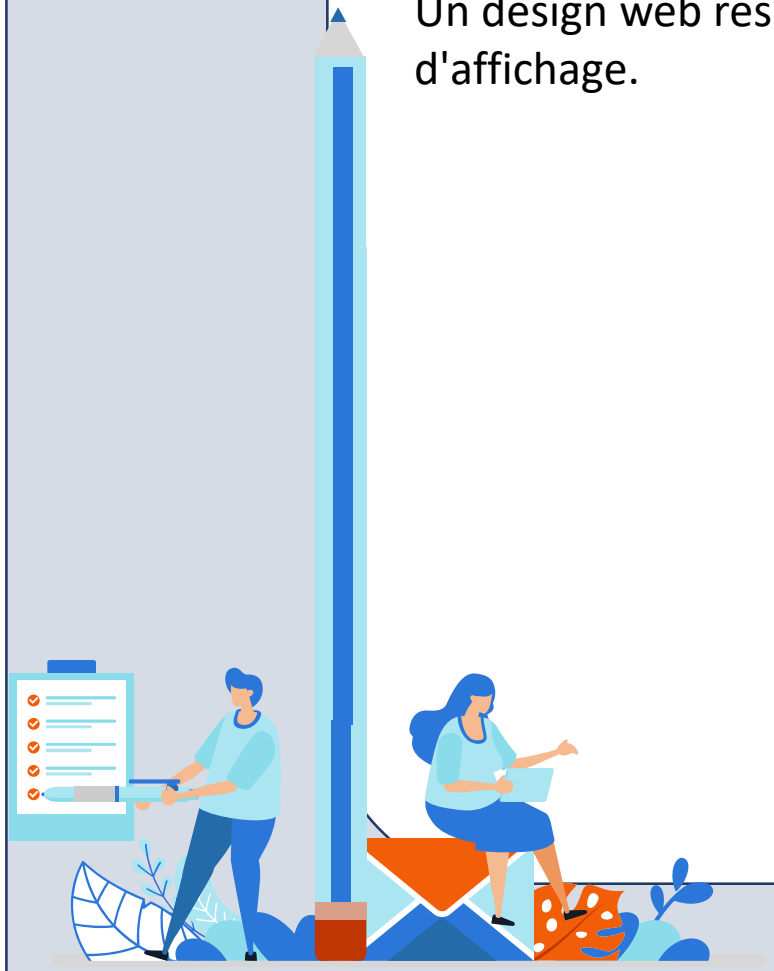
Pratique: integration web

Techniques Responsive

Dr Riadh HARIZI

Le **Responsive Web Design** consiste à créer des pages web qui s'adaptent bien à tous les appareils ! Un design web responsive s'ajustera automatiquement aux différentes tailles d'écran et fenêtres d'affichage.

Comment ?



HTML :Structurer le Web

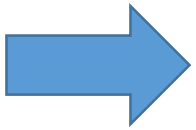
Multimédia et Intégration

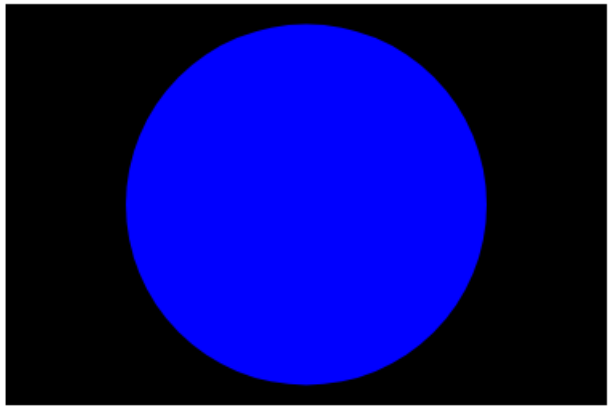
Les images en HTML

```
<figure>
  <img
    src="images/dinosaur.jpg"
    alt="The head and torso of a dinosaur skeleton;
         it has a large head with long sharp teeth"
    width="400"
    height="341" />

  <figcaption>
    A T-Rex on display in the Manchester University Museum.
  </figcaption>
</figure>
```

crée un cercle et un rectangle :

```
<svg
  version="1.1"
  baseProfile="full"
  width="300"
  height="200"
  xmlns="http://www.w3.org/2000/svg">
  <rect width="100%" height="100%" fill="black" />
  <circle cx="150" cy="100" r="90" fill="blue" />
</svg>
```





```
<img
  src="equilateral.svg"
  alt="un triangle avec trois côtés égaux"
  height="87"
  width="100" />
```

Contenu vidéo

```
<video
  controls
  width="400"
  height="400"
  autoplay
  loop
  muted
  poster="poster.png">
  <source src="rabbit320.mp4" type="video/mp4" />
  <source src="rabbit320.webm" type="video/webm" />
  <p>
    Votre navigateur ne prend pas en charge les vidéos HTML5. Voici, à la place,
    un <a href="rabbit320.mp4">lien à la vidéo</a>.
  </p>
</video>
```

Contenu audio

```
<audio controls>
  <source src="viper.mp3" type="audio/mp3" />
  <source src="viper.ogg" type="audio/ogg" />
  <p>
    Votre navigateur ne prend pas en charge l'audio HTML5. Voici, à la place, un
    <a href="viper.mp3">lien sur l'audio</a>.
  </p>
</audio>
```

Des objets aux iframes — autres techniques d'intégration

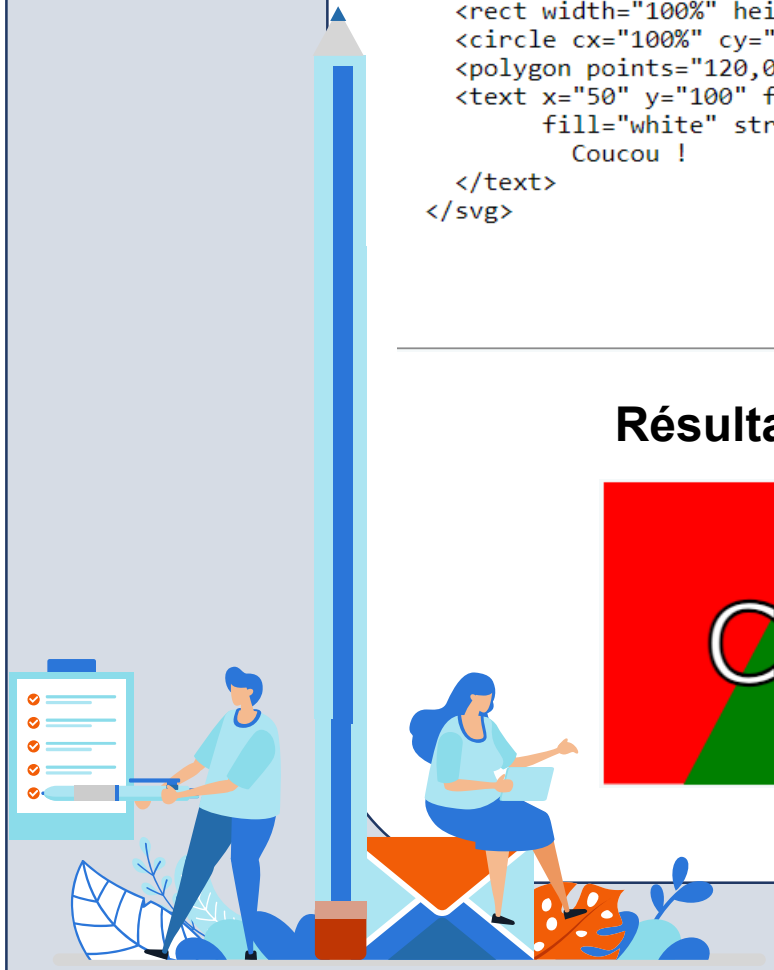
```
<iframe
  src="https://developer.mozilla.org/fr/docs/Glossary"
  width="100%"
  height="500"
  frameborder="0"
  allowfullscreen
  sandbox>
  <p>
    <a href="https://developer.mozilla.org/fr/docs/Glossary">
      Lien de repli pour les navigateurs ne prenant pas en charge iframe
    </a>
  </p>
</iframe>
```

```
<object
  data="mypdf.pdf"
  type="application/pdf"
  width="800"
  height="1200"
  typemustmatch>
  <p>
    Vous ne possédez pas de greffon PDF, mais vous pouvez
    <a href="myfile.pdf">télécharger le fichier PDF.</a>
  </p>
</object>
```

