
Conception de pages web

Découverte de HTML 5 et CSS3

HTML



CSS



Cours et TD

Notions essentielles

Eric Willems – Luc Granier

2022-2023

PARTIE 1 : Le langage HTML5

Qu'est ce que le langage HTML5 ?

Le langage HTML5 est un **langage de description** à l'aide de **balises** de pages web qui peuvent être visualisées à l'aide d'un **navigateur**. Il affiche du contenu **statique**. Il a été lancé en 2009.

HTML signifie « HyperText Markup Language » traduisible par « langage de balises pour l'hypertexte ».

On trouvera notamment des balises :

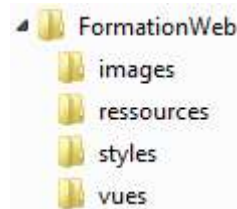
- De **description** de la page (en tête, corps, ...)
- De **mise en forme** des éléments de la page
- De gestion de l'interactivité par des **formulaires**

On découvrira dans ces pages les principales balises HTML5 et CSS3. Pour tout complément, se référer aux sites de référence :

- <https://www.w3schools.com/html> et [/css](https://www.w3schools.com/css)
- <https://developer.mozilla.org/fr/docs/Web/CSS> et [/html](https://developer.mozilla.org/fr/docs/Web/HTML)

Préparation de l'environnement

- Démarrer **WampServer** (ou lancer le navigateur et taper localhost :8080)
- Cliquer sur l'icône de WampServer pour afficher **www directory**
- Créer un dossier **FormationWeb** dans le dossier **www**, puis l'arborescence suivante :



- Démarrer l'éditeur **NotePad++**
- Dans l'onglet **Paramètres / Préférences / Auto complétion**, cliquer sur **activer la complétion automatique**
- Définir le langage pour activer la coloration : **Langage**, choisir **HTML**
- Définir l'encodage : **Encodage / Encoder en UTF-8**
- Ajouter le volet de navigation : **Affichage / Projet / Panneau de projet #1**
- Créer un projet : clic droit sur **Espace de travail / Ajouter un nouveau projet**
- Saisir le nom du projet : **FormationWeb**
- Associer le dossier FormationWeb au projet : **Clic droit sur FormationWeb / Ajouter les fichiers à partir d'un dossier**
- Ouvrir le dossier **FormationWeb** créé dans **C:\wamp\www**

Découverte de HTML5 et CSS3

Conception de pages web

Premier document HTML5

- Dans un **nouveau document**, saisir le code suivant :

```
<!DOCTYPE html>
<!-- Ceci est un commentaire -->
<html lang="fr">
  <head>
    <meta charset="utf-8" />
    <title> Découverte du HTML5 </title>
  </head>
  <body>
    <h1>Bonjour à tous</h1>
    <p> Bienvenue sur la page du <strong> HTML5 </strong></p>
  </body>
</html>
```

- L'enregistrer dans la racine (**FormationWeb**) sous le nom : **exercicesWeb.html**
- Cliquer sur l'icône de **WampServer** pour afficher **Localhost** (ou lancer le navigateur et taper localhost :8080)
- Ouvrir le fichier **exercicesWeb.html** pour le visualiser dans le navigateur

<http://localhost/FormationWeb/exercicesWeb.html>

Remarque : Si vous ne pouvez pas utiliser WAMP, il suffit d'aller dans le répertoire c:\WAMP\FormationWeb et de double cliquer sur votre fichier .html

Explication de la structure d'une page HTML5

Un document HTML5 est structuré en plusieurs parties :

L'entête DOCTYPE

Le **DOCTYPE** ou Document Type est le premier élément de l'entête. Il déclare au navigateur la norme utilisée. Il doit être placé en tout début de la page.

Les balises head

Elle contient des **méta-données** (titre du document, description, mots clés) utilisées par les moteurs de recherche. Le titre (**title**) s'affiche tout en haut dans le navigateur et dans le résultat de recherche sur Google, par exemple.

D'autres éléments peuvent apparaître dans le head :

- Des **références externes** vers des **feuille de styles CSS** :
`<link rel="stylesheet" type="text/css" media="screen" href="css/style.css" />`
- Des **références externes** vers des feuilles de **fichiers JavaScript**
`<script type="text/javascript" src="javascript/monscript.js"></script>`
- Des **références externes** pour afficher une petite **icone** pour les **favoris** du navigateur :
`<link rel="icon" type="image/png" href="images/favicon.png" />`
- Des **méta tags** utilisés pour le référencement par les moteurs de recherche (description, keywords,...).

Le corps du document

Le corps du document est en général le contenu principal de la page.

Il peut être subdivisé en quatre parties différentes :

- Un **entête** commun à toutes les pages du site. Elle peut être une bannière, un logo, ou autre chose qui représente généralement le propriétaire du site
- Un **menu** : point d'accès aux différentes pages du site
- Un **contenu**
- Un **pied de page** : informations d'ordre général comme le copyright, les liens vers les pages d'informations, etc

Principaux concepts utilisés

- **Éléments** : indicatifs définissant la structure et le contenu des objets dans une page. Exemple : ``, `<h1>`, `<a>`...
- **Balises** : marque de début et de fin d'un élément. Exemple : `<p>... </p>`
- **Attributs** : instructions complémentaires pour les éléments

Test des navigateurs

Pour tester l'affichage du site web sur plusieurs versions des principaux navigateurs, utiliser notamment : <http://browsershots.org/>

Test du code avec Fiddle

Utiliser <https://jsfiddle.net/>

Fiddle est une plateforme collaborative permettant de s'entraîner à coder en **HTML**, **CSS** et **JavaScript**, avec vision en temps réel du résultat du code.

Validation W3C

Le **W3C** ou **World Wide Web Consortium**, est un organisme de standardisation des applications Web.

Cette adresse (<http://validator.w3.org>) permet de valider le code source des pages conformément à la norme pour garantir une compatibilité avec les différents navigateurs.

La bibliothèque JavaScript **Modernizr** permet de détecter les éléments HTML5 et CSS3 inconnus des navigateurs et de prévoir une solution de rechange.

Règles d'écriture

- Toute **balise ouvrante** doit obligatoirement être **fermée**.
- Les **balises** et les **attributs** doivent être écrits en **minuscule**.
- Les **valeurs des attributs** doivent être entre **guillemet** ou **apostrophe**.
- Chaque **attribut** doit avoir une **valeur**.
- Chaque élément doit être **imbriqué correctement**.
- Un **commentaire** est placé entre des balises **<!--** et **-->**.

Pour chaque exercice :

- 1. Tester le code avec la plateforme Fiddle***
- 2. Puis copier le code dans le body de la page exercicesWeb.html***
- 3. Enfin tester le résultat dans le navigateur***

Titres et sous titres

Les balises de titre `<h1>...</h1>`, `<h2>...</h2>` jusqu'à `<h6>...</h6>` permettent de structurer le plan d'un document.

→ Exercice n°1 : Titres et sous titres

```
<h1>Le langage HTML5</h1>
<h2>Titres et sous titres</h2>
<h2>Les paragraphes</h2>
<h2>Les listes</h2>
<h3>Listes non ordonnées</h3>
<h3>Listes numérotées</h3>
<h2>Les liens</h2>
<h2>Les images</h2>
<h2>Les tableaux</h2>
<h2>Les liens</h2>
<h2>Le formulaires</h2>
```

***Important** : Ne pas choisir une balise de titre en fonction de la taille. Il est impératif de bien structurer sa page en commençant par un titre de niveau 1 (`<h1>`), puis un titre de niveau 2 (`<h2>`), etc. Pour modifier la taille du texte, on utilisera du CSS.*

Créez un nouveau fichier : exercice2.html

Au fur et à mesure vous insérerez les divers exercices donnés dans le cours dans ce fichier et vous le testerez.

Paragraphes

Créer un paragraphe

Le contenu des paragraphes sera décrit entre des balises `<p>...</p>`.

→ Exercice n°2 : Création d'un paragraphe

```
<p>
HTML5 (Extensible HyperText Markup Language) est un langage de balisage servant à écrire des pages pour le
World Wide Web.
</p>
```

Saut de ligne dans un paragraphe

Si on veut juste sauter une ligne dans un paragraphe, sans pour autant changer de paragraphe, on utilise la balise `
`

→ Exercice n°3 : Saut de ligne

```
<p>
Toute balise ouvrante doit obligatoirement être fermée. <br>
Les balises et les attributs doivent être écrits en minuscule. <br>
Les valeurs des attributs doivent être entre guillemet ou apostrophe. <br>
Chaque attribut doit avoir une valeur. <br>
Chaque élément doit être imbriqué correctement. <br>
</p>
```

Découverte de HTML5 et CSS3

Conception de pages web

Séparer les paragraphes par des lignes

La balise `<hr/>` insère un trait horizontal entre des paragraphes.

→ **Exercice n°4** : Insertion d'une ligne `<hr/ >`

```
...pour le World Wide Web.</p>
<hr/ >
<p>Toute balise... </p >
```

La balise d'accentuation ``

La balise **em** (emphase) permet de mettre en emphase du texte. Il s'agit de marquer l'importance d'un texte.

→ **Exercice n°5** : La balise d'accentuation ``

```
HTML5 <em>Extensible HyperText Markup Language</em> est un langage de balisage servant à écrire des
pages pour le World Wide Web.
```

La balise d'accentuation forte ``

La balise **strong** met un accent plus fort que la balise **em**, pour interpeler vivement le lecteur.

→ **Exercice n°6** : La balise d'accentuation forte ``

```
<strong>HTML5</strong> Extensible HyperText Markup Language est un langage de balisage servant à écrire
des pages pour le <em>World Wide Web</em>.
```

La balise de surlignage `<mark>`

La balise `<mark>` permet de faire ressortir visuellement une portion de texte.

→ **Exercice n°7** : La balise de surlignage `<mark>`

```
<strong>HTML5 </strong> <mark>Extensible HyperText Markup Language</mark> est un langage de balisage
servant à écrire des pages pour le <em>World Wide Web</em>.
```

On pourra changer le style d'affichage en CSS (surligner, couleur, encadrer, etc.)

Les listes

Les listes non ordonnées

Une liste non ordonnée énumère des données sous forme de listes à puces, en utilisant la balise `.........`.

ul signifie *Unordored List* et elle délimite le bloc de liste. Pour distinguer chaque élément de la liste, on utilise **li** qui signifie list item.

→ **Exercice n°8** : Les listes non ordonnées

```
<p> Règles d'écriture du HTML5 </p>
<ul>
    <li>Toute balise ouvrante doit obligatoirement être fermée. </li>
    <li>Toute balise ouvrante doit obligatoirement être fermée. </li>
    ...
</ul>
```

Découverte de HTML5 et CSS3

Conception de pages web

Listes numérotées

Les listes numérotées (1, 2, ...) utilisent la balise **** : *Ordered List* à la place de la balise ****. Pour délimiter les items, on utilise toujours la balise ****.

→ Exercice n°9 : Listes numérotées

```
<p> Etapes de conception d'un site Web </p>
<ol>
    <li>Faire un plan du site</li>
    <li>Sélectionner des images</li>
    <li>Mettre en forme</li>
</ol>
```

Les liens

Faire un lien en HTML5

Le lien constitue une liaison entre une page et une autre.

La balise **<a>** **** inclut un attribut **href** pour spécifier la cible, c'est-à-dire l'URL de la page ou de la ressource vers laquelle sera dirigé le visiteur après son clic sur le lien.

L'attribut **title** ajoute une bulle d'info au survol sur le lien.

Les liens absolus

Dans un lien absolu, on précise entièrement le **chemin**. Souvent utilisé pour un lien vers un site http ou ftp.

→ Exercice n°10 : Les liens absolus

```
<a href="http://www.w3schools.com/" title="W3 schools">Référence HTML5</a>
```

NB : si le lien contient des « & » les remplacer par « & »

Les liens relatifs

Dans un lien relatif, le chemin part du **répertoire courant**. Il s'agit d'un lien relatif au répertoire dans lequel on se trouve. Il est souvent utile d'employer ce lien quand on fait référence à une page interne du site.

Créer dans le dossier **vues** un fichier nommé **presCSS.html** contenant ceci :

→ Exercice n°11 : Les liens relatifs

```
<h1> Le CSS </h1>
<p>
Les feuilles de style "Cascading Style Sheets" (CSS) permettent de gérer la mise en forme des pages d'un site
indépendamment de son contenu. <br />
Un style peut être appliqué à différentes parties de différentes pages. <br />
Ainsi en modifiant le style, tous les éléments auxquels il est appliqué sont modifiés.
</p>
```


Découverte de HTML5 et CSS3

Conception de pages web

Créer dans le fichier **exercice2.html** un lien vers cette page :

➔ **Exercice n°12** : Création d'un lien vers une page

```
<a href="vues/presCSS.html">Présentation du CSS</a>
```

Si le fichier à ouvrir se trouve dans un dossier parent, utiliser le `..` pour remonter dans les dossiers.

Créer dans le fichier **presCSS.html** un lien vers la page exercice2

Exemple d'écriture d'un menu :

```
<ul>

  <li><a href="index.html">Accueil</a></li>

  <li><a href="forum.html">Forum</a></li>

  <li><a href="contact.html">Contact</a></li>

</ul>
```

Les sous menus ont la même structure que les menus.

➔ **Exercice n°13** : Création d'un lien vers une autre page

```
<a href=" ../exercice2.html">Retour index (exercice2) du site</a>
```

Lien interne sur une même page

Une **ancre** est un point pour repérer une partie particulière de la page. L'ajout de l'attribut **id** à une balise permet de créer une ancre qui donne un nom, un identifiant unique à un lien.

