



UN TOPO SFUGGENTE

CoderDOJO su Javascript - Trento



Augusto Ciuffoletti

26 maggio 2015

Un topo nella pagina

- Il tuo browser (quello che usi per navigare) può leggere un file del tuo computer, e mostrarti disegni e parole nella sua finestra
- Devi descriverli usando il linguaggio che capisce lui: HTML
- La novità è che puoi scrivere anche programmi che cambiano il contenuto della finestra!
- Puoi disegnarci dentro un topo e strizzarlo, rimpicciolirlo, farlo scappare e anche scomparire!
- Cominciamo dal principio: una pagina con un topo (o un fantasma, un cane, un pipistrello, scegli tu)
- Fatti consegnare il file l1.zip e estrai il contenuto: ricordati dove lo metti!

Il file che descrive la pagina

- Nella stessa directory, apri un file con notepad o gedit, e scrivici dentro queste istruzioni per il tuo browser

```
<html>
<body>
  
</body>
</html>
```

Il file topo.html

- Il file lo puoi chiamare come vuoi, ma deve finire con .html
 - noi lo abbiamo chiamato topo.html
- Ora vai sul tuo browser (Firefox, Explorer, Chrome) e apri il file topo.html
 - se non sai come si fa chiedilo a un mentor

Un nome alle cose

- Quelle paroline tra `<` `>` si chiamano **tag** e servono a spiegare al browser quello che deve fare
- Ad esempio, tutto quello che sta tra `<body>` e `</body>` è il *corpo* della nostra pagina
- Quello che sta dopo il tag `img` dice al browser che deve mostrare il disegno che sta nel file `topo.png`

Ora fai da solo:

- Se conosci HTML, prova ad aggiungere un testo e un titolo
- Anche se non conosci HTML, cambia il file e fai mostrare un altro dei personaggi nella directory

Inventiamo un nome per le cose

- Diamo un **nome** al tag che contiene il disegno
 - attento: non un nome al disegno (quello c'è già), ma un nome per il tag che contiene il disegno
 - noi lo chiamiamo `ilTopo`, **id** sta per *identificatore*
 - **id** è una **proprietà**, ce ne sono tante e usiamo quelle che ci servono
- Poi diciamo al browser con che **stile** rappresentarlo
 - `position:fixed` è una parolina magica... non te la posso spiegare

```

```

modifica il tag `img` nel tuo file HTML

- Per il nostro tag abbiamo definito tre **proprietà**: `src`, `id` e `style`.
- Controlla con il tuo browser: è cambiato qualcosa?

Le cose possono cambiare

- Non è cambiato nulla ma **ora** possiamo cominciare a dare qualcosa da fare al nostro browser
- Scriviamo un programma che, quando viene **eseguito**, cambia le dimensioni del topo
- Lo mettiamo in fondo alla nostra pagina. Così:

```
...  
<script>  
function StrizzaTopo(event){  
  document.getElementById("ilTopo").style.width=80;  
}  
</script>  
</body>  
</html>
```

Il programma che strizza il topo

- I programmi si chiamano **script** e sono scritti nel linguaggio **JavaScript**

Le cose cambiano (basta toccarle)

- Abbiamo scritto il programma, ora dobbiamo dirgli **quando** eseguirlo
- Vogliamo dirgli di eseguirlo quando clicchiamo sul disegno del topo
- Per farlo mettiamo un'altra **proprietà** nel tag che contiene il nostro disegno
- La nuova proprietà è **onclick**, e significa che quando si clicca sulla rappresentazione di quel tag, si chiama la funzione *StrizzaTopo*

```

```

Modifica il tag `img`

Le cose cambiano (basta sfiorarle)

- La **proprietà** `onclick` definisce un **evento**, il click sul tasto del mouse
- Il **valore** della proprietà indica quale funzione deve essere eseguita quando accade quell'evento
 - il nome della funzione possiamo inventarcela, i nomi degli eventi no
- Esistono tanti altri eventi: prova `onmouseover`, `onmouseenter`, `onmouseleave`, `ondblclick`
- Che differenza c'è tra `onmousedown` e `onmouseup`?

