



SOMMAIRE

INTRODUCTION.....	2
LES TYPES DE BALISES.....	2
Balises En Paires.....	2
Balises Orphelines.....	2
En-tête.....	3
L'encodage.....	3
Commentaire.....	3
Les Listes Et Les Puces	3
Balises de structuration de texte.....	5
Balise Inline Et Block	5
Structurer Une Page	7
Tableaux.....	7
Les Formulaires :	8
Zone de texte monoligne	8
Les libellés	8
Propriété input :	8
Zone de mot de passe.....	8
Zone de texte multiligne	8
Les zones de saisie enrichies	9
Les cases à cocher.....	10
Les zones d'options (radio bouton).....	11
Les listes déroulantes	12
Finaliser Et Envoyer Le Formulaire.....	13
Le Bouton d'envoi.....	14
Résumé :	15
La Vidéo Et L'audio.....	15
Les formats audio	16

Les formats vidéo	16
Insertion d'un élément audio	17
Insertion d'une vidéo	18
Résumé :	19
CSS3	20
Trois manières différentes pour utiliser CSS	20
Commentaires :	20
Les sélecteurs :	20
Utiliser une police personnalisée avec @font-face :	20
Mise en forme du texte :	22
Propriété des boîtes	23
Propriété de couleur et de fond	24
Propriété des listes et des tableaux	24
Propriétés de positionnement et d'affichage	25
Padding Et Margin	25
Les pseudos format :	25
Les Dégradés linéaires :	27
Dégradé circulaire :	28
Border-radius	28
Text-overflow	29
Attributs personnalisés	30
Les Transitions	31
Animations	33

INTRODUCTION

- Tim Berners-Lee est l'inventeur de web.
- HTML (HyperText Markup Language) apparaissait en 1991.
- CSS (Cascading Style Sheets) Feuille de style en français apparaissait en 1996
- Un programme WYSIWYG ex : DreamWeaver, Frontpage,....
- HTML pour le fond, css pour la forme

LES TYPES DE BALISES

Balises En Paires

Balise en paires (balise ouvrante + balise fermante)

Ex : <titre>c'est un titre </titre>

Balises Orphelines

Balise orpheline (n'est pas nécessaire de délimiter le début et la fin)

Ex : <image/>

Le slash n'est pas obligatoire mais c'est pratique .

En-tête

<!DOCTYPE html> il s'agit d'une page web(HTML5) >

Balises d'en-tête

Ces balises sont toutes situées dans l'en-tête de la page web, c'est-à-dire entre <head> et </head> :

Balise	Description
<link />	Liaison avec une feuille de style
<meta />	Métadonnées de la page web (charset, mots-clés, etc.)
<script>	Code JavaScript
<style>	Code CSS
<title>	Titre de la page

L'encodage

<meta charset= "utf-8"/>

Commentaire

<! - - ceci est un commentaire - ->

Les Listes Et Les Puces

Type de liste	Exemple de code	Résultat
Liste non ordonnée	Langage de programmation VB Pascal C/C++ 	Langage de programmation <ul style="list-style-type: none">• VB• Pascal• C/C++
Liste ordonnée	Microsoft Office Word Excel Access 	Microsoft Office <ol style="list-style-type: none">1. Word2. Excel3. Access
Liste de définition	<dl> définition de termes : <dt>HTML <dd>Hyer Text Markup Language <dt>WWW <dd>World Wide Web </dl>	définition de termes : HTML Hyer Text Markup Language WWW World Wide Web

Les types de :

<ul type=?> : disc, square, circle

Les type de :

Type	Résultat
Développement web <ol type = "1"> Html Javascript Java 	Développement web 1. Html 2. Javascript 3. Java
Développement web <ol type = "A"> Html Javascript Java 	Développement web A. Html B. Javascript C. Java
Développement web <ol type = "a"> Html Javascript Java 	Développement web a Html b Javascript c Java
Développement web <ol type = "I"> Html Javascript Java 	Développement web I. Html II. Javascript III. Java

Balises de structuration du texte

Balise	Description
<abbr>	Abréviation
<blockquote>	Citation (longue)
<cite>	Citation du titre d'une œuvre ou d'un évènement
<q>	Citation (courte)
<sup>	Exposant
<sub>	Indice
	Mise en valeur forte
	Mise en valeur normale
<mark>	Mise en valeur visuelle
<h1>	Titre de niveau 1
<h2>	Titre de niveau 2
<h3>	Titre de niveau 3
<h4>	Titre de niveau 4
<h5>	Titre de niveau 5
<h6>	Titre de niveau 6
	Image
<figure>	Figure (image, code, etc.)
<figcaption>	Description de la figure
<audio>	Son
<video>	Vidéo
<source>	Format source pour les balises <audio> et <video>
<a>	Lien hypertexte
 	Retour à la ligne
<p>	Paragraphe
<hr />	Ligne de séparation horizontale
<address>	Adresse de contact
	Texte supprimé
<ins>	Texte inséré
<dfn>	Définition
<kbd>	Saisie clavier
<pre>	Affichage formaté (pour les codes sources)
<progress>	Barre de progression
<time>	Date ou heure

REMARQUE : si le lien contient & il doit être converti en & (validation html5)

Balise Inline Et Block

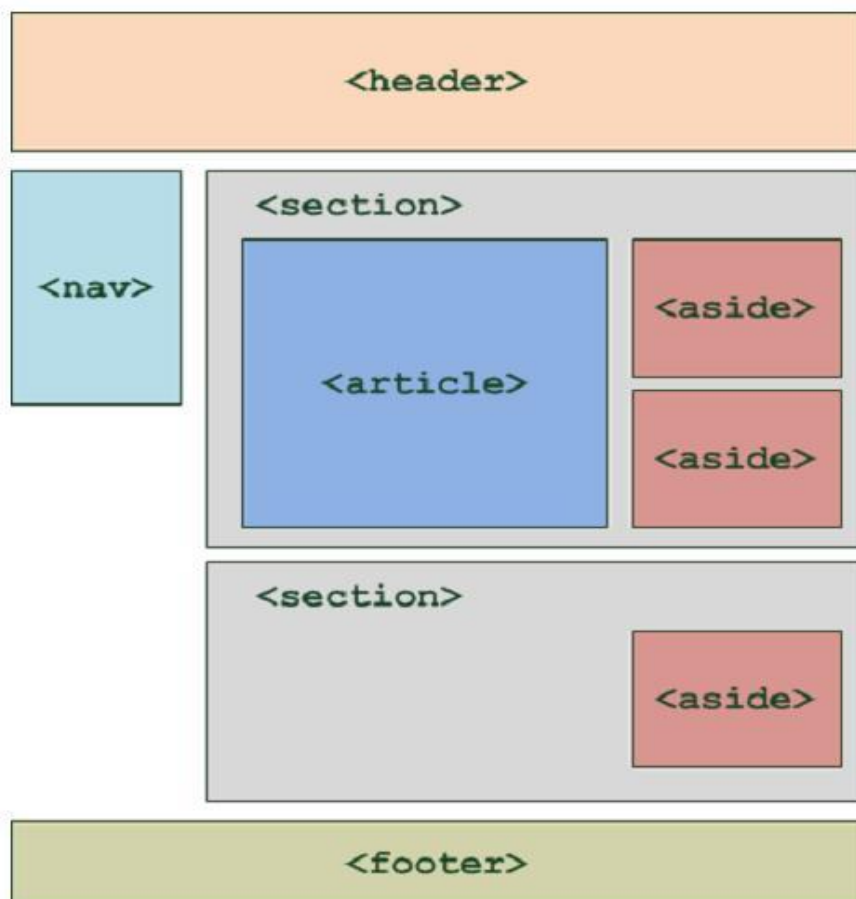
Valeur	Exemples	Description
inline	<a>, , ...	Éléments d'une ligne. Se placent les uns à côté des autres.
block	<p>, <div>, <section>...	Éléments en forme de blocs. Se placent les uns en-dessous des autres et peuvent être redimensionnés.
inline-block	<select>, <input>	Éléments positionnés les uns à côté des autres (comme les inlines) mais qui peuvent être redimensionnés (comme les blocs).
none	<head>	Éléments non affichés.

