

HTML Synthèse

Rappel : le langage HTML

Le HTML est un langage de balisage qui permet de structurer une page web et son contenu.

- Tous les éléments de la structure doivent être explicitement balisés (DOCTYPE, html, head, meta, title, body etc.)
- Toutes balises ouvertes doivent être fermées.
- Les balises auto-fermantes, vides ou orphelines s'écrivent `<balise (espace)/>`
- Tous les noms des balises et attributs doivent être écrits en minuscules
- Tous les attributs doivent avoir une valeur explicite
- Cette valeur doit être entourée de guillemets
- Les balises doivent être correctement imbriquées ex : `<p> emphase et renforcement </p>`
- Les attributs obligatoires doivent être spécifiés (ex : alt pour la balise img)

Quelques règles à respecter dans l'organisation du répertoire à la racine du site

- Le fichier d'ouverture du site doit être intitulé index.html
- Les noms de dossiers et fichiers sont écrits en minuscules sans espaces ni caractères accentués ou spéciaux

Les balises HTML

Éléments de type block vs éléments inline

Il existe deux catégories de balises en HTML, les balises de type « bloc » et les balises « en-ligne ». Le type de la balise influe sur le comportement, le positionnement et l'affichage des éléments; il est important d'en comprendre le fonctionnement pour l'intégration future des styles CSS.

Lorsqu'un navigateur édite une page, il parcourt le contenu du fichier HTML et en donne un rendu visuel selon les instructions de mise en forme CSS. Tous les éléments de la page (caractérisés par des balises) sont lus dans leur ordre d'apparition dans le flux HTML et sont positionnés dans des boîtes rangées les unes à la suite des autres à l'écran.

Les éléments se comportant comme des « blocs »

Outre les balises de la structure de la page, il s'agit des titres, des paragraphes, des listes, des tableaux ...

Les éléments de type bloc vont occuper tout l'espace disponible horizontalement.

Par défaut ils sont positionnés les uns en dessous de autres.

Les paragraphes et les titres ne peuvent pas contenir d'autres éléments de type « bloc ».

Les éléments « inline »

Les éléments en ligne se placent les uns derrière les autres au fil du texte, dans le « flux de la page ».

Ils sont destinés à être affichés dans le contexte de l'élément parent qui les contient et servent à

donner du sens à certaines parties du texte.

Il s'agit des éléments d'emphasis et renforcement , des liens <a> et des images ...

La taille d'un élément en ligne correspond à la taille de son contenu.

Par défaut, ils ont des marges internes et externes nulles.

Une propriété CSS - display et ses valeurs block et inline - permet de modifier cette appartenance à l'une ou l'autre catégorie.

Conformité aux standards du Word Wide Web Consortium

Pour être conforme aux standards du W3C, la page Web doit avoir été codée en HTML pour la structuration de la page et en CSS pour le rendu visuel.

Se conformer aux standards du W3C ne signifie pas que les pages web ainsi réalisées auront la même apparence sur tous les navigateurs. Les différences d'affichage sont dues aux différences d'interprétation et de restitution des navigateurs eux-mêmes, à la diversité des systèmes d'exploitation, aux dimensions et résolutions d'écran variables, aux spécificités physiques du matériel, aux modifications des paramètres de base apportées par les utilisateurs (taille de police, caractère ...).

Mise en conformité aux standards et validation auprès du W3C

Les outils du W3C - <http://validator.w3.org/> pour la validation du HTML et pour <http://jigsaw.w3.org/css-validator/> pour la validation du CSS (cf outil web developer) permettent de vérifier la conformité d'un code à la norme utilisée (DOCTYPE), de déceler les erreurs de syntaxe, les fautes de frappe sur un document soumis à leur analyse. Toutes les d'erreurs apparaissant lors de cette phase doivent être corrigées. Une fois la correction terminée le document est validé.

Pour vérifier si le code d'une page est valide (ce qui est différent de fonctionnel) suivant la norme indiquée dans la balise DOCTYPE, il suffit d'indiquer son url au validateur.

Rappel de la structure d'une page HTML

```
<!DOCTYPE html>
<html lang="fr-FR">
  <head>
    <meta charset="utf-8" />
    <title> </title>
  </head>
  <body>
  </body>
</html>
```

Notion d'héritage

L'héritage des CSS est fondé sur le modèle Parent-Enfant(s), par exemple une div ou un paragraphe situés dans le body, héritent de certaines propriétés définies pour le corps de texte (le type de caractère, la couleur, taille) ... ce qui présente un avantage évident : il n'est pas utile de répéter ces propriétés pour

chaque balise. Cela ne concerne pas toutes les propriétés, les bordures, les espaces ou marges ne font pas partie de l'héritage.

