



# LA MANU

L'ÉCOLE DES MÉTIERS DU NUMÉRIQUE  
Manufacture de compétences



## SUPPORT APPRENANT

*Responsive design et Bootstrap*



**CONFIDENTIEL**

*Ce document est strictement confidentiel et ne doit pas être diffusé  
sans accord préalable écrit*

## Responsive design et Bootstrap

### Ce que l'on a vu...

#### Responsive Web Design

De nos jours, on peut consulter des sites web sur différents types de supports (télévision, ordinateur, tablette, smartphone...). Pour assurer une navigation adaptée à ces différents supports, 3 solutions se présentent pour les développeurs :

1. Réaliser un **site dédié**, c'est-à-dire concevoir un site à part entière pour chacun des supports ciblés (soit un site pour les ordinateurs, un site pour les tablettes, un site pour les smartphones...), mais cette solution a l'inconvénient de dupliquer le travail de développement ainsi que des opérations de maintenance.
2. Réaliser une **application native**, c'est-à-dire concevoir une application dans un langage spécifique (Swift, Java...) et qui sera téléchargeable dans le store des smartphones (AppStore, Google Play...), mais cette solution nécessite également de dupliquer le travail de développement et de maintenance et cela sur des technologies différentes.
3. Réaliser un **site responsive**, c'est-à-dire concevoir un site qui s'adaptera à toutes les tailles d'écrans mais cette solution augmente le temps de développement.

On distingue 4 design différents :

1. Le **design statique** : design fixe qui ne s'adapte pas aux différents supports, utilise les pixels pour définir la taille des éléments.
2. Le **design liquide** : design fluide dont le contenu s'adapte aux écrans mais pas le design, n'est pas optimisé pour les petits écrans, utilise les pourcentages pour définir la taille des éléments.
3. Le **design adaptatif** : design qui s'adapte aux écrans mais pas le contenu, utilise une mise en page différente selon la résolution de l'écran du navigateur.
4. Le **design responsive** : design et contenu qui s'adaptent aux différentes tailles d'écran, utilise les pixels, les pourcentages, les em et rem pour définir la taille des éléments.

Faire du Responsive Web Design (RWD) peut se faire de différentes manières mais nécessite d'utiliser certaines méthodes et technologies. Il faut prévoir des sites qui s'adaptent à tous les écrans et pratiquer la philosophie du « Mobile First », c'est-à-dire concevoir en priorité la version mobile du site et en l'adaptant progressivement pour les écrans plus larges. Cela permet ainsi de réaliser une amélioration progressive en ajoutant des effets ou fonctionnalités en augmentant la taille des écrans plutôt que de réaliser une dégradation progressive en s'adaptant aux petits écrans.

Il faut utiliser la balise `<meta name="viewport" content="width=device-width, initial-scale=1" />` qui permet de contrôler la mise en page d'un site sur différentes tailles d'écrans.

On peut également utiliser des media queries qui permettent de définir des règles CSS spécifiques en fonction de la résolution des écrans et/ou du périphérique utilisé pour consulter le site. Pour cela on utilise une feuille de style indépendante contenant des propriétés CSS spécifiques et qui est liée au document HTML avec une balise `<link/>` dont l'attribut **media** prendra les valeurs et fonctionnalités souhaités (screen, print, tv, all, width, orientation...).

Enfin il est possible de s'aider de frameworks tels que Bootstrap, Materialize, Knacss, Foundation... pour réaliser un site en Responsive Web Design.

## Bootstrap



Bootstrap est un framework HTML, CSS, JavaScript créé par des développeurs de Twitter. Ce framework permet de concevoir des pages web esthétiques et responsives, notamment grâce à l'utilisation de class prédéfinies et grâce à un système de grille sur 12 colonnes. La grille Bootstrap permet d'organiser l'affichage d'éléments HTML en fonction de la taille de l'écran (col-lg, col-md, col-sm...). Les valeurs des propriétés CSS des différentes class Bootstrap sont définies dans un fichier bootstrap.min.css qu'il ne faut pas modifier.

Pour utiliser des éléments Bootstrap il faut inclure ce fichier ainsi que des fichiers script au document HTML dans lequel on souhaite les utiliser. Ils sont à placer dans la balise **<head>** et à la fin de la balise **<body>** avant vos propres fichiers CSS et vos propres scripts afin d'éviter que vos fichiers soient écrasés par ceux de Bootstrap. Ces fichiers peuvent être ajoutés de 2 façons au projet : soit en utilisant le CDN (Content Delivery Network) disponible sur le site de Bootstrap, soit en téléchargeant le NuGet Bootstrap accessible depuis le gestionnaire de package NuGet de Visual Studio, les fichiers utiles pour son utilisation seront installés au projet et il ne restera plus qu'à les inclure au fichier HTML.

## Liens utiles

<https://www.alsacreations.com/article/lire/1615-cest-quoi-le-responsive-web-design.html>  
<https://www.uptilab.com/fr/responsive-ou-adaptive-design/>  
<https://www.usabilis.com/responsive-web-design-site-web-adaptatif/>  
<http://glossaire.infowebmaster.fr/mobile-first/>  
<https://www.alsacreations.com/article/lire/1490-comprendre-le-viewport-dans-le-web-mobile.html>  
<https://www.alsacreations.com/article/lire/930-css3-media-queries.html>  
[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_mediaqueries\\_ex1](https://www.w3schools.com/css/tryit.asp?filename=trycss_mediaqueries_ex1)  
<https://getbootstrap.com/>  
<https://openclassrooms.com/en/courses/1885491-prenez-en-main-bootstrap>  
<https://bootswatch.com/>  
<https://mdbootstrap.com/>  
<http://pikock.github.io/bootstrap-magic/>  
<https://materializecss.com/>  
[https://search.google.com/test/mobile-friendly?utm\\_source=mft&utm\\_medium=redirect&utm\\_campaign=mft-redirect](https://search.google.com/test/mobile-friendly?utm_source=mft&utm_medium=redirect&utm_campaign=mft-redirect)

## Bonnes pratiques

Rappel des principales bonnes pratiques	
<input type="checkbox"/>	Prioriser l'ordre des link dans la balise head (ex : le fichier Style.css doit être en dernier).
<input type="checkbox"/>	Attention à l'utilisation des CDN qui ne fonctionnent pas hors connexion.
<input type="checkbox"/>	1 seul framework par projet.
<input type="checkbox"/>	Ne pas modifier le css du framework.
<input type="checkbox"/>	Utilisation des media queries.
<input type="checkbox"/>	Utilisation d'un framework CSS pour simplifier les développements.
<input type="checkbox"/>	<meta name= "viewport" content="width=device-width, initial-scale=1" />
<input type="checkbox"/>	Vocabulaire spécifique : <ul style="list-style-type: none"><li>• Adaptative VS liquide VS statique VS responsive</li><li>• Site mobile</li><li>• Application native</li><li>• RWD</li></ul>