Balises block	Balises inline
<p>	
<footer>	
<h1>	<mark>
<h2>	<a>
<article>	
...	...

3 Méthodes pour la mise en forme

- La mise en page d'un site web s'effectue en CSS. Plusieurs techniques sont à notre disposition.
- Le positionnement flottant (avec la propriété `float`) est l'un des plus utilisés à l'heure actuelle. Il permet par exemple de placer un menu à gauche ou à droite de la page. Néanmoins, cette propriété n'a pas été initialement conçue pour cela et il est préférable, si possible, d'éviter cette technique.
- Le positionnement `inline-block` consiste à affecter un type `inline-block` à nos éléments grâce à la propriété `display`. Ils se comporteront comme des inlines (placement de gauche à droite) mais pourront être redimensionnés comme des blocs (avec `width` et `height`). Cette technique est à préférer au positionnement flottant.
- Le positionnement absolu permet de placer un élément où l'on souhaite sur la page, au pixel près.
- Le positionnement fixe est identique au positionnement absolu mais l'élément restera toujours visible même si on descend plus bas dans la page.
- Le positionnement relatif permet de décaler un bloc par rapport à sa position normale.
- Un élément A positionné en absolu à l'intérieur d'un autre élément B (lui-même positionné en absolu, fixe ou relatif) se positionnera par rapport à l'élément B, et non par rapport au coin en haut à gauche de la page.

Structurer Une Page



Tableaux

Diagram illustrating the structure of a table:

Nom	Age	Pays
Anne	27 ans	France
Carmen	33 ans	Espagne
Michelle	26 ans	Etats-Unis
Ogasaku Nyagatosoka	18 ans	Japon

Labels in the diagram:

- Tr (ligne)**: Points to a row in the table.
- Td (cellules)**: Points to individual cells in the table.

Passagers du vol 377

Nom	Age	Pays
Carmen	33 ans	Espagne
Michelle	26 ans	Etats-Unis
François	43 ans	France
Martine	34 ans	France
Jonathan	13 ans	Australie
Xu	19 ans	Chine

Labels in the diagram:

- En-tête du tableau**: Points to the header row.
- Corps du tableau**: Points to the body rows.
- Pied du tableau**: Points to the footer row.

EXEMPLE :

```
<h1>Titre (block)</h1>
```

```
<p>Paragraphe blablabla blablabla  
blablabla <a>Lien inline</a> blabla  
blablabla et toujours blabla  
</p>
```

```
<p>Encore un paragraphe (block)
```

```
</p>
```

Les Formulaires :

```
1 | <p>Texte avant le formulaire</p>
2 |
3 | <form method="post" action="traitement.php">
4 |     <p>Texte à l'intérieur du formulaire</p>
5 | </form>
6 |
7 | <p>Texte après le formulaire</p>
```

Zone de texte monoligne

Votre pseudo :

```
1 | <input type="text" />
```

Les libellés

```
1 | <form method="post" action="traitement.php">
2 |     <p>
3 |         <label for="pseudo">Votre pseudo</label> : <input type="
4 |             text" name="pseudo" id="pseudo" />
5 |     </p>
6 | </form>
```

Propriété input :

- On peut agrandir le champ avec **size**.
- On peut limiter le nombre de caractères que l'on peut saisir avec **maxlength**.
- On peut pré-remplir le champ avec une valeur par défaut à l'aide de **value**.
- On peut donner une indication sur le contenu du champ avec **placeholder**. Cette indication disparaîtra dès que le visiteur aura cliqué à l'intérieur du champ.

Zone de mot de passe

```
1 | <form method="post" action="traitement.php">
2 |     <p>
3 |         <label for="pseudo">Votre pseudo :</label>
4 |         <input type="text" name="pseudo" id="pseudo" />
5 |
6 |         <br />
7 |         <label for="pass">Votre mot de passe :</label>
8 |         <input type="password" name="pass" id="pass" />
9 |
10 |     </p>
11 | </form>
```

Votre pseudo :

Votre mot de passe :

Zone de texte multiligne


```

1 | <form method="post" action="traitement.php">
2 |   <p>
3 |     <label for="ameliorer">Comment pensez-vous que je
        pourrais améliorer mon site ?</label><br />
4 |     <textarea name="ameliorer" id="ameliorer"></textarea>
5 |   </p>
6 | </form>

```

Comment pensez-vous que je pourrais améliorer mon site ?

- *Propriété :*

- **En CSS** : il suffit d'appliquer les propriétés CSS `width` et `height` au `<textarea>`.
- **Avec des attributs** : on peut ajouter les attributs `rows` et `cols` à la balise `<textarea>`. Le premier indique le nombre de lignes de texte qui peuvent être affichées simultanément, et le second le nombre de colonnes.

Les zones de saisie enrichies

- *E-mail*

```

1 | <input type="email" />

```

- *Une URL*

```

1 | <input type="url" />

```



FIGURE 16.8 – Clavier de saisie d'URL sur iPhone

- *Numéro de téléphone*

```

1 | <input type="tel" />

```



FIGURE 16.9 – Clavier de saisie de numéro de téléphone sur un iPhone

- *Nombre*

```
1 | <input type="number" />
```

- *Un curseur*

```
1 | <input type="range" />
```

- *Couleur*

```
1 | <input type="color" />
```

- *Date*

Différents types de champs de sélection de date existent :

- `date` : pour la date (05/08/1985 par exemple) ;
- `time` : pour l'heure (13:37 par exemple) ;
- `week` : pour la semaine ;
- `month` : pour le mois ;
- `datetime` : pour la date et l'heure (avec gestion du décalage horaire) ;
- `datetime-local` pour la date et l'heure (sans gestion du décalage horaire).

