

Gioco

credits: <http://www.lostdecadegames.com/how-to-make-a-simple-html5-canvas-game/>

Idea



HTML5 Canvas

- Contenitore per disegnare della grafica
 - cerchi, quadrati, linee, immagini
 - On the fly! (javascript)

HTML

```
<canvas id="myCanvas" width="200" height="100"> </canvas>
```

Disegnare un immagine in un canvas

```
var c = document.getElementById( "myCanvas" );
var ctx = c.getContext( "2d" );
var img = document.getElementById( "scream" );
ctx.drawImage( img, 10, 10 )
```

guida su come disegnare altri oggetti: http://www.w3schools.com/html/html5_canvas.asp

1. Creiamo uno **stato** di gioco:

1. Creiamo due oggetti: hero e monster
2. Settiamo le loro posizioni attuali rispetto al canvas, la loro velocità
3. Creiamo una variabile per contare quante volte l'eroe ha catturato il mostro

2. Creiamo un **main** con un ciclo infinito

1. nel main invochiamo una funzione "update" seguita da una funzione "render"

3. **Update**: spostiamo i due personaggi e vediamo se si sono catturati – aggiorniamo lo stato del gioco

1. Utilizziamo gli eventi "keydown" e "keyup" per capire se l'utente sta premendo una certa freccia

4. **Render**: Visualizziamo le immagini nel canvas in base allo stato del gioco

5. Creiamo una funzione **Reset** per inizializzare il gioco

1. Piazziamo hero e monster in due posizioni random

Polyfill

Problema

- HTML5 introduce gli elementi “article” che i browser vecchi non riconoscono.
- Come farglieli riconoscere senza modificare il browser?

Risposta

- Mettiamo un javascript che naviga il DOM, cerca tutti gli `article` e li modifica con elementi di tipo `div`
- Questo è un esempio di “polyfill”

Altri Esempi:

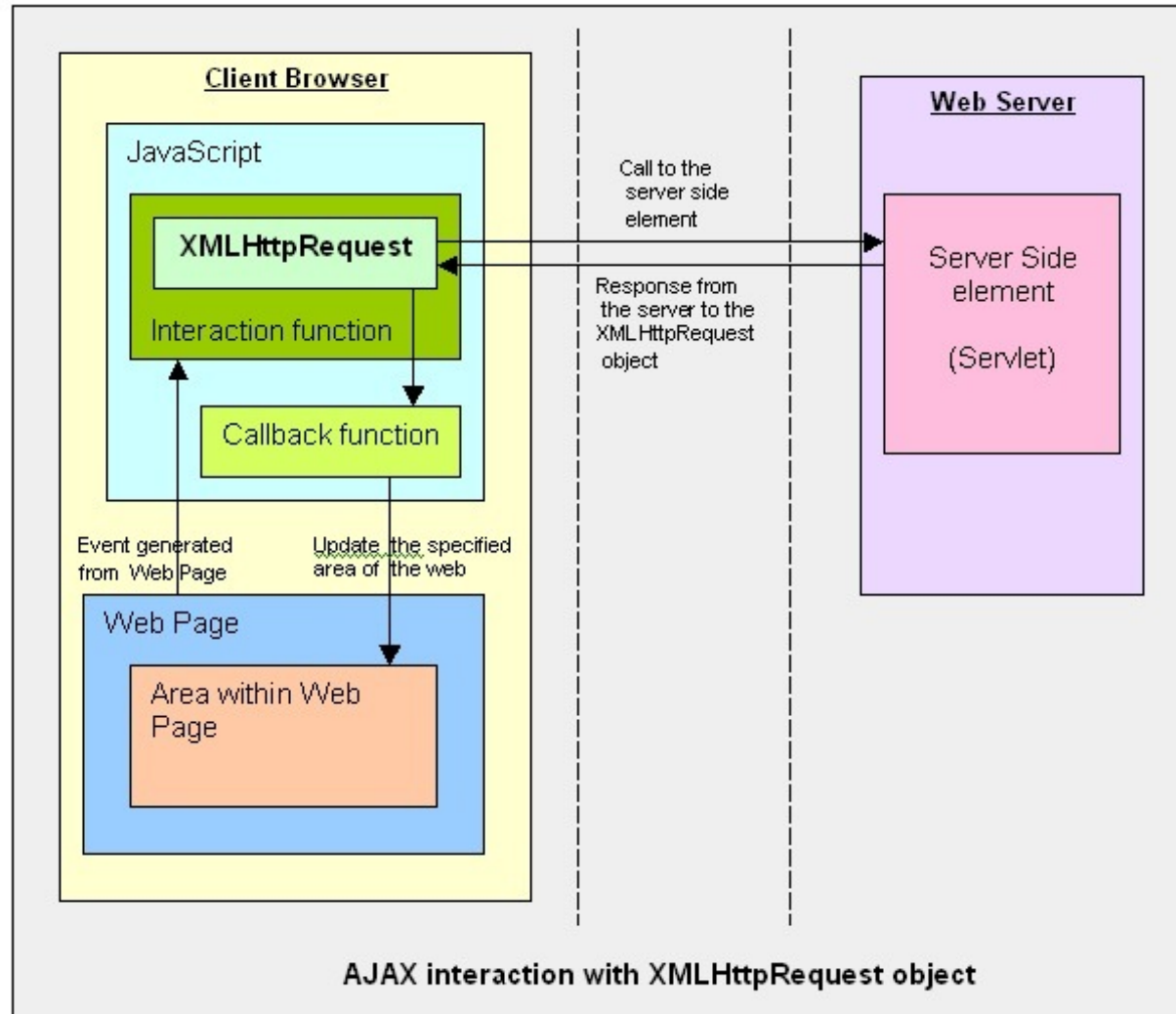
HTML5 polyfill: rende una pagina HTML5 compatibile con vecchi browser non HTML5

