

COURS HTML/CSS SEMAINE 2 JOUR 2

LE DESIGN RESPONSIVE

Le design responsive vise à rendre un site adaptés à tout média.

Les sites aujourd'hui sont majoritairement consultés sur des téléphone portable, plus de 80% des sites internet sont conçus pour être compatibles avec des téléphone portable. On parle alors de conception « mobile first ».

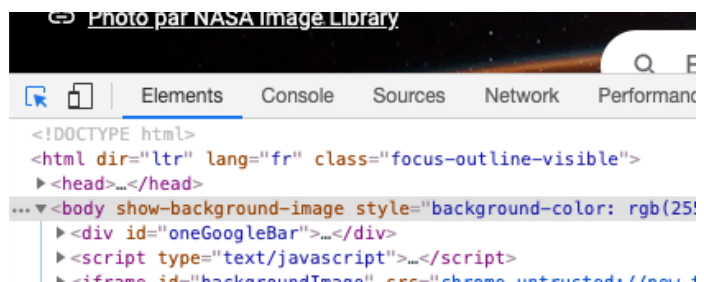
On veillera donc dès la conception du site à penser à des versions pour différents médias. Le site sera donc décliné en trois versions : les ordinateurs de bureaux, les téléphones mobiles et les tablettes tactiles.

Taille des médias les plus courant :

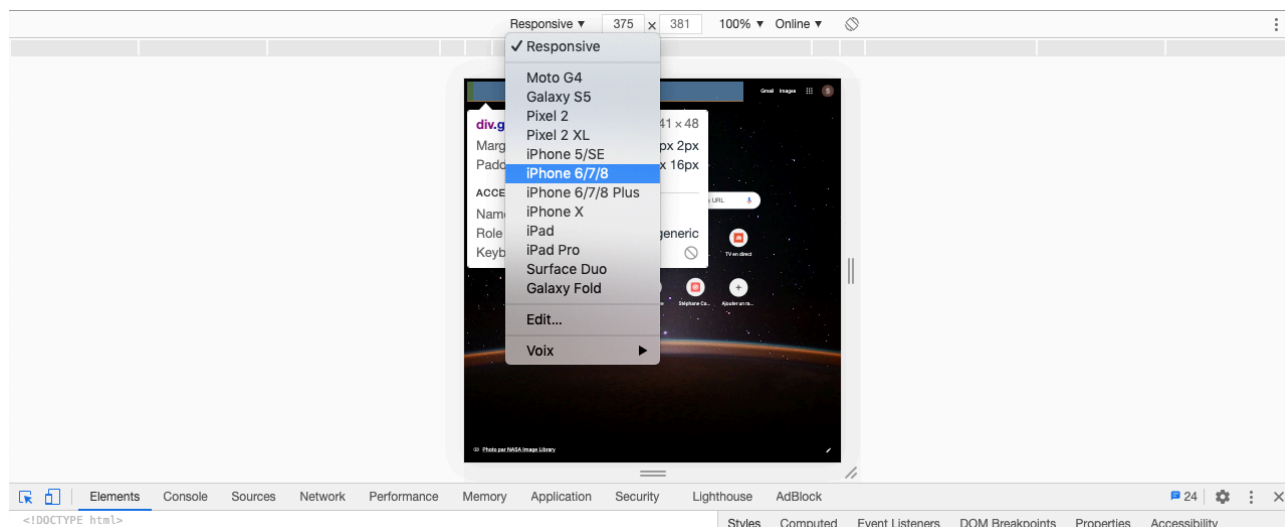
- Téléphones mobiles : en dessous 480px
- Tablettes : jusqu'à 768px
- Tablette en mode portrait : jusqu'à 1024px
- Ordinateur portable : jusqu'à 1600px
- Ordinateur de bureau : plus de 1600px

AVOIR UN APERCU DANS VOTRE NAVIGATEUR DE LA VERSION MOBILE DU SITE

Tous les navigateurs possède une console qui permet d'afficher le code. On y accède avec la touche F12 pour Windows et Cmd Alt i pour Mac. En cliquant sur l'icône **Toggle Device toolbar** (la deuxième en partant de la droite) Vous basculez dans différents modes d'affichage.



Vous pouvez alors choisir le type de média dans une liste déroulante ou cliquer sur l'une des portions de la barre grisée (à gauche les plus grands écran à droite les smartphones)



C'est de cette manière que nous pourrons nous assurer de la compatibilité de notre site avec tous les médias.