Exemple :

```
1 | <input type="date" />
```

- *Recherche*

```
1 | <input type="search" />
```

Les cases à cocher

```
1 | <input type="checkbox" name="choix" />
```

```

1 | <form method="post" action="traitement.php">
2 |     <p>
3 |         Cochez les aliments que vous aimez manger :<br />
4 |         <input type="checkbox" name="frites" id="frites" /> <
5 |             label for="frites">Frites</label><br />
6 |         <input type="checkbox" name="steak" id="steak" /> <label
7 |             for="steak">Steak haché</label><br />
8 |         <input type="checkbox" name="epinards" id="epinards" />
9 |             <label for="epinards">Epinards</label><br />
10 |        <input type="checkbox" name="huitres" id="huitres" /> <
11 |            label for="huitres">Huitres</label>
12 |     </p>
13 | </form>

```

Cochez les aliments que vous aimez manger :

- ☒ Frites
- ☒ Steak haché
- ☐ Epinards
- ☐ Huitres

- *Coché par défaut*

```

1 | <input type="checkbox" name="choix" checked />

```

Les zones d'options (radio bouton)

```

1 <form method="post" action="traitement.php">
2   <p>
3     Veuillez indiquer la tranche d'âge dans laquelle vous
      vous situez :<br />
4     <input type="radio" name="age" value="moins15" id="
      moins15" /> <label for="moins15">Moins de 15 ans</
      label><br />
5     <input type="radio" name="age" value="medium15-25" id="
      medium15-25" /> <label for="medium15-25">15-25 ans</
      label><br />
6     <input type="radio" name="age" value="medium25-40" id="
      medium25-40" /> <label for="medium25-40">25-40 ans</
      label><br />
7     <input type="radio" name="age" value="plus40" id="plus40
      " /> <label for="plus40">Encore plus vieux que ça ?!<
      /label>
8   </p>
9 </form>

```

Veuillez indiquer la tranche d'âge dans laquelle vous vous situez :

- ☐ Moins de 15 ans
- ☐ 15-25 ans
- ☒ 25-40 ans
- ☐ Encore plus vieux que ça ?!



Vous noterez que cette fois on a choisi un id différent de name. En effet, les valeurs de name étant identiques, on n'aurait pas pu différencier les *id* (et vous savez bien qu'un id doit être unique!). Voilà donc pourquoi on a choisi de donner à l'id la même valeur que value.

Les listes déroulantes

```

1 <form method="post" action="traitement.php">
2   <p>
3     <label for="pays">Dans quel pays habitez-vous ?</label><
      br />
4     <select name="pays" id="pays">
5       <option value="france">France</option>
6       <option value="espagne">Espagne</option>
7       <option value="italie">Italie</option>
8       <option value="royaume-uni">Royaume-Uni</option>
9       <option value="canada">Canada</option>
10      <option value="etats-unis">États-Unis</option>
11      <option value="chine">Chine</option>
12      <option value="japon">Japon</option>
13    </select>
14  </p>
15 </form>

```

Dans quel pays habitez-vous ?

- *Coché par défaut :*

```

1 <option value="canada" selected>Canada</option>

```

- **Faire un Groupe :**

```

1 <form method="post" action="traitement.php">
2   <p>
3     <label for="pays">Dans quel pays habitez-vous ?</label><br />
4     <select name="pays" id="pays">
5       <optgroup label="Europe">
6         <option value="france">France</option>
7         <option value="espagne">Espagne</option>
8         <option value="italie">Italie</option>
9         <option value="royaume-uni">Royaume-Uni</option>
10      </optgroup>
11      <optgroup label="Amérique">
12        <option value="canada">Canada</option>
13        <option value="etats-unis">Etats-Unis</option>
14      </optgroup>
15      <optgroup label="Asie">
16        <option value="chine">Chine</option>
17        <option value="japon">Japon</option>
18      </optgroup>
19    </select>
20  </p>
21 </form>

```

Dans quel pays habitez-vous ?

Finaliser Et Envoyer Le Formulaire

Si votre formulaire grossit et comporte beaucoup de champs, il peut être utile de les regrouper au sein de plusieurs balises `<fieldset>`. Chaque `<fieldset>` peut contenir une légende avec la balise `<legend>`. Regardez cet exemple :

```

<fieldset>
  <legend>Vos coordonnées</legend> <!-- Titre du fieldset -->
  -->

```

- **Sélectionner automatiquement un champ**

```
1 | <input type="text" name="prenom" id="prenom" autofocus />
```

- *Rendre un champ obligatoire*

```
1 | <input type="text" name="prenom" id="prenom" required />
```

REMARQUE :



Les anciens navigateurs, qui ne reconnaissent pas cet attribut, enverront le contenu du formulaire sans vérification. Pour ces navigateurs, il sera nécessaire de compléter les tests avec, par exemple, des scripts JavaScript.



On dispose de pseudo-formats en CSS pour changer le style des éléments requis (:required) et invalides (:invalid). N'oubliez pas non plus que vous disposez du pseudo-format :focus pour changer l'apparence d'un champ lorsque le curseur se trouve à l'intérieur.

```
1 | :required
2 | {
3 |     background-color: red;
4 | }
```

Le Bouton d'envoi

Le bouton d'envoi

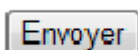
Il ne nous reste plus qu'à créer le bouton d'envoi. Là encore, la balise `<input />` vient à notre secours. Elle existe en quatre versions :

- `type="submit"` : le principal bouton d'envoi de formulaire. C'est celui que vous utiliserez le plus souvent. Le visiteur sera conduit à la page indiquée dans l'attribut `action` du formulaire.
- `type="reset"` : remise à zéro du formulaire.
- `type="image"` : équivalent du bouton `submit`, présenté cette fois sous forme d'image. Rajoutez l'attribut `src` pour indiquer l'URL de l'image.
- `type="button"` : bouton générique, qui n'aura (par défaut) aucun effet. En général, ce bouton est géré en JavaScript pour exécuter des actions sur la page. Nous ne l'utiliserons pas ici.



On peut changer le texte affiché à l'intérieur des boutons avec l'attribut `value`.

```
1 | <input type="submit" value="Envoyer" />
```



En résumé

- Un formulaire est une zone interactive de la page, dans laquelle vos visiteurs peuvent saisir des informations.
- On délimite un formulaire avec la balise `<form>` à laquelle il faut ajouter deux attributs : `method` (mode d'envoi des données) et `action` (page vers laquelle le visiteur sera redirigé après envoi du formulaire et qui traitera les informations).
- Une grande partie des éléments du formulaire peut s'insérer avec la balise `<input />`. La valeur de son attribut `type` permet d'indiquer quel type de champ doit être inséré :
 - `text` : zone de texte ;
 - `password` : zone de texte pour mot de passe ;
 - `tel` : numéro de téléphone ;
 - `checkbox` : case à cocher ;
 - etc.
- La balise `<label>` permet d'écrire un libellé. On l'associe à un champ de formulaire avec l'attribut `for`, qui doit avoir la même valeur que l'id du champ de formulaire.
- On peut rendre un champ obligatoire avec l'attribut `required`, faire en sorte qu'il soit sélectionné par défaut avec `autofocus`, donner une indication dans le champ avec `placeholder`...
- Pour récupérer ce que les visiteurs ont saisi, le langage HTML ne suffit pas. Il faut utiliser un langage « serveur » comme PHP... Si vous voulez aller plus loin, il va donc falloir apprendre un nouveau langage !

