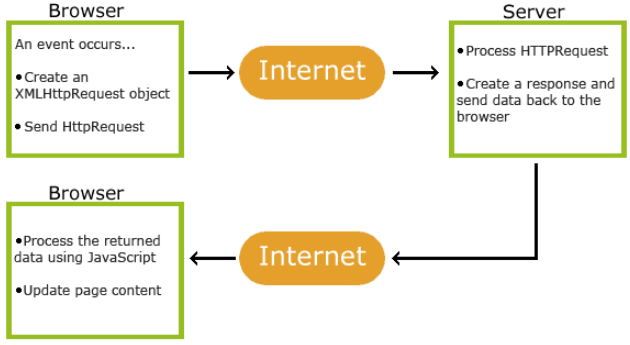
**AJAX (Asynchronous JavaScript And XML) -** [**https://www.w3schools.com/js/js\_ajax\_intro.asp**](https://www.w3schools.com/js/js_ajax_intro.asp)

* Leggi i dati da un server web - dopo che la pagina è stata caricata
* Aggiorna una pagina Web senza ricaricare la pagina
* Invia dati a un server web - in background

AJAX utilizza solo una combinazione di:

* Un oggetto ***XMLHttpRequest*** integrato nel browser (per richiedere dati da un server Web)
* ***DOM*** ***JavaScript*** e ***HTML*** (per visualizzare o utilizzare i dati)



1. Un evento si verifica in una pagina Web (la pagina viene caricata, si fa clic su un pulsante)
2. Un oggetto XMLHttpRequest viene creato da JavaScript
3. L'oggetto XMLHttpRequest invia una richiesta a un server web
4. Il server elabora la richiesta
5. Il server invia una risposta alla pagina web
6. La risposta viene letta da JavaScript
7. L'azione corretta (come l'aggiornamento della pagina) viene eseguita da JavaScript

**Oggetto XMLHttpRequest -** [**https://www.w3schools.com/js/js\_ajax\_http.asp**](https://www.w3schools.com/js/js_ajax_http.asp)

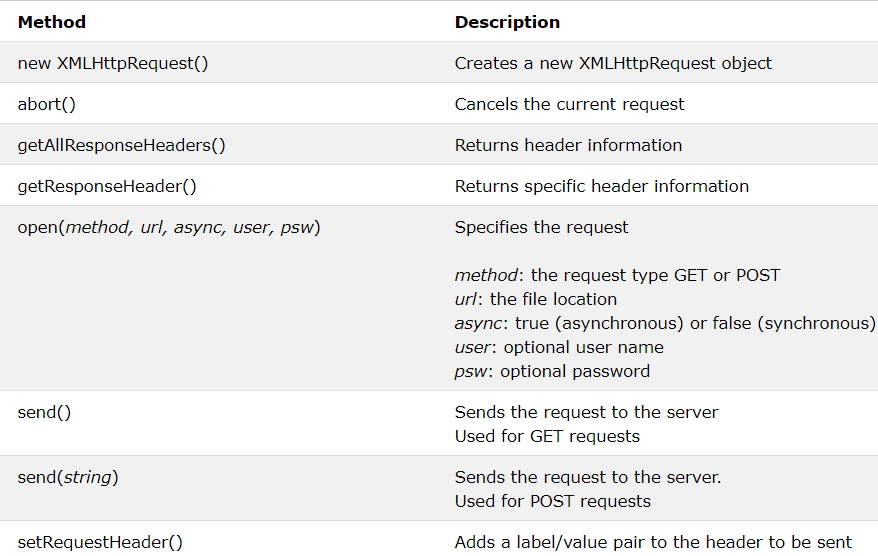
L'oggetto ***XMLHttpRequest*** può essere utilizzato per scambiare dati con un server Web dietro le quinte. Ciò significa che è possibile aggiornare parti di una pagina Web senza ricaricare l'intera pagina.