CSS, Cascading Style Sheets ou feuilles de style en cascade

I. Généralités

Un code HTML écrit de façon soignée, sémantiquement correct permet de profiter de toutes les potentialités des CSS pour la mise en forme des pages Web.

Les feuilles de style permettent de contrôler principalement deux aspects de la présentation d'un fichier HTML:

- La présentation des textes : taille, couleur, type de caractère, interligne, majuscules...
- La mise en page : gestion du positionnement des éléments de rendu « bloc » [en-tête, menu, navigation, centre, pied de page...], des marges, espacements...

A. Déclaration des feuilles de style CSS

Il existe trois différentes façons de déclarer les feuilles de style :

- a. directement dans la balise
- b. en interne dans l'en-tête du fichier HTML
- c. en externe dans un fichier.css indépendant

a. Le code CSS directement à l'intérieur des balises

Il est possible d'appliquer un style directement à une balise HTML à l'aide de l'**attribut « style »** suivi du bloc des propriétés et de leurs valeurs séparées par un point virgule. Ce codage est très proche du html, le style se répète dans chaque balise concernée. La structure et la présentation ne sont pas séparées. Cette méthode est à éviter, elle peut être utilisée lorsque le style ne concerne qu'un élément isolé dans les pages d'un site.

Ex : `<p style="color:#c00; font-family: verdana;"> flash info</p>`

b. Feuille de style interne, le code CSS dans l'en-tête du fichier HTML

Insérées dans l'en-tête de chaque page, les règles de style sont **encadrées par les balises <style> et </style>**.

Les fichiers HTML et CSS ne sont pas séparés. Une modification dans le style n'a d'effet que sur les éléments de la page, pas sur le site entier.

```
<head>
```

```
<style type="text/css" media="screen">
```

```
p {
```

```
color:#c00;
```

```
font-family: verdana;
}
</style>
</head>
```

L'attribut **type** est obligatoire. Il précise le type MIME [Multipurpose Internet Mail Extensions] c'est-à-dire le schéma qui permet au logiciel d'interprétation de savoir de quel type de données il s'agit. Un type MIME est composé d'un type de media et d'un sous type. Le type « text/CSS » est utilisé pour les fichiers texte des feuilles de style.

L'attribut **media** permet de préciser le type de media auquel la feuille de style est destinée : « screen » pour l'écran (la valeur par défaut), « print » pour l'imprimante, « all » pour tous types de medias.

c. *Externe, le code CSS dans un fichier externe*

Ecrire les règles CSS dans un fichier indépendant « fichier.css » semble être la meilleure solution. Le document HTML s'allège, le code devient plus lisible autant dans le fichier HTML que dans le fichier CSS.

Les fichiers CSS, stockés « en cache »* par les navigateurs ne sont téléchargés qu'une seule fois. Les modifications, ajouts ou suppressions dans le fichier CSS se répercutent sur l'ensemble des pages utilisant ce fichier, il est alors possible de modifier une propriété, ajouter une règle concernant toutes les pages du site en une seule fois.

Une page HTML est liée au fichier CSS par une balise <link> située dans l'en-tête du fichier HTML
`<link href="css/style.css" rel="stylesheet" type="text/css" media="screen" />`

L'attribut **href** indique l'emplacement du document auquel la balise s'applique

L'attribut **rel** définit la nature de la relation établie entre deux documents, permet de lier la feuille de style CSS au document HTML. Le navigateur sait ainsi qu'il s'agit des styles applicables au document et interprète les propriétés CSS contenues dans le fichier lié.

** Les mécanismes de mise en cache permettent de « dédier » une partie du disque dur de l'ordinateur au stockage des ressources déjà visionnées lors de précédentes visites. La mise en cache permet d'économiser des ressources machine, réseau et de réduire les temps d'attente.*

II. Syntaxe CSS

Une feuille de style est constituée d'un ensemble de règles de style rédigées selon la syntaxe CSS.

A. Une règle de style CSS

Une règle de style CSS est constituée de deux parties : un sélecteur suivi d'un bloc de déclarations

```
selecteur {  
    propriete1: valeur; /* chaque propriété doit avoir au moins une valeur */  
    propriete2: valeur;  
}
```

Le sélecteur va permettre d'identifier l'élément de la page web auquel les styles seront appliqués.

Chaque déclaration est composée d'une propriété qui indique au navigateur quels sont les éléments à modifier - et de sa valeur qui indique le changement à effectuer.