Les formats audio

Les formats audio

Pour diffuser de la musique ou n'importe quel son, il existe de nombreux formats. La plupart d'entre eux sont compressés (comme le sont les images JPEG, PNG et GIF) ce qui permet de réduire leur poids :

- **MP3** : vous ne pouvez *pas* ne pas en avoir entendu parler ! C'est l'un des plus vieux, mais aussi l'un des plus compatibles (tous les appareils savent lire des MP3), ce qui fait qu'il est toujours très utilisé aujourd'hui.
- **AAC** : utilisé majoritairement par Apple sur iTunes, c'est un format de bonne qualité. Les iPod, iPhone et autres iPad savent les lire sans problème.
- **OGG** : le format Ogg Vorbis est très répandu dans le monde du logiciel libre, notamment sous Linux. Ce format a l'avantage d'être libre, c'est-à-dire qu'il n'est protégé par aucun brevet.
- **WAV (format non compressé)** : évitez autant que possible de l'utiliser car le fichier est très volumineux avec ce format. C'est un peu l'équivalent du Bitmap (BMP) pour l'audio.

Aucun navigateur ne gère tous ces formats à la fois. Retenez surtout la compatibilité pour les MP3 et OGG :

Navigateur	MP3	OGG
Internet Explorer	Oui	-
Chrome	Oui	Oui
Firefox	-	Oui
Safari	Oui	-
Opera	-	Oui

Les formats vidéo

Le stockage de la vidéo est autrement plus complexe. On a besoin de trois éléments :

- **Un format conteneur** : c'est un peu comme une boîte qui va servir à contenir les deux éléments ci-dessous. On reconnaît en général le type de conteneur à l'extension du fichier : AVI, MP4, MKV...
- **Un codec audio** : c'est le format du son de la vidéo, généralement compressé. Nous venons de les voir, on utilise les mêmes : MP3, AAC, OGG...
- **Un codec vidéo** : c'est le format qui va compresser les images. C'est là que les choses se corsent, car ces formats sont complexes et on ne peut pas toujours les utiliser gratuitement. Les principaux à connaître pour le Web sont :
 - **H.264** : l'un des plus puissants et des plus utilisés aujourd'hui... mais il n'est pas 100% gratuit. En fait, on peut l'utiliser gratuitement dans certains cas (comme la diffusion de vidéos sur un site web personnel), mais il y a un flou juridique qui fait qu'il est risqué de l'utiliser à tout va.
 - **Ogg Theora** : un codec gratuit et libre de droits, mais moins puissant que H.264. Il est bien reconnu sous Linux mais, sous Windows, il faut installer des programmes pour pouvoir le lire.
 - **WebM** : un autre codec gratuit et libre de droits, plus récent. Proposé par Google, c'est le concurrent le plus sérieux de H.264 à l'heure actuelle.

Quelle est la compatibilité des codecs vidéo sur les différents navigateurs ? Là encore, vous allez voir que c'est un joyeux bazar :

Navigateur	H.264	Ogg Theora	WebM
Internet Explorer	Oui	-	Oui
Chrome	-	Oui	Oui
Firefox	-	Oui	Oui
Safari	Oui	-	-
Opera	Oui	Oui	Oui

Insertion d'un élément audio

```
1 | <audio src="musique.mp3"></audio>
```

• Attributs

En pratique, c'est un peu plus compliqué que cela. Si vous testez ce code... vous ne verrez rien ! En effet, le navigateur va seulement télécharger les informations générales sur le fichier (on parle de **métadonnées**) mais il ne se passera rien de particulier.

- **controls** : pour ajouter les boutons « Lecture », « Pause » et la barre de défilement. Cela peut sembler indispensable, et vous vous demandez peut-être pourquoi cela n'y figure pas par défaut, mais certains sites web préfèrent créer eux-mêmes leurs propres boutons et commander la lecture avec du JavaScript.
- **width** : pour modifier la largeur de l'outil de lecture audio.
- **loop** : la musique sera jouée en boucle.
- **autoplay** : la musique sera jouée dès le chargement de la page. Évitez d'en abuser, c'est en général irritant d'arriver sur un site qui joue de la musique tout seul !
- **preload** : indique si la musique peut être préchargée dès le chargement de la page ou non. Cet attribut peut prendre les valeurs :
 - **auto** (par défaut) : le navigateur décide s'il doit précharger toute la musique, uniquement les métadonnées ou rien du tout.
 - **metadata** : charge uniquement les métadonnées (durée, etc.).
 - **none** : pas de préchargement. Utile si vous ne voulez pas gaspiller de bande passante sur votre site.



On ne peut pas forcer le préchargement de la musique, c'est toujours le navigateur qui décide. Les navigateurs mobiles, par exemple, ne préchargent jamais la musique pour économiser la bande passante (le temps de chargement étant long sur un portable).

```
1 | <audio src="hype_home.mp3" controls></audio>
```



• En cas d'erreur :

```
1 | <audio src="hype_home.mp3" controls>Veuillez mettre à jour  
   votre navigateur !</audio>
```



Je vous conseille de proposer une solution de secours en Flash, comme Dewplayer. Vous placerez le code correspondant à Flash entre les balises `<audio>` et `</audio>` : ainsi, les anciens navigateurs afficheront le lecteur Flash, tandis que les nouveaux afficheront le lecteur natif.



On a vu que certains navigateurs ne géraient pas le MP3, comment faire?

Il faut proposer plusieurs versions du fichier audio. Dans ce cas, on va construire notre balise comme ceci :

```
1 | <audio controls>
2 |     <source src="hype_home.mp3"></source>
3 |     <source src="hype_home.ogg"></source>
4 | </audio>
```

Insertion d'une vidéo



La balise `<video>` que nous allons découvrir est reconnue par tous les navigateurs récents, y compris Internet Explorer à partir de la version 9 (IE9).

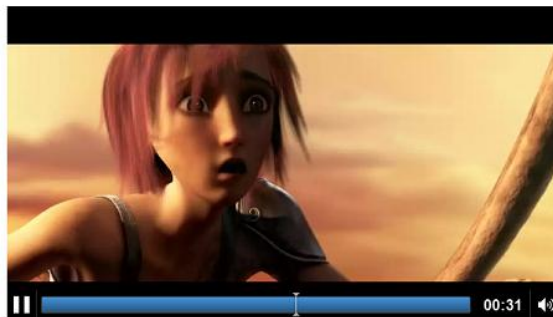
```
1 | <video src="sintel.webm"></video>
```

• Attributs

- **poster** : image à afficher à la place de la vidéo tant que celle-ci n'est pas lancée. Par défaut, le navigateur prend la première image de la vidéo mais, comme il s'agit souvent d'une image noire ou d'une image peu représentative de la vidéo, je vous conseille d'en créer une ! Vous pouvez tout simplement faire une capture d'écran d'un moment de la vidéo.
- **controls** : pour ajouter les boutons « Lecture », « Pause » et la barre de défilement. Cela peut sembler indispensable, mais certains sites web préfèrent créer eux-mêmes leurs propres boutons et commander la lecture avec du JavaScript. En ce qui nous concerne, ce sera largement suffisant !
- **width** : pour modifier la largeur de la vidéo.
- **height** : pour modifier la hauteur de la vidéo.
- **loop** : la vidéo sera jouée en boucle.
- **autoplay** : la vidéo sera jouée dès le chargement de la page. Là encore, évitez d'en abuser, c'est en général irritant d'arriver sur un site qui lance quelque chose tout seul !
- **preload** : indique si la vidéo peut être préchargée dès le chargement de la page ou non. Cet attribut peut prendre les valeurs :
 - **auto** (par défaut) : le navigateur décide s'il doit précharger toute la vidéo, uniquement les métadonnées ou rien du tout.
 - **metadata** : charge uniquement les métadonnées (durée, dimensions, etc.).
 - **none** : pas de préchargement. Utile si vous souhaitez éviter le gaspillage de bande passante sur votre site.