Facciamo ordine

- Di solito gli script si mettono in un file a parte
- Tagliamo il nostro script e incolliamolo in un file che finisce con .js

```
function cambiaTopo(event){  
  document.getElementById("ilTopo").style.width=80;  
}
```

Il file scriptDelTopo.js

- Nel file HTML mettiamo una proprietà `src` nel tag `script` con il nome del file con lo script

```
...  
<script src="scriptDelTopo.js">  
</script>  
</body>  
</html>
```

Riepilogo (i due file)

```
<html>
<body>
  
  <script src="scriptDelTopo.js"></script>
</body>
</html>
```

Il file topo.html

```
function StrizzaTopo(event){
  document.getElementById("ilTopo").style.width=80;
}
function Scompare(event){
  event.target.style.visibility = "hidden";
}
```

CHALLENGE!

Metti nella tua pagina anche una immagine del bottone Ora prova a fare in modo che il topo sia strizzato quando premi il bottone!

Il topo scappa lontano

- Ora proviamo a zoomare il topo ogni volta che lo tocco
 - se prima è largo 200 poi sarà 100 e poi 50 e poi...
- Uso una **variabile** e ci metto dentro quanto è largo
- Poi divido per due la variabile
- Il topo sembrerà scappare via lontano

```
function StrizzaTopo(event){  
  var largo=document.getElementById("ilTopo").width;  
  document.getElementById("ilTopo").style.width=largo/2;  
}
```

Il file scriptDelTopo.js

Risparmiamo le dita

- A proposito di variabili: ne uso un'altra per metterci il **tag** del topo
 - ...così il programma è più corto e si capisce meglio

```
function StrizzaTopo(event){  
  var topo=document.getElementById("ilTopo");  
  var largo=topo.width;  
  topo.style.width=largo/2;  
}
```

Il file scriptDelTopo.js

- Controlla che funzioni ancora...

Il topo scappa di lato

- Il topo può scappare se modifichi la sua posizione
- Per usare la posizione devi usare il **metodo** che si chiama `offsetLeft`
- La **proprietà** che usi per impostare la posizione si chiama `left`
 - ...anche se noi non l'abbiamo ancora usata esiste lo stesso

```
function scappaTopo(event){  
  var topo=document.getElementById("ilTopo");  
  topo.style.left=topo.offsetLeft+200;  
}
```

Il file `scriptDelTopo.js`

- **Attento:** il nome della funzione è cambiato!

Scappa a zig-zag

- Il topo cambia senso di fuga ogni volta che lo tocchi
- Quindi:
 - Se sta andando a destra, andrà poi a sinistra
 - Se sta andando a sinistra, andrà poi a destra
- Il topo non fa **sempre** la stessa cosa: si ricorda quello che ha fatto prima e **decide** che cosa fare
- Si dice che osserva una **condizione**, e poi decide
- La condizione è *Stavo andando a destra?*
 - Se è vera, poi vado a sinistra
 - Altrimenti, poi vado a destra

Se (cioè *if*)

- In Javascript possiamo scrivere **if**, e il computer ci capisce
 - `if (stavoAndando == destra)` vado a sinistra
- Per dirgli altrimenti scriviamo **else**
 - `else` vado a destra
- Scriviamolo bene...

```
stavoAndando="destra";  
function ScappaTopo(event){  
  var topo=document.getElementById("ilTopo");  
  if ( stavoAndando=="destra" ) {  
    topo.style.left=topo.offsetLeft+200;  
  }  
  else {  
    topo.style.left=topo.offsetLeft-200;  
  }  
}
```

Il file scriptDelTopo.js

Non funziona...

- Cosa c'è che non va?

CHALLENGE

- Cosa devi aggiungere al programma per fare andare il topo prima a destra, e poi a sinistra?

- scriviamo nella variabile tutte le volte da che parte andiamo!

```
var stavoAndando="destra";  
function ScappaTopo(event){  
  var topo=document.getElementById("ilTopo");  
  if ( stavoAndando=="destra" ) {  
    topo.style.left=topo.offsetLeft+200;  
    stavoAndando="sinistra"; }  
  else {  
    topo.style.left=topo.offsetLeft-200;  
    stavoAndando="destra"; }  
}
```

Il file scriptDelTopo.js