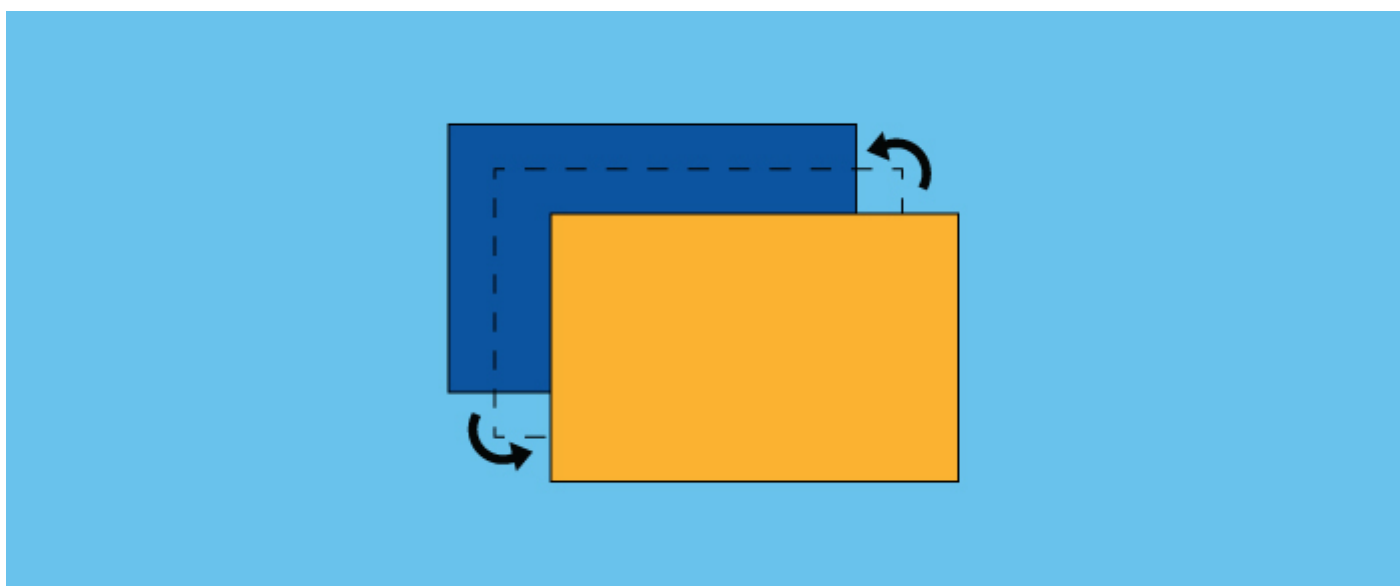


Aggiungere un effetto rotazione alle cards di Bootstrap

lunedì, 23 luglio 2018

Diciamoci la verità, a volte basta davvero poco per dare risalto ad una pagina o ad un sito web. Ma ciò che molti ignorano o tralasciano è che farlo è molto semplice!

Ed ecco come lasciare i tuoi utenti a bocca aperta con pochissime righe di CSS e Javascript, trasformando un'anonimo box in un bellissimo contenitore a due facce, visibili con un effetto animato di rotazione, detto anche **flip effect**. Curioso?



Sommario

Prepariamo il terreno

Passo 1: HTML

Passo 2: CSS

Passo 3: jQuery

Prepariamo il terreno

Per mostrarti come aggiungere l'effetto rotazione animato, utilizzeremo uno dei componenti più diffusi di **Bootstrap**: le cards.

Daremo quindi per scontato che tu conosca già **Bootstrap**, la sua implementazione ed i suoi componenti, ma in caso contrario, ti consiglio vivamente di scoprirne le numerose potenzialità prima di iniziare.