EXEMPLE :

```
1 | <video src="sintel.webm" controls poster="sintel.jpg" width="
   | 600"></video>
```



En cas d'erreur :

```
1 <video src="sintel.webm" controls poster="sintel.jpg" width="
  600">
2   Il est temps de mettre à jour votre navigateur !
3 </video>

1 <video controls poster="sintel.jpg" width="600">
2   <source src="sintel.mp4" />
3   <source src="sintel.webm" />
4   <source src="sintel.ogv" />
5 </video>
```



Les iPhone, iPad et iPod ne reconnaissent à l'heure actuelle que le format H.264 (fichier .mp4)... et uniquement si celui-ci apparaît en premier dans la liste ! Je vous recommande donc d'indiquer le format H.264 en premier pour assurer une compatibilité maximale.

Résumé :

En résumé

- Insérer de la musique ou de la vidéo n'était pas possible autrefois en HTML. Il fallait recourir à un plugin comme Flash.
- Depuis HTML5, les balises `<audio>` et `<video>` ont été introduites et permettent de jouer de la musique et des vidéos sans plugin.
- Il existe plusieurs formats audio et vidéo. Il faut notamment connaître :
 - pour l'audio : MP3 et Ogg Vorbis ;
 - pour la vidéo : H.264, Ogg Theora et WebM.
- Aucun format n'est reconnu par l'ensemble des navigateurs : il faut proposer différentes versions de sa musique ou de sa vidéo pour satisfaire tous les navigateurs.
- Il faut ajouter l'attribut `controls` aux balises `<audio>` et `<video>` pour permettre au visiteur de lancer ou d'arrêter le média.
- Ces balises ne sont pas conçues pour empêcher le téléchargement de la musique et de la vidéo. Vous ne pouvez pas protéger votre média contre la copie.

Trois manières différentes pour utiliser CSS

- 1.

```
<LINK REL="STYLESHEET" HREF="STYLE.CSS"
```

- 2.

```
<HEAD>
  <STYLE>
    P{COLOR:BLUE;}
  </STYLE>
</HEAD>
```

- 3.

```
<P STYLE="COLOR:BLUE";>BONJOUR</P>
```

Commentaires :

```
1  /*
2  style.css
3  -----
4
5  Par Mathieu Nebra
6  */
7
8  p
9  {
10     color: blue; /* Les paragraphes seront en bleu */
11 }
```

Les sélecteurs :

* {}	sélecteur universel
A B {}	sélecteur universel
A + B {}	une balise qui en suit une autre
A[attribut]	une balise qui possède un attribut
A[attribut]="valeur"	une balise, un attribut et une valeur exacte
A[attribut*="Valeur"]	une balise, un attribut et une valeur
A>B	S'adresse uniquement aux fils(non petites fils)

Utiliser une police personnalisée avec @font-face :

- Type de fonts :

- .ttf : *TrueType Font*. Fonctionne sur IE9 et tous les autres navigateurs.
- .eot : *Embedded OpenType*. Fonctionne sur Internet Explorer uniquement, toutes versions. Ce format est propriétaire, produit par Microsoft.
- .otf : *OpenType Font*. Ne fonctionne pas sur Internet Explorer.
- .svg : *SVG Font*. Le seul format reconnu sur les iPhones et iPads pour le moment.
- .woff : *Web Open Font Format*. Nouveau format conçu pour le Web, qui fonctionne sur IE9 et tous les autres navigateurs.

- *Déclarations*

```
1 | @font-face {
2 |     font-family: 'MaSuperPolice';
3 |     src: url('MaSuperPolice.eot') format('eot'),
4 |         url('MaSuperPolice.woff') format('woff'),
5 |         url('MaSuperPolice.ttf') format('truetype'),
6 |         url('MaSuperPolice.svg') format('svg');
7 | }
```

EXEMPLE:

```
1 | @font-face { /* Définition d'une nouvelle police nommée
2 |     LearningCurveProRegular */
3 |     font-family: 'LearningCurveProRegular';
4 |     src: url('LearningCurve_OT-webfont.eot');
5 |     src: url('LearningCurve_OT-webfont.eot?#iefix') format('
6 |         embedded-opentype'),
7 |         url('LearningCurve_OT-webfont.woff') format('woff'),
8 |         url('LearningCurve_OT-webfont.ttf') format('truetype')
9 |         ,
10 |        url('LearningCurve_OT-webfont.svg#
11 |            LearningCurveProRegular') format('svg');
12 | }
13 |
14 | h1 /* Utilisation de la police qu'on vient de définir sur les
15 |     titres */
16 | {
17 |     font-family: 'LearningCurveProRegular', Arial, serif;
18 | }
```

Propriétés de mise en forme du texte

Je résume ici la plupart des propriétés de **mise en forme du texte**.

Qu'est-ce que la mise en forme de texte ? C'est tout ce qui touche à la présentation du texte proprement dit : le gras, l'italique, le souligné, la police, l'alignement, etc.

Propriété	Description
font-family	Nom de police
@font-face	Police personnalisée
font-size	Taille du texte
font-weight	Gras
font-style	Italique
text-decoration	Soulignement, ligne au-dessus, barré ou clignotant
font-variant	Petites capitales
text-transform	Capitales
font	Super propriété de police. Combine : font-weight, font-style, font-size, font-variant, font-family.
text-align	Alignement horizontal
vertical-align	Alignement vertical (cellules de tableau ou éléments inline-block uniquement)
line-height	Hauteur de ligne
text-indent	Alinéa
white-space	Césure
word-wrap	Césure forcée
text-shadow	Ombre de texte

White-space : écrire sur une même ligne.

DECOUPER LES PARAGRAPHE EN COLONNES :

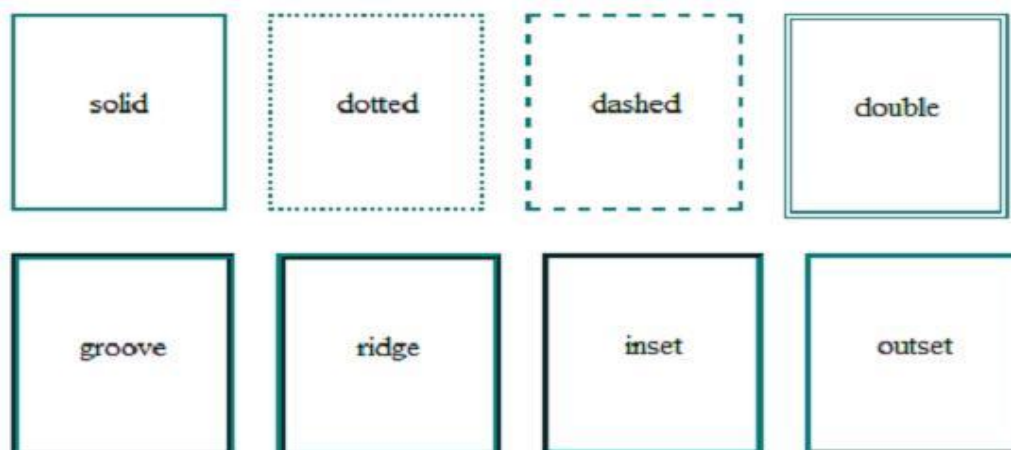
NOMBRE DE COLONNES => columns :4;

