HTML5

HTML Tutorial

RESPONSIVE DESIGN

HTML Tutorial

INTRODUCTION AU RESPONSIVE DESIGN

Définition du « responsive design »

- Les mots « responsive design », en anglais, peuvent être traduits en français par « design adaptable ».
- Par extension, le « responsive design » désigne généralement l'ensemble des techniques nous permettant d'adapter les pages web à toutes les tailles d'écrans afin d'avoir un bon rendu.
- La problématique du responsive design est apparue avec l'apparition des tablettes et des Smartphones pouvant se connecter à Internet : en effet, comment faire pour que mon site s'affiche aussi bien sur un petit écran de téléphone portable que sur un ordinateur de salon ?

Trois façons d'adapter son site sur tous les écrans

Aujourd'hui, il existe trois façons d'adapter son site afin d'avoir un bon rendu visuel sur toutes les tailles d'écrans :

- Créer un site dédié pour chaque terminal différent (un site mobile, un site pour ordinateur, etc.) ;
- Créer une application mobile en plus de notre site web (gérant l'environnement Androïd et iOS) ;
- Utiliser les outils offerts par le HTML et le CSS et créer une version «responsive» du site.

C'est la troisième méthode qui nous intéresse ici.

HTML Tutorial

LE VIEWPORT ET L'UTILISATION
DES POURCENTAGES

Présentation du viewport

- Le « viewport » correspond à la taille de la fenêtre web du visiteur.
- Une première méthode, très simple, pour commencer à optimiser son site web pour un affichage sur mobile ou tablette est d'utiliser et de manipuler le viewport.
- En HTML5, on peut prendre « contrôle » de la taille de la fenêtre grâce à l'élément **meta** et aux attributs **name** et content.
- On demande à ce que nos pages web s'adaptent à la taille de la fenêtre de chacun des visiteurs.
- Même si le résultat n'est pas toujours parfait, c'est une première approche.

Présentation du viewport

- Voilà ce qu'il faut écrire :

```html
<!DOCTYPE html>
<html>
    <head>
        <title>Responsive design</title>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="stylesheet" href="styles.css">
    </head>

    <body>
        <h1>Responsive design</h1>
        <img src="ecosse.jpg" alt="arbre en Ecosse">
        <p>Un petit lorem ipsum pour la route...Lorem ipsum dolor sit amet,
        aenean dictum viverra, vulputate molestie, nullam dolor habitasse
        mattis montes nullam tempus, suscipit et, magna lacus sociis mauris
        ac. Ullamcorper ultrices massa eu sagittis, metus mi pellentesque sit
        ut metus, urna ipsum orci scelerisque mollis fusce duis, tristique donec
        eros urna phasellus nibh nullam, lacinia proin nibh. Accumsan et
        phasellus amet consectetuer eleifend cras, semper ac erat enim neque
        lobortis velit. Ante luctus elit, in viverra. Ut pharetra consequat
        neque ipsum ullamcorper etiam, leo nunc, felis lorem sollicitudin in
        quis nam turpis, id volutpat velit magna, vestibulum justo et vitae.
        Volutpat sed suspendisse ante augue purus velit, in accumsan dis amet.
        Amet cras lorem amet, purus faucibus rutrum sagittis diam, wisi hendrerit
        lacus pretium duis, pede sed duis et doloremque nullam nonummy.
        Magna imperdiet.</p>
    </body>
</html>
```

Présentation du viewport

- Le code **width=device-width** va faire en sorte que la page s'adapte à la taille de l'écran du visiteur de telle façon que les éléments HTML prennent la place disponible.
- Le code **initial-scale=1.0** définit le niveau de zoom initial lorsque la page est chargée par le navigateur.

:

Présentation du viewport

- Voici le résultat avec et sans utilisation de **meta viewport** du code HTML pour un affichage sur mobile :

Utilisation des valeurs en pourcentage en CSS

- En plus de l'emploi de l'élément meta, on peut utiliser les tailles de certains éléments plutôt que des valeurs en pixels.
- Si cela est fait efficacement (*) cela aura pour effet de redimensionner les différents éléments des pages en même temps que la fenêtre va se rétrécir ou s'agrandir.
- Avec cette méthode, ne pas oublier que les tailles exprimées sont un pourcentage de la taille de l'élément parent des éléments ciblés.
- Ainsi, si vous définissez une taille égale à 100% pour votre élément body, puis ensuite une largeur de 50% pour un div enfant direct du body, la taille du div sera toujours égale à 50% de celle du body.
- Si maintenant vous définissez à nouveau une largeur égale à 50% pour les paragraphes à l'intérieur de votre div, les paragraphes occuperont 50% de la largeur du div …

