

Ajax

Asynchronous JavaScript and XML

© Alberto Ferrari

• 1

Definizione

- Asynchronous JavaScript and XML (AJAX) non è di per sé una tecnologia, ma è un termine che descrive un "nuovo" approccio all'utilizzo di diverse tecnologie esistenti, compresi: HTML o XHTML, CSS, JavaScript, DOM, XML, XSLT e l'oggetto XMLHttpRequest.
- Grazie all'utilizzo di queste tecnologie in combinazione con il modello AJAX, le applicazioni web possono eseguire aggiornamenti rapidi e incrementali dell'interfaccia utente senza ricaricare nel browser l'intera pagina.
- Questo rende l'applicazione più performante e più reattiva alle azioni dell'utente.

<https://developer.mozilla.org/it/AJAX>

© Alberto Ferrari

• 2

AJAX

- AJAX è uno strumento di sviluppo per la realizzazione di **applicazioni web interattive** (Rich Internet Application).
- La tecnologia AJAX si basa su uno scambio di dati in background fra web browser e server, che consente l'aggiornamento dinamico di una pagina web senza esplicito ricaricamento da parte dell'utente.
- La richiesta è asincrona: non si deve attendere che sia ultimata per effettuare altre operazioni.
- Normalmente le funzioni richiamate sono scritte con il linguaggio JavaScript.
- AJAX è una tecnica multi-piattaforma.

© Alberto Ferrari

• 3

Caratteristiche

- Sfrutta l'oggetto XMLHttpRequest.
- Offre la possibilità di interpretare e lavorare con i documenti XML
- Reso popolare da Google, a partire dal 2005.
- Utilizzato in Google Maps, Gmail ...
- Il vantaggio di usare AJAX è la grande velocità alla quale un'applicazione risponde agli input dell'utente.

© Alberto Ferrari

• 4

Richiesta HTTP

- La richiesta HTTP al server utilizza un'istanza di una classe.
- Questa classe è stata introdotta originariamente in Internet Explorer come oggetto ActiveX e si chiamava XMLHttpRequest.
- Successivamente Mozilla, Safari e altri browser hanno implementato la classe XMLHttpRequest, che supporta gli stessi metodi e le stesse proprietà della classe di Microsoft.

© Alberto Ferrari

• 5

Come fare

- Per creare un oggetto per la richiesta http occorre scrivere codice diversificato per i diversi browser

```
if (window.XMLHttpRequest) { // Mozilla, Safari, ...  
    http_request = new XMLHttpRequest();  
} else if (window.ActiveXObject) { // IE  
    http_request = new ActiveXObject("Microsoft.XMLHTTP");  
}
```

© Alberto Ferrari

• 6

Stabilire cosa fare con la risposta inviata dal server

- Bisogna dire all'oggetto XMLHttpRequest quale funzione JavaScript elaborerà il codice XML.
- Questo si fa impostando la proprietà onreadystatechange dell'oggetto con il nome della funzione JavaScript.
- `http_request.onreadystatechange = nomeFunzione;`
- In questo modo si sta assegnando un riferimento alla funzione, non la si sta ancora chiamando.

• Alberto Ferrari

• 7

Inviare la richiesta al server

- Per inviare la richiesta bisogna chiamare i metodi `open()` e `send()`

```
http_request.open("GET", "http://www.nameserver.org/qualsiasi.file", true);
http_request.send(null);
```
- `open`
 - Parametro 1: metodo : GET, POST, HEAD o qualsiasi altro metodo che si desidera utilizzare (scritto in lettere MAIUSCOLE).
 - Parametro 2: URL (non è possibile chiamare pagine che si trovino su un dominio differente da quello in cui si trova la pagina corrente).
 - Parametro 3: true per richiesta asincrona, false sincrona
- `send`
 - Il parametro è costituito dai dati che si vogliono inviare al server se la richiesta è di tipo POST.

• Alberto Ferrari

• 8

Come gestire la risposta del server

- La funzione deve controllare lo stato della richiesta. Se lo stato ha un valore di 4, significa che la risposta è stata ricevuta per intero.

```
if (http_request.readyState == 4) {
    // tutto ok, la risposta è stata ricevuta
} else {
    // non ancora pronto
}
```

- Bisogna poi controllare il codice di stato della risposta http: il codice 200 è OK.

```
if (http_request.status == 200) {
    // bene!
} else {
    // problema nella richiesta,
}
```

• Alberto Ferrari

• 9

Come accedere ai dati ricevuti

- `http_request.responseText`
 - restituisce la risposta sotto forma di stringa di testo
- `http_request.responseXML`
 - restituisce la risposta sotto forma di oggetto XMLDocument che si può navigare tramite le funzioni DOM.

• Alberto Ferrari

• 10

Cosa fare con i dati ricevuti

- Selezionare un elemento della pagina
 - `getElementById(string)`
 - è un metodo importantissimo del DOM. Permette di selezionare un elemento mediante il suo `id`. Comportamento differente dei vari browser.
- Modificare il valore dell'elemento
 - `innerHTML`
 - proprietà che permette di ottenere o modificare il testo e le caratteristiche di un nodo.

• Alberto Ferrari

• 11

getElementById

```
function prendiElementoDaId(id_elemento) {
    var elemento; // elemento da restituire

    // se esiste il metodo getElementById
    if(document.getElementById)
        elemento = document.getElementById(id_elemento);

    // altrimenti: vecchio sistema
    else
        elemento = document.all[id_elemento];

    // restituzione elemento
    return elemento;
}
```

• Alberto Ferrari

• 12

getElementById esempio d'uso

- Esempio

```
function cambiaColore(nuovoColore)
{
  elem = document.getElementById("paragrafo1");
  elem.style.color = nuovoColore;
}
```

• Alberto Ferrari

• 13

innerHTML

- La proprietà innerHTML permette di ottenere o modificare il testo e le caratteristiche di un nodo.

• Alberto Ferrari

• 14

innerHTML - esempio

- Cambiare dinamicamente un link

```
function cambiaLink()
{
  document.getElementById('mioLink').innerHTML="sito";
  document.getElementById('mioLink').href="http://www.sito.com";
  document.getElementById('mioLink').target="_blank";
}
```

• Alberto Ferrari

• 15

Esempi

- Esempio 01
 - Visualizzazione di dati presenti sul server in un file di testo
- Esempio 02
 - Analogo al precedente ma richiedendo un file XML
- Esempio 03
 - Uso di innerHTML
- Esempio 04
 - Tooltip

• Alberto Ferrari

• 16