

AJAX

Riccardo Rosati

Linguaggi e tecnologie per il Web
Corso di laurea in Ingegneria informatica e automatica
Sapienza Università di Roma
a.a. 2017/2018

<http://www.dis.uniroma1.it/~rosati/lw/>



SAPIENZA
UNIVERSITÀ DI ROMA

AJAX

- AJAX = Asynchronous Javascript and XML
- Tecnologia basata su JavaScript e sull'interazione **asincrona** tra web client e web server
- Interazione asincrona: il web client (browser) fa una richiesta al server ma, a differenza delle altre forme di interazione previste, NON si interrompe fino all'arrivo della risposta (response) da parte del server
- Inoltre è possibile (ri)caricare solo una piccola parte della pagina attualmente visualizzata dal browser: ciò comporta maggiore velocità di esecuzione del browser e maggiore fluidità nell'interazione con l'utente

AJAX

- XML è il formato inizialmente previsto per lo scambio dei dati tra client e server
- Tuttavia, i dati possono essere scambiati anche in altri formati (JSON o altro)

AJAX

- AJAX è basato sull'interazione asincrona tra server e client
- **XML Http Request Object (XHR)** è l'oggetto che permette la comunicazione asincrona tra client e server
- Per attivare tale tipo di comunicazione, il client deve creare un nuovo oggetto XHR ed usare gli attributi e i metodi di questo oggetto
- XHR non è un oggetto standard del DOM W3C, ma è standard WHATWG ed è attualmente supportato da tutti i browser (ma in modi diversi)

XMLHttpRequest object: attributi

| | |
|--------------------|---|
| readyState | 1 = Open 2 = Sent 3 = Received 4 = Loaded |
| Status | 200 = ok 404 = page not found |
| statusText | Contiene l'etichetta dello status |
| responseText | Contiene i dati caricati (stringa di caratteri). Assume il suo valore finale quando readyState diventa uguale a 4 |
| responseXML | Contiene il documento XML caricato (oggetto XML document). E' significativo solo se readyState = 4, altrimenti è null |
| onreadystatechange | Funzione invocata quando cambia l'attributo readyState |

XHR object: metodi

| | |
|--|---|
| abort() | Interrompe tutte le attività create dell'oggetto e lo resetta |
| getAllResponseHeaders() () | Restituisce tutti gli header in una stringa |
| getResponseHeader(DOMString) | Restituisce gli header dei dati ricevuti dopo l'ultima request |
| open(mode, url, boolean [, login, password]) | mode: tipo della HTTP request (GET, POST, HEAD...) url: indirizzo del file boolean: true (asincrona) / false (sincrona) (login e password opzionali) |
| send("string") | Invia una stringa. Se la request è GET deve essere vuoto o null. Causa una DOMException (INVALID_STATE_ERR) se readyState è diverso da 1 |
| setRequestHeader(DOMString, DomString) | Gli argomenti sono nome dell'header e valore Causa una DOMException (INVALID_STATE_ERR) se readyState è diverso da 1 |

AJAX: esempio

```
<!DOCTYPE HTML>
<html>
  <head><title>AJAX: semplice esempio</title></head>
  <body>
    <div>
      <button>Documento_1</button>
      <button>Documento_2</button>
      <button>Documento_3</button>
      <button>Documento_4</button>
    </div>
    <hr/>
    <div id="zonaDinamica">
      Seleziona il documento da visualizzare
    </div>
    <hr/>
    Resto del documento<br/>
    ...
```

AJAX: esempio (segue)

```
<script>
  var documenti = document.getElementsByTagName("button");
  for (var i = 0; i < documenti.length; i++) {
    documenti[i].onclick = caricaDocumento;
  }

  function caricaDocumento(e) {
    var httpRequest = new XMLHttpRequest();
    httpRequest.onreadystatechange = gestisciResponse;
    httpRequest.open("GET", e.target.innerHTML+".htm", true);
    httpRequest.send();
  }

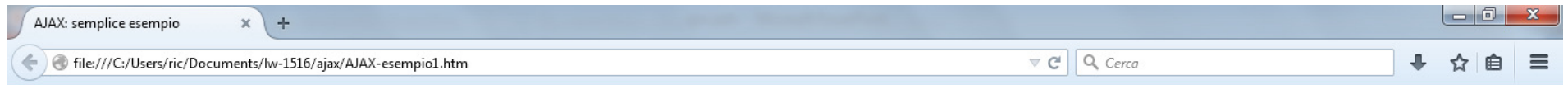
  /* si noti l'uso della proprietà target degli eventi,
  che restituisce l'elemento che ha causato l'evento */
```


AJAX: esempio (segue)

```
function gestisciResponse(e) {  
    if (e.target.readyState == 4 && e.target.status == 200) {  
        document.getElementById("zonaDinamica").innerHTML  
        = e.target.responseText;  
    }  
}  
</script>  
</body>  
</html>
```

