Université Hassan 1er

Faculté des Sciences et Techniques de Settat

COURS CSS

Cascading Style Sheets

Professeur Laachfoubi Nabil

Département des Mathématiques et Informatique

Sommaire

A. Introduction....3
A.1. Importer une feuille de style....3
A.2. Règles CSS....4
A.3. Types de styles....5
A.4. Sélecteurs CSS....5
A.5. CSS en cascade....6
A.6. Héritage....6
A.7. Unités de mesure....7
A.8. Pseudo-classes....8
B. Polices et style du texte....9
B.1. Polices et typographie....9
B.2. Gestion des styles de texte....10
C. Dimensionnement et positionnement....11
C.1. Le dimensionnement des éléments....11
C.2. Positionnement précis des éléments....11
C.3. Visibilité et ordre d'empilement....13
D. Créer des bordures, des marges, des espacements et des contours....13
D.1. Concept du modèle de boîte....13
E. Couleurs et images de fond....15
E.1. Couleurs en CSS....15

A. Introduction

Les feuilles de style en cascade, ou CSS (Cascading Style Sheets) permettent d'appliquer des styles aux pages web pour leur donner exactement l'aspect voulu. Ceci fonctionne grâce au fait que CSS est connecté au modèle objet de document, DOM (Document Object Model).

Avec CSS et son intégration au DOM, on peut rapidement et aisément donner un nouvel aspect, un nouveau look à n'importe quel élément. Ainsi, si on n'aime pas l'aspect des textes dans les balises <h1>, <h2>, ou autres, on peut leur affecter de nouveaux styles pour remplacer les règles prédéfinis de leurs familles de polices, de leurs tailles, des attributs gras ou italiques, et de bien d'autres propriétés.

Une manière d'ajouter du style à une page web consiste à insérer les instructions nécessaires dans l'entête d'une page web, entre les balises <head> et </head>. Par exemple, pour modifier le style de la balise <h1>, on peut écrire le code suivant :

```
<style>
   h1 {color: red ; font-size: 3em ; font-family: Arial ; }
</style>
```

A.1. Importer une feuille de style

Lorsqu'il s'agit de changer le style de tout un site et non d'une seule page, il vaut mieux gérer des feuilles de style à part et de déplacer tous ces styles en dehors du code HTML des pages vers des fichiers instincts, puis d'importer ceux dont on a besoin. Cela permet d'appliquer différentes feuilles de styles à des dispositions différentes (par exemples aux affichages et aux impressions), sans modifier le HTML.

Plusieurs moyens permettent d'atteindre ce but, dont le premier consiste à utiliser la directive CSS *@import*, comme suit :

```
<style>
   @import url('styles.css') ;
</style>
```

Cette instruction indique au navigateur d'aller chercher une feuille de style dénommée *styles.css*. La commande *@import* est suffisamment souple pour permettre de créer des feuilles de styles qui, elles-mêmes, importent d'autres feuilles des styles et ainsi de suite.

Importer du CSS à partir de l'HTML

Une autre possibilité pour inclure une feuille de style en HTML consiste à faire appel à la balise HTML <link>, comme suit :

```
<link rel='stylesheet' tyles='text/css' href="styles.css'>
```

Ceci produit exactement le même effet qu'avec la directive *@import*, avec une petite nuance que *<link>* est une balise en pur HTML, et que ce n'est pas une directive de style valide, donc il n'est pas permis de la placer dans une feuille de style pour en appeler une autre et elle ne peut pas apparaitre entre les balises <style> et </style>.

Tout comme on peut utiliser plusieurs directives @import dans du CSS pour insérer plusieurs feuilles de styles externes, en HTML, on peut utiliser autant d'éléments <link> pour appeler des feuilles de styles externes.

