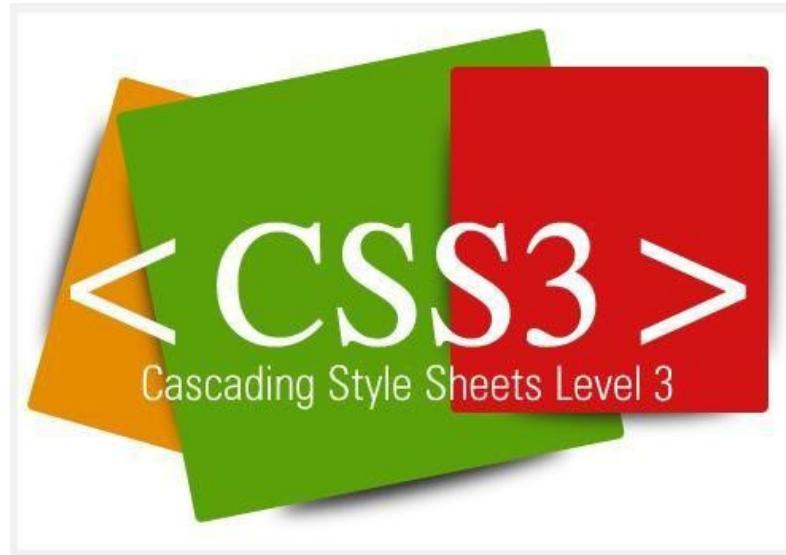


Le CSS, Le CSS3 ses nouveautés



Introduction

- Une brève histoire du CSS
- Principe de cascade

Une introduction au langage CSS

- La syntaxe d'une règle de style
- Emplacements des styles
- Les sélecteurs CSS de style
- Les propriétés et leur ordre
- Valeurs, tailles et unités
- La mise en page

Les nouveautés CSS 3

- Les codes de couleurs et niveaux de transparence
- Les effets pour le texte
- Multicolonnage
- Nouveaux types de sélecteurs
-

Présentation du CSS

- ❑ **CSS** : Cascading Style Sheets (feuilles de style en cascade).
- ❑ Un standard publié par W3C.
- ❑ Il appartient à la famille de HTML5.
- ❑ **Rôle** : mettre en forme un ou des documents web d'un site par l'intermédiaire des **propriétés**.
- ❑ Il dissocie le contenu (HTML) de la mise en forme (CSS).
 - Ne pas répéter le même code de mise en forme dans chaque page.
 - Réduire le temps de connexion
 - Utiliser des styles génériques avec des noms explicites
 - Mise à jour plus facile
 - Faciliter la lecture de code de la page

Principe de cascade

La mise en forme d'une page web peut faire appel à plusieurs feuilles de style.

□ affectation de plusieurs propriétés pour un même élément pour lui donner sa mise en forme finale.

Quelle propriété sera appliquée?!

□ des règles de priorités s'appliquent : le dernier style défini est pris en compte

Ordre de priorité :

CSS local > CSS interne > CSS importée > CSS externe

- **CSS3** est **rétro-compatible** avec les versions précédentes (CSS1, CSS2 et CSS2.1).
 - utilisable avec des pages web existantes sans modifier quoi que ce soit.
 - ajoute des fonctionnalités et des caractéristiques au lieu de modifier la façon d'utiliser CSS.
- Les nouvelles fonctionnalités :
 - *des nouveaux sélecteurs et sélecteurs d'attributs*
 - *modèle de boîtes*
 - *Transformations 2D et 3D*
 - *Animations*
 - *Disposition sur plusieurs colonnes*
 - *Bordures arrondis*

Préfixe de navigateur

- ❑ Les nouveautés CSS3 n'ont pas été implémentées directement dans les navigateurs.
- ❑ Certains navigateurs ont développés leurs propres règles CSS3.
- ❑ Un préfixe CSS pour chaque navigateur et la page doit être testée sur plus grand nombre de navigateur.



Safari

-webkit-



Mozilla

-moz-



Microsoft

-ms-



Chrome

-chrome-

La syntaxe CSS

- Une feuille de style **CSS** fonctionne sous forme des **règles**.
- Chaque règle est composé d'un **sélecteur** et d'un ensemble de **déclarations** de style.
- Une déclaration est un couple de **propriété-valeur**.

```
sélecteur
{
    propriété : valeur;
    /*autres propriétés*/
}
```

- Chaque déclaration se termine par « ; »

- Une règle CSS peut être écrite de trois manières différentes :

La syntaxe CSS

Exemple
CSS

```
<h1>Ceci est un titre</h1>
<p>
  Un paragraphe de texte avec un
  <a href="page2.html">lien</a> vers une autre
  page.  Les mots suivants
  <strong>sont importants</strong>
</p>
```

