

Ad

MRW.it → Web Design → Articoli → MRW Template: un (mini) framework CSS per il responsive design

MRW Template: un (mini) framework CSS per il responsive design

Ad



26/04/2019 Luca Ruggiero

MRW Template è un **framework CSS** di piccole dimensioni, realizzato in **HTML5 e CSS3**, messo a disposizione gratuitamente da Mr. Webmaster per semplificare la vita e velocizzare il lavoro dei webmaster e dei web designer che hanno la necessità di creare un **design responsivo** e non intendono partire da zero ne' tantomeno utilizzare altri framework, sicuramente più potenti e completi, ma anche molto più pesanti e che, in alcuni casi, fanno molte più cose di quelle che in realtà servono, consumando quindi più risorse del necessario.

Indice [nascondi]

- [MRW Template: finalità e aspetto grafico](#)
- [Scaricare il framework](#)
- [Pro e contro di MRW Template](#)
- [Struttura di base della pagina ed inclusione delle librerie](#)
- [Struttura di base del template](#)
- [Creare griglie responsive](#)
- [Referenze: classi e ID proprietari del framework](#)
- [Finestre modali](#)
- [Personalizzazione dello stile: gestione dei colori](#)

MRW Template: finalità e aspetto grafico

L'obiettivo di questo framework, infatti, è fare "lo stretto necessario", gestire header e footer, fornire una struttura pronta all'uso per il menu di navigazione, una griglia responsiva e qualche (piccola) utilità Javascript di uso comune (come, ad esempio, le finestre modali). Fine.

VEDI ANCHE...

- [Framework: cosa sono e come funzionano](#)
- [Caratteristiche del .NET Framework](#)
- [CSS media query](#)
- [Express: uno dei più popolari web framework per Node.js](#)
- [HTTP e classi del .NET Framework - Guida Sviluppo per Windows Phone 7](#)

GUIDE E MANUALI

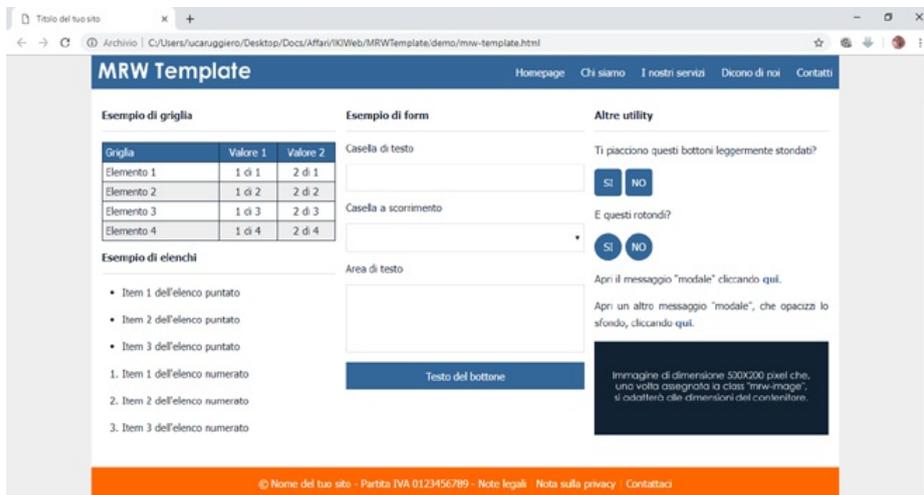
- [Guida Bootstrap](#)
Una guida completa ed esaustiva all'utilizzo della libreria Bootstrap grazie alla quale potremo c...
- [Guida Web Design](#)
Guida di base al design di pagine web che offre una panoramica sui software e sulle tecniche magg...

