

HTML



HTML5

Approfondimento su layout Responsive

Approfondimento su Layout Responsive

In passato la risoluzione di riferimento è stata 1024px mentre per il web design la larghezza di riferimento delle pagine è stata di 960px.

Attualmente questo non è più vero: con l'avvento del mobile c'è l'esigenza di avere un layout che si adatta facilmente a tutte le risoluzioni possibili dei vari dispositivi, smartphone o tablet che siano, a prescindere dal sistema operativo.

Layout liquido

Una delle soluzioni fondamentali per svincolare il layout della pagina dalle ipotesi sui possibili dispositivi utilizzati dall'utente è quella di ricorrere a layout con griglie liquide, le quali non richiedono assunzioni sulla view-port, perché sono appunto, ideati per adattarsi a qualsiasi risoluzione.

Nell'impostazione di questa tipologia di layout nelle nostre definizioni CSS, in riferimento alle dimensioni, dobbiamo ragionare per percentuali, e non per dimensione “fissa”, dovremmo gestire il posizionamento delle nostre sezioni di pagina anche via javascript, faremo ricorso alle mediaquery per meglio applicare le classi CSS in base alle dimensioni di visualizzazione.

Approfondimento su Layout Responsive

Layout liquido

Consideriamo il seguente codice HTML

```
<section id="wrapper">
  <main>
    <section class="intro"></section>
  <section class="elencobox">
  <section class="singleBox"></section>
  <section class="singleBox"></section>
  <section class="singleBox"></section>
  <section class="singleBox"></section>
  <section class="singleBox"></section>
  <section class="singleBox"></section>
  <section class="singleBox"></section>
  <section class="singleBox"></section>
  </section>
  </main>
  <aside></aside>
  <footer></footer>
</section>
</body>
```

```
html {
  margin: 0;
  padding: 0;
}

#wrapper {
  margin: 3% auto;
  width: 90%;
  background: #ffffff;
  background: rgba(255, 255, 255, 0.8);
}

aside {
  float: left;
  width: 34%;
  margin-bottom: 1%;
}

main {
  float: right;
  width: 65%;
  margin-bottom: 1%;
}

.elencobox {
  width: 100%;
}

.elencobox .singleBox {
  margin-left: 1.538462%;
  margin-top: 1.538462%;
  width: 32.307692%;
  float: left;
}
```



Approfondimento su Layout Responsive

Layout liquido

Come si può notare tutte le dimensioni sono state espresse in percentuali, quindi si adatteranno alle dimensioni della finestra di visualizzazione.

Uno dei principali problemi delle griglie liquide consiste nel fatto che, al diminuire della larghezza della viewport, lo spazio a disposizione dei contenuti diventa più stretto e testi e immagini non solo possono sfiorare gli spazi in cui sono contenuti ma, anche quando gli spazi fossero rispettati, testi e immagini possono comunque disporsi in modo da essere scarsamente leggibili.

Smartphone e tablet ingrandiscono o riducono differenti elementi della pagina al fine di mostrare una pagina il più possibile coincidente con quella che si vedrebbe su un PC, con risultati a volte fedeli ma scarsamente fruibile.

Discorso analogo sulle immagini, in quanto avendo per loro natura dimensione fissa in pixel, dobbiamo essere in grado di ridimensionarle in base al contesto di visualizzazione.

Approfondimento su Layout Responsive

Definizione di viewport

Si tratta di un meta tag che viene inserito nell'<head> delle pagine ottimizzate per smartphones per indicare al browser del telefono che la sta mostrando il livello di zoom da utilizzare nella visualizzazione.

Vediamo un esempio pratico. Nella seguente immagine vediamo come una pagina senza meta tag viewport viene visualizzata da un iPhone.