Feuille de style:

```
h1
{
  font-size: 200%;
}

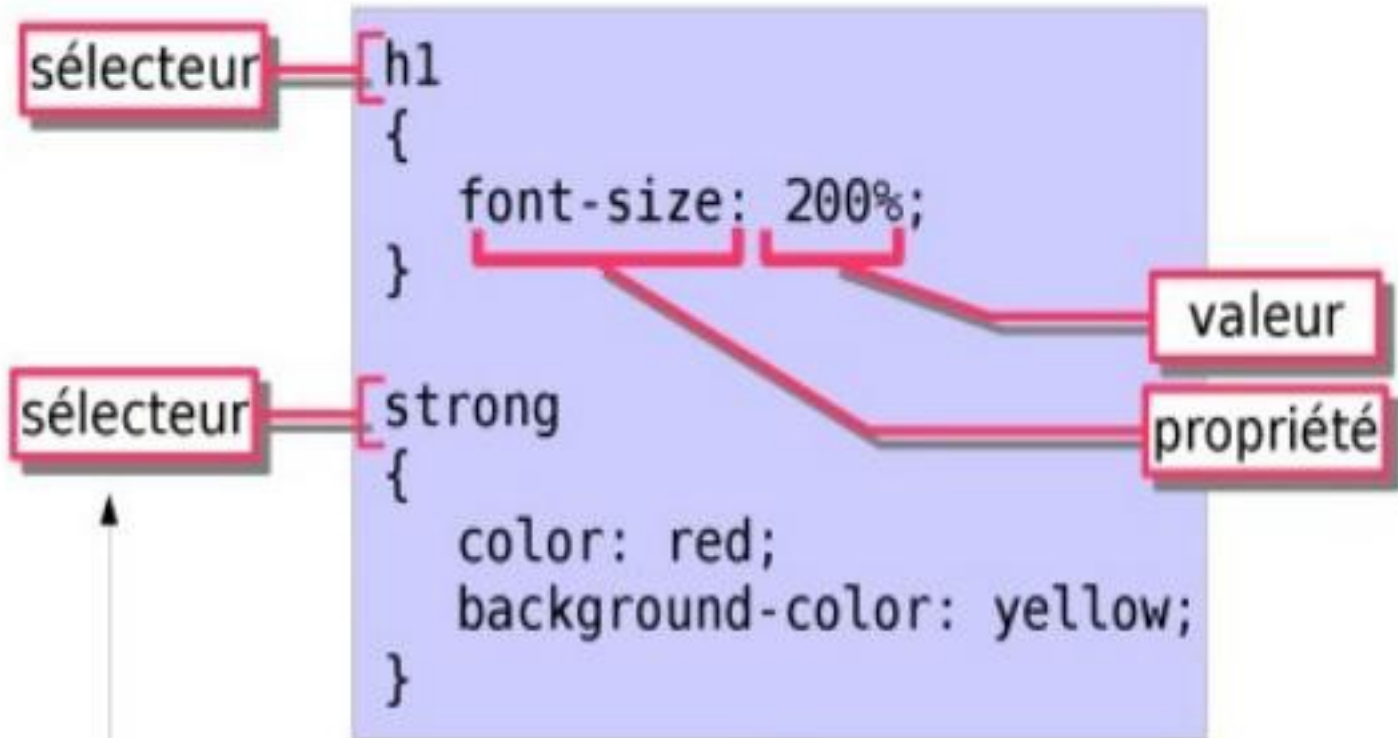
strong
{
  color: red;
}
```

Ceci est un titre

Un paragraphe de texte
avec un [lien](#) vers une autre
page. Les mots suivants
sont importants

La syntaxe CSS

Exemple
CSS



sélecteur:
sur quels éléments
s'appliquent les styles

insensible aux espaces
et fins de ligne
`h1 { font-size: 200%; }`

Spécifier un style

Le style peut s'appliquer de trois manières :

- **Style interne**

```
<head>
  <style type
    ="text/css"> h1
    {
      text-decoration : underline;
    }
  </style>
</head>
```

- **Style local ou en ligne**

Le style est dans la balise elle-même et à éviter que possible

```
.
<h1 style="text-decoration: underline; color : red;" > Titre </h1>
```

Spécifier un style

■ Style externe

On ajoute dans le **<head>** un lien à un fichier distinct
(.css)

```
<link href="style.css" type="text/css"
rel="stylesheet"/>
</head>
```

ou

```
.....
<style type="text/css">
    @import url (mes-styles.css);
</style>
.....
</head>
```

A NOTER

*/ La règle **@import** peut être utilisée dans une feuille de style externe : importer une feuille de style dans une autre.

*/ Avec **<!Doctype HTML>**, l'attribut type est inutile.