I NOSTRI CORSI

- Corso CSS**
Stilizzare siti Web coi CSS e CSS3.
A partire da 39 €
- Corso Web Design**
Webdesign e Responsive Design dei siti Web. A partire da 49 €
- Corso Webmaster Avanzato**
Diventa un Webmaster

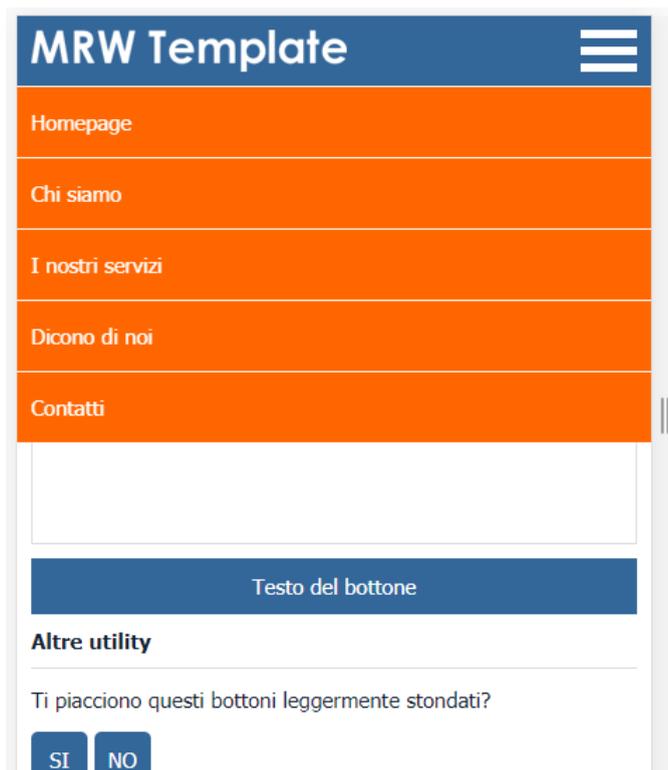
Ad

Ecco un'anteprima che contempla un po' tutte le caratteristiche del framework:



Ad

In versione mobile, col menu aperto:



Per farvi un'idea ancora più precisa del potenziale di MRW Template, è possibile visualizzare la [demo](#), da visitare sia da desktop che dispositivi mobili come tablet e smartphone. Il mio consiglio è di lasciare aperta la pagina della demo nel corso della lettura di questa guida, poichè tutti gli esempi che faremo sono illustrati così come li potete vedere e provare nella demo stessa.

Scaricare il framework

Questo, invece, è il link per effettuare il [download](#) della libreria. Scompattando l'archivio compresso troverete i seguenti file:

- **mrw-template.css** (ed **mrw-template.min.css**) - è il *core* del framework, il foglio di stile che "modella" il design del template;

- **mrw-custom.css** - è il foglio di stile personalizzabile, attraverso cui il web designer può gestire i colori di tutti gli elementi che compongono il template ed aggiungere, se lo desidera, le proprie stilizzazioni;
- **mrw-template.js** (ed **mrw-template.min.js**) - è il file Javascript che contiene le poche, ma indispensabili funzioni utili al template.

Sia il CSS che il Javascript sono disponibili sia in versione *minified* che in versione estesa, intelligibile e facilmente modificabile. Tuttavia, si tenga presente che i tre file insieme, così come sono e non compressi, occupano **meno di 10 Kb** di spazio su disco. Utilizzando le versioni minificate (come è giusto che sia per un sito in produzione) il peso è ancora inferiore.

Continua dopo la pubblicità...

Ad



Cerchi un hosting per WordPress?

Prova SiteGround e ti innamorerai. Super veloce, facile e sicuro.



Segretaria 24h/24. Prova GRATIS

Risparmia il 90% dei costi fissi del tuo ufficio utilizzando una segretaria virtuale.



Corsi di informatica on-line

E-Learning con tutor online ed attestato finale. Iscriviti subito ai corsi!

Alcune note legali: MRW Template è gratuito per uso personale e commerciale a condizione che le note sul copyright, contenute all'interno dei file CSS e JS, non vengano rimosse.

Pro e contro di MRW Template

Come ogni cosa a questo mondo, anche MRW Template ha dei pro e dei contro. I vantaggi sono sicuramente quelli già espressi, ovvero il fatto che si tratta di un framework estremamente leggero, quindi velocissimo da caricare, nonché molto semplice da utilizzare (gli elementi sono pochissimi rispetto ai framework più diffusi e blasonati).

Capita molto spesso, infatti, che i "grandi framework" prevedano le più disparate casistiche per soddisfare allo stesso modo sia le esigenze semplici (che rappresentano la stragrande maggioranza dei casi) che quelle più complesse, avendo di contro il fatto che a prescindere dalla propria esigenza ci si debba comunque "tirare dietro" una marea di codice che, probabilmente, non sfrutteremo mai.

