

R1.02 – Développement d'interfaces web

TP 4 – Responsive design

Le but de ce TP est d'utiliser les principes du "*responsive design*" pour faire en sorte qu'un site web puisse être vu avec le même confort sur le grand écran d'un ordinateur, sur l'écran moyen d'une tablette ou sur le petit écran d'un smartphone, sans que l'affichage ne devienne trop petit et nécessite de faire des zoom et des défilement horizontaux de la page.



On modifiera le site web de cinéma du TP3 pour répondre à ce besoin. On utilisera pour cela les **media queries**, qui nous permettront de savoir quelle est la taille de l'écran de consultation et donc d'adapter les propriétés CSS en conséquence. On ciblera trois types d'écrans :

Appareil	Largeur	Condition
Smartphone	320 à 480 pixels	(max-width: 480px)
Tablette	481 à 768 pixels	(min-width: 481px) and (max-width: 768px)
Ordinateur	769 pixels et plus	(min-width: 769px)

Attention : pour les smartphones et tablettes, il ne s'agit pas de leur véritable définition en pixels qui est en général beaucoup plus élevée (par exemple 640 pixels de large pour l'iPhone 5, 1536 pixels pour l'iPad 3) mais d'une définition virtuelle (device-width) que croit avoir le navigateur mobile. Pour plus d'informations, lisez cet article :

<http://www.alsacreations.com/article/lire/1490-comprendre-le-viewport-dans-le-web-mobile.html>

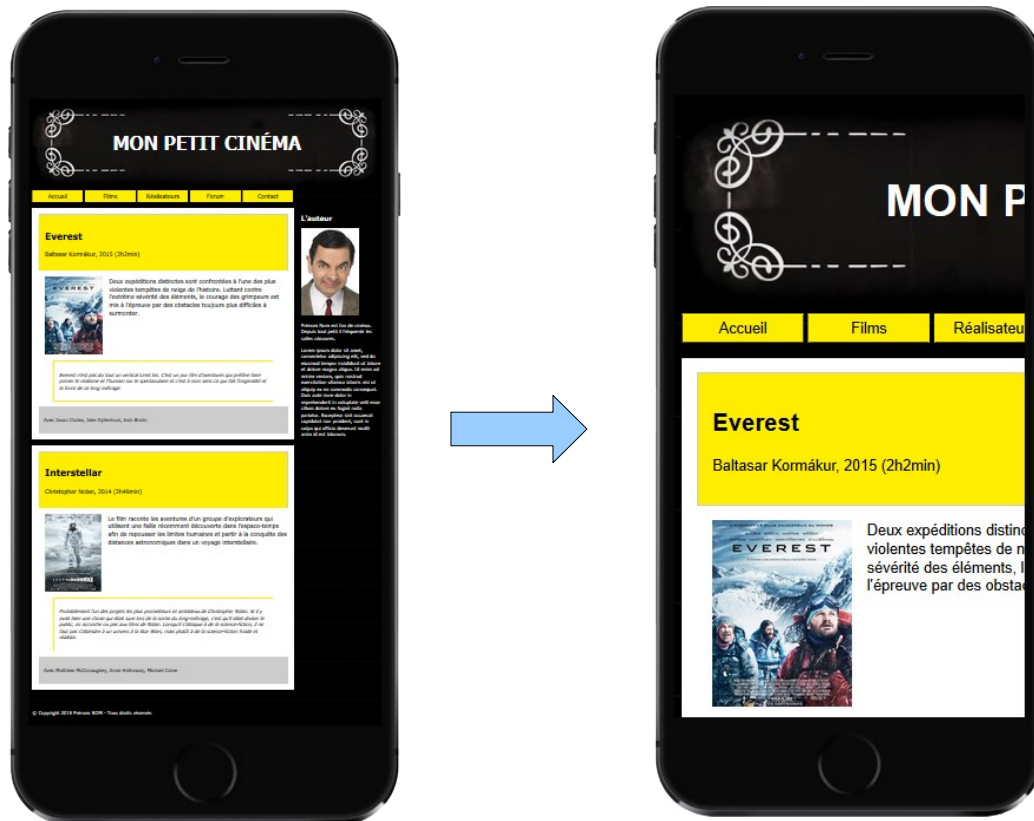
Lancez le site web du TP3 et redimensionnez la fenêtre du navigateur :



Vous pouvez constater que le site n'est pas adaptatif car la page ne s'adapte pas à la nouvelle largeur de la fenêtre. Lorsque la largeur de la fenêtre est inférieure à 900 pixels, un ascenseur horizontal apparaît pour pouvoir faire défiler la page, car nous avons fixé sa largeur à 900 pixels dans les propriétés de l'id **#page**.