ESPACR ENTRE LES COLONNES => column-gap:4px;

STYLE DE FOND(COMME LA PROPRIETE BORDER) => column-rule:red dootted 1px;

Propriétés des boîtes

Propriété	Description
width	Largeur
height	Hauteur
min-width	Largeur minimale
max-width	Largeur maximale
min-height	Hauteur minimale
max-height	Hauteur maximale
margin-top	Marge en haut
margin-left	Marge à gauche
margin-right	Marge à droite
margin-bottom	Marge en bas
margin	Super-propriété de marge. Combine : margin-top, margin-right, margin-bottom, margin-left.
padding-top	Marge intérieure en haut
padding-left	Marge intérieure à gauche
padding-right	Marge intérieure à droite
padding-bottom	Marge intérieure en bas
padding	Super-propriété de marge intérieure. Combine : padding-top, padding-right, padding-bottom, padding-left.
border-width	Épaisseur de bordure
border-color	Couleur de bordure
border-style	Type de bordure
border	Super-propriété de bordure. Combine border-width, border-color, border-style. Existe aussi en version border-top, border-right, border-bottom, border-left.
border-radius	Bordure arrondie
box-shadow	Ombre de boîte



Propriété de couleur et de fond

Propriétés de couleur et de fond

Propriété	Description
color	Couleur du texte
background-color	Couleur de fond
background-image	Image de fond
background-attachment	Fond fixe
background-repeat	Répétition du fond
background-position	Position du fond
background	Super propriété du fond. Combine : background-image, background-repeat, background-attachment, background-position
opacity	Transparence

white	
silver	
grey	
black	
red	
maroon	
lime	
green	
yellow	
olive	
blue	
navy	
fuchsia	
purple	
aqua	
teal	

Propriété des listes et des tableaux

left	Position par rapport à la gauche
right	Position par rapport à la droite
z-index	Ordre d'affichage en cas de superposition

Propriétés des listes

Propriété	Description
list-style-type	Type de liste
list-style-position	Position en retrait
list-style-image	Puce personnalisée
list-style	Super-propriété de liste. Combine list-style-type, list-style-position, list-style-image.

Propriétés des tableaux

Propriété	Description
border-collapse	Fusion des bordures
empty-cells	Affichage des cellules vides
caption-side	Position du titre du tableau

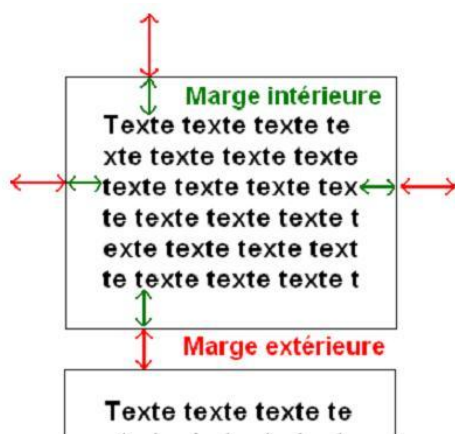
Autres propriétés

Propriété	Description
cursor	Curseur de souris

Propriétés de positionnement et d'affichage

Propriété	Description
display	Type d'élément (block, inline, inline-block, none...)
visibility	Visibilité
clip	Affichage d'une partie de l'élément
overflow	Comportement en cas de dépassement
float	Flottant
clear	Arrêt d'un flottant
position	Positionnement
top	Position par rapport au haut
bottom	Position par rapport au bas

Padding Et Margin



Les pseudos format :

NTH-CHILD(N) : selectionner les éléments enfants selon la position en partant de debut(de parent)

NTH-LAST-CHILD(N) : partant de la fin

NTH-OF-TYPE(N) : le nième élément de ce type

NTH-OF-TYPE(2N+1) : selectionner les impaires

:ONLY-CHILD

:ONLY-CHILD représente un élément qui n'a aucun frère

:ONLY-OF-TYPE renvoie tous les éléments qui sont seuls de leur type parmi leurs frères

Le sélecteur de négation **:NOT()**

EXEMPLE : Sélectionne tous les paragraphes ayant la classe small, exceptés ceux ayant la classe red.

P.SMALL:NOT(.RED)

Autre:

:CHECKED

Il permet de sélectionner les checkbox uniquement si elles sont cochées.

:EMPTY

sélectionne un élément n'ayant aucun enfant : aucune autre balise, aucun texte, rien du tout

:TARGET

sélectionner les éléments qui sont ciblés(id)

::AFTER ET ::BEFORE

EXEMPLE : afficher un contenu (lien)

```
a::after
{
    content: "attr(href)";
}
```

Partie HTML

```
<a href="http://www.alsacreations.com" target="_blank">Le site d'Alsacr ations</a>
```

Partie CSS

```
a[target]:after { /* on ne cible que les liens disposant d'un attribut target */
    content: " (nouvelle fen tre)";
}
```

::SELECTION Texte s lectionn e

:REQUIRED ET :OPTIONAL

:ENABLED, :DISABLED ET :CHECKED

:VALID, :INVALID

```
input:invalid {
    background-color: red;
}
input:valid {
    background-color: green;
}
```

BOX-SHADOW

- Le d calage de l'ombre sur l'axe des abscisses.
- Le d calage de l'ombre sur l'axe des ordonn es.
- Le rayon de flou.
- La distance de l'ombre par rapport   l' l ment (pour donner l'impression que ce dernier est plus ou moins  loign  de notre page).
- La couleur de l'ombre.



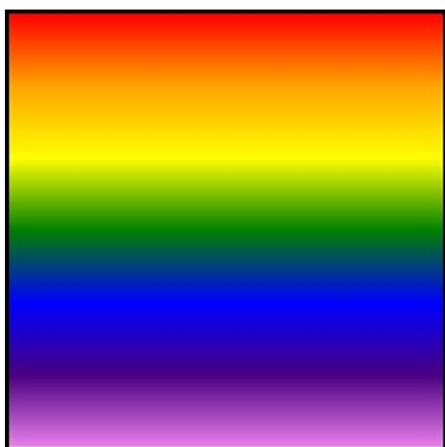
Avec **INSET** ombre interne

- Le premier paramètre indique alors la largeur de l'ombre du côté vertical. Si cette valeur est positive, elle sera collée contre le côté gauche de notre élément. Si elle est négative, elle sera contre le côté droit.
- Le second paramètre indique la hauteur de l'ombre du côté horizontal. Si cette valeur est positive, l'ombre sera collée contre le côté du haut de l'élément. Sinon, elle sera contre le côté du bas.