Inoltre è perfettamente cross-device e cross-browser, a patto che i browser utilizzati abbiano il pieno supporto di HTML5 e CSS3, ovvero le versioni del markup e dello stylesheet utilizzati per scrivere il framework.

I contro di MRW Template sono sostanzialmente due:

1. è adatto a chi ha una buona conoscenza dell'HTML, dei CSS e delle tecniche di design in generale (cosa che, in realtà, dovrebbe valere per qualsiasi framework), visto che oltre alla struttura di base ed alcune utility, non offre per default altri elementi di design che dovranno, pertanto, essere create di volta in volta dallo sviluppatore che, pertanto, deve avere le conoscenze idonee per la loro gestione;
2. essendo volutamente "essenziale" potrebbe rivelarsi, per certi versi, piuttosto "rigido", quindi il suo utilizzo è consigliato per lo sviluppo di siti web per i quali si desidera impostare una grafica non troppo dissimile da quella che potete visualizzare nella demo.

Fatte le doverose premesse, è arrivato il momento di passare alla pratica.

Struttura di base della pagina ed inclusione delle librerie

La struttura HTML di base per creare un template col framweork in oggetto a questa documentazione è tipica di una pagina HTML(5) di base. Segue il codice nel quale sono evidenziate, mediante appositi commenti, le parti salienti:

```
<!DOCTYPE html>
<html lang="it">
  <head>
    <title>Titolo della pagina</title>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
    <!-- MOBILE FRIENDLY -->
    <meta name="viewport" content="width=device-width, initial-scale=1.
    <meta name="HandheldFriendly" content="true"/>
    <meta name="MobileOptimized" content="320"/>
    <!-- CSS -->
    <link rel="stylesheet" type="text/css" href="mrw-template.min.css"/>
    <link rel="stylesheet" type="text/css" href="mrw-custom.css"/>
    <!-- JS -->
    <script src="mrw-template.min.js"></script>
  </head>
  <body>

  <!-- Qui la struttura HTML della pagina -->

</body>
</html>
```

I punti salienti, della struttura di base di una pagina Web costruita con MRW Template, sono da ricercare nell'intestazione della pagina, ovvero il richiamo ai tre file: i due CSS ed il JS presentati in fase introduttiva.

Come anticipato, il compito del file *mrw-custom.css* è quello di offriure allo sviluppatore un'agevole gestione dei colori e di integrare, a seguire, le proprie regole di stile. Vedremo più avanti, nel corso della nostra documentazione, come impostare i colori ai vari elementi del template.

Si faccia attenzione: è importante includere sempre prima il file *mrw-template.min.css* e poi il file "custom" per questioni di ereditarietà, mentre il file Javascript può essere incluso indifferentemente prima o dopo rispetto ai figli di stile.

Da questo momento in poi si da per scontato che la struttura di ogni pagina che creerete, all'interno del vostro sito, sia come quella presentata in questo paragrafo: d'ora in avanti, infatti, agiremo sempre e solo all'interno dei tag `<body>` e la relativa chiusura `</body>`.

Struttura di base del template

Come già visto nella demo, MRW Template consente di gestire un layout centrato con un'header sottile: logo a sinistra e menu a destra che, in versione mobile (cioè a risoluzione schermo ridotta) scompare e verrà aperto a tendina attraverso la classica icona a forma di "hamburger".

Il footer del sito consente di gestire contenuti in modo centrato, mentre il corpo della pagina, il vero e proprio layout, sarà argomento dedicato del prossimo paragrafo.

Guardiamo dunque la porzione di codice da inserire tra i tag `<body>` e `</body>`:

```
<-- CONTENITORE -->
<div id="mrw-template">
  <-- HEADER -->
  <div id="mrw-header">
    <-- LOGO -->
    <div id="mrw-logo">
      
    </div>
    <-- MENU -->
    <div id="mrw-hamburger" onclick="mrwMenu()"><div></div><div></div><div></div><div></div>
    <ul id="mrw-menu">
      <li><a href="#">Link</a></li>
      <li><a href="#">Link</a></li>
      <li><a href="#">Link</a></li>
      <li><a href="#">Link</a></li>
    </ul>
  </div>
  <-- CONTENUTI -->
  <div id="mrw-main">
    <!-- Qui il contenuto del sito -->
  </div>
  <-- FOOTER -->
  <div id="mrw-footer">
    <!-- Qui il contenuto del footer -->
  </div>
</div>
```