1. Copiez vos fichiers **tp3.html** et **style_tp3.css**, appelez les **tp4.html** et **style_tp4.css**. N'oubliez pas de modifier **tp4.html** pour qu'il charge **style_tp4.css** !
2. Dans la suite du TP, pour tester votre page et vérifier qu'elle s'adapte correctement à la largeur du navigateur, plusieurs possibilités s'offrent à vous :
 - 1) Réduisez tout simplement la taille de la fenêtre du navigateur à la souris !
 - 2) Dans Firefox : aller dans le menu > Développement et cocher la case **Vue adaptative**.
 - 3) Dans Google Chrome : aller dans le menu > Outils > Outils de Développement et cliquer sur l'onglet **Emulation**.
 - 4) Il existe aussi de nombreux sites qui permettent de simuler différentes tailles de navigateurs, comme par exemple :
 - <http://www.emulateurmobile.com/>
 - <http://www.responsinator.com/>
 - <http://ami.responsivedesign.is/>
 - <http://quirktools.com/screenfly/>

3. Les navigateurs de mobiles vont afficher la page avec une largeur par défaut, celle du "viewport", qui est environ celle d'un ordinateur (980px pour Safari Mobile, mais cette valeur peut changer d'un navigateur à un autre). Il va donc y avoir un dézoom qui fait que votre site sera affiché en tout petit sur un mobile (1) :



(1) Viewport par défaut

(2) Modification du viewport

Pour éviter ça (2), il suffit de dire au navigateur mobile qu'on veut que la largeur du site respecte celle du mobile, en ajoutant la balise **<meta>** suivante dans la zone **<head></head>** de votre document HTML pour modifier le viewport :

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

4. Dans l'id **#page**, modifiez la propriété **width: 900px** en **max-width: 900px** : ainsi, la largeur ne sera pas fixe de 900 pixels, mais s'adaptera à la largeur de la fenêtre, en étant au maximum large de 900 pixels.
5. Utilisez les media queries pour charger une image de fond différente en fonction de la largeur de l'écran : **background.jpg** pour les grands écrans, et une simple couleur noire pour les tablettes et smartphones.
6. Dans le cas d'un smartphone ou d'une tablette, enlevez l'image de fond de la bannière et mettez une simple couleur de fond noire. Utilisez une taille de balise **<h1>** de 200% pour le titre de la bannière ("mon petit cinéma").

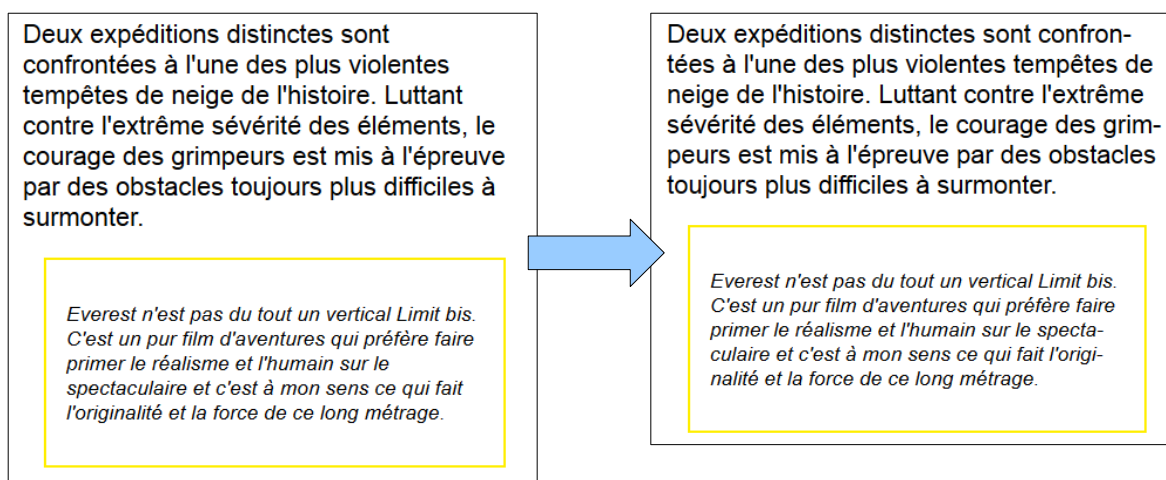
7. Dans le cas d'un smartphone, faites en sorte que les images des articles ne soient pas flottantes, qu'elles soient centrées et qu'elles ne dépassent pas la largeur de l'écran (**max-width : 100%; height : auto;**)
8. Lorsqu'on définit les dimensions d'un élément de type bloc (**<p>**, **<div>**, etc.) avec les propriétés **width** et **height**, ces dimensions sont augmentées lorsqu'on définit une épaisseur de bordure et un padding.
- <https://developer.mozilla.org/fr/docs/Web/CSS/box-sizing>