Esempio

Documento iniziale:



Documento_1 Documento_2 Documento_3 Documento_4

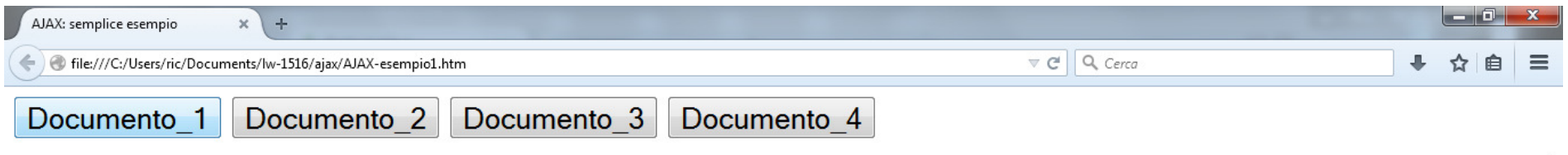
Seleziona il documento da visualizzare

Resto del documento

...

Esempio (segue)

Cliccando sul bottone Documento_1 si ottiene:



Documento 1

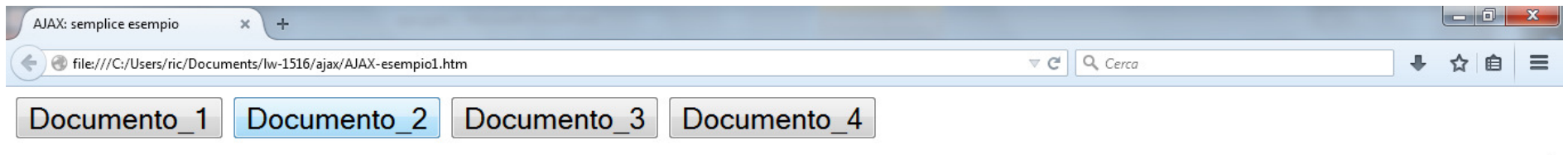
Questo è il documento 1

Resto del documento

...

Esempio (segue)

Cliccando sul bottone Documento_2 si ottiene:



Documento 2

Questo è il documento 2

Resto del documento

...

Esempio: commenti

- Nota bene: ad ogni clic su un bottone, non viene ricaricato tutto il documento
- Solo il contenuto del tag `<div id="target">` del documento iniziale viene modificato dall'interazione asincrona con il server
- Ricordiamo che ad ogni invocazione di funzione associata ad un evento viene automaticamente passato un argomento corrispondente all'evento stesso (argomento e nelle dichiarazioni delle funzioni dell'esempio)
- L'evento ha l'importante proprietà **target** che contiene l'oggetto che ha causato l'evento

Interazione sincrona

- Nel caso di open sincrone (attributo boolean=false), l'interprete JavaScript si ferma fino a che non viene ricevuta risposta dal server
- per questo le open sincrone vanno in genere evitate o comunque limitate

Caricamento di documenti XML

- AJAX e l'oggetto XMLHttpRequest trattano in modo speciale il caso in cui la risorsa oggetto della comunicazione tra client e server sia un documento XML
- In tal caso, il documento è memorizzato, sottoforma di oggetto XML document, nell'attributo **responseXML** (e non in responseText)
- per manipolare tale oggetto si possono usare i metodi del Document Object Model (DOM) di XML

Esempio con risorsa XML

Supponiamo ora che i precedenti documenti da caricare siano file XML.
Supponiamo ad esempio che il file Documento_1.xml sia:

```
<?xml version="1.0"?>
<radiceDocumento>
  <titolo>
    Documento 1
  </titolo>
  <contenuto>
    ...
  </contenuto>
</radiceDocumento>
```


Esempio con risorsa XML (segue)

(stessa struttura per i file Documento_2.xml, Documento_3.xml e Documento_4.xml)

In tal caso tali documenti verrebbero ricevuti nell'attributo responseXML.

La funzione caricaDocumento andrebbe modificata nel seguente modo:

```
function caricaDocumento(e) {  
    var httpRequest = new XMLHttpRequest();  
    httpRequest.onreadystatechange = gestisciResponse;  
    httpRequest.open("GET", e.target.innerHTML + ".xml", true);  
    httpRequest.send();  
}
```

Esempio con risorsa XML (segue)

La funzione `gestisciResponse` andrebbe modificata come segue:

```
function gestisciResponse(e) {  
    if (e.target.readyState == 4 && e.target.status == 200){  
        var resp = e.target.responseXML;  
        var x = resp.getElementsByTagName("titolo");  
        document.getElementById("zonaDinamica").innerHTML=  
            "<h1>" + x[0].childNodes[0].nodeValue + "</h1>";  
    }  
}
```

(viene visualizzato il contenuto testuale dell'elemento titolo del file XML)