(*) cf le site AlsaCréations

Utilisation des valeurs en pourcentage en CSS

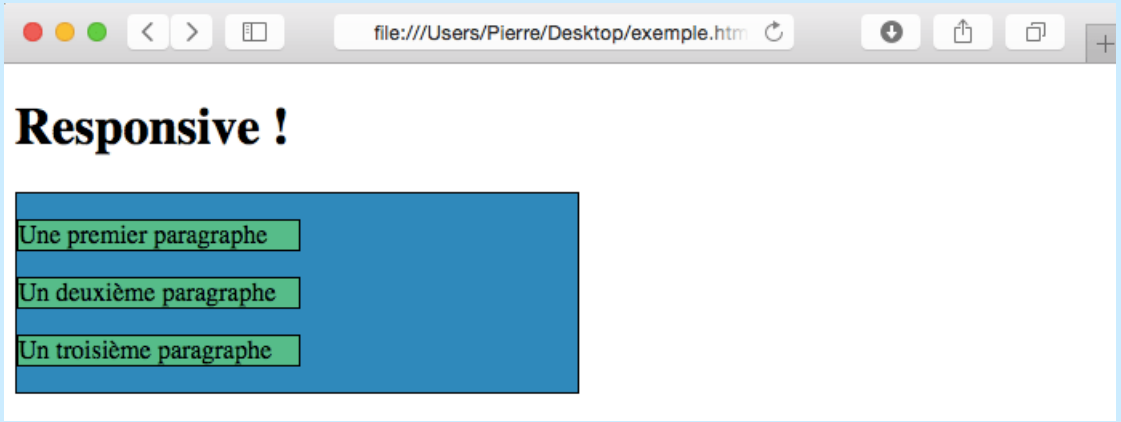
- (Suite)

```html
<!DOCTYPE html>
<html>
    <head>
        <title>Des pourcentages</title>
        <meta charset='utf-8'>
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="stylesheet" href="styles.css">
    </head>
    <body>
        <h1>Responsive !</h1>
        <div>
            <p>Une premier paragraphe</p>
            <p>Un deuxième paragraphe</p>
            <p>Un troisième paragraphe</p>
        </div>
    </body>
</html>
```

```css
body{
    width: 100%;
}

div{
    width: 50%;
    border: 1px solid black;
    background-color: #08B;
}

p{
    width: 50%;
    border: 1px solid black;
    background-color: #0B8;
}
```

Utilisation des valeurs en pourcentage en CSS

- Résultat :



Limites de ces deux premières méthodes

- Cependant, ces deux premières méthodes peuvent ne pas se révéler suffisantes. En effet, selon la façon dont les pages sont construites, il pourrait être plus approprié de les réagencer totalement selon la taille de l'écran du visiteurs.
- Par exemple, pour un affichage sur mobile, on pourra préférer enlever du contenu et afficher les différents éléments en longueur plutôt qu'en largeur.
- La meilleure méthode pour faire cela est d'utiliser ce qu'on appelle les "**media queries**".

HTML Tutorial

MEDIA QUERIES ET RESPONSIVE DESIGN

Présentation des media queries

- Les **media queries** correspondent à une technique introduite en CSS3 et qui se sert de la règle CSS @media.
- Grâce aux media queries, on peut appliquer différentes propriétés ou différentes valeurs à des éléments HTML selon la taille de l'écran et le type de media utilisé (écran, imprimante, etc.) par le visiteur.
- Cela permettra par exemple de ne pas afficher certains éléments pour des tailles d'écran trop petites ou de réorganiser les pages web grâce à des propriétés CSS.

Utilisation de @media en CSS

- @media précise le type de media pour lequel les déclarations CSS doivent s'appliquer. Dans la grande majorité des cas, ce sera **@media screen**, afin d'appliquer les propriétés à tous les medias dotés d'un écran (téléphone portable, tablette, ordinateur de bureau, etc.).
- Cependant, sachez que l'on dispose également de **@media print,** afin de s'adapter à des imprimantes ou encore **@media speech**, pour les ordinateurs spéciaux qui vont « lire à haute voix » une page.