Le bloc déclaratif est entouré d'accolades → {} et chaque déclaration termine par un point-virgule → ;

Application à des éléments précis : les titres h1 du fichier html

Dans la partie body du fichier.html

```
<h1>L'automne</h1> <!-- la taille du titre définie par défaut (32px) est modifiée ainsi que la couleur et la police -->
```

Dans le fichier.css

```
h1 { /* règle de style pour la balise h1 */  
    font-family: arial; /* caractère arial */  
    font-size: 36px; /* taille 36px */  
    color: #c00; /* couleur rouge */  
}
```

Résultat :

L'automne

Tous les titres h1 seront écrits en rouge avec une police de caractère de type arial et une taille de 36px. Il est important de commenter les codes pour plus de clarté, surtout si vous travaillez à plusieurs sur les mêmes pages d'un site .

- Les commentaires HTML	→	<!-- commentaires HTML -->
- Les commentaires CSS	→	/* commentaires CSS */

Principaux sélecteurs pour appliquer les styles

Le sélecteur permet d'identifier la partie de la page web à laquelle les styles seront appliqués.

a. Sélecteur de type ou sélecteur de balise

```
h1 { /* définition d'une règle de style à appliquer aux balises h1 */  
  font-family: arial, sans-serif; /* caractère arial */  
  color: #c00; /* couleur rouge */  
}
```

C'est le sélecteur le plus simple, il reprend le **nom de la balise**. Les règles de style ainsi écrites vont s'appliquer à tous les titres h1.

```
h1, h2, p { /* définition d'une règle de style à appliquer aux trois balises h1, h2 et p */  
  font-family: arial, sans-serif; /* caractère arial */  
  color: #c00; /* couleur rouge */  
  font-size: 23px;  
}
```

/ remarque : On peut grouper les sélecteurs pour leur appliquer un même style. Pour cela, il faut juste les lister et séparer les éléments de la liste par une virgule ; Les règles de style ainsi écrites vont s'appliquer non seulement aux titres h1 mais également à tous les paragraphes p, et titres h2, sans distinction. Les éléments h1, h2, p ont été associés dans ce groupe pour les propriétés qu'ils partagent */*

Si l'on veut appliquer – par la suite - un style différent à un paragraphe ou un titre de la page, cibler de façon plus précise quelles balises sont concernées par la modification, on peut avoir recours à deux attributs qui fonctionnent avec toutes les balises, les attributs « **class** » et « **id** ».

b. Sélecteur de classe

L'attribut « **class** » dans le code HTML permet d'appliquer le style défini en CSS à un ou plusieurs éléments. Le sélecteur de classe est défini dans le code CSS, son écriture commence par un point (.)

Dans le fichier.html

```
<p class="automne">Les couleurs de l'automne</p>  
<p class="printemps"> Les couleurs du printemps</p>  
<h1 class="automne"> Les feuilles mortes ... </h1>  
<h2 class="printemps">Le printemps</h2>
```

Dans le fichier.css

```
.automne { /* sélecteur de classe */  
  color: red;  
  font-family: arial;  
}  
.printemps { /* sélecteur de classe */  
  color: green;
```

```
font-family: verdana;  
}
```

Résultat :

Les couleurs de l'automne

Les couleurs du printemps

Les feuilles mortes ...

Le printemps

Le premier paragraphe ainsi que le titre h1 adopteront la règle de style « .automne », ils seront écrits en rouge, avec un type de police de caractère « arial ». Le second paragraphe et le titre de niveau2 seront écrits selon la règle de style « .printemps » en vert avec une police de caractère « verdana »

Plusieurs classes peuvent être définies pour un même élément html. Pour appliquer plusieurs classes à un élément, il suffit de mettre comme valeur de l'attribut class la liste des noms des classes séparés par un espace.

c. Sélecteur d'identifiant unique

L'attribut « id » fonctionne de la même façon que l'attribut « class ». La seule différence est qu'il ne peut être utilisé qu'une seule fois dans un fichier donné. C'est un identifiant, sa valeur ne concerne qu'un seul élément. Le nom de cet identifiant doit obligatoirement commencer par une lettre suivi de chiffres ou de lettres. L'attribut « id » se positionne sur les éléments spécifiques, souvent pour créer des blocs → en-tête, conteneur, pied de page, menu, navigation ...

L'écriture d'un sélecteur d'identifiant dans le fichier.css commence par un dièse (#)

Dans le fichier.html

```
<div id="content">  
<h2>Titre</h2>  
<p>Contenu du paragraphe</p>  
</div>
```

Dans le fichier.css

```
#content { /* sélecteur d'identifiant unique */  
  background-color:#ccc;  
  color : #8db3e2;  
}
```

Résultat :

d. Sélecteur universel

Le sélecteur universel, noté (*) est utilisée pour appliquer les règles de style à l'ensemble des éléments d'un document. Par exemple pour supprimer les marges ou espacements définis par défaut de tous les éléments de rendu « bloc »

```
→ * {  
    margin: 0;  
    padding: 0;  
}
```

Peu utilisé car problématique avec d'anciennes versions de IE et macintosh.