[Framework Bootstrap](#)

Passo 1: HTML

Come consuetudine, partiamo dalla parte più semplice, il markup HTML. Il concetto alla base del progetto è avere due "facciate", proprio come se avessimo tra le mani un foglio di carta. Costruisci quindi il contenitore principale, che conterrà i due lati:

```
<div class="rotate-container">
  [â€¦] CONTENUTO [â€¦]
</div>
```

e successivamente all'interno del contenitore astratto, posiziona i contenitori delle due facciate:

```
<div class="card card-front">
  <div class="card-body">
    <div class="h5 card-title">Card title</div>
    <div class="h6 card-subtitle mb-2 text-muted">Card subtitle</div>
    <p class="card-text">Some quick example text to build on the card title and make
up the bulk of the card's content.</p>
    <button class="btn btn-secondary btn-rotate">Rotate me!</button>
  </div>
</div>
<div class="card card-back">
  <div class="card-body">
    <div class="h5 card-title">Card title</div>
    <div class="h6 card-subtitle mb-2 text-muted">Card subtitle</div>
    <p class="card-text">Some quick example text to build on the card title and make
up the bulk of the card's content.</p>
    <button class="btn btn-dark btn-rotate">Go back</button>
  </div>
</div>
```

come potrai notare, rispetto alle classi di default di Bootstrap, sono state aggiunte le classi:

- **card-front**: per indicare il lato frontale della card
- **card-back**: per indicare il lato posteriore della card
- **btn-rotate**: utilizzato come trigger della rotazione

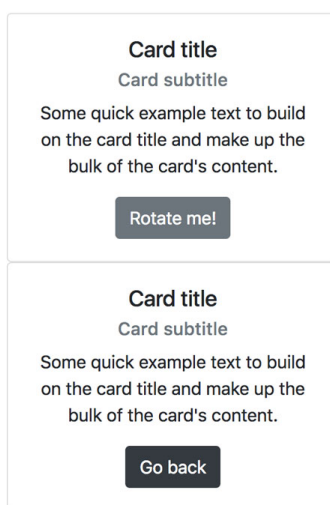
Passo 2: CSS

Inizia a dare un po' di stile adesso!

La prima cosa da fare, è impostare un paio di semplici regole CSS per rendere l'animazione perfettamente fluida:

```
.rotate-container { position: relative; }  
.rotate-container .card-front .btn-rotate,  
.rotate-container .card-back .btn-rotate { position: static; z-index: 0; }
```

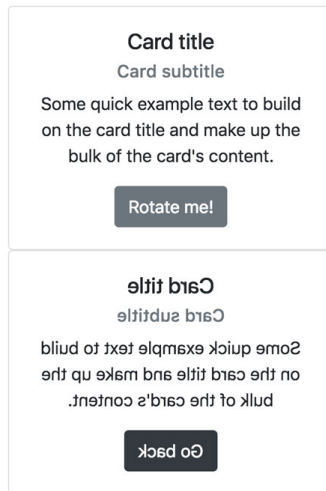
Al momento però l'unico risultato è avere due semplici <div> l'uno sull'altro.



Ma lavorando un po' con la proprietà perspective di CSS3 inizierai a vedere qualcosa di più:

```
.rotate-container .card-front {  
  -webkit-transform: perspective(600px) rotateY(0deg);  
  transform: perspective(600px) rotateY(0deg);  
}  
.rotate-container .card-back {  
  -webkit-transform: perspective(1600px) rotateY(180deg);  
  transform: perspective(1600px) rotateY(180deg);  
}
```

Ecco il risultato:



Ma cosa è successo? Abbiamo utilizzato la proprietà `perspective` di CSS3 per posizionare gli elementi in 3D in una prospettiva spaziale, in poche parole, è come se avessi "diviso" le due facce di un foglio di carta e le avessi messe una sotto l'altra, ruotando la seconda di 180°.

Ora, per ricostruire l'oggetto, è necessario che le due facciate si sovrappongano. Per farlo, è sufficiente aggiungere alla regola CSS già creata le ulteriori direttive (evidenziate):

```
.rotate-container .card-back {  
  -webkit-transform: perspective(1600px) rotateY(180deg);  
  transform: perspective(1600px) rotateY(180deg);  
  position: absolute;  
  top: 0;  
  left: 0;  
  right: 0;  
}
```

ed il gioco è fatto!



Arrivati a questa fase, hai creato i due lati della card, le hai unite in un unico oggetto con due facce (come un foglio) e dato loro la giusta prospettiva spaziale 3D.