Exemple de Code SVG

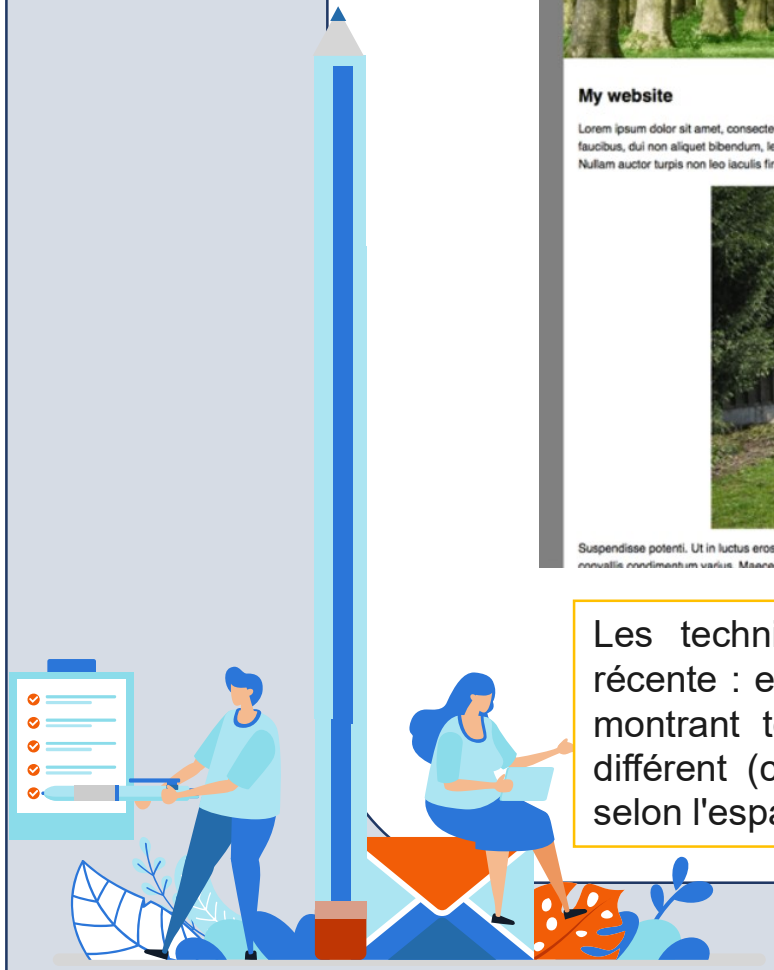
```
<svg width="100%" height="100%">
  <rect width="100%" height="100%" fill="red" />
  <circle cx="100%" cy="100%" r="150" fill="blue" stroke="black" />
  <polygon points="120,0 240,225 0,225" fill="green"/>
  <text x="50" y="100" font-family="Verdana" font-size="55"
        fill="white" stroke="black" stroke-width="2">
          Coucou !
  </text>
</svg>
```

Résultat



Images adaptatives ?

Les techniques d'images adaptatives sont de mise en œuvre récente : elles offrent au navigateur plusieurs fichiers d'images, soit montrant tous la même chose mais avec un nombre de pixels différent (commutation de résolution), soit des images différentes selon l'espace alloué (décisions artistiques).



Comment créer des images adaptatives ?

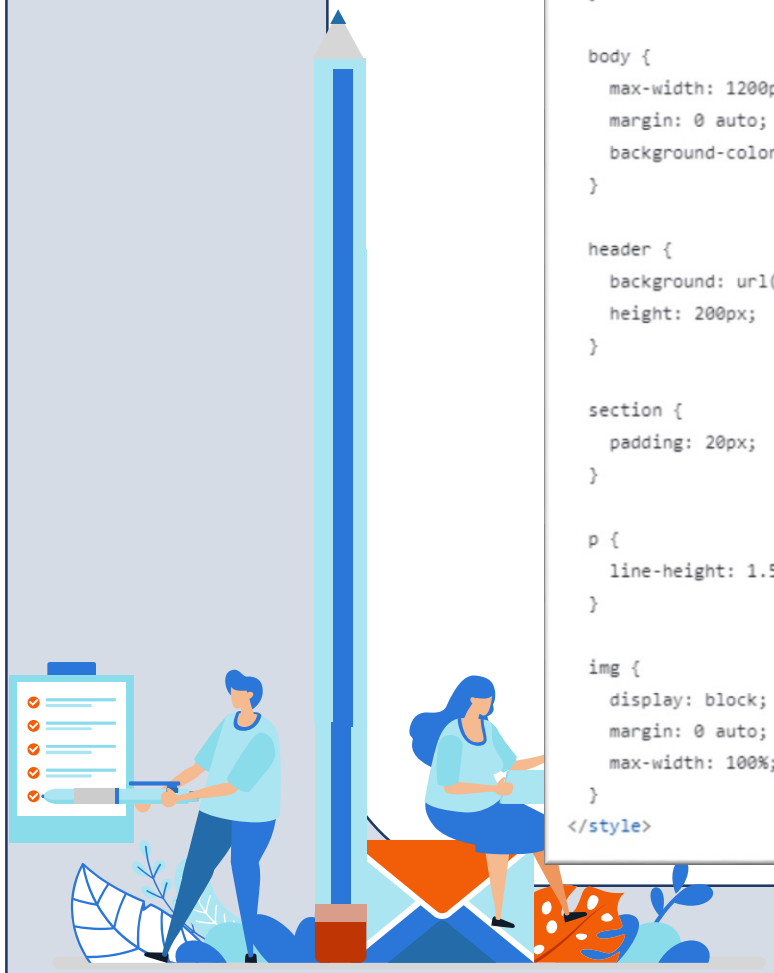
```
<img src="elva-fairy-800w.jpg" alt="Elva habillée en fée" />
```

Mais il est possible d'utiliser deux nouveaux attributs — srcset et sizes — permettant de fournir plusieurs images source avec des indications pour permettre au navigateur de faire le bon choix.

```
<img
  srcset="elva-fairy-480w.jpg 480w, elva-fairy-800w.jpg 800w"
  sizes="(max-width: 600px) 480px,
            800px"
  src="elva-fairy-800w.jpg"
  alt="Elva dressed as a fairy" />
```

srcset définit l'ensemble des images offertes au choix du navigateur, et la taille de chaque image. Avant chaque virgule, nous avons écrit :
1.un nom de **fichier image** (elva-fairy-480w.jpg),
2.un espace,
3.la **largeur intrinsèque en pixels** (480w) — notez l'utilisation de l'unité w, et non px comme vous auriez pu penser. C'est la taille réelle de l'image; qui peut être trouvée en examinant les propriétés du fichier image sur l'ordinateur

sizes définit un ensemble de conditions pour le média (par ex. des largeurs d'écran) et indique quelle taille d'image serait la plus adaptée si certaines conditions sont satisfaites — ce sont les conditions dont nous avons parlé plus haut. Dans ce cas, nous écrivons avant chaque virgule :
1.une **condition pour le média** ((max-width:600px)) — cette condition pour le média décrit un état possible de l'écran. Dans notre cas, nous disons « si la largeur de fenêtre est de 600 pixels ou moins »,
2.une espace,
3.**la largeur de la place** occupée par l'image si la condition pour le média est vraie (480px).



