

Javascript



Impariamo Javascript
con Memory



Giulio Angiani
I.I.S. "Blaise Pascal" - Reggio Emilia



What's Javascript?

2/36

Cos'è Javascript

- Linguaggio per il WEB
- Linguaggio di scripting
- Gestione ad eventi
- General Purpose
- Nasce lato client (browser)
- Numerose librerie e frameworks
- Lato server (NodeJs)



Codice Javascript

- Immerso nel codice HTML (poco!)
- All'interno dei tag (meglio!)
- In file esterni .js (molto meglio!)

```
<a href="#" onClick='fname()'>lancia la funzione</a>
```

HTML

```
<script>
    function fname() {
        alert('una funzione!');
    }

    alert("metto qui il codice Js!");
</script>
```

```
<script type="text/javascript" src="path-to-javascript-file.js"></script>
```

4/36

Perché un gioco

- Apprendimento divertente
- Step-by-step
- Introduce concetti a diversi livelli
- User-friendly



Memory

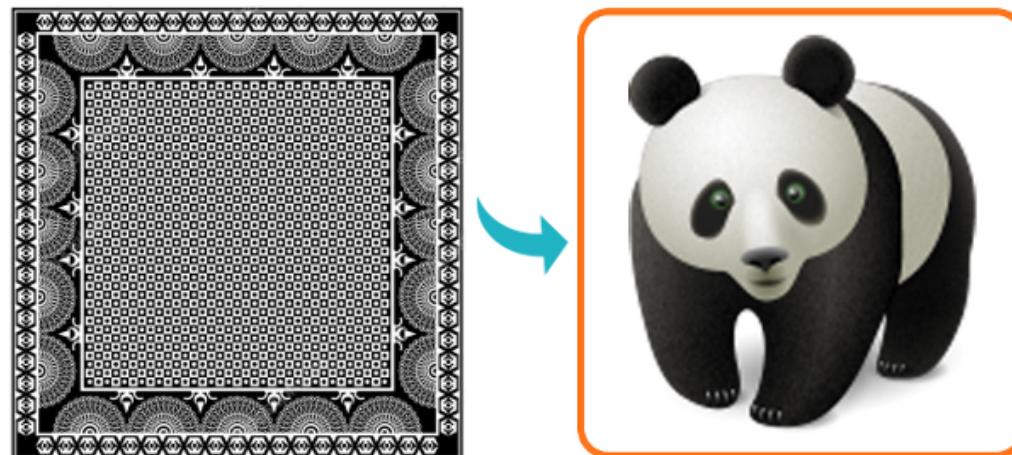
- Gioco 2D
- Regole molto semplici
- Da nessuna animazione a effetti grafici divertenti



6/36

Iniziamo...

- Si gira una carta cliccandoci sopra



Struttura

- Incastriamo l'immagine in una griglia HTML
- Associamo un ID all'immagine

```
<div id='title'>Memory! </div>  
<div id='main'>  
    <table id='contenitore'>  
        <tr>  
            <td>  
                <img class='cards' id='card00' src='images/retro.jpg'>  
            </td>  
        </tr>  
    </table>  
</div>
```

HTML

Gestione degli eventi

- Evento per girare la carta
- nel tag IMG aggancio all'evento **onClick** la chiamata alla funzione **giraLaCarta()**

```
<img class='cards' id='card00' src='images/retro.jpg' onClick='giraLaCarta()'>
```

HTML

```
function giraLaCarta() {  
    // cambia l'attributo src dell'immagine  
    var c = document.getElementById('card00');  
    c.src = 'images/panda.png';  
}
```

JAVASCRIPT

Esercizio

- Cliccando varie volte la carta si gira più volte



Gestione degli eventi

- Modifico l'evento
- Se src vale "panda" metto "retro" altrimenti metto "panda"

```
<img class='cards' id='card00' src='images/retro.jpg' onClick='giraLaCarta()'>
```

HTML

```
function giraLaCarta() {  
    var c = document.getElementById('card00');  
    if (c.src.indexOf('images/panda.png') > -1) { // la stringa è presente  
        c.src = 'images/retro.jpg';  
    }  
    else {  
        c.src = 'images/panda.png';  
    }  
}
```

JAVASCRIPT

Refactoring

- Non legare MAI la logica applicativa ai valori degli attributi HTML
- Usa le variabili interne
- in JS le variabili possono essere dichiarate (ma anche no!!)
 - se usi `var` stai definendo una variabile *locale*
 - se non lo sue stai definendo una variabile *globale*