En bas de la page exercice2, compléter la saisie pour ajouter une ancre :

➔ **Exercice n°14** : ajout d'une ancre

```
<p id="liens">Liens vers des sites de référence</p>
```

```
<a href="http://www.w3schools.com">Référence HTML5</a>
```

En haut de la page exercice2, ajouter un lien vers cette ancre :

➔ **Exercice n°15** : ajout d'un lien vers cette ancre

```
<h1 >Le CSS</h1>
<h1 ><a href="#liens">Liens</a></h1>
```

En bas de la page exercice2, créer un lien pour remonter tout en haut de la page.

Découverte de HTML5 et CSS3

Conception de pages web

→ **Exercice n°16 (en autonomie)** : créer un lien retour

NB : si l'ancre se trouve sur une autre page, il faut préciser le chemin.
Exemple : ``

Ouvrir un lien dans une fenêtre différente

L'attribut **target="_blank"** ouvre un lien dans une nouvelle fenêtre.
La norme W3C de HTML5 recommande de ne plus utiliser cette pratique.

Pour tester malgré tout le fonctionnement, modifier le lien déjà saisi :

→ **Exercice n°17** : Ouvrir un lien dans une fenêtre différente

```
<a href="http://www.w3schools.com" target="_blank">Référence HTML5</a>
```

Des solutions alternatives existent en langage **JavaScript**.

Envoyer un mail

Un clic sur un lien ouvre le client mail par défaut (Outlook par exemple) en renseignant l'adresse email du destinataire.

→ **Exercice n°18** : Envoyer un mail

```
<a href="mailto:adresse@domain.fr">Contactez-moi</a>
```

Cette méthode peut être exploitée par des programmes malveillants ou des robots de SPAM, pour récupérer l'adresse mail. Une solution consiste à mettre une image à la place du texte du lien. L'image affiche sous forme textuelle l'adresse email. Les robots ne peuvent pas interpréter une image pour récupérer le texte. Il existe aussi d'autres solutions réalisables en PHP.

Lien pour ouvrir un fichier PDF

Il est aussi possible de faire un lien vers des fichiers autres que le HTML notamment les fichiers **PDF** (lisible avec **Acrobat Reader**).

- Copier le fichier **css_avancees.pdf** dans le dossier **ressources**.
- Créer dans la page **presCSS** un lien vers ce fichier.

→ **Exercice n°19** : ouvrir un fichier PDF

```
<a href="../ressources/css_avancees.pdf">Le CSS 3</a>
```

L'expression comporte un chemin relatif ("`../ressources/css_avancees.pdf`").
De la même façon on peut provoquer le téléchargement des fichiers au format zip, docx... en utilisant les liens.

Découverte de HTML5 et CSS3

Conception de pages web

Les images

Le **poids** (en Ko, voire en Mo) de l'image et donc son **temps de chargement** sera plus ou moins élevé selon le **format** choisi et la **qualité** de l'image va changer.

Certains formats sont plus adaptés que d'autres selon l'image (photo, dessin, image animée...).

Toutes les images diffusées sur Internet ont un point commun : elles sont **compressées**.

Les principaux formats d'image :

- Le **format JPEG** (Joint Photographic Expert Group) : plus de 16 millions de couleurs différentes avec perte de qualité imperceptible sur une photo. La compression est maximale en jpg.
- Le **format PNG** (Portable Network Graphics) est adapté à la plupart des graphiques (hors photo). Le PNG peut être rendu transparent et il n'altère pas la qualité de l'image. Il remplace avantageusement le GIF. Le PNG existe en deux versions, en fonction du nombre de couleurs que doit comporter l'image : PNG 8 bits : 256 couleurs ; PNG 24 bits : 16 millions de couleurs.
- Le **format GIF** est limité à 256 couleurs et peut être animé.
- Le **format BITMAP** (*.bmp), non compressé, n'est pas adapté au Web.

NB : les noms d'images et de dossiers doivent être écrits en minuscule sans espace.

Afficher une image

La balise utilisée pour afficher une image est la balise ****. Cette balise ne nécessite pas une balise fermante. Elle est placée dans une balise **<p>...</p>**.

→ **Exercice n°20** : Afficher une image

```
<p></p>
```

La balise **img** peut avoir plusieurs attributs mais 2 d'entre eux sont obligatoires :

- **src** : Cet attribut indique l'emplacement de l'image que l'on veut insérer. On peut indiquer un **chemin relatif** ou **absolu**.
- **alt** : texte « alternatif » si l'image ne peut pas s'afficher.

Rendre une image cliquable

A la place du texte, l'image peut très bien servir de lien. Il suffit de mettre une image entre les balises **<a>**. L'insertion de l'image entre ces balises implique l'utilisation de la balise **img**.

Compléter le lien précédent et tester l'image cliquable/.

→ **Exercice n°21** : Rendre une image cliquable

```
<a href="http://www.w3schools.com">  
  <p></p>  
</a>
```

Découverte de HTML5 et CSS3

Conception de pages web

Pour faire une miniature cliquable :

```
<p>
  Vous souhaitez voir l'image dans sa taille d'origine ? Cliquez dessus !<br />
  <a href="images/france.jpg">
    
  </a>
</p>
```

Afficher une bulle sur l'image

Pour afficher une bulle d'aide sur une image, on utilise le même attribut que pour les liens.

Compléter le lien précédent et tester la balise **title**.

➔ **Exercice n°22** : Afficher une bulle sur l'image

```
<a href="http://www.w3schools.com">
  <p></p>
</a>
```

Les figures

Elles se distinguent des images car elles apportent une valeur sémantique (un sens) au contenu. Elles ne sont pas que décoratives.

On utilise la balise **<figure>** associée à la balise **<figcaption>** pour sa légende.

➔ **Exercice n°23** : Les figures

```
<figure>
  
  <figcaption>La carte de France</figcaption>
</figure>
```

PARTIE 2 : Le langage CSS3

Qu'est ce que le CSS ?

Le **CSS** (Cascading Style Sheets) assure la mise en forme et la mise en page du contenu du site.

Un **sélecteur** permet de cibler un ou plusieurs éléments HTML afin de leur appliquer un style particulier.

Le site **www.caniuse.com** propose des tables de compatibilité des fonctionnalités de HTML et CSS sur différentes versions de navigateurs.

Le code en langage CSS peut être implanté à trois endroits différents :

- Dans l'en-tête **<head>** du fichier HTML.
- Directement dans les balises du fichier HTML via un attribut style (méthode la moins recommandée).
- Dans un **fichier .css** (méthode la plus recommandée).

Appel d'un fichier CSS

- Créer et enregistrer dans le dossier **styles** un nouveau fichier nommé **mesStyles.css**.
- Créer et enregistrer dans le dossier **racine** un nouveau fichier **exerciceCSS1.html** structuré comme le fichier **exercicesWeb.html** (ou **index.html**).
- Ajouter dans le **head** de la page la ligne :

```
<link rel="stylesheet" href="..." />
```

- Compléter le paramètre **href** pour appeler le fichier mes **styles.css**.

Soit : `< link rel="stylesheet" href = "styles/mesStyles.css" />`

NB : activer la coloration du code dans Notepad++ : menu Langage / CSS /

Commentaire

Saisir les commentaires entre `/* ... */`.

Pour chaque exercice ci-dessous :

- 1. Tester le code CSS avec Fiddle (optionnel)**
- 2. Puis coller le code dans les pages `exerciceCSS1.html` et `mesStyles.css`**
- 3. Enfin, tester l'affichage dans le navigateur.**

Appliquer un style CSS à une balise HTML

➔ **Exercice n°1** : définition de styles appliqués aux balises HTML

Ce sont des **sélecteurs simples** de type **élément** pour lesquels il suffit simplement de reprendre le nom de l'élément HTML pour lui appliquer des styles.

Code CSS

```
/* style appliqué à la balise body */
body {
    background-color: yellow;
    font-family: "Arial";
}

/* style appliqué à un titre de premier niveau */
h1 { color: blue; }

/* style appliqué au paragraphe p */
p { color : green; }

/* style appliqué à l'emphase */
strong { color : blue; }
```

Code HTML

```
<h1>Exercices HTML5 et CSS3</h1>
<p>Ce texte doit être écrit en <strong>Arial</strong> et en vert sur fond jaune</p>
```

NB : La liste des couleurs est disponible à cette adresse :
https://www.w3schools.com/cssref/css_colors.asp

Pour appliquer un même style à plusieurs balises, remplacer les styles appliqués aux balises h1 et strong précédentes par :

```
h1, strong
{
    color: red;
}
```

Par la suite, nous allons étudier les sélecteurs **#id** et **.class** qui vont permettre de cibler un élément en particulier plutôt qu'un type d'élément.

Découverte de HTML5 et CSS3

Conception de pages web

Les classes CSS

Les classes permettent de définir plusieurs styles différents pour une même balise HTML.

➔ **Exercice n°2** : les noms des classes .vert et .rouge pour un paragraphe.

Code CSS

```
p.vert{
    color : green
}
```

```
p.rouge {
    color : red
}
```

Code HTML

```
<p class="vert"> Ce paragraphe est de couleur verte. </p>
<p class="rouge"> Ce paragraphe est de couleur rouge. </p>
```

Si on omet le nom de la balise (p) dans le sélecteur, le style sera utilisable par d'autres balises HTML.

➔ **Exercice n°3** : avec la classe .vert toute balise porteuse de cette classe sera coloré en vert.

Code CSS

```
.vert{ /* Création d'une classe appelée : vert */
    color:green
}
```

Code HTML

```
<h1 class="vert">Ce titre est coloré en vert</h1>
<p class="vert">Tout comme le titre ci-dessus, ce paragraphe est aussi de couleur verte.</p>
```

Découverte de HTML5 et CSS3

Conception de pages web

➔ **Exercice n°4** : création de 3 classes spécifiques selon la nature d'un texte à afficher

Code CSS

```
.remarque { /* sélecteur */
    font-size:small; /*propriété : valeur */
    text-align:center;
    color:blue;
}
```

```
.note {
    font-size: large;
    text-align:right;
    color:red;
}
```

```
.normal {
    font-size: medium;
}
```

Code HTML

```
<p class="remarque"> Ce paragraphe est une remarque. </p>
<p class="note"> Ce passage est à noter et à retenir. </p>
<p class="normal"> C'est un paragraphe normal. </p>
```

Les id en CSS

Contrairement aux classes, un **id** s'applique à un **objet unique**. Il n'est pas utilisable pour plusieurs éléments HTML.

On utilise généralement un **id** pour placer des blocs dans une page (menu droit, menu gauche, contenu principal, etc...). Il pourra être utilisé par le code JavaScript.

➔ **Exercice n°5** : création d'une div (bloc) associée à un style.

Code CSS

```
#blocInfo { /* Création d'un id appelé : blocInfo */

    background-color: white;
    color : blue ;
}
```

Pour appliquer le style à une balise div, on spécifie le nom de l'id comme valeur de l'attribut id :

Code HTML pour appliquer l'id appelé blocInfo à une div

```
<div id="blocInfo" >Un exemple d'id. <br/> La div contient 2 lignes.</div>
```


Groupement de sélecteurs

Pour grouper plusieurs sélecteurs, il faut les lister et séparer les éléments de la liste par une virgule.

➔ **Exercice n°6** : Groupement des titres, qui auront tous la couleur verte

Code CSS

```
h1,h2,h3 {  
    color: green;  
}
```

Code HTML

```
<h1>Un titre h1 en vert</h1>  
<h2>Un titre h2 en vert aussi</h2>  
<h6>Un titre h6 en vert aussi</h6>
```

On peut même grouper des sélecteurs de types différents.

➔ **Exercice n°7** : Groupement de sélecteurs de types différents.

Code CSS

```
h4, .paragraphe, #idspecial {  
    color : red;  
    font-weight : bold;  
}
```

Code HTML

```
<div id="idspecial">Un id avec le même style</div>  
<h4>Un titre avec le même style</h4>  
<p class="paragraphe">Un paragraphe avec le même style</p>
```

Découverte de HTML5 et CSS3

Conception de pages web

Un **élément HTML** peut avoir **plusieurs classes**. On peut ainsi combiner les styles.

→ **Exercice n°8** : Combinaison de styles par plusieurs classes.

Code CSS

```
.vert {  
    color:green;  
}
```

```
.centre{  
    text-align : center;  
}
```

```
.gras {  
    font-weight:bold;  
}
```

```
.fondjaune {  
    background-color:yellow;  
}
```

Code HTML

```
<h1 class="vert centre fondjaune">Application de 3 classes pour le titre </h1>  
<p class="vert gras">On n'applique que deux styles au paragraphe : .vert et .gras.</p>  
<p class="vert">Et là, c'est un paragraphe avec un seul style. Donc une seule classe : .vert.</p>
```

Règles de priorité et héritage

Priorité des styles

- Priorité entre **classes** et **sélecteurs de type** : Le style de classe est prioritaire par rapport au style standard de la balise p.
- Priorité entre **deux classes** de **même nom** : Si deux classes de même nom sont définies avec des propriétés différentes, c'est celle de la dernière classe définie qui sera prise en compte.
- Priorité entre les **classes** et les **id** : Le style de l'id qui est prioritaire sur la classe.
- Priorité entre les **styles internes** et la **feuille externe css** : Le style défini en interne qui est prioritaire par rapport au style externe. A éviter !

L'héritage

La notion d'héritage signifie que tout élément HTML enfant va hériter, « en cascades », des styles de ses parents.

