Cours CSS

3WAcademy

Cours d'intégration Web

6. Le Responsive Web Design

6.1. Présentation

Le Responsive Web Design est un ensemble de pratiques permettant de visualiser de façon optimum une même page web sur différents appareils ayant des tailles d'écran très variées. En particulier, cela permet d'adapter une page web pour l'afficher sur un téléphone mobile ou sur une tablette.

D'autres stratégies sont possibles pour diffuser du contenu sur les appareils mobile, comme la création d'applications dédiées au système d'exploitation de l'appareil (IOS ou Android) ou encore la création d'un site spécifique (mobile.mon-domaine.com). Toutes ces stratégies ont un coût plus élevé à la création et à la maintenance pour l'entreprise éditrice ce qui explique l'engouement pour le Responsive Web Design.

6.2. Les différentes tailles des écrans

6.2.1. La surface physique

Un écran a une surface physique qui correspond au nombre de pixels réels qui composent l'écran. Cette surface est propre à chaque appareil, elle est définie par le constructeur. On parle de définition de l'écran (ou de résolution).

6.2.2. La surface de rendu de l'appareil

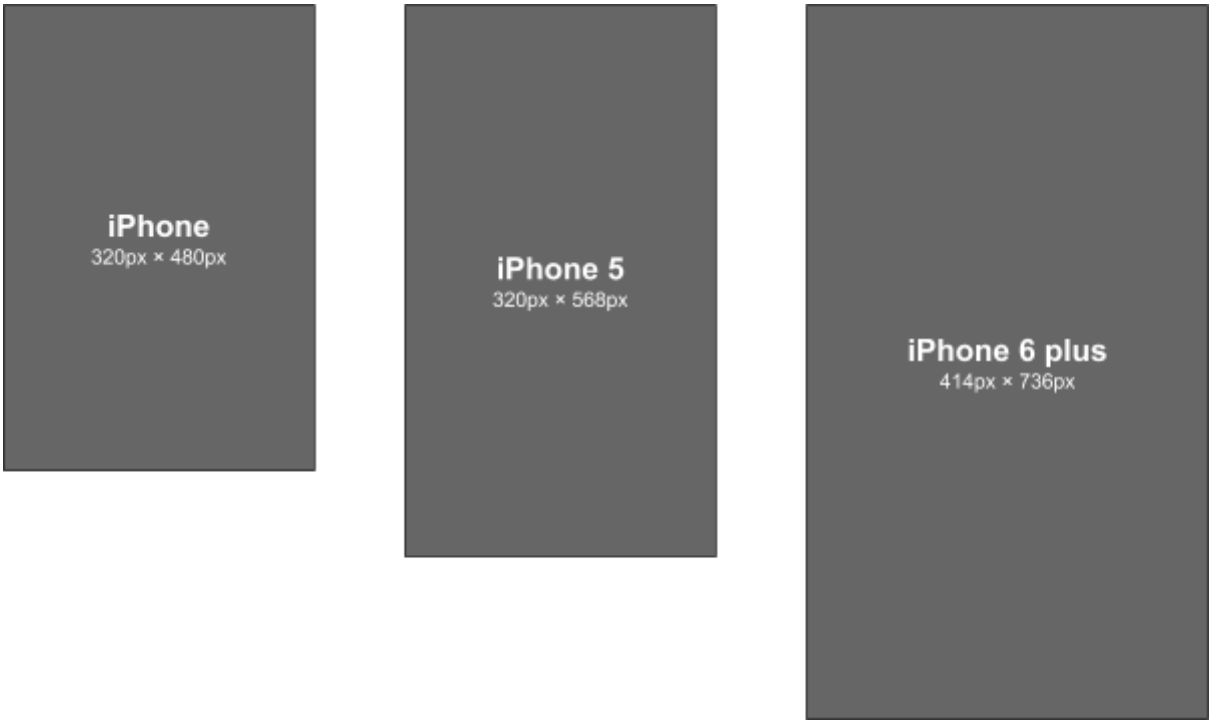
Les appareils mobiles comme les téléphones mobile ou les tablettes possèdent en outre une surface virtuelle nommée surface du rendu de l'appareil (*device-width*). Cette taille est définie par le constructeur de l'appareil, elle est généralement inférieure à la taille physique de l'écran. On peut admettre pour simplifier que tous les téléphones ont une largeur de rendu inférieure à 480px. (pour un téléphone tenu verticalement).

Les téléphones mobile ont maintenant des tailles physiques très grandes parfois supérieures au standard HD. Ces écrans ont une grande densité d'affichage, ce qui améliore la lisibilité à l'écran, néanmoins la taille du rendu de l'appareil reste petite pour que son contenu puisse être accessible et lisible (ceci est lié à la largeur de l'appareil qui doit tenir dans la main).

La taille physique de l'appareil n'est pas accessible depuis de navigateur et elle est d'ailleurs inutile. La taille qui nous intéresse est la taille du rendu.

