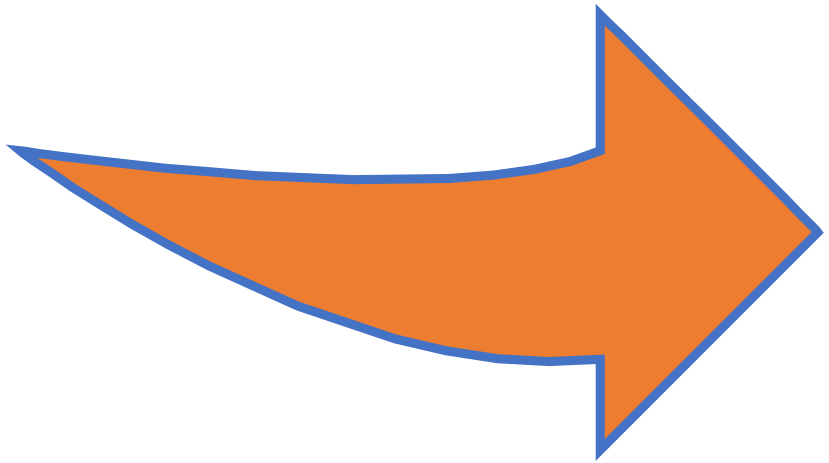
JSON

formato per lo scambio di dati tra le applicazioni



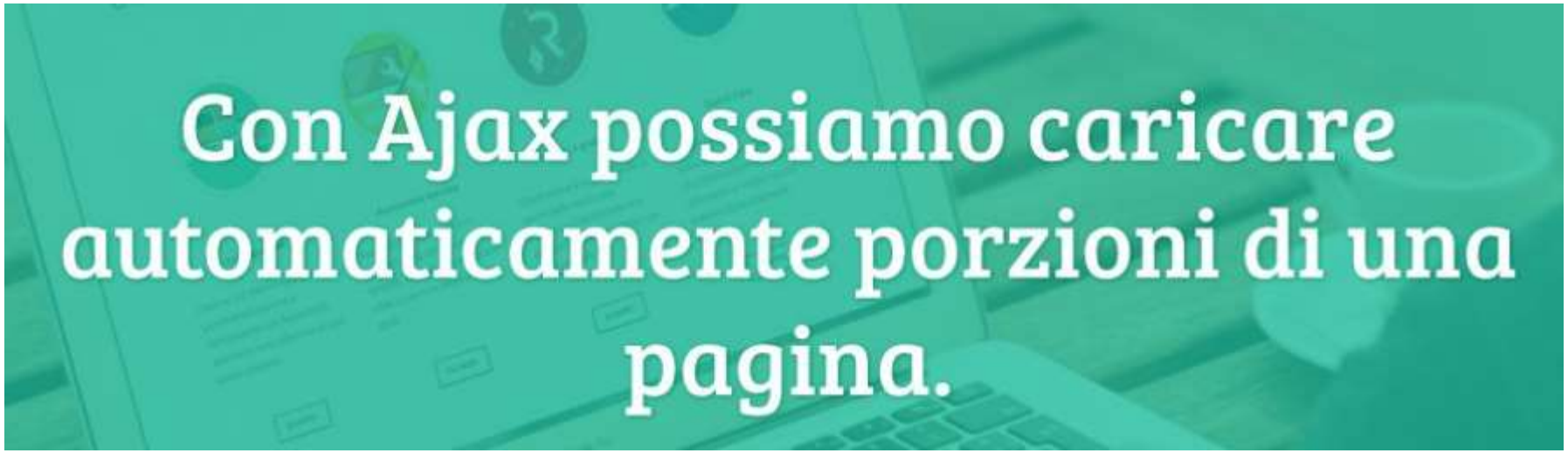
vs JSON: JavaScript Object Notation

Fino a poco tempo fa per utilizzare Ajax si faceva un largo uso di XML, ma oggi la tecnologia è migliore!

JSON è un **formato di interscambio di dati**, ed è spesso comparato con l'XML che però è un linguaggio di markup.

