



R 1.02

2023 - 2024

Développement d'interfaces web

**TD n°7
Framework CSS**



ANNE Jean-François

Le but de ce TD est de se familiariser avec la création de pages et de site WEB.

Framework Front-End

A. Normalisation

I. Reset & Normalize

Nous allons parler de deux outils que vous serez sûrement amené à utiliser lorsque vous ferez du CSS : Reset et Normalize. Ces outils permettent de contrebalancer le problème des styles par défaut imposés par les navigateurs (par exemple, les paragraphes ont une marge par défaut, marges qui ne sont pas forcément constantes entre les navigateurs).

1°) Reset

L'objectif de cette approche est de contrer l'apparence que le navigateur impose par défaut à la plupart des élément HTML. Ceci permet ensuite, lorsque l'on travaille sur le CSS, de ne pas avoir à penser aux styles qui pourraient être appliqués par défaut aux éléments. Si par exemple vous souhaitez styliser un titre, vous ne serez pas obligés de retirer les marges avant de travailler. Il existe plusieurs feuilles de styles vous proposant une réinitialisation :

- [Le reset original](#)
- [Minireset](#)

2°) Le reset Original :

L'objectif d'une réinitialisation de la feuille de style est de réduire les incohérences du navigateur dans des domaines tels que les hauteurs de ligne par défaut, les marges et les tailles de police des en-têtes, etc... Les styles de réinitialisation apparaissent assez souvent dans les Framework CSS, et la "réinitialisation Meyerweb" originale a été introduite dans Blueprint, entre autres.

Les styles de réinitialisation donnés ici sont intentionnellement très génériques. Par exemple, il n'y a pas de couleur ou d'arrière-plan par défaut pour l'élément body. Je ne recommande pas particulièrement que vous utilisiez simplement ceci dans son état non modifié dans vos propres projets. Il doit être modifié, édité, étendu et mis au point de manière à correspondre à votre base de réinitialisation spécifique. Remplissez vos couleurs préférées pour la page, les liens, etc.

En d'autres termes, il s'agit d'un point de départ et non d'une boîte noire.

Si vous voulez utiliser mes styles de réinitialisation, n'hésitez pas ! Tout est explicitement dans le domaine public. Vous pouvez récupérer une copie du fichier à utiliser et la modifier selon vos besoins. Si vous êtes plutôt du type copier-coller ou si vous souhaitez simplement un aperçu sur la page de ce que vous obtiendrez, le voici :

```
/* http://meyerweb.com/eric/tools/css/reset/
v2.0 | 20110126 License : none (public domain)
*/

html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
```

```

dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup,
menu, nav, output, ruby, section, summary,
time, mark, audio, video {
    margin: 0;
    padding: 0;
    border: 0;
    font-size: 100%;
    font: inherit;
    vertical-align: baseline;
}

/* HTML5 display-role reset for older browsers */
article, aside, details, figcaption, figure,
footer, header, hgroup, menu, nav, section {
    display: block;
}

body {
    line-height: 1;
}

ol, ul {
    list-style: none;
}

blockquote, q {
    quotes: none;
}

blockquote:before, blockquote:after,
q:before, q:after {
    content: '';
    content: none;
}

table {
    border-collapse: collapse;
    border-spacing: 0;
}

```

3°) Normalize

Cet outil propose une approche différente qui consiste à essayer de normaliser l'apparence des éléments HTML à travers les différents navigateurs. L'objectif est d'avoir le comportement le plus prévisible possible. Cette feuille de style va redéfinir certaines propriétés (par exemple spécifier les marges sur les titres, paragraphes ou autre) afin d'offrir une expérience constante pour les utilisateurs quel que soit le navigateur utilisé.

- [Normalize.css](#)

4°) Quelle approche utiliser ?

On peut alors se demander quelle approche nous devons privilégier dans notre CSS. Comme d'habitude, le choix va dépendre de la situation :

- Si vous avez une maquette spécifique qui définit concrètement comment doivent fonctionner l'ensemble des éléments HTML il vaudra mieux passer par un système de reset pour ensuite pouvoir définir ses propres règles.
- En revanche, si vous n'avez pas de style prédéfinis pour l'ensemble des éléments HTML, il vaudra mieux utiliser Normalize (quitte à redéfinir certaines propriétés pour des éléments spécifiques comme par exemple des titres de premier niveau).

B. -Prefix-Free

Prefix-free est un petit fichier JavaScript (2 Ko) développé par [Lea Verou](#), développeuse Web impliquée dans les standards Open Source, CSS3 et JavaScript. Le rôle de ce script est d'ajouter automatiquement en arrière-plan, , uniquement lorsque cela est nécessaire, des préfixes propriétaires nécessaires à la reconnaissance de certaines propriétés de la spécification CSS3 par les différents navigateurs modernes. -prefix-free vous permet donc d'utiliser uniquement des propriétés CSS non préfixées partout.



II. Mais quel est le problème

Les propriétés CSS3 avancées sont intégrées dans les différents navigateurs de manière non concertée et selon le bon vouloir des développeurs de ces butineurs ! Pour ne pas entraver la prise en compte de ces propriétés non validées par le W3C on a autorisé une syntaxe personnalisée afin de permettre leur intégration progressive. Le choix a été fait d'utiliser un préfixe propriétaire pour identifier les propriétés CSS3 prises en compte par chaque type de navigateur (selon leur moteur de rendu). Les préfixes principaux :

- -webkit- (pour Chrome, Safari,...)
- -moz- (pour Firefox, Flock,...)
- -o- (pour Opera)
- -ms- (pour Internet Explorer)

Les développeurs Web souhaitant d'ores et déjà utiliser ces propriétés CSS sont donc contraints à écrire plusieurs lignes pour que la prise en compte de leur feuille de style se fasse le plus largement possible. Un exemple pour illustrer cela. Ci-dessous la fonction de zoom d'élément (ici doublement de la taille de l'élément ayant pour id `zoom`) si on ne tient compte que de la spécification officielle :

```
#zoom {
  transform: scale(2);
}
```

Pour être certain que cette transformation s'opère dans un maximum de navigateurs, il va falloir ajouter plusieurs lignes supplémentaires :

```
#zoom {
  -moz-transform: scale(2);
  -webkit-transform: scale(2);
  -o-transform: scale(2);
  transform: scale(2);
}
```

III. Utilité de -prefix-free

-prefix-free est un petit fichier JavaScript qui ajoute à la volée le bon préfixe propriétaire selon le navigateur affichant la page. Plus besoin, donc, d'ajouter dans la feuille de style les différentes propriétés propriétaires ! -prefix-free adapte la propriété CSS3 au navigateur. Un exemple, voici la propriété définissant un fond en dégradé :

```
#degrade {
  background: linear-gradient(#96C7CC, #FFF);
}
```

Cette ligne de la feuille de style sera remplacée de la manière suivante pour les utilisateurs de Firefox :

```
#degrade {
  background: -moz-linear-gradient(#96C7CC, #FFF);
}
```

Et ainsi pour les utilisateurs d'Opera :

```
#degrade {
  background: -o-linear-gradient(#96C7CC, #FFF);
}
```

Les avantages de cette technique :

- Les feuilles de styles feront une cure d'amaigrissement non négligeable (une seule ligne pour une même propriété de style).
- Les développeurs web expérimentés apprécieront le gain de temps (pas besoin d'ajouter les lignes avec propriétés préfixées).
- Les débutants en codage CSS3 ne seront pas déboussolés par les pléthores de préfixes.

IV. Comment faire fonctionner -prefix-free

La procédure d'installation de -prefix-free est relativement simple. Placer le fichier à un endroit de votre hébergement, puis insérer la ligne suivante dans la partie <head> de vos pages HTML.

```
<script type="text/javascript" src="chemin/vers/prefixfree.min.js"></script>
```

Ou alors plus simplement (sans héberger le fichier) en appelant le fichier directement à partir de Github :

```
<script type="text/javascript"
src="https://raw.githubusercontent.com/LeaVerou/prefixfree/master/prefixfree.min.js">
</script>
```

Pour minimiser l'effet FOUC - c'est un phénomène décrit par bluerobot.com qui se traduit par l'affichage de la page brute avant celui de la page stylée - il est recommandé de placer la ligne d'appel du fichier JS **juste après la déclaration de la CSS**.

V. Conditions d'utilisation

Comme tout script "patchant", l'utilisation de ce genre d'outil est soumise à certaines conditions...

Avantages :

- Traite les styles CSS appelés par une balise <link> ou contenus dans une balise <style>, mais également ceux des attributs style à l'intérieur des balises.

- Prend en compte les nouveaux éléments `<link>` et `<style>` et les modifications des attributs `style` ([nécessite un plugin](#)).
- Est compatible avec jQuery ([nécessite un plugin](#)).
- Est compatible avec [CSS3 Pie](#) (de manière générale d'ailleurs, ne traite pas les propriétés déjà préfixées).
- Les navigateurs cibles compatibles sont : IE9 et supérieur, Opera 10 et supérieur, Firefox 3.5 et supérieur, Safari 4 et supérieur, Chrome pour Windows et Mac-OS et Arora, Midori, Chromium, Epiphany, Rekonq pour Linux (testés sur Ubuntu 11.10). En règle générale, tout navigateur prenant en charge les propriétés avancées CSS3 devrait être compatible.
- Pour les navigateurs non compatibles avec ce genre de propriétés (IE8 par exemple) et pour les propriétés étant déjà préfixées manuellement, aucun préfixe ne sera ajouté.

Inconvénients :

- Ajoute une couche JavaScript, donc une requête de fichier et des traitements de données du côté client.
- Ne traite pas les styles CSS incorporés par l'intermédiaire de la règle `@import`
- Ne traite pas non plus les CSS provenant d'un autre domaine.
- Pas de traitement des styles incorporés dans les balises (attribut `style`) pour Internet Explorer et Firefox < 3.6.

Pour terminer ce petit tutoriel, [une page de démonstration de -prefix-free](#)

Introduction aux Framework Front End

A. Introduction aux Framework Front-End

Les **Frameworks ou cadriceils** en bon français sont des outils de plus en plus courants dans le processus de création de sites web. On distingue au moins deux types de Frameworks : **front-end et back-end**.

Les Frameworks back-end (comme Symfony, Laravel ou CakePhp) sont déjà utilisés depuis plusieurs années avec succès, mais qu'en est-il des Frameworks front-end ?

Depuis quelques années, ces Frameworks sont devenus presque incontournables. Nous allons découvrir, ce qu'est un Framework front-end, pourquoi en utiliser, identifier les différents Framework et enfin en analyser les limites.

I. Qu'est-ce qu'un Framework front-end ?

Un Framework, c'est une base d'outils et de fonctionnalités conçus pour s'agencer entre eux. En partant d'une grille, qui est la plupart du temps prévue pour être responsive, vous pouvez ajouter facilement des éléments qui ont été maintes et maintes fois testés par d'autres développeurs et dont la plupart des sites ont besoin (des comptes utilisateurs, des authentifications, des connexions à diverses sources de données, des moteurs de recherche interne, des boutons de partage sur les réseaux sociaux...). Le tout sans avoir besoin de les coder vous-même, ce qui représente un gain de temps considérable.