Al tag principale che ospiterà il template assegneremo l'ID **mrw-template**, immediatamente seguito dal tag con ID **mrw-header**, che a sua volta contiene gli elementi **mrw-logo** (che contiene l'immagine col logo rappresentativo del sito), **mrw-hamburger** (all'interno di cui creeremo tre *div* vuoti e diventa l'icona del menu in visualizzazione mobile) e, in fine, **mrw-menu** (associato ad un elenco, contenente i link del menu di navigazione del sito).

Così com'è, l'header della pagina è già perfettamente funzionante sia dal punto di vista estetico che funzionale; il menu può agevolmente ospitare, mediamente, 5/6 link dal testo ragionevolmente conciso. E' chiaro che se l'esigenza fosse quella di avere un numero di link molto più alto e/o dal testo più lungo del normale, la soluzione ottimale sarebbe quella di un menu con dei sottomenu; tuttavia, questa possibilità non è attualmente offerta da MRW Template.

Due note sull'icona *hamburger* del menu: non necessita di immagini ne' di font grafici importati esternamente, ma è disegnata direttamente via CSS. Tuttavia, necessita dell'attributo *onclick* per scatenare la funzione **mrwMenu()** che apre e chiude il menu a tendina in versione mobile.

Chiuso l'header della pagina, abbiamo due macro blocchi: **mrw-main** ed **mrw-footer** che, come i loro nomi lasciano ad intendere, ospiteranno rispettivamente i contenuti veri e propri della pagina e quelli previsti per il footer.

Creare griglie responsive

Il layout della pagina, come suddetto, viene creato all'interno dell'elemento con ID *mrw-main*, all'interno del quale potremo avere infiniti blocchi contrassegnati da elementi a cui assegneremo la classe **mrw-cols**, all'interno dei quali potremo creare fino a cinque colonne a seconda della struttura di cui si necessita.

Di seguito una rappresentazione schematica di esempio che anticipa i nomi delle classi utili allo scopo in oggetto:

mrw-col-1-1				
mrw-col-1-2		mrw-col-1-2		
mrw-col-2-3			mrw-col-1-3	
mrw-col-1-3	mrw-col-2-3			
mrw-col-1-3	mrw-col-1-3		mrw-col-1-3	
mrw-col-3-4			mrw-col-1-4	
mrw-col-1-4	mrw-col-3-4			
mrw-col-1-4	mrw-col-1-4	mrw-col-1-4	mrw-col-1-4	
mrw-col-1-5	mrw-col-4-5			
mrw-col-4-5			mrw-col-1-5	
mrw-col-2-5		mrw-col-3-5		
mrw-col-3-5			mrw-col-2-5	
mrw-col-1-5	mrw-col-1-5	mrw-col-1-5	mrw-col-1-5	mrw-col-1-5

In sostanza possiamo creare un layout a colonna unica, larga quanto l'intero template, piuttosto che a due colonne di egual misura, o di tre, quattro e cinque colonne di egual misura, o ancora a due o più colonne, ma di misure diverse.

Un trucco per ricordare i nomi delle classi delle colonne: la colonna singola "1-1" può essere letta come "uno di uno"; la colonna "1-2" può essere letta come "uno di due"; quella "2-5" può essere letta come "due di cinque", quindi cioè "due colonne di cinque totali". E così via per il resto delle classi.

Vediamo dunque qualche esempio, iniziando dal layout ad una colonna:

```
<div class="mrw-cols">
  <div class="mrw-col-1-1">
    Contenuto della colonna...
  </div>
</div>
```

Adesso creiamo un layout a due colonne di egual misura:

```
<div class="mrw-cols">
  <div class="mrw-col-1-2">
    Contenuto della colonna...
  </div>
  <div class="mrw-col-1-2">
    Contenuto della colonna...
  </div>
</div>
```