Sintassi per la creazione di un oggetto ***XMLHttpRequest***: var xhttp = new XMLHttpRequest();

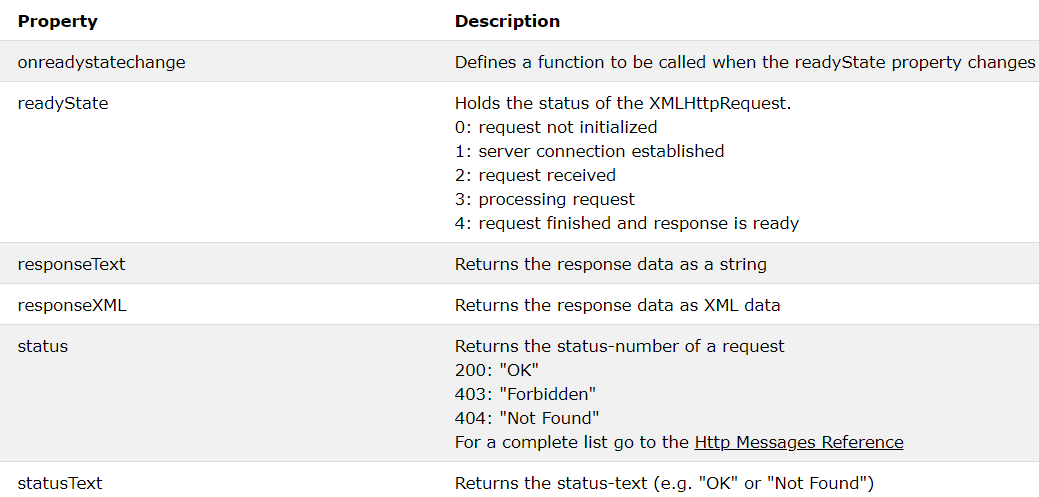
Per motivi di sicurezza, i browser moderni non consentono l'accesso ai domini.

Ciò significa che sia la pagina Web che il file XML che tenta di caricare devono trovarsi sullo stesso server.

***Metodi oggetto XMLHttpRequest:***



***Proprietà dell'oggetto XMLHttpRequest:***



**AJAX Request -** [**https://www.w3schools.com/js/js\_ajax\_http\_send.asp**](https://www.w3schools.com/js/js_ajax_http_send.asp)

Per inviare una richiesta a un server, utilizziamo i metodi ***open()*** e ***send()*** ***XMLHttpRequest*** dell'oggetto:

***Col GET:***

xhttp.open("GET", "ajax\_info.txt", true);  
xhttp.send();

Potresti ottenere un risultato memorizzato nella cache. Per evitare ciò, aggiungi un ID univoco all'URL:

xhttp.open("GET", "demo\_get.asp?t=" + Math.random(), true);  
xhttp.send();

Se si desidera inviare informazioni con il metodo ***GET***, aggiungere le informazioni all'URL:

xhttp.open("GET", "demo\_get2.asp?fname=Henry&lname=Ford", true);  
xhttp.send();

***Col POST:***

xhttp.open("POST", "demo\_post.asp", true);  
xhttp.send();

Per i dati POST come un modulo HTML, aggiungere un'intestazione HTTP con ***setRequestHeader().*** Specificare i dati che si desidera inviare nel metodo ***send()***:

xhttp.open("POST", "ajax\_test.asp", true);  
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");  
xhttp.send("fname=Henry&lname=Ford");

***L’URL:***

Il parametro ***url*** del metodo ***open()*** è un indirizzo di un file su un server. Il file può essere qualsiasi tipo di file, come .txt e .xml, o file di scripting del server come .asp e .php (che possono eseguire azioni sul server prima di inviare la risposta).