Par ailleurs, le Framework est automatiquement mis à jour pour s'adapter aux nouvelles pratiques du net (comme ce fut le cas pour le passage aux sites responsive, notamment). Concrètement, c'est **un ensemble de fichiers et dossiers** livré de façon structurée ; le tout est codé en respectant les standards actuels et fonctionne uniformément sur toutes les plateformes et navigateurs actuels. Ce pack de fichiers fournit un lot de comportements et d'éléments couramment utilisés dans un site web.

Il y a 3 types de fichiers dans le Framework : **HTML, CSS et JavaScript**.

Le langage HTML **structure** les données, le **CSS stylise** ces données pour les rendre visuellement compréhensibles et esthétiques, le Javascript permet de produire des **animations et interactions** améliorant l'expérience utilisateur.

1°) Le responsive design et la grille



Illustration d'une page web au contenu identique se réorganise automatiquement en fonction du support utilisé.

Lorsqu'une page web est conçue pour réorganiser son contenu de façon optimale pour différentes tailles d'écran, il s'agit alors de **responsive design**.

Un site devient ainsi accessible sur un ordinateur classique (desktop), sur une tablette ou encore un téléphone mobile (type iPhone, Android) tout en réorganisant un contenu unique, certains blocs de contenus seront sur deux ou trois colonnes, cacher ou modifier selon le terminal utilisé.

La grille régit le contenu sous forme de colonnes. Les blocs de contenu seront réorganisés afin de s'adapter à la taille de l'écran et d'offrir une expérience utilisateur optimale.

2°) «Le Framework front-end répond au besoin de chacune des composantes d'un site web»

Maintenant que nous avons découpé clairement les composantes d'une page web classique, il est plus aisé de comprendre ce que peut apporter un Framework front-end.

Un Framework propose des éléments préformatés et paramétrables pour chacune des composantes identifiées ci-dessus :

- Un layout complet et manipulable
- Des modules couvrant tous les aspects et besoins courants
- Des états standards
- Des comportements interactifs multiples
- Un design responsive paramétrable

Le Framework est en d'autres termes un "kit de démarrage front-end" composé de briques personnalisables. On peut aisément sortir de l'aspect graphique par défaut proposé par le Framework.

II. Les principaux acteurs

Il existe actuellement une multitude de Framework front-end.

- | | |
|----------------------------|---------------------|
| 1. <i>Bootstrap.</i> | 7. <i>UIKit.</i> |
| 2. <i>Foundation.</i> | 8. <i>PURE.</i> |
| 3. <i>Skeleton.</i> | 9. <i>Miligram.</i> |
| 4. <i>Semantic UI.</i> | 10. <i>Susy.</i> |
| 5. <i>Materialize CSS.</i> | 11. <i>Bulma</i> |
| 6. <i>Material UI.</i> | |

III. Bien choisir son Framework

Si le fonctionnement des différents Framework présents sur internet est relativement similaire, les disparités sont suffisamment fortes pour nécessiter de choisir correctement son Framework de départ. Car changer de Framework implique de s'initier à tout un nouveau langage, à de nouvelles interfaces, à des modularités différentes... Bref, cela demande un temps certain que vous n'aurez peut-être pas envie de perdre pour chaque nouveau site à mettre en place.

3°) Bootstrap.

B Bootstrap

Le Framework Bootstrap, créé pour Twitter en 2011, compte parmi les plus populaires. Open Source, il présente un nombre important de modules et d'extensions. La documentation disponible est tout aussi impressionnante. Il fonctionne sur un système de grille en douze colonnes avec un affichage adapté pour différentes tailles d'écran et il est possible de personnaliser les fichiers CSS. Il est le Framework le plus utilisé dans le monde. Il utilise des éléments de CCS, HTML, JQuery et Javascript.

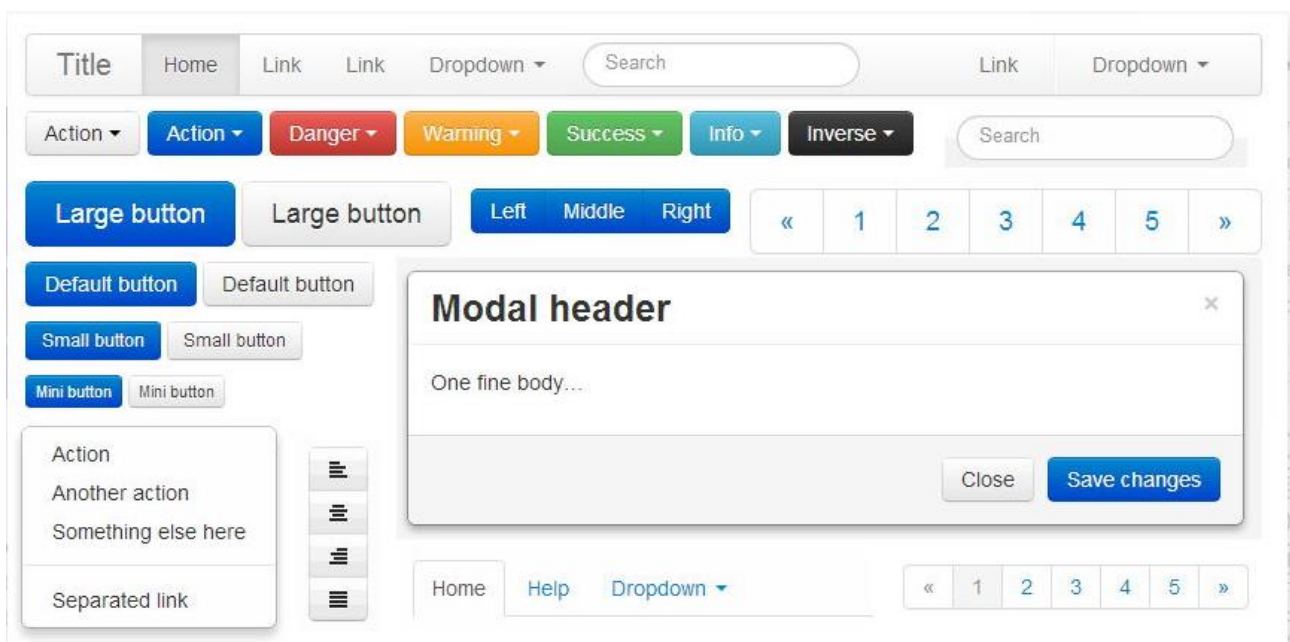
Avantages :

- Permet le Responsive Design.
- Une très grande communauté, permettant d'obtenir de l'aide ou consulter des documentations très facilement.
- Mis à jour régulièrement et toujours aussi populaire.
- Version téléchargeable modulable afin d'inclure ou non certains paquets, possibilité également depuis le fichier config.
- Supporte SASS et SCSS.
- Flexible.

Inconvénients :

- Plus lourd que les autres Framework CSS.
- Demande à être configuré pour ne pas se retrouver avec un site web des plus banals.

Exemple :



© <https://www.slant.co/options/521/alternatives/~materialize-alternatives>

4°) Foundation.



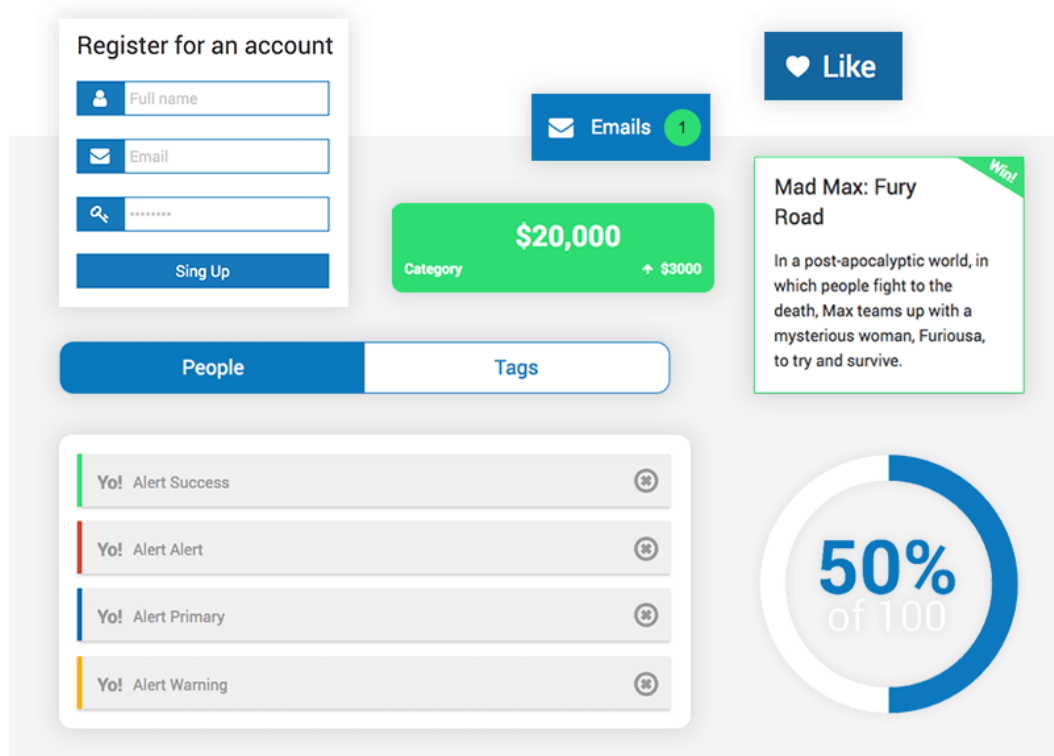
Framework Open Source géré par ZURB. Il dispose d'une grande flexibilité. Il est proposé avec un fichier démo utile pour les débutants et dispose de son propre centre de formation. Il est cependant légèrement plus complexe d'utilisation.

Avantages :

- Du CSS par défaut pour ne pas avoir à démarrer de zéro.
- Moins populaire que Bootstrap, ce qui permet un style moins identifiable de site Web.
- Une grille flexible, permettant d'appliquer plus de personnalisation dans les déclarations de classes.
- Une bonne documentation en ligne, couplée à une aide et des cours proposés de la part du service technique de Foundation.
- Supporte le SCSS.
- Permet le responsive.

Inconvénients :

- Ne supporte pas le SASS.
- Demande beaucoup d'efforts pour une customisation du site complète.
- Grille de base peu modulable.



© <https://assets.justinmind.com/wp-content/uploads/2017/09/foundation-framework-ui-elements-prototype-websites.png>

5°) Skeleton.



Framework créée par Dave Gamache pour les besoins de petits projets. comme **Milligram**. Ce sont des Frameworks très légers et simples à utiliser. Ils seront pourtant réservés à la création de sites qui ne demandent pas un trop grand nombre de fonctionnalités.

Avantages :

- Framework le plus simple d'approche.
- Choix le plus approprié pour de petits projets.
- Code épuré et très clair.
- Un très bon responsive.

Inconvénients :

- Prend davantage de temps à développer pour une customisation plus complète.
- Un manque cruel de CSS intégré par défaut.
- Templates limités de par la simplicité de la grille proposée.
- Non adapté au Prototypage dû à son peu de flexibilité.