Ora, invece, creiamo due elementi *mrw-cols*, il primo da 5 colonne, aggiungendo di seguito un layout a 3 colonne:

```
<div class="mrw-cols mrw-space">
  <div class="mrw-col-1-5">
    Contenuto della colonna...
  </div>
  <div class="mrw-col-1-5">
    Contenuto della colonna...
  </div>
</div>
<div class="mrw-cols">
  <div class="mrw-col-1-3">
    Contenuto della colonna...
  </div>
  <div class="mrw-col-1-3">
    Contenuto della colonna...
  </div>
  <div class="mrw-col-1-3">
    Contenuto della colonna...
  </div>
</div>
```

E così via, fino ad arrivare alla struttura di propria esigenza. Naturalmente le colonne saranno affiancate in versione desktop, ma visualizzate una sotto l'altra in versione mobile.

Ad

Notiamo, infine, che al primo blocco *mrw-cols* aggiungiamo la classe **mrw-space** che, come vedremo in seguito, serve a dare una spaziatura inferiore rispetto al layout successivo. L'ultimo div con classe *mrw-cols* solitamente non necessita di questa classe aggiuntiva, poichè tale spaziatura è gestita nativamente dalla classe del contenitore principale.

Referenze: classi e ID proprietari del framework

Nel corso dei paragrafi precedenti già abbiamo anticipato una nutrita schiera di classi ed ID che caratterizzano e compongono MRW Template. In questo paragrafo li referenzieremo tutti, in ordine alfabetico, compresi quelli di cui abbiamo già discusso. Al termine dell'elenco faremo alcuni esempi per la gestione di elementi dei quali non abbiamo già parlato.

Iniziamo dagli ID e, a seguire, le classi.

- **mrw-footer** - footer, chiusura, "piè di pagina" del sito;
- **mrw-hamburger** - icona per la gestione del menu in versione mobile, contenuto all'interno di "mrw-header";
- **mrw-header** - header, inizio, intestazione del sito;
- **mrw-logo** - ospita il logo del sito, contenuto all'interno di *mrw-header*;
- **mrw-main** - ospita il layout a colonne che rappresenta il contenuto vero e proprio delle pagine;
- **mrw-menu** - assegnato ad un elenco (preferibilmente puntato), ospita i link del sito ed è contenuto all'interno di *mrw-header*;
- **mrw-template** - tag principale del sito, va aperto subito dopo <body> e chiuso immediatamente prima di </body>.

Veniamo dunque all'elenco delle classi.

- **mrw-block** - assegna larghezza 100% ad un elemento che ha una larghezza specificata o predefinita;
- **mrw-center** - centra un elemento rispetto al box in cui è contenuto;
- **mrw-circle** - arrotonda un box fino a renderlo perfettamente circolare (a condizione, ovviamente, che altezza e larghezza siano eguali);
- **mrw-col-1-1** - layout ad 1 colonna, contenuto all'interno di un elemento con classe "mrw-cols";
- **mrw-col-1-2** - layout a 2 colonne, contenuto all'interno di un elemento con classe "mrw-cols";
- **mrw-col-1-3** - layout a 3 colonne, contenuto all'interno di un elemento con classe "mrw-cols";
- **mrw-col-1-4** - layout a 4 colonne, contenuto all'interno di un elemento con classe "mrw-cols";
- **mrw-col-1-5** - layout a 5 colonne, contenuto all'interno di un elemento con classe "mrw-cols";
- **mrw-col-2-3** - layout con colonna larga due terzi dello spazio a disposizione, contenuto all'interno di un elemento con classe "mrw-cols".
- **mrw-col-2-5** - layout con colonna larga due quinti dello spazio a disposizione, contenuto all'interno di un elemento con classe "mrw-cols".
- **mrw-col-3-4** - layout con colonna larga tre quarti dello spazio a disposizione, contenuto all'interno di un elemento con classe "mrw-cols".
- **mrw-col-3-5** - layout con colonna larga tre quinti dello spazio a disposizione, contenuto all'interno di un elemento con classe "mrw-cols".
- **mrw-col-4-5** - layout con colonna larga quattro quinti dello spazio a disposizione, contenuto all'interno di un elemento con classe "mrw-cols".
- **mrw-cols** - struttura di base di un layout a colonne;
- **mrw-grid** - crea una griglia utilizzando una classica tabella HTML;
- **mrw-header-fixed** - viene applicata tag <body> nel caso in cui si desideri fissare l'header della pagina (contenente logo e menu di navigazione) allo scroll di pagina, per lasciarlo sempre fisso ed in primo piano;
- **mrw-image** - viene assegnata ad un'immagine che si intende rendere responsiva, fluida, che si adatti quindi al 100% delle dimensioni del box che la contiene;
- **mrw-list** - rende cross-browser un elenco, sia esso puntato o numerato, in termini di spaziatura sotto l'aspetto del box-model (va applicata ai tag oppure , ma non a);
- **mrw-modal** - genera una finestra modale secondo la stessa filosofia di un "alert" nativo del Javascript;
- **mrw-rounded-L** - stonda un box in modo piuttosto evidente;
- **mrw-rounded-M** - stonda un box in modo abbastanza evidente;
- **mrw-rounded-S** - stonda un box in modo lieve;
- **mrw-space** - crea una spaziatura dopo un elemento che, per default, prevede il ritorno a capo;