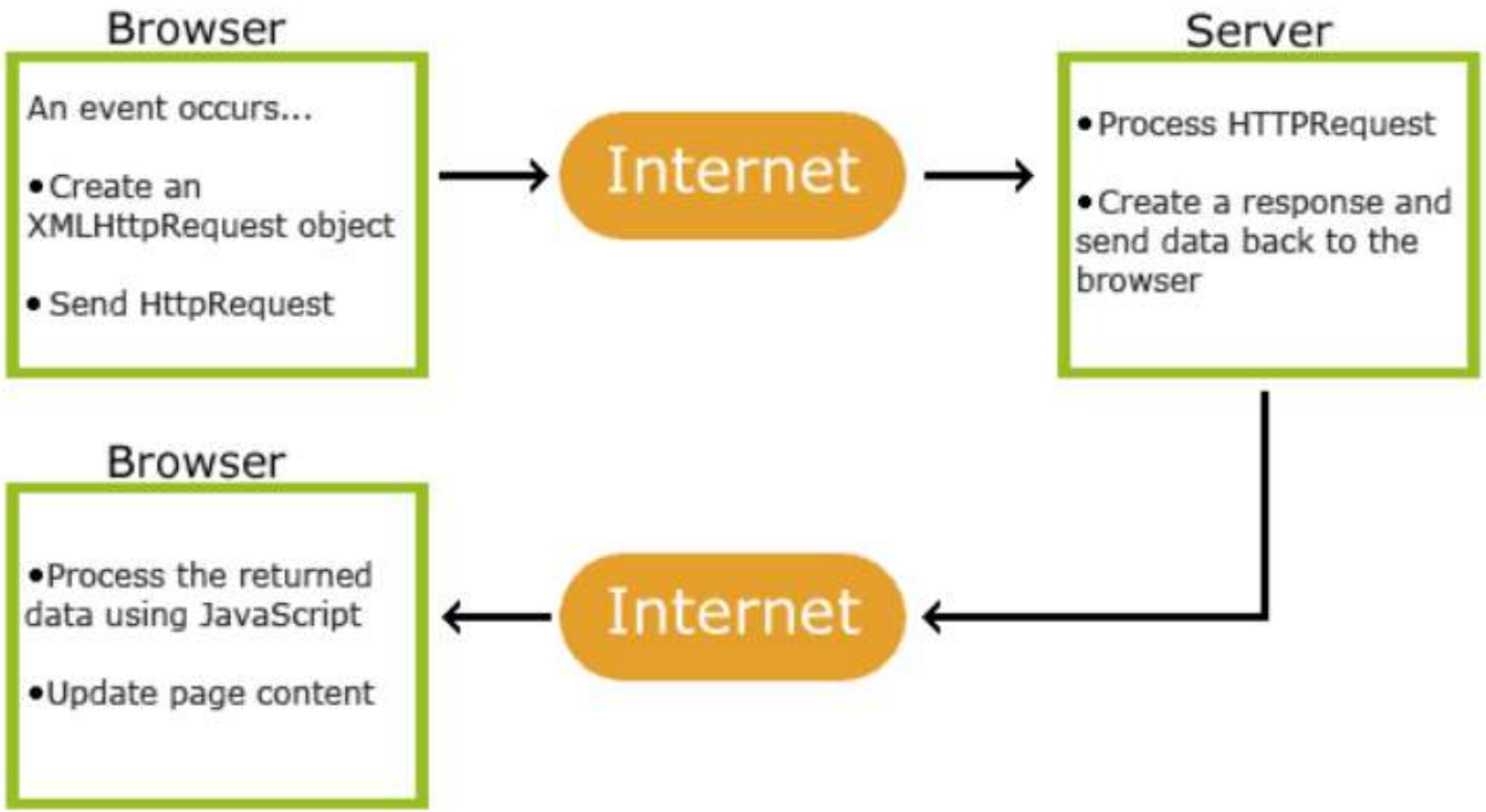
Entrambi non hanno un sistema di rappresentazione dei *dati binari*, per cui è compito del programmatore adottare convenzioni appropriate (es. Base64) per convertire i dati binari in **forma testuale**.

AJAX: Asynchronous JavaScript and XML



Tecnica di sviluppo software per la realizzazione di applicazioni web interattive (Rich Internet Application). Lo sviluppo di applicazioni HTML con 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.

Come lavora AJAX ?



JSON: introduzione

Praticamente ci fornisce una collezione di dati organizzati logicamente

JSON: due strutture

Basato su un sottoinsieme del linguaggio JavaScript, un file JSON può essere costituito solo da due strutture:

- Un **insieme di coppie nome/valore** (ad esempio un oggetto, una struct, una tabella hash, ecc.);

```
"name":"John"
```

- Un **elenco ordinato di valori** (come un array, un vettore, una lista, ecc.).

```
{
"employees":[ "John", "Anna", "Peter" ]
}
```

