



Ajax



Com'era una volta?



Click su invio






Com'è oggi?



🔍 Roger Fe

✕ 🗣️

-  Roger Federer
Tennista
- 🔍 roger federer news
- 🔍 roger federer patrimonio
- 🔍 roger federer ritiro
- 🔍 roger federer nazionalità
-  Roger Federer è esistito davvero
-  Roger Fenton
Fotografo
- 🔍 roger federer età
- 🔍 roger federer logo
- 🔍 roger federer instagram

Cerca con Google Mi sento fortunato

Segnala previsioni inappropriate
[Ulteriori informazioni](#)



Com'è oggi?



- Aggiornamento automatico dei risultati suggeriti
- Senza aggiornare tutta la pagina
- Senza dover premere alcun invio

Com'è possibile?

Cos'è AJAX?

È possibile farlo grazie all'utilizzo di **AJAX**



Cos'è AJAX?

- Non è un linguaggio di programmazione
- Non è una libreria JavaScript
- È semplicemente una combinazione di tecnologie utilizzate per poter creare pagine web dinamiche

Cosa significa AJAX?

A - Asynchronous
J - JavaScript
A - And
X - XML



Cosa significa AJAX?

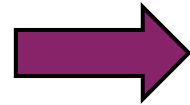
A - Asynchronous 

Asincrono, in quanto possiamo mandare una richiesta al server e intanto continuare a comunicare con l'utente, senza aspettare la risposta del server. Quando la risposta arriva, una specifica area dell'interfaccia utente verrà aggiornata (senza aggiornare l'intera pagina).



Cosa significa AJAX?

J - JavaScript

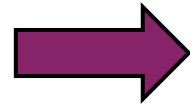


JavaScript, in quanto il linguaggio di programmazione che sta dietro al funzionamento di un'applicazione AJAX è proprio Javascript



Cosa significa AJAX?

A - And



E, con valore puramente
aggiuntivo, coordina elementi
della stessa frase

Cosa significa AJAX?

X - XML

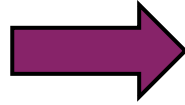


XML, eXtensible Markup Language, è un linguaggio di markup, utile alla codifica di documenti attraverso l'utilizzo di TAG (Proprio come HTML)



Cosa significa AJAX?

X - XML



XML, eXtensible Markup Language, è un linguaggio di markup, utile alla codifica di documenti attraverso l'utilizzo di TAG (Proprio come HTML)

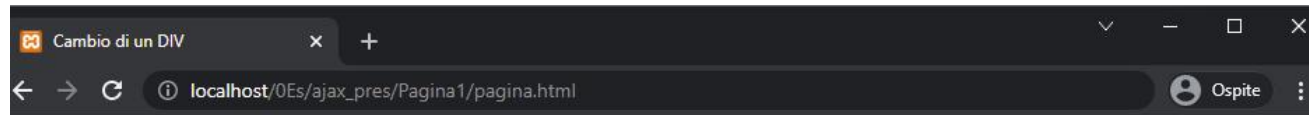
```
<?xml version="1.0"?>
<quiz>
  <qanda seq="1">
    <question>
      Who was the forty-second
      president of the U.S.A.?
    </question>
    <answer>
      William Jefferson Clinton
    </answer>
  </qanda>
  <!-- Note: We need to add
  more questions later.-->
</quiz>
```

XML

N.B.: Come suggerisce il sito W3C, AJAX è un nome di per sé ingannevole, in quanto non si deve per forza usare il formato XML per trasportare dati, ma è anzi comune trasportare i dati come semplice testo o con altri formati.

AJAX is a misleading name. AJAX applications might use XML to transport data, but it is equally common to transport data as plain text or JSON text.

Vediamo un esempio

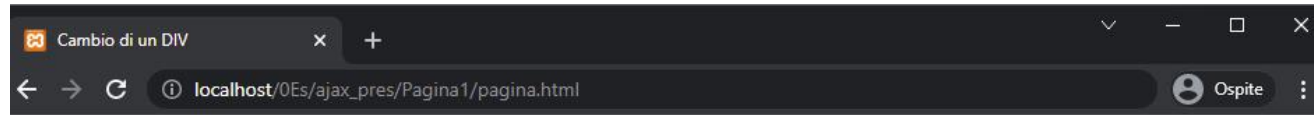


Prova AJAX



Name	Status	Type	Initiator	Size	Time	Waterfall	
pagina.html	200	document	Other	205 B	6 ms		
stile.css	304	stylesheet	pagina.html	204 B	2 ms		
funzione.js	200	script	pagina.html	(memory ca...)	0 ms		
gatto.jpg	200	jpeg	pagina.html	(memory ca...)	0 ms		
favicon.ico	200	x-icon	Other	(disk cache)	0 ms		