Ainsi en appliquant une couleur à la balise **body**, toutes les balises incluses (h1, p, ...) posséderont par défaut cette même couleur.

Les éléments html de type block et de type inline

En HTML, tout élément est soit de **type « block » (bloc)**, soit de **type « inline » (en ligne)**.

Élément de type block

Un élément de type block va toujours commencer sur une nouvelle ligne et prendre toute la largeur disponible dans la page.

De plus, un élément de type block peut contenir d'autres éléments de type block ainsi que des éléments de type inline.

Les éléments HTML de type block les plus communs sont les suivants : **p**; **h1**, **h2**, etc.; **ol** et **ul**; **form**; **div**.

Élément de type inline

Au contraire des éléments de type block, les éléments de type inline ne vont pas commencer sur une nouvelle ligne mais s'insérer dans la ligne actuelle.

Les éléments de type inline prennent uniquement la largeur qui leur est nécessaire (c'est-à-dire la largeur de leur contenu).

Les éléments HTML de type inline les plus communs sont les suivants : **strong** et **em** ; **a**; **img**; **span**.

→ Créer un nouveau fichier **exerciceCSS2.html** identique fichier **exerciceCSS1.html** (ou index.html) et un nouveau fichier **mesStyles2.css**

La balise Div

La balise **div** associée à des styles est utilisée pour faire des blocs de structure ou boîte. L'élément HTML **div** est un élément de type **block**.

➔ **Exercice n°9** : Utilisation d'une Div

Création d'une div autour de deux paragraphes. Ensuite, application des styles CSS directement à la div.

Code CSS

```
.div-para{
  background-color: #88BB11;
  font-weight: bold;
}
```

Code HTML

```
<h1>Div et span</h1>

<div class="div-para">
  <p>L'élément p est un élément de type block</p>
  <p>Un <strong>autre</strong> paragraphe</p>
</div>
```

La balise Span

La balise **span** va servir de conteneur pour mettre en forme une portion de texte à l'intérieur d'un élément sans pouvoir la cibler de façon simple.

L'élément **span** sera souvent utilisé avec un attribut **class** afin de pouvoir le cibler plus facilement en CSS.

➔ **Exercice n°10** : Surligner en vert un mot dans une div

Code CSS

```
.boiteEncadre {
  border: solid 2px #990000 ; /* taille absolue en pixel(px), cm ou mm */
                             /* taille relative en pourcentage */
}

.stylespan {
  background-color:#00FF00;
  text-decoration: underline;
  font-weight: bold;
}
```

Code HTML

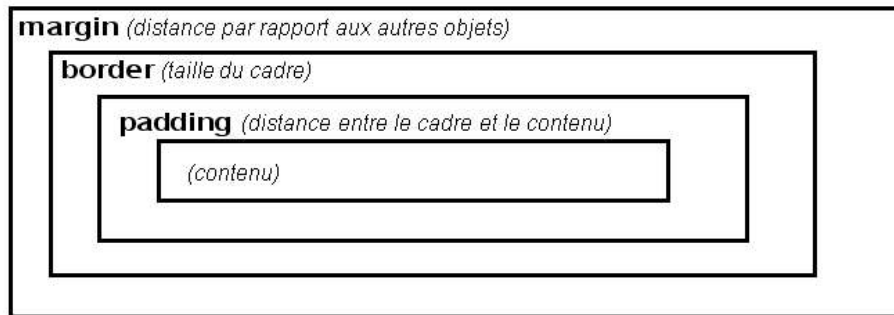
```
<div class="boiteEncadre">Ce <span class="stylespan">mot</span> sera surligné en vert, mis en gras et souligné pour marquer son importance.</div>
```

La gestion des boîtes

Les modèles de boîtes

Les modèles de boîtes sont un concept essentiel et central pour la mise en page d'une page web. Ils permettent de positionner les différents éléments d'une page.

Une boîte possède l'anatomie suivante :



→ Exercice n°11 : Création d'une boîte

Code CSS

```
div{
  background-color: #088;      /*Couleur de fond (bleu-vert)*/
  width: 300px;               /*Largeur de l'élément en soi*/
  padding: 20px;              /*Marge intérieure*/
  border: 20px solid #0C0;    /*Bordure (vert)*/
  margin: 50px;               /*Marge extérieure*/
}
```

Code HTML

```
<div>
  <h1>Le modèle des boîtes</h1>
  <p>Un premier paragraphe.</p>
  <p>Un autre paragraphe</p>
</div>
```

Affichage



Découverte de HTML5 et CSS3

Conception de pages web

La largeur et la hauteur

La **hauteur par défaut** d'un élément HTML est déterminée par la hauteur de son contenu.

La **largeur d'un élément**, en revanche, est avant tout déterminée par son type : les éléments de type **block** occuperont par défaut tout l'espace disponible. Les éléments de type **inline** n'occuperont que la largeur nécessaire à leur contenu.

La **hauteur** d'un élément est gérée par la propriété **height**, qui peut prendre des valeurs **absolues**, **relatives** (utiles pour un site responsive) ou la valeur **auto** (hauteur calculée automatiquement par le navigateur).

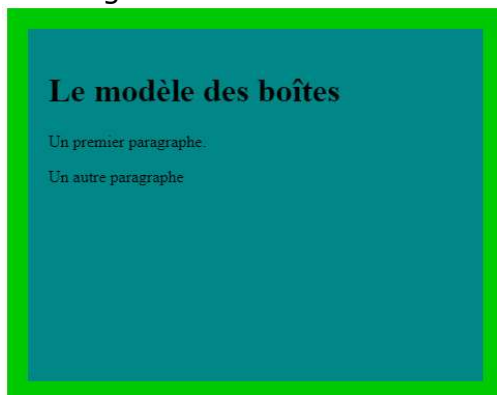
La **largeur** d'un élément est gérée la propriété **width**. Cette propriété peut prendre les mêmes valeurs que la propriété height.

➔ **Exercice n°12** : Gestion de la largeur et hauteur des boîtes

Code CSS (compléter le CSS précédent)

```
div{  
  ...  
  width: 400px;  
  height:300px;  
}
```

Affichage



Découverte de HTML5 et CSS3

Conception de pages web

Les bordures

La définition des bordures d'un élément utilise les propriétés **border**, **border-style**, **border-width** et **border-color**.

➔ **Exercice n°13** : Gestion des bordures

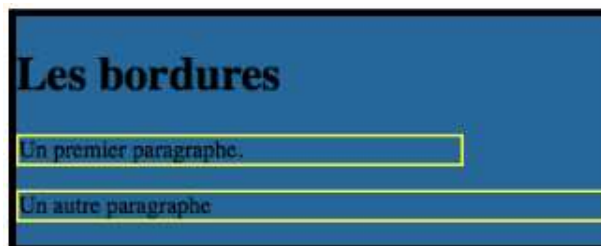
Code CSS

```
div{
  background-color: #269;
  width: 400px;
  border: 5px solid black;
}
.para1{
  /*Le contenu du paragraphe va prendre 75% de la largeur de son élément parent, c'est-à-dire du div*/
  width: 75%;
  border-width: 2px;
  border-style: solid;
  border-color: yellow;
}
.para2{
  border: 2px solid yellow;
}
```

Code HTML

```
<div>
  <h1>Les bordures</h1>
  <p class="para1">Un premier paragraphe.</p>
  <p class="para2">Un autre paragraphe</p>
</div>
```

Affichage



La propriété **border** nécessite de préciser dans l'ordre les valeurs associées à **border-width**, **border-style** et enfin **border-color**.

Découverte de HTML5 et CSS3

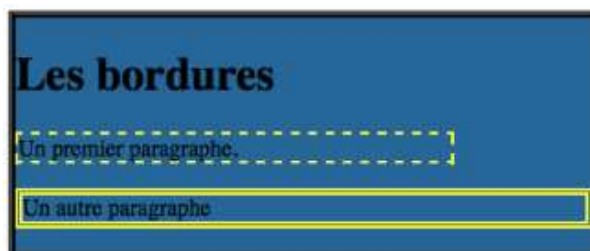
Conception de pages web

La propriété **border-style** peut prendre huit valeurs différentes :



➔ **Exercice n°14** : Gestion du style des boîtes

Remplacer le **style** de la bordure du premier paragraphe par **dashed** et observez le résultat.



Le CSS permet de définir des bordures différentes au-dessus (**top**), à droite (**right**), en dessous (**bottom**) et à gauche (**left**) des éléments HTML.

➔ **Exercice n°15** : Gestion des côtés des bordures

Code CSS

```
.para1{
  width: 75%;
  border-top-width: 2px;
  border-bottom-width: 10px;
  border-right-width: 5px;
  border-left-width: 5px;
  border-style: inset;
  border-color: yellow;
}
```

Affichage



Enfin, on peut créer des **bordures arrondies** avec la propriété **border-radius**. Cette propriété va prendre une valeur en pixels qui va correspondre à la valeur du radius de la bordure, et va s'appliquer avec la propriété border, ou avec les mots clefs **top-right**, **bottom-right**, **bottom-left**, **top-left**.

Découverte de HTML5 et CSS3

Conception de pages web

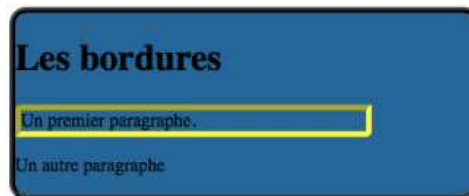
➔ Exercice n°16 : Gestion des côtés des bordures

Code CSS

```
div{
  background-color: #269;
  width: 400px;
  border: 5px ridge #444;
  border-radius: 15px;
}

.para1{
  width: 75%;
  border-width: 5px;
  border-style: inset;
  border-color: yellow;
  border-top-left-radius: 5px;
  border-bottom-right-radius: 5px;
}
```

Affichage



Les marges

Deux notions de marges :

- La **marge externe** (***margin***) définit l'espace entre le bloc et son extérieur
- La **marge interne** (***padding***) définit l'espace entre le bloc et son contenu

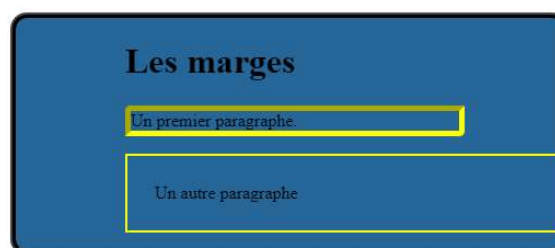
➔ Exercice n°17 : définir une marge intérieure

Code CSS

```
div{
  ...
  padding-left: 100px;
}

.para2{
  ...
  padding: 25px;
}
```

Affichage



➔ Exercice n°18 : définir une marge extérieure

Découverte de HTML5 et CSS3

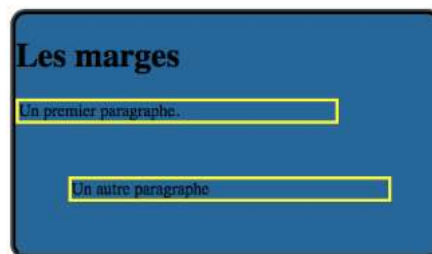
Conception de pages web

- La **valeur** de la marge (longueur ou % de la taille du conteneur ou auto selon le navigateur) est donnée pour les quatre côtés à la fois.
- Si on ne spécifie qu'une valeur dans la propriété **margin**, celle-ci s'applique à tous les côtés.

Code CSS

```
div{  
  ...  
  margin-top: 50px;  
  margin-left: 50px;  
}  
.para2{  
  ...  
  margin: 50px;  
}
```

Affichage



Remarque : on peut définir des marges intérieures et extérieures de tailles différentes pour chaque côté de nos éléments en utilisant directement les propriétés **padding** et **margin** et en spécifiant les quatre valeurs à la suite. Dans l'ordre : **marge haute, marge droite, marge basse, marge gauche**.

Code CSS

```
div{  
  background-color: #269;  
  width: 400px;  
  border: 5px ridge #444;  
  border-radius: 15px;  
  margin: 50px 0px 0px 20px;  
  padding: 0px 0px 0px 80px;  
}
```

Affichage



Centrer un bloc

Pour centrer horizontalement un élément par rapport à son élément parent, il faut :

- Utiliser la valeur auto
- L'élément doit être de type block
- L'élément doit posséder une largeur définie

➔ **Exercice n°19** : centrer une div

Code CSS

```
div{
  background-color: #269;
  width: 400px;
  border: 5px ridge #444;
  border-radius: 15px;
  margin: auto; /* au centre par rapport à l'élément parent */
}
```

Affichage



Définir une ombre à une boîte

On peut créer des ombres autour des boîtes (à l'extérieur ou à l'intérieur des bordures) grâce à la propriété CSS **box-shadow** qui va prendre quatre valeurs différentes dans l'ordre suivant :

- Le déplacement **horizontal** (vers la droite ou la gauche) de l'ombre
- Le déplacement **vertical** (vers le bas ou le haut) de l'ombre
- Le rayon de **propagation** de l'ombre
- La **couleur** de l'ombre

La valeur **inset** ajoutée à **box-shadow** permet de créer des ombres intérieures.

