

# MÉDIAS ET ÉLÉMENTS RÉACTIFS

Redimensionner une image en utilisant une largeur en %  
Images réactives via cookies et JavaScript  
Adapter une vidéo à la largeur de l'écran  
Redimensionner une image avec les requêtes média  
Modifier les éléments de navigation avec les requêtes média

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean nec eros magna, non fringilla lacus. Ut hendrerit, nisl id sagittis laoreet, turpis arcu posuere ipsum, non dictum tortor nisi sed lorem.



Lorem ipsum dolor sit amet

## 1.1 Images fluides avec CSS

# Exercice

Copiez ce fichier  
dans votre  
répertoire  
personnel.

- 01\_images\_fluides.zip
  - HTML
  - CSS

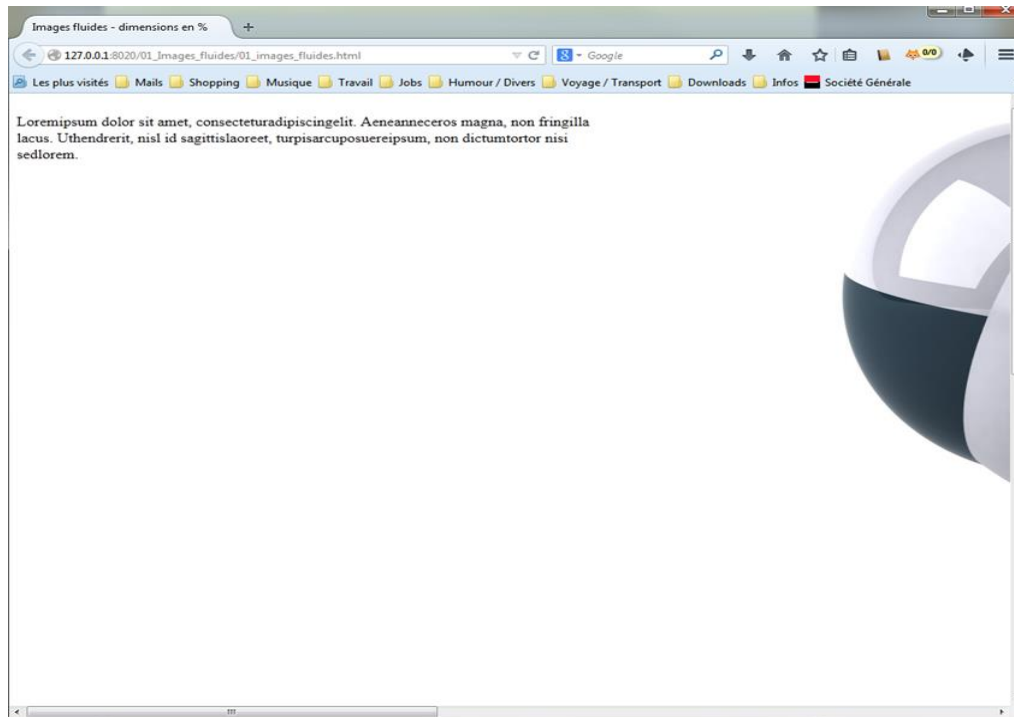
# Etude fichier HTML

```
1  <!DOCTYPE HTML>
2  <html>
3    <head>
4      <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5      <title>Images fluides - dimensions en %</title>
6      <link rel="stylesheet" type="text/css" href="01_Images_fluides.css" />
7    </head>
8    <body>
9
10     <p class="text">
11       Loremipsum dolor sit amet, consecteturadipiscingelit. Aeneanneceros magna, non fringilla lacus. Uthendrerit, nis
12     </p>
13
14     <div class="img-wrap" >
15       
16     <p>
17       Loremipsum dolor sit amet
18     </p>
19   </div>
20 </body>
21 </html>
```

# Etude du fichier CSS

```
1 p.text {  
2     float:left;  
3     width:60%;  
4 }  
5 div.img-wrap{  
6     float:right;  
7     width:40%;  
8 }
```

# Aperçu



Aperçu de l'image dans Firefox

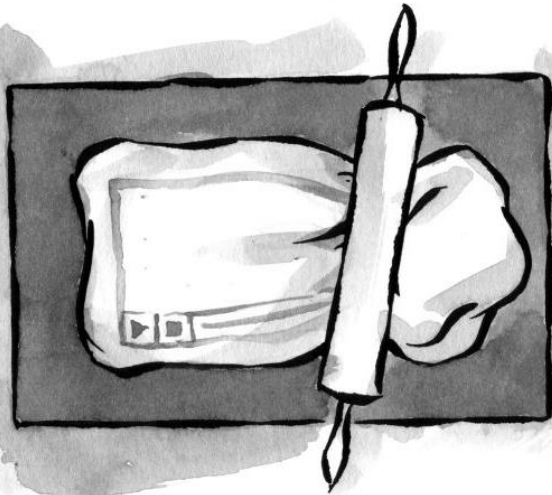
## Problèmes:

- Mise en page fluide mais pas de comportement réactif pour l'image
- L'image est affichée avec ses dimensions initiales statiques

## Exercice:

- Ecrire le code de la classe CSS `img.responsive`

Issue N°  
**284**



## Creating Intrinsic Ratios for Video

by **THIERRY KOBLENTZ** · May 26, 2009

Published in *CSS, HTML, JavaScript, Responsive Design* · 24 Comments

# 1.2 VIDÉOS FLUIDES

Source: <http://alistapart.com/article/creating-intrinsic-ratios-for-video>

# Exercice

Copiez ce fichier  
dans votre  
répertoire  
personnel.