Réglages de style intégrés

Rien n'empêche de régler ou de remplacer individuellement certains styles dans la page courante au cas-par-cas. On peut en effet insérer des déclarations de style directement dans du HTML, comme ceci :

```
<div style='font-style : italic ; color : blue ; '> Bonjour tout le monde </div>
```

Utiliser des identifiants(id)

Une meilleure solution pour définir le style d'un élément déterminé consiste à lui affecter un identifiant (id) en HTML, comme suit :

```
<div id='bienvenue'> Bonjour tout le monde </div>
```

Ceci établit que le contenu de la division d'identifiant bienvenue doit recevoir le style défini par le réglage de style bienvenue. L'instruction CSS correspondant à cet identifiant peut prendre la forme suivante :

```
#bienvenue {font-style : italic ; clor : blue ; }
```

Utiliser les classes

Lorsque on souhaite appliquer le même style à plusieurs éléments HTML, il n'est pas nécessaire de leur donner des identifiants différents, parce que on peut préciser une classe pour les gérer tous, comme ceci :

```
<div class='bienvenue'> Bonjour tout le monde </div>
```

Cela signifie que le contenu de cet élément, et de tous les autres qui utilisent cette classe, doit recevoir le style définit dans la classe bienvenue. Lorsqu'une classe s'applique, on peut définir la règle CSS suivante, que ce soit dans l'entête de page HTML ou dans une feuille de styles externe, pour préciser les styles de la classe :

```
.bienvenue {font-style : italic ; clor : blue ; }
```

Au lieu du symbole #, réservé aux identifiants, les déclarations de classes reçoivent le préfixe point (.).

Utiliser les points-virgules

En CSS, les points-virgules séparent plusieurs instructions CSS sur la même ligne. Mais s'il n'y a qu'une seule instruction dans une règle, on peut s'en passer, de même qu'à la fin de la dernière instruction du même groupe.

Cependant, pour éviter des erreurs difficiles à détecter, il vaut mieux en placer partout après chaque instruction.

A.2. Règles CSS

Chaque instruction d'une règle CSS débute par un sélecteur, c'est-à-dire l'élément auquel la règle s'applique. Par exemple, dans l'affectation suivante, h1 est le sélecteur qui reçoit une taille de police 240% plus grande que la valeur prédéfinie :

```
h1{ font-size : 240% ; }
```

Détaillons cette instruction :

1. La propriété identifie ce qui sera défini dans le style adopté. Ces propriétés sont énumérées dans les spécifications CSS. Il existe de nombreuses propriétés portant, par exemple, sur la police de caractère (font), sur le texte (text), sur l'arrière-plan (background), sur la bordure (border), etc.
2. La propriété est séparée de sa valeur par un double point.

3. Les espaces sont permis. Ainsi certains auteurs ont pris l'habitude de mettre un espace entre le double point et la valeur pour une meilleure lisibilité du code.
4. La valeur identifie la nature de l'effet de style souhaité. La valeur s'exprime par un mot-clé, un pourcentage, une grandeur en fonction de la propriété à laquelle elle est assignée.
5. Une déclaration de style se termine toujours par un point-virgule et elle est entourée par deux accolades.
6. Il est possible de définir plusieurs déclarations de style pour un sélecteur. (Par exemple : propriété1 : valeur ; propriété2 : valeur ; propriété3 : valeur ;).

A.3. Types de styles

Styles prédéfinis

Le niveau le plus faible de préséance s'applique aux styles prédéfinis, par défaut, proposés par un navigateur. Ces styles servent de solution de repli, quand une page web ne définit aucun style ; ils forment un ensemble générique de styles qui permettent d'afficher un contenu web raisonnablement bien dans la plupart des cas.

Feuilles de style externes

Les styles de préséance plus élevée suivante sont ceux affectés par une feuille de style externe. Ces réglages écrasent ceux déjà affectés par le navigateur. Les feuilles de styles externes constituent la voie recommandée de définition de styles, parce qu'elles permettent de produire et d'appliquer des feuilles de styles différents selon le but souhaité

Styles internes

Viennent ensuite les styles internes, ceux qu'on crée entre les balises <style>...</style>, qui ont la priorité sur tous les types de styles précédents.

