

Classes : SEIOT2

Matière : Programmation WEB

Enseignants TP : Med. A. DAHDEH, Yossra HATTAB

Nb pages : 5

TP n° 4 CSS : Responsive design

Le « responsive design » ou design adaptable désigne l'ensemble des techniques permettant d'adapter les pages web à toutes les tailles d'écrans (Ordinateurs, Tablettes, Smartphones) afin d'avoir un bon rendu.

Il existe trois façons d'adapter un site :

- Créer un site dédié pour chaque terminal différent (un site mobile, un site pour ordinateur, etc.) ;
- Créer une application mobile en plus de notre site web (gérant l'environnement Android et iOS) ;
- Utiliser les outils offerts par le HTML et le CSS et créer une version « responsive » de notre site.

I. Utilisation du viewport

Le « viewport » correspond à la taille de la fenêtre web des visiteurs. Une première méthode, très simple, pour commencer à optimiser votre site web pour un affichage sur mobile ou tablette est d'utiliser et de manipuler le viewport.

En HTML5, nous pouvons prendre « contrôle » de la taille de la fenêtre grâce à l'élément **meta** et aux attributs **name** et **content**. Cela permet de remplir la fenêtre ayant n'importe quelle taille avec le contenu de la page.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Le code `width=device-width` va faire en sorte que la page s'adapte à la taille de l'écran avec les éléments HTML prenant la place disponible.

Le code `initial-scale=1.0` définit le niveau de zoom initial lorsque la page est chargée par le navigateur.



Quelques règles supplémentaires à suivre:

- Ne pas utiliser une largeur fixe pour les grands éléments.
- Ne pas fixer la largeur du viewport à une valeur particulière.
- Utilisez les Media Queries de CSS pour appliquer un style différent pour les petits et grands écrans.

II. Utilisation des valeurs en pourcentage en CSS

En plus de l'emploi de l'élément meta, nous pouvons utiliser des valeurs en pourcentages en CSS pour définir les tailles de certains éléments plutôt que des valeurs en pixels.

Lorsque vous utilisez cette méthode, pensez bien que les tailles exprimées sont un pourcentage de la taille de l'élément parent des éléments que vous ciblez actuellement.

Ainsi, si vous définissez une taille égale à 100% pour votre élément body, puis ensuite une largeur de 50% pour un div enfant direct du body, la taille du div sera toujours égale à 50% de celle du body.

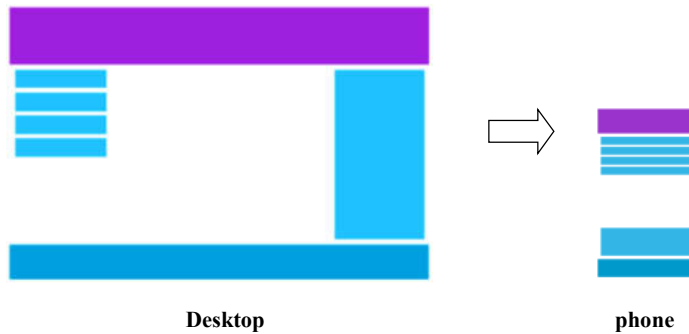
Si maintenant vous définissez à nouveau une largeur égale à 50% pour les paragraphes à l'intérieur de votre div, les paragraphes occuperont 50% de la largeur du div.

III. Utilisation des Media Queries

Grâce aux media queries, nous allons pouvoir appliquer différentes propriétés ou différentes valeurs à des éléments HTML selon la taille de l'écran et le type de média utilisé (écran, imprimante, etc.).



Cela va nous permettre par exemple de ne pas afficher certains éléments pour des tailles d'écran trop petites ou de réorganiser nos pages web grâce aux propriétés CSS.