B. Les balises génériques ou universelles et <div>

a. La balise : balise générique en ligne

La balise , balise « en-ligne » utilisée avec les attributs « class » et/ou « id », va permettre d'appliquer des règles de style à un mot ou groupe de mots « libre dans le flux » c'est-à-dire non encadré de balises.

Dans le fichier.html

```
<p>Peindre d'abord <span class="automne">une cage </span>peindre ensuite <span  
class="automne">quelque chose de joli </span></p>
```

Dans le fichier.css

```
.automne { /* sélecteur de classe */  
    color: red;  
    font-family: arial;  
}
```

Résultat :

Peindre d'abord **une cage** peindre ensuite **quelque chose de joli**

b. La balise <div>, balise générique de type bloc

<div> pourrait être traduit par « division » → diviser pour délimiter des zones, des regroupements d'objets.

En CSS la balise <div>, balise de type « bloc », est souvent utilisée pour effectuer la mise en page. De nombreuses divisions partagent l'espace en zones logiques : *la zone du haut – header avec l'en-tête, le logo ... , la navigation, le contenu dans la zone centrale, le pied de page ou footer...* Les blocs sont positionnés sur la page, juxtaposés, imbriqués ... pour les « visualiser » (même de façon provisoire) il suffit de les entourer de bordures ou de leur attribuer une couleur de fond.

Ex : Deux conteneurs identifiés par « bloc1 » et « bloc2 »

Dans le fichier. html

```
< div id="bloc1">
<p>Bloc de 30px de hauteur, avec une couleur de fond grise et une police de caractère de couleur
rouge</p>
</div>
<div id="bloc2">
<p>Bloc 2, 100px de hauteur, avec une couleur de fond rouge et une police de caractère grise</p>
</div>
```

Dans le fichier.css

```
#bloc1 { /* sélecteur d'identifiant */
  height : 30px;
  background-color:#ccc;
  color:#c00;
}
#bloc2 { /* sélecteur d'identifiant */
  height : 100px;
  background-color:#c00;
  color:#ccc;
}
```

Résultat :

Bloc1 30px de hauteur, avec une couleur de fond grise et une police de caractère de couleur rouge

Bloc2, 100px de hauteur, avec une couleur de fond rouge et une police de caractère grise

C. Liste des propriétés de formatage du texte

1. Propriétés de mise en forme des caractères → Les propriétés **font**

Les propriétés de mise en forme du texte vont permettre de définir le type de caractère, la taille, la couleur, le style, l'épaisseur et aussi la dimension de l'interlignage...

Les propriétés : **font-style** (italique), **font-variant** (petites majuscules), **font-weight** (caractère gras), **font-size/line-height** (taille police/dimension interligne), **font-family** (type de police de caractère)

Choix des polices de caractères → La propriété **font-family**

La police de caractère utilisée doit être reconnue par l'ordinateur de l'utilisateur. C'est la raison pour laquelle il est important de définir plusieurs noms de police d'une même famille générique (sans-serif, serif, monospace) séparés par une virgule. La première police déclarée est la police de référence, les suivantes sont des polices alternatives approchantes et la dernière est le nom de la famille générique.

```
body {
    font-family: Verdana, Arial, Helvetica, "Trebuchet MS", Tahoma, sans-serif ;
} /* sans empattement les extrémités des caractères sont droites */
body {
    font-family: Georgia, Times, "Times New Roman", Garamond, "Century
Schoolbook", serif ; /* à empattement les extrémités des caractères comportent de petits
crochets décoratifs */
} /* Un nom de police construit avec plusieurs mots entrecoupés d'espaces doit être
mis entre guillemets */
body {
    font-family: Courier, "Courier New", monospace ;
} /* à chasse fixe, la place occupée par un caractère est constante */
```

Rappel notion d'héritage :

Lorsque la police de caractère est appliquée à la balise <body> « élément parent », elle est systématiquement appliquée, par héritage, à ses descendants « éléments enfants ».

*Si l'on veut indiquer des caractères spécifiques à d'autres éléments de la page, un titre, un paragraphe ... il est possible de le faire par la suite à l'aide d'identifiant ou de classe, **la dernière déclaration prévaut sur les précédentes.***

Des exceptions à cet héritage : les marges, bordures, dimensions... ne peuvent « s'hériter ».

