CSS Phase 7 : Le web responsive

Objectifs

- Connaître les contraintes du web responsive
- Maîtriser les media queries CSS
- Adapter une page HTML pour le responsive

Cheminement

Assistez à la présentation sur le web responsive par votre formateur.

Pensez *responsive*

Tous les éléments composant une page web doivent être redimensionnés automatiquement. Il ne faut donc pas oublier :

- Les polices de caractères
- Les images
- Les autres médias : vidéos, animations (canvas, Flash...)
- Les tableaux HTML

Le viewport

Pour pouvoir rendre une page responsive, il faut au préalable ajouter une balise HTML meta (donc dans la partie `<head>`) nommée **viewport**.

Le code de celle-ci est en général le suivant :

```
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
```

Expliquons les valeurs de l'attribut `content` :

- `width=device-width` : fixe la largeur de la fenêtre = celle de l'appareil
- `initial-scale=1.0` : niveau de zoom initial (ici échelle = 1, soit 100%)
- `shrink-to-fit=no` : règle un bug spécifique du navigateur Apple Safari mobile sous le système d'exploitation iOS 9 (iPhone/iPad).

> La notion de viewport est un peu plus complexe techniquement, lire (facultatif) cette page qui explique de façon détaillée son fonctionnement.

Media queries

Les media queries permettent d'appliquer des règles CSS différentes selon le dispositif et d'autres critères (appelés fonctionnalités).

Listes des dispositifs :

- `screen` : écrans
- `handheld` : périphériques mobiles ou de petite taille
- `print` : impression
- `aural` (CSS 2.0) / `speech` (CSS 2.1) : synthèses vocales
- `braille` : plages braille
- `embossed` : imprimantes braille
- `projection` : projecteurs (ou présentations avec slides)
- `tty` : terminal/police à pas fixe
- `tv` : téléviseur
- `all` : tous les précédents

Liste des fonctionnalités :

- `color` : support de la couleur (bits/pixel)
- `color-index` : périphérique utilisant une table de couleurs indexées
- `device-aspect-ratio` : ratio du périphérique de sortie (par exemple 16/9)
- `aspect-ratio` : ratio de la zone d'affichage
- `device-height` : dimension en hauteur du périphérique
- `device-width` : dimension en largeur du périphérique
- `grid` : périphérique bitmap ou grille (ex : lcd)
- `height` : dimension en hauteur de la zone d'affichage
- `monochrome` : périphérique monochrome ou niveaux de gris (bits/pixel)
- `orientation` : orientation du périphérique (portait ou landscape)
- `resolution` : résolution du périphérique (en dpi, dppx, ou dpcm)
- `scan` : type de balayage des téléviseurs (progressive ou interlace)
- `width` : dimension en largeur de la zone d'affichage

Les fonctionnalités peuvent être préfixées par `min-` ou `max-`.

Ces critères peuvent être combinés via les opérateurs suivants :

- `and` : et
- `only` : uniquement
- `not` : non

> Pour le web, nous serons surtout concernés par les dispositifs de type écrans et les largeurs minimum et maximum.

Exemples :

```
@media screen and (max-width: 576px)
{
    h1
    {
        font-size : 20px;
    }
}

@media screen and (max-width: 768px)
{
    h1
    {
        font-size : 40px;
    }
}
```

Ce code CSS affichera le texte des balises `h1` sur 20 pixels pour les écrans inférieurs à 576 pixels puis sur 40 pixels pour les écrans de 577 à 768 pixels. Au-delà les `h1` auront une taille normale.

> Attention au sens de lecture des préfixes `min-` (s'appliquera à ce qui est au-dessus de la valeur indiquée) et `max-` (s'appliquera à ce qui est en dessous de la valeur indiquée).