Utilisation de @media en CSS

- On peut également utiliser **@media all** pour appliquer des déclarations à tous les types de media.
- Avec @media screen, on doit préciser une taille ou un intervalle de tailles d'écran au sein duquel les propriétés vont s'appliquer. C'est cette fonctionnalité qui permet de créer un site responsive en soi.

Utilisation de @media en CSS

- Voici un premier exemple d'utilisation des media queries et de @media.
- HTML :

```html
<!DOCTYPE html>
<html>
    <head>
        <title>Responsive design</title>
        <meta charset="utf-8">
        <link rel="stylesheet" href="styles.css">
    </head>

    <body>
        <h1>Responsive design</h1>

    </body>
</html>
```

- Pour tester le code des pages suivantes : http://jsbin.com/gopega/146/edit?html,css,output

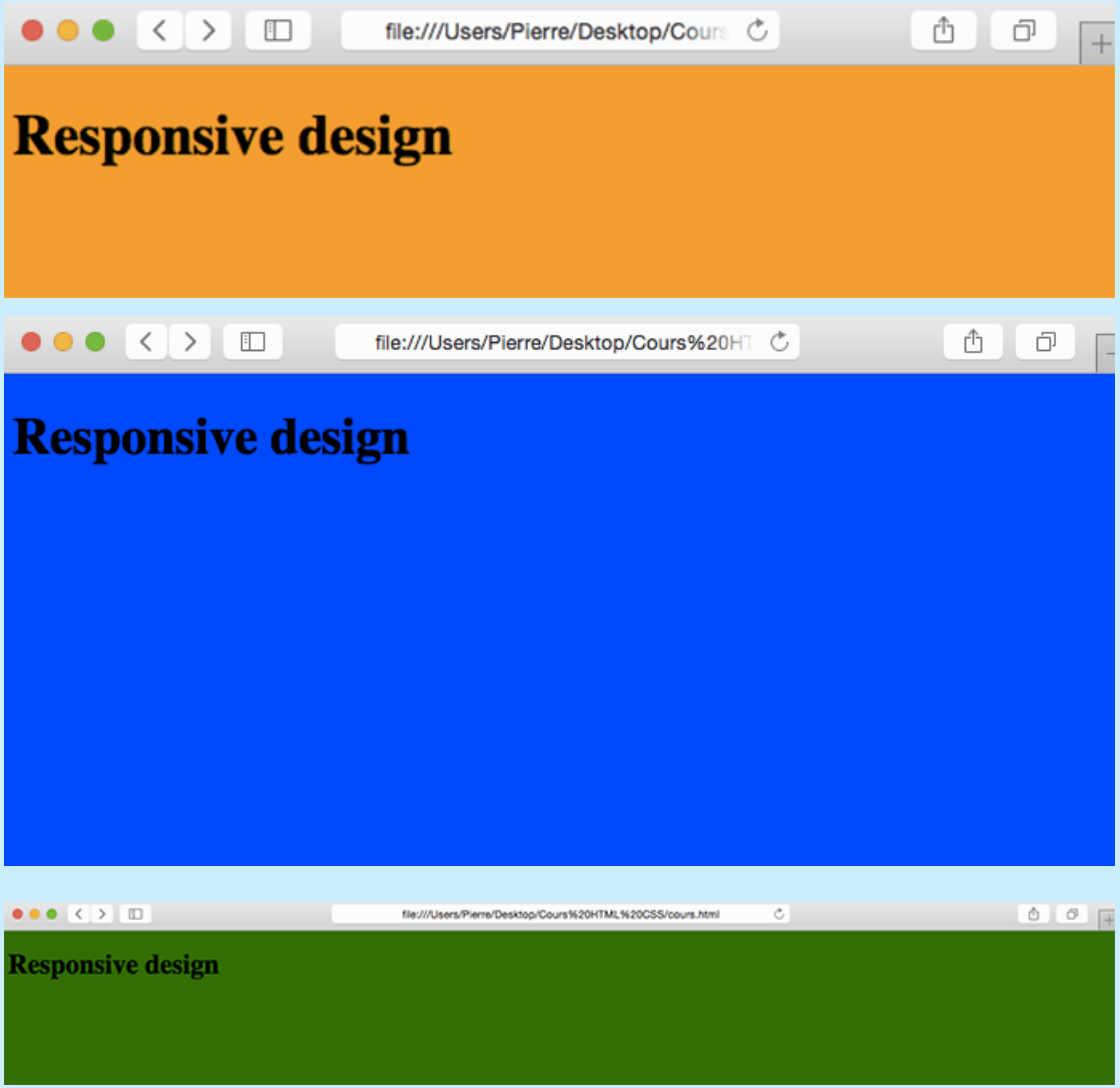
Utilisation de @media en CSS

- CSS :

```css
/*Si la taille de l'écran d'un visiteur est inférieure à 480px, la couleur
 *de fond de notre élément body (donc de notre page) sera orange*/
@media screen and (max-width: 680px){
    body{
        background-color: orange;
    }
}

/*Pour les écrans ayant une taille entre 481px et 1179px, la couleur de
 *fond de notre page sera bleue*/
@media screen and (min-width: 681px) and (max-width: 1279px){
    body{
        background-color: blue;
    }
}

/*Pour les écrans d'une taille supérieure à 1280px, le fond de la page sera
 *de couleur verte*/
@media screen and (min-width: 1280px){
    body{
        background-color: green;
    }
}
```

Utilisation de @media en CSS

- Résultat :



Utiliser les media queries pour changer l'agencement des éléments HTML

- Les media queries permettent de masquer certains éléments ou demodifier leur agencement selon la taille de l'écran cible.
- Pour cela, on utilise les propriétés **display**, **position** et les valeurs relatives en pourcentages de façon générale.

Utiliser les media queries pour changer l'agencement des éléments HTML

- Prenons par exemple l'hypothèse d'une page web composée de trois div côte à côte et d'un pied de page.

```html
<!DOCTYPE html>
<html>
    <head>
        <title>Responsive design</title>