## — 02\_Videos\_fluides.zip

- HTML
- CSS



# Balise <video> présentation

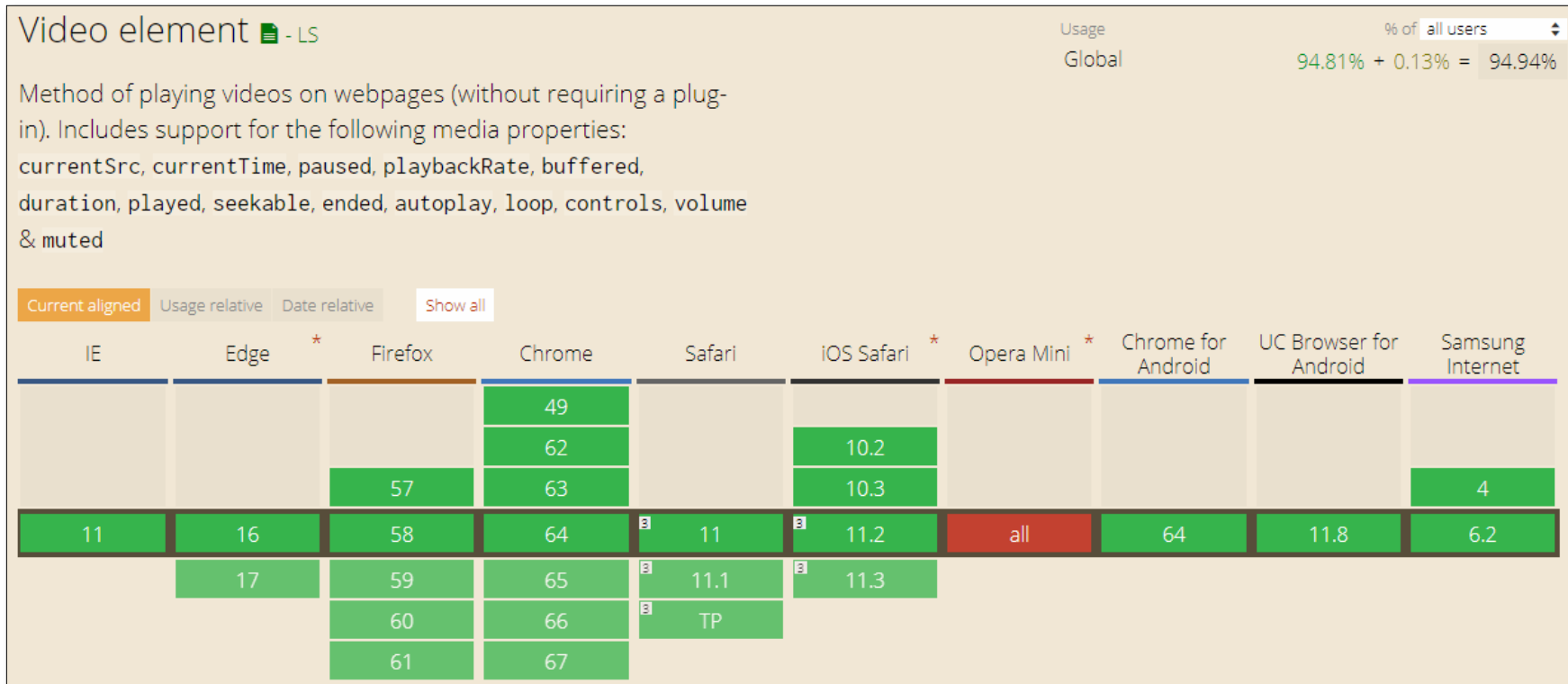
- Elle permet d'encapsuler dans une page Web des fichiers ou des flux vidéo
- Utilisation:

```
<video src="media/speaker.mp4" controls></video>
```

- La balise <video> comprend des attributs supplémentaires par rapport à la basie <audio>, comme **poster**, **width**, et **height**
  - Les autres attributs sont identiques à ceux de la balise <audio>

```
<video id="playerVideo" width="480" height="270" controls autoplay>  
  <source src="media/speaker.ogv"  
    type='video/ogg; codec="theora, vorbis"'>  
  <source src="media/speaker.webm"  
    type='video/webm; codec="vp8, vorbis"'>  
</video>
```

# Prise en charge de la balise <video>



Source: <http://caniuse.com/#feat=video>

# Attributs supplémentaires de la balise <video>

Attribut	Description
<b>poster</b>	Permet d'afficher une image lorsqu'il n'y a pas de données vidéo disponibles. Si l'attribut poster n'est pas spécifié, le navigateur affichera la 1 <sup>e</sup> image de la vidéo. Doit contenir une URL valide vers l'image.
<b>width</b>	Définit la largeur en pixels de l'élément <video>
<b>height</b>	Définit la hauteur en pixels de l'élément <video>

```
<video controls poster="snowy-tree-poster.gif" width="300" height="210">  
  <source src="snowy-tree.mp4" type="video/mp4">  
  <source src="snowy-tree.webm" type="video/webm">  
</video>
```

# Exemples de contrôlés natifs



Firefox 5



Safari 5.0.5



Chrome 12

```
<video src="snowy-tree.mp4" width="300" height="176" controls></video>
```



Opera 11.50



IE9

# Exemples d'utilisation de <video>

## Exemples basiques

- Généralement, la lecture en boucle, la lecture auto et cacher les contrôles sont déconseillés
- Exemple 1: couper le son du fichier lors de son chargement
  - `<video src="match.webm" controls autoplay muted></video>`
- Exemple 2: empêcher le pré-chargement de la source audio
  - `<video src="match.webm" controls preload="none"></video>`
- Exemple 3: préciser des dimensions pour l'élément <video>
  - `<video src="snowy-tree.webm" controls width="300" height="210">`

# Exemples d'utilisation de <video>

## Lecture vidéo avec sources différentes

- ─ Problème: fournir une source vidéo dans plusieurs formats à la fois (MP4, OGG et WebM)
- ─ Solution: utiliser l'élément **<source>** en conjonction avec l'élément **<video>**

```
<video id="playerVideo" controls autoplay>  
  <source src="match.ogv" type="video/ogg">  
  <source src="match.mp4" type="video/mp4">  
  <source src="match.webm" type="video/webm">  
</video>
```

- ─ L'attribut **type** est déplacé sur l'élément **<source>** au lieu de **<video>**
  - **type** peut aussi contenir les informations relatives au codec utilisé pour encoder le fichier ou le flux vidéo

# Exemples d'utilisation de <video>

## Lecture audio avec support arrière

- ─ Problème: permettre la lecture de la source audio sur des navigateurs anciens (e.g. IE6 à 8) ou ne supportant pas HTML5
- ─ Solutions:
  - Soit fournir directement le lien vers la source audio, pour que l'utilisateur la télécharge

```
<audio controls>
  <source src="sayHello.ogg" type="audio/ogg">
  <source src="sayHello.mp3" type="audio/mp3">
  <object type="application/x-shockwave-flash"
    data="player.swf?audioUrl=sayHello.mp3&autoPlay=true"
    height="27" width="320">
    <param name="movie"
      value="player.swf?audioUrl=sayHello.mp3&autoPlay=true">
  </object>
  <a href="sayHello.mp3">Download the audio file</a>
</audio>
```

# Exemples d'utilisation de <video>

## Lecture vidéo avec support arrière (1)

- Fournir un lien vers le fichier vidéo pour le télécharger

```
<video controls autoplay>
  <source src="snowy-tree.mp4" type="video/mp4">
  <source src="snowy-tree.webm" type="video/webm">
  <figure>
    <a href="snowy-tree.mp4">
      
      <figcaption>Télécharger la vidéo: snowy-tree.mp4</figcaption>
    </a>
  </figure>
</video>
```

- Encapsuler la vidéo dans Flash, en utilisant une balise <object> ou <embed>

```
<video controls autoplay>
  <source src="snowy-tree.mp4" type="video/mp4">
  <object type="application/x-shockwave-flash"
    data="player.swf?videoUrl=snowy-tree.mp4&autoPlay=true"
    height="210" width="300">
    <param name="movie"
      value="player.swf?videoUrl=snowy-tree.mp4&autoPlay=true">
  </object></video>
```