Vediamo un esempio



Prova AJAX



Name	Status	Type	Initiator	Size	Time	Waterfall	
pagina.html	200	document	Other	205 B	6 ms		
stile.css	304	stylesheet	pagina.html	204 B	2 ms		
funzione.js	200	script	pagina.html	(memory ca...)	0 ms		
gatto.jpg	200	jpeg	pagina.html	(memory ca...)	0 ms		
favicon.ico	200	x-icon	Other	(disk cache)	0 ms		
info.txt	200	xhr	funzione.js:10	502 B	3 ms		
ajax.png	200	png	funzione.js:6	77.9 kB	3 ms		

Codice di «pagina.html»

```
Pagina1 > <> pagina.html > ...
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="UTF-8">
5      <title>Cambio di un DIV</title>
6      <link href="stile.css" rel="stylesheet" type="text/css">
7
8      <script src="funzione.js"></script>
9    </head>
10
11   <body>
12     <h1>Prova AJAX</h1>
13
14     <div id="divCambio">
15       <h2>Titolo Iniziale</h2>
16       
17       <br>
18       <button onclick="cambio()">Clicca qua</button>
19     </div>
20   </body>
21 </html>
```

Niente di nuovo
nel codice della
pagina

Codice di «funzione.js»

```
Pagina1 > <> pagina.html > ...
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Cambio di un DIV</title>
6     <link href="stile.css" rel="stylesheet" type="text/css">
7
8     <script src="funzione.js"></script>
9   </head>
10
11   <body>
12     <h1>Prova AJAX</h1>
13
14     <div id="divCambio">
15       <h2>Titolo Iniziale</h2>
16       
17       <br>
18       <button onclick="cambio()">Clicca qua</button>
19     </div>
20   </body>
21 </html>
```

```
1 function cambio()
2 {
3   const xhttp = new XMLHttpRequest();
4
5   xhttp.onload = function () {
6     document.getElementById("divCambio").innerHTML = this.responseText;
7   }
8
9   xhttp.open("GET", "info.txt", true);
10  xhttp.send();
11 }
```

- ▶ È l'elemento fondamentale di AJAX
- ▶ È un oggetto che permette al codice JS Client-side di effettuare richieste HTTP, senza aggiornare tutta la pagina
- ▶ Ha vari metodi e proprietà, noi useremo principalmente i seguenti





Ritorniamo al codice

Creo una funzione che verrà eseguita all'attivarsi di "onload"

("onload" è come "onreadystatechange", ma che si esegue solo nel momento in cui il "readyState" è 4, quindi quando la richiesta è completata e la risposta spedita correttamente).

```
1 function cambio()  
2 {  
3     const xhttp = new XMLHttpRequest();  
4  
5     xhttp.onload = function () {  
6         document.getElementById("divCambio").innerHTML = this.responseText;  
7     }  
8  
9     xhttp.open("GET", "info.txt", true);  
10    xhttp.send();  
11 }
```

Creiamo e spediamo la richiesta al server

```
1 <h2>Cosa significa AJAX</h2>  
2   
3 <br>  
4 <b>A</b> - Asynchronous  
5 <br>  
6 <b>J</b> - Javascript  
7 <br>  
8 <b>A</b> - And  
9 <br>  
10 <b>X</b> - XML
```

Ritorniamo al codice

```
xhttp.onload = function () {  
    document.getElementById("divCambio").innerHTML = this.responseText;  
}
```

Quando la funzione è richiamata (ovvero quando readyState è a 4), andiamo a prendere l'elemento con id="divCambio" nella pagina, e cambiamo l'html contenuto con il testo contenuto nel file di risposta.

```
Pagina1 > info.txt  
1 <h2>Cosa significa AJAX</h2>  
2   
3 <br>  
4 <b>A</b> - Asynchronous  
5 <br>  
6 <b>J</b> - Javascript  
7 <br>  
8 <b>A</b> - And  
9 <br>  
10 <b>X</b> - XML
```



```
1 function cambio()  
2 {  
3     const xhttp = new XMLHttpRequest();  
4  
5     xhttp.onload = function () {  
6         document.getElementById("divCambio").innerHTML = this.responseText;  
7     }  
8  
9     xhttp.open("GET", "info.txt", true);  
10    xhttp.send();  
11 }
```

```
Pagina1 > pagina.html > ...  
1 <!DOCTYPE html>  
2 <html>  
3 <head>  
4     <meta charset="UTF-8">  
5     <title>Cambio di un DIV</title>  
6     <link href="stile.css" rel="stylesheet" type="text/css">  
7  
8     <script src="funzione.js"></script>  
9 </head>  
10  
11 <body>  
12 <h1>Prova AJAX</h1>  
  
    <div id="divCambio">  
        <h2>Titolo Iniziale</h2>  
          
        <br>  
        <button onclick="cambio()">Clicca qua</button>  
    </div>  
  
</body>  
</html>
```