Adapter les styles aux médias

L'adaptation des feuilles de style au périphérique de sortie est possible soit dans :

- La feuille de style directement

```
@media screen {  
  /*règle de style */;  
}
```

- L'élément `<link>` grâce à l'attribut **média**.

```
<link      rel="stylesheet"  href="style.css"  
          media="screen"  />
```

- Les types de médias possibles : **all** (), **print** (imprimante), **handheld** (appareils portatifs), **tv** (télévision), **projection** (présentations projetées), **screen** (écran d'ordinateur en couleur),

Les sélecteurs de style

- Un **sélecteur CSS** permet de désigner les balises concernées de la mise en forme éventuellement de nature différente ou une relation entre deux éléments.

Exemple :

- sélectionner tous les titres de niveau 2 dans le menu,
 - ou encore tous les éléments que l'on a marqués comme étant en rouge.
- Les sélecteurs CSS trient les éléments de la page selon : *leur type, leurs attributs, leur imbrication dans le code de la page ou selon l'interaction avec l'utilisateur pour les liens.*
- Il est possible de combiner les différents types de sélecteurs (décrits ci-après).
- CSS définit plusieurs types de sélecteurs pour parcourir l'arbre de document HTML.

Les sélecteurs de style

Attributs *id* et *class*

id: nom pour un élément unique

```
<h1>Un essai</h1>
<h2>Titre section-1</h2>
<p id="para1">bla bla bla bla</p>
<h2 class="actu">Titre section-2</h2>
<p>bli bli bli bli bli</p>
<h2>Titre section-3</h2>
<p>Qu'est qu'on s'amuse!</p>
<h2 class="actu">Titre section-4</h2>
<p>abc def ghi jkl mno</p>
<h2 class="actu">Titre section-5</h2>
```

class: nom permettant de désigner plusieurs éléments

#para1 {color: green;}

p {color: red;}

#unID
nombalise
.uneClasse

```
<h1>Un essai</h1>
<h2>Titre section-1</h2>
<p id="para1">bla bla bla bla</p>
<h2 class="actu">Titre section-2</h2>
<p>bli bli bli bli bli</p>
<h2>Titre section-3</h2>
<p>Qu'est qu'on s'amuse!</p>
<h2 class="actu">Titre section-4</h2>
<p>abc def ghi jkl mno</p>
<h2 class="actu">Titre section-5</h2>
```

.actu {color: blue;}

Sélecteur d'identifiant

```
#nom_id
{ /* déclaration(s) */ }

élément#nom_id
{ /* déclaration(s) */ }
```

HTML

```
<div id="contenu">
...
</div>
```

CSS

```
#contenu
{ background-color: yellow; }
```



La règle s'applique à la balise `<div>` ayant un identifiant « contenu »

Il doit être seul et unique dans la page.

A NOTER

Un sélecteur pour autres attributs est déterminé par **[nom_attribut]** à un Exemple de type `input[name] {border : 1px solid silver}.`

Sélecteur de classe

.nom_classe

```
{ /* déclaration(s) */ }
```

Élément.nom_classe

```
{ /* déclaration(s) */ }
```

HTML

```
<p class="stylep">...</p>
<ul class="stylepp" >
  <li>...</li>
  <li>...</li>
</ul>
```

N'importe quel élément HTML de la page peut utiliser cette classe.

CSS

.stylep

```
{ color: yellow; }
```

p.stylepp

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

Les éléments de classe *stylep* sont affichés en gris et les paragraphes de cette classe ont en plus leur texte en italique.

Sélecteur d'attribut

Le sélecteur d'attribut est référençable en ajoutant **[nom- attribut]** à un sélecteur de type.

élément **[attribut]**
{ /* déclaration(s) */ }

→ Sélectionner tous les éléments qui possèdent un attribut

«
attribut ».

élément **[attribut="valeur"]**

{ déclaration(s) }

→ Sélectionner tous les éléments qui possèdent un attribut

« *attribut* » égale à la valeur « *valeur* ».

élément **[attribut*="valeur"]**

{ déclaration(s) }

Sélectionner tous les éléments qui possèdent un attribut

Sélecteurs génériques

Sélecteur de type

Le sélecteur de type est le nom d'une balise dans la page web.

élément

{ déclaration(s) }

```
.....  
<h2>  
Titre en niveau 2  
</h2>  
.....
```

CSS

```
h2  
{ font-size: 150%; }
```



La règle s'applique à toutes les balises `<h2>` et fixe la taille de la police à 150% de la police normale.