Découverte de HTML5 et CSS3

Conception de pages web

→ Exercice n°20 : ombrer une boîte

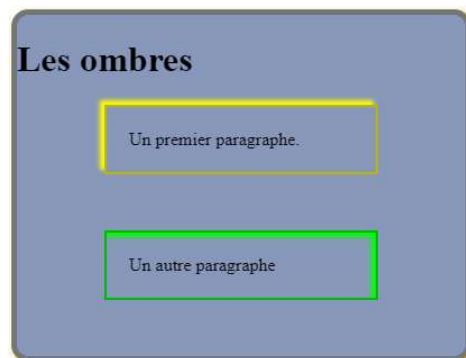
Code CSS

```
div{
  background-color: #89B;
  width: 400px;
  border: 5px solid #777;
  border-radius: 15px;
  /* Ombre orange autour de la div, projection de 3 pixels à droite et en bas pour cette 1ère ombre,
   et rayon de propagation de 9 pixels */
  box-shadow: 3px 3px 9px orange;
}

.para1{
  width: 50%;
  margin: 0px auto;
  padding: 20px;
  border: 2px solid #BB0;
  /* déplacements horizontal et vertical négatifs, ce qui projette l'ombre en haut à gauche
  plutôt qu'en bas à droite. */
  box-shadow: -4px -4px 5px yellow;
}

.para2{
  width: 50%;
  margin: 50px auto;
  padding: 20px;
  border: 2px solid #0B0;
  /* la valeur inset pour l'ombre du 2nd paragraphe crée une ombre intérieure plutôt qu'extérieure. */
  box-shadow: -4px 4px 5px lime inset;
}
```

Affichage



Le mode d’Affichage display

La propriété **display** peut prendre de nombreuses valeurs permettant de gérer précisément la façon dont chaque élément va être affiché.

Display : inline

Avec `display:inline`, l’élément ciblé va se comporter comme un élément de type inline et donc n’occuper que la largeur qui lui est strictement nécessaire.

Par exemple, en appliquant un **display:inline** à des éléments HTML **p**, ceux-ci vont venir se coller les uns à côté des autres selon la place disponible.

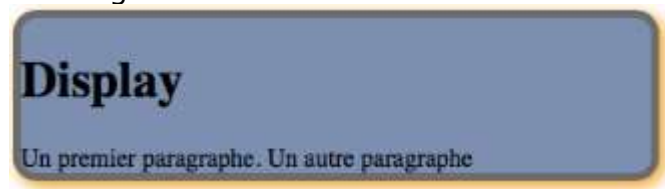
➔ **Exercice n°21** : affichage d’un bloc inline

Code CSS

```
div{
  background-color: #89B;
  width: 400px;
  border: 5px solid #777;
  border-radius: 15px;
  box-shadow: 3px 3px 9px orange;
}

.para1, .para2{
  display: inline;
}
```

Affichage



Display : block

Avec `display:block`, les éléments ciblés vont se comporter comme des éléments HTML de type block et ainsi prendre toute la largeur disponible.

On peut par exemple appliquer un **display:block** à un élément HTML **span** (qui est par défaut de type inline) afin d’observer la modification du comportement de celui-ci.

Découverte de HTML5 et CSS3

Conception de pages web

→ Exercice n°22 : affichage block

Code CSS

```
div{
  background-color: #89B;
  width: 400px;
  border: 5px solid #777;
  border-radius: 15px;
  box-shadow: 3px 3px 9px orange;
}

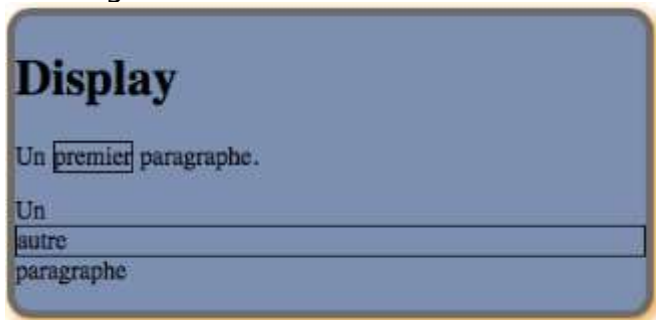
span{
  border: 1px solid black;
}

#sp2{
  display: block;
}
```

Code HTML

```
<div>
  <h1>Display</h1>
  <p class="para1">Un <span>premier</span> paragraphe.</p>
  <p class="para2">Un <span id="sp2">autre</span> paragraphe</p>
</div>
```

Affichage



Une bordure est appliquée aux deux éléments HTML **span** afin de voir la modification du comportement.
Ensuite, **display:block** est appliqué au deuxième élément **span**. Celui-ci se comporte alors comme un élément de type block et occupe toute la largeur disponible.

Display : inline-block

La valeur **inline-block** va permettre d'emprunter certaines propriétés des éléments de type **block** et certaines propriétés des éléments de type **inline**.

Ainsi, l'élément en soi va être de type inline tandis que ce qu'il contient (« l'intérieur de la boîte ») va être considéré comme étant de type block.

Cela permet notamment d'afficher plusieurs éléments côte-à-côte tout en maîtrisant précisément la taille de chacun d'entre eux.

Centrage de 3 div dans une div principale.

Découverte de HTML5 et CSS3

Conception de pages web

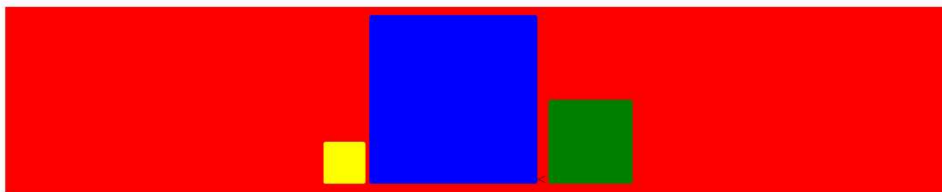
Code CSS

```
body{
    padding : 30px;
    background-color: #fcfcfc;
}
.outer-div{ /* extérieur */
    padding : 10px;
    text-align: center; /* centre le contenu dans le bloc
    background-color: red;
}
.inner-div1{ /* intérieur */
    display: inline-block; /* l'élément se met à coté */
    background-color: yellow;
    border-radius: 3px;
    width: 50px;
    height: 50px;
}
.inner-div2{
    display: inline-block;
    padding : 50px;
    background-color: blue;
    border-radius: 3px;
    width: 100px;
    height: 100px;
}
.inner-div3{
    display: inline-block;
    padding : 50px;
    background-color: green;
    border-radius: 3px;
}
}
```

Code HTML

```
<div class="outer-div">
    <div class="inner-div1"> </div>
    <div class="inner-div2"> </div>
    <div class="inner-div3"> </div>
</div>
```

Affichage



Découverte de HTML5 et CSS3

Conception de pages web

→ Exercice n°23 : Gestion des côtés des bordures

Code CSS

```
div{
  background-color: #89B;
  width: 400px;
  border: 5px solid #777;
  border-radius: 15px;
  box-shadow: 3px 3px 9px orange;
  /* text-align : center ; permet de centrer l'ensemble considéré comme du texte */
}

.para1{
  display: inline-block;
  border: 1px solid black;
  width: 55%;
}

.para2{
  display: inline-block;
  border: 1px solid black;
  width: 40%;
}
```

Code HTML

```
<div>
  <h1>Display</h1>
  <p class="para1">Un premier paragraphe.</p>
  <p class="para2">Un autre paragraphe</p>
</div>
```

Affichage



Display : none

La valeur none permet de ne pas afficher un élément.

Le positionnement

La propriété **position** permet de positionner les différents éléments HTML de façon absolue ou relativement par rapport à d'autres éléments HTML ou par rapport à leur place d'origine.

La propriété CSS position supporte quatre valeurs différentes :

- **static** : par défaut
- **relative**
- **fixed** (toujours à la même place en cas de scrolling)
- **absolute** (par rapport à son parent le plus proche)

La propriété **position** sera utilisée conjointement aux propriétés **top**, **right**, **bottom** et **left** afin de positionner précisément nos éléments.

Ces quatre propriétés vont pouvoir prendre des valeurs en pixels qui vont indiquer un déplacement d'un élément par rapport à sa position initiale sur un certain bord.

Position : static

La valeur **static** correspond au positionnement **par défaut** des éléments HTML dans une page. Aucun effet possible sur les positions

➔ **Exercice n°24** : Positionnement statique

Code CSS

```
div{
  background-color: #89B;
  width: 400px;
  border: 5px solid #777;
  border-radius: 15px;
}
.para1{
  position: static;
  top: 100px;
  left: 50px;
}
```

Affichage



Les propriétés **top**, **right**, **bottom** et **left** n'ont aucun effet sur un élément positionné avec **position:static**.

Découverte de HTML5 et CSS3

Conception de pages web

Position : relative

Un élément positionné grâce à **position:relative** va être repositionné relativement par rapport à sa position par défaut.

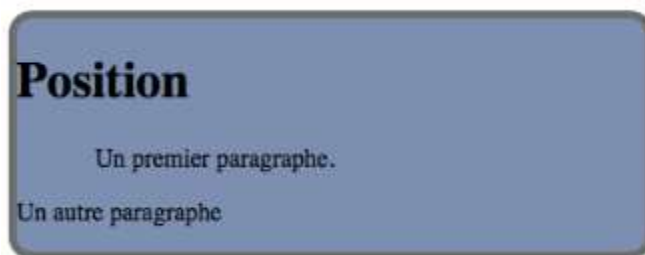
➔ **Exercice n°25** : Positionnement relatif

Code CSS

```
div{
  background-color: #89B;
  width: 400px;
  border: 5px solid #777;
  border-radius: 15px;
}

.para1{
  position: relative;
  left: 50px;
}
```

Affichage



Découverte de HTML5 et CSS3

Conception de pages web

Position : fixed

Un élément HTML possédant un positionnement fixed va toujours rester à la même place, même si l'un des visiteurs descend « scrolle » dans la page.

➔ **Exercice n°26** : Positionnement fixed

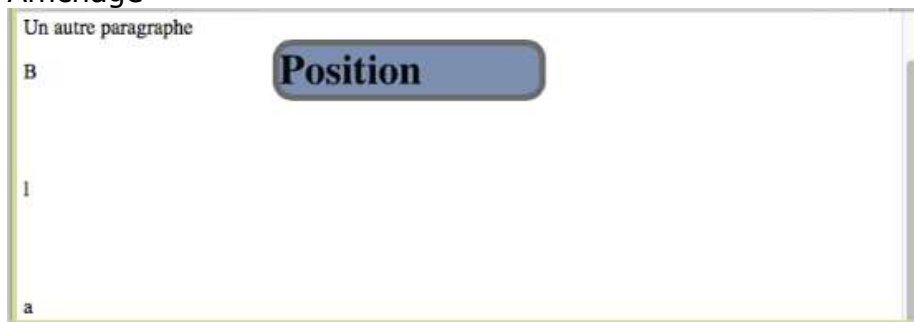
Code CSS

```
h1{
  background-color: #89B;
  width: 200px;
  border: 5px solid #777;
  border-radius: 15px;
  position: fixed;
  top: 0px;
  left: 200px;
}
```

Code HTML

```
<div>
  <h1>Position</h1>
  <p class="para1">Un premier paragraphe.</p>
  <p class="para2">Un autre paragraphe</p>
  <p>B<br><br><br><br>I<br><br><br><br>a<br><br><br><br></p>
  <p>B<br><br><br><br>I<br><br><br><br>a<br><br><br><br></p>
  <p>B<br><br><br><br>I<br><br><br><br>a<br><br><br><br></p>
</div>
```

Affichage



La **position:fixed** est appliquée au titre placé en haut de à 200 pixels vers la droite. Les espaces ajoutés dans la page permettent de tester la propriété CSS position. Lorsque l'on descend dans la page, le titre h1 conserve sa position.

Découverte de HTML5 et CSS3

Conception de pages web

Position : absolute

La valeur **absolute** de la propriété position positionne un élément de façon relative par rapport à son parent le plus proche auquel on a appliqué un positionnement spécifique (relative, fixed ou absolute). Ou à défaut par rapport à la page entière.

➔ **Exercice n°27** : Positionnement absolu

Code CSS

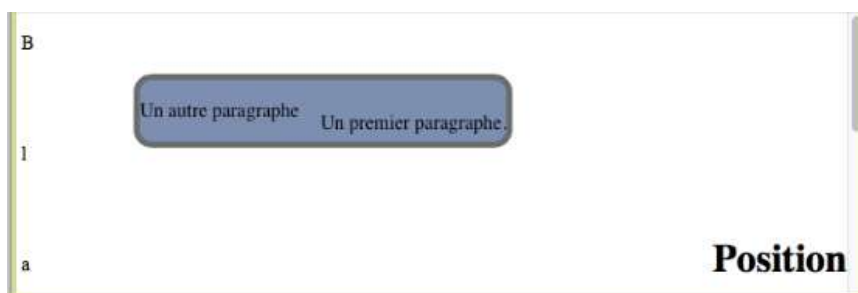
```
h1{
  position: absolute;
  bottom: 0px;
  right: 0px;
}

div{
  background-color: #89B;
  width: 300px;
  border: 5px solid #777;
  border-radius: 15px;
  position: fixed;
  top: 50px;
  left: 100px;
}

.para1{
  position: absolute;
  right: 0px;
  top: 10px;
}
```

Code HTML

```
<h1>Position</h1>
<!-- Attention, contrairement aux exemples précédents, h1 n'est plus à l'intérieur de notre élément div ! -->
<div>
  <p class="para1">Un premier paragraphe.</p>
  <p class="para2">Un autre paragraphe</p>
</div>
<p>B<br><br><br><br><br>I<br><br><br><br><br>a<br><br><br><br><br></p>
<p>B<br><br><br><br><br>I<br><br><br><br><br>a<br><br><br><br><br></p>
<p>B<br><br><br><br><br>I<br><br><br><br><br>a<br><br><br><br><br></p>
```



Découverte de HTML5 et CSS3

Conception de pages web

Le z-index

La propriété **z-index** définit l'ordre de superposition des éléments, par un nombre. Un nombre plus grand positionnera un élément devant un autre ayant un z-index plus petit.