Styles « à la volée »

Enfin, viennent les styles à la volée (inline), c'est-à-dire ceux où on affecte directement une propriété à un élément. Ils ont la plus haute priorité parmi les types de styles et s'utilisent comme suit :

```
<a href="http//www.google.ma" style="color : blue ; ">Google Maroc</a>
```

A.4. Sélecteurs CSS

Sélecteur d'attribut de texte

```
p { text-align : justify ; }
```

Sélecteur contextuel

```
p b { color : red ; }
ul li b { color : blue ; }
```

Sélecteur enfant

```
<p><i><b>Bonjour</b>à tous </i></p>
p b { color : red ; }
p > b { color : red ; }
```

Sélecteur d'identifiant (id)

```
#madiv { text-decoration : underline ; }
div#madiv { text-decoration : underline ; }
```

Sélecteur de classe

```
.maclasse { margin-left : 10px ; }
```

Sélecteur d'attribut

```
[type="submit"] { width : 100px ; }
form input[type="submit"] { width : 100px ; }
```

Sélecteur universel

```
* {border : 1px solid green ; }
#cadrevert * p {border : 1px solid green ; }
```

Sélecteur par group

```
p, #nomid,.nomclasse {border : 1px dotted orange ; }
```

A.5. CSS en cascade

À la lecture des points précédents, on constate que l'on peut avoir plusieurs définitions de style soit en ligne, interne(s) ou externe(s). En cas de concurrence entre plusieurs éléments de style, intervient alors la notion de "cascade" (le

Cascading de Cascading Style Sheets) ou d'ordre de priorité.

Le navigateur prend d'abord en considération les spécifications des feuilles de style externes (avec l'extension css), ensuite celles de style internes (soit celles à l'intérieur des balises <head>) et ensuite des feuilles de style en ligne (celles liées à un élément Html5).

Ainsi, en cas de conflit entre une spécification de style définie à la fois dans une feuille de style externe et dans une feuille de style interne, c'est la spécification de la feuille de style interne qui sera retenue par le navigateur. De même, en cas de conflit entre une feuille de style interne et en ligne, c'est cette dernière qui l'emportera.

L'ordre croissant de priorité (de la plus basse à la plus haute) est :

1. Propriétés du navigateur.
2. Les feuilles de style externes.
3. Les feuilles de style internes.
4. Les feuilles de style en ligne.

Les styles déclarés dans la feuille de style en ligne ont donc bien la plus haute priorité.

A.6. Héritage

L'héritage est le fait qu'un élément enfant possède les mêmes styles que l'élément qui le contient (son parent dans la hiérarchie des éléments d'une page). Soit par exemple, le style suivant :

```
div {color : white ; background : blue ; }
```

Le code HTML correspondant :

```
<div>
   Texte <p>Premier paragraphe <span class="italique">HTML5</span> et <b>CSS</b> cours </p>
</div>
```

L'élément <p> est enfant de et les éléments <pn> et <b> sont eux-mêmes enfants de <p>. Par héritage et faute d'avoir définis des styles propres pour les éléments <p>, et <b>, ceux-ci ont un contenu qui possède les styles définis pour leur parent direct ou indirect . Ils sont donc tous en blanc sur fond bleu. Si on crée un style différent pour l'élément <p>, ses éléments enfants héritent alors de ces styles et non plus de ceux de l'élément .

L'héritage concerne un grand nombre de propriétés CSS, mais toutes les propriétés ne sont pas systématiquement héritées, par exemple les marges, les bordures, ou les dimensions et la position pour des raisons de mise en page. Depuis CSS2, la quasi-totalité des propriétés peut prendre la valeur *inherit* permettant de définir explicitement l'héritage de la valeur que possède la même propriété dans l'élément parent.

A.7. Unités de mesure

Les feuilles de style CSS permettent d'utiliser de nombreuses unités de mesure soit en pouces (inches), en centimètres, en millimètres, en points, en picas, en pixels et en pourcentage.

On distingue les valeurs relatives qui peuvent varier selon l'ordinateur utilisé et les valeurs absolues qui restent constantes quel que soit le matériel ou le software utilisé.