Sélecteur universel

Le sélecteur (*) sélectionne toutes les balises du document web et modifie le style de tout le document.

*

```
{ /* déclaration(s) */ }
```

HTML

```
<p>.....</p>  
<h2>  
Titre en niveau 2  
</h2>  
.....
```

CSS

*

```
{ font-family : serif; }
```

Sélecteurs hiérarchiques

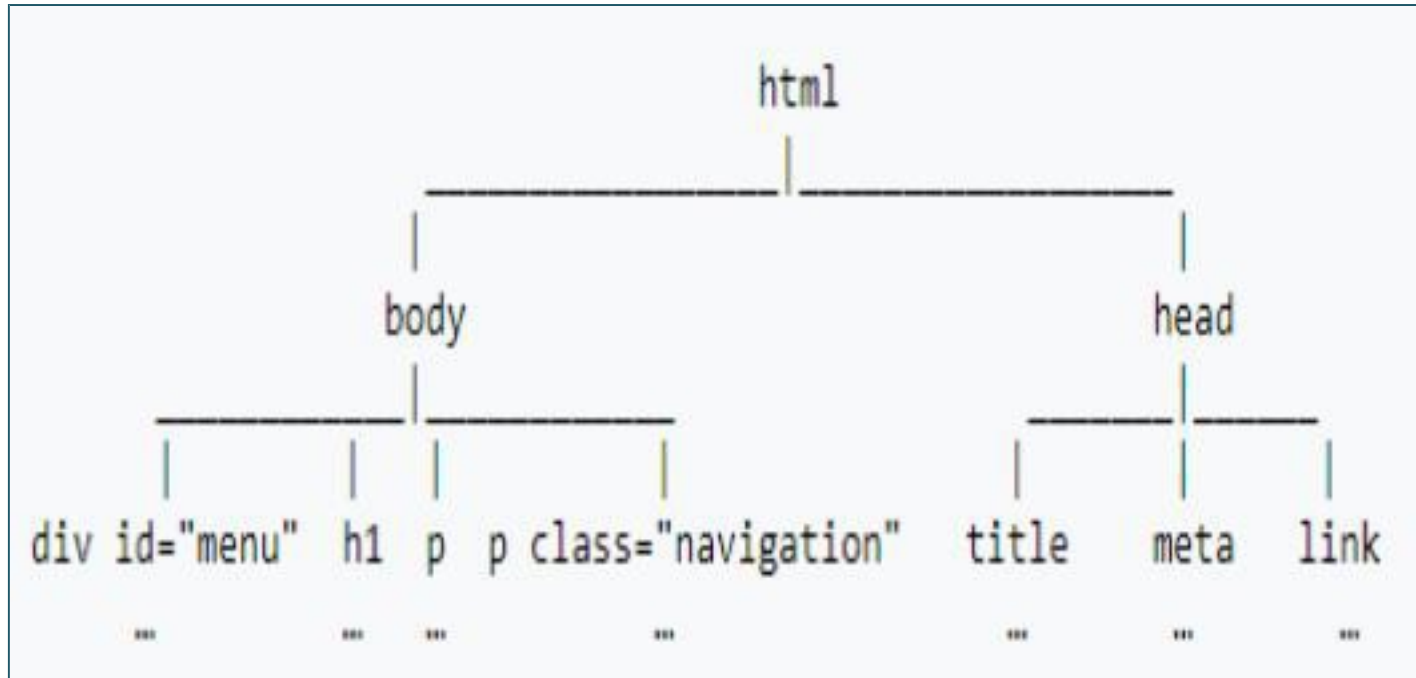
```

<!DOCTYPE html>
<html lang="fr" xml:lang="fr">
  <head>
    <title>Didacticiel pour le CSS</ title>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
    <link rel="stylesheet" type="text/css" href="essai.css" />
    <!-- Inclusion de la feuille de style externe -->
  </head>
  <body>
    <div id="menu">
      <!-- Menu -->
      <ul>
        <li><a href="#1">Premier point</ a></li>
        <li><a href="#2">Deuxième point</ a></li>
        <li><a href="#3">Troisième point</ a></li>
      </ul>
    </div>
    <!-- Corps du texte -->
    <p> Ceci est un fichier <abbr title="Hypertext Markup
      Language">HTML</abbr> utilisant une feuille de style. </ p>
    <h1><a name="1">Premier point</ a></h1>
    <p> L'objet de ce fichier est de fournir un support au didacticiel < abbr
      title="Cascading Style Sheets" >CSS</abbr>. </p>
    <p class="navigation">
      | <a href="#menu">menu</ a> | </p>
    <h1><a name="2">Deuxième point</ a></h1>
    <p> Outre mesure, ce fichier ne présente strictement
    <em>aucun</em>intérêt.
    </p>
    <p class="navigation"> | <a href="#menu">menu</ a> | </p>
    <h1><a name="3">Troisième point</ a></h1>
    <p> Nous vous souhaitons une bonne journée. </p>

```

Sélecteurs hiérarchiques

Arbre de document



A NOTER

Chaque élément a un élément ***parent***, sauf l'élément racine `html`, et un ou plusieurs ***élément(s) enfant(s)***, sauf les éléments terminant une ramification.