→ **Exercice n°28** : Superposition de boîtes

Code CSS

```
h1{
  background-color: #AA2;
  position: absolute;
  top: 10px;
  left: 130px;
  z-index: 2; /* le numéro prioritaire est le numéro le plus élevé */
}

div{
  background-color: #89B;
  width: 300px;
  border: 5px solid #777;
  border-radius: 15px;
  position: fixed;
  top: 50px;
  left: 100px;
  z-index: 1;
}
```

Code HTML

```
<h1>Position et z-index</h1>

<div>
  <p class="para1">Un premier paragraphe.</p>
  <p class="para2">Un autre paragraphe</p>
</div>
```

Affichage



La propriété **z-index** va être particulièrement utile dans le cas d'un menu déroulant (afin que les rubriques s'affichent devant le reste de la page).

Flottement

La propriété : float

La propriété **float** va nous permettre de faire « flotter » des éléments HTML à gauche (**left**) ou à droite (**right**) dans une page web.

La propriété **float** va faire sortir un élément du flux normal de la page. Il y a trois choses à retenir lorsque l'on utilise cette propriété :

- Les éléments suivants un élément flottant vont se positionner à côté de celui-ci par défaut. La propriété **clear** annule ce comportement.
- Les éléments positionnés de façon absolue avec **position:absolute** ignoreront la propriété **float**
- Un élément flottera toujours dans les limites de la largeur de son élément parent conteneur.

➔ Exercice n°29 : Éléments flottants

Code HTML

```
<h1>Alignement & flottement</h1>

<div>
  <p class="para1">Un premier paragraphe.</p>
  <p class="para2">Un autre paragraphe</p>
</div>
```

Code CSS

```
h1{
  background-color: #AA2;
  width: 200px;
  float: left;
}
div{
  background-color: #89B;
  border: 5px solid #777;
  width: 200px;
  float: right;
}
.para1{
  float:right;
}
```

Affichage



L'élément **h1** flotte sur la gauche de la page. Ainsi, s'ils ont la place nécessaire, les éléments suivants l'élément h1 viendront se positionner par défaut à côté de celui-ci.

La **div** flotte sur la droite. Celui-ci se positionne donc bien par défaut sur la « même ligne » (imaginaire) que h1 mais à droite de notre page.

Finalement, on fait flotter le premier **paragraphe** à droite dans la **div**. Cela a pour effet de placer le 2nd deuxième paragraphe sur la même ligne que le premier et sur sa gauche.

La propriété : clear

La propriété **clear** permet d'empêcher des éléments de venir se positionner aux côtés d'éléments flottants.

Cette propriété peut prendre les valeurs suivantes :

- **Left** : neutralise l'effet d'un **float:left**
- **Right** : neutralise l'effet d'un **float:right**
- **Both** : neutralise l'effet d'un **float:left** et d'un **float:right**

Attention : La propriété **clear** doit d'appliquer à l'élément qui va venir flotter par défaut autour de l'élément flottant, pas à celui qui possède la propriété float.

Découverte de HTML5 et CSS3

Conception de pages web

→ Exercice n°30 : Propriété clear sur élément flottant

Code HTML

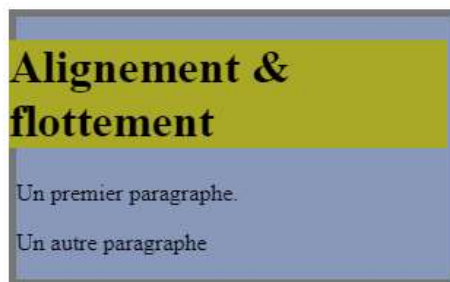
```
<h1>Alignement & flottement</h1>

<div>
  <p class="para1">Un premier paragraphe.</p>
  <p class="para2">Un autre paragraphe</p>
</div>
```

Code CSS

```
h1{
  background-color: #AA2;
  width: 300px;
  float: left;
}
div{
  background-color: #89B;
  border: 5px solid #777;
  width: 300px;
}
```

Affichage



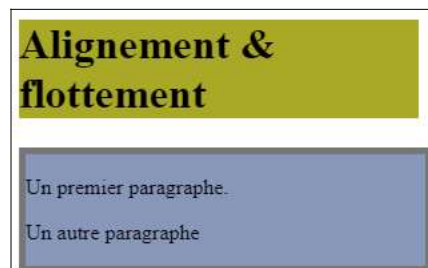
Le titre **h1** flotte sur la gauche. Par défaut, le **div** en dessous du **h1** va se placer sur la même ligne que le titre et donc autour de notre **h1**.

Pour éviter ce comportement, il faut spécifier un **clear:left** à notre élément **div**, afin de l'empêcher de venir flotter autour de notre **h1**.

Code CSS (ajouter le clear :left)

```
div{
  ...
  clear: left;
}
```

Affichage



Découverte de HTML5 et CSS3

Conception de pages web

Gestion du débordement (overflow)

La propriété float va sortir complètement un élément HTML du flux normal de la page web.

Ainsi, si un élément flottant est plus grand que son élément parent conteneur, celui-ci va déborder de son conteneur verticalement.

Afin d'éviter ce comportement, la propriété overflow va permettre de cacher (ou pas) ce qui va dépasser.

Cette propriété peut prendre les valeurs suivantes :

- **Visible** : valeur par défaut (rien ne sera coupé)
- **Hidden** : ce qui dépasse sera coupé
- **Scroll** : coupe de qui dépasse et ajoute une barre de défilement afin d'avoir accès à tout le contenu
- **Auto** : Laisse le navigateur décider du meilleur choix
- **Initial** et **inherit**

➔ **Exercice n°31** : L'overflow ou débordement

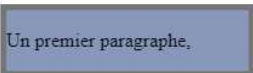
Code CSS

```
h1{
  background-color: #AA2;
  width: 300px;
}
div{
  background-color: #89B;
  border: 5px solid #777;
  width: 200px;
  height: 50px;
  overflow: hidden;
}
.para1, .para2{
  float: left;
}
```

Code HTML

```
<h1>Alignement & flottement</h1>
<div>
  <p class="para1">Un premier paragraphe, <br><br>
  qui va dépasser de notre div.</p>
  <p class="para2">Un autre paragraphe, <br><br>
  qui va également dépasser.</p>
</div>
```

Affichage



Tester l'affichage en modifiant la propriété **overflow** à **scroll**, puis à **visible**.

Alignement

Il existe 3 manières principales pour aligner ou centrer des éléments HTML dans une page :

- Avec la propriété **margin**
- Avec la propriété **position**
- Avec la propriété **float**

Centrer horizontalement avec margin

Il faut indiquer deux valeurs à la propriété margin :

- une valeur pour les marges haute et basse (au choix)
- la valeur auto pour les marges gauche et droite, ce qui va avoir pour effet de centrer notre élément HTML.

Attention : l'élément à centrer doit être de type block et posséder une largeur définie pour pouvoir être centré correctement.

Pour centrer le contenu de l'élément dans le bloc (par exemple le texte dans h1), il faut utiliser la propriété CSS **text-align**.

➔ **Exercice n°32** : Centrage horizontal avec margin

Code CSS

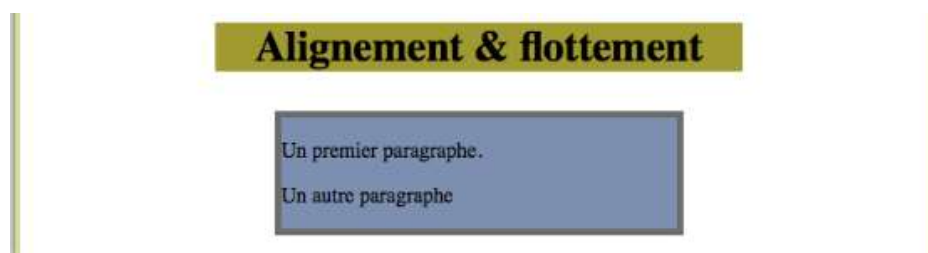
```
h1{
  background-color: #AA2;
  width: 400px;
  margin: 0px auto;
  text-align: center;
}

div{
  background-color: #89B;
  border: 5px solid #777;
  width: 300px;
  margin: 30px auto;
}
```

Code HTML

```
<h1>Alignement & flottement</h1>
<div>
  <p class="para1">Un premier paragraphe.</p>
  <p class="para2">Un autre paragraphe</p>
</div>
```

Affichage



Découverte de HTML5 et CSS3

Conception de pages web

Aligner horizontalement un élément HTML avec CSS position

On utilise la valeur **absolute** de la propriété position.

Il faut penser à effectuer un « reset CSS » sur la page, c'est-à-dire à définir des marges intérieures et extérieures du body égales à 0px afin d'éviter des différences d'affichage entre navigateurs.

Pour centrer horizontalement un élément en utilisant la propriété position, on peut également attribuer la même valeur, en pourcentage, aux propriétés left et right. Dans ce cas, il ne faut pas donner de largeur définie à l'élément HTML ciblé (celui-ci va se recadrer de lui-même selon la valeur attribuée à left et à right et la place disponible).

➔ Exercice n°33 : Centrage horizontal avec CSS position

Code CSS

```
h1{
  background-color: #AA2;
  position: absolute;
  left: 20%;
  right: 20%;
  text-align: center;
}

div{
  background-color: #89B;
  border: 5px solid #777;
  width: 300px;
  position: absolute;
  right: 10px;
  bottom: 10px;
}
```

Code HTML

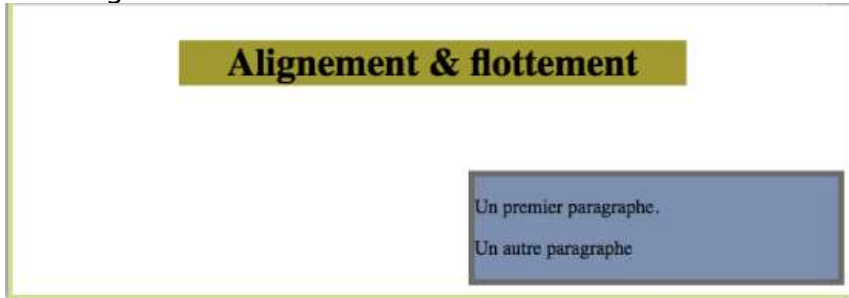
```
<h1>Alignement & amp; flottement</h1>

<div>
  <p class="para1">Un premier paragraphe.</p>
  <p class="para2">Un autre paragraphe</p>
</div>
```

Découverte de HTML5 et CSS3

Conception de pages web

Affichage



Aligner horizontalement en utilisant float

On peut aligner un élément sur la gauche ou la droite de la page en utilisant la propriété **float** et ses valeurs **left** et **right**. Il faut effectuer un reset CSS des marges intérieures et extérieures sur l'élément body.

➔ **Exercice n°34** : Centrage horizontal avec float

Code HTML

```
<h1>Alignement & flottement</h1>

<div>
  <p class="para1">Un premier paragraphe.</p>
  <p class="para2">Un autre paragraphe</p>
</div>
```

Code CSS

```
h1{
  background-color: #AA2;
  width: 200px;
  float: left;
}

div{
  background-color: #89B;
  border: 5px solid #777;
  width: 200px;
  float: right;
}

.para1{
  float:right;
}
```

Affichage



Découverte de HTML5 et CSS3

Conception de pages web

Les types de balises

Balises imbriquées

➔ **Exercice n°35** : Tester les types de balises

Code CSS

```
h3 em {color:red;} /* balises imbriquées, séparées par un espace */
em {color:blue;}
```

Les balises **** situées à l'intérieur d'une balise **<h3>** seront en rouge.
En dehors, elles seront en bleu.

Code HTML

```
<h3>Titre avec <em>textes importants</em></h3>
<em>textes importants hors h3</em>
```

Balises suivant une autre balise

Code CSS

```
h3 + p
{
    color:red;
}
```

Les balises **p** situées après un **h3** seront en rouge. Les autres resteront en vert comme défini dans le style **p** initial.

Code HTML

```
<p>Cette balise p précède le h3</p>
<h3>Titre de niveau 3</h3>
<p>Cette balise p suit le h3</p>
```

Balises avec un attribut

Code CSS

```
a[title] { color:red; }
```

Code HTML

```
<a href="http://www.google.fr" title="Moteur de recherche">Google</a>
```

Les liens hypertextes **a** possédant un attribut **title** seront en rouge.

Avec l'écriture :

```
a[title="Moteur de recherche"] { color:red; }
```

Seuls les liens hypertextes **a** possédant l'attribut **title** égal à "Moteur de recherche" seront en rouge.

Avec l'écriture :

```
a[title*="recherche"] { color:red; }
```

L'attribut **title** doit contenir mot « recherche » (peu importe sa position).

Découverte de HTML5 et CSS3

Conception de pages web

Balises de texte

➔ **Exercice n°36** : Tester les types de balises de texte

Taille du texte

Taille absolue :

font-size=12px

Taille relative (s'adapte au navigateur) :

font-size=valeur ;

Les valeurs possibles sont :

- **xx-small** : minuscule; **x-small** : très petit; **small** : petit; **medium** : moyen; **large** : grand; **x-large** : très grand; **xx-large** : gigantesque.
- x **em**. Exemple : 1em correspond à une taille normale. 1.5em, 0.8em...

D'autres unités existent : ex, %...