Oggetti JSON: il profilo di un utente

Gli oggetti JSON sono rappresentati da una serie non ordinata di nomi e valori ed iniziano con una parentesi graffa sinistra e finiscono con una parentesi graffa destra. La distinzione tra il nome e il valore di un oggetto avviene tramite la separazione tra i due valori con il simbolo dei due punti (*:*)

```
{
   "profile": {
      "username": "string",
      "firstName": "string",
      "lastName": "string",
      "gender": "string",
      "city": "string",
      "country": "string",
      "town": "string",
      "address": "string",
      "link": "string",
      "locale": "string",
      "image": "string",
      "imageUri": "string"
   }
```

Accesso ai valori di un oggetto: dot notation

```
myObj = { "name":"John", "age":30, "car":null };
x = myObj.name;
```

oppure

```
x = myObj["name"];
```

Per accedere a tutti i valori (*for-in*):

```
for (x in myObj) {
  document.getElementById("demo").innerHTML += x;
}
```

oppure

```
for (x in myObj) {
  document.getElementById("demo").innerHTML += myObj[x];
}
```

JSON e JavaScript: tipi di dati validi

JSON

```
{ "name":"John" }
```

Valid Data Types

- a string
- a number
- an object (JSON object)
- an array
- a boolean
- *null*

JSON values **cannot** be

- a function
- a date
- *undefined*

JavaScript

```
{ name:'John' }
```

ogni espressione valida compreso:

- a string
- a number
- an object
- an array
- a boolean
- null

- a function
- a date
- undefined

Invio al Server come oggetto JSON di dati immagazzinati in oggetto JS

JAVASCRIPT

JS JSON

JSON Intro

Sending Data

```
var myObj = {name: "John", age: 31, city: "New York"};
var myJSON = JSON.stringify(myObj);
window.location = "demo_json.php?x=" + myJSON;
```

JSON.stringify()

Conversione di un dato in formato JSON ricevuto dal Server, in oggetto JS

JAVASCRIPT

JS JSON

JSON Intro

Receiving Data

```
var myJSON = '{"name":"John", "age":31, "city":"New York"}';
var myObj = JSON.parse(myJSON);
document.getElementById("demo").innerHTML = myObj.name;
```

JSON.parse()

JSON: immagazzina dati in forma testuale

```
// Storing data:
myObj = {name: "John", age: 31, city: "New York"};
myJSON = JSON.stringify(myObj);
localStorage.setItem("testJSON", myJSON);

// Retrieving data:
text = localStorage.getItem("testJSON");
obj = JSON.parse(text);
document.getElementById("demo").innerHTML = obj.name;
```

Consultare dati in modo semplice

JavaScript

```
var andrea = {
    "eta" : 32,
    "citta" : "Livorno",
    "sesso" : "maschile"
}
```

Per esempio, se desidero inserire all'interno del mio documento HTML una stringa che dichiara la mia età potrei scrivere qualcosa del genere:

JavaScript

```
document.write( "Andrea ha " + andrea.eta + " anni." );
```

Rispettando completamente la sintassi di JavaScript sono stato in grado di prendere la proprietà `eta` dell'oggetto `andrea` che mi è stato passato attraverso JSON. Se vuoi notare tu stesso la differenza e la semplicità di questo tipo di notazione puoi confrontare le seguenti dichiarazioni.

JSON e APP

Grazie a JSON e alla tecnologia Ajax posso ottenere facilmente informazioni e usarle in una app.

permette di ottenere facilmente

informazioni e di utilizzarle all'interno di una nostra applicazione

(web, mobile o di qualsiasi altro tipo).

JSONP: JSON with Padding …

```
<script src="demo_jsonp.php">
```

File su SERVER: demo_jsonp.php

```
<?php
$myJSON = '{ "name":"John", "age":30, "city":"New York" }';

echo "myFunc(".$myJSON.");";
?>
```

Funzione lato CLIENT:

```
function myFunc(myObj) {
  document.getElementById("demo").innerHTML = myObj.name;
}
```

… pagina esempio con uso JSONP

```
<!DOCTYPE html>
<html>
<body>
<h2>Request JSON using the script tag</h2>
<p>The PHP file returns a call to a function that will handle the JSON data.</p>
<p id="demo"></p>
<script>
        function myFunc(myObj) {
          document.getElementById("demo").innerHTML = myObj.name;
        }
</script> <script src="demo_jsonp.php"></script>
</body>
</html>
```

JQuery: libreria di JavaScript

Sintassi:

$(selector).action()

```
$(document).ready(function(){
  $("p").click(function(){
    $(this).hide();
  });
});
```

Sintassi per usare funzioni:

```
$(document).ready(function(){
        // jQuery methods go here...
});
```

oppure in forma compatta:

```
$(function(){
        // jQuery methods go here...
});
```