Un sélecteur suit ces ramifications, en partant du parent HTML, pour désigner une ou des branches — qui sont des éléments HTML

Sélecteur de descendant

- Noté par un ou plusieurs espaces
- Désigne les éléments qui ont une relation de parenté.
- Il suffit que la 2^{ème} balise soit, descendant, direct ou éloigné, de la 1^{ère} balise.

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



Justifier seulement le texte des éléments `<p>` qui sont inclus dans un bloc `<div>`.

```
#sommaire a { color: gray; }
```



Mettre en gris uniquement les liens contenus dans l'élément d'identifiant « sommaire ».

Sélecteur d'enfant

- Le **sélecteur d'enfant**, noté (`>`), permet de désigner un élément par filiation directe à un autre élément (non les petits-enfants).

```
div > p { font-style: italic; }
```



Mettre en italique le texte de l'élément `<p>` dans le **1^{er} niveau** d'imbrication à l'intérieur d'une balise `<div>` (*enfant direct*). Cette règle ne s'applique pas sur des balises `<p>` ayant d'autres niveau d'imbrication.

```
#wiki > h1 { font-style: italic; }
```



Mettre en italique le titre de l'élément `<h1>` dans le **1^{er} niveau** d'imbrication à l'intérieur d'un élément d'identifiant « wiki ».

Sélecteur de frère adjacent

Le **sélecteur de frère adjacent**, noté (+), permet de définir un style pour un élément enfant suivant immédiatement un autre élément enfant du même élément parent.

□ La relation de filiation est donc à nouveau directe.



Le style s'applique à l'élément `<h2>` qui suit immédiatement l'élément `<h1>`.

```
<div class="wiki">
  <h1> Titre 1 </h1>
  <h2> Titre 1.1 </h2>
  <p>Paragraphe...</p>
  <h2> Titre 1.2 </h2>
  <p>Paragraphe...</p>
</div>
```


Les pseudo-classes

- Le **pseudo-classe**, accolée à une balise, apporte des précisions au sélecteur (affiner le style).
- Les pseudo-classes sont prédéfinies (on ne peut pas créer d'autres pseudo-classe).

```
a : hover { color: red; }
```



Mettre en rouge le texte d'une balise `<a>` au moment de survol de curseur par la souris.

```
a.menu : hover { color: red; }
```



Mettre le texte en rouge au passage de la souris mais uniquement sur les balises `/>`.

Pseudo-classes pour les liens

`:link` = lien hypertexte non visité

`:visited` = lien visité

Pseudo-classes d'interaction avec

l'utilisateur élément survolé par la souris

`:hover` =

`:focus` = élément qui possède le focus

`:active` = élément activé

Autres pseudo-classes

`:first-child` = premier élément enfant de son conteneur

`:lang(fr)` = élément qui possède un attribut `lang="fr"`

Les pseudo-éléments

- Le **pseudo-élément**, accolée à une balise, apporte d'autres types de précisions au sélecteur.
- CSS 3 ajoute « : » aux pseudo-éléments pour les différencier des pseudo-classes.

::first-letter = première lettre du bloc

::first-line = première ligne du bloc

::before = insérer (texte ou image) avant une balise spécifiée

::after = insérer (texte ou image) après la balise spécifiée

Unité de taille

- Les unités sont utilisées pour les propriétés de : taille des caractères, marge, bordure, dimensions, largeur/hauteur de blocs, images,
- Le choix de l'unité dépend du média auquel s'applique la feuille de style.
 - ✓ Les unités absolues sont destinées aux feuilles de styles d'impression.
 - ✓ Le pixel est destiné aux feuilles de styles d'affichage
- Les unités de taille utilisées pour le style d'affichage sont :
px (pixels), **em** (taille de police), **%** (pourcentage).

Unités de mesure

pt (1 point = 0,35 mm = 1/72 in) : entier

pc (1pica = 12 pt = 4,22 mm) : réel

in (1 inch ou 1 pouce = 2.54 cm) : réel

cm (1 cm = 10 mm) : entier

mm (1 mm = 24 pc) : entier

Unités de taille absolue

Unités de taille relative

em (la taille de police) : réel

ex (la taille de la lettre minuscule de l'élément) : réel

% (à la taille de l'objet parent) : entier

px (la taille de l'écran et la résolution employée) : entier

1 em = 100% = 16 px

medium (taille standard), **xx-small**, **x-small**, **small**, **large**, **x-large**, **xx-large**

Taille définie par
mots-clés

Les couleurs

Les couleurs sont définies à l'aide :

- **Des noms** : blue, red, white, ...
- **Des codes**

```
p {color : #f00; }  
p {color : #ff0000; }  
p {color : rgb(255,0,0); }  
p {color : rgb(100%,0%,0%); }
```