I caratteri sono molto piccoli e quasi illeggibili. Il motivo è che il telefono, per rendere il più possibile del sito visibile, setta una larghezza dello schermo di 800 o 900 pixel (a seconda del browser), rendendo dunque pressochè inutilizzabile qualunque sito (l'idea è che si farà poi uno zoom nella parte che ci interessa).



Approfondimento su Layout Responsive

Definizione di viewport

Se invece noi vogliamo rendere il nostro sito già ben visibile dagli smartphones ci basterà aggiungere questa riga nell'<head> della pagina:

```
<meta name="viewport"
content="width=device-width , user-scalable=no">
```

e questo è quello che vedremo:



Approfondimento su Layout Responsive

Definizione di viewport

Attraverso il meta tag viewport abbiamo detto al browser di usare come larghezza dello schermo quella del telefono (nel caso dell'iPhone 320px), senza fare dunque nessuno zoom-out.

Ecco perchè i caratteri sono molto più leggibili e grandi.

Abbiamo poi anche detto (attraverso il comando `user-scalable=no`) di non consentire all'utente di cambiare il livello di zoom - questo per evitare che inavvertitamente il nostro visitatore faccia uno zoom-in o zoom-out scombinando il nostro sito ottimizzato.



Approfondimento su Layout Responsive

Definizione di viewport

Ecco di seguito tutti i valori che possono essere usati nell'attributo "content" di questo meta tag:

Valore	Descrizione
width	la larghezza del viewport. Si può specificare un valore in pixel (es: "320px") oppure "device-width". Quest'ultima maniera è quella consigliata dal momento che così facendo prende la larghezza del device specifico, iPhone, iPad, Android o qualsiasi cosa questo sia.
height	stesso discorso della width ma per l'altezza. Oltre ai pixel si può usare "device-height". In genere questo valore non viene indicato.
initial-scale	in una scala da 0 a 10, indica il livello di zoom iniziale (con 1 = nessuno zoom). Il consiglio è lasciare questo valore vuoto o settarlo a "1".
user-scalable	yes/no. Se no l'utente non può cambiare il livello di zoom della pagina.
minimum-scale	da 0 a 10, se user-scalable è "yes" indica il livello minimo di zoom che l'utente può impostare.
maximum-scale	da 0 a 10, se user-scalable è "yes" indica il livello massimo di zoom che l'utente può impostare.

Approfondimento su Layout Responsive

Media Query

La specifica CSS3 ha introdotto meccanismi per l'analisi del dispositivo usato dall'utente che si basano su proprietà quali la risoluzione o l'orientamento dello schermo.

Questi meccanismi, detti media query, sono particolarmente utili per la realizzazione di siti responsive.

Fino alla specifica CSS2 l'unico strumento a disposizione dei web designer per differenziare l'esperienza utente fra PC e smartphone o altri dispositivi era l'utilizzo di una fra le seguenti tecniche:

```
<style type="text/css">
@import "screen.css" screen;
@import "handheld.css" handheld;
</style>
```

```
<style type="text/css">
@media screen { /* Recole CSS per il media screen */ }
@media handheld { /* Recole CSS per il media handheld */ }
</style>
```

```
<style type="text/css" media="screen"></style>
<style type="text/css" media="handheld"></style>
```

```
<link rel="stylesheet" type="text/css" media="screen" href="screen.css" />
<link rel="stylesheet" type="text/css" media="handheld" href="handheld" />
```

Approfondimento su Layout Responsive

Media Query

Nonostante i numerosi tipi di media previsti, questo tipo di soluzione non ha mai sfondato tra i produttori di browser mobile, tanto meno tra i web designer.

La specifica CSS3 ha però introdotto nuovi meccanismi per l'analisi del tipo di dispositivi usato dall'utente, meccanismi che si basano su proprietà del device quali la risoluzione o l'orientamento dello schermo; questi meccanismi sono definiti “media query” e sono particolarmente utili per i dispositivi mobile e come base per i layout responsive.

