

TP1 – Conception de la charte graphique

1 Problématique

On voudrait développer la page HTML suivante :



Nous avons deux divs l'un à côté de l'autre. Ecrire le code HTML nécessaire pour avoir ce résultat. On vous donne le code HTML :

```
<div class="container">
  <div id="one">Div one</div>
  <div id="two">Div two</div>
</div>
```

Voici quelques indices :

- (a) La bordure du conteneur est de 2px, ligne pleine (solid), noir :
`border : 2px solid #000;`
- (b) Pour le div one :
 - (1) couleur de fond : gris
 - (2) flottaison : gauche
 - (3) marge à droite : 20px
 - (4) bordure à droite de 2px, solid et noire
 - (5) hauteur : 194 px
 - (6) largeur : 200px
- (c) Pour le div two
 - (1) Débordement : caché
 - (2) marge (des 4 côtés) : 10px
 - (3) bordure grise de 2px, style « dashed »
 - (4) hauteur : 170px ;

On voudrait maintenant que le div two devienne au dessus du div one quand la largeur de la fenêtre (ou de l'écran) est inférieure à 768px, voici le résultat souhaité :



Le div one ne flotte plus, la marge devient nulle, idem pour des 4 bordures qu'on remplace par une bordure en bas de 2px

Ajouter (et compléter) alors les règles CSS suivantes :

```
@media (max-width: ???px) {
  #??? {
    float: ???;
    margin-right: ???;
    width: auto;
    border: ???;
    border-??? : 2px solid #000;
  }
}
```

CONCLUSION : beaucoup de code pour une si petite page

2 Solution : Bootstrap

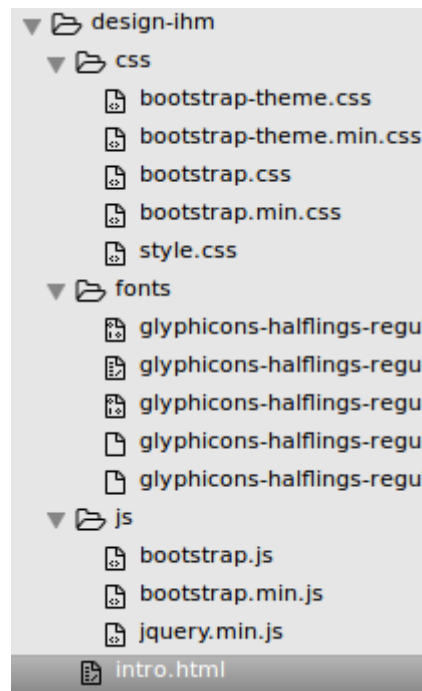
Nous allons télécharger la version 3 de Bootstrap, disponible à l'URL getbootstrap.com (360 ko)
 Dans votre dossier WWW créer le dossier « bloggy » et y copier le contenu du fichier téléchargé.
 Bien analyser son contenu et en discuter avec votre enseignant.

Dans le même dossier créer un nouveau fichier, le sauvegarder sous le nom « index.html » et générer le code HTML (via sublime text)

Faire appel aux fichiers suivants, nécessaires pour le bon fonctionnement de Bootstrap :

- css/bootstrap.min.css
- js/bootstrap.min.js
- js/jquery.min.js
- css/style.css (fichier vide à créer, il contiendra toutes le code CSS personnalisé)

Voici un aperçu de l'arborescence du projet :



Créer et compléter le fichier « intro.html », ensuite le visualiser sous votre navigateur, bien analyser le comportement de la page lors du changement de la taille de la fenêtre :

```
<!DOCTYPE html>
<html>
<head>
  <title>Mon 1er TP Bootstrap</title>
  <link rel="stylesheet" type="text/css" href="???">
  <script type="text/javascript" src="???"></script>
  <script type="text/javascript" src="???"></script>
</head>
<body>
  <div class="container">
    <div class="row">
      <div id="one" class="col-sm-4"> Div 1 </div>
      <div id="two" class="???-??-?"> Div 2 </div>
    </div>
  </div>

  <style type="text/css">
    #one, #two {
      border: 2px solid #00FF00;
      background-color: #ccc;
      padding : 5px;
    }
  </style>
</body>
</html>
```