Les valeurs absolues sont :

Unité	Nom	Description	Valeur	Exemple
pt	Point	72 pt = 1 inch	entier	48pt
pc	Pica	1 pc = 12 pt	réel	4.5pc
mm	millimètre	1 mm =.24 pc	entier	60mm
cm	centimètre	1 cm = 10 mm	entier	6cm
in	Inch	1 in = 2.54 cm	réel	0.1in

Les valeurs relatives sont :

Unité	Description	Valeur	Exemple
em	Unité relative se basant sur la taille de police par défaut de la page (px/16=em).	Réel	1.8em
ex	Unité relative à la hauteur de la minuscule de l'élément sélectionné.	Réel	1.3ex
px	Le pixel est la plus petite partie d'une image. Dépend de la résolution d'écran.	Entier	220px
%	Pourcentage	Entier	80%

Remarques :

- Les unités de mesure sont toujours notées par une abréviation comportant deux lettres.
- Il ne peut y avoir d'espace entre la valeur et l'unité. Si un espace est mis entre la valeur et l'unité, la feuille de style ne sera pas acceptée et ne sera donc pas affichée.
- Certaines propriétés acceptent des valeurs négatives.
- Il est généralement recommandé d'utiliser l'unité *em* pour décrire la taille des polices de caractères pour une plus grande stabilité entre les différents systèmes d'exploitation et navigateurs.

A.8. Pseudo-classes

Il existe, dans un document HTML, des éléments non définis par une balise ou par un attribut qui pourraient se révéler intéressants pour leur appliquer un effet de style comme, par exemple, le lien visité ou la première lettre d'un paragraphe.

Le W3C a introduit, pour certains de ces éléments non référencés de la page, la notion de pseudo-classes. On parle aussi de pseudo-style ou pseudo-élément.

La syntaxe des pseudo-classes est :

```
sélecteur : pseudo-classe { déclaration(s) de style }
```

Soit le sélecteur, double point, le nom de la pseudo-classe suivi par la déclaration de style entre accolades.

Les pseudo-classes peuvent également être utilisées avec des classes. Elles se noteront alors :

```
sélecteur.classe : pseudo-classe { déclaration(s) de style }
```

Soit le sélecteur, point, le nom de la classe, double point, le nom de la pseudo-classe suivi par la déclaration de style entre accolades.

Tout comme pour les classes, les pseudo-classes sont définies à l'intérieur de balises <style type="text/css"> … </style> dans le head de la page ou dans une feuille de style externe.

Pseudo-classes de lien

- **a:link**

  La pseudo-classe *a : link* permet de définir l'apparence d'un lien qui n'a pas encore été sélectionné.

```
a : link { text-decoration : none ; color : black ; }
```

- **a:visited**

  La pseudo-classe *a : visited* permet de définir l'apparence d'un lien qui a déjà été visité.

```
a : visited {text-decoration : line-through ; color : black ; }
```

- **a:hover**

  La pseudo-classe *a : hover* permet de définir l'apparence d'un lien au passage du curseur de la souris.

```
a : hover { font-weight : bold ; font-style : italic ; font-size : 1.5em ; }
```

- **a:active**

  La pseudo-classe **a : active** permet de définir l'apparence d'un lien au moment où il est cliqué par le visiteur.

```
a : active { text-decoration : underline overline ; }
```

Pseudo-classes de paragraphe

- **:first-letter**

  Cette pseudo-classe *: first-letter* permet de définir l'apparence de la première (et uniquement la première) lettre d'un paragraphe. On crée ainsi un effet de lettrine.

- **:first-line**

  Cette pseudo-classe *: first-line* permet de définir l'apparence de la première (et uniquement la première) ligne d'un paragraphe.

La longueur de la première ligne est variable, elle dépendra de la largeur de l'élément boîte, de la taille de police, etc.

Insérer un contenu

- **:before**

  La pseudo-classe *: before*, associée à la propriété content, permet d'insérer un contenu avant un élément.

- **:after**

  La pseudo-classe *: after*, associée à la propriété content, permet d'insérer un contenu après un élément.