Police de caractère

font-family: Impact, "Arial Black", Arial, Verdana, sans-serif;

Utilisation de la police Impact ou, si elle n'y est pas, Arial Black, ou sinon Arial, ou sinon Verdana, ou à défaut une police standard (sans-serif)

NB : avec CSS, il est possible de télécharger sur le poste client une police spécifique, sous réserve des droits d'auteur et du poids du fichier police.

Attribut de caractère (italique, gras, souligné)

font-style : valeur;

Les valeurs possibles sont : **italic**, **oblique**, **normal**

font-weight : valeurs ;

Les valeurs possibles sont : **bold** (pour gras), **normal**

text-decoration : valeur ;

Les valeurs possibles sont **underline** (souligné), **line-through** (barré), **overline** (ligne au-dessus), **blink** (clignotant), **none** (normal par défaut).

Alignement du texte

text-align : valeur ;

Les valeurs possibles sont **left** (aligné à gauche, par défaut), **center** (centré), **right** (aligné à droite), **justify** (justifié).

Découverte de HTML5 et CSS3

Conception de pages web

Couleur du texte

Elle est appliquée par la balise **color**.

Code CSS

```
h1 { color : maroon; }
h2 { color : #000000; }
p { color: rgb(240,96,204); }
```

Les couleurs peuvent être exprimées par leur nom, en hexadécimal ou en RGB. Ainsi, #000000 correspond à la couleur noire et #FFFFFF à la couleur blanche. Une liste des couleurs est disponible à :

http://www.w3schools.com/html/html_colornames.asp

Code HTML

```
<h1>Les couleurs</h1>
<h2>Plusieurs formats d'écriture du code de la couleur</h2>
<p> Par leur nom, en hexadécimal ou en RGB. </p>
```

Image de fond

Elle est appliquée par la balise **background-image :url("nomImage.png")**.

➔ **Exercice n°37** : Ajouter une image de fond

Code CSS

```
body { background-image: url("../ressources/fond.jpg"); }
```

Pour fixer le fond, utiliser la propriété **background-attachment**.

Le texte glisse alors par-dessus le fond.

Deux valeurs sont disponibles :

- **fixed** : l'image de fond reste fixe ;
- **scroll** : l'image de fond défile avec le texte (par défaut).

Pour définir la répétition du fond, utiliser la propriété **background-repeat**. Par défaut, l'image de fond est répétée en mosaïque.

Autre propriétés :

- **no-repeat** : le fond ne sera pas répété. L'image sera donc unique sur la page.
- **repeat-x** : le fond sera répété uniquement sur la première ligne, horizontalement.
- **repeat-y** : le fond sera répété uniquement sur la première colonne, verticalement.
- **repeat** : le fond sera répété en mosaïque (par défaut).

Découverte de HTML5 et CSS3

Conception de pages web

Pour indiquer la position de l'image de fond, utiliser la propriété **background-position**.

La position peut être définie en pixels à partir du haut gauche, par exemple : **background-position: 30px 50px;**

Ou en utilisant des valeurs en anglais (qui peuvent être combinés) :

- **top** : en haut
- **bottom** : en bas
- **left** : à gauche
- **center** : centré
- **right** : à droite

➔ **Exercice n°38** : Ajouter une image fixe

Pour afficher une image fixe en haut à droite de l'écran :

Code CSS

```
background-image: url("../ressources/HTML5CSS3.jpg");
background-attachment: fixed; /* Le fond restera fixe */
background-repeat: no-repeat; /* Le fond ne sera pas répété */
background-position: top right; /* Le fond sera placé en haut à droite */
```

Pour affecter plusieurs images de fond, utiliser la propriété background.

Code CSS

```
body { background: url("../ressources/HTML5CSS3.jpg ") fixed no-repeat top right, url("../ressources/fond.jpg ") fixed; }
```

Opacité

La propriété **opacity : valeur** définit le niveau transparence des éléments. La valeur varie entre 0 (transparent) et 1 (opaque).

Code CSS

```
.opacite{
background-color: black;
color: white;
opacity: 0.3;}
```

Code HTML

```
<p class="opacite">Ce texte a un niveau de transparence</p>
```

Les pseudo-formats

Ils modifient l'apparence au survol, lors du clic, lors du focus, lorsqu'un lien a été consulté.

L'information est ajoutée après le nom de la balise (ou de la classe) dans le CSS.

```
a:hover { }
```

- **:hover** : au survol
- **:active** : au moment du clic
- **:focus** : lorsque l'élément est sélectionné, après un clic
- **:visited** : lorsque le lien a été visité

➔ **Exercice n°39** : Créer un lien hypertexte et tester les pseudo-formats

Code CSS

```
/* Liens par défaut (non survolés) */
a {
  text-decoration: none;
  color: red;
  font-style: italic;}

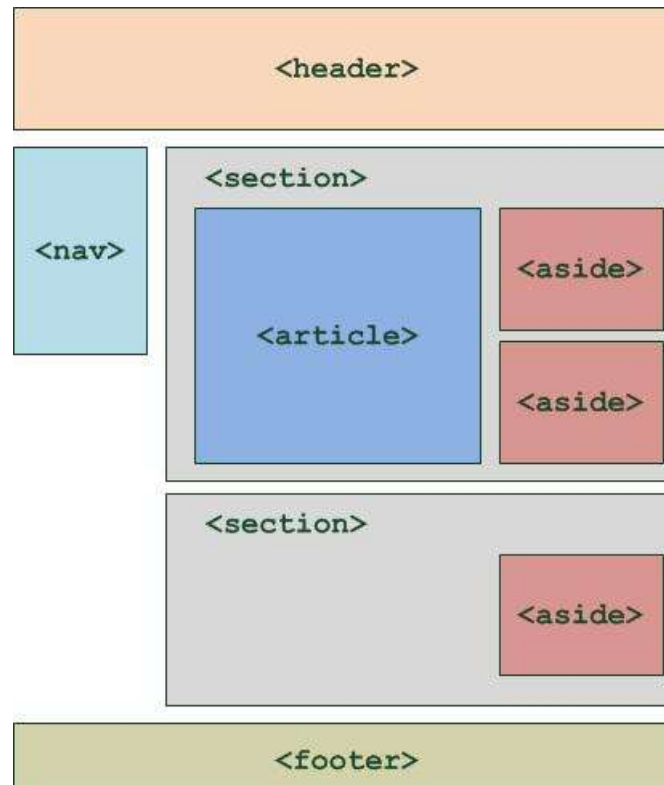
/* Apparence au survol des liens */
a:hover {
  text-decoration: underline;
  color: green;}

/* Quand le visiteur clique sur le lien */
a:active {
  background-color: #FFCC66;
}

/* Quand le visiteur sélectionne le lien */
a:focus {
  background-color: #FFCC66;
}

/* Quand le visiteur a déjà vu la page concernée */
a:visited {
  color: #AAA; /* Appliquer une couleur grise */
}
```

Les balises structurantes :



Sans utiliser la balise **footer** :

```
<div class="footer">
  alain.taulelle.free.fr - Tous droits réservés
</div>
<style type="text/css">
  .footer{
    color:green;
    background-color:red;
    text-align : center;
  }
</style>
```

En utilisant la balise **footer**

Découverte de HTML5 et CSS3

Conception de pages web

```
<footer>
    alain.taulelle.free.fr - Tous droits réservés
</footer>
<style type="text/css">
    footer{
        color:green;
        background-color:red;
        text-align : center;
    }
</style>
```

PARTIE 3 : Tableaux et formulaires

Les tableaux

Création d'un tableau en HTML5

Un tableau en HTML5 est créé à partir de 4 balises principales :

- La balise **<table>< /table>**. Elle délimite le début et la fin du tableau.
- L'attribut **<caption>** permet de mettre un titre sur le tableau.
- La balise **<tr></tr>** permet de définir une ligne.
- La balise **<td></td>** correspond aux cellules en colonne.
- La balise **<th></th>** s'utilise juste après la balise table pour les cellules de titre des colonnes du tableau.

➔ **Exercice n°1** : Créer le tableau ci-dessous à 2 colonnes et 4 lignes.

Balise	Fonction
table	Création d'un tableau
th	Cellule d'en-tête
...	...

Tableau HTML

Code HTML

```
<table border=1>
<caption>Tableau HTML</caption>
<tr>
    <th>Balise</th>
    <th>Fonction</th>
</tr>
<tr>
    <td>table</td>
    <td>Création d'un tableau</td>
</tr>
<tr>
    <td>th</td>
    <td>Cellule d'en-tête</td>
</tr>
<tr>
    <td>...</td>
    <td>...</td>
</tr>
</table>
```

Découverte de HTML5 et CSS3

Conception de pages web

Bordures

➔ **Exercice n°2** : Gérer les bordures du tableau

Code CSS

```
table{
  border-collapse: collapse; /* Bordures du tableau collées */
}
th, td /* Toutes les cellules des tableaux... */
{
  border: 1px solid black; /* bordure de 1px */
}
caption {caption-side : bottom; } /* top : titre placé au-dessus du tableau ; bottom : au-dessous */
```

Fusionner les colonnes d'un tableau

Pour fusionner plusieurs colonnes, on ajoute à l'attribut **td** ou **th** un attribut **colspan="n"** où n est le nombre de colonnes à fusionner.

➔ **Exercice n°3** : Fusionner une première ligne

Balises de création d'un tableau	
Balise	Fonction
table	Création d'un tableau
th	Cellule d'en-tête
...	...

```
<table>
<tr>
  <td colspan=2>Balises de création d'un tableau</td>
</tr>
<tr>
  <th>Balise</th>
  <th>Fonction</th>
</tr>
...
```

La première ligne s'étend sur 2 cellules.

Découverte de HTML5 et CSS3

Conception de pages web

Fusion de ligne

En appliquant **rowspan="n"** à une balise **td** ou **th**, on obtient les mêmes résultats sur les lignes. La première colonne s'étend sur 4 lignes.

Balise générales	Balise	Fonction
	table	Création d'un tableau
	th	Cellule d'en-tête

➔ **Exercice n°4** : Fusionner les lignes de la 1^{ère} colonne

```
<table border=1>
<tr>
  <td rowspan=4>Balise générales</td>
  <th>Balise</th>
  <th>Fonction</th>
</tr>
<tr>
  <td>table</td>
  <td>Création d'un tableau</td>
</tr>
<tr>
  <td>th</td>
  <td>Cellule d'en-tête</td>
</tr>
<tr>
  <td>...</td>
  <td>...</td>
</tr>
</table>
```

Les objets

La balise **<object>** définit un objet dans une page web html. Un objet peut être une animation flash, une applet java, un fichier audio...

Intégrer un objet multimedia

➔ **Exercice n°5** : Insérer un objet multimedia

```
<!DOCTYPE html>
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
  </head>
  <body>
    <h2> Ma France </h2>
    <audio src="Ma France.mp3" controls></audio>
    <hr />
    <br/><br/><br/>
    <h2> Loguivy de la mer </h2>
    <video src="LOGUIVY.3gp" width="200" height="200" controls></video>    </body>
</html>
```

➔ **Exercice n°6** : Réaliser une page similaire en utilisant des objets de votre choix.

Intégrer une carte (Google Map)

Ouvrir Google Map

Lorsque la carte souhaitée est choisie,

Cliquer sur le menu, dans l'angle supérieur gauche.

Puis cliquer sur Partager ou intégrer la carte

Puis cliquer sur Intégrer la carte dans la partie supérieure

Choisir la taille souhaitée

Copiez le code d'intégration HTML et collez le dans le code source de votre site à l'endroit que vous voulez.

Les Formulaires

Un **formulaire** permet à un utilisateur de saisir des données pour des recherches, une inscription, l'envoi de mails...

Le langage **HTML** est avant tout destiné à la présentation de documents et ne dispose d'aucun moyen de la modifier de manière dynamique.

Grace aux formulaires, HTML offre néanmoins un mécanisme de communication permettant de transmettre des informations de l'utilisateur vers le serveur. Ainsi HTML va permettre de saisir des données sur un navigateur et de les exporter vers le serveur.

HTML et dynamisme :

- **HTML** pour l'affichage de données.
- **JavaScript** pour un premier contrôle de la saisie
- **PHP** (ou **ASP** ou **script CGI**) au niveau du serveur pour le traitement de l'information.

HTML dispose de balises permettant de déléguer les traitements

- **<script>** et **</script>** pour un dynamisme coté client
- **< ? php >** et **< ? >** pour un dynamisme coté serveur

L'internaute valide sa saisie en cliquant sur un bouton de soumission (**submit**).

Les informations seront alors transmises à un **programme de traitement** (en **php**, par exemple).

Pour la suite, créez un nouveau fichier : form.html

Découverte de HTML5 et CSS3

Conception de pages web

Création de la structure du formulaire

Un formulaire est créé avec la balise **<form></form>**.

L'attribut **name** affecte un nom au formulaire.

L'attribut **method** détermine la méthode du formulaire :

- **get** : les données envoyées transitent par l'URL.
- **post** : les données envoyées ne sont pas visibles dans l'URL.

L'attribut **action** définit l'action qui doit être réalisée lors de la soumission du formulaire.

Par exemple, ouvrir un programme de traitement PHP qui va récupérer les données saisies et les envoyer dans une base de données ou à une adresse mail.

Pour cela il suffit d'écrire :

```
<form action="traForm.php" method="post" enctype="text/plain">
```

Pour le moment nous n'utiliserons pas les fichiers **.php**

➔ **Exercice n°7** : Création d'un formulaire

```
<form action="#" method="post" enctype="text/plain">
...
</form>
```

Toutes les instructions qui suivent seront saisies entre les balises **<form>** et **</form>**.