MODERNIZR: testa la presenza di features nel browser e carica polyfill se servono

SELECTIVIZR: fa capire a vecchie versioni di IE query CSS3 complesse

Comunicazione asincrona

Evitare il “full page refresh”



XMLHttpRequest

- Crea una richiesta web
- Metodi/attributi più utilizzati:
 - **open**('GET', 'http://www.uniroma2.it', false)
Il terzo parametro dice se la richiesta deve essere asincrona (ovvero se la funzione si deve bloccare finchè non ritorna). Se `async=true`, dobbiamo gestire manualmente utilizzando listener di eventi
 - **send()**
Invia la richiesta
 - **responseText**
Il test di risposta (DOMString)

Guida: <https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest>

Esempio XMLHttpRequest (sync)

```
function Get(yourUrl) {  
    var Httpreq = new XMLHttpRequest();  
    Httpreq.open("GET", yourUrl, false);  
    Httpreq.send(null);  
    return Httpreq.responseText;  
}
```

```
console.log( Get('http://  
www.example.org/prova.json') );
```

Esempio XMLHttpRequest (async)

```
var xhr= new XMLHttpRequest();
xhr.onreadystatechange = function() {
    if (xhr.readyState==4 &&
    xhr.status==200) {
        //facciamo delle operazioni
        // per esempio leggiamo la
        xhr.responseText
    }
}
xhr.open( "GET",yourUrl , true);
```

onreadystatechange

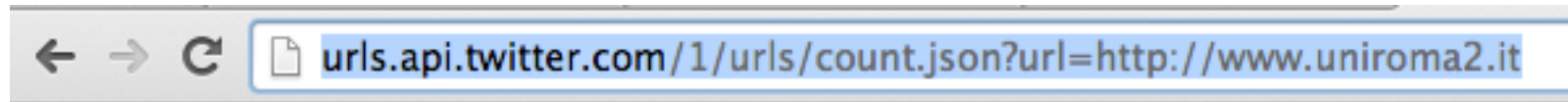
readyState	
0	request not initialized
1	server connection established
2	request received
3	processing request
4	request finished and response is ready