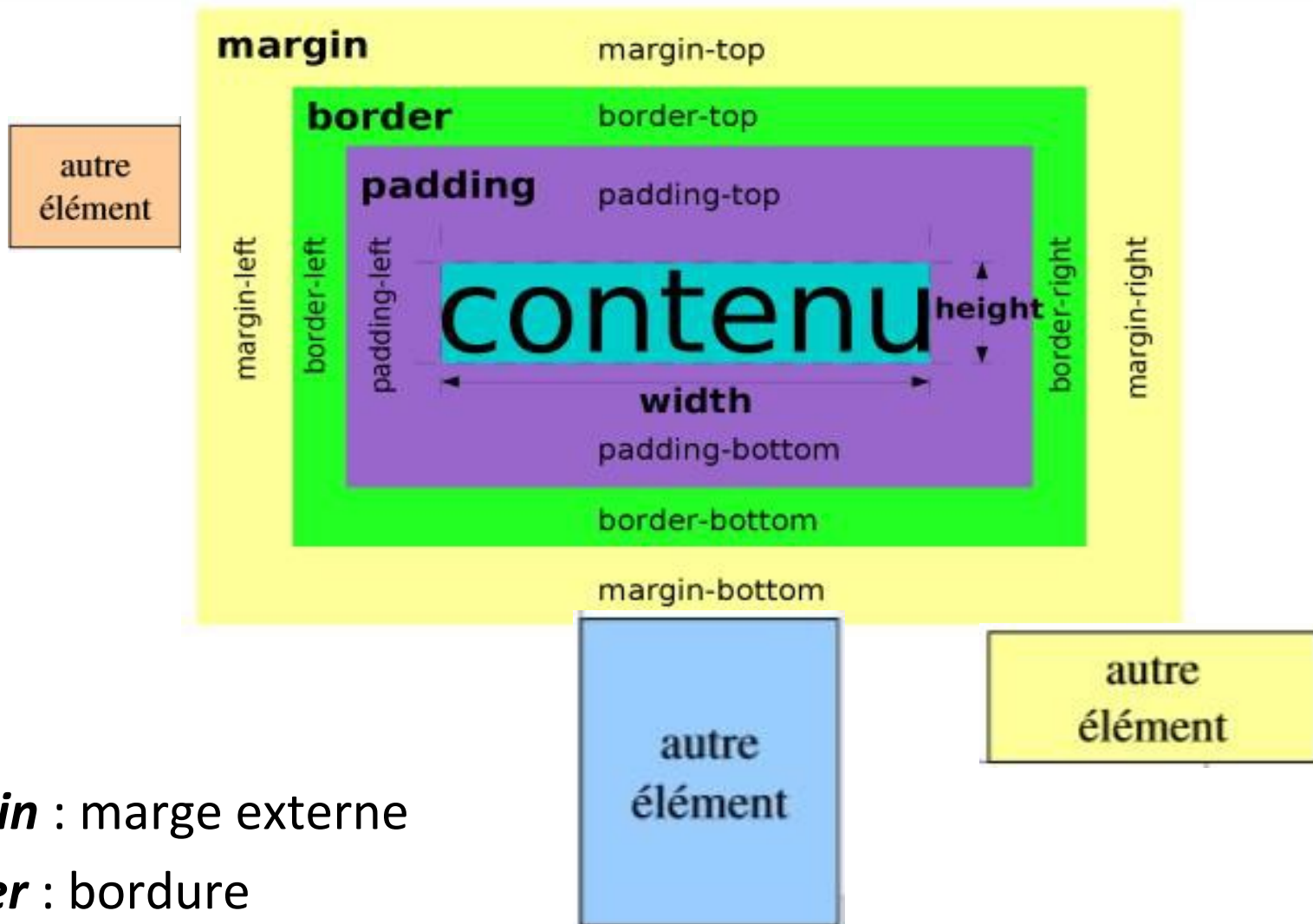
- ✓ **#RVB** : chaque lettre R, V et B est un chiffre hexadécimal entre 0 et F.
- ✓ **#RRVVBB** : chaque paire de lettres RR, VV et BB est un nombre hexadécimal entre 00 et FF.
- ✓ **rgb(R, V, B)** : chaque lettre R, V et B est un nombre décimal entre 0 et 255.
- ✓ **rgb(R%, V%, B%)** : chaque lettre R, V et B est un nombre décimal entre 0 et 100.

La spécification CSS3 ajoute d'autres notations de couleurs :

- ✓ **rgba(R, V, B,a)** : chaque lettre R, V et B est un nombre décimal entre 0 et 255, avec a : degré d'opacité ou de transparence entre 0 et 1.
- ✓ **HSL(Hue, Saturation, Luminance)** : TSL (Teinte, saturation, Luminosité). La notation HSL correspond en 3 valeurs :
 - **valeur1** : exprimée en degré de 0° à 359°. Elle correspond à une couleur dans la roue chromatique (*rouge (0°), jaune (60°), vert (120°) cyan(180°) bleu (240°) et magenta (300°)*).
 - **valeur2** et **valeur 3** : exprimée en pourcentage et notent respectivement la saturation et la luminosité.
- ✓ **HSLa (R%, V%, B%)** : ajoute une valeur comprise entre 0 et 1 pour la transparence ou l'opacité.