Una media query è costituita dalla regola @media seguita da una parola chiave che rappresenta il tipo di medium (ad esempio “screen”) eventualmente seguita da una o più espressioni precedute dalla parola chiave “and”, le quali verificano condizioni poste su alcune caratteristiche del medium indicato (dette “media feature”): quando le condizioni sono vere per quel medium allora verranno applicate quelle regole CSS di formattazione.

Una media query può anche iniziare con la parola chiave “only” che ha il solo scopo di nascondere il foglio di stile ai browser che non supportano le media query (ad esempio IE8 e precedenti); possiamo anche avere la parola chiave “not” che nega il risultato delle media query.

Approfondimento su Layout Responsive

Media Query

Esempi di dichiarazioni CSS con media query:

```
<style type=text/css>
@media screen { /* Recole CSS per il media screen */ }
@media all and (color) { /* regole */ }
@media only screen and (min-device-width: 800px) { /* regole */ }
@media only screen and (device-width: 320px) and (orientation: portrait) {
  /* regole */
}
</style>
```

Le espressioni sono racchiuse tra parentesi tonde e possono contenere una media feature (ad esempio “color”) oppure una media feature e un valore con cui fare il confronto (ad esempio “min-device-width”).

Le media query sono utili per associare il giusto foglio di stile non ad uno specifico medium (screen o handheld) ma ad un medium che abbia determinate caratteristiche.

A questo punto dovrebbe essere chiara la grande utilità del tag meta “viewport” : senza l'uso di tale tag gli smartphone e tablet non risponderebbero correttamente alle media query.

Approfondimento su Layout Responsive

Internet Explorer 8 e Media Query

Internet Explorer 8 (e precedenti) non supporta le media query, per cui è necessario trovare delle soluzioni per ovviare al problema.

La prima soluzione consiste nell'utilizzare librerie Javascript che forniscono supporto per le media query anche per IE; le librerie più interessanti sono “Respond.js” (<https://github.com/scottjehl/Respond>) e “css3-mediaquery.js” (<https://code.google.com/p/css3-mediaqueries-js>).

Respond.js è più rapida ma supporta solamente le media feature “min-width” e “max-width” che sono quelle più utili ai fini del responsive web design.

Un ultimo approccio consiste nel caricare file CSS supplementari, contenente le regole aggiuntive, solo a Internet Explorer 8 e precedenti per mezzo dei commenti condizionali:

```
<!--[if lte IE8]>  
<link rel="stylesheet" href="ie8-specific.css" >  
<![endif]-->
```

Approfondimento su Layout Responsive

Immagini Responsive

Nel processo di realizzazione di un sito di adattarsi a qualsiasi dispositivo non può essere ignorato il problema della gestione delle immagini.

Molto del peso di una pagina è concentrato nelle immagini in essa contenute, pertanto utilizzare immagini sufficientemente grandi (grandi almeno quanto la massima dimensione alla quale saranno visualizzate), così da poterle riproporre a diverse risoluzioni e diversi dispositivi comporta un inutile aumento del peso della pagina, con conseguente (e a volte grave) riduzione delle performance.

Nel processo di realizzazione di una pagina in grado di adattarsi con efficacia alle varie modalità attraverso le quali questa può essere fruita non è possibile ignorare le performance, sia dal punto di vista del caricamento delle immagini stesse, sia dal punto di vista del traffico generato, ma anche dell'occupazione di memoria del browser; dispositivi come smartphone e tablet possono collegarsi al Web mediante connessioni Wi-Fi veloci o connessioni via rete cellulare, a volte lente e instabili.

Fornire mediante una connessione lenta e a consumo un'immagine 748x470 pixel quando magari è sufficiente una a 310x195 pixel, fornita con il solo scopo di dare una resa grafica migliore, non può essere considerata una buona pratica di responsive web design.