JSON

- Voglio inviare/richiedere dati (senza markup!)
 - Ad esempio voglio la lista degli esami ma non in formato HTML
 - Non voglio mostrare le informazioni all'utente così come mi vengono date, ma definire dei comportamenti (“se ci sono più di 0 esami, mostro qualcosa, altrimenti mostro la lista gli esami in una MIA tabella”)
- JSON: Javascript object notation
 - Formato popolare per la comunicazione client-server
 - Ad esempio: [{name: 'Programmazione Web'}, { name: 'Fondamenti di Informatica'}]

Esempio JSON

<http://urls.api.twitter.com/1/urls/count.json?url=http://www.uniroma2.it>



```
{"count":8,"url":"http://www.uniroma2.it/"}
```

Ritorna il numero di volte che un certo URL e' presente in twitter

<http://www.bbc.co.uk/radio1/playlist.json>
(playlist BBC)

<https://itunes.apple.com/search?term=beyonce&entity=musicVideo>
Video itunes beyonce

JSON in Javascript

```
function Get(yourUrl) {  
    var Httpreq = new XMLHttpRequest();  
    Httpreq.open("GET", yourUrl, false);  
    Httpreq.send(null);  
    return Httpreq.responseText;  
}
```

```
var json_obj = JSON.parse(Get(yourUrl));  
console.log("this is the author name: "+  
    json_obj.author_name);
```


Same origin policy

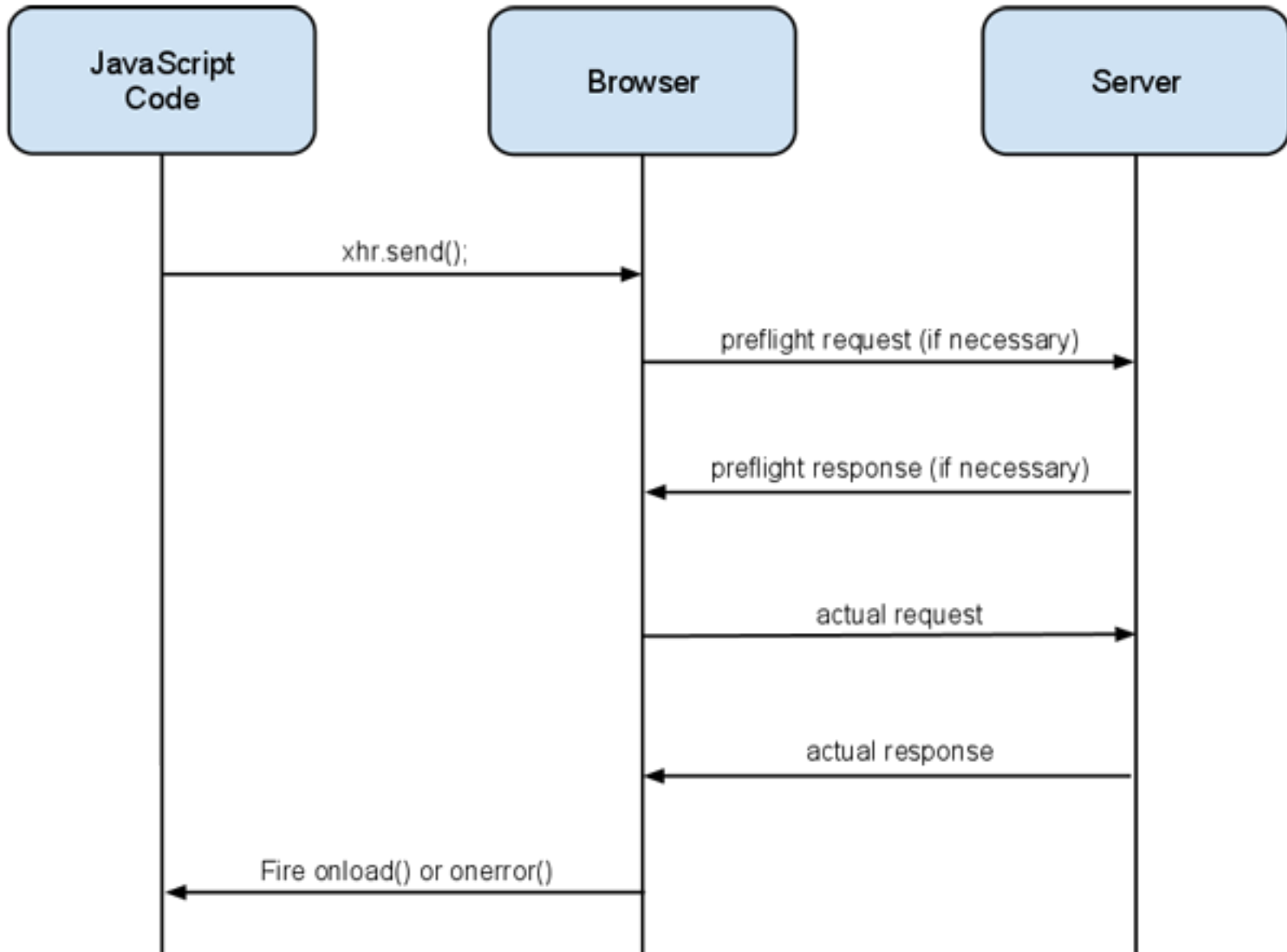
- Un browser permette agli script contenuti in una pagina web di accedere ai dati contenuti in un'altra risorsa web (altra pagina web, json ecc) solo se entrambe le pagine hanno la stessa origine