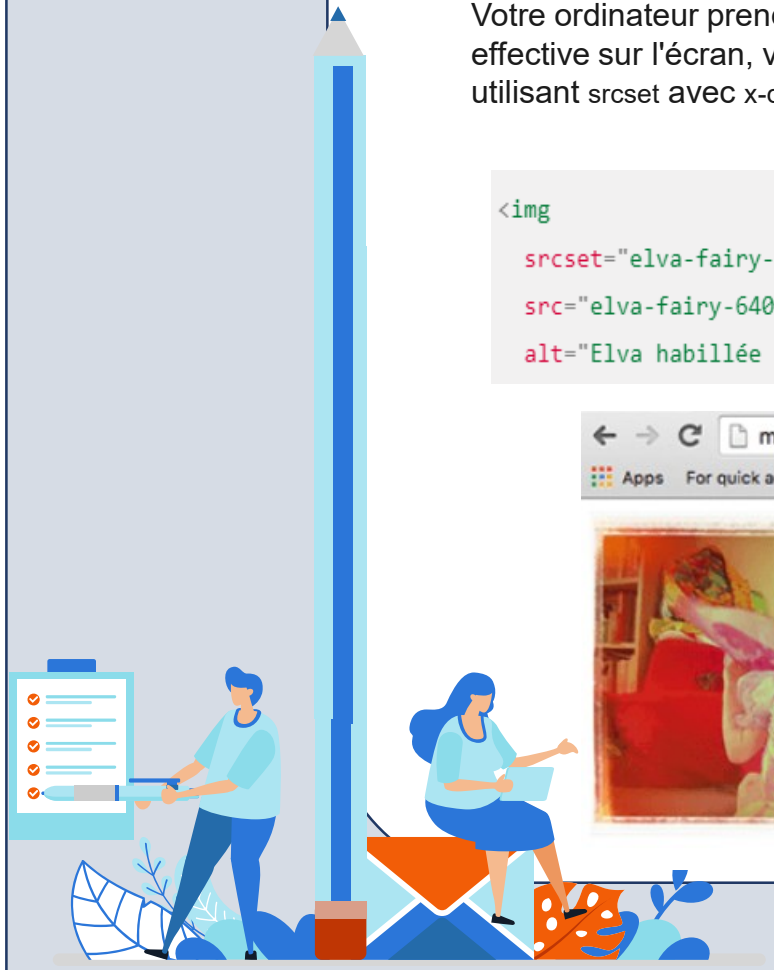
```css
<style>
  html {
    font-family: sans-serif;
    background-color: gray;
  }

  body {
    max-width: 1200px;
    margin: 0 auto;
    background-color: white;
  }

  header {
    background: url(header.jpg) no-repeat center;
    height: 200px;
  }

  section {
    padding: 20px;
  }

  p {
    line-height: 1.5;
  }

  img {
    display: block;
    margin: 0 auto;
    max-width: 100%;
  }
</style>
```

```html
<body>
  <header>

  </header>

  <main>
    <section>
      <h1>My website</h1>

      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris eget venenatis ligula. Ut lacinia at dolor

      <picture>
        <source media="(max-width: 799px)" srcset="elva-480w-close-portrait.jpg">
        <source media="(min-width: 800px)" srcset="elva-800w.jpg">
        <img src="elva-800w.jpg" alt="Chris standing up holding his daughter Elva">
      </picture>

      <p>Suspendisse potenti. Ut in luctus eros. Mauris pulvinar vehicula aliquet. Etiam imperdiet eleifend luctus.

      <img srcset="elva-fairy-480w.jpg 480w,
                   elva-fairy-800w.jpg 800w"
           sizes="(max-width: 600px) 480px,
                  800px"
           src="elva-fairy-800w.jpg" alt="Elva dressed as a fairy">

      <p>Header image originally by <a href="https://www.flickr.com/photos/miwok/17086751527/">Miwok</a>.</p>
    </section>
  </main>
</body>
</html>
```

Commutation de résolution : même taille, résolutions différentes

Votre ordinateur prend en charge plusieurs résolutions d'affichage, mais que tout le monde voit l'image avec la même taille effective sur l'écran, vous pouvez permettre au navigateur de choisir une image de résolution appropriée en utilisant srcset avec x-descriptors et sans sizes — une syntaxe un peu plus facile en quelque sorte !

```
<img
  srcset="elva-fairy-320w.jpg, elva-fairy-480w.jpg 1.5x, elva-fairy-640w.jpg 2x"
  src="elva-fairy-640w.jpg"
  alt="Elva habillée en fée" />
```

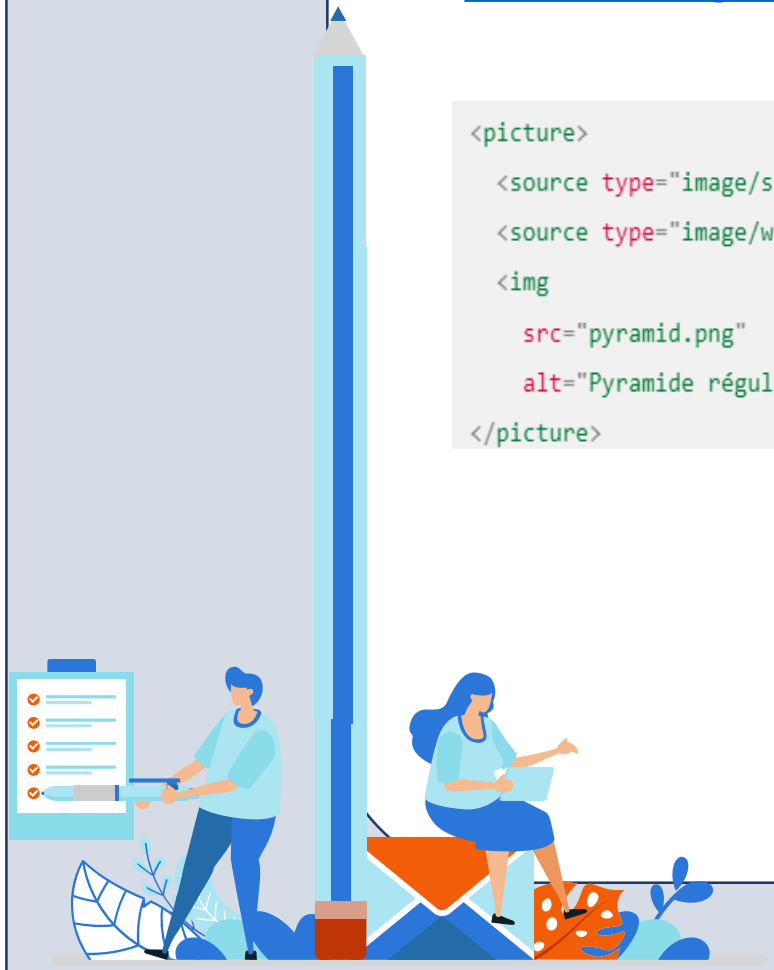


Pourquoi **ne peut-on pas réaliser cela avec le CSS ou du JavaScript ?**

```
<!doctype html>
<html lang="en-US">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <title>Responsive HTML images demo</title>
    <style>
      img {
        width: 320px;
      }
    </style>
  </head>
  <body>
    <img
      srcset="
        elva-fairy-320w.jpg,
        elva-fairy-480w.jpg 1.5x,
        elva-fairy-640w.jpg 2x
      "
      src="elva-fairy-640w.jpg"
      alt="Elva dressed as a fairy"
      id="img"
      onload="showUrl()"
    />
    <script>
      function showUrl() {
        const img = document.getElementById("img");
        const p = document.createElement("p");
        const fileName = img.currentSrc?.substring(
          img.currentSrc.lastIndexOf("/"),
        );
        p.textContent = `Loaded: ${fileName}`;
        document.body.appendChild(p);
      }
    </script>
  </body>
</html>
```

Utilisez largement les formats d'image modernes

```
<picture>
  <source type="image/svg+xml" srcset="pyramid.svg" />
  <source type="image/webp" srcset="pyramid.webp" />
  <img
    src="pyramid.png"
    alt="Pyramide régulière constituée de quatre triangles équilatéraux" />
</picture>
```



Police de caractères adaptative avec CSS et les media queries

C'est grâce aux ***media queries*** (requêtes média) en CSS3 que vous pouvez adapter votre police d'écriture aux différentes tailles d'écran. Les media queries sont des règles à appliquer pour **reconnaître les différentes caractéristiques d'un écran**, comme sa **résolution**, la **largeur** et la **hauteur** de la fenêtre du navigateur, ou si l'écran est en position **verticale ou horizontale**.
Une police de caractères adaptative doit pouvoir être **flexible** et c'est pourquoi les unités en ***em*** (cadratin) et en ***rem*** (root em) doivent déterminer l'unité de mesure de la police d'écriture.