# Vidéos réactives - 1

- Via HTML5, il est très facile d'intégrer une vidéo dans une page et la rendre réactive:
  - La balise `<video>` supporte les largeurs en %
  - Cela nécessite toutefois que la source de la vidéo soit hébergée sur votre site

```
1 <style>
2   video {
3     max-width: 100%;
4     height: auto;
5   }
6 </style>
7
8 <video width="320" height="240" controls="controls">
9   <source src="movie.mp4" type="video/mp4">
10  <source src="movie.ogv" type="video/ogg">
11   Votre navigateur ne prend pas en charge la balise video.
12 </video>
```

# Vidéos réactives - 2

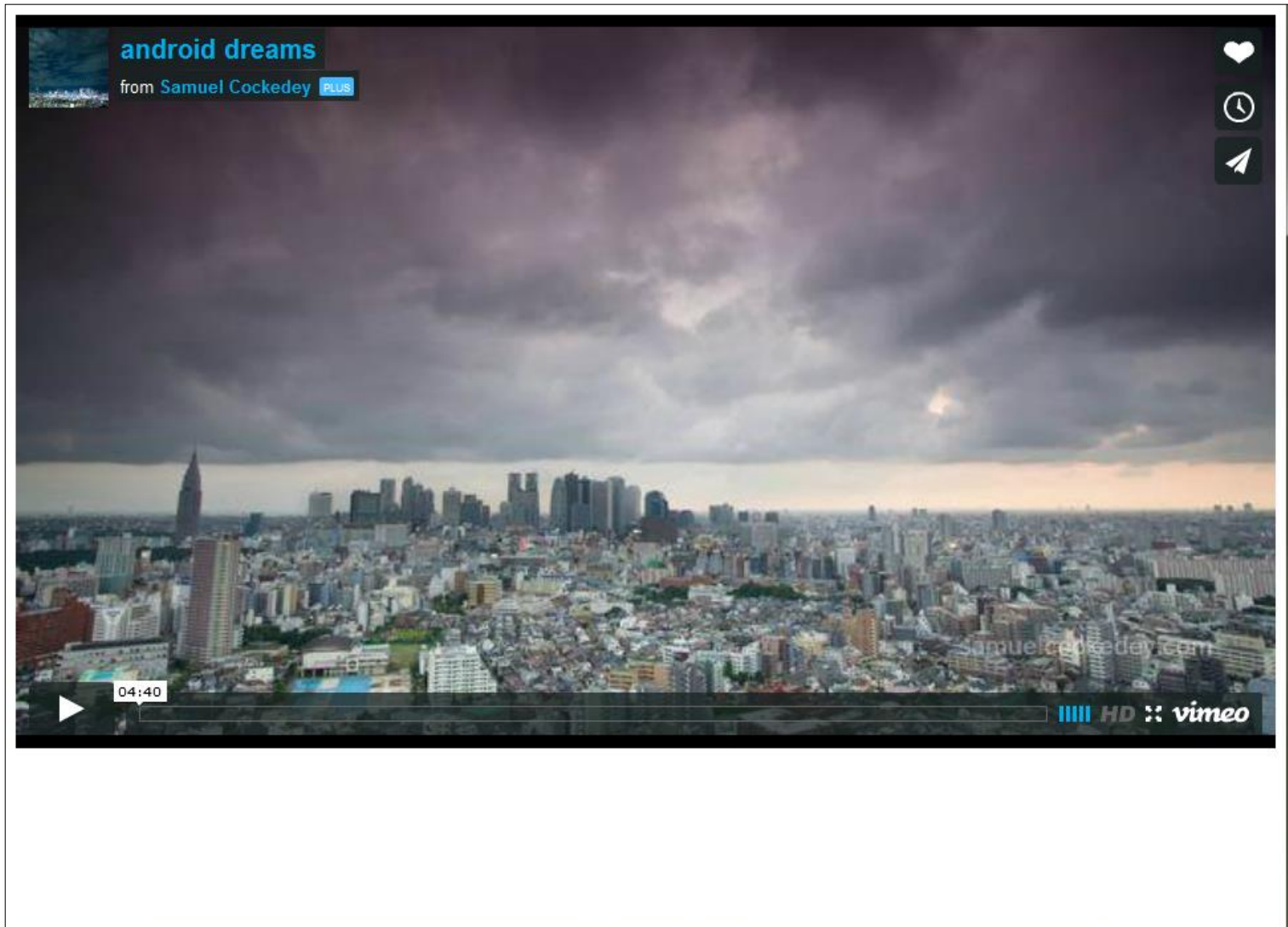
- Toutefois il est souvent plus intéressant d'utiliser un autre fournisseur, tel que YouTube ou Vimeo:
  - Permet de s'affranchir des contraintes de bande passante ou espace disque sur le serveur d'hébergement
  - De tels sites permettent de transformer des vidéos plus facilement
- Pour intégrer une telle vidéo, il est nécessaire d'utiliser une balise **<iframe>** ou **<object>**
  - Impossible dans une balise **<video>**
  - Pour rendre réactif le comportement de ces balises, c'est un peu plus difficile...

# Etude des fichiers HTML et CSS

```
1 <!DOCTYPE HTML>
2 <head>
3   <title>Vidéos fluides</title>
4   <meta charset="utf-8">
5   <link rel="stylesheet" type="text/css" href="style.css"/>
6 </head>
7 <body>
8   <div class="video-wrap">
9     <iframe src="//player.vimeo.com/video/30300114" width="500" height="281" frameborder="0"></iframe>
10  </div>
11 </body>
```

```
1 .video-wrap {
2   position: relative;
3   padding-bottom: 55%;
4   padding-top: 30px;
5   height: 0;
6   overflow: hidden;
7 }
8 .video-wrap iframe, .video-wrap object, .video-wrap embed {
9   position: absolute;
10  top: 0;
11  width: 100%;
12  height: 100%;
13 }
```