***Asincrono:***

Le richieste del server devono essere inviate in modo asincrono. Il parametro ***async*** del metodo ***open()*** deve essere impostato su ***true***: xhttp.open("GET", "ajax\_test.asp", true);

Inviando in modo asincrono, il JavaScript non deve attendere la risposta del server, ma può invece:

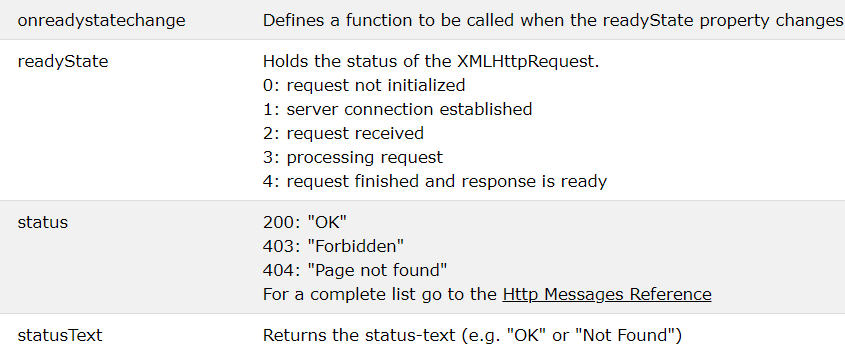
* eseguire altri script mentre si attende la risposta del server
* affrontare la risposta dopo che la risposta è pronta

**AJAX Response -** [**https://www.w3schools.com/js/js\_ajax\_http\_response.asp**](https://www.w3schools.com/js/js_ajax_http_response.asp)

La proprietà ***readyState*** contiene lo stato di ***XMLHttpRequest***.

*La proprietà* ***onreadystatechange*** *definisce una funzione da eseguire quando il* ***readyState*** *cambia, eseguendo una eventuale funzione di callback.*

La proprietà ***status*** e la proprietà ***statusText*** mantengono lo stato dell'oggetto ***XMLHttpRequest***.



Quando ***readyState*** è 4 e lo stato è 200, la risposta è pronta:

L' evento onreadystatechange viene attivato quattro volte (1-4), una volta per ogni modifica nel readyState.

***Utilizzo di una funzione di richiamata:***

Una funzione di ***callback*** è una funzione passata come parametro ad un'altra funzione.

Se si dispone di più di un'attività AJAX in un sito Web, è necessario creare una funzione per l'esecuzione ***XMLHttpRequest*** dell'oggetto e una funzione di ***callback*** per ciascuna attività AJAX.

*La chiamata di funzione dovrebbe contenere l'URL e quale funzione chiamare quando la risposta è pronta.*

loadDoc("url-1", myFunction1);

function loadDoc(url, cFunction) {  
  var xhttp;  
  xhttp = new XMLHttpRequest();  
  xhttp.onreadystatechange = function() {  
    if (this.readyState == 4 && this.status == 200) {  
      cFunction(this);  
    }  
  };  
  xhttp.open("GET", url, true);  
  xhttp.send();  
}  
function myFunction1(xhttp) {  
  // action goes here  
}  <https://www.w3schools.com/js/tryit.asp?filename=tryjs_ajax_callback>

La proprietà ***responseText*** restituisce la risposta del server come una stringa JavaScript e può essere utilizzata di conseguenza: document.getElementById("demo").innerHTML = xhttp.responseText;

Il metodo ***getAllResponseHeaders()*** restituisce tutte le informazioni di intestazione dalla risposta del server.

var xhttp = new XMLHttpRequest();  
xhttp.onreadystatechange = function() {  
  if (this.readyState == 4 && this.status == 200) {  
    document.getElementById("demo").innerHTML =  
    this.getAllResponseHeaders();  
  }  
}; <https://www.w3schools.com/js/tryit.asp?filename=tryjs_ajax_header>

Il metodo ***getResponseHeader()*** restituisce informazioni specifiche sull'intestazione dalla risposta del server.

var xhttp = new XMLHttpRequest();  
xhttp.onreadystatechange = function() {  
  if (this.readyState == 4 && this.status == 200) {  
    document.getElementById("demo").innerHTML =  
    this.getResponseHeader("Last-Modified");  
  }  
};  
xhttp.open("GET", "ajax\_info.txt", true);  
xhttp.send(); <https://www.w3schools.com/js/tryit.asp?filename=tryjs_ajax_lastmodified>