Cette technique permet de modifier la taille du texte de base à l'aide de **pourcentages**, qui sont une autre unité de mesure permettant également de déterminer la taille des caractères.

Commandes CSS.

•Avec ***@font-face***, vous pouvez intégrer la **police souhaitée**.
•Pour que la police d'écriture puisse **s'adapter au format de l'écran** sur lequel votre site est affiché, la requête média ***@media*** est nécessaire. En combinant *@media* et ***screen*** ou ***min-width*** (largeur minimale) ou ***max-width*** (largeur maximale),

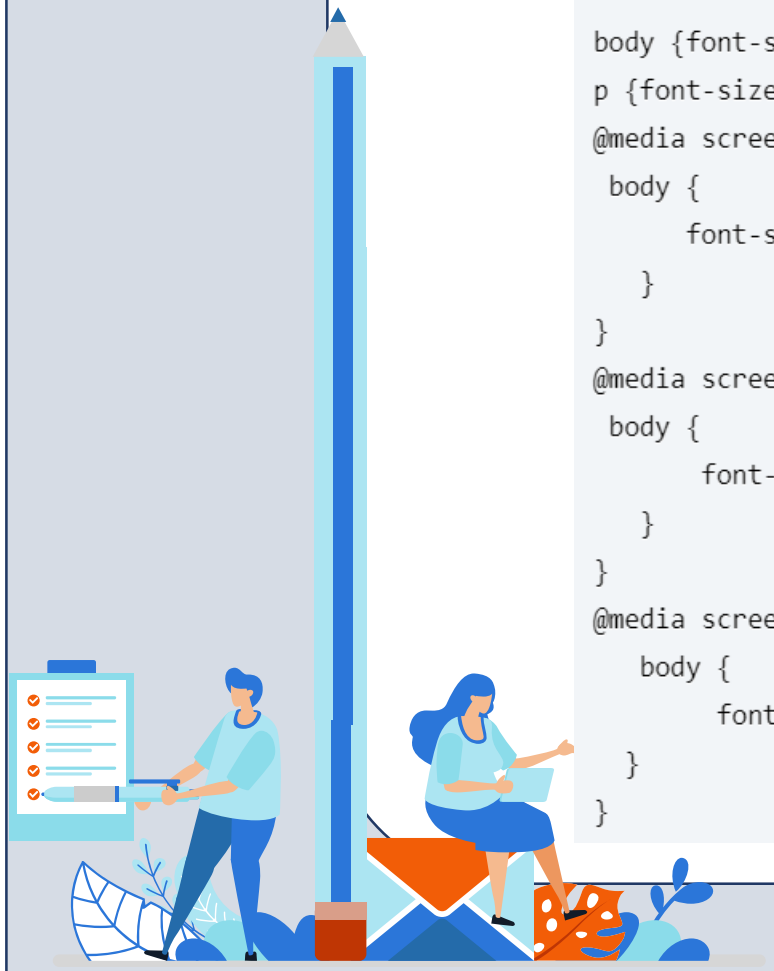
•il est possible de déterminer le **format de la police de caractères** en fonction de **différentes tailles d'écran**. Par exemple avec : ***@media screen*** *and* ***(min-width: 800px)***.

•La propriété ***font-size*** définit la taille affichée de la police d'écriture, en pixels ou en unités de mesure comme ***em***. Cette unité de mesure se rapporte à la taille de la police et permet d'avoir des feuilles de style plus facilement adaptables d'un média à l'autre.

• Si la taille de la police de caractères est définie en *em*, alors elle s'adapte en fonction de la taille de l'élément parent.

Un corps de texte adaptatif en *em*

```
body {font-size: 100%}
p {font-size: 1.5em;}
@media screen and (max-width: 64em) {
 body {
      font-size: 90%;
   }
}
@media screen and (max-width: 50em) {
 body {
       font-size: 75%;
   }
}
@media screen and (max-width: 30em) {
   body {
        font-size: 50%;
  }
}
```



La valeur du corps de texte (*body*) est fixée à 100% et est de l'ordre de 16 pixels sur la plupart des navigateurs.
La **taille de base des polices de caractères** du texte (*font-size: 1.5em*) est de **24 pixels** (1,5 x 16 = 24) dans cet exemple.
A partir des commandes *font-size* (*90%*, *75%*, *50%*) en rapport avec le *body*, la taille de polices de caractères s'adapte à la largeur de la fenêtre.
Cette courte manipulation permet de rendre votre police de caractères adaptative pour des projets en Responsive Web design.

```
body {font-size: 100%}
p {
    font-size: 1.5em;
    line-hight: 1.3 em;
}
h1, h2, h3 {line-height: 1.2em;}
```

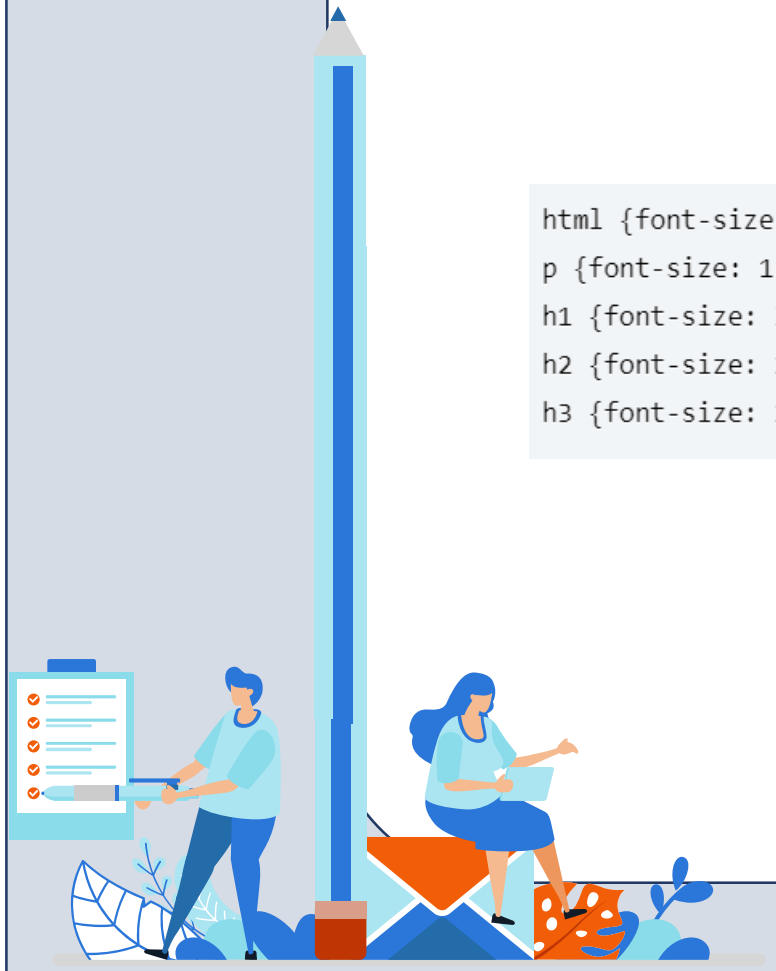
lors que le corps de texte est adaptatif, il va de soi qu'il est important de veiller à ce que les **interlignes** le soient aussi. Une des règles de base en typographie consiste à équilibrer l'espacement des interlignes en fonction de la longueur des lignes du texte.