Compared URL	Outcome	Reason
http://www.example.com/dir/page.html	Success	Same protocol and host
http://www.example.com/dir2/other.html	Success	Same protocol and host
http://www.example.com:81/dir/other.html	Failure	Same protocol and host but different port
https://www.example.com/dir/other.html	Failure	Different protocol
http://en.example.com/dir/other.html	Failure	Different host
http://example.com/dir/other.html	Failure	Different host (exact match required)
http://v2.www.example.com/dir/other.html	Failure	Different host (exact match required)

✖ ▶ XMLHttpRequest cannot load json.html:8
<http://urls.api.twitter.com/1/urls/count.json?url=http://www.uniroma2.it>. No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'null' is therefore not allowed access.

Cross Origin Resource Sharing

- Come fare se vogliamo espressamente permettere il resource sharing tra due siti diversi? → CORS
 - Standard W3C per condividere risorse tra domini diversi
 - Prevede richiesta di autorizzazione (client) e autorizzazione (server)
- Viene implementato inviando degli header HTTP in req/resp
 - Nel caso di richieste GET c'è la classica richiesta/risposta
 - Nel caso di richieste più complesse (PUT/DELETE) viene inviata una richiesta di tipo OPTIONS prima della richiesta vera (preflight request)



Simple CORS requests (GET)

```
var url = 'http://api.alice.com/cors';  
var xhr = createCORSRequest('GET', url);  
xhr.send();
```

Access-Control-Allow-Origin: http://api.bob.com
Access-Control-Allow-Credentials: true
Access-Control-Expose-Headers: FooBar
Content-Type: text/html; charset=utf-8

GET /cors HTTP/1.1
Origin: http://api.bob.com
Host: api.alice.com
Accept-Language: en-US
Connection: keep-alive
User-Agent: Mozilla/5.0...

Client

Server

Complex CORS requests (PUT)

OPTIONS /cors HTTP/1.1

Origin: http://api.bob.com

Access-Control-Request-Method: PUT

Host: api.alice.com

...

Access-Control-Allow-Origin: http://api.bob.com

Access-Control-Allow-Methods: GET, POST, PUT

Content-Type: text/html; charset=utf-8

Preflight request/response

PUT /cors HTTP/1.1

Origin: http://api.bob.com

Host: api.alice.com

X-Custom-Header: value

Access-Control-Allow-Origin: http://api.bob.com

Content-Type: text/html; charset=utf-8

Client

Server

Esempio

- Creiamo una pagina per votare e aggiungere delle citazioni famose
 - vediamo anche il debugger js

- 5/6 basi linguaggio
 - 8/6 basi linguaggio
 - 12/6 basi linguaggio + gioco
 - 15/6 esempio
-
- Lunedì' venerdì' dalle 14 alle 16 aula B3