',  
  // Core Effect properties  
  duration: 0.8  
})
```

Demo



Demo





Librerie per interfacce grafiche

- Ce ne sono tante
- Fanno praticamente di tutto
- Alcune commerciali alcune open source
- Non faccio pubblicità, vi mostro qualcosa assemblata da me

Librerie per interfacce grafiche

- Ce ne sono tante
- Fanno praticamente di tutto
- Licenze commerciali ed open source
- Non faccio pubblicità, vi mostro qualcosa assemblato da me