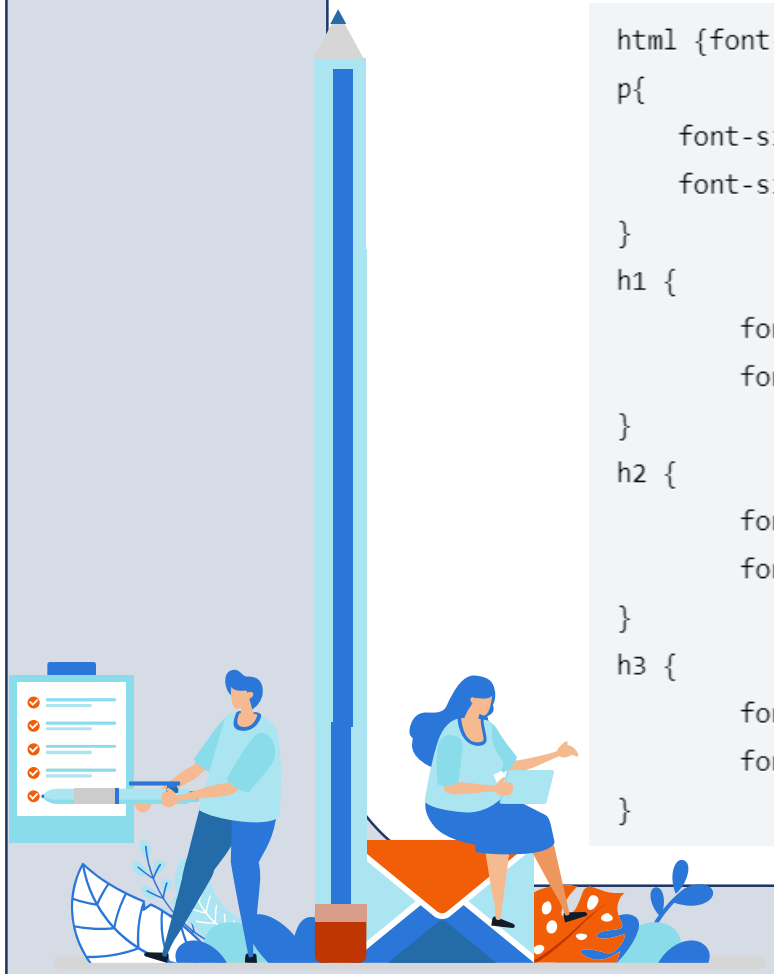
la commande ***line-height*** (hauteur des lignes)

Un corps de texte adaptatif avec l'unité *rem*

```
html {font-size: 100%;}
p {font-size: 1.5rem;}
h1 {font-size: 3rem;}
h2 {font-size: 2.5 rem;}
h3 {font-size: 2rem;}
```

L'unité relative de mesure ***rem*** (« **root em** ») est également adaptée aux corps de textes adaptatifs. Cette unité est **orientée en fonction de l'élément racine *html*** (au regard des unités comme les *em*). Par conséquent, la taille de la police d'écriture est directement liée au contenu racine.





```
html {font-size: 100%}
p{
    font-size: 24px;
    font-size: 1.5rem;
}
h1 {
        font-size: 48px;
        font-size: 3rem;
}
h2 {
        font-size: 40px;
        font-size: 2.5rem;
}
h3 {
        font-size: 32px;
        font-size: 2rem;
}
```

Lors d'une problématique de compatibilité de l'unité *rem* avec les anciennes versions de navigateurs. Cette unité est néanmoins compatible avec Firefox, Chrome, Safari, Edge ou encore Opera. Pour Internet Explorer, une assistance des unités *rem* n'est seulement disponible qu'à partir de la **Version 9**. Pour que les utilisateurs des anciennes versions d'Internet Explorer puissent tout de même bénéficier d'une navigation agréable sur votre site Web, il est possible d'intégrer un **fallback**, qui est une solution de repli.

Exemple: paramétrer la taille des titres grâce aux media queries

```
body {font-size: 100%}
h1 {font-size: 3em;}
@media screen and (max-width: 64em) {
h1 {
        font-size: 2.5em;
    }
}
@media screen and (max-width: 50em) {
h1 {
        font-size: 2em;
    }
}
@media screen and (max-width: 30em){
h1 {
        font-sitze: 1.5em;
    }
}
```

Cet exemple appliqué au *h1* permet de donner quatre variantes du ***font-size*** (taille de la police de caractères) du ***h1*** (*3em*; *2.5em*; *2em*; *1.5em*).
Vous pouvez déterminer la taille de la police de caractères en fonction de la largeur du navigateur qui l'affichera.
Pour cela, déterminez la taille maximale (***max-width***). Dans cet exemple illustratif, la largeur est déterminée par l'unité de mesure *em*. Sa valeur est définit en fonction de la taille standard du navigateur, généralement de l'ordre de 16 pixels. En suivant cette logique, notre exemple donne ***1em = 16px***. Ainsi, vous pouvez calculer les différentes tailles d'*em* en pixels en fonction des largeurs des fenêtres d'écran (min-width/max-width).
Calculez la largeur de la fenêtre en pixels et **divisez** (ou multipliez) la donnée tout simplement **par 16**.
Dans l'exemple, la taille du h1 se base sur quatre tailles de la fenêtre différentes :

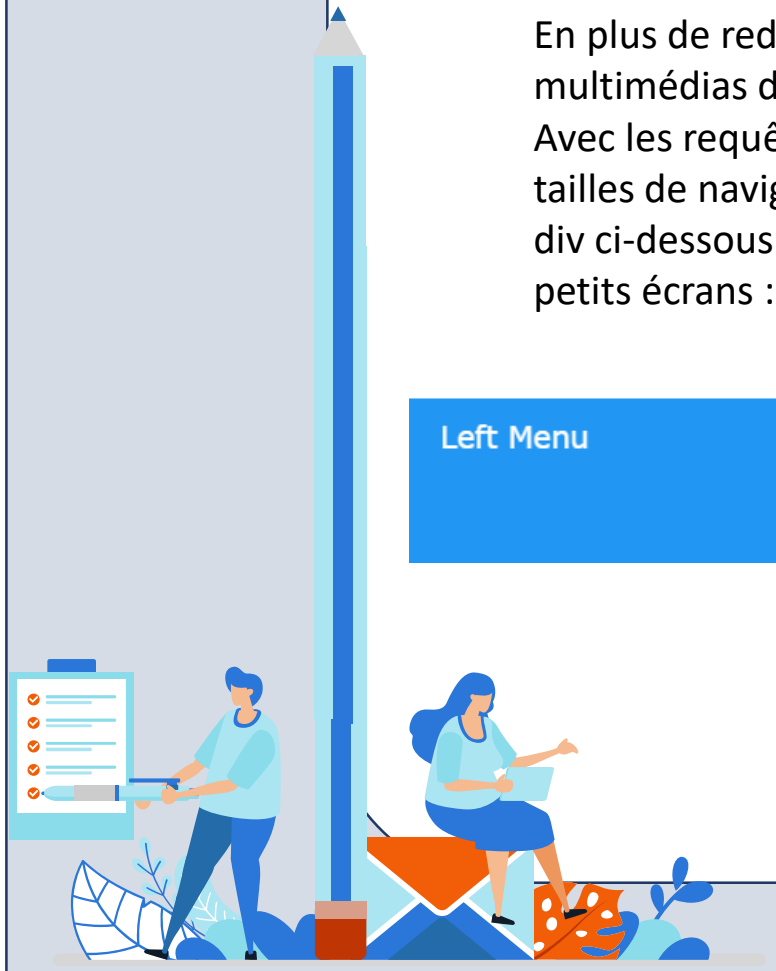
- plus de 1024 pixels
- jusqu'à pixels (1024 : 16 = *64em*)
- jusqu'à 800 pixels (800 : 16 = *50em*)
- jusqu'à 480 pixels (480 : 16 = *30em*)

vous pouvez modifier le code et l'appliquer au cas par cas (**breakpoints**). Les breakpoints seraient par exemeple en 480, 800 ou 1024 pixels. Ainsi, il vous sera possible d'adapter au plus près vos titres.

