Responsive design: come gestire layout, tipografia ed elementi multimediali?

Nella precedente lezione, abbiamo realizzato una pagina responsive attraverso la quale abbiamo analizzato il layout e le media queries offerte dal codice CSS3. La lezione di oggi è dedicata ai contenuti: le unità di misura, la tipografia, le immagini e i video.

Unità di misura del layout

Percentuali o pixel? La scelta delle dimensioni del layout è da analizzare molto attentamente in quanto possono esserci pareri differenti riguardo l'unità di misura da adottare. Da una parte c'è chi dice che un sito responsive debba essere totalmente flessibile. Quindi è d'obbligo usare le percentuali e mai i pixel:

In questo esempio, tutte le dimensioni delle classi e degli id sono espresse in percentuale. Il layout sarà quindi totalmente adattabile a qualunque risoluzione dello schermo.

Dall'altra parte invece si nota che l'uso esclusivo dei pixel o la combinazione pixel / percentuali è un aiuto per lavorare meglio su un layout fisso del quale possiamo conoscere fin dall'inizio il comportamento dei box:

Quale delle due soluzioni adottare? Effettivamente un sito responsive dovrebbe usare le percentuali e quindi permettere un adattamento a tutte le risoluzioni degli schermi. In questo modo l'uso delle media queries ci permette di spostare i box come si vuole e di lasciare che sia lo schermo a deciderne le dimensioni in base ai valori percentuali scritti nel codice CSS. Nasce, a mio avviso, un piccolo problema.

Fino a 1200px di risoluzione dello schermo i contenuti si possono adattare e gestire senza problemi. La sidebar potrebbe essere più larga e mostrare qualche informazione (per esempio una foto) in più o il riquadro dei contenuti principali potrebbe mostrare uno slide più grande e un numero di caratteri più elevato su un'unica riga riducendo l'effetto di scrolling in verticale. Cosa fare però se lo schermo dovesse avere una risoluzione elevata (Full-HD) o decisamente elevata (Retina Display)? Gestire i contenuti a risoluzioni così elevate non è sempre facile.

La risposta che ritengo più ragionevole alla questione su quale strada intraprendere (pixel, percentuali, o entrambi) è "a seconda dei casi". Un sito può essere progettato per essere visto a pieno schermo, e in questo caso è sicuramente doveroso usare le percentuali. Oppure pensato per lo scrolling in verticale, e in questo caso è possibile usare le dimensioni fisse con o senza l'uso

delle percentuali.

Tipografia

Altro aspetto da non tralasciare quando si progetta e si realizza un sito, in particolar modo un sito responsivo, è la tipografia.

Unità di misura

L'unità di misura da adottare su un sito responsivo (ma anche non) deve essere sempre quella relativa e mai fissa. Quindi **la dimensione dei font deve essere** espressa in em. Questo perché il px, l'unità di misura del display, è un'unità di misura fissa; l'em invece è un'unità di misura relativa che si adatta al dispositivo utilizzato. Quindi per il body possiamo impostare il font-size al 100%.

Per tutti gli altri testi, dai titolo ai paragrafi, dobbiamo invece usare gli em.

Utilizzando questa soluzione sarà sufficiente variare la dimensione del font-size nel body per modificare gli altri testi.

I Font Family

La scelta dei font su un sito responsive è la parte più delicata. I font disponibili su PC sono molti più numerosi rispetto agli smartphone, alcuni dei quali hanno giusto un font per ogni famiglia generica (sans-serif, serif, monospace).

Molto probabilmente uno smartphone equipaggiato con un sistema Android non visualizzerà il titolo h1 in Arial nè in Helvetica ma in Droid Sans. Come risolvere il problema?

Il CSS3 e vari servizi, come Google Web Fonts, ci hanno permesso di integrare nel sito font non installati sui dispositivo. Questa soluzione è molto comoda ma solo se si dispone di una connessione velocissima e senza limiti. Con gli smartphone e con i tablet non è molto praticabile.

Il progetto potrebbe complicarsi, visto che scegliere un font adeguato alla comunicazione del sito non è mai facile. In un sito responsive si potrebbe quindi decidere di usare due font differenti: uno specifico e personalizzato per i PC, ed uno generico per i dispositivi mobile.

Elementi multimediali

Riprendiamo il layout che abbiamo realizzato nella terza lezione.

