

19/01/2021

HTML 5 – Partie 3

Les images

HTML



MODULE B1.WEB

Introduction.....	2
Les images : 	2
La balise 	2
Les formats et poids des images	3
La balise <figure> et <figcaption>	4
Les styles sur image.....	4
image d'arrière plan.....	4
L'opacité d'une image	5
Les dimensions d'une image	5
L'alignement des images	6

INTRODUCTION

Certains éléments en HTML ne vont être constitués que d'une balise qu'on appelle alors orpheline. Cela va être le cas pour certains éléments qui ne possèdent pas de contenu textuel comme l'élément `
` par exemple qui sert simplement à créer un retour à la ligne en HTML. C'est le cas de la balise `` qui n'a pas pour vocation de contenir du texte par définition.

Voici quelques exemples :

```
<br />
<hr />
<img ... />
```

LES IMAGES :

LA BALISE

L'insertion d'une image est obtenue par la balise ``.

C'est une balise unique de type en ligne. Donc chaque image va se coller les unes aux autres dans la page en fonction de ses dimensions

La syntaxe de base est :

```
 <!--le / de fin n'est pas obligatoire -->

```

- L'attribut `src` indique que l'image provient d'un fichier externe. Celui-ci précise l'adresse relative ou absolue (comme pour la balise `<a>`) du fichier image qui doit être affiché dans la page.

Exemples :

```


<a href="lien"></a>
```

Si le lien vers l'image est dit "mort" une image cassée s'affichera :



- Les attributs `height` et `width` permettent de définir la hauteur et la largeur en pixels. Sans précisions de taille, l'image prend sa dimension normale. Plus l'image est volumineuse, plus elle mettra du temps à s'afficher dans la page. Il est donc important d'utiliser des outils de retouche d'image pour adapter celle-ci pour le Web.

Pour une image destinée à un fond de site qui occupera 100% de la page, il faut arriver à un compromis entre taille/poids et qualité sur tous les écrans. Nous recommandons une taille de 1920 x 1080px qui permet un affichage correct sur la majorité des écrans.

```

```

Si l'image dépasse la résolution de l'écran des scroll barre verticale et/ou horizontale sont ajoutés à la page ce qui esthétiquement n'est pas le plus adapté.



Une image trop petite avec une résolution modifiée avec `width` et `height` perdra en netteté.

Exemple : la même image affichée avec la même largeur et hauteur, mais avec une résolution différente. La différence est claire et nette

```


```

Visuel :



Figure 2 : Résolution de 1920 x 1080

Figure 1 : résolution de 19x10

Pour obtenir une image qui prend toute la largeur de la page et qui s'adapte à votre écran, le pixel n'est pas l'unité adaptée. Ici la solution est d'utiliser le % sur la largeur de l'image. Modifier la hauteur dans l'exemple n'aurait pas de sens et déformerait l'image en cassant les proportions.

Exemple :

```

```

- L'attribut `alt` est obligatoire. Il contient une description de l'image. Cet attribut sert de texte de remplacement si, pour une raison ou une autre, l'image ne peut pas s'afficher. Elle est aussi utile aux moteurs de recherche pour indexer les images. Enfin, les liseuses de site (pour les malvoyants) lisent cet attribut pour remplacer les images.
- L'attribut `title` permet d'afficher une infobulle lors du survol de l'image avec la souris. Peut prendre la même valeur que `alt`

LES FORMATS ET POIDS DES IMAGES

Pour que le téléchargement des images soit rapide il faut des formats d'image qui sont compressés : GIF, JPG, PNG.

- GIF : 256 couleurs, animation, mode transparent : utilisé pour les logos, les boutons, ou animations
- JPG: 16.7 millions de couleurs, perte de qualité après compression, pas d'animation, ni transparence : idéal pour les photos.
- PNG : format plus récent, recommandé par le W3C, 16.7 millions de couleurs, compression sans perte de qualité, permet la transparence, pas d'animation : bon compromis entre GIF et JPG.
- Tous autres formats, notamment le BMP est à éviter, car trop lourd.