Refactoring

```
var chiusa = true;
var img_chiusa = "images/retro.jpg";
var img_aperta = "images/panda.png";
```

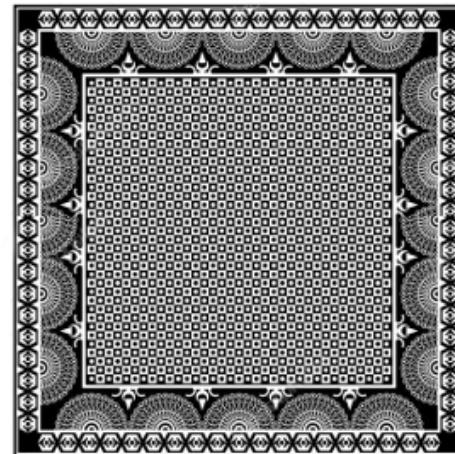
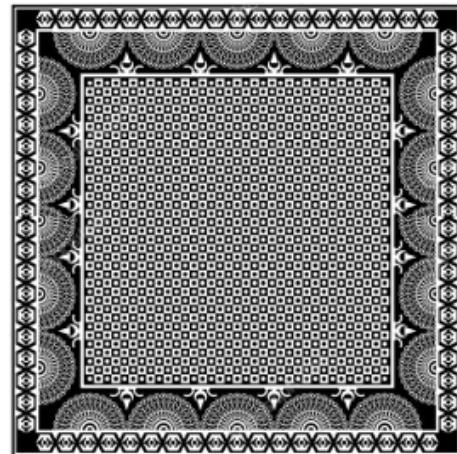
JAVASCRIPT

```
function giraLaCarta() {
    var c = document.getElementById('card00');
    if (!chiusa) {
        c.src = img_chiusa;
        chiusa = true;
    }
    else {
        c.src = img_aperta;
        chiusa = false;
    }
}
```

13/36

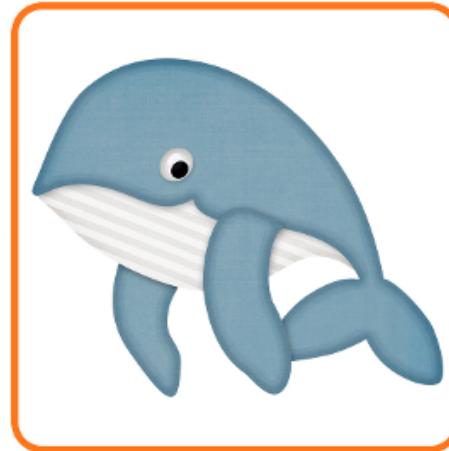
Esercizio

- Proviamo con due carte (un panda e una balena)



Esercizio

- Proviamo con due carte (un panda e una balena)



Facile!! Copincolla!!!!

```
<table id='contenitore'>  
  <tr>  
    <td>  
      <img class='cards' id='card00' src='images/retro.jpg' onClick='giraLaCarta1()'>  
      <img class='cards' id='card01' src='images/retro.jpg' onClick='giraLaCarta2()'>  
    </td>  
  </tr>  
</table>
```

HTML

16/36

Facile!! Copincolla!!!!

```
var chiusal = true;
var img_chiusal = "images/retro.jpg";
var img_apertal = "images/panda.png";

function giraLaCarta1() {
    var c = document.getElementById('card00');
    if (!chiusal) {
        c.src = img_chiusal;
        chiusal = true;
    }
    else {
        c.src = img_apertal;
        chiusal = false;
    }
}
```

JAVASCRIPT

17/36

Facile!! Copincolla!!!!

```
var chiusa2 = true;
var img_chiusa2 = "images/retro.jpg";
var img_aperta2 = "images/balena.png";

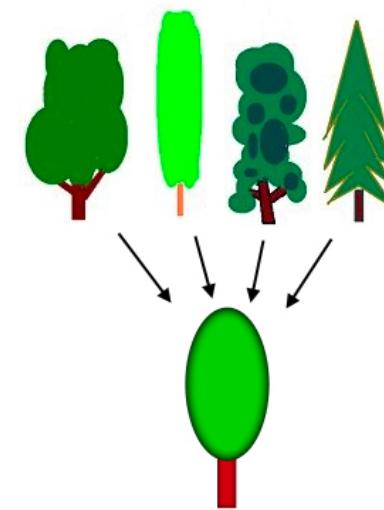
function giraLaCarta2() {
    var c = document.getElementById('card01');
    if (!chiusa2) {
        c.src = img_chiusa2;
        chiusa2 = true;
    }
    else {
        c.src = img_aperta2;
        chiusa2 = false;
    }
}
```