Mise en page du site

Modèle de boîtes (box model)



margin : marge externe

border : bordure

padding : marge interne

Les marges externes

- Les marges externes sont situées au-delà des bordures d'un bloc.
- Elles servent à espacer les blocs entre eux.
- Les marges externes sont définies sur chacun des côtés.
- Les éléments en ligne n'ont pas de marge supérieure, ni inférieures (*margin-top = margin-bottom = auto = 0*).
- Il est possible de superposer des blocs en utilisant des valeurs négatives pour les marges externes

Propriétés	<i>margin-left , margin-right, margin-top , margin-bottom</i>
Valeurs possibles	auto : (<i>margin-left = margin-right</i>) □ centrer l'élément taille relative (conseillée) taille fixe
Propriété raccourci	margin
Valeurs possibles	une valeur : toutes les marges 2 valeurs : <i>margin-top = margin-bottom, margin-left = margin-right</i> 3 valeurs : <i>margin-top, margin-left = margin-right , margin-bottom</i> 4 valeurs : <i>margin-top, margin-right, margin-bottom, margin-left</i>

Les marges internes

- Les marges internes se trouvent à l'intérieur des bordures d'un bloc.
- Leur présence évitent que le contenu ne soit pas collé au cadre du bloc qui le contient.
 - Les marges internes ne peuvent pas être négatives.
 - Les éléments en ligne n'ont ni marge supérieure ni inférieure.
 - La valeur par défaut des marges internes est 0.
 - La valeur auto n'existe pas.

Propriétés	<i>padding-left , padding-right, padding-top , padding-bottom</i>
Valeurs possibles	taille relative (conseillée) taille fixe
Propriété raccourci	padding
Valeurs possibles	une valeur : <i>toutes les marges</i> 2 valeurs : <i>padding-top = padding-bottom, padding-left = padding-right</i> 3 valeurs : <i>padding-top, padding-left = padding-right , padding-bottom</i> 4 valeurs : <i>padding-top, padding-right, padding-bottom, padding-left</i>

Largeur d'un élément

- La propriété **width** fixe la largeur disponible pour le contenu d'un bloc de texte ou pour une image.
- Elle ne comprend pas les marges internes et externes ni l'épaisseur de la bordure.
- Les valeurs de width ne peuvent pas être négatives.

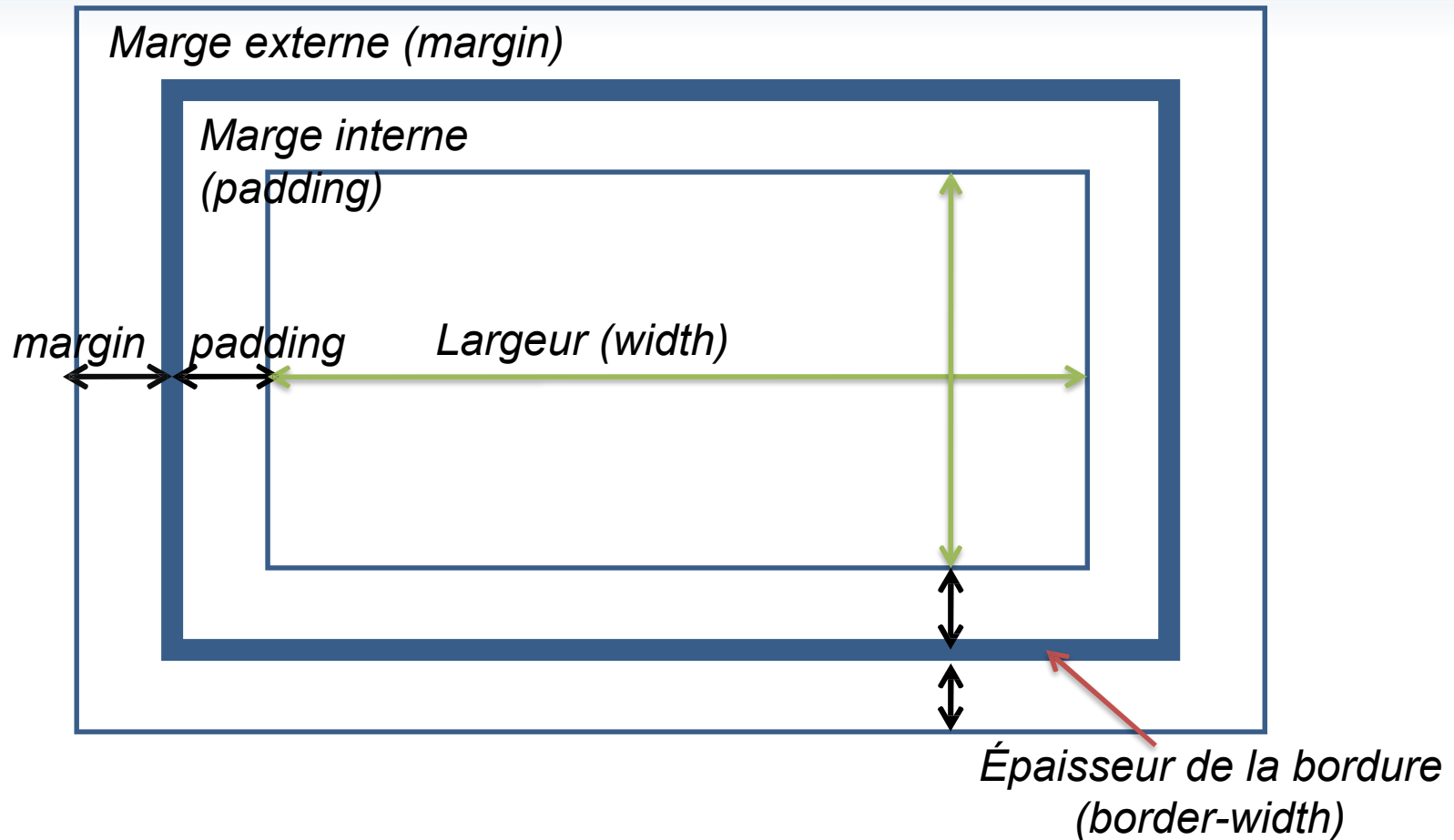
Propriété	<i>width</i>
Valeurs possibles	auto (par défaut) taille relative (conseillée) taille fixe
Autres propriétés	<i>max-width, min-width</i>
Valeurs possibles	none (= 0 : valeur par défaut) taille relative (conseillée) taille fixe