6°) Semantic UI.



Framework Open-source se voulant comme son nom l'indique moins programmatique, et davantage naturel. . Son principe repose sur l'utilisation d'un langage naturel, accessible aux néophytes, qui permet d'obtenir un code HTML facile à lire et à comprendre.

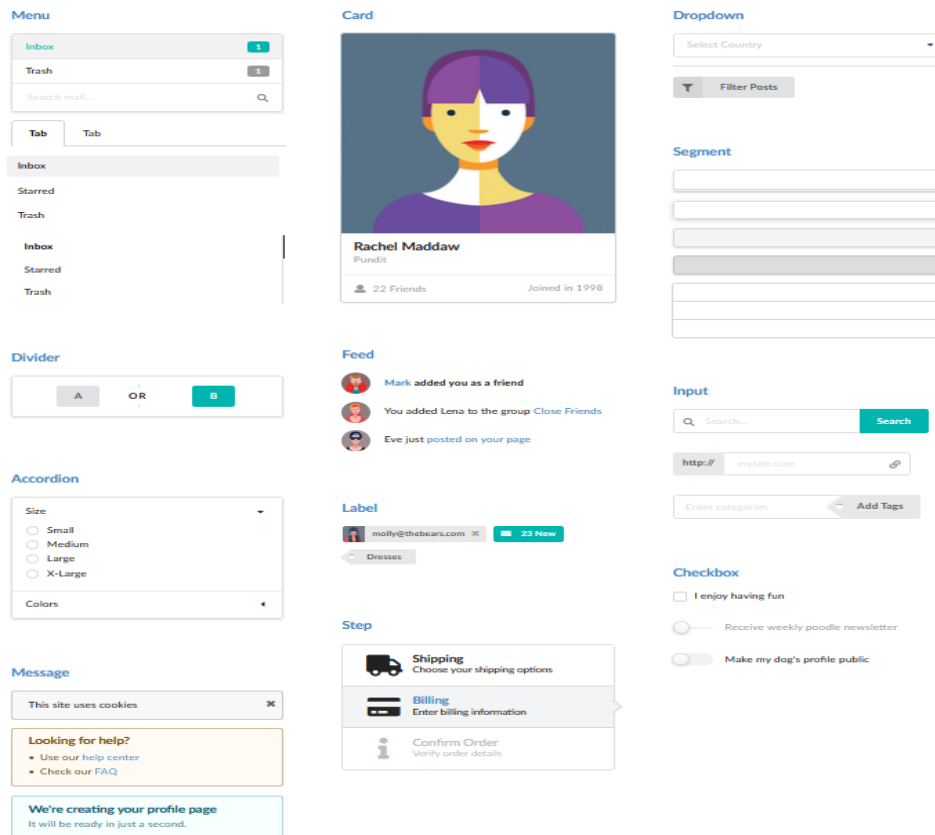
Avantages :

- Léger.
- Esthétique des contrôles.
- Large palette de couleurs de base.
- Fluidité des menus.
- Contenus intégrés, par exemple les drapeaux.
- Prend en compte le responsive.
- Noms des classes intuitifs ; compréhensibles.

Inconvénients :

- Contenus intégrés, (Version de Font Awesome obsolète par exemple).

- Manière de coder très stricte.



© <https://semantic-ui.com/>

7°) Materialize.



Framework Open source édité par Google, axé Material design et inspiré de Bootstrap. Il s'appuie sur les recommandations design de Google et est fourni avec de nombreux composants prêts à l'emploi.

Avantages :

- Inclut une série de composants JavaScript permettant de réaliser une multitude d'effets additionnels comme les vagues sur les boutons ou encore des effets parallaxe.
- Code simplifié sur les formulaires et Navbars.
- Facile à prendre en main.
- Supporte le SCSS et le SASS.
- Flexible.
- Très léger.
- De très bonnes documentations.

- Permet le responsive.

Inconvénients :

- Mauvais support du SASS.
- Dans certaines situations la grille a du mal à s'adapter.

8°) Material UI.



Material UI : Framework CSS basé inspiré du Material Design de Google. Il est également basé sur les principes du Material Design de Google, mais il sera d'un abord plus complexe, d'autant qu'il nécessite de comprendre au préalable React.js pour pouvoir l'utiliser.

Avantages :

- Hautement customisable.
- Large palette de couleurs.
- Le meilleur moyen de se familiariser avec le Material design de Google.

Inconvénients :

- Demande des connaissances en ReactJS pour être correctement utilisé.
- N'est pas destiné à être utilisé comme point de départ pour les projets de web design complet, nécessite de le coupler à un autre Framework.

9°) uikit.



Framework Open-source. C'est un Framework créé par YOOftheme, un studio de création de thèmes pour WordPress et de Templates pour Joomla! très réputé. Ce Framework responsive nous propose tous les composants attendus d'un Framework :

- Une grille de mise en page responsive.
- Des composants pour afficher des panneaux, des articles structurés, des commentaires...
- Des barres de navigation, des fils d'Ariane, des éléments de pagination et des composants à onglet.
- Des éléments de mise en forme pour le texte : des listes, des tableaux élaborés et des formulaires.
- Des éléments graphiques avec des mises en forme pour les images, les boutons, les fenêtres d'alerte, les éléments animés...

- Des composants interactifs en JavaScript : des menus déroulants, des fenêtres modales, des bulles d'aide, des boutons à bascule...

Avantages :

- Hautement customisable du fait de la possibilité de combiner les modules.
- Permet d'ajouter des composants au CSS sans impacter le style général.
- Possibilité de créer des interfaces utilisateurs poussées.

Inconvénients :

- Framework récent, très peu de ressources et de documentations.

10°) Pure.



Framework CSS édité par l'équipe de développement de Yahoo. Il est plus concentré sur la création de sites rapides et responsive, il ne supporte pas les plugins en JS ou JQuery.

Avantages :

- Très léger.
- Se charge extrêmement vite sur les appareils mobiles.
- Très flexible en milieu CSS.
- S'adapte très bien à n'importe quel type de projet.
- Permet un responsive simple et efficace.

Inconvénients :

- Framework exclusivement CSS, n'inclut pas de plugins JQuery ou Javascript.

11°) Milligram.



Framework Open-source similaire à Skeleton. Ce sont des frameworks très légers et simples à utiliser. Ils seront pourtant réservés à la création de sites qui ne demandent pas un trop grand nombre de fonctionnalités.

Avantages :

- Très léger, seulement 2kb.
- Utilise Flexbox CSS comme système de grille.

Inconvénients :

- Tout comme Ulkit celui-ci manque de ressources et de documentations, dû à sa récente apparition.
- Peu de composants de style disponibles comparés aux autres framework.

12°) Susy.



Framework destiné à la création de grille.

Avantages :

- Permet la création complète de la grille de par sa flexibilité.
- Se charge d'effectuer les calculs.

Inconvénients :

- Ne prend pas en compte tous les aspects du Web design et requiert l'utilisation conjointe d'un autre Framework.
- Aucune grille définie par défaut.

13°) Bulma.



L'un des avantages de Bulma est qu'il est purement CSS, basé sur Flexbox, pas de JavaScript. Cela signifie que vous devez simplement inclure un fichier .css dans votre projet pour pouvoir démarrer, aucun fichier .js requis.

Avantages :

- Entièrement responsive et orienté mobile.
- Modulaire, moderne, construit avec Sass, utilisez uniquement ce dont vous avez besoin.
- Gratuit, Open Source.

Inconvénients :

- Jeune.

14°) Conclusion.

Libre à vous d'utiliser bien évidemment le Framework qui correspond le mieux à votre projet. Toutefois, il est important en tant que débutant de prendre en compte qu'une bonne documentation est toujours appréciée et qu'un Framework populaire est peut-être le meilleur choix pour s'y aventurer. En effet, beaucoup de personnes ont déjà répondu

sur les forums à la plupart des problèmes rencontrés et de larges communautés axées sur l'entraide existent.

Mais nous allons nous attarder sur les deux Framework les plus populaires : **Bootstrap** de **Twitter** et **Foundation** de **Zurb**.

15°) Bootstrap de Twitter

Il est le plus connu et le plus utilisé. Certainement boosté par la notoriété de Twitter qui a conçu son propre Framework et a choisi de le mettre à disposition du public. Bootstrap est ainsi né fin 2011, depuis sa notoriété n'a cessé de grandir, en raison de sa simplicité d'implémentation et de la diversité de ses composants.



Bootstrap est fourni avec des fichiers CSS, JS et une police. Il est également possible de télécharger des fichiers HTML d'exemple afin de découvrir ce dont est capable le Framework.

Bootstrap propose :

- une grille responsive, des styles CSS pour tous les éléments HTML standards (formulaire, bouton, texte, image ...)
- des composants JS comme des listes déroulantes dynamiques, des navigations interactives ou un carrousel
- une police qui est en fait un set d'icônes. L'avantage est que les icônes n'étant pas des images mais des polices, elles sont vectorisées et donc toujours nettes sur tout type d'appareils (on pense notamment aux appareils mac disposant d'un écran Retina).

Il est possible de coupler Bootstrap avec le préprocesseur de CSS appelé [LESS CSS](#). Cet outil permet de créer des fichiers CSS encore plus rapidement grâce à un langage de programmation bien plus dynamique et intelligent que le CSS standard. Ceci est une option, mais Bootstrap couplé à LESS devient un ensemble d'outil encore plus performant.

À noter que Bootstrap est également compatible avec le préprocesseur [SASS](#) (qui se comporte de façon identique à LESS).

Une des options proposées par Bootstrap est de pouvoir personnaliser les couleurs et les aspects graphiques du Framework avant même de le télécharger.

Bootstrap est mis à jour régulièrement, au rythme moyen d'une nouvelle version majeure tous les 12 à 18 mois.

[En savoir plus sur Bootstrap](#)

[Tous les composants de Bootstrap](#)

16°) Foundation de Zurb

Il est moins populaire que Bootstrap, car un peu plus pointu. Mais Foundation est en tout point semblable à Bootstrap.



Foundation propose :

- une grille responsive, des style CSS pour les éléments HTML courants
- des composants JS d'ailleurs un peu plus fournis que sur Bootstrap

Par contre pas de set d'icônes fourni par défaut. Il est de toute façon très simple d'en insérer un.

Il est possible de consulter de [nombreux Template de pages](#) conçus avec Foundation. Ce Framework quant à lui, n'est pas compatible avec le préprocesseur de CSS appelé [LESS CSS](#). Foundation supporte uniquement [SASS](#).

Dans l'ensemble Foundation est un outil qui semble plus complet mais moins évident à prendre en main que Bootstrap. Il s'adresse aux [développeurs front-end](#) (ou [intégrateur web](#)) plus pointus. Foundation semble également plus à même de répondre pleinement à des projets web d'envergure demandant plus de souplesse.