# Test 1



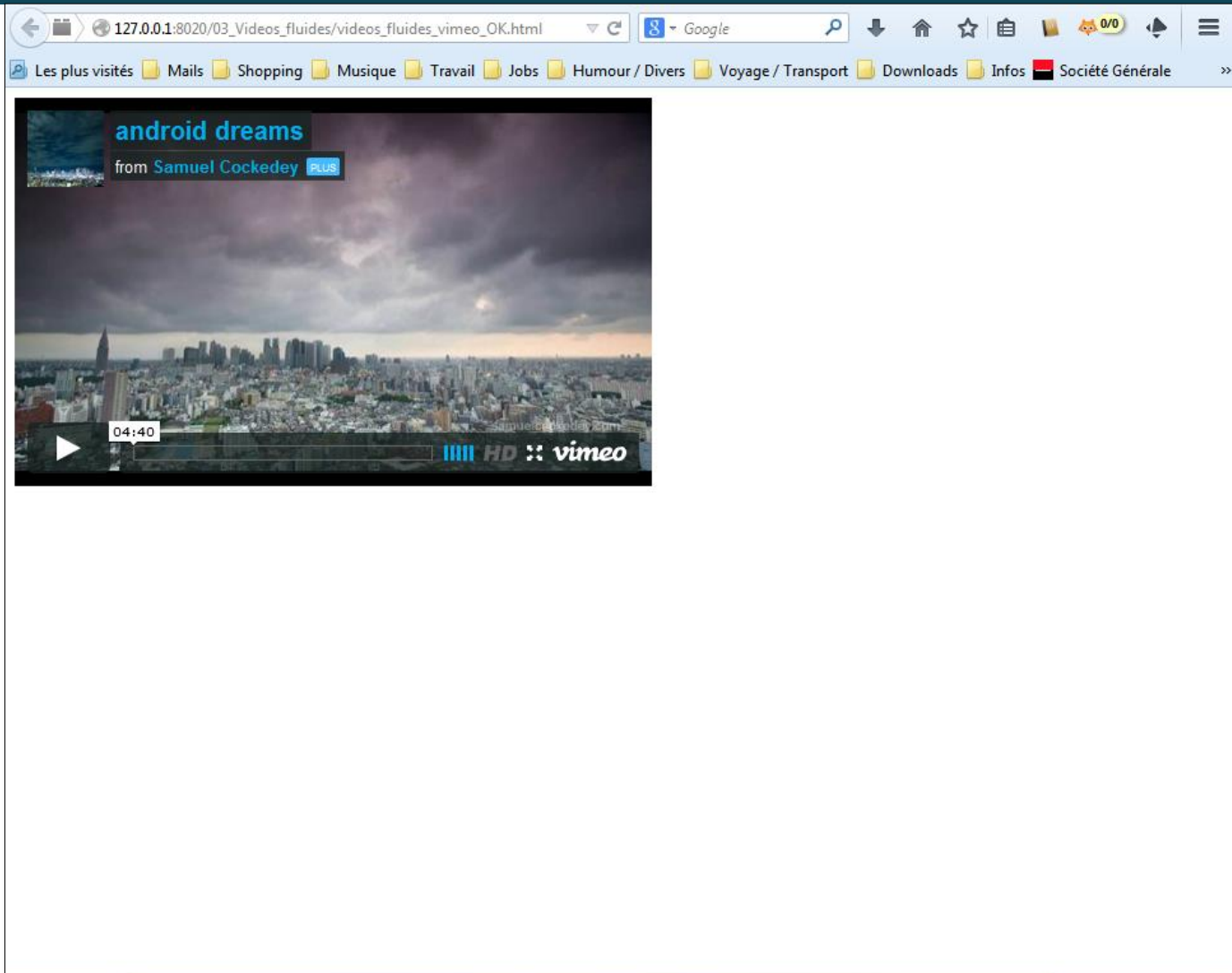
# Améliorations

- Si on ne souhaite pas que la vidéo prenne toute la largeur de la page
  - Placer l'élément video-wrap dans une autre div avec une largeur fixe et largeur max à 100%

```
1  <!DOCTYPE HTML>
2  <head>
3    <title>Vidéos fluides</title>
4    <meta charset="utf-8">
5    <link rel="stylesheet" type="text/css" href="style.css"/>
6  </head>
7  <body>
8    <div class="video-outer-wrap">
9      <div class="video-wrap">
10        <iframe src="//player.vimeo.com/video/30300114" width="500" height="281" frameborder="0"></iframe>
11      </div>
12    </div>
13  </body>
```

```
15  .video-outer-wrap {
16    width: 500px;
17    max-width: 100%;
18  }
```

# Test 2



# Remarques

- Cette méthode est appelée **Intrinsic Ratio for Videos**, et a été créée par **Thierry Koblentz** sur le blog A List Apart.
  - <http://alistapart.com/article/creating-intrinsic-ratios-for-video>
- Le principe est de placer la vidéo dans un conteneur ayant un ratio d'aspect intrinsèque, puis de donner à la vidéo une position absolue.
  - Ceci permet de verrouiller le ratio d'aspect, tout en permettant à la taille d'être fluide.

# 1.3 REDIMENSIONNER UNE IMAGE AVEC DES REQUÊTES MÉDIAS



# Exercice

Copiez ce fichier  
dans votre  
répertoire  
personnel.

- 03\_redim\_image\_req\_media.zip
  - HTML
  - CSS

# Principe

- Les requêtes média représentent une méthode très puissante et hautement personnalisable pour fournir des images réactives.
  - Elles sont utilisées ici différemment que lors d'une adaptation de la mise en page à une largeur d'écran
  - Elles s'appliquent dans les cas où la mise en page repose sur l'utilisation d'images avec des largeurs spécifiques prédéfinies pour les différentes tailles d'écrans visées
- La méthode nécessite une seule image
- Le redimensionnement est effectué côté client

# Mise en place

- Créez un fichier HTML contenant une balise image + associez-lui une classe CSS, par exemple « responsive »
  - C'est cette classe qui permettra un comportement différent selon les requêtes média
- Dans le fichier CSS associé, écrivez les requêtes média permettant d'identifier les écrans:
  - D'une largeur supérieure à 1024px
  - D'une largeur inférieure à 1024px
- Pour chacune des requêtes média, spécifiez une largeur différente pour la classe « **reponsive** »
- En dehors des requêtes média, spécifiez **height:auto** pour la classe « **responsive** ».

# Etude du code

## Fichier HTML

```
1 <!DOCTYPE HTML>
2 <head>
3     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4     <title>Images fluides - redimensionnement avec requêtes média</title>
5     <link rel="stylesheet" type="text/css" href="style.css" />
6 </head>
7 <body>
8
9     
10
11 </body>
```

## Fichier CSS

```
@media screen and (max-width: 1024px) {
    img.responsive {
        width: 200px;
    }
}

@media screen and (min-width: 1025px) {
    img.responsive {
        width: 300px;
    }
}

img.responsive {
    height: auto;
}
```

# Améliorations

## Fichier HTML

```
1 <!DOCTYPE HTML>
2 <head>
3   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4   <title>Images fluides - redimensionnement avec requêtes média</title>
5   <link rel="stylesheet" type="text/css" href="style.css" />
6 </head>
7 <body>
8
9   
10
11 </body>
```

## Fichier CSS

```
1 @media screen and (max-width: 1024px) {
2   img.responsive {
3     width: 200px;
4   }
5 }
6
7 @media screen and (min-width: 1025px) and (max-width: 1280px) {
8   img.responsive {
9     width: 300px;
10  }
11 }
12
13 @media screen and (min-width: 1081px) {
14   img.responsive {
15     width: 400px;
16   }
17 }
18
19 img.responsive {
20   height: auto;
21 }
```

# Fonctionnement

- Grâce aux requêtes média CSS3, vous pouvez utiliser des conditions logiques CSS basées sur les propriétés du viewport du navigateur.
  - Il est ainsi possible d'afficher différents styles selon les propriétés de la fenêtre du navigateur.
- Dans cette méthode, nous utilisons ces propriétés pour définir des tailles d'images différentes.