Pour faire en sorte que ces dimensions ne soient pas modifiées, on peut utiliser la propriété **box-sizing** en lui donnant la valeur **border-box**. Pour faire en sorte que tous les blocs soient affectés, utilisez le sélecteur universel :

```
* {  
    box-sizing : border-box;  
}
```

9. Avec la propriété **hyphens** le navigateur peut effectuer des césures (des coupures dans les mots) pour un affichage plus esthétique :

```
p {  
    hyphens : auto;  
}
```



Sans hyphens:auto

Avec hyphens:auto

Pour que cela fonctionne correctement, il faut que le navigateur connaisse la langue dans laquelle est rédigé le site. Vous pouvez le préciser dans le document HTML, en attribut de la balise **<html>** qui se trouve au début du document. Par exemple pour du français :

```
<html lang="fr">
```

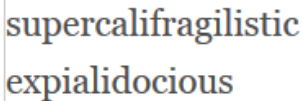
10. Lorsqu'on affiche du texte dans un bloc très étroit, il se peut que des mots trop longs dépassent du cadre. Pour éviter cela, on peut utiliser la propriété **word-wrap** en lui donnant la valeur **break-word** pour permettre des césures au milieu de mots :

```
p {  
    word-wrap : break-word;  
    overflow-wrap : break-word;  
}
```



supercalifragilisticexpialidocious

*Sans **word-wrap***



supercalifragilistic
expialidocious

*Avec **word-wrap : break-word;***

11. On peut faire disparaître un élément de la page web en lui ajoutant la propriété **display: none;** dans le CSS. On peut par exemple créer une classe **.petit_ecran** qui contient cette propriété si la largeur d'écran est trop petite :

```
@media screen and (max-width: 768px) {  
    .petit_ecran { display: none; }  
}
```

On peut ensuite affecter plusieurs classes à un élément HTML en séparant leurs noms par des espaces :

```
<div class="nomClasse petit_ecran">
```

Dans le cas du bloc **<aside>** ayant pour id **#sidebar**, on peut aussi lui appliquer la classe **.petit_ecran** comme suit :

```
<aside id="sidebar" class="petit_ecran">
```

Utilisez ce principe pour faire disparaître en mode smartphone et tablette le bloc **<aside>** de droite contenant la biographie de l'auteur.

12. Dans le cas d'un smartphone ou d'une tablette, faites en sorte que la **<section>** prenne toute la largeur de l'écran, étant donné qu'on n'affiche plus le bloc **<aside>** avec la biographie.
13. Dans le cas d'un smartphone, réduisez les marges et les padding des différents éléments de l'interface pour profiter au maximum de la zone d'affichage. Par exemple, pour les articles, utilisez un padding de 0.5em au lieu de 1em.
14. Dans le cas d'un smartphone, faites en sorte de disposer à la verticale les options du menu et qu'elles occupent toute la largeur de l'écran.

Note :

Des frameworks CSS permettent de simplifier cette tâche de création de sites adaptatifs, comme *Bootstrap* que nous utiliserons dans un prochain TP, ou *Blueprint*, *Foundation*, *Knacss*, etc.