Une photo prise d'un appareil photo pèse entre 4 et 8 Mo. Il faudra réduire le poids de la photo avant de l'intégrer dans votre page Web. Des logiciels gratuits comme GIMP, Paint.net, etc. permettent de retoucher rapidement une image.

Remarque : reprenons l'image de la figure 2 de 1920 x 1080 px. Sa taille sur le disque fait par exemple 700ko. Même en réduisant sa taille d'affichage à 100px en largeur (la hauteur s'adapte en respectant les proportions), il s'agit toujours de la même image. En réalité nous n'avons qu'effectué un "zoom moins" et le poids de l'image est toujours de 700ko

```

```

700ko ce n'est pas une image trop importante. Quand est-il d'une image de 4Mo ou plus. Une solution consiste à minimiser l'image de base.

 paysage2.jpeg	19/01/2021 16:58	Fichier JPEG	724 Ko
 paysage2-mini.jpeg	19/01/2021 22:08	Fichier JPEG	195 Ko

Dans la page web, on propose d'afficher d'abord l'image mini en lien cliquable. Lorsqu'on clique dessus, l'image principale s'affiche dans une nouvelle fenêtre par exemple.

Exemple :

```
<a href="paysage2.jpeg" target="_blank">
  
</a>
```

Rappelons aussi que vous hébergez votre site. Suivant ce dernier vous disposez d'un espace serveur et d'un débit limités. Il vous faudra donc déjà faire attention à ne pas le saturer avec des images inutilement lourdes.

LA BALISE <FIGURE> ET <FIGCAPTION>

L'élément HTML <figure> permet d'insérer des éléments d'illustration. C'est une balise du type conteneur multimédia : <video>, <audio>, et . L'élément <figcaption> permet d'indiquer une légende.

La balise <figure> a un rôle avant tout sémantique. Cela veut dire qu'elle indique au moteur de recherche que l'image a du sens et qu'elle est importante pour la bonne compréhension du texte.

```
<figure>
  
  <figcaption>un paysage de montagne</figcaption>
</figure>
```

LES STYLES SUR IMAGE

IMAGE D'ARRIÈRE PLAN

La propriété css **background-image** permet d'ajouter une image en fond d'un élément html. Souvent appliqué sur la balise <body> elle peut s'appliquer à tous type de boîtes <p>, <h>, <div> etc...

On utilise cette propriété lorsque l'image n'apporte pas de sens à la page et que celle-ci est purement décorative.

Remarque :

On ne met jamais en fond une image porteuse d'information. Ces images se mettent "en dur" dans le code html avec un texte alternatif (alt) qui va bien. Attention aussi à utiliser une image qui ne gênera pas la lecture de la page avec une couleur de fond différente de celle de la police de caractère.

Contrairement aux couches d'images qui peuvent être multiples, on ne peut avoir qu'une seule couche de couleur pour un élément donné. Donc si une couche de couleur est définie avec **background-color**, elle est dessinée en dessous de l'image et n'apparaîtra que si l'image n'est pas chargée.

De plus la propriété **background-image** autorise plusieurs

images. Dans ce cas elles vont se superposer les unes aux autres. Bien entendu si les images ne gèrent pas la transparence dans ce cas, c'est la première image qui sera affichée selon le principe des calques.

La propriété **background-repeat** définit la façon dont les images utilisées en arrière-plan sont répétées (**repeat**). Une image d'arrière-plan pourra ainsi être répétée sur l'axe horizontal **repeat-x**, l'axe vertical **repeat-y** ou bien ne pas être répétée, avec la valeur **no-repeat**.

background-size définit la taille des images d'arrière-plan pour l'élément. Avec la valeur **contain** l'image est redimensionnée afin qu'elle soit la plus grande possible et que l'image conserve ses proportions, Les zones éventuellement vides sont remplies avec la couleur d'arrière-plan. Avec **cover** les parties qui dépassent sont rognées. Cette propriété permet aussi de définir une taille personnelle en px ou %.

