



L'UTILISATEUR **AU** **CENTRE** DU WEB

LE RESPONSIVE DESIGN

L'UTILISATEUR AU CENTRE DU WEB

- #Les media queries pour s'adapter à l'écran P.3
- #Les différents types de media P.4
- #Configurer une page Web Responsive..... P.5
- #Quatre tailles pour gérer tous les écrans P.6
- #Unités de mesure relative : em, rem et % P.7

LES MEDIA QUERIES POUR S'ADAPTER A L'ECRAN

Il existe plusieurs moyens de connaître la taille de l'écran d'un utilisateur dans les différents langages Web. Le Responsive Design est une technique qui permet d'adapter les styles d'une page Web pour qu'elle ait un impact optimisé sur ses différents supports de lecture, c'est pourquoi il doit être défini en CSS plutôt qu'en JavaScript par exemple. Il faut alors faire appel aux requêtes Media Queries qui permettent de définir des propriétés différentes selon la taille de l'écran (ou de la fenêtre).

Media Queries dans la balise <head>

Les feuilles de style seront chargées uniquement si la largeur de l'écran respecte les conditions définies dans la balise media=".."

Cette technique implique l'écriture de toutes les propriétés CSS dans plusieurs fichiers CSS

```
<link rel="stylesheet"
      media="screen and (max-width: 980px)"
      href="styleLarge.css" />

<link rel="stylesheet"
      media="all and (orientation: portrait)"
      href="stylePortrait.css" />
```

Media Queries dans le fichier CSS

Toutes les tailles sont définies dans le même fichier CSS, il faut intégrer le principe du "Mobile First" en commençant par définir les styles des largeurs les plus petites.

Avec cette technique il est possible de conserver des propriétés communes aux différents formats pour ne redéfinir que certaines propriétés CSS

```
@media screen and (max-width: 980px){
    p {
        font-size: 12px;
        color : red;
    }
}

@media all and (orientation: portrait){
    p {
        font-size: 14px;
    }
}
```

LES DIFFERENTS TYPES DE MEDIA

La règle CSS @media permet de spécifier le support sur lequel doivent s'appliquer des propriétés CSS. Initialement, la règle CSS @media permettait de définir une feuille de style spécifique pour l'impression avec le media print et une autre pour les écrans d'ordinateur avec media screen. Avec l'apparition de nouveaux lecteurs de page Web, la règle CSS @media a intégré de nouveaux types de média pour permettre une plus grandes optimisations.

@media all

Permet de spécifier des propriétés pour tous les types de media



```
@media all {  
  body {  
    font-family: 'Arial', sans-serif;  
  }  
}
```

@media print

Permet de spécifier des propriétés destinées à l'impression



```
@media print {  
  body {  
    font-size: 14pt  
  }  
}
```

@media screen

Permet de spécifier des propriétés destinées aux écrans d'ordinateurs










```
@media screen {  
  body {  
    font-size: 12px  
  }  
}
```

D'autres types de media permettent par exemple d'optimiser une page Web pour les supports en braille ou pour l'affiche sur des télévisions ou des projecteurs.

CONFIGURER UNE PAGE WEB RESPONSIVE

Associé à la règle @media CSS, la balise HTML meta viewport permet de contrôler la mise en page sur les navigateurs mobiles. Cette balise est essentielle pour maîtriser l'affichage des CSS sur un smartphone car elle permet d'initialiser et de bloquer le zoom. En effet, contrairement à un ordinateur, un smartphone permet de zoomer dans l'écran, il est donc primordial de maîtriser le niveau de zoom initial pour pouvoir configurer efficacement les CSS.

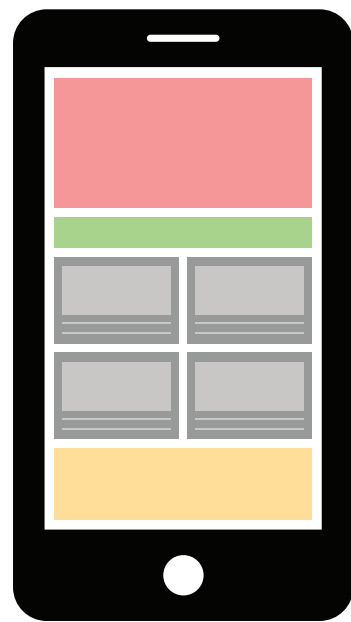
META VIEWPORT Balise à placer dans la balise <head>..</head>	 <code><meta name="viewport"</code>
CONTENT Permet d'effectuer la configuration du viewport	 <code>content="</code>
WIDTH Largeur de la fenêtre, peut être défini en pixel	 <code>width=device-width,</code>
HEIGHT hauteur de la fenêtre, peut être défini en pixel	 <code>height=device-height,</code>
INITIAL-SCALE Niveau de zoom de base	 <code>initial-scale=1,</code>
MAXIMUM-SCALE Niveau de zoom maximal	 <code>maximum-scale=3,</code>
USER-SCALABLE Possibilité ou non de zoomer dans la page Web	 <code>user-scalable=yes" ></code>

La configuration de base du viewport pour le responsive design est width=device-width et initial-scale=1.0

QUATRE TAILLES POUR GERER TOUS LES FORMATS

Selon les constructeurs, les supports mobiles et autres n'ont pas tous la même taille, il est néanmoins possible de définir des formats qui correspondront à tous les supports sans faire de Media Queries pour chaque modèle disponible.