        <meta charset="utf-8">
        <link rel="stylesheet" href="styles.css">
    </head>

    <body>
        <h1>Responsive design</h1>
        <div class="div1">
            <p>Je suis un paragraphe appartenant au premier div.</p>
        </div>
        <div class="div2">
            <p>Je suis un paragraphe appartenant au deuxième div.</p>
        </div>
        <div class="div3">
            <p>Je suis un paragraphe appartenant au troisième div.</p>
        </div>
        <div class="footer">
            <p>Je suis un pied de page.</p>
        </div>
    </body>
</html>
```

Utiliser les media queries pour changer l'agencement des éléments HTML

- CSS :

```css
body{
    margin: 0px;
    padding: 0px;
}

.div1, .div2, .div3{
    display: inline-block;
    width: 30%;
    height: 100px;
    border: 3px solid #AAA;
    margin: 1%;
}

.div1{
    background-color: orange;
}
.div2{
    background-color: #28B;
}
.div3{
    background-color: #2B8;
}

.footer{
    width: 100%;
    height: 200px;
    margin-top: 20px;
    border-top: 1px solid black;
    text-align: center;
    background-color: #CCC;
}
```

Utiliser les media queries pour changer l'agencement des éléments HTML

- Résultat :

- Pour tester le code : http://jsbin.com/gopega/147/edit?html,css,output

Utiliser les media queries pour changer l'agencement des éléments HTML

- Noter :
  - L'utilisation des valeurs en pourcentages pour les éléments div afin que ceux-ci grandissent ou rappetissent en même temps que la page web.
  - Le reset CSS des marges intérieures et extérieures sur notre élément body afin d'être sûr de tous travailler sur les mêmes bases.

- Pour l'affichage sur mobile, il faut afficher la page sur une seule colonne et enlever le pied de page afin de faciliter la lisibilité de la page
- Si l'on considère les écrans de mobiles d'une taille intérieure à 780px, on va utiliser @media avec les bonnes propriétés CSS afin d'avoir l'affichage souhaité sur téléphone portable , comme indiqué page suivante .
- Cela fonctionne de la façon suivante : les règles définies dans @media deviennent prioritaires sur les règles « générales » dès que la taille de l'écran passe sous les 780px.