Par exemple:

Appareil	Densité	Taille physique de l'écran	Taille du rendu de l'appareil
iPhone	1	320px × 480px	320px × 480px
iPhone 5	2	640px × 1136px	320px × 568px
iPhone 6+	2.6	1080px × 1920px	414px × 736px



Surfaces de rendu relatives de différents téléphones

Pour connaître les dimensions pour les autres appareils, il existe des sites qui référencent les valeurs comme https://www.mydevice.io, http://screensiz.es ou http://viewportsizes.com

6.2.3. Le viewport du navigateur

Le Viewport du navigateur correspond à la surface de la fenêtre du navigateur. Cependant, sur mobile, la notion de fenêtre est différente de celle sur un écran de bureau, il n'y a pas de cadre ni de barres de défilement car tout est prévu pour naviguer de manière tactile.

C'est le navigateur qui défini par défaut la taille de son viewport, par exemple 980px pour le navigateur Safari.

Cette taille permet d'afficher les pages non gérées en Responsive en entier. La navigation se fait alors en zoomant sur la partie de la page que l'on souhaite consulter.

6.3. La balise meta viewport

Toute l'ingéniosité du Responsive Web Design est de faire correspondre la taille du Viewport avec celle de la surface de rendu de l'appareil (le device-width) afin d'avoir un espace moins large correspondant plus à la taille physique de l'appareil (soit quelques centimètres de large). Pour faire cela, on déclare la correspondance dans une balise meta dédiée au viewport.

```
<meta name="viewport" content="width=device-width"/>
```

Ici la valeur du content signifie: la largeur du *viewport* doit être égale à la largeur du *device-width*.

Dans le cas d'un site spécifique, il est possible de mettre une taille fixe:

```
<meta name="viewport" content="width=320"/>
```

On peut aussi préciser d'autres informations comme le niveau de zoom initial:

```
<meta name="viewport" content="width=device-width,initial-scale=1"/>
```

Voici l'ensemble des propriétés disponibles:

width	Largeur de fenêtre viewport.	`width="device-width"`
height	Hauteur de fenêtre viewport.	`height="device-height"`
initial-scale	Niveau de zoom initial.	`initial-scale="1.0"`
minimum-scale	Niveau de zoom minimal.	`minimum-scale="0.5"`
maximum-scale	Niveau de zoom maximal. Attention, la valeur "1.0" interdit le zoom et peut rendre vos pages inaccessibles.	`maximum-scale="3.0"`
user-scalable	Possibilité à l'utilisateur de zoomer. Attention, la valeur "no" interdit le zoom et peut rendre vos pages inaccessibles.	`user-scalable="yes"`

6.4. Les Media Queries

Maintenant que l'espace disponible dans le viewport du navigateur est celui du device-width,l est nécessaire de faire en sorte que la page web s'affiche correctement quelle que soit cette taille.

On va utiliser pour cela les *Media Queries*. Il s'agit d'écrire des déclarations CSS en fonction de certains critères. La syntaxe est proche d'un test conditionnel de programmation.

6.4.1. Un exemple simple

Pour cibler uniquement les appareil dont la largeur du viewport est inférieur à 480px:

```
@media screen and (max-width: 480px) {
    /* Mettre ici les différentes règles concernées */
}
```

Le @media fait référence aux divers médias disponibles pour les feuilles de styles. ici la requête concerne les écran (@media screen).

6.4.2. Les critères disponibles

width	Largeur de la zone d'affichage (min et max)
height	Hauteur de la zone d'affichage (min et max)
orientation	Orientation du périphérique (portait ou landscape)
resolution	Résolution du périphérique (en dpi, dppx, ou dpcm)
aspect-ratio	Ratio de la zone d'affichage
color	Support de la couleur (bits/pixel)
grid	Périphérique bitmap ou grille (ex : lcd)
monochrome	Périphérique monochrome ou niveaux de gris (bits/pixel)

6.4.3. Les opérateurs disponibles

and	Pour combiner plusieurs critères (*ET* logique)
only	Pour spécifier uniquement une seule option
not	Pour la négation (NON logique)
,	Permet de faire un *OU* logique (séparer les critères par une virgule)

Par exemple, pour cibler uniquement les appareil dont la largeur du viewport est supérieur à 480px ET inférieur à 960px (des tablettes)

```
@media screen and (min-width: 480px) and (max-width: 960px) {
    /* Mettre ici les différentes règles concernées */
}
```

6.4.4. L'équivalence avec une balise link

Si l'on souhaite séparer le code concerné par la media-query au sein d'un fichier css distinct, il est possible d'utiliser la balise link avec l'attribut media.

