

Responsive Design: come e perchè progettare siti responsive?

Il tuo sito è **Responsive**? I tuoi visitatori riescono ad accedere comodamente da qualunque dispositivo? La parola d'ordine è mobilità, il fornire applicazioni, soprattutto web, su qualunque dispositivo, in qualunque angolo dell'Universo.

Prima di parlare, però, direttamente di **Responsive Design**, ritengo necessario scrivere una premessa per rispolverare un po' di passato, capire perché sia nato e come nel corso degli ultimi mesi questa tecnica si stia diffondendo.

Un po' di storia... ieri

Nel 2003-2004 coloro che avevano un accesso al web possedevano un PC desktop o un notebook. La dimensione degli schermi più diffusi variava dai 14" ai 17" e le risoluzioni dai 1024 ai 1280 pixel in larghezza. Accantonata l'obsoleta risoluzione a 800x600px, realizzare un sito web divenne più "semplice". Si puntò molto su un layout classico a 960px, il più diffuso, oppure su un design fluido, più difficile da gestire. E per 4-5 anni la situazione rimase stabile.

Dal 2007 in poi arrivarono con prepotenza e per qualunque tasca netbook, smartphone, tablet, schermi Full-HD e Retina Display, e Web-TV, stravolgendo il modo di navigare - immagino che per Jakob Nielsen si sarà accumulato molto materiale da analizzare per aggiornare i suoi studi sulla navigazione e sull'usabilità.

Oggi...

Le conseguenze le conosciamo di già. Imbottigliati nel traffico diamo uno sguardo alle email o alle ultime notizie. In vacanza ci siamo dimenticati dei passanti indigeni e ci facciamo consigliare dai sempre più numerosi portali che trattano di tutto o da Siri. In cucina non apriamo più il ricettario della nonna ma ora basta un app o un sito specializzato. E la sera in salotto annoiati dalle inutili trasmissioni televisive entriamo su Google e vagabondiamo per la rete.

Oggi gli accessi ai siti web da ciò che è diverso da un PC desktop o notebook sono in costante aumento e su alcuni siti sono addirittura superiori alla controparte meno mobile.

Cos'è il Responsive Design?

A questo punto concordiamo tutti, sia dal lato utente che dal lato progettazione, che un sito web debba essere realizzato per funzionare correttamente su più dispositivi. Nasce così il **Responsive**

Design che, come Ajax, non è un nuovo linguaggio di programmazione ma l'integrazione di strumenti già a disposizione da qualche anno: JavaScript, in particolar modo la libreria jQuery, e il CSS3. Inoltre, io aggiungerei anche un pizzico di organizzazione e studio, prima di iniziare a scrivere il codice, per disporre correttamente i contenuti all'interno della pagine.

Il termine **Responsive** in italiano è traducibile come **capacità di adattarsi**, e nel caso del web è evidente che debbano essere i contenuti ad adattarsi in larghezza e altezza al proprio contenitore fisico, il display.

Se sei un grafico o un esperto di usabilità non dovrai fare altro che progettare il layout per gli smartphone, per i tablet, per i netbook, e infine per i comuni schermi da PC desktop e notebook (l'ordine non è casuale ma voluto). Per iniziare bastano quattro versioni per capire come il sito potrebbe comportarsi visto da schermi differenti. L'approccio migliore è pensare il sito prima per il mobile, e via via progettarlo per le risoluzioni più grandi.

Dal lato CSS e HTML, invece, non dovrai imparare quasi nulla di nuovo, ma devi partire dal fatto che il codice dovrà essere scritto per più dispositivi, o meglio per più tipologie di risoluzioni del display.

In che modo scrivere il CSS? Prima di iniziare a scrivere il codice è necessario analizzare i layout grafici del sito progettati per i vari dispositivi ed evidenziare gli elementi in comune e quelli più importanti. Questi elementi comuni possono essere il background, il colore dei testi e dei link, i bordi delle immagini, per citarne solo alcuni. I più importanti sicuramente il logo, la barra di navigazione, e il contenuto principale della pagina.

La strada migliore che potrai intraprendere per realizzare il tuo sito **Responsive** è usare il CSS3. In questo caso non sarà necessario scrivere più fogli di stile. Ne basterà uno. La prima parte di codice va riservata agli elementi in comune e poi occorre suddividere il resto del codice per tipologia di display.

In CSS3 il richiamo alla tipologia del display lo si fa semplicemente con la dichiarazione

Dal lato HTML non cambia nulla. Questo perché è il CSS a fare tutto il lavoro, e in teoria anche i siti più vecchi possono essere adeguati alle nuove esigenze.

L'uso solo ed esclusivo del CSS3 però comporta alcuni problemi di compatibilità. Quello che vogliamo dal nostro sito è che sia visibile da quante più persone possibili. Giusto? I selettori CSS3, tra cui i media queries, non sono compatibili con le versioni 6, 7 e 8 di Internet Explorer.

Occorre quindi ricorrere a jQuery. Usando la libreria *Selectivizr* possiamo attivare i selettori CSS3 anche nelle vecchie versioni di IE.

Se deciderai di utilizzare jQuery per rilevare la risoluzione del display, potrai scrivere in un foglio di stile - che per comodità chiamiamo *screen.css* - tutti gli elementi che accomunano le versioni del sito. Mentre in fogli di stile separati (*mobile.css*, *1920.css*, ...) dovrai scrivere le caratteristiche relative ad ogni singolo dispositivo. L'uso esclusivo del jQuery però può comportare alcuni problemi a chi ha disattivato volutamente JavaScript e le caratteristiche Responsive si perderebbero.

La libreria [Adapt.js](#) di 960.gs analizza la risoluzione e richiama il file css desiderato:

Semplice, no?

Piccola curiosità...

E' però interessante notare il fatto che un sito web che non usi nessun selettore CSS, in pratica uno di quei vecchi siti scarni degli anni '90, sia già Responsive. E anche a voler aggiungere qualche foto e video la libreria jQuery ci viene in nostro aiuto. Non sarà però il nostro caso visto che il sito che vogliamo progettare deve essere completo.

Conclusioni

Oggi abbiamo "chiacchierato" sulla nascita del **Responsive Web Design**, accennando agli strumenti usati per renderlo immediatamente fattibile, almeno dal punto di vista del codice.

Dalla progettazione iniziale del layout al codice finale, ai trucchi per estendere il supporto anche ai vecchi browser per poi passare infine ai test e agli strumenti basilari, l'argomento **Responsive** è davvero interessante e utile.

Durante la prossima lezione, invece, analizzeremo nel dettaglio i dispositivi da prendere in considerazione, come organizzare i contenuti e come progettare il layout.

E tu hai già realizzato qualche sito Responsive? Se sì, quali sono le principali difficoltà che hai incontrato?

Indice della guida

Gli argomenti che tratteremo sul **Responsive Web Design** saranno i seguenti:

1. [Introduzione: come nasce il Responsive Design, accenno ai linguaggi di programmazione usati](#)
2. [Analisi dei dispositivi, studio del layout e gestione dei contenuti](#)
3. [I CSS3 media queries](#)

4. [Gestione del layout, tipografia ed elementi multimediali](#)

5. [Compatibilità e testing](#)