Apparentemente sembra poca roba, ma tutto sommato, considerando che lo scopo di MRW Template è quello di fornire uno strumento per velocizzare delle esigenze più o meno semplici (non quello di soddisfare tutto lo scibile delle esigenze), è anche più del

necessario. Altri elementi (come paragrafi, form, semplici link linee di separazione orizzontali, titoli, tabelle e così via) sono gestiti per default, senza necessità di applicare classi.

Ad

Passiamo dunque alla pratica, facendo alcuni esempi che contemplino elementi non ancora analizzati nel corso dei precedenti paragrafi. Abbiamo già visto tutto relativamente all'utilizzo degli ID, per cui ci concentreremo su quelle classi di cui non abbiamo ancora visto nulla di pratico.

Iniziamo con un esempio che "faccia fuori" in un colpo solo due classi; creiamo un box il cui contenuto sarà centrato ed abbia una spaziatura rispetto agli elementi sottostanti:

```
<div class="mrw-center mrw-space">Testo centrato con spaziatura verso il ba
<div class="mrw-center mrw-space">Altro testo centrato con spaziatura verso
```

Si provi a rimuovere dal primo box la classe *mrw-space* per notare che i due box non avranno spaziatura, bensì solo il ritorno a capo. La classe *mrw-space* è molto comoda in tantissimi casi e si consiglia la sua applicazione combinata (oltre che a se stante, ove opportuno) soprattutto insieme alle seguenti classi:

- *mrw-cols*, per distanziare un insieme di colonne rispetto agli elementi sottostanti;
- *mrw-grid*, per distanziare la griglia rispetto agli elementi sottostanti.

Vediamo come fissare l'header allo scroll di pagina, utilizzando la classe **mrw-header-fixed** che verrà applicata al corpo della pagina che diventerà:

```
<body class="mrw-header-fixed">
```

Ed il gioco è fatto, tutto qui: nulla di più e nulla di meno.

Ad

Per rendere un'immagine al 100% fluida e che mantenga le dimensioni del box che la contiene, è sufficiente applicare la classe **mrw-image** come nell'esempio che segue:

```

```

Anche nel caso in cui vengano esplicitamente assegnate le dimensioni al tag che la mostra a video, l'immagine verrà gestita dal template in modo responsivo:

```

```



Passiamo alla gestione degli elenchi che, come anticipato, non assegna per default particolari aspetti stilistici alla lista, sia essa puntata o numerata, ma la normalizza in modo che sia visibile allo stesso modo, in termini di margini e spaziature, su qualsiasi browser. Vediamo un esempio:

```
<ul class="mrw-list">
  <li>Item 1 dell'elenco puntato</li>
  <li>Item 2 dell'elenco puntato</li>
  <li>Item 3 dell'elenco puntato</li>
</ul>
```

E' dunque sufficiente assegnare la classe **mrw-list** al tag ``. Stesso discorso per il tag `` per quanto riguarda gli elenchi numerati:

```
<ol class="mrw-list">
  <li>Item 1 dell'elenco numerato</li>
  <li>Item 2 dell'elenco numerato</li>
  <li>Item 3 dell'elenco numerato</li>
</ol>
```

Facciamo un esempio di griglia, creando quindi una tabella a cui assegneremo la classe **mrw-grid**:

Ad

```

<table class="mrw-grid mrw-space">
  <tr>
    <td style="width: 50%;">Griglia</td>
    <td class="mrw-center" style="width: 25%;">Valore 1</td>
    <td class="mrw-center" style="width: 25%;">Valore 2</td>
  </tr>
  <tr>
    <td>Elemento 1</td>
    <td class="mrw-center">1 di 1</td>
    <td class="mrw-center">2 di 1</td>
  </tr>
  <tr>
    <td>Elemento 2</td>
    <td class="mrw-center">1 di 2</td>
    <td class="mrw-center">2 di 2</td>
  </tr>
  <tr>
    <td>Elemento 3</td>
    <td class="mrw-center">1 di 3</td>
    <td class="mrw-center">2 di 3</td>
  </tr>
  <tr>
    <td>Elemento 4</td>
    <td class="mrw-center">1 di 4</td>
    <td class="mrw-center">2 di 4</td>
  </tr>
</table>

```

Una volta creata la tabella ed impostata la classe, non abbiamo bisogno di altre classi proprietarie per ottenere tutti gli effetti visti nella demo: prima riga evidenziata a mo' di titoli delle colonne, righe di colori alternati che si evidenziano con un colore differente al passaggio del mouse.

Utilizziamo la classe "mrw-center" per centrare il contenuto delle celle che "è bene" siano centrate, quindi usiamo un normalissimo stile in linea per dimensionare le colonne (applicandolo solo alla prima riga, visto che le successive ereditano lo stesso stile).

Vediamo un esempio che utilizzi le classi **mrw-block** ed **mrw-rounded-S** (l'utilizzo di -M ed L è identico, cambia solo il grado di stonatura) applicandolo ad un **form**. Segue il codice presente nella demo linkata all'inizio della guida, ma con l'aggiunta della stonatura:

```

<form onsubmit="return false">
  <p>Casella di testo</p>
  <div class="mrw-space"><input type="text" name="testo" class="mrw-block
  <p>Casella a scorrimento</p>
  <div class="mrw-space">
  <select name="selezione" class="mrw-block mrw-rounded-S">
    <option value=""></option>
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
  </select>
  </div>
  <p>Area di testo</p>
  <div class="mrw-space"><textarea name="area" class="mrw-block mrw-round
  <div class="mrw-space"><button class="mrw-block mrw-rounded-S" onclick=
</form>

```

Creiamo dunque un form HTML senza preoccuparci di impostare particolari classi, ma solo

"mrw-block" ed "mrw-rounded-S" che, rispettivamente, faranno sì che i campi di input (input, select, textarea e button) saranno larghi il 100% dello spazio a disposizione ed appaiano leggermente stondati.

Qualora volessimo applicare ad un box l'effetto perfettamente rotondo, utilizzeremo la classe **mrw-circle**, partendo dal presupposto che il box abbia già un suo stile in termini quantomeno di dimensioni e colori:

Ad

```
<div class="mrw-circle" style="background: #000; width: 100px; height: 100px">
```

Finestre modali

Concludiamo con il sistema di apertura di finestre modali, ossia la simulazione di un tipico messaggio di avviso di tipo *alert* nativo del Javascript, ma dall'aspetto più professionale ed in linea col design del sito, ottenibile grazie all'utilizzo della classe **mrw-modal** che applicheremo ad un "div" come di seguito indicato:

```
<p>Apri la <a href="javascript:mrwModal('mio-id', 'Titolo...', 'Messaggio...')">Apri la finestra</a>
<div class="mrw-modal" id="mio-id"></div>
```

Nulla di più semplice, dunque: un "div" a cui assegneremo la classe in questione ed un ID a nostro piacimento. Guardiamo, piuttosto, come gestire la funzione Javascript **mrwModal()** che apre la finestra. Abbiamo quattro parametri:

1. l'ID che abbiamo scelto;
2. il titolo del box di avviso;
3. il testo del messaggio che intendiamo dare al nostro visitatore;
4. un booleano che può avere valore **1** oppure **0** a seconda che si voglia o meno opacizzare lo sfondo della pagina nel momento in cui la finestra si apre.

Nell'esempio il quarto parametro è zero, quindi lo sfondo opacizzato non apparirà. Impostando uno, invece, questi sarà impostato.