[En savoir plus sur Foundation](#)

[Tous les composants de Foundation](#)

IV. Pourquoi utiliser un Framework front-end en 5 points

Un Framework propose de multiples avantages :

1°) Gain de temps

Prenons un cas concret. Un nouveau projet arrive sur la table, le développeur front-end doit évaluer le temps de développement mais souvent le temps presse. Alors le Framework front-end peut être une bonne alternative. Car le développeur n'aura pas à partir de zéro. Plutôt que de coder et créer l'ensemble des composantes de la page, il aura la tâche d'assembler les éléments préformatés et proposés par le Framework. C'est comme si l'on éliminait une étape dans le processus de travail du développeur front-end.

On élimine l'étape zéro, celle où le développeur front-end part de rien ou presque pour créer le site web au complet.

2°) Standardisation et performance

Un site développé en utilisant un Framework sera performant car le Framework est optimisé pour les multiples navigateurs web du marché. Plutôt que d'un site performant on parlera plutôt d'un site optimal, ceci sera expliqué dans les limites du Framework front-end.

3°) Évolutivité et mise à jour

Le Framework est régulièrement mis à jour afin de respecter les derniers standards du web mais aussi afin de proposer des modules pertinents et dans l'air du temps. En utilisant un Framework, on bénéficie d'un ensemble d'améliorations bénéfiques pour la construction et la mise en place d'un site web devant respecter les derniers standards en vigueur.

4°) Gratuit et libre

Les Frameworks front-end sont (au moins actuellement) gratuits et libres. Cela signifie qu'il est possible de les utiliser sans payer de licence et de pouvoir les modifier à son gré. C'est du moins le cas pour Bootstrap et Foundation.

5°) Fiabilité

Les Frameworks les plus populaires par des centaines de milliers de développeurs qui attestent de la qualité et de la fiabilité de l'outil. Les bugs sont presque inexistants.

En bref, utiliser un Framework permet d'éliminer une étape parfois laborieuse qui est celle de concevoir les bases visuelles du site et de créer les différents modules de zéro. En plus de gagner du temps, il fournit une structure éprouvée, solide, et évolutive testée par des milliers d'utilisateurs.

V. Les limites du Framework front-end en 5 points

1°) Des sites qui se ressemblent

Une critique récurrente rencontrée sur nombre de forums spécialisés vise l'aspect visuel similaire qu'ont les sites développés avec un Framework front-end. Effectivement, si le développeur ne personnalise pas un minimum l'aspect visuel du site, alors le style par défaut (les couleurs, les angles arrondis, les contours, les ombres, la typographie) sera appliqué. Donc, rien ne ressemble plus à un site Bootstrap avec le style par défaut qu'un autre site Bootstrap avec le style également par défaut.

2°) Du code inutilisé qui alourdit les pages

Un Framework fournit un lot d'éléments qui ne sont parfois pas utilisés dans un projet web. Or, même si l'on n'utilise pas certains composants du Framework, ils sont tout de même chargés par défaut, notamment leurs styles CSS et également les fichiers JS. On se retrouve avec code lourd et partie inutile. Les effets négatifs sont notamment que les pages web se chargeront moins rapidement, et que le développeur en charge de faire évoluer le projet aura parfois du mal à s'y retrouver. Une bonne pratique veut que le développeur qui démarre le projet tâche de rendre muet les appels vers des composants pas nécessaires et qu'il commente et documente un maximum le code généré.

3°) Nécessité de modifier ou ajouter du code

Le Framework n'a pas réponse à tous les cas de figures. Et il est quasiment certains que le développeur devra modifier ou créer des segments de code supplémentaires pour se rapprocher du design souhaité. Cependant il y a un ensemble de paramètres faciles à modifier au cours du projet, pour peu que l'on utilise un préprocesseur CSS.

4°) Modifier ou adapter le code originel du Framework peut être long voire laborieux

Lorsque cela est nécessaire, la modification du code ou l'adaptation de celui-ci peut s'avérer laborieux, car on tente de défaire une architecture en imposant de nouveaux critères. Et il n'est pas toujours simple et rapide de modifier des éléments dont le code mêle à la fois du CSS et du HTML et parfois même du JS.

5°) Risque de conflits avec d'autres bibliothèques ou plugins

Sur un projet mêlant un CMS comme par exemple WordPress et des plugins comme par exemple Visual Composer (qui permet d'éditer rapidement le contenu des pages). Donc sur ce type de projet, il risque d'y avoir des conflits notamment en raison de déclarations [CSS](#) utilisant les mêmes noms de classes ou utilisant des styles CSS peu compatibles avec le CSS du Framework front-end choisi. Dans ce cas il faudra non seulement modifier ou adapter le code du Framework mais également apporter des modifications au plugin posant conflit. Dans tous les cas, la solution ne sera pas simple.

VI. Conclusion

Le Framework front-end est un outil de plus dans la panoplie du développeur front-end. Il est très pratique pour créer des wireframes, des prototypes ou des maquettes simples dans le cadre de projet web léger. Le temps économisé au début du projet est indéniable. Il faut connaître les limites du Framework avant de le choisir pour un projet web plus conséquent. L'idéal est que les designers et wireframers qui imaginent et conçoivent la structure et l'apparence du site web connaissent le Framework afin de proposer des wireframes et des maquettes graphiques compatibles. Dans tous les cas, l'utilisation d'un Framework pose un cadre de travail standardisé en suivant des règles préétablies. Cela permet de transmettre le projet d'un développeur à un autre en minimisant le temps de réappropriation. Enfin, le Framework tire sa pleine puissance en étant utilisé en combinaison avec un préprocesseur CSS (LESS ou SASS). Le code créé sera encore plus modulaire et facilement modifiable.

Premiers pas avec le Framework Bootstrap (CSS)



I. Présentation de Bootstrap

Bootstrap (<http://getbootstrap.com/>) est un Framework destiné à faciliter la création d'applications Web. Il regroupe une collection d'outils fournis sous la forme de classes CSS et de bibliothèques JavaScript. Bootstrap a été créé par deux développeurs du réseau social Twitter.

A. Prise en main de Bootstrap

1°) Installation

Pour pouvoir utiliser Bootstrap, il est nécessaire que le Framework soit accessible pour nos pages

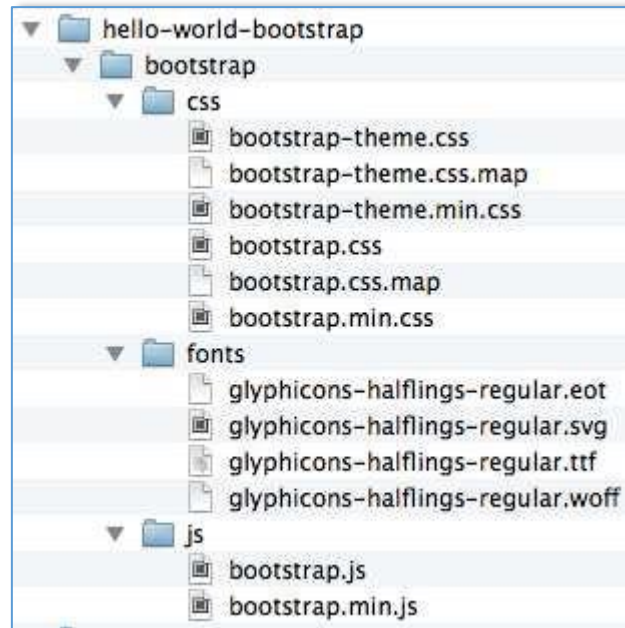
Web. Pour cela, on peut y accéder en ligne via un CDN (*Content Delivery Network*) ou bien l'installer localement. Nous allons utiliser cette seconde solution.

Créez dans votre répertoire personnel un sous-répertoire nommé [hello-world-bootstrap](#) qui sera le répertoire racine de notre exemple. Dans ce répertoire, créez un sous-répertoire [Bootstrap](#).

Ensuite, téléchargez la version courante de Bootstrap sur cette page :

■ <https://getbootstrap.com/docs/4.3/getting-started/download/>

Ensuite, décompressez le contenu de l'archive dans le répertoire [Bootstrap](#) créé précédemment. Vous devez obtenir l'arborescence ci-dessous.



On observe que le Framework se compose de définitions CSS et JavaScript ainsi que d'images compressées. Les fichiers contenant le mot min correspondent aux versions minimales des sources, compressées mais peu lisibles pour le développeur. On les utilise souvent en production afin de réduire le volume de données nécessaire pour afficher une ressource Web.

VII. Une première page Web

Créez dans le répertoire `hello-world-bootstrap` un fichier HTML nommé `index.html`. Recopiez le code ci-dessous dans ce fichier.

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <title>Hello world with Bootstrap</title>
  </head>
  <body>
    <h1>Hello world!</h1>