Intégration des media queries

L'appel de media queries dans votre page peut se faire de 2 manières :

- Un seul fichier CSS externe global contenant toutes les media queries.
- Plusieurs fichiers CSS externes, un par media querie/dispositif/point d'arrêt, ciblé par l'attribut `media` de la balise `<link>` :

```
<link rel="stylesheet" media="screen and (max-width: 575px)" href="mobile.css">
<link rel="stylesheet" media="screen and (min-width: 576px)" href="tablette.css">
<link rel="stylesheet" media="screen and (min-width: 992px)" href="desktop.css">
```

Taille des polices de caractères

La gestion de la taille des polices de caractères en responsive peut s'avérer complexe (elle l'est d'ailleurs aussi en "non responsive"). La problématique est cependant commune avec les autres éléments : il s'agit de rendre flexible (élastique) pour le redimensionnement, c'est-à-dire utiliser des unités de mesure relatives et non absolues (pt, cm etc.).

Le cas particulier des pixels

Traditionnellement, l'unité de taille de police la plus utilisée sur le web est le pixel. Celle-ci est bien une unité dite « relative » c'est-à-dire redimensionnable; mais elle dépend en effet de la résolution et de la taille de l'écran.

Les autres unités relatives : `%`, `em` et `rem`

Le problème des unités relatives `%` (pourcentage) et `em` (cadratin) est qu'elles sont proportionnelles à leurs parent. Cela se révèle ingérable en responsive, où le principe est de tout redimensionner. Les résultats obtenuis ne sont pas focément ceux attendus par le concepteur d'une page web.

Pour y remédier, l'unité `rem` se base sur la taille de l'élément racine ("root em") de la page, c'est-à-dire la balise `<html>`. Tous les calculs de taille pour les autres éléments HTML sont donc effectués ensuite sur cette base.

Nouvelles unités `vw` et `vh`

Le CSS 3 a introduit de nouvelles unités spécialement dédiée au responsive : `vw`, `vh`, `vmin` et `vmax` basées sur la notion de viewport. La solution la plus simple pour des polices responsive est donc d'utiliser ces dernières unités. Revoir le cours CSS Polices et Couleurs pour plus de détails.

Quelle unité choisir pour le responsive ?

Les unités basées sur le viewport sont un bon choix puisque spécialement dédié au responsive. La seule limite est leur prise en charge sous d'anciennes versions de navigateurs. Autre point, le redimensionnement peut s'avérer trop petit sur mobile, il faut donc veiller à laisser des tailles lisibles.

Beaucoup préconisent l'utilisation de l'unité `rem` en fixant à 100% la racine `<html>` :

```
html {
    font-size: 100%;
}
```

Puis pour les autres éléments, on utilisera l'unité `rem`, par exemple :

```
h1 {
    font-size: 2rem;
}
```

Il convient toutefois là aussi de s'assurer de la compatibilité navigateur.

Conversion

Il existe des outils de conversion d'une unité de taille à une autre, par exemple (pixels vers rem).

Sources

- https://blog.lesieur.name/pourquoi-j-utilise-l-unite-rem-et-non-l-unite-pixel
- http://wdfriday.com/wdfriday.com/blog/2012/02/font-size-responsive-accessibilite/index.html

Images

On rend une image responsive en appliquant les instructions CSS suivantes. Au préalable, il est nécessaire de supprimer les attributs de dimensions `width` et `height` :

```
img
{
    max-width: 100%;
    height: auto;
}
```

- `max-width: 100%` : cette propriété est utilisée pour définir la largeur maximale d'un élément donné. Elle empêche la valeur de la propriété `width` de devenir supérieure à la valeur spécifiée par `max-width` (autrement dit, `max-width` est une borne supérieur pour `width`). Ainsi, l'image s'adaptera à la largeur de son conteneur sans jamais la dépasser même si la largeur de l'image en pixels est supérieure à la largeur de la balise qui contient l'image.
- `height: auto` : calcule automatiquement la hauteur, en responsive, permet d'ajuster la hauteur en rapport à la largeur de l'image, sinon l'image présenterait une distorsion.

> Ceci est la solution standard mais basique. Il existe des alternatives plus poussées.

Exercice

Reprenez la maquette du site *Jarditou* pour la rendre responsive en intégrant les contraintes suivantes :

- Un seul breakpoint : 768 pixels.
- La colonne de droite sera masquée sur les dispositifs mobiles.
- Sur mobile, les menus doivent s'afficher verticalement.
- Les polices doivent être redimensionnées

Une fois l'exercice terminé, demandez au formateur de vérifier votre travail.