Choix de la taille des caractères → la propriété **font-size**

Les unités de dimension relatives - em, % - permettent d'établir la taille de la police en fonction de la taille de l'élément parent ou la taille par défaut si rien n'a été spécifié.

Les unités de mesure absolue – px - fixent la taille du texte sans tenir compte des autres paramètres. Les unités pouce (in), centimètre (cm), millimètre (mm), point (pt), pica (pc) ne sont pas utilisées sur le web. Le pt est une unité réservée à l'impression.

Tableau de correspondance des unités de dimension relatives. Les navigateurs fixent la taille du texte par défaut à 16px, ce qui équivaut à 1em ou 100%

Pixels	Ems	Percent
8px	0.5em	50%
9px	0.55em	55%
10px	0.625em	62.5%
11px	0.7em	70%
12px	0.75em	75%
13px	0.8em	80%
14px	0.875em	87.5%
15px	0.95em	95%
16px	1em	100%
17px	1.05em	105%
18px	1.125em	112.5%
19px	1.2em	120%
20px	1.25em	125%
21px	1.3em	130%

Pixels	Ems	Percent
21px	1.3em	130%
22px	1.4em	140%
23px	1.45em	145%
24px	1.5em	150%
26px	1.6em	160%
29px	1.8em	180%
32px	2em	200%
35px	2.2em	220%
36px	2.25em	225%
37px	2.3em	230%
38px	2.35em	235%
40px	2.45em	245%
42px	2.55em	255%
45px	2.75em	275%
48px	3em	300%

Choix de la couleur des caractères → la propriété **color**

Les valeurs attribuées à la propriété **color** peuvent s'exprimer par un nom de couleur ou sous une autre forme.

- 17 noms de couleurs sont validées par le W3C: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, *orange*, purple, red, silver, teal (turquoise), white et yellow. Transparent est également une valeur admise (c'est la valeur par défaut des couleurs de fond)

- La valeur RGB en notation hexadécimale - *combinaison de lettres et de chiffres* - est formée d'un caractère '#' immédiatement suivi par trois ou six caractères hexadécimaux. *Les 2 premiers indiquent une quantité de rouge, les 2 suivants une quantité de vert, et les 2 derniers une quantité de bleu. Toute couleur est issue de la combinaison des 3 couleurs de base : rouge, vert et bleu. En mélangeant une quantité de chacune de ces composantes on peut obtenir la couleur que l'on désire.*

- La valeur RGB en notation fonctionnelle est formée par les caractères « rgb » suivis par trois valeurs numériques (trois entiers *compris entre 0 et 255* - ou trois pourcentages – 0 à 100%).

HTML

<p> Peindre d'abord une cage </p>

CSS /* l'une des propriétés suivantes */

```
p {color:green ;}  
p {color: #009900 ;} /* #rrggbb notation hexadecimale */  
p {color: #090 ;} /* #rgb notation hexadecimale */  
p {color:rgb(0,153,0) ;} /* rgb notation fonctionnelle */
```

Résultat

Peindre d'abord une cage

Installer l'extension colorZilla de Mozilla Firefox.

Lors de votre navigation sur le web, il vous arrive de trouver une couleur intéressante. Grâce à cet outil, il suffit de placer le curseur de la souris sur la couleur, pour que le code couleur correspondant apparaisse, vous pouvez alors le récupérer.

Mise en gras → la propriété **font-weight**

La propriété **font-weight** → valeurs: **normal** ou **bold** ou bolder, lighter, 100 ... 900, etc.
Propriété héritée. La valeur « normal » est utilisée pour arrêter l'héritage.

HTML

<p>Peindre d'abord une cage</p>

CSS

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

Résultat

Peindre d'abord **une cage**

Mise en italique → la propriété **font-style**

La propriété **font-style** → valeurs: **normal** ou **italic** ou **oblique**
Propriété héritée. La valeur « normal » est utilisée pour arrêter l'héritage.

HTML

<p> Avec une porte ouverte, peindre ensuite quelque chose de joli</p>

CSS

```
.italic {  
    font-style:italic;  
}
```

Résultat

Avec une porte ouverte, *peindre ensuite* quelque chose de joli

Transformation du caractère en petites majuscules → la propriété **font-variant**

La propriété **font-variant** → valeurs: **normal**, **small-caps**.

Propriété héritée. La valeur « normal » est utilisée pour arrêter l'héritage.

HTML

<p> Quelque chose de simple</p>

CSS

```
p {  
    font-variant:small-caps;  
}
```

Résultat

QUELQUE CHOSE DE SIMPLE

Interligne → la propriété **line-height**

La propriété **line-height** → valeurs: **normal** ou **nombre positif** en px, em, %.