# 1.4 NAVIGATION ADAPTATIVE AVEC DES REQUÊTES MÉDIA

# Exercice

Copiez ce fichier  
dans votre  
répertoire  
personnel.

- 04\_req\_media\_nav.zip
  - HTML
  - CSS



# Principe

- Les requêtes média sont utiles à bien plus de choses que le simple redimensionnement d'images
- Ici, nous allons les utiliser afin d'afficher un menu réactif sur la base de différentes tailles d'écrans
  - A partir de deux menus différents, nous allons afficher un menu dynamique pour trois tailles d'écrans différentes

# Mise en place

## HTML

- 1 div par menu (x2)
- 1 div pour le contenu

## CSS

- Requêtes média
- Classes par défaut pour menus (x2) et contenu

# Mise en place

## Fichier HTML - 1

- Pour les terminaux avec des tailles de fenêtres réduites (e.g. terminaux mobiles, tablettes):
  - Menu simple select, occupant un espace vertical limité
  - Menu reposant sur une balise `<form>` pour les options de navigation, qui déclenche un code JavaScript pour charger la page correspondant à la sélection

```
<form>
  <select name="URL" onchange="window.location.href=this.form.URL.options[this.form.URL.selectedIndex].value">
    <option value="op_1.html">Option 1</option>
    <option value="op_2.html">Option 2</option>
    <option value="op_3.html">Option 3</option>
  </select>
</form>
```

- Menu à intégrer dans un div à laquelle on associe une classe CSS

# Mise en place

## Fichier HTML - 2

- Pour les tailles de fenêtres plus importantes :
  - Menu reposant sur une balise `<ul>` pour les options de navigation, qui sera stylisé via du CSS
    - Le menu aura une mise en forme et un aspect différent grâce aux requêtes média

```
<ul>
  <li>
    <a href="op_1.html">Option 1</a>
  </li>
  <li>
    <a href="op_2.html">Option 2</a>
  </li>
  <li>
    <a href="op_3.html">Option 3</a>
  </li>
</ul>
```

- Menu à intégrer dans une `<div>` à laquelle on associera une classe CSS

# Mise en place

## Fichier CSS

### — Requêtes média à créer:

- Pour fenêtres inférieures à 800px – afficher seulement la div contenant le formulaire `<select>` (menu réduit)
- Pour toutes les autres fenêtres – afficher seulement la div contenant le menu liste `<ul>`

```
@media screen and ( max-width: 800px ) {  
    .petit-menu { display: inline; }  
    .grand-menu { display: none; }  
}  
  
@media screen and ( min-width: 801px ) and ( max-width:  
1024px ) {  
    .petit-menu { display: none; }  
    .grand-menu { display: inline; }  
}  
  
@media screen and ( min-width: 1025px ) {  
    .petit-menu { display: none; }  
    .grand-menu { display: inline; }  
}
```

## Menus écrans de taille moyenne

- Afficher les éléments sous forme de liste horizontale, en haut de la page, avec une largeur à 100%
- Ajouter les lignes suivantes dans la requête média correspondante:
  - `.grand-menu ul { list-style-type: none; }`
  - `.grand-menu ul li { display: inline; }`
  - Préciser une largeur à 100% pour la classe `.grand-menu`

```
.grand-menu {  
  display: inline;  
  width: 100%;  
}
```

## Menus écrans de grande taille

- Afficher les éléments sous forme de liste verticale, flottant à gauche de son élément parent, avec une largeur à 20%
- Ajouter les lignes suivantes dans la requête média correspondante:

```
.grand-menu {  
    display: inline;  
    float: left;  
    width: 20%;  
}
```

# Améliorations

## Mise en forme du contenu

- Ecrans de taille moyenne
  - Taille de l'élément contenant le paragraphe à 100%
- Ecrans de grande taille
  - Taille de l'élément contenant le paragraphe à 80%
  - Flottement à droite de l'élément parent
- Ajouter les lignes suivantes dans les requêtes média correspondantes:
  - `.contenu: { width: 100%; }`
  - `.contenu{`  
`float: right;`  
`width: 80%;`  
`}`



Option 1 ▾

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean neceros magna, non fringilla lacus. Uthendrerit, nisl id sagittis laoreet, turpis arcu posuere ipsum, non dictum tortor nisi sed lorem. Nam sitamethendrerit lacus. In rutrum, mauris laoreet bibendum sodales, augue augue fermentum nunc, at feugiat neque turpis ut purus. Praesent mollis urna nisl. Aliquam posuere lacus at elit tempus sed aliquam augue venenatis. Nullam pretium nisl vitae vestibulum est. Nullam pulvinar pellentesque vel it neque condimentum.

[Option 1](#) [Option 2](#) [Option 3](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean neceros magna, non fringilla lacus. Uthendrerit, nisl id sagittis laoreet, turpis arcu posuere ipsum, non dictum tortor nisi sed lorem. Nam sitamethendrerit lacus. In rutrum, mauris laoreet bibendum sodales, augue augue fermentum nunc, at feugiat neque turpis ut purus. Praesent mollis urna nisl. Aliquam posuere lacus at elit tempus sed aliquam augue venenatis. Nullam pretium nisl sed lorem consequat ut sollicitudin felis gravida. Cras vitae vestibulum est. Nullam pulvinar pellentesque vel it neque condimentum.

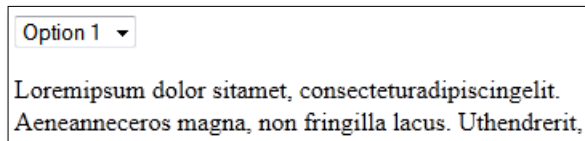
- [Option 1](#)
- [Option 2](#)
- [Option 3](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean neceros magna, non fringilla lacus. Uthendrerit, nisl id sagittis laoreet, turpis arcu posuere ipsum, non dictum tortor nisi sed lorem. Nam sitamethendrerit lacus. In rutrum, mauris laoreet bibendum sodales, augue augue fermentum nunc, at feugiat neque turpis ut purus. Praesent mollis urna nisl. Aliquam posuere lacus at elit tempus sed aliquam augue venenatis. Nullam pretium nisl sed lorem consequat ut sollicitudin felis gravida. Cras vitae vestibulum est. Nullam pulvinar pellentesque vel it neque condimentum.

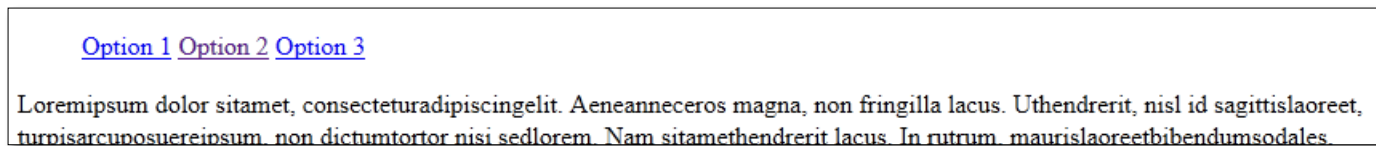
# Fonctionnement

- Chaque version de la navigation exploite les requêtes média CSS3 pour maximiser l'espace disponible pour l'affichage du menu et du contenu associé.