3 Projet

Il s'agit de développer la charte graphique d'un blog, ce dernier est composé de 5 pages :

- Page d'accueil (home) : on y liste un aperçu des derniers articles du blog
- Détails article
- Page « A propos » (page de contenu statique)
- Formulaire de contact

Détaillons la page d'accueil :



On peut composer la page en 4 lignes (rows) :



Ce qui change entre les pages c'est uniquement la 3ème ligne.

Créer un fichier appelé accueil.html. Dans la balise « body » créer la balise `<div class="container"></div>`

Dans cette balise insérer ce code :

```
<div class="row" id="menu">
  <div class="col-md-9"></div>
  <div class="col-md-1">Accueil</div>
  <div class="col-md-1">A propos</div>
  <div class="col-md-1">Contact</div>
</div>
```

Explications : on a créé une ligne composée de 4 cellules. Une ligne peut contenir 12 cellules, cependant dans notre cas on n'en a que 4, on va donc fusionner les 9 premières, et laisser les 3 autres. Donc pour résumer, la 1ère colonne fait 9/12^{ème} de la fenêtre, et les autres chacune 1/12^{ème}.

Pour centrer un texte dans un div, appliquer à ce dernier la classe « text-center »

On voudrait avoir une ligne horizontale grise (#ccc) au dessous du menu principal, on sait qu'en CSS on peut appliquer une bordure (border) à un élément, ce dernier possède alors 4 côté : bottom, top, left et right. On va donc appliquer un cadre uniquement à la partie inférieure du div contenant le menu. Créer la règle CSS permettant d'appliquer au div du menu une bordure basse ayant (dans l'ordre) les caractéristiques suivantes :

- Épaisseur de 1px
- Ligne continue (solid)
- Couleur grise #ccc

On remarque enfin que chaque menu est dans un cadre, sauf le dernier :

	Accueil	A propos	Contact
--	---------	----------	---------

On utilisera le même principe : chaque menu aura une bordure gauche.

Problème : en réduisant la largeur de la fenêtre à 991 px (ou moins) on remarque que le menu est cassé : c'est parce qu'on a pris en compte uniquement le cas d'un écran normal (col-md-* : md = medium) Ajouter une nouvelle règle pour chaque cellule afin que pour une taille d'écran plus petite (sm = small) on ait l'arrangement suivant : 6/12, le reste à décomposer sur 3 cellules.

Encore un problème pour une taille inférieure à 767px : il faut alors créer une règle, trouvez alors la bonne répartition, sachant que xs = extra small

Nous venons de créer un menu **responsiv design** :)

Continuer à créer le reste de la page en se basant sur la décomposition suivante :



NOTE : div#sidebar : appliquer une police italique (font-style) de 10px

Remarquons que le H2 et le H3 sont dans une police non standard. Il s'agit de nouvelles polices importées depuis Google Fonts, ajouter ces instructions dans le header :

```
<link href='http://fonts.googleapis.com/css?family=Irish+Grover'
rel='stylesheet' type='text/css'>
<link href='http://fonts.googleapis.com/css?
family=La+Belle+Aurore' rel='stylesheet' type='text/css'/>
```

Sachant que :

- l'option « font-family » permet de définir la police d'un texte
- dans les urls ci-dessous, le « + » représente un espace
- le nom de la police doit être entre ' et '

Appliquer la 1ère police au H2 du header, et la 2ème police au H3, définir aussi la taille de chaque balise afin de s'approcher de l'image ci-dessus.