B. Polices et style du texte

B.1. Polices et typographie

Les polices possèdent quatre propriétés qui permettent d'en modifier l'apparence en CSS : la famille, le style, la taille et l'épaisseur du trait. Ces propriétés permettent d'ajuster avec précision la manière dont un texte s'affiche sur une page web ou à l'impression.

Type de police : font-family

```
p { font-family : Arial ; }
p { font-family : Arial, Helvetica, Georgia ; }
p { font-family : 'Courrier New', 'Trebuchet MS', 'Lucida Console' ; }
```

Famille de police

Au lieu de spécifier le nom de la police, on peut utiliser des familles de police ou des noms de polices génériques :

1. **Serif** : Les polices de type serif ont des terminaisons au bout de leurs traits et un espacement des lettres proportionnel (Times New Roman, Bodoni, Georgia, Garamond, …).
2. **Sansserif** : Les polices sans serif ont les bouts des traits pleins, sans évasement et un espacement des lettres proportionnel (Arial, Verdana, Helvetica, Trebuchet, …).
3. **Cursive** : Famille de polices dont le résultat est proche d'une écriture manuscrite (Script, Adobe Poetica, …).
4. **Fantasy** : Les polices fantaisies ou décoratives (Wingdings, …).
5. **Monospace** : Toutes les lettres ont les mêmes dimensions. L'aspect est semblable à celui obtenu avec une machine à écrire manuelle et on emploie souvent ce type de police pour écrire du code informatique (Courier, Courier New).

Le navigateur choisira lui-même la police disponible sur la machine de l'utilisateur qui correspond aux critères du nom générique.

Style de police : font-style

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

Variante de police : font-variant

```
.caps { font-variant : small-caps ; }
.normal { font-variant : normal ; }
```

Taille de police : font-size

```
h1 { font-size : 40px ; }
h1 { font-size : 2.5em ; }
```

Epaisseur du trait : font-weight

```
p.normal { font-weight : normal ; }
p.lighter { font-weight : lighter ; }
p.bold { font-weight : bold ; }
p.bolder { font-weight : bolder ; }
p.bold900 { font-weight : 900 ; } /* de 100 à 900*/
```

B.2. Gestion des styles de texte

Décoration : text-decoration

```
h1 { text-decoration : overline ; }
h2 { text-decoration : line-through ; }
h3 { text-decoration : underline ; }
```

Espacements

```
p {
   line-height : 125% ;
   word-spacing : 30px ;
   letter-spacing : 3px ;
}
```

Justification : text-align

```
h1 { text-align : center ; }
h2 { text-align : left ; }
h3 { text-align : right ; }
```

Transformation : text-transform

```
p.uppercase { text-transform : uppercase ; }
p.lowercase { text-transform : lowercase ; }
p.capitalize { text-transform : capitalize ; }
```

Retrait : text-indent

```
p { text-indent : 50px ; }
```

C. Dimensionnement et positionnement

C.1. Le dimensionnement des éléments

Longueur et hauteur

Les propriétés de style *width* et *height* vont fixer respectivement la largeur et la hauteur de l'élément boîte.

width :	auto ou valeur de longueur ou pourcentage
Height :	auto ou valeur de longueur ou pourcentage

En plus de *width* et *height*, il existe à partir de CSS2 des propriétés pour déterminer la taille minimale et maximale d'un élément de type bloc ou d'un élément remplacé dynamiquement par du JavaScript.

max-height : min-height : max-width : min-width :	Fixe la hauteur maximale Fixe la hauteur minimale Fixe la largeur maximale Fixe la largeur minimale

Dépassement

La propriété overflow détermine ce que le navigateur doit faire lorsqu'un élément est plus grand que l'élément parent qui le contient.

overflow :	hidden ; ou scroll ; ou visible ; ou auto ;

C.2. Positionnement précis des éléments

Un élément peut se positionner de quatre façons différentes.

Positionnement statique

C'est le positionnement normal de l'élément selon la façon habituelle de procéder du navigateur.

Le positionnement statique se détermine par :

```
position : static ;
```

Le concepteur n'a donc pas le contrôle. L'élément ne peut être positionné ou repositionné et sa visibilité ne peut être modifiée. Il n'est également pas possible d'utiliser du JavaScript pour changer la position de l'élément.