Propriété héritée. La valeur « normal » est utilisée pour arrêter l'héritage

```
Ex : p {  
    font-family: verdana,sans-serif;  
    font-size: 12px;  
    line-height: 180%  
}
```

Notion de propriété raccourcie

Une « propriété raccourcie » permet de définir en une seule déclaration, la mise en forme de plusieurs propriétés d'un élément.

Les propriétés raccourcies s'appliquent aux éléments « font », « border », « margin » et « padding » ou encore « background », « list-style »...

Propriété raccourcie → la propriété **font**

La **propriété raccourcie font** va permettre de spécifier en une déclaration les divers propriétés des caractères et la taille de l'interlignage

sélecteur {

font: 'valeur de font-style', 'valeur de font-variant', 'valeur de font-weight', 'valeur de font-size'/'valeur de line-height', 'valeur de font-family' ;
}

HTML

<p> Peindre d'abord une cage

avec une porte ouverte</p>

CSS

```
p {  
  font: italic small-caps bold 1em/2em verdana,sans-serif;  
}
```

Résultat

**PEINDRE D'ABORD UNE CAGE
AVEC UNE PORTE OUVERTE**

NB : pour que cette propriété raccourcie fonctionne il est nécessaire de préciser les valeurs de font-size et de font-family.

2. Autres propriétés de mise en forme du texte

Quelques autres propriétés relatives à la mise en forme du texte

Transformation du texte en majuscules ou minuscules → la propriété **text-transform**

La propriété **text-transform** → valeurs: **none**, **capitalize** (première lettre de chaque mot en majuscule), **lowercase** (tout le texte en minuscules), **uppercase** (tout en majuscules).

Propriété héritée. La valeur « none » est utilisée pour arrêter l'héritage.

HTML

```
<p> <span class="majitalic">Quelque chose </span> de beau</p>
```

CSS

```
.majitalic {  
    text-transform:uppercase;  
    font-style:italic ;  
}
```

Résultat

QUELQUE CHOSE de beau

Décoration d'un texte → la propriété **text-decoration**

La propriété **text-decoration** → valeurs: **none**, **underline** (souligné), **overline** (surligné), **blink** (clignotant), **line-through** (barré).

Propriété héritée. La valeur « none » est utilisée pour arrêter l'héritage.

HTML

```
<p>Les <span class="decoration">feuilles mortes</span> se ramassent à la pelle</p>
```

CSS

```
.decoration {  
    text-decoration: underline overline ;  
    color:#cccccc ;  
}
```

Résultat : Les feuilles mortes se ramassent à la pelle

Espacement entre les mots → la propriété **word-spacing**

La propriété **word-spacing** → valeurs: **normal**, **entier positif ou négatif** permet d'augmenter ou diminuer l'espace entre les mots en px, em, %.

Propriété héritée. La valeur « normal » est utilisée pour arrêter l'héritage

HTML

```
<p>Quelque chose d'utile pour l'oiseau</p>
```

CSS

```
p {  
    word-spacing:15px;  
}
```

Résultat

Quelque chose d'utile pour l'oiseau

Espacement entre les caractères formant les mots → la propriété **letter-spacing**

La propriété **letter-spacing** → valeurs: **normal**, **entier positif ou négatif** permet d'augmenter ou diminuer l'espace entre les lettres en px, em, %. Propriété héritée. La valeur « normal » arrête l'héritage

HTML

<p>Placer ensuite la toile contre un arbre</p>

CSS <!--valeur négative pour diminuer l'espace entre les lettres -->

```
p {  
  letter-spacing:-1px;  
}
```

Résultat

Placer ensuite la table contre un arbre

Et aussi

HTML

<p>dans un jardin dans un bois ou dans la forêt</p>

CSS <!--valeur positive pour augmenter l'espace entre les lettres -->

```
p {  
  letter-spacing:10px;  
}
```

Résultat

d a n s u n j a r d i n d a n s u n b o i s o u d a n s l a f o r ê t

Indentation de la première ligne du texte → la propriété **text-indent**

La propriété **text-indent** → valeurs: **0** (valeur par défaut), **entier positif ou négatif** pour un retrait vers la droite ou la gauche de la première ligne d'un texte, en px, em, %.
Propriété héritée. La valeur « 0 » est utilisée pour arrêter l'héritage

HTML

<p>Se cacher derrière l'arbre

Sans rien dire

Sans bouger</p>

CSS /* indentation négative */

```
p {  
  text-indent:-20px;  
}
```

Résultat

Se cacher derrière l'arbre
Sans rien dire
Sans bouger

Et aussi

HTML

```
<p>Parfois l'oiseau arrive vite<br />  
Mais il peut mettre de longues années <br/>  
Avant de se décider</p>
```

