

Creare un menu off-canvas con CSS3

venerdì, 13 gennaio 2017

L'off-canvas è una tipologia di menu di navigazione in cui l'elemento HTML che contiene il menu stesso è posto al di fuori dell'area visibile (viewport) e viene visualizzata all'utente su sua esplicita azione di richiesta, ad esempio, cliccando su un pulsante. Questa tecnica è spesso utilizzata nel mobile, dove l'ottimizzazione degli spazi è una priorità, ma ultimamente si sta diffondendo molto anche su web app fruibili da dispositivi con risoluzioni desktop.



Sommario

Passo 1: HTML

Passo 2: CSS

Passo 3: Rendiamo tutto in movimento

Passo 1: HTML

La struttura del markup HTML è molto semplice. Per prima cosa, costruisci un pulsante, che diventerà il trigger (attivatore) per il nuovo menu off-canvas; per realizzarlo, però, ecco una tecnica un po' inusuale che ti permetterà di utilizzare solamente il CSS per attivare/disattivare la visualizzazione del menu:

```
<input type="checkbox" id="offcanvas-toggle" />
<label for="offcanvas-toggle">&equiv;</label>
```

Adesso, costruisci un semplice menu, fatto di qualche link demo:

```
<ul class="offcanvas">
  <li><a href="#">Lorem ipsum</a></li>
  <li><a href="#">Lorem ipsum</a></li>
  <li><a href="#">Lorem ipsum</a></li>
  <li><a href="#">Lorem ipsum</a></li>
  <li><a href="#">Lorem ipsum</a></li>
</ul>
```

In realtà, l'elemento `<input>` viene nascosto da CSS (come vedrai in seguito), ma è necessario che venga posizionato gerarchicamente in modo da avere lo stesso elemento padre rispetto al menu off-canvas. Il pulsante vero e proprio, è costituito dalla `<label>` dell'elemento `<input>` (associati tra loro tramite l'attributo `for=""` dell'elemento `<label>` che "punta" all'ID dell'elemento `<input>`).

Inoltre, è necessario che tutto il contenuto non off-canvas, sia inserito in un elemento separato, ad esempio:

```
<div class="wrapper">
  [â€¦] CONTENUTO PAGINA [â€¦]
</div>
```

Tenendo conto di questa piccola premessa, puoi creare un markup un po' più elaborato e completo, come ad esempio:

```
<input type="checkbox" id="offcanvas-toggle" />
<ul class="offcanvas">
  <li><a href="#">Lorem ipsum</a></li>
  <li><a href="#">Lorem ipsum</a></li>
  <li><a href="#">Lorem ipsum</a></li>
  <li><a href="#">Lorem ipsum</a></li>
  <li><a href="#">Lorem ipsum</a></li>
</ul>
<div class="wrapper">
```

```
<nav>
  <label for="offcanvas-toggle">&equiv;</label>
</nav>
<div class="container">
  [â€¦] CONTENUTO PAGINA [â€¦]
</div>
</div>
```

Passo 2: CSS

Ora non ti resta che dare un po' di stile, con particolare attenzione alle regole necessarie al funzionamento del menu di navigazione off-canvas.

Partendo dal menu off-canvas:

```
.offcanvas {  
  bottom: 0;  
  left: 0;  
  position: fixed;  
  top: 0;  
  width: 320px;  
  transform: translateX( -100% );  
}
```

In particolare, con la regola alla riga 7, "spostiamo" il menu al di fuori della viewport, così da renderlo non visibile in fase iniziale.

Per quanto riguarda il pulsante, invece:

```
#offcanvas-toggle { display: none; }  
#offcanvas-toggle:checked ~ .offcanvas { transform: translateX( 0 ); }  
#offcanvas-toggle:checked ~ .wrapper { transform: translateX( 320px ); }
```

Tre semplicissime righe, ma che sono la colonna portante di tutto il lavoro!

- Nella riga 1, viene nascosto l'elemento `<input>` (come anticipato in precedenza)
- Nella riga 2, viene specificato che, nel caso l'elemento `<input>` sia selezionato (checked), il menu off-canvas subisce uno "spostamento" orizzontale che lo faccia partire la pixel zero (quindi dall'estrema sinistra della viewport)
- Nella riga 3, come per la riga precedente, ma a subire lo spostamento è il contenitore principale, che viene spostato di tanti pixel quanto è la grandezza del menu off-canvas.

Da sottolineare, l'utilizzo del selettore `~` (chiamato "*general sibling*") che ha il significato di:

Seleziona gli elementi con classe "offcanvas" che sono successivi all'elemento con ID "offcanvas-toggle" che risulta selezionato. Inoltre, i due elementi devono avere lo stesso elemento padre

```

1 <!DOCTYPE html>
2 <html lang="it">
3 <head>
4 <meta charset="utf-8">
5 <title>demo</title>
6 <style type="text/css">
7
8     /* IL SELETTORE ~ SEPARA DUE ELEMENTI E SELEZIONA IL SECONDO ELEMENTO SOLO SE È PRECEDUTO DAL PRIMO ED ENTRAMBI CONDIVIDONO LO STESSO ELEMENTO PADRE. */
9     <strong ~ p { color: red; }
10
11 </style>
12 </head>
13 <body>
14 <div>
15 <div>
16 <div><strong>Lorem ipsum all the fact</div> <!-- Non verrà selezionato: ha lo stesso elemento padre di <strong> ma non è successivo a <strong> -->
17 <div><strong>Lorem ipsum all the fact</div> <!-- Non verrà selezionato: ha lo stesso elemento padre di <strong> ma non è successivo a <strong> -->
18 <div><strong>Lorem ipsum all the fact</strong> <!-- Verrà selezionato: ha lo stesso elemento padre di <strong> ed è successivo a <strong> -->
19 <div><strong>Lorem ipsum all the fact</div> <!-- Verrà selezionato: ha lo stesso elemento padre di <strong> ed è successivo a <strong> -->
20 </div>
21 <div><strong>Lorem ipsum all the fact</div> <!-- Non verrà selezionato: non ha lo stesso padre di <strong> anche se è successivo a <strong> -->
22 <div><strong>Lorem ipsum all the fact</div> <!-- Non verrà selezionato: non ha lo stesso padre di <strong> anche se è successivo a <strong> -->
23 </body>
24 </html>

```

Passo 3: Rendiamo tutto in movimento

Non resta che creare qualche animazione un po' più fluida, ma come promesso, senza utilizzare jQuery o Javascript.

Ti basta aggiungere una semplice regola di CSS3 agli elementi che subiranno gli spostamenti, ovvero all'elemento contenitore `.wrapper` ed al menu di navigazione `.offcanvas`:

```
.wrapper,  
.offcanvas { transition: all 0.25s; }
```

Ed ecco l'effetto finale, davvero molto molto bello ed elegante, guardalo su JSFiddle.

[Guarda la demo su JSFiddle](#)