Per inserire tale libreria (un file .js) nella pagina:

```
<head>
<script src="jquery-3.4.0.min.js"></script>
</head>
```

hide() metodo JQuery per nascondere l'elemento selezionato con sintassi **$(*selector*).hide(*speed,easing,callback*)**

jQuery CDN

Due modi per inserire Jquery in pagina web:

- Scaricare la libreria da JQuery.com
- Includerla in un CDN (Content Delivery Network) tipo Google

Google CDN: ***pagina di test per inserire in Google***

```
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js">
</script>
</head>
```

JQuery: metodo per eseguire una richiesta asincrona

Per verificare che tutto funzioni correttamente non ci resta che, una volta collegato jQuery alla nostra pagina, aggiungere il seguente blocco di codice:

$.ajax()
esegue una richiesta
AJAX asincrona
con scrittura alla console

HTML DOM
metodo **console.log()**

JavaScript

```
$.ajax({
    url: 'http://randomuser.me/api/',
    dataType: 'json',
    success: function(data){
        console.log(data);
    }
});
```

.... pagina di test

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"></script>
<script>
            $(document).ready(function(){
              $("button").click(function(){
                $.ajax({url: "demo_test.txt", success: function(result){
                  $("#div1").html(result);
                }});
              });
            });
</script>
</head>
<body>
<div id="div1"><h2>Let jQuery AJAX Change This Text</h2></div>
<button>Get External Content</button>
</body>
</html>
```

```
$("button").click(function(){
  $.ajax({url: "demo_test.txt", success: function(result){
    $("#div1").html(result);
  }});
});
```

JQuery: libreria di JavaScript

JQuery AJAX - tra i metodi utili load() per caricare dati da Server e inserirli in un elemento

Sintassi :

$(*selector*).load(*URL,data,callback*);

```
$("button").click(function(){
  $("#div1").load("demo_test.txt");
});
```

JQuery HTML/CSS - tra i metodi utili html() per cambiare o ritornare il contenuto di un elemento selezionato

Sintassi per ritornare:

$(*selector*).html();

o settare

$(*selector*).html(*content*);

o settore usando funzione

$(*selector*).html(function*(index,currentcontent)*)

```
$("button").click(function(){
  $("p").html("Hello <b>world</b>!");
});
```

Altri metodi per manipolare elementi HTML (DOM)

DOM = Document Object Model

The DOM defines a standard for accessing HTML and XML documents:

"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."

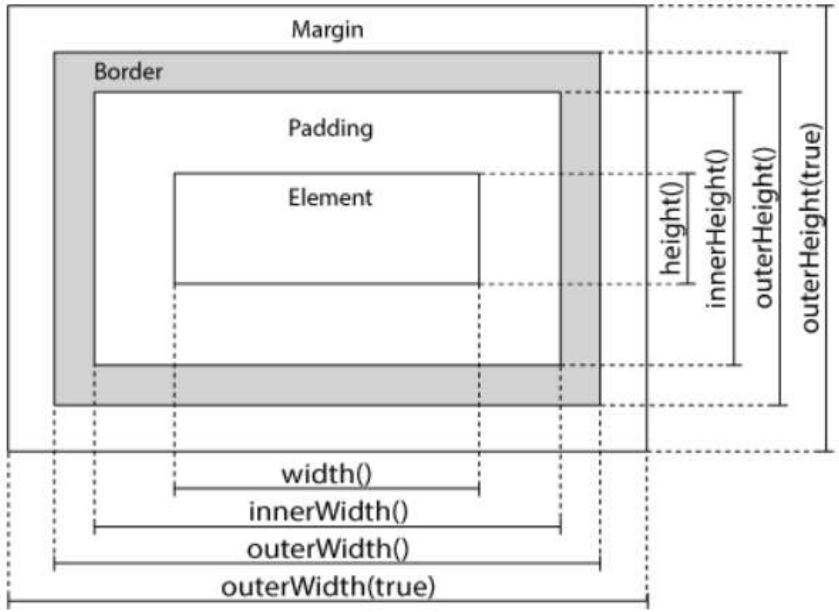
jQuery HTML

- jQuery Get
- jQuery Set
- jQuery Add
- jQuery Remove
- jQuery CSS Classes
- jQuery css()
- jQuery Dimensions

Get Content - **text(), html(), and val()**

Get Attributes - **attr()**

- `width()`
- `height()`
- `innerWidth()`
- `innerHeight()`
- `outerWidth()`
- `outerHeight()`



JQuery: altri effetti