Les Dégradés linéaires :

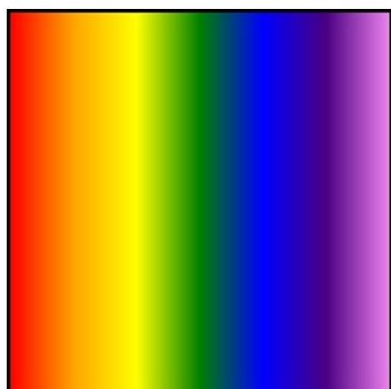
EXEMPLE : sans préfixe

BACKGROUND-IMAGE: LINEAR-GRADIENT (RED, ORANGE, YELLOW, GREEN, BLUE, INDIGO, VIOLET);



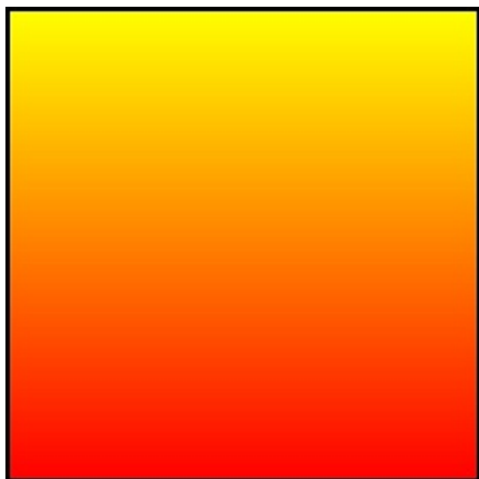
EXEMPLE : Utilisation des degrés

BACKGROUND-IMAGE: LINEAR-GRADIENT (0DEG, RED, ORANGE, YELLOW, GREEN, BLUE, INDIGO, VIOLET);

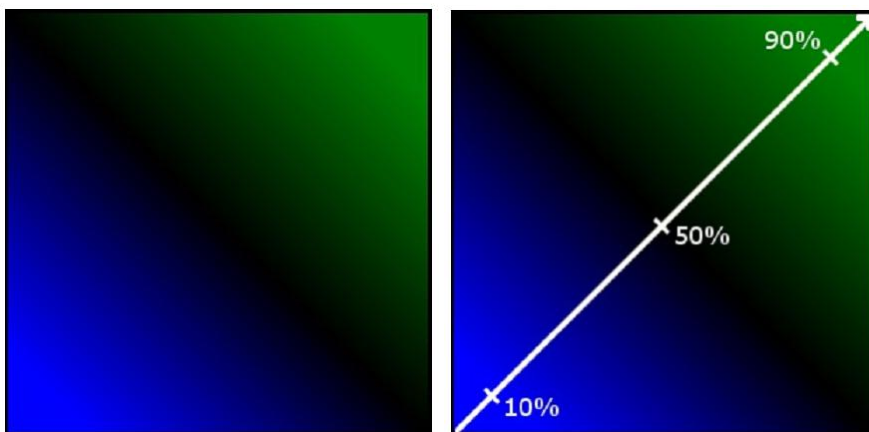


EXEMPLE : Utilisation des Pourcentage :

BACKGROUND-IMAGE: LINEAR-GRADIENT (YELLOW 0%, RED 100%);

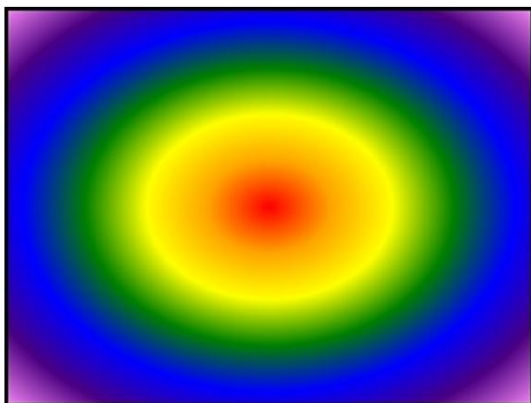


LINEAR-GRADIENT (45DEG, BLUE 10%, BLACK 50%, GREEN 90%);



Dégradé circulaire :

RADIAL-GRADIENT (RED, ORANGE, YELLOW, GREEN, BLUE, INDIGO, VIOLET);

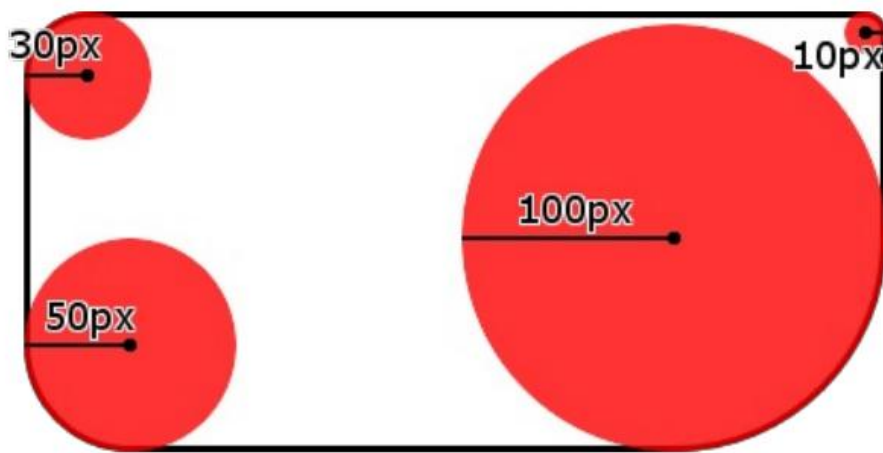


- *Spécifier la forme* : ellipse, circle

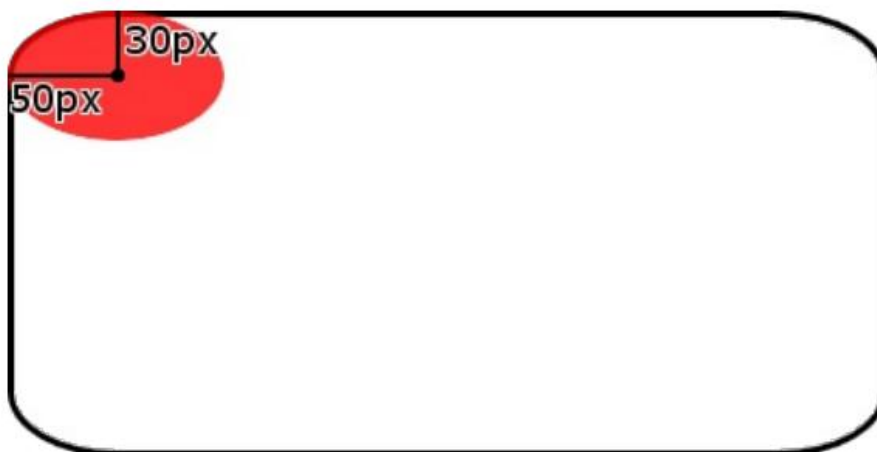
RADIAL-GRADIENT (CIRCLE, RED, ORANGE, YELLOW, GREEN, BLUE, INDIGO, VIOLET);

Border-radius

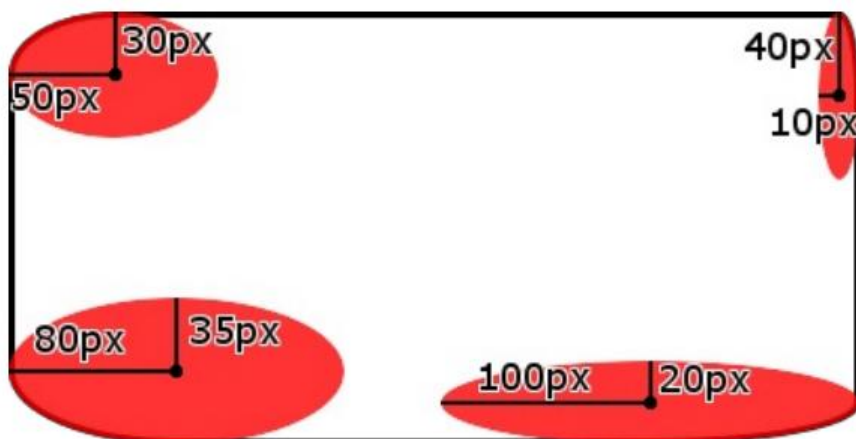
BORDER-RADIUS: 30PX 10PX 100PX 50PX;



BORDER-RADIUS: 50px / 30px;



BORDER-RADIUS: 50px 10px 100px 80px / 30px 40px 20px 35px;



Text-overflow

text-overflow

Dans certains contextes particuliers, pour ne pas dénaturer la mise en page d'un document, nous sommes amenés à masquer les contenus d'un élément dimensionnés à l'aide de la règle `overflow` (`hidden`, `scroll` ou `auto`).