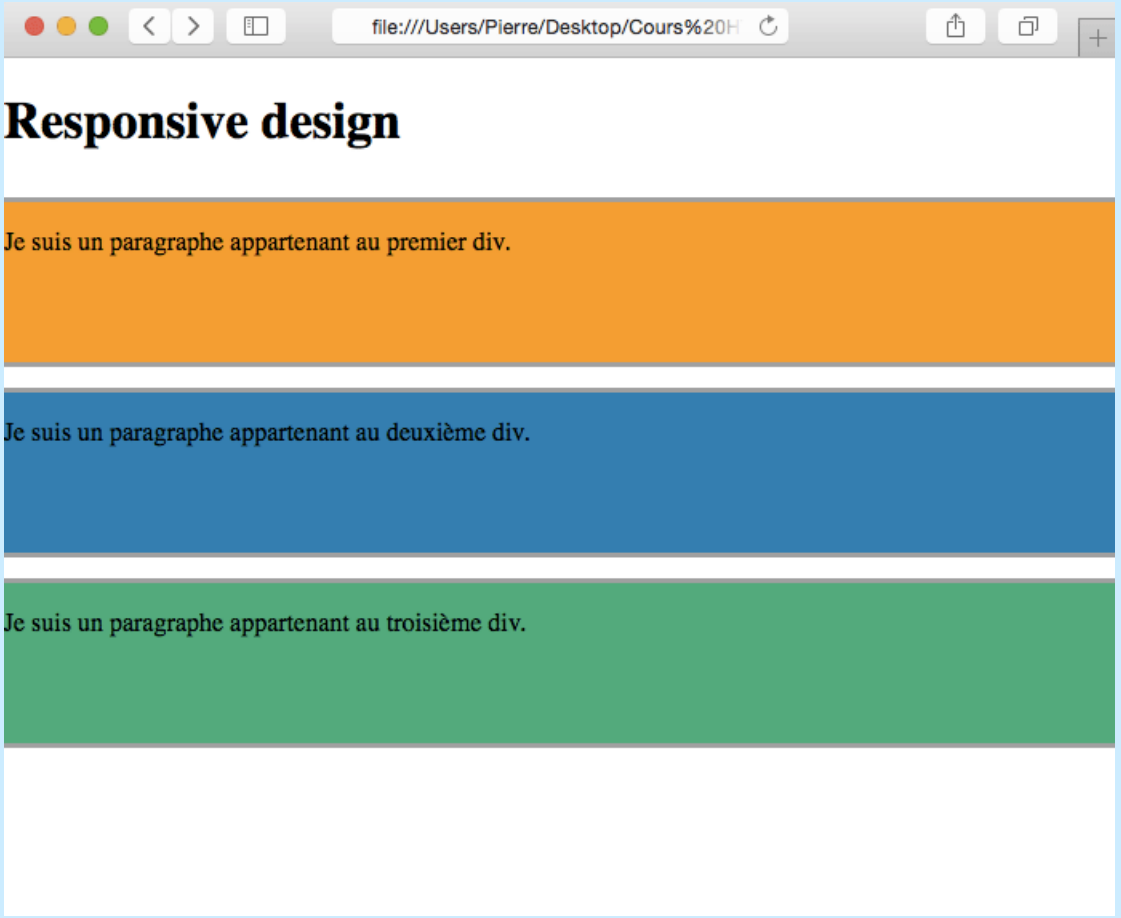
Utiliser les media queries pour changer l'agencement des éléments HTML

- (Suite)

```css
body{
    margin: 0px;
    padding: 0px;
}

.div1, .div2, .div3{
    display: inline-block;
    width: 30%;
    height: 100px;
    border: 3px solid #AAA;
    margin: 1%;
}

.div1{
    background-color: orange;
}
.div2{
    background-color: #28B;
}
.div3{
    background-color: #2B8;
}

.footer{
    width: 100%;
    height: 200px;
    margin-top: 20px;
    border-top: 1px solid black;
    text-align: center;
    background-color: #CCC;
}

@media screen and (max-width: 780px){
    .div1, .div2, .div3{
        width: 100%;
        border-right: none;
        border-left: none;
        margin-left: 0px;
        margin-right: 0px;
    }
    .footer{
        display: none;
    }
}
```

Utiliser les media queries pour changer l'agencement des éléments HTML

- Résultat :



- Pour tester le code : http://jsbin.com/gopega/148/edit?html,css,output

@media et orientation

- On peut également définir des règles spécifiques selon l'orientation de l'écran de vos visiteurs (utile pour l'affichage sur tablette ou iPhone entre autres).
- Il suffit d'utiliser « **orientation : landscape** » pour définir des règles spécifiques pour les écrans tournés en mode « paysage ».

```
@media screen and (orientation:landscape){
    /*Des déclarations CSS...*/
}
```

- (Suite)