

# Responsive web design: compatibilità e testing

Compatibilità e browser: penso che per noi sviluppatori web questi due termini messi insieme significhino davvero tanto. Durante la fase di sviluppo scegliamo di controllare la resa visiva del sito sul nostro browser preferito (il mio è Firefox). A fine lavoro verifichiamo invece il risultato con gli altri browser. Ed è qui che iniziano le sorprese.

Il test con Opera, Safari, e Chrome solitamente si risolve con una rapida visualizzazione delle pagine e tutto si presenta esattamente come programmato. Accade sia perché Chrome (tra i più usati) si aggiorna costantemente e autonomamente e ciò significa che tutti hanno sempre l'ultima versione disponibile, sia perché Opera e Safari, sempre al passo coi tempi, sono poco usati e chi li usa li aggiorna costantemente.

Poi **arriva il momento di Internet Explorer**. E qui iniziano i grattacapi. Non tanto con la versione 9 e la recente versione 10 dove il lavoro finale è quello atteso, bensì con le precedenti, soprattutto con la 8 (6 e 7 sono quasi trascurabili grazie all'aggiornamento forzato voluto da Microsoft).

Il problema di fondo è che le media queries non funzionano con i vecchi browser, e di conseguenza **una pagina web responsive realizzata con CSS3 può essere visualizzata correttamente solo da chi ha un browser recente**. Fortunatamente esistono due soluzioni JavaScript che ci vengono incontro: Respond.js e adaptive.960.js. Vediamole insieme.

## Respond.js

La prima soluzione è la più semplice. Un modo per far digerire le istruzioni media queries del CSS3 anche alle vecchie versioni di Internet Explorer è l'utilizzo dello script [respond.js](#).

Questo script, chiamato dalla pagina come riportato sopra, aggiunge il supporto alle media queries, min-width, max-width e tutti i media types (es. screen) per i vecchi browser. In questo modo le istruzioni riportate al loro interno possono funzionare correttamente.

Per la sua semplicità di utilizzo e di integrazione può sembrare una soluzione fin troppo banale, ma per molti progetti rappresenta una soluzione ideale. Questo, dal mio punto di vista, perché il codice CSS3 è letto dai più recenti browser per PC e da tutti i browser pensati per smartphone. Il solo compito dello script è quello di far leggere le istruzioni all'interno delle media queries ai vecchi browser, tra i quali l'unico da tenere veramente in considerazione per la maggior parte dei progetti è Internet Explorer 8.

## adaptive.960.js

La seconda soluzione che andremo ad analizzare è molto diversa dalla precedente e soprattutto non fa uso del CSS3, quindi non fa uso delle media queries. Il concetto è però del tutto simile alle media queries del CSS3.

Se nel CSS3 usiamo un unico file, con questa libreria dobbiamo separare il codice in più file CSS, uno per ogni gruppo di risoluzione dei dispositivi. La libreria analizza la risoluzione del dispositivo e richiama il CSS realizzato per quella risoluzione.

[Scarichiamo lo script dal sito](#) e vediamo subito come funziona. Attiviamo la libreria nella nostra pagina:

E subito dopo copiamo il seguente codice:

Risalta subito la lista dei file CSS, all'interno di range, uno per ogni raggruppamento di risoluzioni. Ognuno di questi file CSS contiene le proprietà inerenti solo al dispositivo in questione, proprio come il codice CSS racchiuso all'interno delle media queries.

Le istruzioni CSS comuni ai vari layout, quelle che solitamente scriviamo in cima al CSS3 vanno inserite in un file CSS, che per comodità possiamo chiamare *master.css* e richiamato prima dell'integrazione della libreria, come un normale CSS:

Quale tra le due soluzioni adottare? Entrambe le soluzioni sono valide allo stesso modo. La prima, *respond.js*, però ci permette di scrivere e gestire un unico CSS, a differenza della seconda soluzione che invece richiede più file CSS. Considerando che nel giro di un paio d'anni i vecchi browser saranno irrilevanti nella progettazione di un sito web e che i browser mobile avranno la meglio sulla controparte desktop, è preferibile secondo me lavorare già con le media queries ed estendere il supporto ai vecchi browser con i plugin JavaScript disponibili.

## Test: la parte finale di ogni progetto

Quali sono gli strumenti da usare per testare il nostro sito?

Il modo più veloce e affidabile per verificare il corretto funzionamento del ridimensionamento del sito responsive è allargare e stringere la finestra del browser. È un sistema spartano, ma molto utile in fase di progettazione, soprattutto quando il sito è ancora in locale.

Esistono poi dei servizi web based per testare un sito responsive che simulano il ridimensionamento della finestra del browser. Uno di questi è <http://responsive.is/>.

## Your Inspiration Web

Web Design Community, ispirazione, tutorial, guide e risorse gratuite  
<http://www.yourinspirationweb.com>

---

Accedendo troviamo in alto al centro delle icone rappresentanti i principali terminali. Sul lato destro invece si inserisce l'indirizzo web da testare. Il sito è davvero minimale e facilissimo da usare.

Leggermente differente è invece l'approccio usato dal sito <http://www.studiopress.com/responsive/>. Inserendo l'indirizzo web, appariranno affiancati i diversi tipi di visualizzazione.

Oltre alle soluzioni web based appena viste, esistono anche degli applicativi che possono essere installati direttamente sui nostri PC. Tra i software più precisi da poter installare sul proprio computer, va sicuramente ricordato [Opera Mobile Emulator](#), disponibile per Mac, Linux e Windows.

Utilizzarlo è davvero semplice. Dopo l'installazione e l'apertura dell'applicativo, troviamo una lista di dispositivi che si possono emulare. Non troveremo però quelli della Apple, ma rimane comunque un ottimo e valido strumento.



In fine il test dei test: la prova diretta. Quella inconfutabile, quella che ci permette di vedere esattamente come si comporta il sito. Un computer desktop ce lo abbiamo già, e uno smartphone è difficile non averlo. A completare la triade manca il tablet. E' ovvio che non si richiede di possedere una lunga serie di dispositivi, ma un lavoro fatto veramente bene comporta dei test validi.

## Conclusioni e considerazioni

Eccoci giunti alle conclusioni finali. Dopo queste cinque lezioni sul Responsive Design, avete già messo in pratica alcune delle soluzioni proposte?

Leggendo i vostri commenti ho riscontrato alcune perplessità sull'uso delle media queries, dovute a volte all'impossibilità di applicare una simile strategia a tutti i siti. E' vero, le media queries non sono e non devono rappresentare la definitiva e l'unica soluzione per realizzare un sito web che funzioni su qualunque dispositivo.

Le media queries sono un ottimo strumento per migliorare l'esperienza di navigazione, visto che ci consentono di modificare il layout a seconda della risoluzione del display. Quanto riportato in

queste lezioni è stato soprattutto un modo per estremizzare l'uso delle media queries.

Qualunque tecnica che ci viene messa a disposizione non deve mai penalizzare il contenuto che nei nostri siti deve essere sempre il protagonista. Tutto il resto, elementi grafici e tecniche usate, sono solo un abile contorno per migliorare la comunicazione e raggiungere il più alto numero di utenti. Non è così?

*Alla prossima...*

## Indice della guida

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](#)