CSS /* indentation positive */

```
p {  
  text-indent:20px;  
}
```

Résultat : Parfois l'oiseau arrive vite
 Mais il peut mettre de longues années
 Avant de se décider

Alignement horizontal du texte → la propriété **text-align**

La propriété **text-align** → valeurs: **left** (valeur par défaut), **right**, **center**, **justify**.
Une valeur « left » « right » ou « justify » est utilisée pour arrêter l'héritage

Exemple pour justifier un texte dans un paragraphe → `p { width:600px ; text-align:justify ; }`

Alignement vertical du texte → la propriété **vertical-align**

La propriété **vertical-align** → valeurs: **top** ou **bottom**, **middle**, **baseline**. Utilisée pour positionner le contenu des cellules des tableaux.

Surlignage du texte → la propriété **background-color**

background-color → valeurs: **couleur**, transparent (transparent est la valeur par défaut).

HTML

```
<p>et vous <span class="surligne">écrivez votre nom </span>dans un coin du tableau</p>
```

CSS

```
.surligne {
```

```
background-color:#cccccc;
}
```

Résultat

et vous écrivez votre nom dans un coin du tableau

D. Propriétés de mise en forme de l'arrière plan

Couleur et image d'arrière-plan → la propriété **background**

Les propriétés background permettent de définir divers aspects d'un fond de page, d'une cellule de tableau, d'un paragraphe ...

Les propriétés : **background** (propriété raccourcie), **background-color** (couleur de fond), **background-image** (image de fond), **background-repeat** (répétition de l'image), **background-attachment** (fixation de l'image), **background-position** (position de l'image)

- Définir une couleur de fond → **background-color** : **transparent** (par défaut), **couleur**
- Définir une image de fond → **background-image** : **url(image.jpg)**
- Gérer la répétition de l'image de fond → **background-repeat** : **repeat** (par défaut), **repeat-x**, **repeat-y**, **no-repeat**

repeat-x : répétition horizontale

repeat-y : répétition verticale

no-repeat : pas de répétition

- Gérer le défilement de l'image de fond → **background-attachment** : **scroll** (par défaut), **fixed**
cette propriété permet de "fixer" le fond, pour ne pas qu'il défile en même temps que le texte
→ **scroll** : L'utilisation de la barre de défilement entraîne un mouvement du texte, l'image d'arrière-plan suit le mouvement.
→ **fixed** : L'utilisation de la barre de défilement entraîne le mouvement du texte. L'image d'arrière-plan est fixée, elle reste en permanence visible à l'écran.

- Gérer la position de l'image de fond → **background-position** : **top**, **right**, **bottom**, **center** ... ou les valeurs en px, %.

Position des images d'arrière-plan sur une page web

top left 0 0	25 0	top center 50 0	75 0	top right 100 0
0 25	25 25	50 25	75 25	100 25
center left 0 50	25 50	center center 50 50	75 50	center right 100 50
0 75	25 75	50 75	75 75	100 75
bottom left 0 100	25 100	bottom center 50 100	75 100	bottom right 100 100

- background-position: top right; /* position haut à droite */
- background-position: 100% 0%; /* position haut à droite */

Il est possible d'utiliser ces propriétés de façon indépendante.

Une **propriété raccourcie** permet de définir les divers aspects de l'arrière plan en une seule déclaration :

```

selecteur {
    background : 'valeur de background-color' 'valeur de background-image' 'valeur de
background-repeat' 'valeur de background-attachment' 'valeur de background-position';
}

```

Exemple : CSS → `body { background:#c00 url(fond.png) no-repeat fixed 100% 100%;}`

L'arrière plan de la page aura une image de fond « fond.png » qui ne se répétera pas et qui est fixée. Elle est positionnée au coin inférieur droit.

Exemple : `p { background:#c00 url(image.png) repeat-x; }`

Dans tous les paragraphes une image de fond s'affiche. Elle se répète à l'horizontal sur toute la largeur. La couleur rouge (#c00) apparaît aux endroits non couverts par l'image.

Exercices :

Reprendre le fichier recette.html, insérer une image de fond de page (balise body), la positionner puis mettre en forme les textes.

```
<style type="text/css" media="screen">
```

```
body {
```

```

background-color:#couleur choisie;          /*Couleur du fond*/
background-image:url( image-fond-page.jpg);   /*Insertion d'image de fond*/
background-repeat:valeur;                    /* no-repeat, repeat x, repeat y) */
background-position:valeur;                  /*image de fond positionnée */
}
</style>

```

E. Les Pseudo-formats

L'écriture d'un pseudo-format dans le fichier.css commence par deux points (:). Les pseudos-formats constituent des classes prédéfinies en CSS.