Saisie sur 1 ligne de texte

La balise **<input />** est utilisée pour saisir des données sur 1 ligne.

L'identification se fait par un attribut **name**, une valeur par défaut : **value** peut être donnée, ainsi qu'un certain nombre de caractères visibles en saisie : **size**, un nombre maximal de caractères : **maxlength** (Exemple : 5 pour un code postal).

```
Nom : <input type="text" name="nom" value="" size="10" />
```

Découverte de HTML5 et CSS3

Conception de pages web

Les principaux attributs possibles sont :

- **name** : nom unique de l'élément. Cet attribut est utilisé pour collecter le contenu de la ligne de texte lors de la soumission du formulaire.
- **size** : nombre de caractères visibles de la ligne de texte. La valeur par défaut de size est de 20. L'utilisateur peut néanmoins encoder autant de caractères qu'il souhaite.
- **maxlength** : nombre maximal de caractères que l'utilisateur peut encoder dans la ligne de texte. Ex : 5 pour un code postal
- **value** : valeur par défaut affichée sur la ligne de texte
- **readonly** : valeur par défaut affichée non modifiable par l'utilisateur.

De nouveaux attributs sont apparus en HTML5. Il faut vérifier leur compatibilité avec les navigateurs :

placeholder

Afficher une suggestion de saisie qui apparaît en grisé dans le champ de formulaire au chargement de la page. Ce texte disparaît dès que l'utilisateur donne le focus à l'élément concerné.

Nom :

Différence avec le **value** :

- Lorsque l'utilisateur commence à écrire dans le champ, le **placeholder** se vide.
- Avec **value**, en cas de soumission du formulaire sans modification de cette valeur par défaut, c'est celle-ci qui sera transmise.

autofocus (cible de la saisie)

Donne le focus à l'élément lors du chargement de la page.

Nom : Prénom : Age :

required

Rend la saisie de l'élément **obligatoire** par l'utilisateur.

Email (obligatoire) :

Au clic sur **Soumettre**, si l'Email n'a pas été renseigné on obtient :



Découverte de HTML5 et CSS3

Conception de pages web

pattern


Des patterns (ou masques) peuvent être créés par une expression régulière pour valider la saisie.

Exemple : saisie d'un code postal

Code postal :

- **pattern="[0-9]"** : la valeur de l'élément doit être comprise entre 0 et 9.
- **{5}** : 5 chiffres doivent être saisis.

Code postal :

 Veuillez respecter le format requis.

Le type **password** masque la saisie des caractères.

<label for="pass">Mot de passe : </label>
<input type="password" name="pass" size="8"/>

Ce formulaire de mot de passe ne protège en aucun cas les données car elles seront transmises en clair.

Saisie multi lignes

La balise **<textarea>** est utilisée pour saisir des données sur plusieurs lignes.

<label for="comment">Vos commentaires :</label>
<textarea name="comment" ></textarea>

Les attributs possibles sont :

- **name** : nom unique de l'élément. Cet attribut est utilisé pour collecter le contenu de la ligne de texte lors de la soumission du formulaire.
- **cols** : nombre de caractères visibles en largeur de la zone de texte **width**.
- **rows** : nombre de lignes visibles en hauteur de la zone de texte **height**.
- **readonly** : valeur attribuée par défaut à la ligne de texte ne pourra être modifiée par l'utilisateur.
- **autofocus** : donne le focus à l'élément lors du chargement de la page.
- **maxlength** : nombre maximal de caractères que l'utilisateur peut saisir
- **required** : saisie obligatoire.
- **wrap** : mode de gestion des retours automatiques à la ligne du texte lors de la soumission de la zone de texte. Avec **wrap="hard"**, un caractère de passage à la ligne est transmis avec le texte. Avec **wrap="soft"** (défaut), aucun caractère de changement de ligne n'est transmis.
- **placeholder** : affiche un texte dans le champ de formulaire au chargement de la page. Ce texte s'efface automatiquement dès que l'utilisateur donne le focus à la zone de texte

Les listes déroulantes

La balise **<select>** délimite la liste déroulante. Plusieurs choix sont proposés par des balises **<option>**.

```
<select name="titre">
  <option value="Mme" selected="selected">Madame</option>
  <option value="Mlle">Mademoiselle</option>
  <option value="Mr">Monsieur</option>
</select>
```

Les attributs possibles sont :

- **name** : nom pour la liste déroulante en vue d'un traitement ultérieur.
- **size** : par défaut égal à 1, cet attribut définit le nombre d'éléments visible du menu. Une barre de défilement permettra l'accès aux autres.
- **multiple** : sélection de plusieurs choix en maintenant la touche [Ctrl]
- **size** Cet attribut doit être égal au nombre des balises **<option>**
- **selected** : élément de la liste sélectionné par défaut
- **value** : l'élément choisi placé derrière **<option>** est transmis lors de l'envoi du formulaire. Une autre valeur (généralement numérique) peut être transmise avec l'attribut **value="valeur"**.

Les cases à cocher checkbox (choix multiple)

A l'inverse, les cases à cocher permettent à l'utilisateur de choisir plusieurs valeurs à la fois. Chaque **checkbox** possède le même attribut **name**.

```
<p> Quels sont vos centres d'intérêts ? <br />

<input type="checkbox" name="interet" id="musique" />
<label for="musique">musique</label>

<input type="checkbox" name="interet" id="politique" />
<label for="politique">politique</label>

<input type="checkbox" name="interet" id="informatique" />
<label for="informatique">informatique</label>

<input type="checkbox" name="interet" id="cuisine" />
<label for="cuisine">cuisine</label>
```

Les attributs possibles sont :

- **name="nom"** : obligatoire, des noms différents sont indispensables pour leur utilisation dans un script.
- **checked** : présélectionne une case à cocher.
- **value="valeur"** : En vue d'un traitement ultérieur, on attribue une valeur à chaque bouton checkbox.

Découverte de HTML5 et CSS3

Conception de pages web

Les boutons radios (choix unique)

Les boutons radios permettent de choisir une et une seule valeur.

```
<p>
Vous voulez vous être recontacté ?<br/>
<input type="radio" name="question" value="oui" id="oui" /> <label for="oui">oui</label>
<input type="radio" name="question" value="non" id="non" /> <label for="non">non</label>
</p>
```

Les attributs possibles sont :

- **name** : obligatoire, le nom doit être identique pour tous les boutons.
- **checked** : présélectionne un bouton radio.
- **value** : En vue d'un traitement ultérieur, on attribue une valeur à chaque bouton radio par l'attribut value="valeur".

Les boutons d'action (envoi et annulation)

Dans un formulaire, on lance un **bouton de soumission** du formulaire et un **bouton d'effacement** de la saisie.

```
<input type="submit" name="valider" value="Valider" />

<input type="reset" name="annuler" value="Annuler" />
```

La balise « datalist »

La balise **<datalist>** ouvre une liste de suggestions à la saisie des 1^{ères} lettres. L'utilisateur peut retenir une suggestion ou saisir une valeur de son choix.

```
Vos centres d'intérêt :
<input type="text" list="interet">
<datalist id="interet">
    <option value="Musique"></option>
    <option value="Cinéma"></option>
    <option value="Lecture"></option>
    <option value="Sport"></option>
</datalist>
```

La balise **<datalist>** est reliée à un champ de formulaire par un identifiant **id** qui renvoie à l'attribut **list** de celui-ci.

Découverte de HTML5 et CSS3

Conception de pages web

Organisation des champs de formulaire

- La balise **<fieldset> ... </fieldset>** englobe les champs de formulaires que vous déterminez. Ces champs sont alors visualisés à l'écran par une bordure.
- La balise **<legend> ... </legend>**, qui se place directement derrière la balise **<fieldset>**, ajoute une légende qui vient s'insérer dans la bordure dessinée par le **<fieldset>**.

Exemple

```
<fieldset >
  <legend>Vos centres d'intérêts</legend>

  <input type="checkbox" name="interet" id="musique" />
  <label for="musique">musique</label>

  <input type="checkbox" name="interet" id="cuisine" />
  <label for="cuisine">cuisine</label>
</fieldset>
<fieldset>
  <legend>Vos compétences</legend>

  <input type="checkbox" name="competences" id="informatique" />
  <label for="informatique">informatique</label>

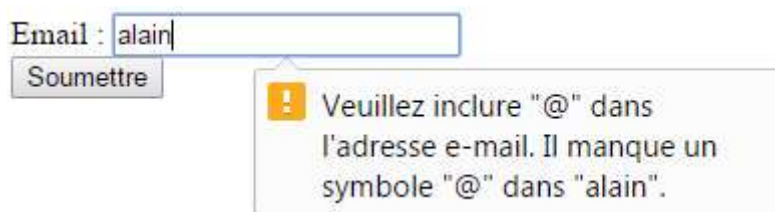
  <input type="checkbox" name="competences" id="gestion" />
  <label for="gestion">gestion</label>
</fieldset>
```

Adresse email valide

La balise **<input type="email">** contrôle la validité de la saisie de l'adresse mail.

```
<form action="">
  Email : <input type="email" name="mail" required><br>
  <input type="submit" value="Soumettre">
</form>
```

Exemple de saisie invalide et de message associé :



Avec l'attribut **required**, une autre infobulle signale que le champ doit être obligatoirement complété.

Découverte de HTML5 et CSS3

Conception de pages web

Saisie semi-automatique

L'attribut **"autocomplete=on|off"** est disponible dans la balise **form** ainsi que dans plusieurs balises de formulaires. Par défaut **"autocomplete=on"**

```
<form action="" autocomplete="on">  
  Email : <input type="email" name="mail" required><br>  
  <input type="submit" value="Soumettre">  
</form>
```

Saisie d'une URL

La balise `<input type="url">` est destinée à la saisie et au contrôle de validité des adresses Web (url).

```
<form action="">  
  Url de votre site : <input type="url" name="web">  
  <input type="submit" value="OK">  
</form>
```

Exemple de saisie erronée :



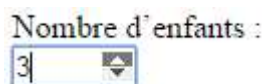
Les compteurs numériques

Cette nouvelle balise **<input type="number">** a des attributs spécifiques :

- **min** : valeur minimale du compteur.
- **max** : valeur maximale du compteur.
- **step** : pas d'avancement à chaque pression de la souris sur la flèche.
- **value** : valeur de départ du compteur.

Exemple

```
<form action="">  
  Nombre d'enfants : <br>  
  <input type="number" name="nbEnfant" min="0" max="20" step="1" value="0">  
</form>
```



Découverte de HTML5 et CSS3

Conception de pages web

Saisie des dates

Le Html5 propose la saisie de dates par un calendrier.

La balise **<input type="date">** sélectionne une date complète.

```
<form action="">  
    Date de naissance : <input type="date" name="dateNaissance">  
    Date d'expiration : <input type="month" name="dateExp">  
    Semainesouhaitée : <input type="week" name="semOk">  
</form>
```

Date de naissance :

septembre 2015

lun.	mar.	mer.	jeu.	ven.	sam.	dim.
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4

La balise **<input type="month">** sélectionne un mois et une année. Utile par exemple pour la date d'expiration d'une carte de crédit.

Date d'expiration :

septembre 2015

lun.	mar.	mer.	jeu.	ven.	sam.	dim.
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4

La balise **<input type="week">** sélectionne une semaine.

Semaine souhaitée :

septembre 2015

Semaine	lun.	mar.	mer.	jeu.	ven.	sam.	dim.
36	31	1	2	3	4	5	6
37	7	8	9	10	11	12	13
38	14	15	16	17	18	19	20
39	21	22	23	24	25	26	27
40	28	29	30	1	2	3	4

Découverte de HTML5 et CSS3

Conception de pages web

Saisie des heures

```
<form action="">
  Livraison de la pizza à :
  <input type="time" min="11:00" max="23:00" step="900" value="11:00">
</form>
```

Les attributs possibles sont :

- **min** : fixe une heure de début du compteur
- **max** : fixe une heure de fin
- **step** : fixe l'avancement en secondes à chaque pression sur la flèche.

A quelle heure êtes-vous disponible :

Curseur

```
<form action="">
  Quantités commandées<br>
  <input name="curseur" type="range">
</form>
```

La balise **<input type="range">** a des attributs spécifiques, identiques à ceux de **<input type="number">** : min, max, step, value.

Quantités commandées

Sélecteur de fichier

Afficher un champ de sélection de fichier qui permet de choisir un seul fichier :

```
<form action="">
  Select a file: <input type="file" name="myFile">
</form>
```

Découverte de HTML5 et CSS3

Conception de pages web

➔ Exercice n°8 : Création d'un formulaire d'inscription

En vous basant sur les différents éléments testés ci-dessus, créez un formulaire d'inscription à un site d'informations culturelles à partir de l'exemple ci-après :

Titre	<input type="text" value="Madame"/>
Nom	<input type="text"/>
Mot de passe :	<input type="password"/>
Vos commentaires :	<input type="text"/>
Quels sont vos centres d'intérêts ?	<input type="checkbox"/> musique <input type="checkbox"/> politique <input type="checkbox"/> informatique <input type="checkbox"/> cuisine
Vous voulez vous être recontacté ?	<input type="radio"/> oui <input type="radio"/> non
<input type="button" value="Valider"/> <input type="button" value="Annuler"/>	

Vous pouvez tester le bouton **Annuler** uniquement, qui vide les champs après saisie.

Vous pouvez le compléter et vous devez mettre notamment en œuvre les **fieldset** (avec CSS), les contrôles de saisie, les zones de saisie obligatoires.