Ti starai chiedendo: ma perché vedo la parte posteriore anziché quella frontale? Giusta osservazione!

Bisogna impostare il foglio di stile affinché "comprenda" che la visibilità dei due lati è alternativa: se è attivo il contenitore **card-front** la parte posteriore deve essere nascosta, se è attivo il contenitore **card-back** la parte anteriore deve essere nascosta. Ma come fare? È presto detto:

```
.rotate-container .card-front,  
.rotate-container .card-back {  
  width: 100%;  
  -webkit-backface-visibility: hidden;  
  backface-visibility: hidden;  
  transition: all 0.5s linear 0s;  
}
```

La proprietà **backface-visibility** definisce se la parte posteriore di un elemento deve essere visibile o meno quando si trova di fronte all'utente, nel caso in esame è stata impostata a hidden, in poche parole, abbiamo nascosto la parte posteriore così che non sia visibile all'utente.

Impostando questa regola su entrambe le facciate della nostra card, ti sei assicurato che l'altra parte non venga visualizzata.

Inoltre, è stato aggiunto ad entrambi i lati un effetto di transizione lineare, per prepararsi all'animazione.

Ora non resta che gestire la rotazione. Partiamo da un ragionamento geometrico (ahinoi!): ruotare l'oggetto che hai appena costruito, equivale a ruotare di 180° entrambe le facciate sull'asse delle

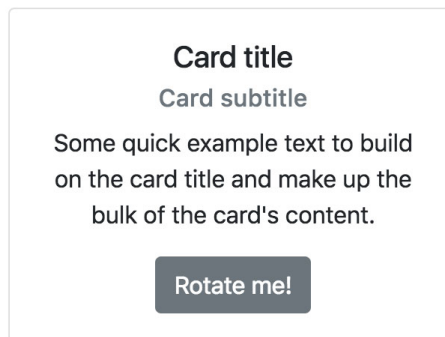
ordinate (Y). Già sembra complicato così, figuriamoci scriverlo in CSS!

Invece no, è piuttosto semplice:

```
.rotate-container .card-front.rotate {  
  -webkit-transform: perspective(1600px) rotateY(-180deg);  
  transform: perspective(1600px) rotateY(-180deg);  
}  
.rotate-container .card-back.rotate {  
  -webkit-transform: perspective(1600px) rotateY(0deg);  
  transform: perspective(1600px) rotateY(0deg);  
}
```

Come fatto in precedenza, è stata utilizzata la proprietà `perspective` di CSS3, modificando però i parametri di posizionamento spaziale: la parte frontale (che partiva da una rotazione nulla, 0°) è stata modificata a -180° (il valore decrescente indica una rotazione verso sx), mentre la parte posteriore (che partiva da una rotazione di 180°) è stata modificata a 0° (il valore crescente indica una rotazione verso dx).

In pratica, adesso le due facce si trovano in posizione opposta e speculare rispetto a prima!



Passo 3: jQuery

Ma come passare da un lato all'altro? I più attenti avranno notato che le regole che modificano la prospettiva dei due lati, fanno riferimento ad una classe CSS nuova, di cui non si era ancora parlato: **.rotate**.

Ebbene sì, questa è la classe che utilizzerai per attivare l'animazione!

Per farlo, è sufficiente un piccolo frammento di codice Javascript, basato su jQuery:

```
( function( $ ) {  
  $( '.btn-rotate' ).click( function( e ) {  
    e.preventDefault();  
    $( this ).parents( '.rotate-container' ).find( '.card-front, .card-back'  
  ).toggleClass( 'rotate' );  
  } );  
} )( jQuery );
```

In pratica, cliccando su un elemento che ha la classe btn-rotate (precedentemente discussa), lo script aggiunge (se non presente) oppure elimina (se presente) la classe rotate dagli elementi con classe card-front e card-back appartenenti allo stesso oggetto con classe rotate-container.

In questo modo, vengono applicate le regole CSS che modificano il valore prospettico delle facce dell'oggetto, con l'effetto di transizione precedentemente applicato:

transition: all 0.5s linear 0s;

[Guarda la demo su JSFiddle](#)