Un esempio più completo



```
Pagina2 > <> pagina.html > ...
1  <!DOCTYPE html>
2  <html>
3
4      <head>
5          <meta charset="UTF-8">
6          <title>Cambio di un DIV</title>
7          <link href="stile.css" rel="stylesheet" type="text/css">
8
9          <script src="funzione.js"></script>
10     </head>
11
12     <body>
13         <h1>Seconda prova AJAX</h1>
14
15         <div>
16             <h2>Inserisci un cognome della classe</h2>
17             <input type="text" id="input" onkeyup="mostraSuggerimento(this.value)">
18         </div>
19
20         <div>
21             <h2>Testo in aggiornamento</h2>
22             <p id="parCambio"></p>
23         </div>
24     </body>
25
26 </html>
```

Link alla pagina

Html e JS come vediamo sono pressoché identici a prima

Notiamo però che la richiesta richiede una pagina php, mandando a essa un dato in GET

```
Pagina2 > JS funzione.js > mostraSuggerimento
1  function mostraSuggerimento(stringa)
2  {
3      if (stringa.length == 0) //se la lunghezza della stringa è pari a 0, mettiamo "" (vuoto) nel paragrafo da cambiare
4      {
5          document.getElementById("parCambio").innerHTML = "";
6      }
7      else
8      {
9          const xhttp = new XMLHttpRequest();
10         xhttp.onload = function () {
11             document.getElementById("parCambio").innerHTML = this.responseText;
12         }
13
14         xhttp.open("GET", "suggerimento.php?sugg="+stringa, true);
15         xhttp.send();
16     }
17 }
18
```



Cosa ci sta dietro?

[Link alla pagina](#)

"suggerimento.php?sugg="+stringa,

```
1 <?php
2 $cognomi = array();
3 $cognomi = caricaVett("cognomi.csv"); //Funzione per caricare il vettore con i cognomi presenti nel file
4
5 if(isset($_GET["sugg"]))
6 {
7     $str=$_GET["sugg"]; //Prendiamo la stringa che è stata passata
8     $ritorno=""; //Questa sarà la variabile che restituirò come risposta alla richiesta
9
10    if($str != "") //Controllo che la stringa passata non sia vuota
11    {
12        $str = strtolower($str); //Imposto tutta la stringa passata in lowercase (Es: Anna --> anna)
13        $dim = strlen($str); //Salvo la lunghezza della stringa passata
14
15        foreach($cognomi as $cognome)
16        {
17            //La funzione seguente mi restituisce la substring di cognome, dalla posizione 0 fino alla posizione $dim, che sarebbe la dimensione della stringa passata
18            $substring = substr($cognome, 0, $dim);
19
20            if(strpos($str, $substring)) //Questa funzione serve a cercare se nella prima stringa passata come parametro è contenuta la seconda
21            {
22                $cognome = ucfirst($cognome); //Metto la maiuscola al cognome da inserire
23                if($ritorno == "") //Se è il primo cognome da aggiungere, lo aggiungo e basta
24                {
25                    $ritorno = $cognome;
26                }
27                else //Altrimenti lo concateno a quelli già inseriti, separandoli con un trattino
28                {
29                    $ritorno = $ritorno." - ".$cognome;
30                }
31            }
32        }
33    }
34
35    if($ritorno=="")
36    {
37        echo "Nessun suggerimento trovato";
38    }
39    else
40    {
41        echo $ritorno;
42    }
43 }
44 else
45 {
46     echo "Errore";
47 }
```

```
function caricaVett($fileName)
{
    $vett = array();
    $file = fopen($fileName, "r");
    if($file)
    {
        while(!feof($file))
        {
            array_push($vett, fgets($file));
        }
    }
    fclose($file);

    return $vett;
}
```