background-position définit la position de départ de l'image sur l'axe des x et des y. Par défaut, l'image se positionne en haut à gauche de l'écran. Sa valeur peut être un mot-clé (**top**, **right**, **bottom**, **left** et **center**), une longueur ou un pourcentage.

```
body{
  height: 100vh;
}
.bg2{
  background-image: url("../image/paysage2.jpeg");
  background-color: rgba(0, 0, 0, 0.5);
}
.bg3{
  background-image: url("../image/voiture-ancienne-
png.png");
  background-repeat: no-repeat;
  background-color: rgba(0, 0, 0, 0.5);
  background-size: contain;
}
.bg4{
  background-image: url("../image/voiture-ancienne-
png.png");
  background-repeat: repeat-y;
  background-color: rgba(0, 0, 0, 0.5);
}
.bg5{
  background-image: url("../image/voiture-ancienne-
png.png");
  background-repeat: no-repeat;
  background-color: rgba(0, 0, 0, 0.5);
  background-size: 200px;
  background-position: center center;
  background-attachment: fixed;
}
p{
  color: white;
}
```

La propriété **background-attachment** définit si la position de l'image d'arrière-plan est fixée dans la zone d'affichage.

L'OPACITÉ D'UNE IMAGE

La propriété CSS **opacity** permet de gérer la transparence d'une image ou de tous autres éléments HTML. Cette propriété est la même dans **background-color**. C'est une valeur comprise entre **0** (complètement transparent) et **1** (complètement visible). Cette propriété est prise en charge par la plupart des navigateurs, mais certains plus anciens utilisent une autre propriété nommée **filter**.

Exemple : ici les images ne sont pas transparentes (1) en temps normal. En cas de survol de l'image, l'opacité passe à 0.5.

```
img.opacite{
  opacity: 1;
}
img.opacite:hover{
  opacity: 0.5;
}
```



Lorem ipsum dolor sit amet, consectetur adipisicing
occaecat cupidatat non proident, sunt in culpa qui o



Lorem ipsum dolor sit amet, consectetur adipisicing
occaecat cupidatat non proident, sunt in culpa qui o

Attention une image même complètement transparente (opacity à 0) reste un bloc à part entière dans la page. Elle garde sa position dans la page. Dans l'exemple le <p> suivant, l'image restera à la même position.

LES DIMENSIONS D'UNE IMAGE

Nous l'avons vu dans la partie html il est possible de modifier la largeur et la hauteur d'une image avec les attributs **width** et **height**. Ces mêmes attributs existent en CSS pour tout type de balise html y compris une image.

Si seulement une seule de ses propriétés est définie (width par exemple) le navigateur redimensionne également l'autre dimension (height par exemple) en respectant les proportions de l'image.

L'idéal pour ces propriétés est d'utiliser comme unité le % ce qui va permettre à l'image de s'adapter à la résolution de l'écran. Dans l'exemple qui suit 2 images avec 2 résolutions d'écran différentes. L'image de gauche est en px, celle de droite en %.



Figure 3 : Résolution de 900x900px

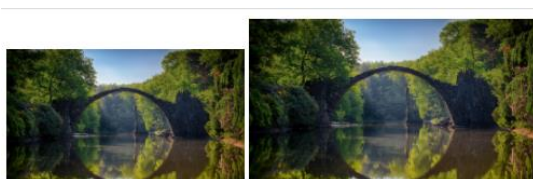


Figure 4 : résolution de 500x900px



Figure 5 : 300 x 900px

Si l'on réduit trop (ou agrandi trop) la résolution de l'écran, l'image risque de ne plus être visible correctement. Pour éviter cela il existe 4 propriétés qui fixent des limites à la hauteur et à la largeur : **min-width**, **max-width**, **min-height**, **max-height** qui permettent de définir la largeur minimum et maximum, ainsi que la hauteur minimum et maximum que l'on ne doit pas dépasser.

Dans cet exemple, le premier à une largeur de 200px, l'autre image est en pourcentage (50%) avec une largeur minimum de 200px. Après avoir baissé la résolution de l'écran, les 2 images ont la même largeur.