    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script
>
    <script src="bootstrap/js/bootstrap.min.js"></script>
  </body>
</html>
```

Il s'agit d'une page Web classique avec quelques éléments spécifiques :

- une balise `<link>` vers la feuille de style Bootstrap ;
- une balise `<meta name="viewport">` définissant le viewport, c'est-à-dire la surface de la fenêtre du navigateur qui affiche la page. Cela va permettre à la page de s'adapter à la dimension du terminal client ;

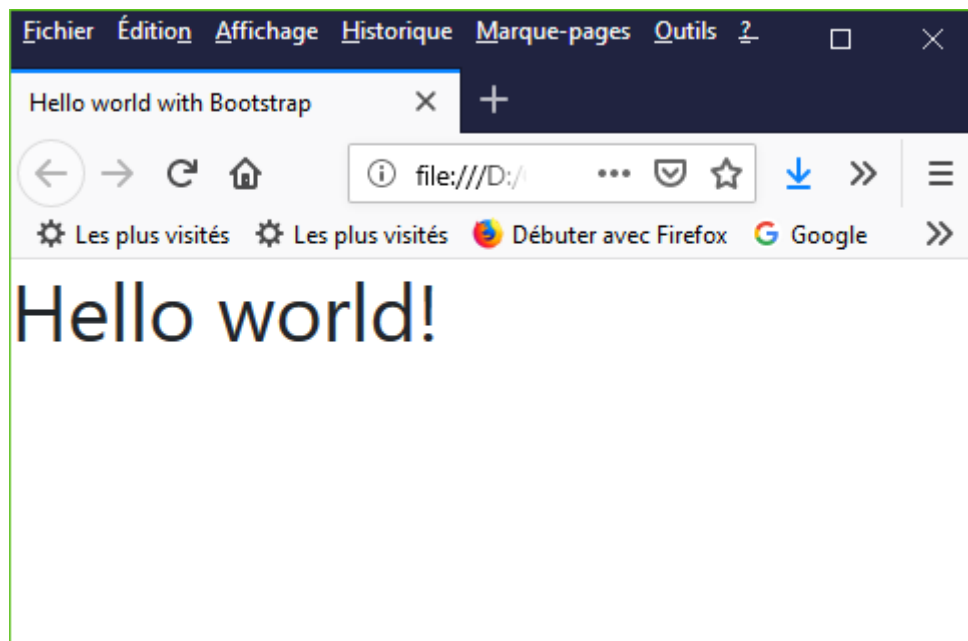
- une balise <meta> permettant un affichage correct sur le navigateur Internet Explorer ;
- des balises <script> qui activent les plugins jQuery de Bootstrap. Ces plugins sont nécessaires pour animer les composants Bootstrap.

Remarque

jQuery est ici inclus en ligne via le CDN Google. Il est également possible de le télécharger et de l'installer localement.

Ouvrez cette page dans votre navigateur Web favori. Vous obtenez le résultat ci-dessous.

Rendu desktop :



B. Principe de fonctionnement

Bootstrap est avant tout une bibliothèque CSS qui fournit plusieurs classes CSS prédéfinies. Il suffit de réutiliser ces classes pour profiter des fonctionnalités de Bootstrap.

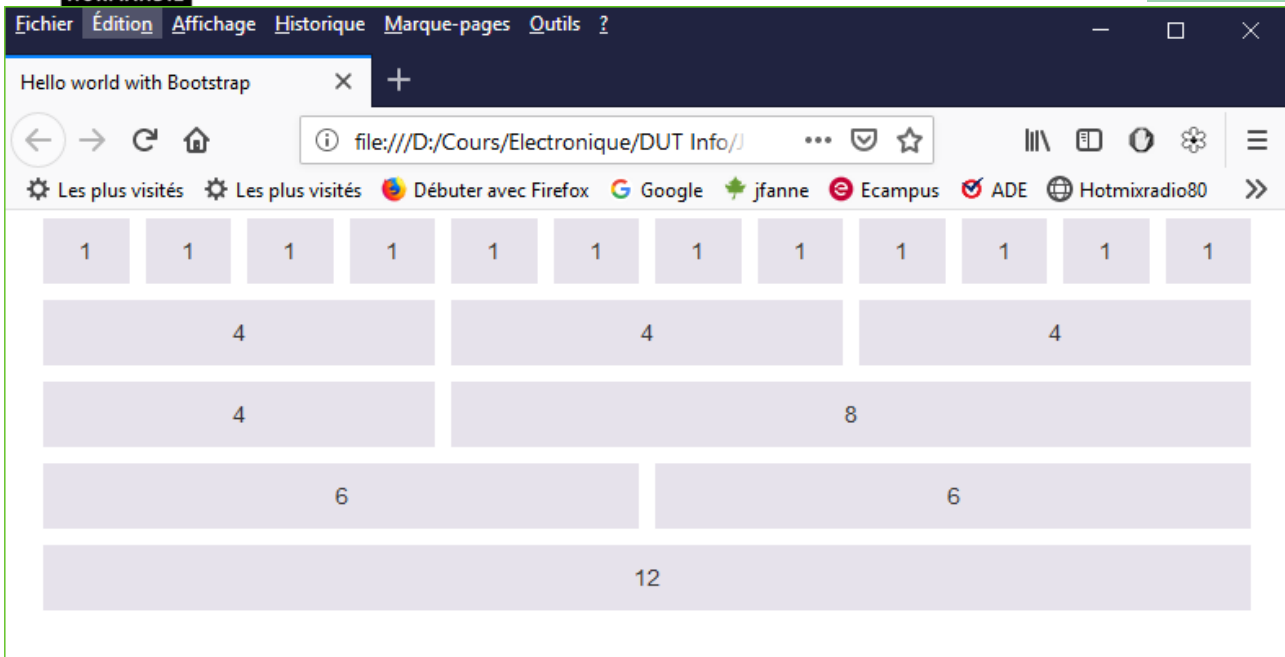
1°) Le conteneur et la grille Bootstrap

Bootstrap fournit une classe CSS **container** utilisée pour regrouper d'autres éléments.

```
<div class="container">
  <!-- ... -->
</div>
```

A l'intérieur d'un **container**, Bootstrap gère la zone d'affichage sous la forme d'une **grille** de 12 colonnes. La classe Bootstrap **row** crée une **ligne** dans cette grille, et chaque classe Bootstrap **col-*** crée une **cellule** qui regroupe * colonnes dans cette ligne.

Dans chaque ligne, la taille totale des colonnes doit être égale à 12. On peut donc choisir de diviser la surface d'affichage en 12 colonnes de taille 1, ou bien en 3 colonnes de taille 4, ou encore en une colonne de taille 4 et une colonne de taille 8.



VIII. Démonstration

Créez dans le répertoire [hello-world-bootstrap](#) et un fichier CSS [style.css](#) contenant le code ci-dessous.

```
.row {
  margin-bottom: 15px;
}

.row [class^="col-"] {
  padding-top: 10px;
  padding-bottom: 10px;
  background-color: #eee;
  border: 1px solid #ddd;
  background-color: rgba(86,61,124,.15);
  border: 1px solid rgba(86,61,124,.2);
}
```

Ce fichier permettra de mieux visualiser les éléments de la grille Bootstrap.

Ensuite, éditez le fichier [index.html](#) comme indiqué ci-dessous.

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <link href="style.css" rel="stylesheet">
  <title>Hello world with Bootstrap</title>
</head>
<body>
  <div class="container">
    <h1>Hello world!</h1>
    <div class="row">
      <div class="col-md-3">
        Zone de gauche
      </div>
```

Développement d'interfaces web

```

<div class="col-md-7">
  Zone du centre
</div>
<div class="col-md-2">
  Zone de droite
</div>
</div>
</div>

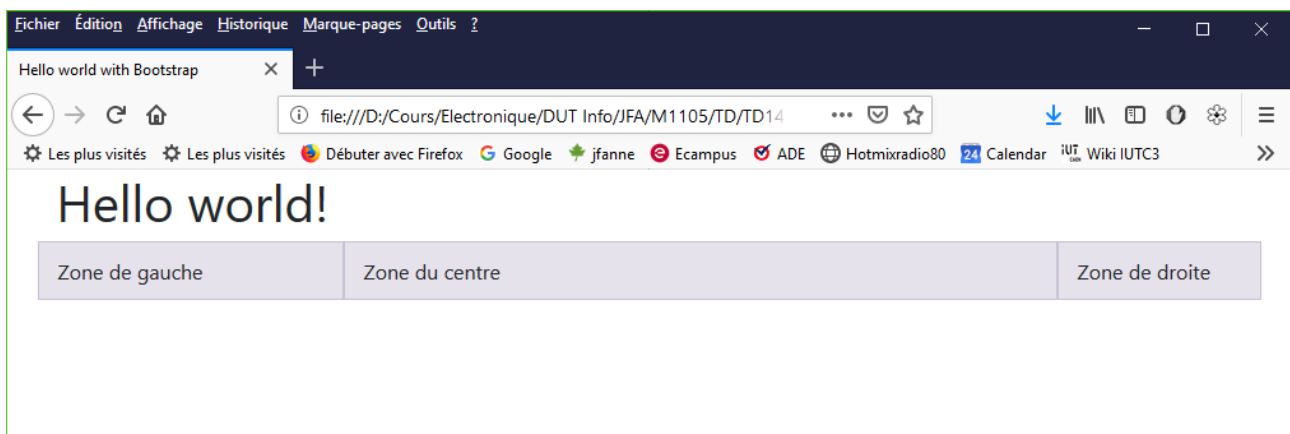
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></
script>
  <script src="bootstrap/js/bootstrap.min.js"></script>
</body>
</html>

```

On remarque au passage que le total des tailles des colonnes (3+7+2) est bien égal à 12. Ouvrez cette page dans un navigateur Web. Le placement des colonnes s'adapte à la surface d'affichage disponible.

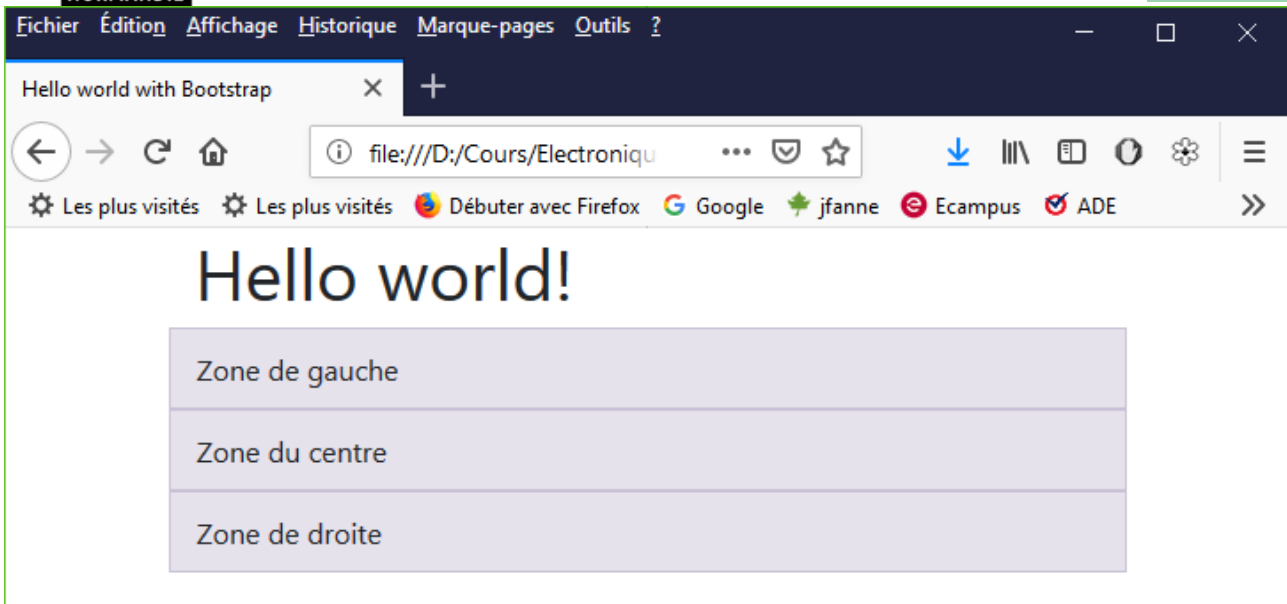
Voici le rendu avec une surface d'affichage importante.

Rendu desktop :



Rendu tablette ou smartphone :

Voici le rendu lorsque la surface d'affichage diminue.



IX. Bootstrap et le design Web adaptatif

D'où vient le comportement que l'on vient d'observer ? Depuis sa version 3, Bootstrap est pensé d'abord pour les terminaux mobiles dont la surface d'affichage est limitée. Il adapte automatiquement la disposition des éléments à la zone d'affichage, détectée grâce à la balise `<meta name="viewport">` vue plus haut.