JAVASCRIPT

18/36

Generalizziamo

- Nel caso 1 immagine:
 - un elemento HTML (tipo IMG) col suo ID
 - una variabile booleana
 - una stringa col nome del file immagine
 - non abbiamo bisogno di passare parametri all'**unica** funzione
- Nel caso N immagini:
 - un elemento HTML (tipo IMG col proprio ID) per ogni immagine
 - un **array** di variabili booleane
 - un **array** di stringhe coi nomi dei file immagine
 - abbiamo bisogno di **passare parametri** alla funzione



Generalizziamo

Per la parte HTML

```
<table id='contenitore'>  
    <tr>  
        <td>  
            <img class='cards' id='card00' src='images/retro.jpg' onClick='giraLaCarta(this.id)'>  
            <img class='cards' id='card01' src='images/retro.jpg' onClick='giraLaCarta(this.id)'>  
        </td>  
    </tr>  
</table>
```

HTML

Funzioni in Javascript

```
function nomefunzione(param1, param2, ...) {  
}
```

JAVASCRIPT

- esempio:

```
function myFunction(p1, p2) {  
    return p1 * p2;    // The function returns the product of p1 and p2  
}
```

JAVASCRIPT

- non si dichiara il *tipo di ritorno*
- può restituire un valore (è una *funzione*)
- può non restituire niente (è una *procedura*)
- i nomi di funzioni possono contenere lettere, numeri, underscores e il simbolo \$

Funzioni in Javascript

- modifichiamo la funzione che gestisce l'evento

```
function giraLaCarta(object_id) {  
    // object_id contiene l'ID dell'oggetto HTML  
    // sul quale ho scatenato l'evento Click  
    var c = document.getElementById(object_id);  
}
```

JAVASCRIPT

- devo scegliere cosa fare in funzione dell'ID selezionato

Funzioni in Javascript

```
var chiusa0 = true;
var chiusal = true;
var img_chiusa = "images/retro.jpg";
var img_aperta0 = "images/panda.png";
var img_apertal = "images/balena.png";
function giraLaCarta(object_id) {
    var c = document.getElementById(object_id);
    if (object_id == 'card00') {
        if (chiusa0) c.src=img_aperta0;
        else c.src=img_chiusa;
        chiusa0 = !chiusa0;
    }
    if (object_id == 'card01') {
        if (chiusal) c.src=img_apertal;
        else c.src=img_chiusa;
        chiusal = !chiusal;
    }
}
```

JAVASCRIPT



© 1960 Peanuts Worldwide LLC. Dist. by Universal Uclick

Esercizio

- Refactoring del codice precedente per eliminare gli IF...
- suggerimenti:
 - due variabili booleane? => Array()
 - due immagini ? => Array()
 - ID card00 è elemento 0, ID card01 è elemento 1
 - String.replace
 - parseInt
- esempio:

```
var s = "javascript";
s = s.replace("java", "my");
console.log(s)
```

JAVASCRIPT

myscript

OUTPUT

Esercizio

- Refactoring del codice precedente per eliminare gli IF...
- suggerimenti:
 - due variabili booleane? => Array()
 - due immagini ? => Array()
 - ID card00 è elemento 0, ID card01 è elemento 1
 - String.replace
 - parseInt
- esempio:

```
var s = "1234";
var n = parseInt(s);
console.log(s)
```

JAVASCRIPT

1234

OUTPUT

25/36

Array in Javascript

- Array come collezione di oggetti qualsiasi
- Non omogeneo
- Ordinato

```
var v = Array(); // array vuoto                                JAVASCRIPT
var v = Array(1,2,'tre',4.5); // array con elementi disomogenei
var v = Array( 1,2, Array('tre', 'quattro'), 5.6); // array a due livelli

v.push(10) // procedura inserisce in coda
var last = v.pop() // stacca e restituisce l'ultimo elemento
v.length // lunghezza dell'array
v.sort() // procedura di ordinamento
v.indexOf(x) // restituisce la posizione di x in v se esiste, altrimenti -1
v.map(fn) // restituisce un nuovo array che è l'applicazione di fn a tutti gli elementi di v
```

[apri test](#)