Les Media Queries sont des expressions dont la valeur est vraie ou fausse. Lorsqu'un Media Query a pour valeur true, les instructions situées entre les accolades qui le suivent sont exécutées. Elles sont ignorées dans le cas contraire. Si nécessaire, vous pouvez associer plusieurs expressions à l'aide d'opérateurs logiques :

Opérateur	Signification
and	et
only	uniquement : masque la suite sur les navigateurs non compatibles avec les Media Queries
not	non
or	ou

Les Media Queries reposent sur la propriété CSS3 @media. Le tableau suivant rassemble quelques-unes des syntaxes utilisables :

Syntaxe	Signification
@media (max-width: largeur) { ... }	Largeur de la fenêtre inférieure à la largeur spécifiée
@media (max-device-width: largeur) { ... }	Largeur du périphérique inférieure à la largeur spécifiée
@media (min-width: largeur1) and (max-width: largeur2) { ... }	Largeur de la fenêtre comprise entre les deux largeurs spécifiées
@media (max-device-width: largeur) and (orientation: landscape) { ... }	Largeur du périphérique inférieure à la largeur spécifiée, et écran tenu horizontalement

On peut associer un style selon le type de media pour lequel les déclarations CSS doivent s'appliquer. Dans l'immense majorité des cas, nous utiliserons **@media screen**, afin d'appliquer nos propriétés à tous les medias dotés d'un écran (téléphone portable, tablette, ordinateur de bureau, etc.).



Cependant, sachez que l'on peut également utiliser **@media print**, par exemple, afin de n'appliquer certaines règles que pour les imprimantes ou encore **@media speech**, pour les ordinateurs spéciaux qui vont « lire à haute voix » une page. On peut également utiliser **@media all** pour appliquer des déclarations à tous les types de media.

Exemple : changer la couleur du corps de la page en rouge si la largeur de la fenêtre est inférieure à 600px

```
@media screen and (max-width:600px) {
    body{
        background-color: red;
    }
}
```

Les media queries vont être particulièrement intéressants pour masquer certains éléments ou pour modifier leur agencement selon la taille de l'écran de vos visiteurs.

Pour faire cela, nous allons pouvoir utiliser les propriétés **display**, **position** et les valeurs relatives en pourcentages de façon générale.

On peut également définir des règles spécifiques selon l'orientation de l'écran de vos visiteurs (utile pour l'affichage sur tablette ou iPhone entre autres).

Pour cela, on peut utiliser « orientation : landscape » pour définir des règles spécifiques pour les écrans tournés en mode « paysage » ou « orientation : portrait » en mode « portrait ».

```
@media screen and (orientation:landscape) { ... }
```

```
@media screen and (orientation:portrait) { ... }
```

Travail demandé:

1. Rajouter la balise <meta> permettant d'adapter le contenu de la page à la largeur de la fenêtre.
2. Copier le code CSS de la feuille de style **monstyles.css** dans **monstyle2.css**, puis modifier votre nouvelle feuille CSS pour changer la largeur des différents blocs de votre page index proportionnellement au corps de la page ou à leurs éléments parents.

Proposition :

- `body = width : 95%`
- `header = margin : 5%`
- `#content = margin : 5%`
- `nav :width : 20% ; margin-right : 2%`
- `article : width : 50% ; margin-right : 2%`
- `aside : width 20%`
- `footer : 85% ; margin : 5%`

3. Réarranger votre page comme suit si la largeur de la fenêtre devient inférieure à 768px





Prenom Nom
prenom.nom@gmail.com

Bienvenue sur ma page personnelle

[Accueil](#)
[Mon CV](#)
[Mes fichiers](#)
[Mes projets](#)

A propos de moi

Je suis un étudiant en SI3 à la faculté des sciences de Bizerte et ceci ma première exeperience en HTML et CSS...

Mes centres d'intérêt

Je suis passionné de

1. Sport
2. Lecture
3. Jeux vidéo
4.

biablablablablablablabla....

Contactez-moi

- [Email](#) 
- [Facebook](#) 
- [Twitter](#) 

Liens utiles

- [Apprenez à créer votre site web avec HTML5 et CSS3](#)
- [Liste des balise HTML5](#)

Notion complémentaire : Responsive Grid-View....Allez chercher !

Référence :

http://www.w3schools.com/css/css_rwd_intro.asp

<http://pierre-giraud.com/html-css/cours-complet/introduction-responsive-design.php>