SMARTPHONE (PORT.)



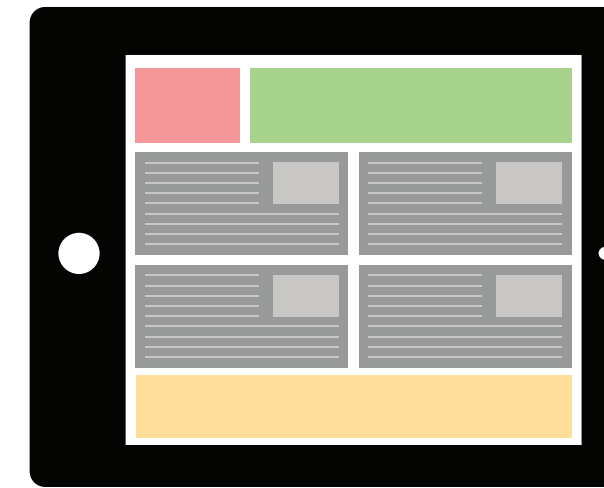
```
@media screen and  
(max-width: 320px) {  
    /* Propriétés CSS */  
}
```

SMARTPHONE (PAYS.) & TABLETTE (PORT.)



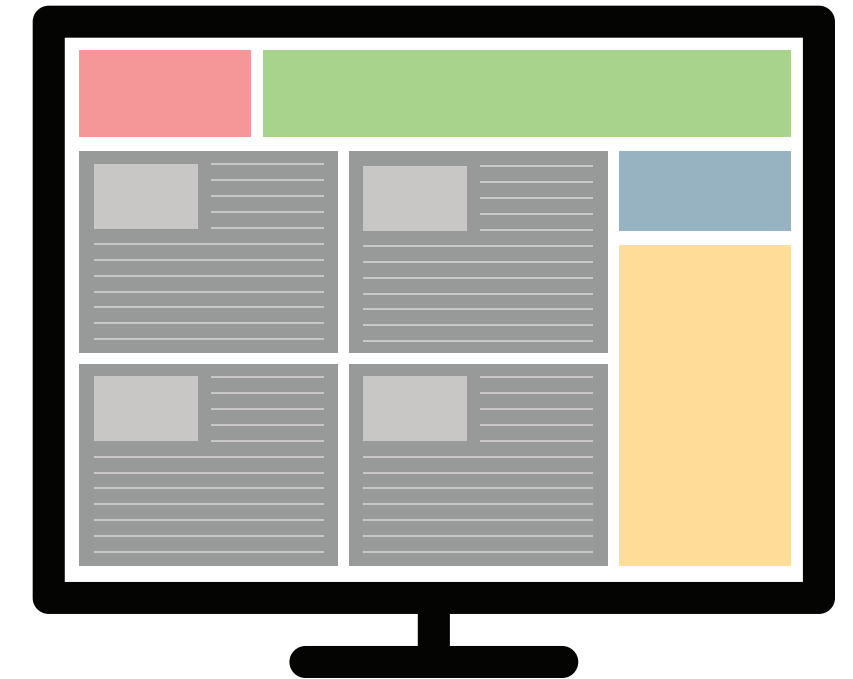
```
@media screen and  
(min-width: 520px) {  
    /* Propriétés CSS */  
}
```

TABLETTE (PAYS.)



```
@media screen and  
(min-width: 720px) {  
    /* Propriétés CSS */  
}
```

DESKTOP



```
@media screen and  
(min-width: 1060px) {  
    /* Propriétés CSS */  
}
```

En responsive design il faut prendre le format le plus petit comme base, le media query pour les smartphone devient donc inutile

UNITES DE MESURE RELATIVE : EM, REM ET %

Il existe différentes unités de mesure en CSS dites absolues ou relatives. Les unités absolues comme le pixel (px) ou le point (pt) ne seront pas modifiées selon les possibilités d'affichage et sont donc utilisées pour maîtriser le rendu d'impression. En Responsive Design il est essentiel d'utiliser des unités de mesure relatives comme le cadratin (em, rem) ou le pourcentage (%) car elles ne sont pas fixes mais proportionnelles à la taille du texte du parent.

Taille par défaut

La taille par défaut des navigateurs est de 16px, pour lui donner une valeur de 10px nous le réduisons à 62,5%



```
html {  
  font-size: 62.5%;  
}
```

Taille des balises <p>

L'unité rem est proportionnelle à la balise HTML, les balises <p> font donc 12px (10 x 1,2 = 12)



```
p {  
  font-size: 1.2rem;  
}
```

Taille des balises <h1>

L'unité rem est proportionnelle à la balise HTML, les balises <h1> font donc 22px (10 x 2,2 = 22)



```
h1 {  
  font-size: 2.2rem;  
}
```

Taille des balises

L'unité em est proportionnelle à la balise parent, les balises font donc 11px (22 x 0,5 = 11)



```
h1 span {  
  font-size: 0.5em;  
}
```

Pour utiliser l'unité de mesure rem il faut absolument spécifier la taille de la balise <html>