

Esercizio 44 Media Query

Obiettivo: Rendere Responsive il sito Andreass00

- Sui tutti i contenitori di primo livello sostituire le **width** con **max-width**
Le sezioni float left e float right non sembrano essere responsive
- Aggiungere sul top frame **background-size: contain**, in modo che l'immagine diventi automaticamente responsive riducendosi proporzionalmente alla larghezza di wrapper

Confronto iniziale Crome Firefox (da visualizzare affiancati)

- Quando si scende sotto la dimensione di wrapper (772px) l'emulatore di Chrome continua a visualizzare una piccola porzione dell'immagine di sfondo di body, l'emulatore di Firefox NO
- L'immagine di sfondo di header, essendo dichiarata **background-size: contain**, viene automaticamente riscalata sia da Firefox che da Chrome.
- Rispetto a topFrame, l'emulatore di Chrome riscalca anche **height**, l'emulatore di Firefox NO, per cui sotto l'immagine viene a crearsi un fastidiosissimo spazio bianco che dovrà essere eliminato
- L'emulatore di Chrome riscalca anche il **font-size** (ma il sito diventa illeggibile), l'emulatore di Firefox NO

Aggiungere al fondo del CSS due media Query tali per cui:

- Sotto la larghezza soglia di 700px :
 - Applicare a wrapper una larghezza **100%** e nascondere l'immagine di sfondo del tag body. Applicare un colore di sfondo bianco.
 - Nascondere la sezione sinistra e la sezione destra
 - Visualizzare la mainSection a piena larghezza (**max-width: 98%**) .
attenzione che occorre anche riportare a **10px** il margin-left che era 160px
 - Ridurre l'altezza del topFrame proporzionalmente rispetto alla larghezza del wrapper, (problema di Firefox indicato sopra). Inizialmente (full screen) l'altezza risulta essere il 35% rispetto alla larghezza del wrapper (275 contro 775). Però purtroppo non è possibile impostare una altezza come % di width allora si usa il seguente trucchetto:
height: 0, padding-bottom: 35%;
Testo e img deborderanno automaticamente nel padding-bottom
 - Tra il TopFrame ed il MainFrame visualizzare il **leftFrame** (che dovrà essere **float: none** e anch'esso con **width: 98%**) contenente però soltanto i tre pulsanti di primo livello disposti orizzontalmente
Occorre pertanto:
nascondere **.menu2** e **.latestNews**
assegnare alle voci di menu1 la proprietà **display: inline-block;**
allargare il contenitore menu1 che aveva larghezza fissa di 150px
aumentare leggermente il **margin-top** di menu (16px contro i 10px precedenti dovuti al margin-bottom di topFrame. Tenere presente che margini verticali consecutivi si sovrappongono)

- Sotto la larghezza soglia di 480px :
 - Ridurre il font-size del body da 9pt a **8pt** (Firefox non lo riduce automaticamente)
Tutti i font-size di TopFrame e MainFrame sono espressi in **em** e dunque dovrebbero adeguarsi automaticamente (em esprime la dimensione del font-size rispetto al font-size del genitore). Sul TopFrame impostare invece **6pt**;
 - All'interno del LeftFrame, nascondere menu1 e visualizzare un BurgerButton **ancorato a destra** (già inserito nell'HTML prima di menù1) che consentirà poi, tramite javascript, di aprire con un click il menù, visualizzando i tre pulsanti di primo livello uno per riga. Assegnare al burger button un margine destro e inferiore pari a 5px;
 - Impostando float:right sul burgerButton, mainSection gli finisce sopra perché 'NON SENTI' gli elementi FLOAT. Per risolvere il problema si può dichiarare leftFrame come **overflow:auto** oppure dichiarare mainFrame come **clear:both** per interrompere la fluttuazione (meglio ancora!).
 - In corrispondenza dell'hover sul burgerButton visualizzare i tre pulsanti di primo livello **uno per riga** come se si trattasse di un menù aperto. Nota: dal momento che l'evento hover sul telefono non ha molto senso, l'emulatore di Chrome trasforma sostanzialmente l'evento hover in un evento click. In Firefox l'evento hover continua a comportarsi normalmente.