26/36

Array in Javascript

- Esempio

```
function inc(x) {  
    return x+1;  
}  
  
var v = Array(1,2,4.5); // array con elementi disomogenei  
var v2 = v.map(inc);  
console.log(v2);
```

JAVASCRIPT

Array(3) [2, 3, 5.5]

OUTPUT

Memory - gestione dei click

- Problema
 - giro la prima carta => aspetto
 - giro la seconda carta => controllo

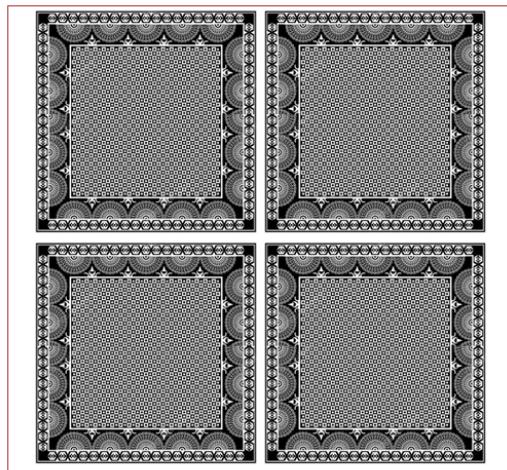
```
// due variabili inizializzate a null
var first = null;
var second = null;

...
function controllaCarte(idcartacliicata) {
    ...
}
```

JAVASCRIPT

Memory - 4 carte

Situazione corrente



Memory - 4 carte

Cliccando su prima e terza... ma cliccando su prima e seconda...



Memory - 4 carte

- struttura HTML

```
<table id='contenitore'>                                              HTML
  <tr>
    <td>
      <img class='cards' id='card00' src='images/retro.jpg' onClick='giraLaCarta(this.id)'>
      <img class='cards' id='card01' src='images/retro.jpg' onClick='giraLaCarta(this.id)'>
    </td>
  </tr>
  <tr>
    <td>
      <img class='cards' id='card02' src='images/retro.jpg' onClick='giraLaCarta(this.id)'>
      <img class='cards' id='card03' src='images/retro.jpg' onClick='giraLaCarta(this.id)'>
    </td>
  </tr>
</table>
```

Memory - 4 carte

- strutture dati JS

```
var chiuse = Array(true, true, true, true);  
var images_path = "images/";  
var img_chiusa = images_path + "retro.jpg";  
var images = ["panda.png", "balena.png", "balena.png", "panda.png"];  
  
var visibili = Array(true, true, true, true);  
var first = null;  
var second = null;
```

JAVASCRIPT

- Facciamo sparire le carte indovinate
 - aggiungiamo un array di boolean per sapere quali sono da visualizzare e quali no

Memory - 4 carte

- Chiamo la funzione **controllaCarte(n)** ad ogni click

```
function controllaCarte(n) {  
    if (first == null) first = n;  
    else if (second == null) second = n;  
    if (first != null && second != null) {  
        if (images[first] == images[second]) {  
            // segno nell'array "visibili" le carte da rendere invisibili  
            visibili[first] = visibili[second] = false;  
        }  
        else {  
            alert("DIVERSE")  
            // richiudi le schede  
            chiuse[first] = chiuse[second] = true;  
        }  
        first=second=null;  
    }  
    mostraCarte();  
}
```

JAVASCRIPT

Memory - 4 carte

- Chiamo sempre la funzione **mostraCarte()**
 - gestisce l'interfaccia verso l'utente

```
function mostraCarte() {  
    // gestisce l'interfaccia  
    for (i = 0; i < 4; i++) {  
        var object_id = "card0" + i;  
        var c = document.getElementById(object_id);  
        if (!visibili[i]) c.style.visibility = 'hidden';  
        else c.style.visibility = 'visible';  
  
        if (chiuse[i]) c.src = img_chiusa  
        else c.src = images_path + images[i];  
    }  
  
    controllaFinePartita(); // controlla se la partita è finita  
}
```

JAVASCRIPT

Memory - 4 carte

- Aggiungo anche una funzione che controlla se ho finito o no

```
// controlla fine partita
function controllaFinePartita() {
    var fine = true;
    for (i = 0; i < 4; i++) {
        if (visibili[i]) fine = false;
    }
    if (fine) {
        document.getElementById('main').innerHTML = 'HAI FINITO!';
    }
}
```

JAVASCRIPT



Giulio Angiani
I.I.S. "Blaise Pascal" - Reggio Emilia