Approfondimento su Layout Responsive

Immagini Responsive

Il problema è più complesso e occorre considerare:

- Servire immagini delle giuste dimensioni è importante quando si ha a che fare con connessioni lente e a consumo, mentre con connessioni ad alta velocità il problema non si pone;
- La giusta dimensione non dipende solo dalla viewport messa a disposizione dai dispositivi utilizzati, ma dal browser e dalla connessione disponibile;
- La velocità di connessione non è facilmente valutabile e implica la sua misurazione prima di utilizzarla come variabile per la determinazione del tipo di immagine da visualizzare

I display retina (o comunque con elevata densità di punti per pollice) introdotti ad Apple con iPhone e iPad e successivamente implementati in altri dispositivi, sono caratterizzati da una elevata densità di pixel che però non riflette le dimensioni della viewport: iPhone5 ha una viewport di 320x568 pixel ma il display è costituito da 640x1136 pixel; pertanto per visualizzare un'immagine a tutto schermo, senza che appaia sgranata, ne occorre una da 640x1136 pixel.

Approfondimento su Layout Responsive

Immagini Responsive

La resa di un'immagine non predisposta per i display retina su è visibilmente peggiore di una predisposta per essi; pertanto può essere necessario discriminare le immagini da servire all'utente non solo in base alla viewport e alla velocità di connessione ma anche in base alla tipologia del display (retina o meno).

Immagini Responsive: Possibili soluzioni

Una delle soluzioni classiche per fornire specifiche informazioni a specifici browser consiste nell'inserirle nell'HTML e tramite CSS selezionare quali mostrare e quali nascondere. In questo modo però la classica soluzione CSS non solo non è utile, ma risulta dannosa in quanto i browser precaricano nella cache tutte le immagini inserite nell'HTML, prima ancora che i CSS siano applicati; è ancor più dannosa in quanto, inserire tutte le possibili varianti di un'immagine comporta l'aumento del numero di immagini da scaricare.

Una soluzione Javascript che si occupasse di riscrivere gli attributi "src" delle immagini presenti nella pagina non sarebbe molto efficiente, poiché i browser avviano il precaricamento delle immagini presenti sull'HTML prima ancora che le istruzioni JS possano modificare il DOM: in questo modo verrebbero caricate sia le immagini presenti inizialmente, sia quelle da sostituire.

Il problema non è affatto banale e non c'è ancora una soluzione definitiva e stabile.

Approfondimento su Layout Responsive

Immagini Responsive: Standard W3c in via di definizione

Presso il W3C sono in corso di definizione alcune proposte, tra cui l'attributo “srcset” ideato da Apple per supportare le immagini per display Retina:

```

```

In pratica l'attributo “srcset” consente di indicare una lista di possibili immagini da utilizzare in base alla dimensione della viewport e alla densità dei pixel del display: un valore del tipo 100w indica che la larghezza minima della viewport è di 100pixel.

Approfondimento su Layout Responsive

Immagini Responsive: Standard W3c in via di definizione

Un'altra proposta del W3C in via di definizione è l'elemento <picture>:

```
<picture width="500" height="500">  
  <source media="(min-width:45em)" src="lg.jpg" />  
  <source media="(min-width:18em)" src="md.jpg" />  
  <source src="sm.jpg" />  
    
  <p>Accesible text</p>  
</picture>
```

Il file da usare come fonte di dati per l'elemento <picture> è quello indicato nell'attributo "src" dell'elemento <source> il cui attributo "media" viene selezionato dalle media query correnti di visualizzazione della pagina.



Approfondimento su Layout Responsive

Immagini Responsive: Soluzioni Javascript

Qui un'elenco delle maggiori librerie Javascript utilizzate al momento:

- Picturefill: <https://github.com/scottjehl/picturefill>
- Filament Group Responsive Images:
<https://github.com/filamentgroup/Responsive-Images>
- jQuery Pictures: <http://jquerypicture.com/>
- HiRSC: <https://github.com/1Marc/hisrc>