Pour chaque taille de colonne (de 1 à 12), Bootstrap fournit quatre classes `col-*` dont le comportement diffère en fonction du terminal d'affichage utilisé pour afficher la page Web.

| | Smartphones | Tablettes | Desktop | Desktop large |
|--------------------------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Limite de largeur d'affichage | < 768px | >= 768 px | >= 992 px | >= 1170 px |
| Préfixe des classes colonne | <code>col-xs-*</code> | <code>col-sm-*</code> | <code>col-md-*</code> | <code>col-lg-*</code> |
| Placement des colonnes | Horizontal | (*) | (*) | (*) |

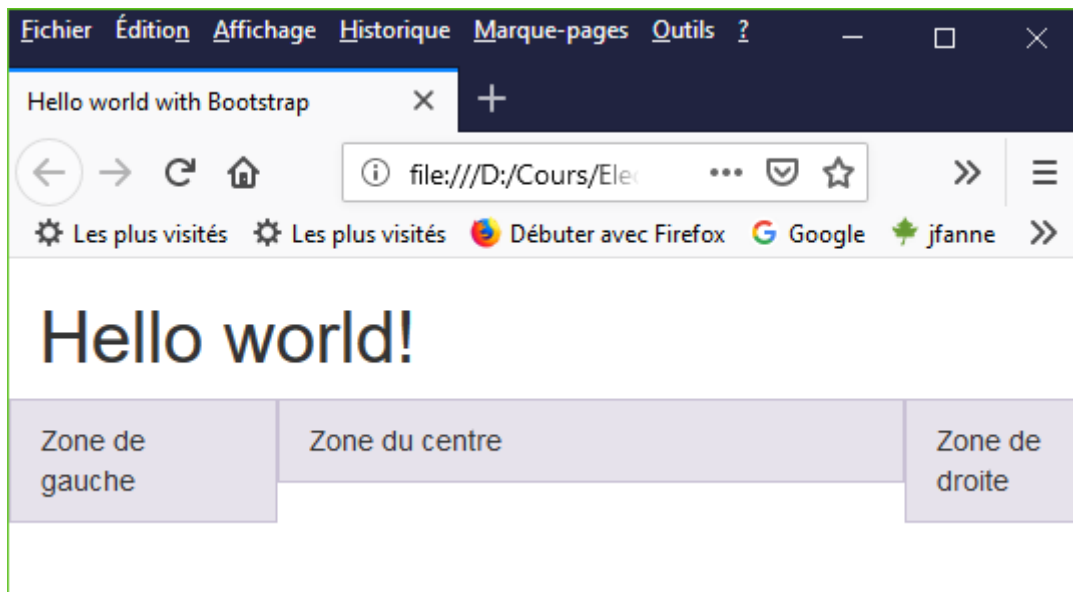
(*) : Les colonnes seront placées horizontalement (côte à côte) uniquement si la largeur d'affichage est suffisante.

Par exemple, la classe Bootstrap `col-md-6` désigne une colonne de taille 6 affichée horizontalement uniquement si la largeur d'affichage est au minimum celle d'un écran de bureau (desktop). Dans le cas contraire, la colonne sera placée verticalement, c'est-à-dire positionnée sous les autres.

Dans notre exemple, nous avons utilisé des colonnes desktop (`col-md-*`). Le tableau ci-dessus montre que ces colonnes ne sont placées horizontalement que si la largeur d'affichage est supérieure ou égale à 992px. Dans le cas contraire, elles se superposent verticalement. Cela explique le comportement observé.

Modifiez le fichier `index.html` et remplacez les classes `col-md-*` par des classes `col-xs-*` de même taille. Ouvrez ce fichier dans un navigateur. Quelle que soit la surface d'affichage, les colonnes restent placées horizontalement.

Rendu :



Comme indiqué dans le tableau ci-dessus, les colonnes `col-xs-*` ne sont jamais superposées verticalement.

X. Aller plus loin

Il est possible de contrôler plus finement l'organisation de la page Web en combinant plusieurs classes colonne. Imaginons par exemple qu'on souhaite garder notre design actuel pour un affichage sur écran de bureau, mais agrandir la zone de gauche et diminuer la taille de la zone centrale pour un affichage sur tablette. Voici le code à modifier dans le fichier [index.html](#).

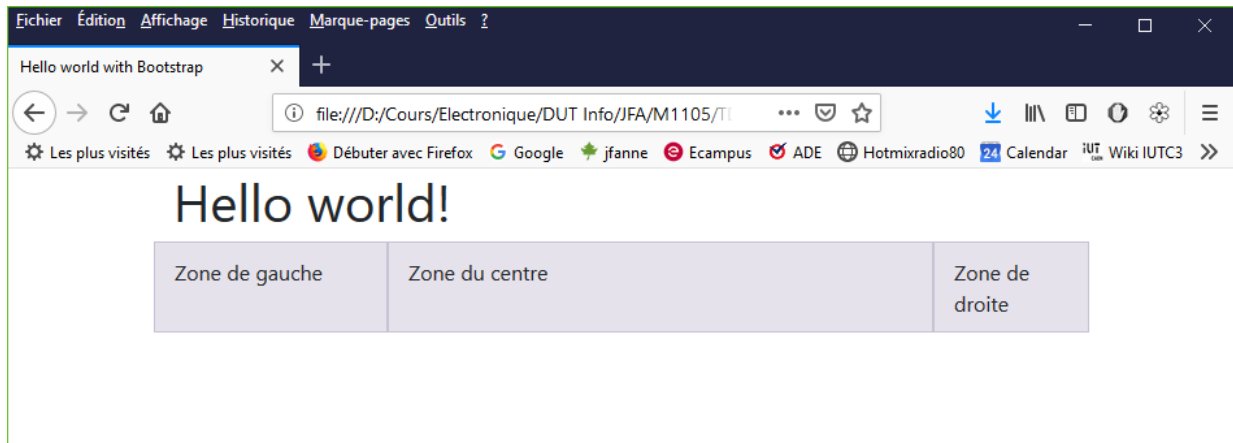
```

<!-- ... -->
<div class="row">
  <!-- Taille 4 sur tablettes, 3 sur desktop et plus -->
  <div class="col-sm-4 col-md-3">
    Zone de gauche
  </div>
  <!-- Taille 6 sur tablettes, 7 sur desktop et plus -->
  <div class="col-sm-6 col-md-7">
    Zone du centre
  </div>
  <!-- Taille 2 sur tablettes et plus -->
  <div class="col-sm-2">
    Zone de droite
  </div>
</div>
<!-- ... -->

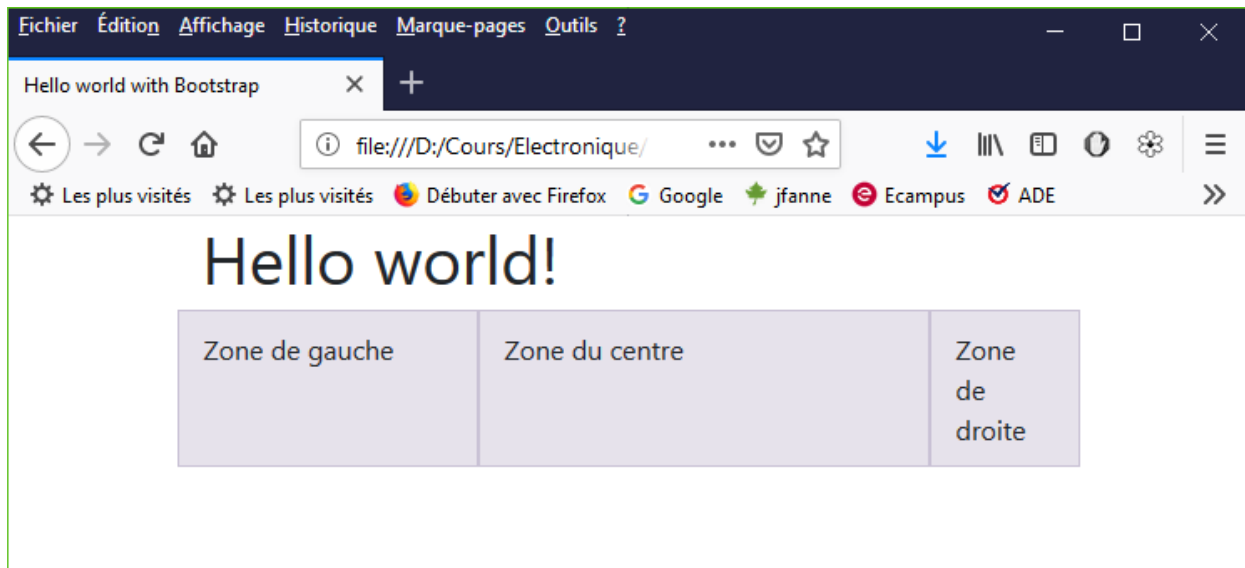
```

Modifiez [index.html](#) puis affichez-le dans un navigateur. On constate que la colonne de gauche passe de 25% à 33% de la largeur totale lorsque la surface d'affichage correspond à celle d'une tablette.

Rendu desktop :



Rendu tablette :



Rendu smartphone :



Remarque

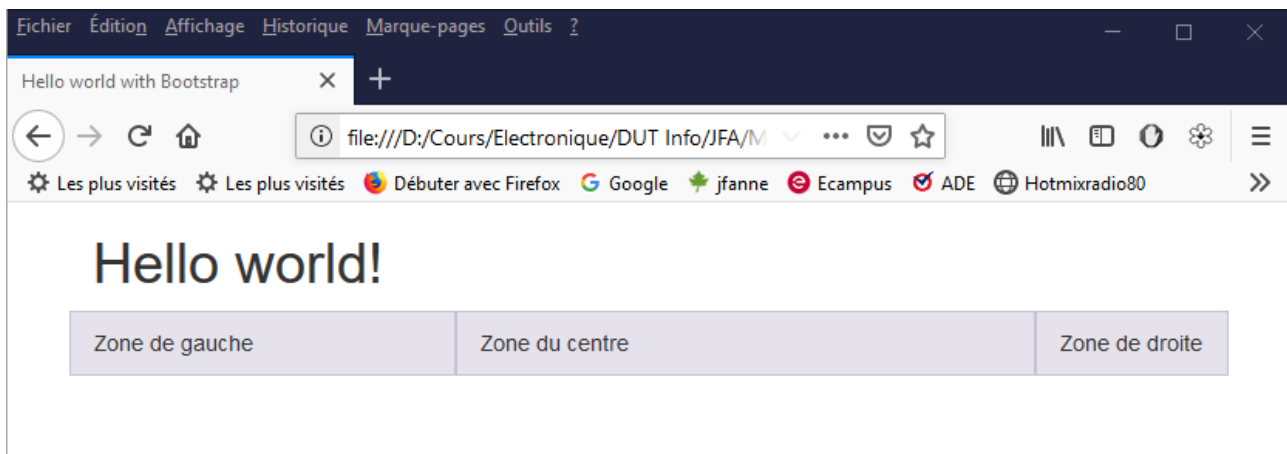
On vérifie bien que la somme des tailles des colonnes est égale à 12 dans chaque configuration d'affichage (4+6+2 en mode tablette, 3+7+2 en desktop et plus).

Il est même possible de masquer ou de faire apparaître certains éléments en fonction de la surface d'affichage grâce aux classes **visible-*** et **hidden-***. Par exemple, on souhaite qu'en contexte smartphone, la zone de droite soit masquée et que les zones de gauche et du centre prennent chacune 50% de la place disponible. Voici le code source associé.