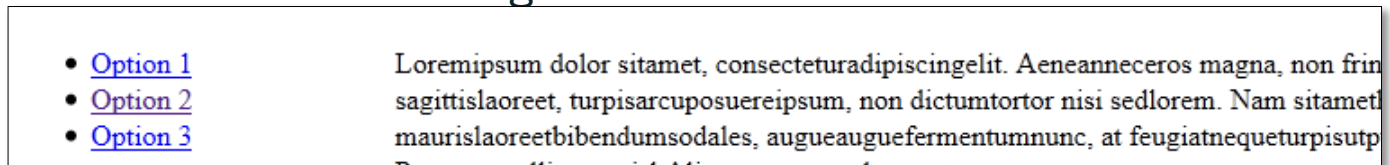
- Pour les fenêtres de tailles < 1024px, les éléments de navigation sont contenus dans un élément `<select>` d'une balise `<form>`



- Pour les fenêtres de taille moyenne (1025px à 1280px), les éléments de navigation sont affichés horizontalement, et occupent toute la largeur de la page; ils sont suivis par le contenu de la page



- Pour les fenêtres de tailles supérieures (> 1280px), les éléments du menu flottent à gauche de la page et occupent 20% de sa largeur; le contenu s'étend sur 80% de la largeur et flotte à droite



# 2. MISE EN PAGE RÉACTIVE

Utilisation des propriétés min-width et max-width  
Contrôler la mise en page avec un padding relatif  
Requêtes média pour chaque terminal  
Créer une mise en page réactive avec des requêtes média  
Cacher un élément avec les requêtes média

## 2.1 UTILISATION DES PROPRIÉTÉS MIN- WIDTH ET MAX-WIDTH

# Exercice

Copiez ce fichier  
dans votre  
répertoire  
personnel.

## — 05\_Min\_max.zip

- HTML
- CSS

# Principe

- La plupart des techniques de mise en page réactive peuvent être assez complexes et compliquées
- Nous verrons ici une méthode simple, reposant sur l'utilisation des propriétés **min-width** et **max-width** appliquées à 3 éléments flottants
  - Autre chose que changer le nombre de colonnes pour les viewport réduits
- Cette méthode permet d'avoir un affichage compatible avec la plupart des terminaux mobiles et écrans de bureau

# Mise en place

## Fichier HTML

- ─ Créez une page simple, contenue dans un seul élément `<article>` ayant:
  - Une en-tête `<h1>`
  - Trois `<div>` chacune avec sa propre classe CSS
    - La première `<div>` contiendra une image
    - Les deux autres du texte

```
1  <!DOCTYPE HTML>
2  <head>
3      <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4      <title>Utilisation des propriétés min-width et max-width</title>
5      <link rel="stylesheet" type="text/css" href="06_min_max.css" />
6  </head>
7  <article>
8      <h1>Utilisation des propriétés min-width et max-width</h1>
9      <div class="un">
10         
11     </div>
12     <div class="deux">
13         Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec a diam lectus. Sed
14     </div>
15     <div class="trois">
16         Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos
17     </div>
18 </article>
```

# Mise en place

## Fichier CSS

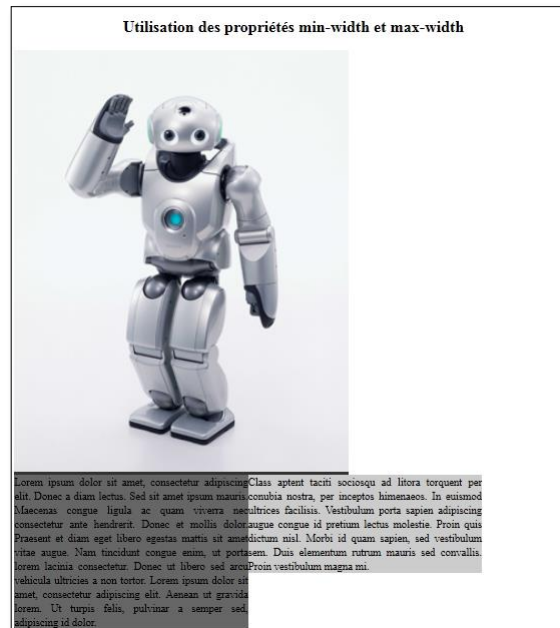
- Une classe pour **<article>**
- Une classe pour **<h1>**
- Une classe pour chacune des 3 **<div>**

```
1 article {  
2     width: 100%;  
3     max-width: 1280px;  
4     margin: 0 auto;  
5 }  
6 h1 {  
7     text-align: center;  
8 }  
9 img {  
10    width: 100%;  
11    height: auto;  
12 }  
13 .un {  
14     background-color: #333;  
15     min-width: 500px;  
16 }  
17 .deux {  
18     background-color: #666  
19 }  
20 .trois {  
21     background-color: #ccc  
22 }
```



# Exercice

- Ecrivez le code CSS de la classe float
  - Précise la largeur maxi pour un affichage sur une seule colonne
  - Définit le flottement
  - Définit l'alignement du texte
- Appliquez cette classe aux éléments HTML concernés
- Résultat attendu :



## 2.2 CONTRÔLER LA MISE EN PAGE AVEC UN PADDING RELATIF

# Exercice

Copiez ce fichier  
dans votre  
répertoire  
personnel.

## — 06\_Padding.zip

- HTML
- CSS

# Principe

- Nous souhaitons créer une mise en page réactive pour un site de type blog
  - Avec des commentaires et des réponses aux commentaires
- Nous allons utiliser exclusivement les marges relatives (relative padding)

# Mise en place

## Fichier HTML

```
<div class="contenu">
  <header>Contrôler la mise en page avec un padding relatif</header>
  <p>Curabitur ...</p>
  <aside>
    <h1>Commentaires</h1>
    <div class="commentaire">
      <p>Pellentesque ...</p>
      <div class="commentaire">
        <p>Pellentesque ...</p>
        <div class="commentaire">
          <p>Pellentesque ...</p>
        </div>
      </div>
    </div>
    <div class="commentaire">
      <p>Pellentesque ...</p>
    </div>
    <div class="commentaire">
      <div class="commentaire">
        <p>Pellentesque ...</p>
        <p>Pellentesque ...</p>
      </div>
    </div>
  </aside>
</div>
```

