



Cours HTML

3WAcademy

Cours d'intégration Web

6. Le Responsive Web Design

6.1. Présentation

6.2. Les différentes tailles des écrans

6.2.1. La surface physique

6.2.2. La surface de rendu de l'appareil

6.2.3. Le viewport du navigateur

6.3. La balise meta viewport

6.4. Les Media Queries

6.4.1. Un exemple simple

6.4.2. Les critères disponibles

6.4.3. Les opérateurs disponibles

6.4.4. L'équivalence avec une balise link

6.5. Mise en oeuvre

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

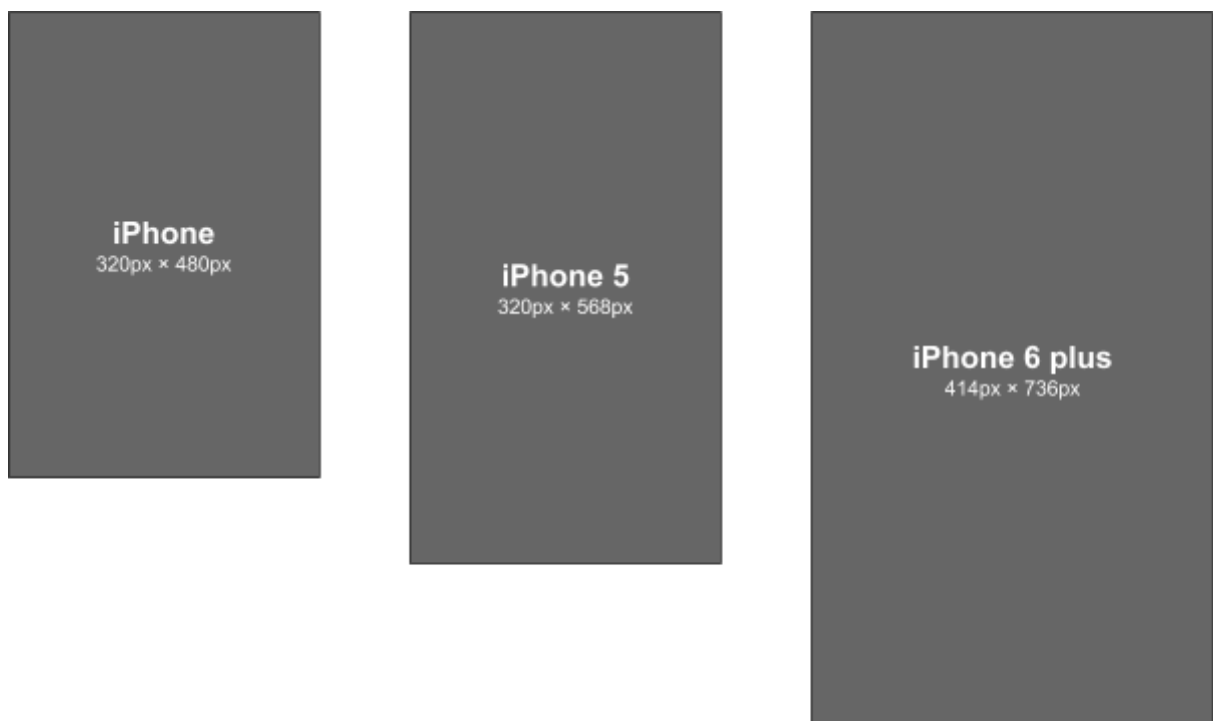
Les appareils mobiles comme les téléphones mobile ou les tablettes possèdent en outre une surface virtuelle nommée surface du rendu de l'appareil (*device-width*). Cette taille est définie par le constructeur de l'appareil, elle est généralement inférieure à la taille physique de l'écran. On peut admettre 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 <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 définit 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. | <code>width="device-width"</code> |
| height | Hauteur de fenêtre viewport. | <code>height="device-height"</code> |
| initial-scale | Niveau de zoom initial. | <code>initial-scale="1.0"</code> |
| minimum-scale | Niveau de zoom minimal. | <code>minimum-scale="0.5"</code> |
| maximum-scale | Niveau de zoom maximal. Attention, la valeur "1.0" interdit le zoom et peut rendre vos pages inaccessibles. | <code>maximum-scale="3.0"</code> |
| user-scalable | Possibilité à l'utilisateur de zoomer. Attention, la valeur "no" interdit le zoom et peut rendre vos pages inaccessibles. | <code>user-scalable="yes"</code> |

6.4. Les Media Queries

Maintenant que l'espace disponible dans le viewport du navigateur est celui du device-width, il 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 appareils dont la largeur du viewport est inférieure 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 l'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 (portrait 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 (<i>ET</i> logique) |
| only | Pour spécifier uniquement une seule option |
| not | Pour la négation (<i>NON</i> logique) |
| , | Permet de faire un <i>OU</i> logique (séparer les critères par une virgule) |

Par exemple, pour cibler uniquement les appareils 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 */  
}
```

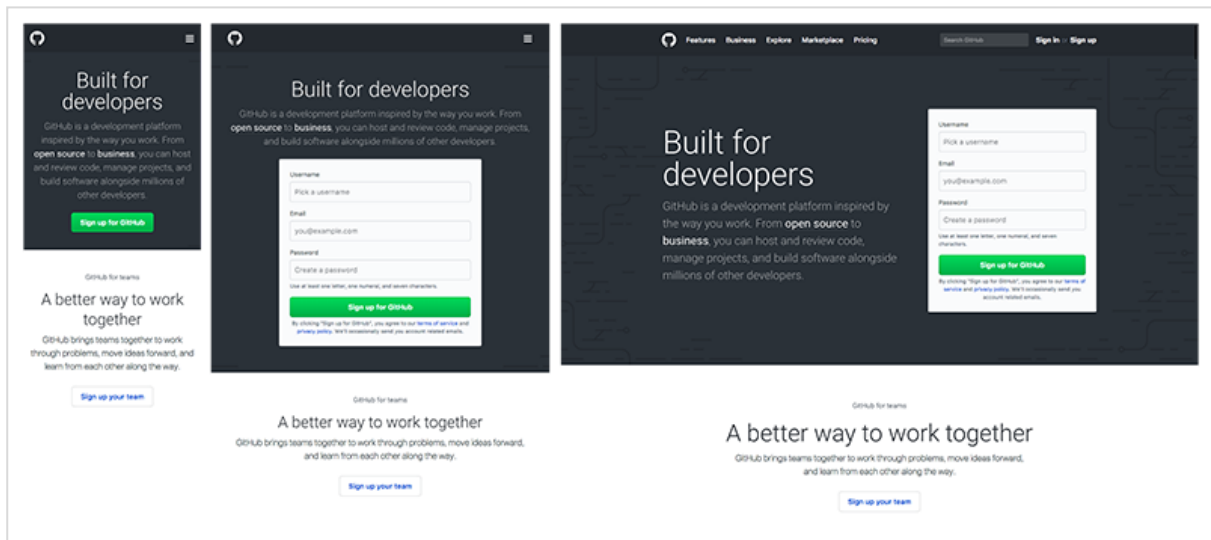
Par cette déclaration dans le HEAD du document HTML

```
<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. 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.