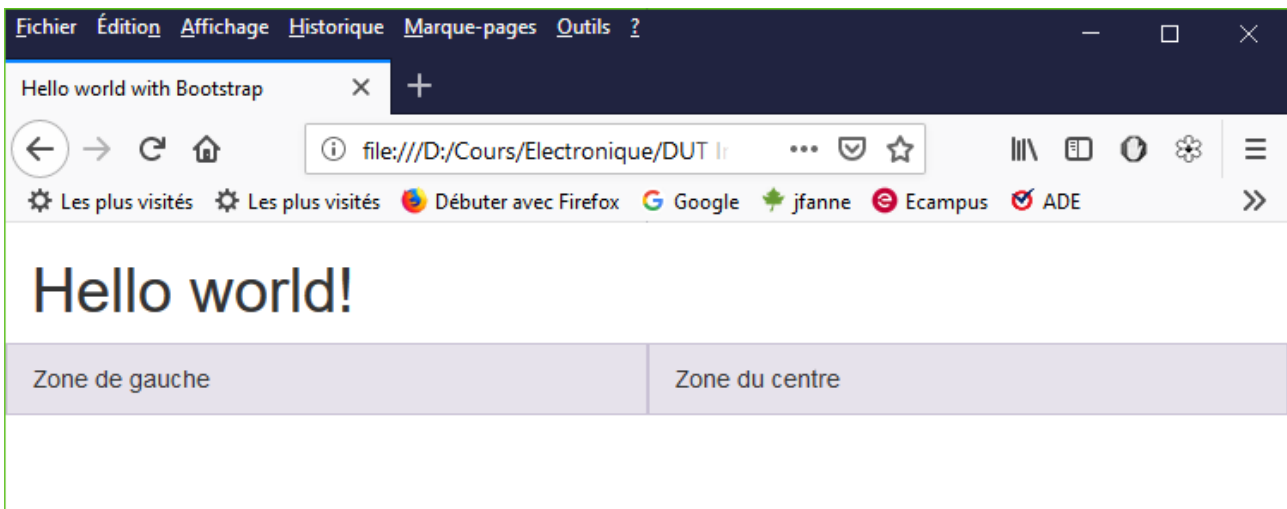
```
!-- ... -->
<div class="row">
  <!-- Taille 6 sur smartphones, 4 sur tablettes, 3 sur desktop et plus -->
  <div class="col-xs-6 col-sm-4 col-md-3">
    Zone de gauche
  </div>
  <!-- Taille 6 sur smartphones et tablettes, 7 sur desktop et plus -->
  <div class="col-xs-6 col-md-7">
    Zone du centre
  </div>
  <!-- Masquée sur smartphones, taille 2 sur tablettes et plus -->
  <div class="col-sm-2 hidden-xs">
    Zone de droite
  </div>
</div>
!-- ... -->
```

Vous pouvez tester ce code pour observer le comportement décrit plus haut.

Rendu desktop :



Rendu desktop :



XI. Conclusion

En remplaçant la gestion CSS manuelle du positionnement (`float : left;` , etc) par l'utilisation des classes Bootstrap `row` et `col-*`, on rend la mise en page Web à la fois plus simple et plus souple.

Les exemples ci-dessus ne donnent qu'un aperçu des possibilités de Bootstrap. Consultez la documentation (<http://getbootstrap.com/css>) du Framework pour plus de détails.

C. Aperçu de quelques composants Bootstrap

Outre les classes de mise en page déjà étudiées, Bootstrap fournit de nombreux composants prêts à être intégrés dans nos pages Web.

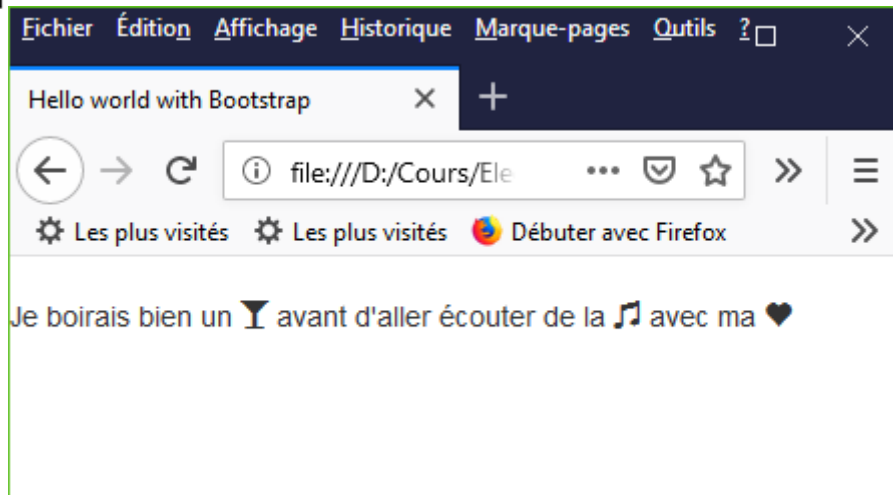
XII. Images et icônes

Bootstrap est livré avec environ 200 mini-images (*glyphicons*) librement utilisables.

```
Je boirais bien un <span class="glyphicon glyphicon-glass"></span> avant
d'aller écouter de la <span class="glyphicon glyphicon-music"></span> avec
ma <span class="glyphicon glyphicon-heart"></span>
```

Rendu :

Ce qui donne le résultat suivant :



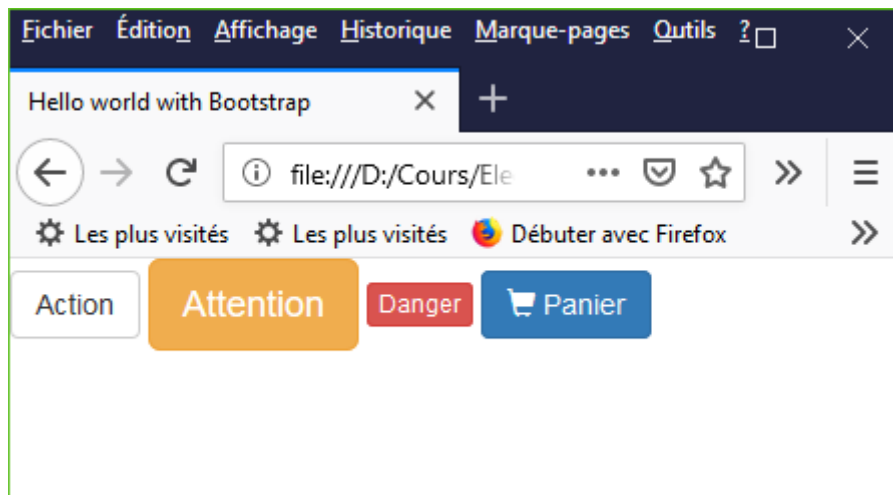
De plus, Bootstrap fournit une classe **img-responsive** qui permet aux images (balises HTML ``) de se redimensionner automatiquement à la taille de leur conteneur parent.

XIII. Boutons

Bootstrap fournit un moyen simple d'obtenir des boutons élégants et personnalisables.

```
<button type="button" class="btn btn-default">Action</button>
<button type="button" class="btn btn-warning btn-lg">Attention</button>
<button type="button" class="btn btn-danger btn-xs">Danger</button>
<button type="button" class="btn btn-primary"><span class="glyphicon
glyphicon-shopping-cart"></span> Panier</button>
```

Rendu :

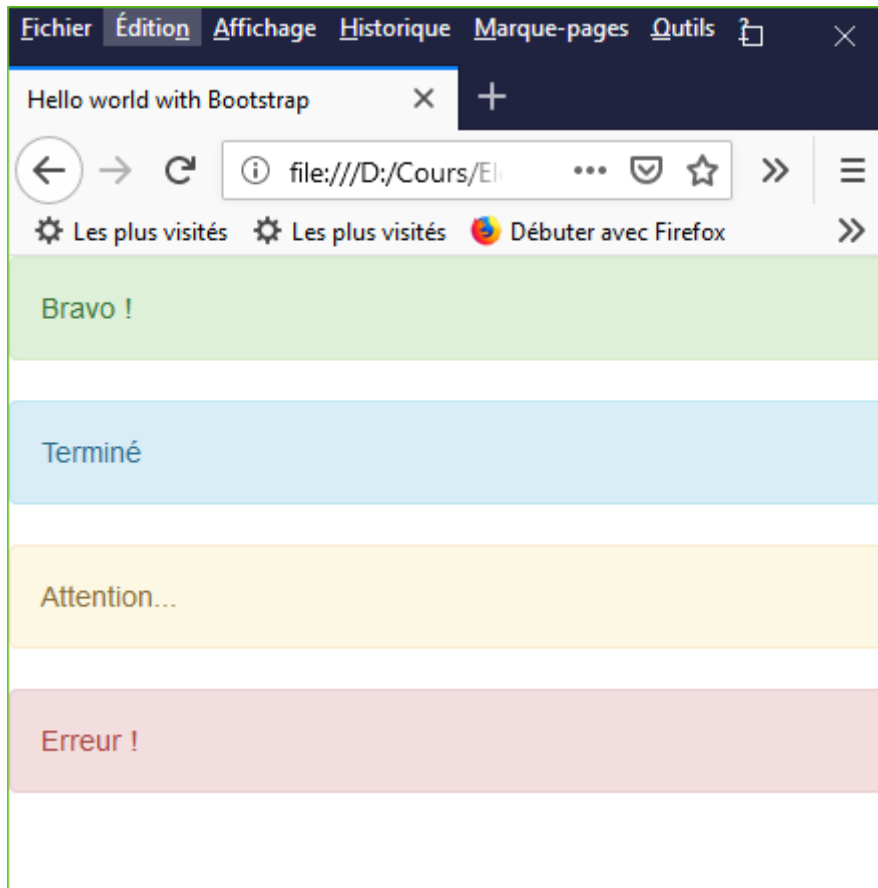


XIV. Messages

Bootstrap permet de définir des messages faciles à repérer dans une page Web.

```
<div class="alert alert-success">Bravo !</div>
<div class="alert alert-info">Terminé</div>
<div class="alert alert-warning">Attention...</div>
<div class="alert alert-danger">Erreur !</div>
```

Rendu :



XV. Menus de navigation

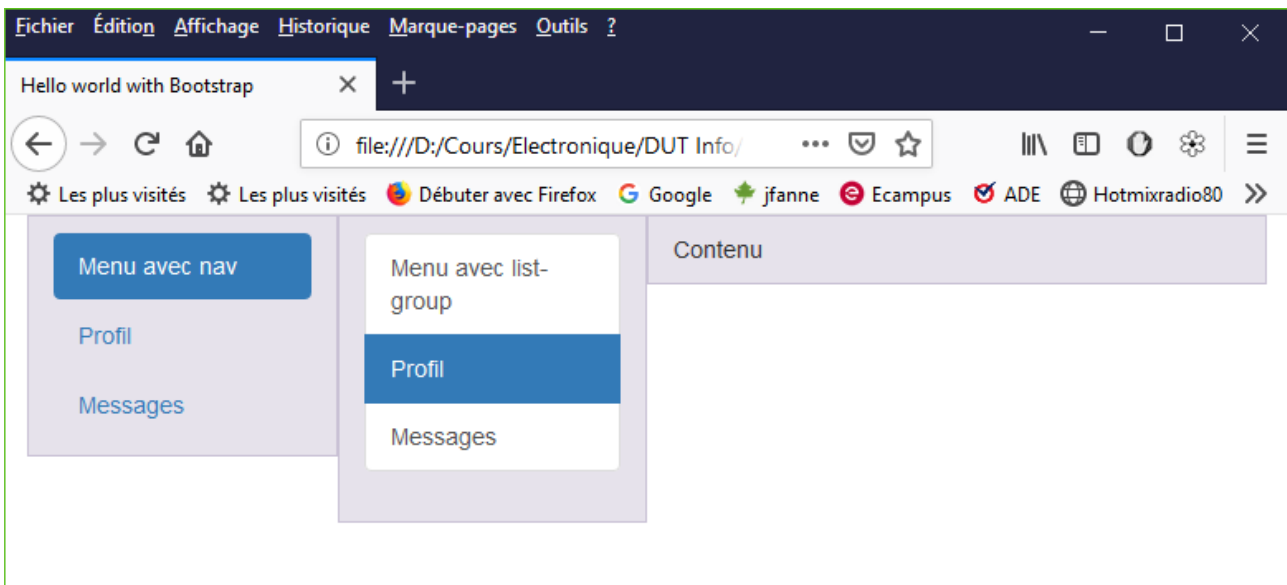
Il existe deux possibilités pour concevoir un menu de navigation :