Positionnement relatif

C'est le positionnement d'un élément par rapport à sa position normale ou statique.

Cet élément reste dans le flux des données mais est en quelque sorte excentré par rapport à sa position normale.

La position est définie par les coordonnées (x , y) où :

- x est la distance par rapport au bord gauche de l'élément parent ou de la fenêtre du navigateur (axe horizontal). Ainsi, left détermine la distance entre la gauche de l'élément et la gauche de la page et right, la distance entre la droite de l'élément et la droite de la page.
- y est la distance par rapport au bord supérieur de l'élément parent ou de la fenêtre du navigateur (axe vertical). Ainsi top détermine la distance entre le bord supérieur de l'élément et le bord supérieur de la page et bottom, la distance entre le bord inférieur de l'élément et le bord inférieur de la page.

position : relative ;	left : valeur ou %; top : valeur ou %; right : valeur ou %; bottom : valeur ou %;

Positionnement absolu

Le positionnement absolu crée un élément indépendant du reste du document. Les éléments définis en position absolue sont retirés du flux normal et se positionnent à l'emplacement exact défini par le concepteur.

La position est définie par les coordonnées (x , y) où :

- x est la distance par rapport au bord gauche de l'élément parent ou de la fenêtre du navigateur (axe horizontal). Ainsi, left détermine la distance entre la gauche de l'élément et la gauche de la page et right, la distance entre la droite de l'élément et la droite de la page.
- y est la distance par rapport au bord supérieur de l'élément parent ou de la fenêtre du navigateur (axe vertical). Ainsi top détermine la distance entre le bord supérieur de l'élément et le bord supérieur de la page et bottom, la distance entre le bord inférieur de l'élément et le bord inférieur de la page.

position : absolu ;	left : valeur ou %; top : valeur ou %; right : valeur ou %; bottom : valeur ou %;

Positionnement fixe

Le positionnement fixe crée aussi un élément indépendant dont on peut définir la position. Ici l'élément reste fixe même lorsque l'on fait défiler le document.

position : fixe ;

Flottement

La propriété float retire un élément boîte du flux normal pour la placer le plus à droite ou le plus gauche possible dans son élément parent, soit son conteneur.

float :	right ; ou left ; ou none ;

- La valeur right aligne l'élément concerné à droite, poussant les autres éléments à s'aligner à gauche.
- La valeur left aligne l'élément concerné à gauche, poussant les autres éléments à s'aligner à droite.
- La valeur none ne spécifie rien et rend la main au navigateur.
- La position float ne peut s'appliquer en cas de positionnement absolu.

Dégagement

La propriété clear permet d'annuler le flottement introduit par la propriété float.

clear :	right ; ou left ; ou both ; ou none ;

- La valeur right annule le flottement à droite.
- La valeur left annule le flottement à gauche.
- La valeur both annule le flottement des deux côtés.
- La valeur none annule toute propriété de flottement.

C.3. Visibilité et ordre d'empilement

Superposition

La propriété z-index ajoute un axe en profondeur permettant de positionner des éléments au-dessus ou en dessous d'un autre élément.

Ainsi, l'élément avec une propriété z-index : 2 apparaîtra avant ou au-dessus de l'élément avec une propriété z-index : 0.

z-index :	un nombre entier (positif);

Cette propriété ne fonctionne qu'avec un positionnement absolu des éléments.

Visibility

La propriété CSS visibility détermine si un élément est visible ou caché.

visibility :	visible ; ou hidden ;

N.B : Attention ! Malgré la valeur hidden, l'élément occupe toujours sa place dans le document et un rectangle blanc apparaît là où il pourrait être affiché. De cette manière, la mise en page est conservée malgré l'absence de l'image et les autres éléments de la page conservent leur position relative initiale. La valeur

hidden diffère en cela de la propriété display : none ; qui retire purement et simplement l'élément de la page et ne réserve pas de place pour celui-ci.