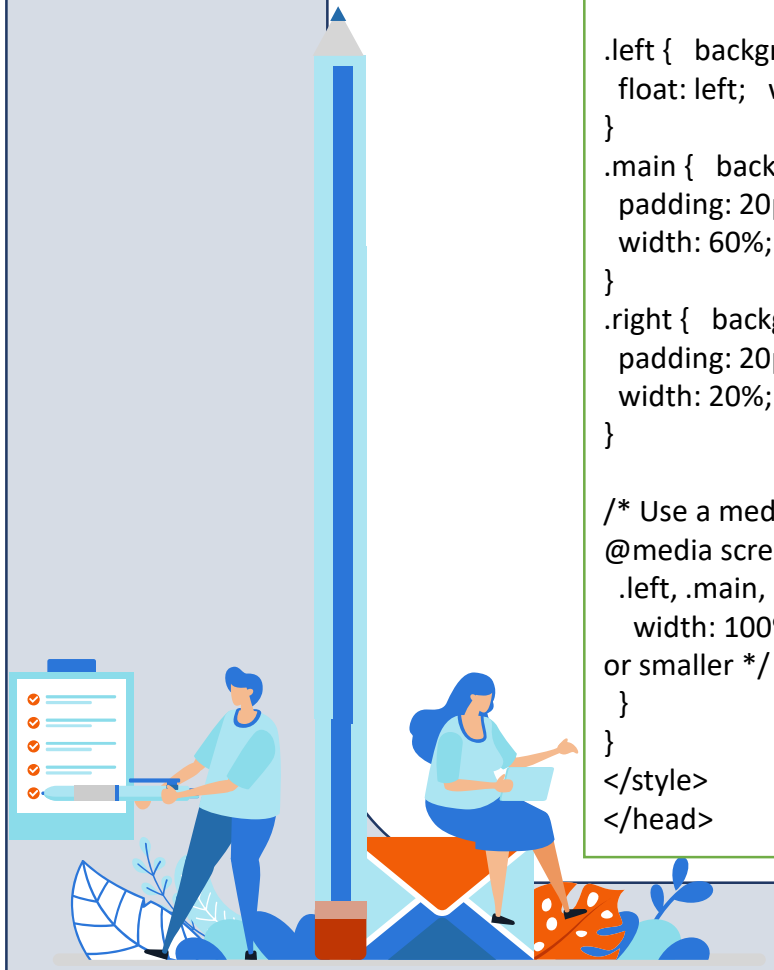
Cas pratique: Media Queries

En plus de redimensionner le texte et les images, il est également courant d'utiliser des requêtes multimédias dans les pages Web réactives.
Avec les requêtes multimédias, vous pouvez définir des styles complètement différents pour différentes tailles de navigateur.Exemple : redimensionnez la fenêtre du navigateur pour voir que les trois éléments div ci-dessous s'afficheront horizontalement sur les grands écrans et s'empileront verticalement sur les petits écrans :

Left Menu	Main Content	Right Content

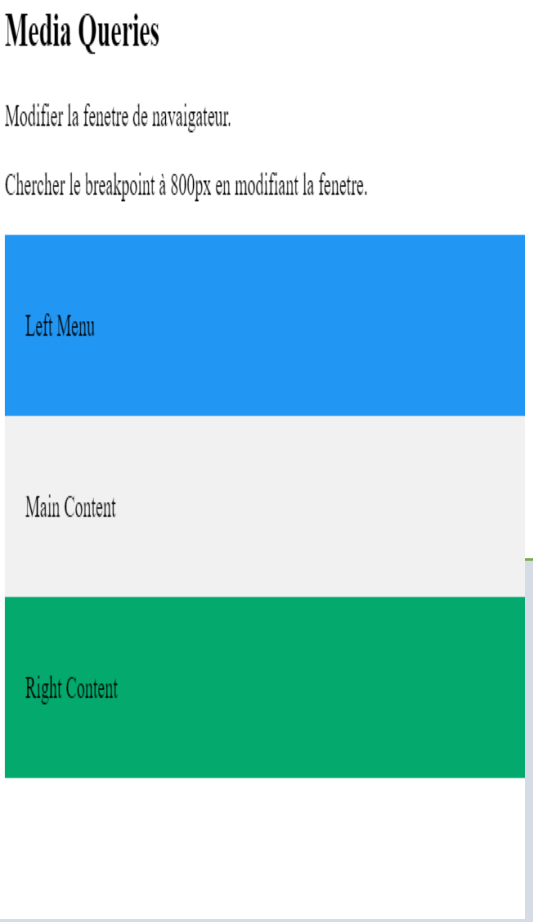


```css
.left, .right {
  float: left;
  width: 20%; /* The width is 20%, by default */
}

.main {
  float: left;
  width: 60%; /* The width is 60%, by default */
}

/* Use a media query to add a breakpoint at 800px: */
@media screen and (max-width: 800px) {
  .left, .main, .right {
    width: 100%; /* The width is 100%, when the viewport is 800px or smaller */
  }
}
```



```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
* {  box-sizing: border-box; }

.left {  background-color: #2196F3;  padding: 20px;
 float: left;  width: 20%; /* The width is 20%, by default */
}
.main {  background-color: #f1f1f1;
 padding: 20px;  float: left;
 width: 60%; /* The width is 60%, by default */
}
.right {  background-color: #04AA6D;
 padding: 20px;  float: left;
 width: 20%; /* The width is 20%, by default */
}

/* Use a media query to add a break point at 800px: */
@media screen and (max-width: 800px) {
 .left, .main, .right {
  width: 100%; /* The width is 100%, when the viewport is 800px or smaller */
 }
}
</style>
</head>
```

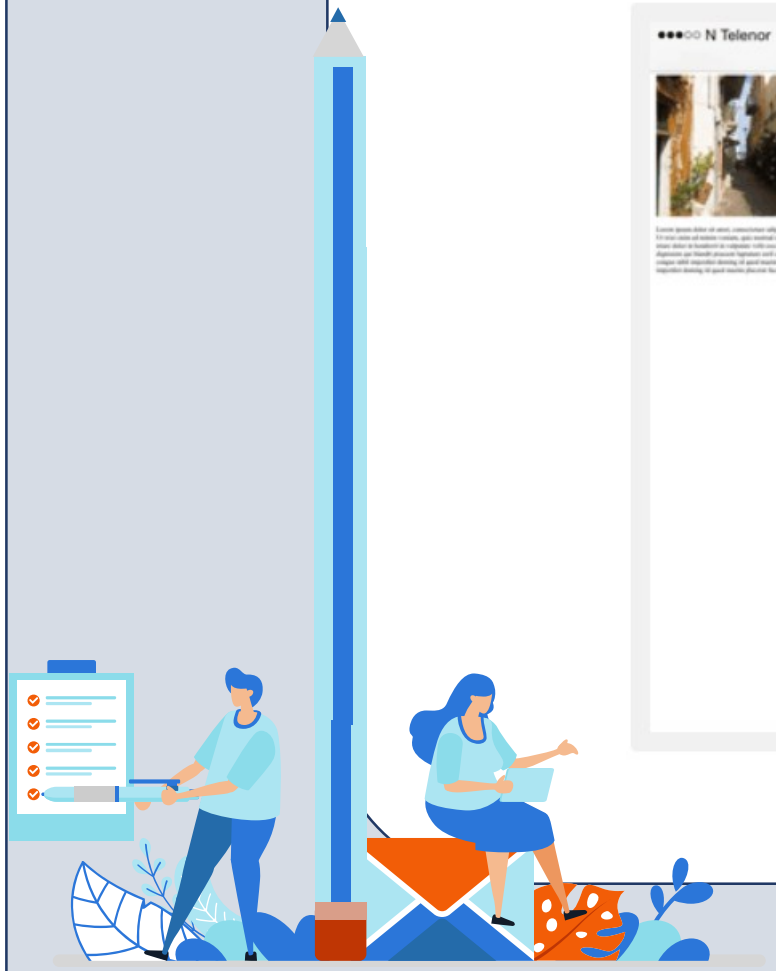
```
<body>

<h2>Media Queries</h2>
<p>Modifier la fenetre de navaigateur.</p>

<p>Chercher le breakpoint à 800px en modifiant la fenetre.</p>

<div class="left">
 <p>Left Menu</p>
</div>

<div class="main">
 <p>Main Content</p>
</div>

<div class="right">
 <p>Right Content</p>
</div>

</body>
</html>
```