Voici comment mettre en œuvre cette propriété :

```
p.hiding {
  overflow: hidden;
  -o-text-overflow: ellipsis; /* pour Opera */
  text-overflow: ellipsis; /* pour le reste du monde */
}
```

Figure 8-3

*Points de suspension
avec text-overflow*

L'adresse du lien suivant
devrait être masqué mais
suggéré via des points de
suspension

<http://www.supercalifra...>



overflow-x et overflow-y

Les propriétés `overflow-x` et `overflow-y` sont des variantes spécifiques de la propriété `overflow` définie en CSS 2, qui agissent sur un seul axe à la fois : `overflow-x` gère les débordements horizontaux tandis que `overflow-y` se charge des dépassement verticaux.

Les valeurs admises sont les mêmes que pour `overflow` :

- `visible` : le bloc est automatiquement redimensionné selon le contenu, aucune barre de défilement n'apparaît. Il s'agit de la valeur par défaut.
- `scroll` : une barre de défilement apparaît constamment, même si le contenu ne dépasse pas du bloc.
- `hidden` : le contenu excédentaire est purement et simplement masqué, sans qu'un ascenseur n'apparaisse.
- `auto` : les dimensions du bloc sont fixes et la barre de défilement apparaît uniquement dans le cas d'un contenu long.

Attributs personnalisés

Attributs personnalisés

N'avez-vous jamais été tenté d'employer un attribut créé de toutes pièces, non référencé dans les spécifications ? HTML 5 autorise la mise en œuvre de tels attributs personnalisés sous certaines conditions :

- Le nom de l'attribut doit être composé d'au moins un caractère et être préfixé de la chaîne `data-` afin d'être valide.
- Il ne doit pas contenir de caractères en majuscules.
- La valeur de l'attribut personnalisé peut être n'importe quelle chaîne de caractères.

L'une des nombreuses applications pratiques de ces attributs paramétrables pourrait être l'affichage d'infobulles personnalisées (figure 7-25) :

Partie HTML 5

```
<span data-tooltip="ceci est une infobulle">survolez-moi !</span>
```

Partie CSS

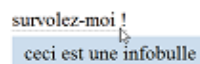
```
[data-tooltip] { /* on cible les attributs data-tooltip */
  position: relative;
  cursor: help;
  border-bottom: 1px dotted gray;
}
[data-tooltip]:hover:after {
  content: attr(data-tooltip); /* on récupère et affiche la valeur de data-tooltip */
  position: absolute;
  top: 1.5em; left: 0;
  white-space: nowrap; /* on empêche les retours à la ligne */
  padding: 5px 10px;
  background: #cde;
}
```

Visualiser le résultat

► <http://www.ie7nomore.com/css2only/tooltip/>

Figure 7-25

*Une infobulle
personnalisée valide*



Les Transitions

EXEMPLE :

```
img {
  width: 50px; /* état par défaut */
  transition-property: width; /* transition de la propriété width au survol
  ➡ de la souris */
  transition-duration: 1s; /* durée de la transition */
}
img:hover {
  width: 200px;
}
```

Propriété à animer : transition-property

La propriété `transition-property` accepte trois valeurs :

- `all` (valeur par défaut) : toutes les propriétés possibles seront animées ;
- `propriété` : le nom d'une propriété pouvant être animée ;
- `none` : aucune propriété ne sera animée.

Durée de l'animation : transition-duration

La propriété `transition-duration` indique la durée de la transition. Si plusieurs propriétés ont été définies à l'aide de `transition-property`, il est possible de préciser leurs valeurs en les séparant d'une virgule.

Les deux unités de temps acceptées sont :

- `s` : la seconde ;
- `ms` : la milliseconde.

Pour préciser une durée de 5 secondes pour la transition, nous pouvons alors utiliser la règle suivante :

Accélération

La propriété `transition-timing-function` détermine la fluidité de l'animation. Les mots-clés admissibles pour cette propriété sont :

- `ease` : rapide sur le début et ralenti sur la fin ;
- `linear` (valeur par défaut) : vitesse constante sur toute la durée de l'animation ;
- `ease-in` : lent sur le début et accélère de plus en plus vers la fin ;
- `ease-out` : rapide sur le début et décélère sur la fin ;
- `ease-in-out` : le départ et la fin sont lents.

• Transition sur une ligne (raccourci)

Notation raccourcie : transition

La notation raccourcie `transition` permet de décrire facilement et de manière concise les différentes propriétés en jeu à l'aide d'une seule propriété. La syntaxe est la suivante :

```
selecteur {  
  transition: <transition-property> <transition-duration> <transition-timing-function>  
    ➡ <transition-delay>;  
}
```

En pratique, voici comment créer une transition d'une seconde de toutes les propriétés d'une image lors de son survol :

```
img {  
  transition: all 1s ease-in;  
}
```


CSS Animations

Définies comme une extension du module Transitions, les animations CSS apportent une dimension supplémentaire en introduisant le concept d'étapes et de nombre d'itérations.

Ce chapitre nécessiterait un ouvrage dédié et dépasse quelque peu le cadre de cet ouvrage « général » sur les techniques CSS avancées. En outre, ce module ne fonctionne actuellement que sur le moteur WebKit (ainsi que sur le tout récent Firefox 4), ce qui en réduit l'usage. Cependant, je ne résiste pas à la tentation de vous en montrer très brièvement le fonctionnement.

Le cœur des animations CSS réside dans les différentes images-clés (*keyframes*) qui sont autant d'étapes dans le cheminement de l'action. Dans un premier temps, donnons le nom de `disparition` à notre animation et définissons ses différentes étapes :

```
@-webkit-keyframes disparition {
  0% {opacity: 1;} /* opacité complète */
  50% {opacity: 0;} /* opacité nulle à la moitié de l'animation */
  100% {opacity: 1;} /* retour à l'état initial */
}
```

Il nous suffit à présent d'appliquer l'animation à l'élément souhaité, par exemple lors du survol des images :

```
img {
  -webkit-animation-name: disparition; /* nom de l'animation */
  -webkit-animation-duration: 2s; /* durée totale */
  -webkit-animation-iteration-count: infinite; /* nombre d'itérations */
}
```

Ce comportement d'animation en CSS est très attrayant, mais n'a pas encore atteint une maturité suffisante pour être véritablement exploité en production.