Affichage

La propriété display permet de contrôler l'affichage des éléments dans la page.

display :	block ; ou inline ; ou none ;

D. Créer des bordures, des marges, des espacements et des contours

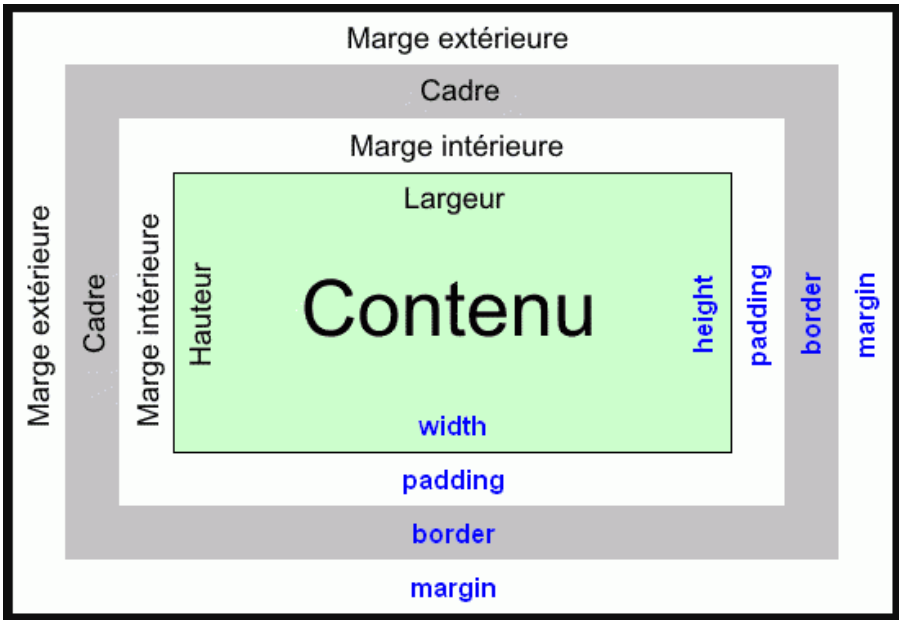
D.1. Concept du modèle de boîte

L'élément boîte est une notion importante en CSS. Cette notion, un peu abstraite, doit être correctement intégrée pour la suite de notre étude et tout particulièrement en vue du positionnement.

Cet élément boîte est aussi appelé un conteneur ou un bloc (block).

Le W3C définit un élément boîte comme une zone rectangulaire constituée :

- D'un contenu,
- D'une marge intérieure (padding),
- D'une bordure (border),
- Et d'une marge extérieure (margin).



Marges externes

Cette propriété permet de définir la marge extérieure de l'élément boîte.

margin : margin-top : margin-right : margin-bottom : margin-left :	auto ou valeur de longueur ou pourcentage

Marges internes

Cette propriété permet de définir la marge intérieure de l'élément boîte.

padding : padding-top : padding-right : padding-bottom : padding-left :	auto ou valeur de longueur ou pourcentage

Couleur de la bordure

Ces propriétés de style permettent de définir la couleur des quatre bordures de l'élément boîte ou de chaque bordure prise individuellement

border-color : border-top-color : border-right-color : border-bottom-color : border-left-color :	nom de couleur notation hexadécimale sous sa forme #xxxxxx notation hexadécimale abrégée sous la forme #xxx notation en mode RVB avec des entiers de 0 à 255 notation en mode RVB avec des % de 0 à 100 ou notation en mode RVBa où a est compris entre 0 et 1. notation en mode HSL notation en mode HSLa où a est compris entre 0 et 1. transparent

Épaisseur de la bordure

Cette propriété permettra de définir l'épaisseur des quatre bordures simultanément ou de chaque bordure prise individuellement.

border-width : border-top-width : border-right-width : border-bottom-width : border-left-width :	valeur de longueur; thin ; medium ; thick ;

Style de la bordure

Cette propriété CSS apporte une variété de bordures par rapport au seul style de bordures du Html.

border-style : border-top-style : border-right-style : border-bottom-style : border-left-style :	solid ou dashed ou dotted ou double ou groove ou ridge ou inset ou outset ou hidden ou none;