Code CSS:

```
#img2{width: 200px;} //image de gauche Figure3,4 et 5: taille fixe
#img3{
  width: 50%; // l'image de droite s'adapte à 50% (Figure 3 et 4)
```



```
min-width: 200px; //, mais pas moins que 200px : figure 5.
}
```

Par contre si les 2 propriétés sont définies les proportions choisies risquent de déformer l'image.

Parfois on est contraint de mettre une image dans une "boîte" avec une dimension fixe (par exemple 200x200px). Comme nous l'avons expliqué précédemment le risque est d'avoir une image déformée. La propriété **object-fit** avec la valeur **cover** permet de réduire cette déformation tout en ayant l'image dans la "boîte". On note que l'image de droite est beaucoup plus nette que celle de gauche, mais elle n'est pas affichée dans sa globalité.

Code :

```
#img4{
  height: 200px;
  width: 200px;
}
#img5{
  height: 200px;
  object-fit: cover;
  width: 200px;
}
```



L'ALIGNEMENT DES IMAGES

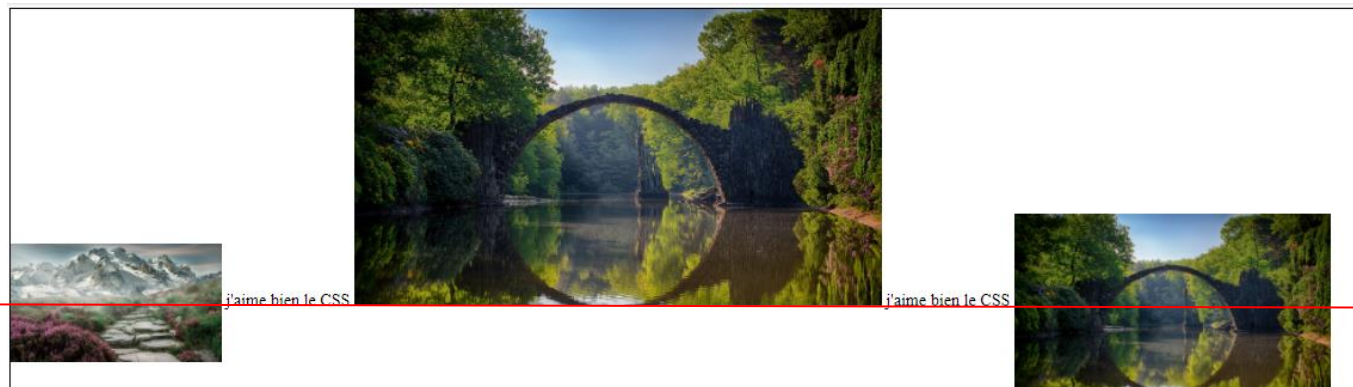
La propriété **vertical-align** permet de définir l'alignement des éléments de type **inline** (en ligne) . Les `` et `` en font parties.

Rappel : un élément **inline** prend en largeur et en hauteur uniquement la place nécessaire pas plus, ni moins.

Prenons l'exemple suivant : 3 images de tailles différentes et du texte entre les images dans une balise conteneur `<div>`. On constate que par défaut les images et le texte se placent sur une ligne imaginaire représentée ici en rouge que l'on appellera ligne de base fluctuante. Par défaut tous nos éléments ont donc leur propriétés **vertical-align** à la valeur **baseline**



Supposons que la 1^{ère} et la dernière image soient alignées verticalement au milieu (**middle**). Le positionnement de cette image va se faire par rapport ici à notre ligne de base ici en rouge.



Avec les 3 images positionnées au **milieu**. Les éléments sont restés sur **baseline** ce qui nous permet de repérer la ligne de base.

Pour bien comprendre encore un exemple. La 2^e image est positionnée au milieu. Les autres éléments sont sur baseline. Cela donne logiquement.



Pour mieux comprendre ce style :

https://www.w3schools.com/cssref/playit.asp?filename=playcss_vertical-align&preval=baseline