Exemple de formulaire pour PPE0

FICHE DE CONTACT

Formulaire de contact

Coordonnées	
<input type="text" value="Madame"/>	
Nom (obligatoire):	<input type="text"/>
Prenom (obligatoire):	<input type="text"/>
Mail (obligatoire):	<input type="text"/> <input type="button" value="Soumettre"/>
Objet	
<input type="text" value="Recherche information"/>	
Quels sont vos centres d'intérêts ?	
<input type="checkbox"/> programmation <input type="checkbox"/> réseau <input type="checkbox"/> base de données <input type="checkbox"/> autre	
Vous voulez vous être recontacté ?	
<input type="radio"/> oui <input type="radio"/> non	
<input type="button" value="Valider"/> <input type="button" value="Annuler"/>	

Découverte de HTML5 et CSS3

Conception de pages web

Code HTML

```
<!DOCTYPE html>
<!-- Fiche de contact pour PPE0 -->
<html lang="fr">
  <head>
    <title>Formulaire de contact</title>
    <meta charset="UTF-8">

    <link rel="stylesheet" href="css/styles.css">
  </head>
  <body>
    <h1>FICHE DE CONTACT</h1>
    <form action="#" method="post" enctype="text/plain">
      <fieldset class="global">
        <legend>Formulaire de contact </legend>
        <fieldset class="coord">
          <legend>Coordonnées </legend>
          <select name="titre">
            <option value="Mme" selected="selected">Madame</option>
            <option value="Mlle">Mademoiselle</option>
            <option value="Mr">Monsieur</option>
          </select> <br/><br/>
          Nom (obligatoire): <input type="text" name="nom" required value="" size=15
        </br/><br/>
          Prenom (obligatoire): <input type="text" name="prenom" required value=""
        size="15" /><br/><br/>
          Mail (obligatoire): <input type="email" name="mail" required size="30"/>
          <input type="submit" value="Soumettre" />
          <br/>
        </fieldset>

        <fieldset class="objet">
          <legend>Objet </legend>
          <select name="titre">
            <option value="Recherche_info" selected="selected">Recherche
information</option>
            <option value="Recherche_presta">Recherche prestation</option>
            <option value="Contact_étudiant">Contact étudiant</option>
          </select> <br/><br/>
        </fieldset>
        <fieldset class="centre">
          <legend>Quels sont vos centres d'intérêts ? </legend>

          <input type="checkbox" name="programmation" id="programmation" />
          <label for="programmation">programmation </label>
          <input type="checkbox" name="reseau" id="reseau" />
          <label for="reseau">réseau </label>
          <input type="checkbox" name="bdd" id="bdd" />
          <label for="bdd">base de données </label>
          <input type="checkbox" name="autre" id="autre" />
          <label for="autre">autre </label>

        </fieldset>
        <fieldset class="contact">
          <legend> Vous voulez vous être recontacté ?</legend>
          <input type="radio" name="question" value="oui" id="oui" />
          <label for="oui">oui</label>
          <input type="radio" name="question" value="non" id="non" />
          <label for="non">non</label>
        </fieldset><br/>
          <input type="submit" name="valider" value="Valider" />
          <input type="reset" name="annuler" value="Annuler" />
        </fieldset>
      </form>
    </body>
```

Découverte de HTML5 et CSS3

Conception de pages web

Code CSS

```
/*style.css*/
h1{
    color: blue;
    position: absolute;
    left: 3cm;
    top :1cm
}
.global{
    position: absolute;
    top :3cm;
    right: 28cm;
    left: 1cm;
    width : 80%;
}
.coord{      width : 60%;    }
.objet{      width : 50%;    }
.centre{     width : 60%;    }
.contact{    width : 50%;    }
```

Autres balises

Horodatage

Les balises `<input type="datetime">` et `<input type="datetime-local">` combine l'affichage d'un calendrier et d'un compteur horaire.

La balise `<input type="color">` affiche une palette de couleur, et au choix d'une couleur, la valeur de celle-ci est reportée.

La balise `<cite>` pour les références

Lorsque l'on cite une œuvre, un livre, une référence, etc, on utilise la balise **cite**. Attention, il ne s'agit pas d'une citation. Pour cela, on utilise les balises `<q>` et `<blockquote>`.

Source du JO : <code><cite></code> RNCP JO du 19/12/2006. <code></cite></code>
--

La balise `<q>` pour citations courtes

On utilise la balise **q** pour mettre une citation courte. Le texte de citation est donc mis entre guillemets automatiquement quand on l'insère entre ces balises. Dans le cas des citations de plusieurs lignes, ou de plusieurs paragraphes, il est préférable d'utiliser la balise **blockquote**.

<code><q></code> Le chargé de projet informatique exerce dans une Société de Services en Ingénierie Informatique (SSII), ou au sein d'une entreprise utilisatrice. <code></q></code>
--

La balise `<blockquote>` pour citations de plusieurs lignes

La balise **blockquote** a le même sens que la balise **q** sauf que celle-ci est utilisée pour les extraits de plusieurs lignes.

<code><blockquote></code> La certification atteste que le titulaire a la capacité de maîtriser les compétences techniques, conceptuelles, humaines et personnelles des cinq blocs d'activités du référentiel : <code>
</code> A1- Organiser et conduire un projet informatique tout au long de son cycle de vie <code>
</code> A2- Analyser et concevoir un développement logiciel appliqué <code>
</code> A3- Administrer, maintenir et faire évoluer un système informatique <code>
</code> A4- Administrer et optimiser les schémas conceptuels liés au système d'information (base de données) <code>
</code> A5- Développer des applications multimédia <code></blockquote></code>

La balise `<acronym>`

La balise **acronym** sert à mettre un acronyme. Son utilisation n'a vraiment de sens que si l'on indique sa signification à l'aide de son attribut **title**.

Le <code><acronym title=" eXtensible HyperText Markup Language " lang="en"></code> HTML5 <code></acronym></code> est la fusion du <code><acronym title=" HyperText Markup Language " lang="en"></code> HTML <code></acronym></code> et du <code><acronym title=" eXtensible Markup Language " lang="en"></code> XML. <code></acronym></code>
--

Remarque : les mots entre **acronym** sont légèrement soulignés. En passant la souris sur les acronymes, leur signification apparaît automatiquement.

Découverte de HTML5 et CSS3

Conception de pages web

La balise <abbr>

La balise **abbr** est utilisée pour indiquer que le texte en question est une abréviation. De même que pour les acronymes, on indique la signification de l'abréviation en question dans l'attribut **title**. En pointant la souris sur les mots, leur signification apparaît automatiquement.

Les `<abbr title="clients">clts</abbr>` et les `<abbr title="fournisseurs">frns</abbr>` se sont mis d'accord.

La balise <dfn>

La balise **dfn** est utilisée pour dire que le texte est une définition d'un mot. Plus exactement la balise **dfn** montre le mot à définir. Attention, il existe les listes de définitions **<dl>** qui peuvent être utilisées lorsqu'il s'agit de définitions détachées du texte et non de sa continuité.

Le maki est un mammifère du sous-ordre des lémuriens vivant à Madagascar.
Source : Le `<dfn>maki</dfn>` est un mammifère du sous-ordre des lémuriens vivant à Madagascar.

La balise <sub>

La balise **<sub>** est utilisée pour mettre un texte en indice

Une suite géométrique de raison q est une suite de cette forme : $U_{n} = q * U_{0}$

La balise <sup>

La balise **<sup>** permet de mettre un texte en exposant.

Source : La dérivée de x^3 vaut $3x^2$.

Liste de définition

Les listes de définitions servent à indiquer un terme à définir et la définition de ce terme. Elles ont été conçues pour les entrées de glossaire. Pour autant, leur utilisation n'est pas limitée à ces dernières. Le principe est tout simplement de faire correspondre les termes aux définitions. La structure d'une liste de définitions est donnée par trois balises :

- La balise conteneur **<dl>** qui délimite la bloc de liste
- La balise **<dt>** ou **Definition Term** contiendra un terme de définition
- La balise **<dd>** ou **Definition Data** qui servira à la description de définition

```
<p>Voici une liste de termes et de leurs définitions :</p>
<dl>
  <dt>Développeur</dt>
  <dd>Un développeur est un informaticien qui programme des logiciels.</dd>
  <dt>Web</dt>
  <dd>Réseau maillé de serveurs d'informations formant une toile d'araignée.</dd>
</dl>
```

On peut mettre plusieurs **dd** pour un seul **dt**, et inversement. En effet, toutes les combinaisons sont réalisables.

Caractères spéciaux dans les URL

Soit l'URL : page.php?a=valeur1&b=valeur2. En HTML5, le & doivent s'écrire dans le code source &, ce qui donne un lien :

<code>Lien avec paramètre vers la page</code>
--

Caractères spéciaux :

à : à
è : è
é : é
ê : ê
ë : ë
ç : ç
ù : ù
espace :

Sommaire

PARTIE 1 : Le langage HTML5

Qu'est ce que le langage HTML5 ?	2
Préparation de l'environnement	2
Premier document HTML5	3
<i>L'entête DOCTYPE</i>	4
<i>Les balises head</i>	4
<i>Le corps du document</i>	4
Test des navigateurs	4
Test du code avec Fiddle	5
Validation W3C	5
Règles d'écriture	5
Titres et sous titres	6
Paragraphes	6
<i>Créer un paragraphe</i>	6
<i>Saut de ligne dans un paragraphe</i>	6
<i>Séparer les paragraphes par des lignes</i>	7
<i>La balise d'accentuation </i>	7
<i>La balise d'accentuation forte </i>	7
<i>La balise de surlignage <mark></i>	7
Les listes	7
<i>Les listes non ordonnées</i>	7
<i>Listes numérotées</i>	8
Les liens	8
<i>Faire un lien en HTML5</i>	8
<i>Les liens absolus</i>	8
<i>Les liens relatifs</i>	8
<i>Lien interne sur une même page</i>	9
<i>Ouvrir un lien dans une fenêtre différente</i>	10
<i>Envoyer un mail</i>	10
<i>Lien pour ouvrir un fichier PDF</i>	10
Les images	11
<i>Afficher une image</i>	11
<i>Rendre une image cliquable</i>	11
<i>Afficher une bulle sur l'image</i>	12
Les figures	12

PARTIE 2 : Le langage CSS3

Qu'est ce que le CSS ?	13
Appel d'un fichier CSS	13
Commentaire	13
Appliquer un style CSS à une balise HTML	14
Les classes CSS	15
Les id en CSS	16
Groupement de sélecteurs	17
Règles de priorité et héritage	19
<i>Priorité des styles</i>	19
<i>L'héritage</i>	19
Les éléments html de type block et de type inline	19
<i>Élément de type block</i>	19
<i>Élément de type inline</i>	19
La balise Div	20
La balise Span	20
La gestion des boîtes	21
<i>Les modèles de boîtes</i>	21
<i>La largeur et la hauteur</i>	22
<i>Les bordures</i>	23
<i>Les marges</i>	25
<i>Centrer un bloc</i>	27
<i>Définir une ombre à une boîte</i>	27
<i>Le mode d'affichage display</i>	29
Le positionnement	33
<i>Position : static</i>	33
<i>Position : relative</i>	34
<i>Position : fixed</i>	35
<i>Position : absolute</i>	36
<i>Le z-index</i>	37
Flottement	38
<i>La propriété : float</i>	38
<i>La propriété : clear</i>	39
<i>Gestion du débordement (overflow)</i>	41
Alignement	42
<i>Centrer horizontalement avec margin</i>	42
<i>Aligner horizontalement un élément HTML avec CSS position</i>	43
<i>Aligner horizontalement en utilisant float</i>	44
<i>Les types de balises</i>	45
<i>Balises imbriquées</i>	45
<i>Balises suivant une autre balise</i>	45
<i>Balises avec un attribut</i>	45
<i>Balises de texte</i>	46
<i>Taille du texte</i>	46
<i>Police de caractère</i>	46
<i>Attribut de caractère (italique, gras, souligné)</i>	46
<i>Alignement du texte</i>	46
<i>Couleur du texte</i>	47
<i>Image de fond</i>	47
<i>Opacité</i>	48
<i>Les pseudo-formats</i>	49
Les balises structurantes :	50

PARTIE 3 : Tableaux, objets et formulaires

Les tableaux.....	51
Création d'un tableau en HTML5.....	51
Bordures.....	52
Fusionner les colonnes d'un tableau.....	52
Fusion de ligne.....	53
Les objets.....	54
Intégrer un objet multimedia.....	54
Intégrer une carte (Google Map).....	54
Les Formulaires.....	55
Création de la structure du formulaire.....	56
Saisie sur 1 ligne de texte.....	56
Saisie multi lignes.....	58
Les listes déroulantes.....	59
Les cases à cocher checkbox (choix multiple).....	59
Les boutons radios (choix unique).....	60
Les boutons d'action (envoi et annulation).....	60
La balise « datalist ».....	60
Organisation des champs de formulaire.....	61
Adresse email valide.....	61
Saisie semi-automatique.....	62
Saisie d'une URL.....	62
Les compteurs numériques.....	62
Saisie des dates.....	63
Saisie des heures.....	64
Curseur.....	64
Sélecteur de fichier.....	64
Autres balises.....	68
Horodatage.....	68
La balise <cite> pour les références.....	68
La balise <q> pour citations courtes.....	68
La balise <blockquote> pour citations de plusieurs lignes.....	68
La balise <acronym>.....	68
La balise <abbr>.....	69
La balise <dfn>.....	69
La balise <sub>.....	69
La balise <sup>.....	69
Liste de définition.....	69
Caractères spéciaux dans les URL.....	70