E. Couleurs et images de fond

E.1. Couleurs en CSS

Les feuilles de style CSS proposent de multiples notations pour déclarer une couleur. Soit :

- La notation hexadécimale classique soit #ffcc00. Elle définit la couleur, ou plutôt ses trois composantes rouge (r pour red), vert (g pour green) et bleu (b pour blue), via une notation hexadécimale de type #rrggbb.
- La notation hexadécimale abrégée soit #fd3. Chaque chiffre est alors implicitement dupliqué. Par exemple, #fd3 correspond à la notation classique #ffdd33. On comprend donc qu'on ne pourra pas abréger une couleur comme #cfe4f5.
- La notation décimale soit, par exemple, color : rgb(0, 0, 255). Le code RGB de la couleur n'est plus encodé en valeur hexadécimale mais par un nombre entier compris entre 0 et 255. C'est donc l'équivalent décimal de la notation hexadécimale.
- La notation en pourcentage soit, par exemple, color : rgb(25%, 50%, 0%). La valeur 0% signifie l'absence de la composante, 100% qu'elle est à son maximum.
- Les mots-clés soit, par exemple, color : red. Ces couleurs, au nombre de 17, sont désignées par un nom en anglais et constituent les couleurs basiques. On y retrouve : green (vert), yellow (jaune), blue (bleu moyen), orange (orange), white (blanc), red (rouge), black (noir), maroon (brun marron), lime (vert citron), aqua (turquoise), teal (cyan foncé), navy (bleu marine), olive (olive), fuchsia (fuchsia), purple (pourpre), silver (gris clair) et gray ou grey (gris foncé).
- La notation RGBa, qui obéit aux mêmes règles de fonctionnement que la notation classique RGB, mis à part qu'une composante est ajoutée à la valeur : rgb(0,0,0). Celle-ci devient donc rgba(0,0,0,0). La dernière valeur indiquant le degré d'opacité ou de transparence entre 0 et 1.
- La notation HSL (Hue Saturation Luminance),qui consiste en trois valeurs. La première est exprimée en degrés de 0° à 359°. Elle correspond à une couleur dans la roue chromatique : le vert (120°), le cyan (180°), le bleu (240°) et le magenta (300°). Les secondes et troisièmes valeurs sont exprimées en pourcentage et notent respectivement la saturation et la luminosité. Par exemple, color : hsl(0, 100%, 50%) pour le rouge.

- La notation HSLa, qui ajoute une valeur comprise entre 0 et 1 pour la transparence ou l'opacité. Par exemple, color : hsl(0, 100%, 50%, 0.5) pour un rouge semi-transparent.

Couleur d'arrière-plan

Cette propriété permet de donner une couleur de fond à un élément.

background-color :	nom de couleur ou notation hexadécimale sous sa forme #xxxxxx ou notation hexadécimale abrégée sous la forme #xxx ou notation en mode RVB avec des entiers de 0 à 255 ou notation en mode RVB avec des % de 0 à 100 ou notation en mode RVBa où a est compris entre 0 et 1. notation en mode HSL notation en mode HSLa où a est compris entre 0 et 1. transparent

Insertion d'une image d'arrière-plan

Cette propriété insère une image de fond disposée de façon classique, soit en mosaïque.

background-image :	url(fichier_image) ; none ;

- On peut utiliser l'adressage relatif ou absolu pour indiquer l'adresse (URL) de l'image.
- Les images peuvent être au format GIF, JPEG ou PNG.
- La valeur none n'affiche aucune image d'arrière-plan.

Répétition de l'image

Cette propriété donne le contrôle au concepteur quant à la répétition de l'image qui ne sera ainsi plus nécessairement disposée en mosaïque.

background-repeat :	repeat ; repeat-x ; repeat-y ; no-repeat ;

Positionnement de l'image

Cette propriété va permettre de positionner précisément l'image de fond d'un élément, au pixel près.

Background-position :	valeur de longueur valeur de longueur; pourcentage pourcentage; left ou center ou right / top ou center ou bottom ;