LA BALISE META VIEWPORT

La première chose à faire, sera de déclarer une balise méta pour prendre en considération les appareils mobiles.

Cette balise **meta viewport** devra être placée dans l'entête du document HTML :

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

LES MEDIA QUERIES

Les media queries ou requêtes media sont apparues avec l'avènement de la version CSS3. Ils nous permettent d'établir de nouvelles règles CSS qui vont s'appliquer à différentes résolutions d'écran.

APPLIQUER UNE REQUÊTE MEDIA : OÙ & COMMENT

Une première solution consistera à créer une nouvelle feuille de style pour les media queries en indiquant un lien dans l'en-tête du fichier HTML (<head>)

Ce lien possédera en plus un attribut **media** qui va spécifier une résolution d'écran :

Ex:

```
<link rel="stylesheet" media="screen and (max-width: 1024px)" href="gros-smartphones.css" />
```

L'autre solution, plus simple, consiste à écrire les requêtes médias dans les feuilles de style existantes. Dans ce cas on inscrira dans la feuille de style une requête concernant une résolution d'écran:

Ex:

```
@media screen and (max-width: 1024px)
{
    .selecteur-flex
    {
        wrap: nowrap;
    }
}
```

Dans la mesure où le code est lu de haut en bas, il faudra veiller à mettre ces nouvelles règles en queue de votre feuille de style, sans quoi, les modifications ne pourraient pas être prise en compte.

Les spécifications dans `@media` sont nombreuses. Elles peuvent indiquer une hauteur ou une largeur de la fenêtre (height ou width), une hauteur et une largeur minimum (max-height et max-width) ou l'orientation du périphérique (portrait ou paysage).

Ex:

```
@media all and (min-width: 1024px) and (max-width: 1280px)
```

Exercice :

Vous allez déterminer à la fin de votre feuille de style un ensemble de règles pour les médias inférieurs à 1024px

```
@media screen and (max-width: 1024px)
```

```
{
    p
    {
        color: orange;
        background-color: navy;
        font-size: 0.5em;
    }
}
```

Avec votre navigateur, vous pourrez tester l'application de cette règle de deux manières : Vous pouvez simplement réduire la fenêtre de votre navigateur ou utiliser la console du navigateur

Ensuite, vous aller créer de nouvelles requêtes médias que vous mettrez dans l'ordre : des plus grandes tailles d'écran au plus petites.

1024 pour les tablettes en mode paysage, 760 pour les tablettes en mode portrait ou des grand smartphone en mode paysage et 480 px pour les téléphones mobile.

Vous y appliquerez différentes règles pour mettre en évidence les points de bascule. On parle de break-point.

Vous serez alors en mesure de réajuster la mise en page en jouant sur les marges intérieures & extérieur, la taille des polices, les dimensions de largeur et hauteur.

RENDRE LES IFRAMES RESPONSIVE

Les iframes possèdent tous des valeurs embarquées de hauteur et de largeur qui peut poser problème pour les rendre plus petites. La première chose à faire sera de supprimer dans la balise iframes les attributs de largeur et de hauteur et plutôt les déclarer dans notre feuille de style.

Le problème d'adaptation des iframes peut aussi être résolu en utilisant des propriétés de positionnement.

La première étape consiste à insérer votre balise iframe sans attributs de hauteur et de largeur dans un conteneur, comme <div>

```
<div class="container">
  <iframe class="responsive-iframe" src="https://www.youtube.com/embed/
tgbNymZ7vqY"></iframe>
</div>
```

L'étape suivante consiste à ajouter du CSS

Ajoutez une valeur en pourcentage pour padding-top afin de conserver le rapport hauteur/largeur du DIV conteneur. L'exemple suivant créera un rapport d'aspect de 16:9, qui est le rapport d'aspect par défaut des vidéos YouTube.

```
.container {
  position: relative;
  /*overflow: hidden;*/
  /*width: 100%;*/
  padding-top: 56.25%; /* 16:9 Aspect Ratio (9 divisé 9 par 16 = 0.5625) */
}
```

/* Ensuite, rentrez ces quelques règles CSS à l'iframe pour qu'il remplisse complètement le conteneur dans sa largeur et sa hauteur. */

```
.responsive-iframe {
  position: absolute;
  top: 0;
  left: 0;
  bottom: 0;
  right: 0;
  width: 100%;
  height: 100%;
}
```