Dans ce cas remplacer la déclaration suivante présent dans la feuille de style principale

```
@media screen and (max-width: 480px) {
    /* Mettre ici les différentes règles concernées */
}
```

Par cette déclaration dans le HEAD du document HTML

```
<link rel="stylesheet" media="screen and (max-width: 480px)"
href="mobile.css" />
```

Il faut ensuite placer les règles directement dans le fichier mobile.css

6.5. Les images

6.5.1. Les images fluides

Pour adapter la taille des images à des largeurs de viewport différentes, on peut simplement ajuster leur taille avec du pourcentage.

Par exemple:

Dans le CSS

```
img.image-responsive {
    width: 100%;
    height: auto;
}
```

La propriété *height* avec la valeur "auto" permet de faire respecter l'aspect ratio de l'image par tous les navigateurs.

6.5.2. L'attribut srcset

L'attribut srcset permet de proposer au navigateur différentes sources de la même image avec des tailles différentes. Cela lui permet de charger l'image la mieux adaptée à l'écran.

L'utilisation de l'attribut srcset est assez simple, il faut lister les différentes sources de l'image avec à la suite de la source un descripteur permettant au navigateur de choisir la source.

Le descripteur "x" permet d'indiquer la densité ou pixel ratio. Par exemple, en indiquant, 2x cela cible les écrans dits "rétina".

Dans le HTML

```
<img src="photo.jpg" srcset="photo-2x.jpg 2x" alt="Exemple"/>
```

Si le navigateur prend en charge l'attribut srcset et que l'écran est un écran avec une densité double, alors il chargera l'image photo-2x.jpg, sinon, il chargera l'image photo.jpg.

Le descripteur "w" permet d'indiquer la largeur de l'image proposée.

Dans le HTML

```
<img src="photo.jpg" srcset="photo-240.jpg 240w, photo-480.jpg 480w"
alt="Exemple"/>
```

6.5.3. L'élément picture

L'élément picture est une alternative plus complète qui possède la même structure en conteneur que les balises *audio* et *video*.

Voici un exmple:

Dans le HTML

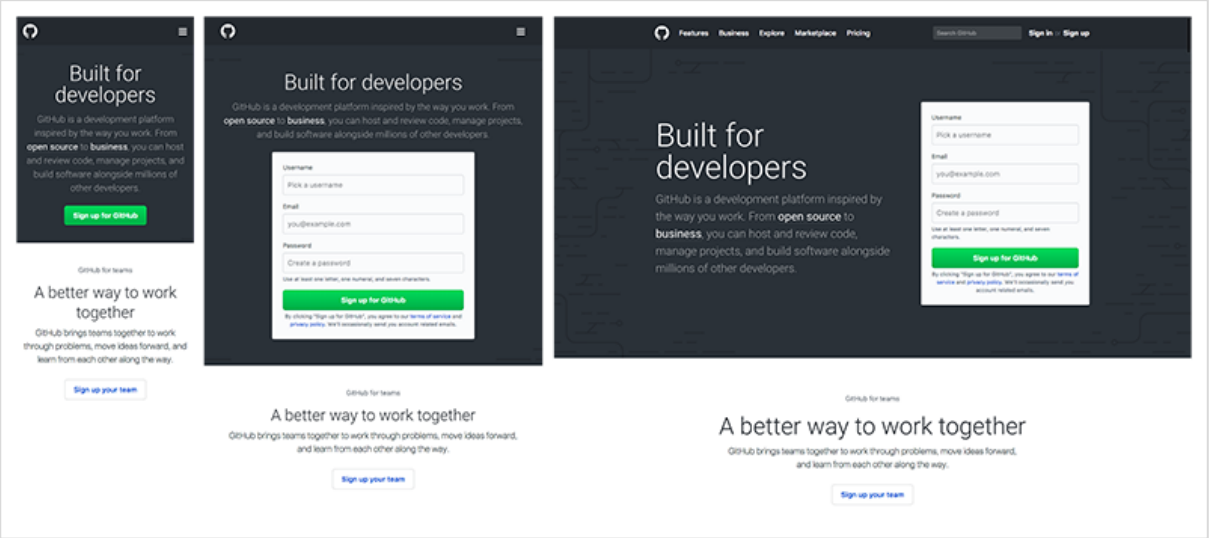
```
<picture>
    <source media="(max-width: 799px)" srcset="photo-480.jpg">
    <source media="(min-width: 800px)" srcset="photo-800.jpg">
    <img src="photo-800.jpg" alt="Exemple">
</picture>
```

On place un élément img comme alternative si le navigateur ne prend pas en charge l'élément picture.

Chaque élément source contient un attribut media qui peut contenir les mêmes critères que les media-queries. En outre chaque source contient également un srcset qui contient la source de l'image correspondante. Le srcset peut lui-même proposer plusieurs sources avec des résolutions différentes mais cette possibilité rarement utilisée.

6.6. Mise en oeuvre

Pour faire correctement l'intégration du page en Responsive Web Design, il est préférable de travailler à partir d'un design qui a été préalablement décliné pour les différents supports.



Il peut être judicieux de préparer le travail à l'aide d'un diagramme d'état reprenant les variations du design en fonction du support.