Contrôler la mise en page avec un padding relatif
Curabitur at mauris nec sapien malesuada sollicitudin. Nunc scelerisque dignissim augue aliquam interdum. Donec id aliquam augue. Proin commodo volutpat velit, vitae vulputate ipsum luctus vel. Aenean tincidunt iaculis tellus, sed egestas eros tincidunt et.
<b>Commentaires</b>
Pellentesque eleifend facilisis odio ac ullamcorper. Nullam ut enim ut massa tincidunt luctus. Ut nulla libero, eleifend vel ultrices at, volutpat quis quam. In eleifend velit bibendum sapien consectetur fermentum iaculis eu eros. Nullam eget dui eros, et semper justo. Nunc ut condimentum felis. Donec elementum euismod nisl vitae commodo. Morbi ac orci purus. Phasellus vehicula vulputate metus a fermentum. Fusce justo massa, pretium sed sollicitudin sit amet, matta vitae ligula. Praesent venenatis viverra leo et convallis. Curabitur at mauris nec sapien malesuada sollicitudin. Nunc scelerisque dignissim augue aliquam interdum. Donec id aliquam augue. Proin commodo volutpat velit, vitae vulputate ipsum luctus vel. Aenean tincidunt iaculis tellus, sed egestas eros tincidunt et.
Pellentesque eleifend facilisis odio ac ullamcorper. Nullam ut enim ut massa tincidunt luctus. Ut nulla libero, eleifend vel ultrices at, volutpat quis quam. In eleifend velit bibendum sapien consectetur fermentum iaculis eu eros. Nullam eget dui eros, et semper justo. Nunc ut condimentum felis. Donec elementum euismod nisl vitae commodo. Morbi ac orci purus. Phasellus vehicula vulputate metus a fermentum. Fusce justo massa, pretium sed sollicitudin sit amet, matta vitae ligula. Praesent venenatis viverra leo et convallis. Curabitur at mauris nec sapien malesuada sollicitudin. Nunc scelerisque dignissim augue aliquam interdum. Donec id aliquam augue. Proin commodo volutpat velit, vitae vulputate ipsum luctus vel. Aenean tincidunt iaculis tellus, sed egestas eros tincidunt et.
Pellentesque eleifend facilisis odio ac ullamcorper. Nullam ut enim ut massa tincidunt luctus. Ut nulla libero, eleifend vel ultrices at, volutpat quis quam. In eleifend velit bibendum sapien consectetur fermentum iaculis eu eros. Nullam eget dui eros, et semper justo. Nunc ut condimentum felis. Donec elementum euismod nisl vitae commodo. Morbi ac orci purus. Phasellus vehicula vulputate metus a fermentum. Fusce justo massa, pretium sed sollicitudin sit amet, matta vitae ligula. Praesent venenatis viverra leo et convallis. Curabitur at mauris nec sapien malesuada sollicitudin. Nunc scelerisque dignissim augue aliquam interdum. Donec id aliquam augue. Proin commodo volutpat velit, vitae vulputate ipsum luctus vel. Aenean tincidunt iaculis tellus, sed egestas eros tincidunt et.
Pellentesque eleifend facilisis odio ac ullamcorper. Nullam ut enim ut massa tincidunt luctus. Ut nulla libero, eleifend vel ultrices at, volutpat quis quam. In eleifend velit bibendum sapien consectetur fermentum iaculis eu eros. Nullam eget dui eros, et semper justo. Nunc ut condimentum felis. Donec elementum euismod nisl vitae commodo. Morbi ac orci purus. Phasellus vehicula vulputate metus a fermentum. Fusce justo massa, pretium sed sollicitudin sit amet, matta vitae ligula. Praesent venenatis viverra leo et convallis. Curabitur at mauris nec sapien malesuada sollicitudin. Nunc scelerisque dignissim augue aliquam interdum. Donec id aliquam augue. Proin commodo volutpat velit, vitae vulputate ipsum luctus vel. Aenean tincidunt iaculis tellus, sed egestas eros tincidunt et.

# Mise en place

## Fichier CSS

```
.contenu {  
    ...  
}  
  
aside {  
    ...  
}  
  
.commentaire {  
    ...  
    background-color: #ccc;  
    border: 1px solid #333;  
}
```

# Exercice

- Ajoutez des marges relatives pour les différentes classes :
  - Une marge gauche (5) et une marge droite (5) pour la classe `.contenu`
  - Une marge gauche (20) et une marge droite (10) pour la classe `aside`
  - Une marge gauche (10) pour la classe `.commentaire`

```
1 ① .contenu {  
2      padding: 0 5% 0 5%;  
3  }  
4 ② aside {  
5      padding: 0 10% 0 30%;  
6  }  
7 ③ .commentaire {  
8      padding: 8px 3px 8px 10%;  
9      background-color: #ccc;  
10     border: 1px solid #333;  
11 }
```

# Fonctionnement

Contrôler la mise en page avec un padding relatif

Curabitur at mauris nec sapien malesuada sollicitudin. Nunc scelerisque dignissim augue aliquam interdum. Donec id aliquam augue. Proin commodo volutpat velit, vitae vulputate ipsum luctus vel. Aenean tincidunt iaculis tellus, sed egestas eros tincidunt et.

## Commentaires

Pellentesque eleifend facilisis odio ac ullamcorper. Nullam ut enim ut massa tincidunt luctus. Ut nulla libero, eleifend vel ultrices at, volutpat quis quam. In eleifend velit bibendum sapien consectetur fermentum iaculis eu eros. Nullam eget dui eros, et semper justo. Nunc ut condimentum felis. Donec elementum euismod nisl vitae commodo. Morbi ac orci purus. Phasellus vehicula vulputate metus a fermentum. Fusce justo massa, pretium sed sollicitudin sit amet, mattis vitae ligula. Praesent venenatis viverra leo et convallis. Curabitur at mauris nec sapien malesuada sollicitudin. Nunc scelerisque dignissim augue aliquam interdum. Donec id aliquam augue. Proin commodo volutpat velit, vitae vulputate ipsum luctus vel. Aenean tincidunt iaculis tellus, sed egestas eros tincidunt et.

Pellentesque eleifend facilisis odio ac ullamcorper. Nullam ut enim ut massa tincidunt luctus. Ut nulla libero, eleifend vel ultrices at, volutpat quis quam. In eleifend velit bibendum sapien consectetur fermentum iaculis eu eros. Nullam eget dui eros, et semper justo. Nunc ut condimentum felis. Donec elementum euismod nisl vitae commodo. Morbi ac orci purus. Phasellus vehicula vulputate metus a fermentum. Fusce justo massa, pretium sed sollicitudin sit amet, mattis vitae ligula. Praesent venenatis viverra leo et convallis. Curabitur at mauris nec sapien malesuada sollicitudin. Nunc scelerisque dignissim augue aliquam interdum. Donec id

Contrôler la mise en page avec un padding relatif

Curabitur at mauris nec sapien malesuada sollicitudin. Nunc scelerisque dignissim augue aliquam interdum. Donec id aliquam augue. Proin commodo volutpat velit, vitae vulputate ipsum luctus vel. Aenean tincidunt iaculis tellus, sed egestas eros tincidunt et.