- Utiliser la classe `nav` ;
- Utiliser la classe `list-group` ;

Le code ci-dessous définit une mise en page sur trois colonnes, avec deux menus et une zone de contenu.

```
<div class="row">
  <div class="col-xs-3">
    <ul class="nav nav-pills nav-stacked">
      <li class="active"><a href="">Menu avec nav</a></li>
      <li><a href="">Profil</a></li>
      <li><a href="">Messages</a></li>
    </ul>
  </div>
  <div class="col-xs-3">
    <div class="list-group">
      <a class="list-group-item" href="">Menu avec list-group</a>
      <a class="list-group-item active" href="">Profil</a>
      <a class="list-group-item" href="">Messages</a>
    </div>
  </div>
  <div class="col-xs-6">
    Contenu
  </div>
</div>
```

Rendu :

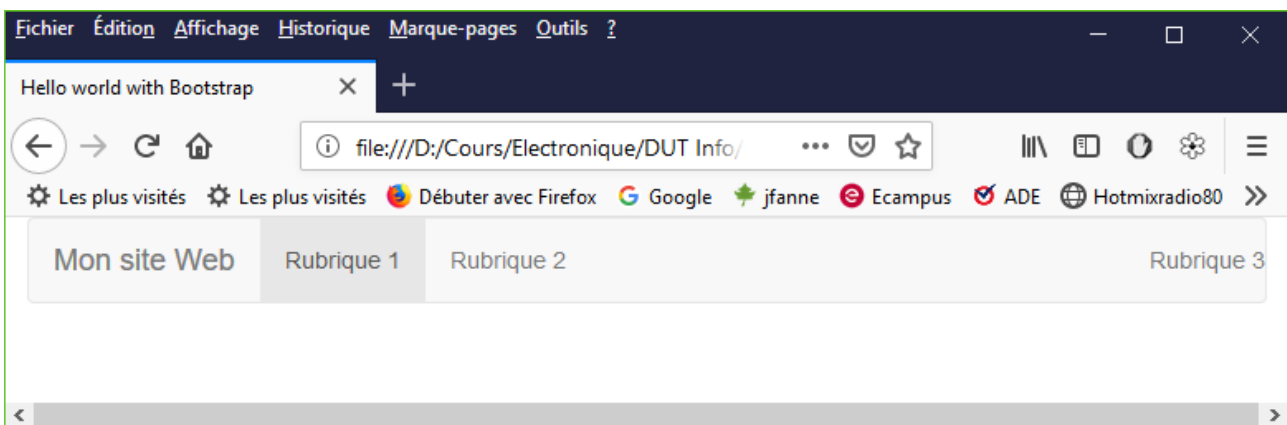


XVI. Barre de navigation

On peut ajouter facilement une barre de navigation horizontale grâce à la classe Bootstrap navbar .

```
<nav class="navbar navbar-default" role="navigation">
  <a class="navbar-brand" href="#">Mon site Web</a>
  <ul class="nav navbar-nav">
    <li class="active"><a href="#">Rubrique 1</a></li>
    <li><a href="#">Rubrique 2</a></li>
  </ul>
  <ul class="nav navbar-nav navbar-right">
    <li><a href="#">Rubrique 3</a></li>
  </ul>
</nav>
```

Rendu :



Il est également possible d'obtenir une barre de navigation adaptative. Son contenu est masqué et remplacé par un simple bouton d'accès lorsque la surface d'affichage est insuffisante.

```
<div class="navbar navbar-default" role="navigation">
  <!-- Partie de la barre toujours affichée -->
```

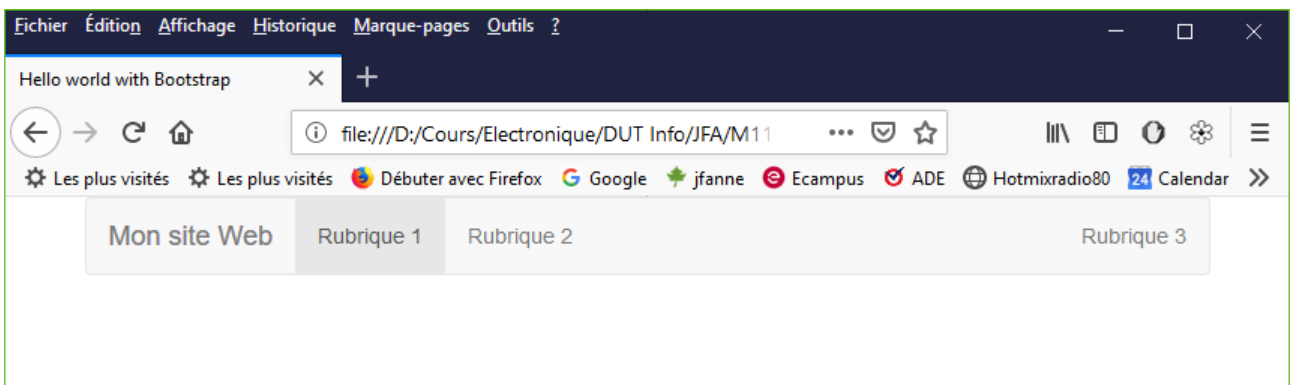


```

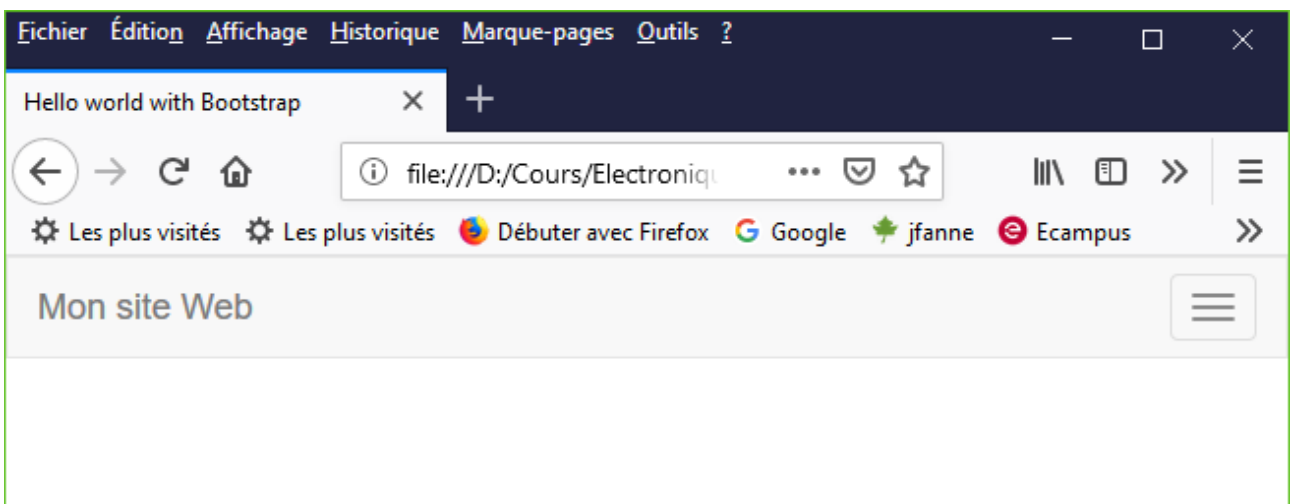
<div class="navbar-header">
  <!-- Bouton d'accès affiché à droite si la zone d'affichage est
trop petite -->
  <button type="button" class="navbar-toggle" data-toggle="collapse"
data-target="#navbar-collapse-target">
    <span class="sr-only">Activer la navigation</span>
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
  </button>
  <a class="navbar-brand" href="#">Mon site Web</a>
</div>
<!-- Partie de la barre masquée si la surface d'affichage est
insuffisante -->
<div class="collapse navbar-collapse" id="navbar-collapse-target">
  <ul class="nav navbar-nav">
    <li class="active"><a href="#">Rubrique 1</a></li>
    <li><a href="#">Rubrique 2</a></li>
  </ul>
  <ul class="nav navbar-nav navbar-right">
    <li><a href="#">Rubrique 3</a></li>
  </ul>
</div>
</div>

```

Rendu desktop :



Rendu tablette ou smartphone :



D. Conclusion

Nous n'avons fait que survoler les possibilités de Bootstrap. Il existe de nombreuses ressources sur ce Framework en ligne, à commencer par sa documentation officielle (<http://getbootstrap.com>) ou encore les tutoriels des sites OpenClassrooms et Codecademy.

E. Webographie

- <http://leaverou.github.io/prefixfree/#plugins>
- <https://www.alsacreations.com/article/lire/1306-prefix-free-prefixes-CSS3.html>
- <https://www.supinfo.com/articles/single/2694-comparatif-10-frameworks-utilises-front-end>
- <https://www.alticreation.com/bootstrap-foundation-Frameworks-front-end/>
- <http://prof.bpesquet.fr/tutoriel/premiers-pas-Framework-bootstrap/#>
- <http://getbootstrap.com/docs/3.3/components/#input-groups>
- <http://bootstrapdocs.com/>
- <https://github.com/daneden/animate.css>
- <https://www.alticreation.com/introduction-a-material-design-lite/>
- <https://tutowebdesign.com/mise-page-bootstrap.php>
- <https://www.codeur.com/blog/front-end-framework/>
- <https://www.codeur.com/blog/creation-template-bootstrap/>
- <https://www.layoutit.com/>
- <https://graphiste.com/blog/meilleurs-frameworks-css>
- <https://www.editions-eni.fr/open/mediabook.aspx?idR=2c3f03c3c6f065ca0d54ce599f7ce75c>
- <http://usablica.github.io/front-end-frameworks/compare.html>
- <https://www.pierre-giraud.com/bootstrap-apprendre-cours/>
- <https://bootstrap-doc.prauds.fr/index3.php#panels>
- <https://www.pierre-giraud.com/bootstrap-apprendre-cours/>
- <https://www.pierre-giraud.com/html-css-apprendre-coder-cours/media-queries-responsive-design/>
- https://developer.mozilla.org/fr/docs/Web/CSS/Media_Queries/Using_media_queries
- https://developer.mozilla.org/fr/docs/Learn/CSS/CSS_layout/Responsive_Design
- https://developer.mozilla.org/fr/docs/Learn/CSS/CSS_layout/Media_queries