Personalizzazione dello stile: gestione dei colori

Non ci resta che imparare a personalizzare il nostro MRW Template agendo sul file **mrw-custom.css** di cui segue il codice completo:

```

body { background: #DDD; }
div, h1, h2, h3, p, li, input, select, textarea, button { color: #000; }
a { color: #00F; }
a:hover { color: #F00; }
input[type="text"], select, textarea { border: solid 1px #000; }
input:focus, select:focus, textarea:focus { background: #FFC; }
button { background: #000; color: #FFF; }
button:hover { background: #555; color: #FFF; }
hr { border-bottom: solid 1px #000; }
*::selection { background: #F00; }

/* ***** */

#mrw-header { background: #000; }
#mrw-hamburger div { background: #FFF; }
#mrw-menu a { color: #FFF; }
#mrw-main { background: #FFF; }
#mrw-footer { background: #555; color: #FFF; }
#mrw-footer li { border-top: dotted 1px #FFF; }
#mrw-footer li::after { color: #777; }
#mrw-footer a { color: #FFF; }
#mrw-footer a:hover { color: #EEE; }

/* ***** */

.mrw-grid { border-top: solid 1px #000; border-left: solid 1px #000; border
.mrw-grid tr { background: #FFF; border-bottom: solid 1px #000; }
.mrw-grid tr:nth-child(odd) { background: #777; }
.mrw-grid tr:first-child { background: #000; color: #FFF; }
.mrw-grid tr:not(:first-child):hover { background: #EEE; color: #000; }
.mrw-grid tr td { border-right: solid 1px #000; }
.mrw-modal div:first-child { background: #000; color: #FFF; }
.mrw-modal div:first-child::after { color: #FFF; }
.mrw-modal div:last-child { background: #777; color: #000; }
.mrw-modal-background { background: rgb(0, 0, 0); background: rgba(0, 0, 0,

/* ***** */

@media screen and (min-width: 320px) {
#mrw-menu li { border-top: solid 1px #FFF; }
#mrw-menu a { background: #777; }
}
@media screen and (min-width: 800px) {
#mrw-menu li:hover { background: #333; }
}

```

Come possiamo vedere è abbastanza corposo, ma il suo unico scopo è quello di gestire il colore di sfondi, testi (e link) e bordi. Il codice è stato diviso in quattro blocchi: tag di base, ID, class e media query. Esaminiamoli nel dettaglio.

Il primo blocco assegna i colori al corpo della pagina, agli elementi testuali, ai link, agli elementi dei form, alla linea di separazione orizzontale ed all'atto della selezione degli elementi della pagina.

Il secondo blocco imposta i colori agli elementi strutturali della pagina per ID, quindi in sostanza all'header, al corpo, al footer ed al menu di navigazione del sito.

Il terzo blocco assegna i colori alle classi della pagina: griglie e finestre modali.

Il quarto ed ultimo blocco imposta una serie di caratteristiche aggiuntive al menu, a seconda che sia visualizzato da dispositivo mobile (o tablet di piccole dimensioni) piuttosto che da dispositivo desktop (o tablet di medio-grandi dimensioni).

Suggerisco di salvare una copia di backup del file e di provare a personalizzarlo usando i propri colori preferiti, impostando ogni elemento col colore giusto per il suo scopo in funzione dello scopo finale che si ha in mente.

I COMMENTI DEGLI UTENTI

I **commenti sono liberi**: non è necessario iscriversi per poter commentare su questa pagina. Tutti i commenti, tuttavia, sottoposti alle **linee guida di moderazione** e prima di essere visibili devono essere approvati da un moderatore.

SCRIVI UN COMMENTO

← PRECEDENTE

Mostrare le immagini allo scrolling di pagina

SUCCESSIVO →

Mostrare una schermata di loading in attesa del caricamento della pagina



© 2003 - 2021 **Mr. Webmaster**

Mr. Webmaster ® è un marchio registrato.

E' vietata ogni forma di riproduzione.

IKIweb Internet Media S.r.l. - P.IVA

02848390122

Parliamo di noi: chi siamo / cronologia

Contatti: pubblicità / contattaci / segnala abusi

Note Legali: condizioni d'uso / privacy / cookie

Altro: sondaggi / feed / preferenze ADV



IKIweb