## Commentaires

Pellentesque eleifend facilisis odio ac ullamcorper. Nullam ut enim ut massa tincidunt luctus. Ut nulla libero, eleifend vel ultrices at, volutpat quis quam. In eleifend velit bibendum sapien consectetur fermentum iaculis eu eros. Nullam eget dui eros, et semper justo. Nunc ut condimentum felis. Donec elementum euismod nisl vitae commodo. Morbi ac orci purus. Phasellus vehicula vulputate metus a fermentum. Fusce justo massa, pretium sed sollicitudin sit amet, mattis vitae ligula. Praesent venenatis viverra leo et convallis. Curabitur at mauris nec sapien malesuada sollicitudin. Nunc scelerisque dignissim augue aliquam interdum. Donec id aliquam augue. Proin commodo volutpat velit, vitae vulputate ipsum luctus vel. Aenean tincidunt iaculis tellus, sed egestas eros tincidunt et.

Pellentesque eleifend facilisis odio ac ullamcorper. Nullam ut enim ut massa tincidunt luctus. Ut nulla libero, eleifend vel ultrices at, volutpat quis quam. In eleifend velit bibendum sapien consectetur fermentum iaculis eu eros. Nullam eget dui eros, et semper justo. Nunc ut condimentum felis. Donec elementum euismod nisl vitae commodo. Morbi ac orci purus. Phasellus vehicula vulputate metus a fermentum. Fusce justo massa, pretium sed sollicitudin sit amet, mattis vitae ligula. Praesent venenatis viverra leo et convallis. Curabitur at mauris nec sapien malesuada sollicitudin. Nunc scelerisque dignissim augue aliquam interdum. Donec id aliquam augue. Proin commodo volutpat velit, vitae vulputate ipsum luctus vel. Aenean tincidunt iaculis tellus, sed egestas eros tincidunt et.

Pellentesque eleifend facilisis odio ac ullamcorper. Nullam ut enim ut massa tincidunt luctus. Ut nulla libero, eleifend vel ultrices at, volutpat quis quam. In eleifend velit bibendum sapien consectetur fermentum iaculis eu eros. Nullam eget dui eros, et semper justo. Nunc ut condimentum felis. Donec elementum euismod nisl vitae commodo. Morbi ac orci purus. Phasellus vehicula vulputate metus a fermentum. Fusce justo massa, pretium sed sollicitudin sit amet, mattis vitae ligula. Praesent venenatis viverra leo et convallis. Curabitur at mauris nec sapien malesuada sollicitudin. Nunc scelerisque dignissim augue aliquam interdum. Donec id aliquam augue. Proin commodo volutpat velit, vitae vulputate ipsum luctus vel. Aenean tincidunt iaculis tellus, sed egestas eros tincidunt et.

Pellentesque eleifend facilisis odio ac ullamcorper. Nullam ut enim ut massa tincidunt luctus. Ut nulla libero, eleifend vel ultrices at, volutpat quis quam. In eleifend velit bibendum sapien consectetur fermentum iaculis eu eros. Nullam eget dui eros, et semper justo. Nunc ut condimentum felis. Donec elementum euismod nisl vitae commodo. Morbi ac orci purus. Phasellus vehicula vulputate metus a fermentum. Fusce justo massa, pretium sed sollicitudin sit amet, mattis vitae ligula. Praesent venenatis viverra leo et convallis. Curabitur at mauris nec sapien malesuada sollicitudin. Nunc scelerisque dignissim augue aliquam interdum. Donec id aliquam augue. Proin commodo volutpat velit, vitae vulputate ipsum luctus vel. Aenean tincidunt iaculis tellus, sed egestas



## 5.3 REQUÊTES MÉDIA POUR CHAQUE TERMINAL

# Exercice

Copiez ce fichier  
dans votre  
répertoire  
personnel.

- 07 Tous terminaux.zip
  - HTML
  - CSS

# Principe

- Nous allons écrire les requêtes média pour tous les terminaux et les positions (portrait / paysage)
  - Pas tous... mais une grande partie !
  - Téléviseur HD, smartphone, ancien téléphone, imprimante, etc.
- Pour les terminaux non pris en compte dans cet exercice:
  - <http://www.w3.org/TR/css3-mediaqueries/>

# Mise en place

## Fichier HTML

### — Créez une page HTML simple

- Titre **<h1>**
- **<div>** contenant:
  - Une image
  - Du texte

```
1  <!DOCTYPE HTML>
2  <html>
3    <head>
4      <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5      <title>Requêtes média pour chaque terminal</title>
6      <link rel="stylesheet" type="text/css" href="08_tous_terminaux.css" />
7    </head>
8    <body>
9      <h1>Requêtes média pour chaque terminal</h1>
10     <div>
11       
12       Suspendisse lectus leo, consectetur in tempor sit amet, placerat quis
13     </div>
14   </body>
15 </html>
```

# Mise en place

## Fichier CSS

```
@media print {...}
```

```
@media (orientation: portrait) {...}
```

```
@media (orientation: landscape) {...}
```

```
@media (height:500px) {...}
```

```
@media (device-width:500px) {...}
```

```
@media (device-height:500px) {...}
```

```
@media screen and (device-aspect-ratio: 16/9) {...}
```

```
@media tv {...}
```

```
@media screen and (max-width:960px) {...}
```

```
@media screen and (min-width:961px) and (max-width:1280px) {...}
```

```
@media screen and (min-width:1281px) {...}
```

# Exercice

- Spécifiez des styles différents pour chacune des requêtes précédentes, pour les éléments:
  - `body`
  - `h1`
  - `img`
  - `p`
- Testez le fonctionnement

## Requêtes média pour chaque terminal

Suspendisse lectus leo, consectetur in tempor sit amet, placerat quis neque.

### Requêtes média pour chaque terminal

Suspendisse lectus leo, consectetur in tempor sit amet, placerat quis neque.

Etiam luctus porttitor lorem, sed suscipit est rutrum non. Curabitur lobortis nisl a enim congue semper. Aenean commodo ultrices imperdiet. Quisque mi metus, ornare sit amet fermentum et, tincidunt et orci. Fusce eget orci a orci congue vestibulum. Ut dolor diam, elementum et vestibulum eu, porttitor vel elit. Curabitur venenatis pulvinar tellus gravida ornare. Sed et erat faucibus nunc euismod ultricies ut id justo. Nullam cursus suscipit nisi, et ultrices justo sodales nec. Fusce venenatis facilisis lectus ac semper. Aliquam at massa ipsum.



tor lorem, sed suscipit est rutrum non. Curabitur lobortis nisl  
per. Aenean commodo ultrices imperdiet. Quisque mi metus,  
mentum et, tincidunt et orci. Fusce eget orci a orci congue  
lor diam, elementum et vestibulum eu, porttitor vel elit.  
is pulvinar tellus gravida ornare. Sed et erat faucibus nunc  
ut id justo. Nullam cursus suscipit nisi, et ultrices justo  
e venenatis facilisis lectus ac semper. Aliquam at massa