Avevamo aggiunto sotto la barra di navigazione un'immagine che si adattava allo schermo grazie all'uso dell'istruzione background-size:containt;. Oggi vediamo in dettaglio come realizzare immagini responsive per il nostro sito, sostituiamo l'immagine grande sotto l'header con uno slider e aggiungiamo anche un video. Il tutto in maniera responsiva.

Come rendere le immagini responsive?

Ci sono, teoricamente, quattro possibilità per rendere le immagini responsive: l'istruzione *object-fit* del CSS3, il tag *picture* dell'HTML5, il tag figure dell'HTML5 appoggiato da jQuery ed infine una tecnica che utilizza il *max-width* del CSS2.

Iniziamo dal CSS3:

Con il codice sopra riportato **adattiamo le immagini ai loro contenitori grazie a *object-fit: contain***. Questa è una possibile soluzione facile da implementare ma comporta un enorme spreco di risorse per chi naviga in mobile, in quanto l'immagine originale deve essere di buona qualità. Inoltre l'object-fit è fruibile solo da pochi browser (in particolare Opera 10.6+, e a breve da Firefox 18+ e Chrome 24+).

Il W3C ha proposto come possibile soluzione un nuovo tag da aggiungere all'HTML5, picture**, che consentirà di includere più immagini, mostrando quella più adatta al dispositivo**.

Questa soluzione è però solo una bozza e richiederà un po' di tempo prima che venga completata e accolta dalle software house.

La terza soluzione alla quale possiamo ricorrere comprende **il tag** figure **dell'HTML5 e il plugin jQuery Picture**.

Il codice da riportare nella pagina HTML5 è:

Possiamo notare, nel codice dell'esempio, l'utilizzo di quattro immagini differenti, una per ogni tipologia di risoluzione, con l'immagine large.png resa predefinita.

mentre il codice che inizializza il plugin è

jQuery Picture può essere utilizzato anche con il tag picture discusso in precedenza:

e per inizializzare il plugin:

L'ultima soluzione è disponibile **utilizzando istruzioni già presenti in CSS2**:

Come per l'object-fit, anche in questo caso si ripete il problema del peso dell'immagine. A favore invece ci sono compatibilità con i browser e facilità di implementazione.

Dei quattro esempi, le migliori soluzioni sono le ultime due, quindi: il plugin jQuery Picture e le dimensioni relative dell'immagine (height, max-width, e width). Entrambi sono facili da implementare. jQuery Picture permette un risparmio di banda ma richiede JavaScript e il ritaglio di quattro immagini. Il CSS2 consuma più banda ma è immediato e funziona anche con JavaScript disabilitato.

Slider responsivo

Miglioriamo il nostro tema aggiungendo uno slider responsivo. Ho scelto ResponsiveSlides in quanto molto veloce e altamente compatibile, anche con Internet Explorer versioni 6,7,8. Dopo aver scaricato il plugin dal sito e scelto alcune immagini (possiamo anche usare quelle fornite dallo sviluppatore del plugin), installiamolo sul sito.

Il plugin

La lista delle immagini

Attivazione dello slider

Le istruzioni CSS

Et voilà… abbiamo installato il nostro slider!

Video responsivo

Infine condividiamo sul nostro sito responsive un video pubblicato su YouTube o su Vimeo. Come adattarlo? Le soluzioni non sono moltissime e presentano ancora qualche difetto di gioventù soprattutto quando lo schermo è troppo piccolo. La soluzione al momento migliore è offerta da fitvids.js, un plug-in jQuery semplice e veloce da installare pensato proprio per il responsive.

Il plugin

Il video da condividere

Anche se le dimensioni del video vengono definiti nel codice html, lo script che segue richiama la funzione fitVids indicando dove si trova il video indicando le nuove dimensioni.

Nell'esempio puoi vedere come il video si adatta al ridimensionamento del browser.

Conclusioni

Con la lezione di oggi abbiamo migliorato la nostra pagina con uno slider di immagini e con un video. Il tutto responsivo.

Cosa manca per completare il percorso che abbiamo affrontato sul responsive design? I test, utili a capire il comportamento del layout della nostra pagina responsive con i vari dispositivi. Nella prossima ed ultima lezione affronteremo questa parte.

Indice della guida