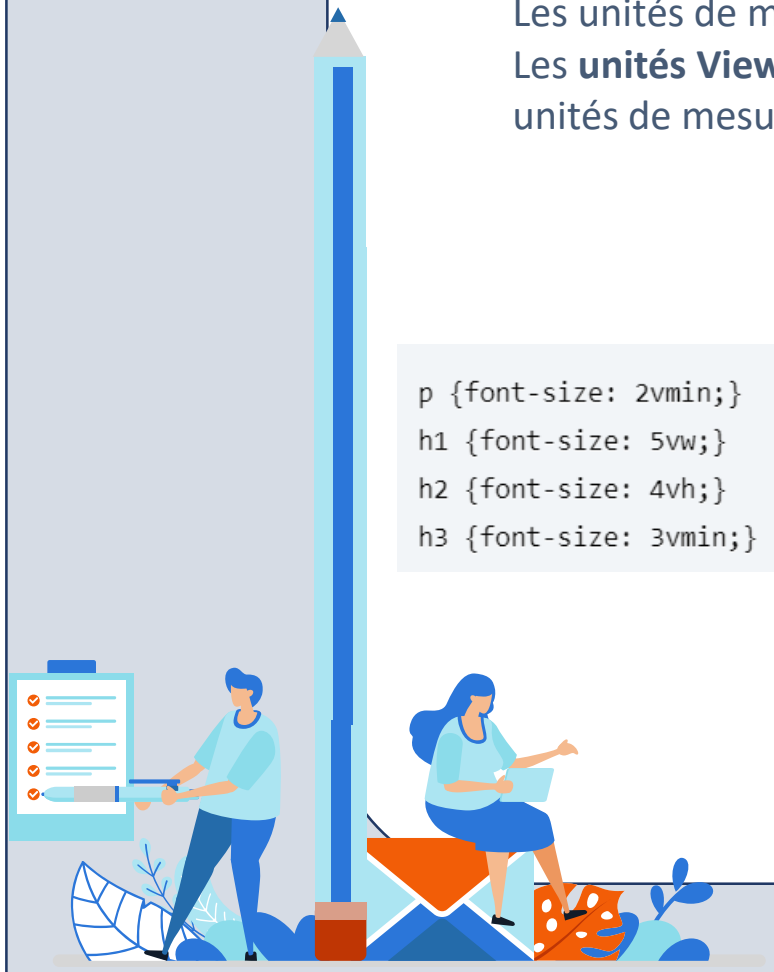
Encoder une police de caractères adaptative via CSS Viewport

Une variante pour la mise en place d'une police de caractères adaptative consiste à utiliser des unités de **CSS Viewport** (c'est-à-dire la taille de la fenêtre en Web design). CSS Viewport permet également d'adapter et de redimensionner les éléments de contenus de la page à la largeur de la fenêtre. L'avantage de cette solution par rapport aux média queries réside dans sa rapidité et une prise en main plus facile. Les unités de CSS Viewport sont exprimées en ***vw*** (« viewport width »), en ***vh*** (« viewport height »), parfois en ***vmax*** (« viewport maximum ») et en ***vmin*** (« viewport minimum »). Ces deux dernières permettent d'adapter la police de caractères soit à la hauteur, soit à la largeur de la fenêtre. La valeur minimale est déterminée par *vmin*, la valeur maximale par *vmax*. Ces quatre unités Viewport sont également exprimées en pourcentages. Toujours à titre d'exemple : *10vw* = 10% de la largeur de la fenêtre. Avec un Viewport de 640 x 480 pixels, le *vmax* serait de 640 pixels (la valeur maximale). Avec une fenêtre de navigateur de cette taille, la valeur *10vmax* est équivalente à *64px* (640 : 10). De manière similaire aux unités de mesure *rem*, les unités Viewport ne sont pas compatibles avec **toutes les versions de navigateurs**.

Mise en place des titres et des corps de texte adaptatifs avec CSS Viewport



Les unités de mesure relatives de CSS Viewport sont adaptées à la fois aux titres et au corps de texte.
Les **unités Viewport** s'effectuent de la même manière que les unités *rem* avec les requêtes média : Ces deux unités de mesure sont relatives.



```
p {font-size: 2vmin;}
h1 {font-size: 5vw;}
h2 {font-size: 4vh;}
h3 {font-size: 3vmin;}
```

Ces commandes CSS permettent de faire en sorte que la police du corps de texte (*p*) soit définie comme suit : (*font-size: 2vmin*) en fonction de la plus petite valeur, soit 2% de la largeur ou la hauteur de la fenêtre.
On garantit avec cela que les différentes écritures gardent la même proportion selon la taille de la fenêtre du navigateur.
La taille du titre principal (*h1*) reste à 5% de la largeur de la fenêtre de l'écran (*font-size*: *5vw*) tandis que la deuxième (*h2*) s'adapte à 4% de la hauteur de la fenêtre (*font-size*: *4vh*).
Le troisième et dernier titre (*h3*) est défini par la valeur (*3vmin)* et est **toujours plus grand que le corps de texte** (en 2vmin), à hauteur de 1%.

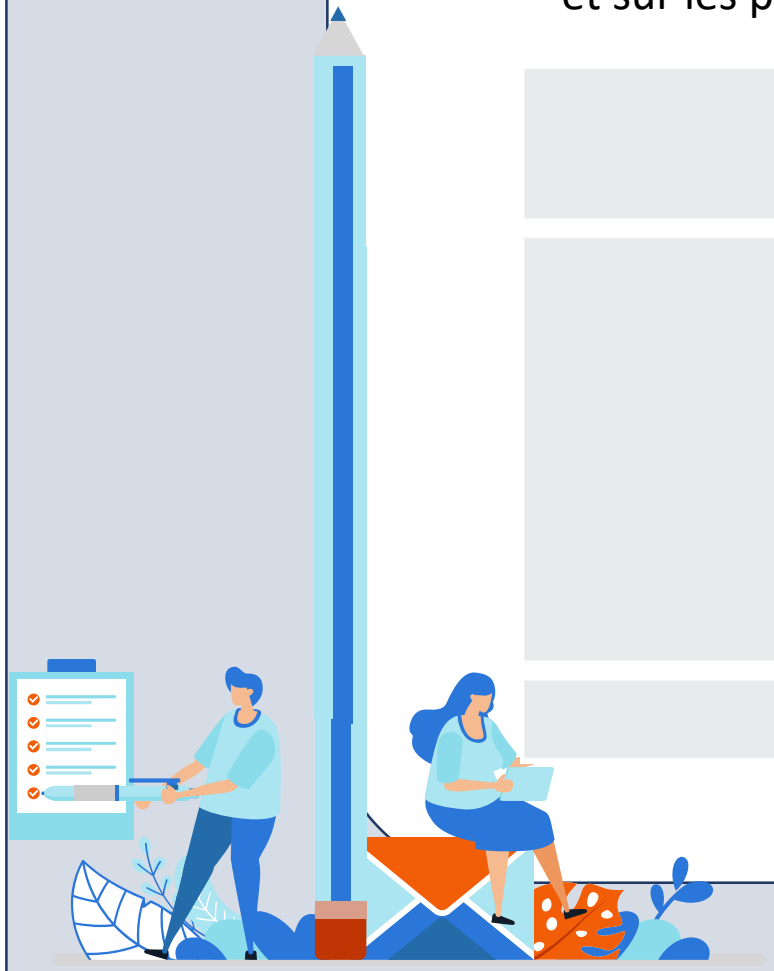
Ces quatre commandes CSS permettent de veiller à ce que la taille de la police de caractères s'adapte toujours à la taille de la fenêtre.