4 Responsive design

En réduisant la taille de la fenêtre on peut avoir le cas suivant :

Le blog de la mort qui tue la vie

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiatcitatation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiatcitatation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Sidebar Lorem ipsum, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat

(c) 2016 - Bloggy

Problème : le sidebar s'affiche au dessous, ce qui est bien, mais on a une séparateur vertical à droite ! Il faut le remplacer par un séparateur horizontal quand la largeur de l'écran est inférieure à 991px, ceci est possible en utilisant le code suivant :

```
@media (max-width: 991px) {
  #content_text {
    border-right: none;
    border-bottom: 1px solid #ccc;
  }
}
```

Ajouter du code CSS afin de faire disparaître le sidebar quand la largeur est inférieure à 766px.

5 Correction du menu

Notre menu est bien adapté à un affiche sur petit écran, cependant si on avait plus d'options dans le menu on n'aurait pas assez de place, il faut dans ce cas cacher le menu et afficher à la place une icône permettant de faire dérouler le menu. Cette icône s'appelle « menu hamburger » car elle ressemble à un Hamburger de 4 tranches !

Remplacer alors le code du div#menu par le code suivant :

```

<nav class="navbar navbar-default">
  <div class="navbar-header">
    <button type="button" class="navbar-toggle" data-toggle="collapse" data-
target="#main_menu">
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
    </button>
  </div>

  <div class="collapse navbar-collapse" id="main_menu">
    <ul class="nav navbar-nav navbar-right">
      <li><a href="#"> Accueil </a></li>
      <li><a href="#"> A propos </a></li>
      <li><a href="#"> Contact </a></li>
    </ul>
  </div>
</nav>

```

On voudrait insérer une icône juste avant le texte « Contact », insérer alors le code suivant :

```
<span class="glyphicon glyphicon-send menu_item"></span>
```

Faire de même avec les 2 autres menu, trouvez la bonne icône depuis la liste suivante :
getbootstrap.com/components/

6 Formulaire de contact

○ Version 1

Dupliquer la page d'accueil et sauvegarder le fichier sous le nom « contact.html » Supprimer le corps de la page et insérer une balise « form »

Bootstrap propose des mises en forme pour les formulaire, il appliquer les principes suivant :

- Le champ ainsi que son libellé doivent être encapsulé par un div ayant la classe «form-group »
- Chaque champ doit avoir la classe « form-control »

Voici le résultat souhaité :

Bloggy

Le blog de la mort qui tue la vie

Contactez moi

E-mail *:

Sujet *:

Message *:

☐ M'inscrire à la newsletter

Envoyer

Sidebar L
exercitatio
aliquip ex

Les 3 premiers champs de saisie sont obligatoires, attention au type du 1^{er} champ.

○ Version 2

Définir la classe «form-horizontal» pour le formulaire (balise « form »)

Ensuite ajouter les classes « control-label » et « col-sm-2 » à tous les libellés (sauf la case à cocher)

Enfin entourer les 3 premiers champs dans un div qui prend le reste de la largeur de la fenêtre.

Ci-contre le résultat souhaité :

Contactez moi

E-mail *:

Sujet *:

Message *:

☐ M'inscrire à la newsletter

○ Améliorations

Le champ e-mail est très important, on va donc le mettre en valeur : Identifier le div contenant le code relatif au 1^{er} champ et appliquer (en ajoutant) la classe « has-feedback ». Juste après le code de l'input ajouter le code suivant :

```
<span class="glyphicon glyphicon-warning-sign form-control-feedback">
```

Voici le résultat final :

Contactez moi

E-mail *:

Sujet *:

7 Détails d'un article

Proposer et développer le gabarit de la page article, elle doit afficher les données suivantes : Date, titre, image principale et texte complet.

Un internaute doit pouvoir commenter un article, dans la fiche article on doit donc prévoir un espace pour afficher la liste des commentaires existant ainsi qu'un formulaire pour ajouter un nouveau commentaire (avec le minimum de champs possible)

Sources :

http://www.w3schools.com/bootstrap/bootstrap_forms.asp

http://www.w3schools.com/bootstrap/bootstrap_forms_inputs.asp