Pseudo-classes

Elles s'appliquent à certains éléments dont l'état peut changer. Les plus connues sont les liens hypertextes qui possèdent plusieurs états : lien non visité, lien visité, lien survolé, lien actif.

Divers aspects du lien

- **a** → lien à l'origine, non visité
- **a:visited** → lien visité
- **a:hover** → lien survolé, modification de l'aspect du lien au passage de la souris
- **a:active** → lien actif, modification lors de l'activation du lien

L'ordre de déclaration des propriétés - a, a:visited, a:hover, a:active - doit être respecté

Dans le fichier.html

```

<ul>
<li><a href="#">Home</a></li>
<li><a href="#">Archives</a></li>
<li><a href="#">Contacts</a></li>
</ul>

```

Dans le fichier.css

```

a, a:visited {
  color:#333333 ;
  text-decoration: none;
}
a:hover {
  color:#cc0000 ;
  text-decoration: underline;
}
a:active {
  background-color:#00ee00;
}

```

```
text-decoration: none;
}
```

Résultat:

lien & lien visité	lien survolé	lien actif

Home	<u>Home</u>	Home
Archives	<u>Archives</u>	Archives
Contacts	<u>Contacts</u>	Contacts

Le CSS va permettre de donner un aspect visuel aux liens, modifier la puce, l'enlever, déterminer la couleur, souligner, surligner ...

Ces pseudo-formats peuvent être appliqués à toutes les autres balises, le navigateur IE6 ne les reconnaît que pour les liens.

Exercice : Site paris-nature**Modifiez l'aspect des liens (navigation)**

F. Quelques propriétés CSS spécifiques

a. Quelques propriétés appliquées aux listes – propriétés des balises « ul », « ol », « li »...

Nous avons abordé la syntaxe HTML spécifique des trois types de listes – ordonnée, à puces, de définition et les différentes combinaisons possibles.

Le langage CSS va permettre de donner du style à ces listes grâce à trois propriétés spécifiques : **list-style-position** pour le positionnement, **list-style-type** pour définir les caractères précédant les éléments de la liste, puce ou numérotation, **list-style-image** pour modifier la représentation de la puce à l'aide d'une image.

La propriété **list-style-position** permet de définir un retrait positif ou négatif de la liste.

Les valeurs → **outside** (valeur par défaut) : la puce, l'image ou le numéro seront positionnés à gauche du paragraphe, en retrait et **inside** : La puce ... fait partie du paragraphe et la liste est en retrait sur la première ligne.

La propriété **list-style-type** permet de modifier l'apparence des puces ou des valeurs précédant les éléments de la liste.

Les valeurs en fonction du type de liste :

liste à puces (ul) → valeurs: **disc** (rond noir), **circle** (cercle vide), **square** (carré noir), none (pas de puce)

listes ordonnées (ol) → valeurs: **decimal** (1,2,3 ...), **lower-roman** (i, ii, iii ...), **upper-roman** (I, II, III ...), **lower-alpha** (a, b, c ...), **upper-alpha** (A, B, C ...).

Exemples :


Liste non ordonnée : **puce carrée noire**

<u>Dans fichier.css</u>	<u>Dans un fichier.html</u>	Résultat :
<pre>ul { list-style-type: square; }</pre>	<pre> Item 1 Item 2 Item 3 </pre>	<ul style="list-style-type: none">▪ Item1▪ Item 2▪ Item 3

Liste ordonnée : **lettres alphabétiques minuscules**

<u>Dans fichier.css</u>	<u>Dans un fichier.html</u>	Résultat :
<pre>ol { list-style-type: lower-alpha; }</pre>	<pre> Item a Item b Item c </pre>	<ol style="list-style-type: none">Item aItem bItem c

La propriété **list-style-image** permet de modifier l'apparence des puces par insertion d'une image

Liste non ordonnée : nécessite au préalable la création d'une **puce « image »** - nommée ici « check.jpg → 

<u>Dans fichier.css</u>	<u>Dans un fichier.html</u>	Résultat :
<pre>ul { list-style-image: url("check.jpg"); }</pre>	<pre> Item 1 Item 2 Item 3 </pre>	<ul style="list-style-type: none">✓ Item 1✓ Item 2✓ Item 3

Il est possible également d'appliquer des propriétés de formatage ou mise en forme du texte, couleur, taille du caractère ...

Listes des propriétés CSS sur le site openclassrooms :

<http://openclassrooms.com/courses/apprenez-a-creeer-votre-site-web-avec-html5-et-css3/memento-des-proprietes-css>

Exercices :

Utilisez les propriétés CSS pour modifier l'aspect visuel

- des pages du site paris-nature