La police de caractères adaptative ne peut être négligée

L'application de quelques commandes CSS suffit à vous procurer un design adaptatif pour vos polices de caractères. Pensez à vérifier leur bon fonctionnement sur **différents appareils** (ou émulateurs) avant de lancer votre site Web.

Le recours aux requêtes média ou bien à CSS Viewport dépendra de la nature de votre projet. En ce qui concerne les media queries, leur échelonnage prend un peu plus de temps qu'avec les unités Viewport et sont bien plus flexibles. Vous pouvez également utiliser des Element Queries ou une extension jQuery pour adapter la taille de votre police de caractères. Toutes ces solutions permettent d'adapter la taille de la police d'écriture à celle de la fenêtre.

Une page Web réactive doit avoir une belle apparence sur les grands écrans de bureau et sur les petits téléphones mobiles.



Solution

Hello World

Link 1

Link 2

Link 3

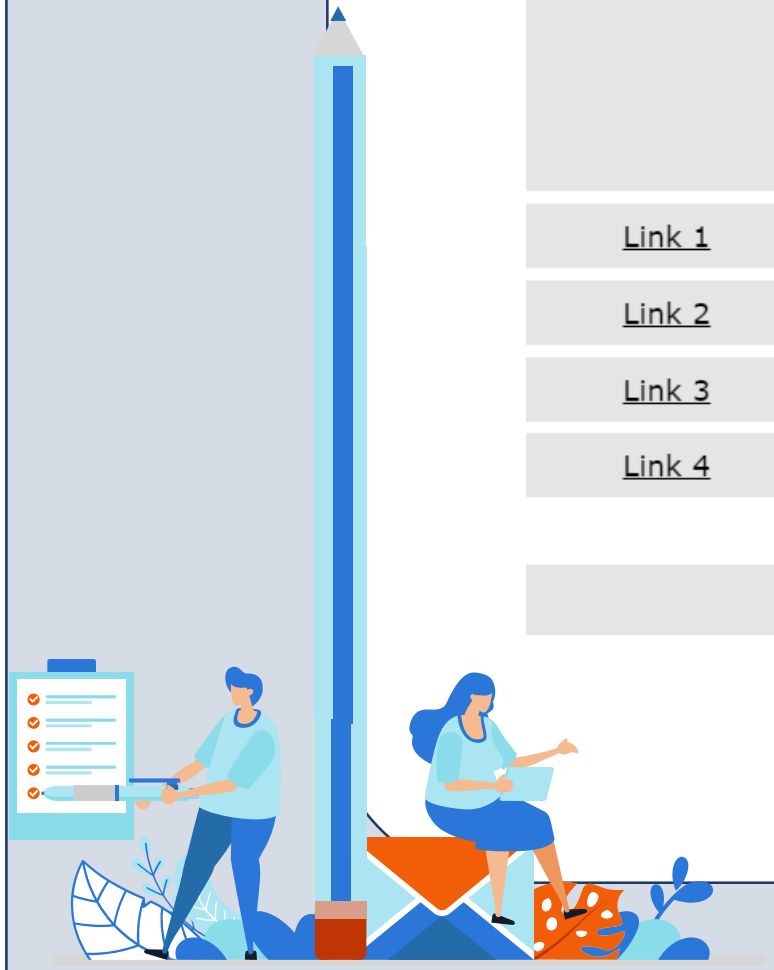
Link 4

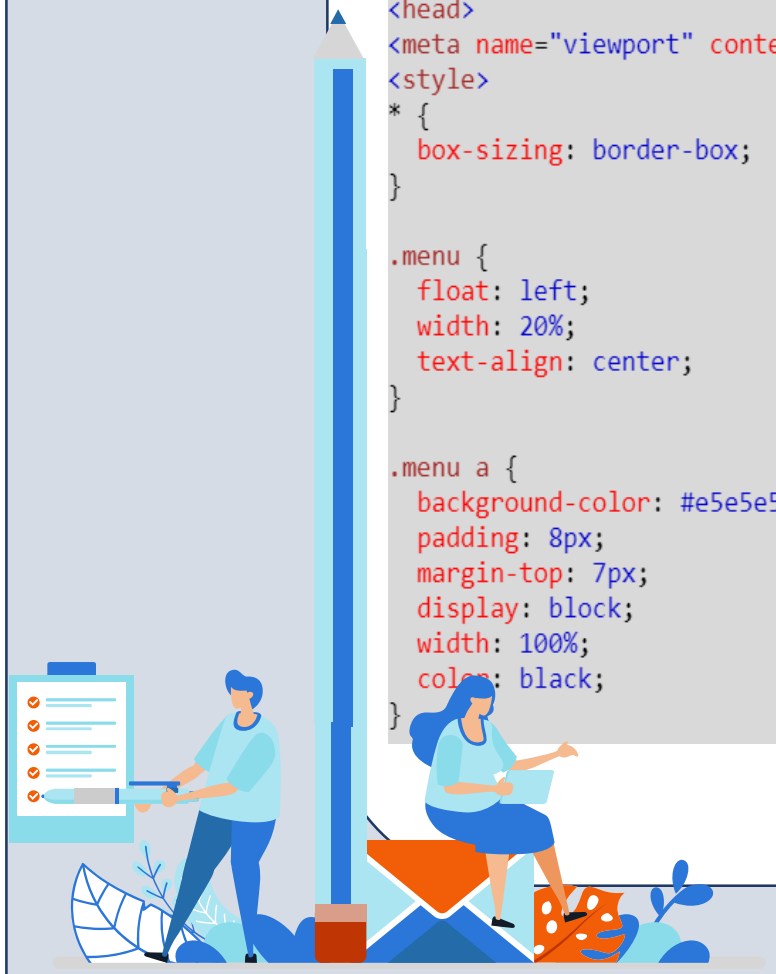
Lorum Ipsum

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

About

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.





```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
* {
  box-sizing: border-box;
}

.menu {
  float: left;
  width: 20%;
  text-align: center;
}

.menu a {
  background-color: #e5e5e5;
  padding: 8px;
  margin-top: 7px;
  display: block;
  width: 100%;
  color: black;
}
```

```
.main {
  float: left;
  width: 60%;
  padding: 0 20px;
}

.right {
  background-color: #e5e5e5;
  float: left;
  width: 20%;
  padding: 15px;
  margin-top: 7px;
  text-align: center;
}

@media only screen and (max-width: 620px) {
  /* For mobile phones: */
  .menu, .main, .right {
    width: 100%;
  }
}
</style>
</head>
<body style="font-family:Verdana;color:#aaaaaa;">
```

```
<div style="background-color:#e5e5e5;padding:15px;text-align:center;">
  <h1>Hello World</h1>
</div>

<div style="overflow:auto">
  <div class="menu">
    <a href="#">Link 1</a>
    <a href="#">Link 2</a>
    <a href="#">Link 3</a>
    <a href="#">Link 4</a>
  </div>

  <div class="main">
    <h2>Lorum Ipsum</h2>
    <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod
tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>
  </div>

  <div class="right">
    <h2>About</h2>
    <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit.</p>
  </div>
</div>

<div style="background-color:#e5e5e5;text-align:center;padding:10px;margin-top:7px;">© copyright
w3schools.com</div>

</body>
</html>
```