Hauteur d'un élément

- La propriété **height** fixe la hauteur disponible pour le contenu d'un bloc de texte ou pour une image.
- Elle ne comprend pas les marges internes et externes ni l'épaisseur de la bordure.
- Les valeurs de height ne peuvent pas être négatives.

Propriété	<i>height</i>
Valeurs possibles	auto (par défaut) taille relative (conseillée) taille fixe
Autres propriétés	<i>max-height</i> , <i>min-height</i>
Valeurs possibles	none (= 0 : valeur par défaut) taille relative (conseillée) taille fixe

Dimensions des boîtes



Largeur totale = $2 * (\text{margin} + \text{border-width} + \text{padding}) + \text{width}$

Hauteur totale = $2 * (\text{margin} + \text{border-width} + \text{padding}) + \text{height}$

Positionnement des éléments

- **Flux normal des éléments**

- ✓ **éléments en ligne** : placés les uns à la suite des autres

- , , ,*

- ✓ **éléments en bloc** : placés les uns en dessous des autres

- <p>, <div>, <h1>, <h2>,*

- Les blocs qui se succèdent dans le flux normal sont séparés entre eux par leurs marges extérieures.

- Chaque bloc peut être placé de différentes façons à l'intérieur de la page web par rapport à :

- ✓ des blocs juxtaposés

- ✓ un endroit précis du bloc qui le contient

- ✓ un emplacement fixe sur la page

HTML : élément « div »

div: élément block pour regrouper plusieurs éléments inline ou block

(souvent : id ou class)

exemples:

```
<div id="sites-amis">
  <h2>Sites amis</h2>
  <ul>
    <li>...</li>
    <li>...</li>
  </ul>
</div>
```

```
<div id="introduction">
  <h2>bla bla</h2>
  <p>bla bla bla</p>
  <h2>bla bla</h2>
  <p>bla bla bla</p>
</div>
```

Associer un **id** ou **class** aux `<div>` et `` pour la présentation avec CSS.

HTML : élément « span »

span: élément inline pour regrouper plusieurs éléments inline

(souvent : id ou class)

exemples:

regrouper du texte:

```
<span class="tel">
  01.49.40.12.34
</span>
<span class="date">
  1/1/2010
</span>
```

Principe du positionnement

- Les blocs de texte et les éléments remplacés (comme images) peuvent être positionnés.
- Pour positionner des éléments en ligne, il faut les transformer en blocs à l'aide de la propriété : **display**
- Tous les éléments HTML ont une valeur *display par défaut*.

Propriété	<i>display</i>
Valeurs possibles	none : élément invisible et dimensions nulles inline block list-item : élément de liste inline-block : élément en ligne remplacé run-in : bloc ou élément table : tableau inline-table : tableau en ligne table-cell : cellule de tableau table-row : ligne du tableau table-column : colonne du tableau table-caption : titre du tableau table-row-group : groupe de lignes de tableau table-column-group : groupe de colonnes de

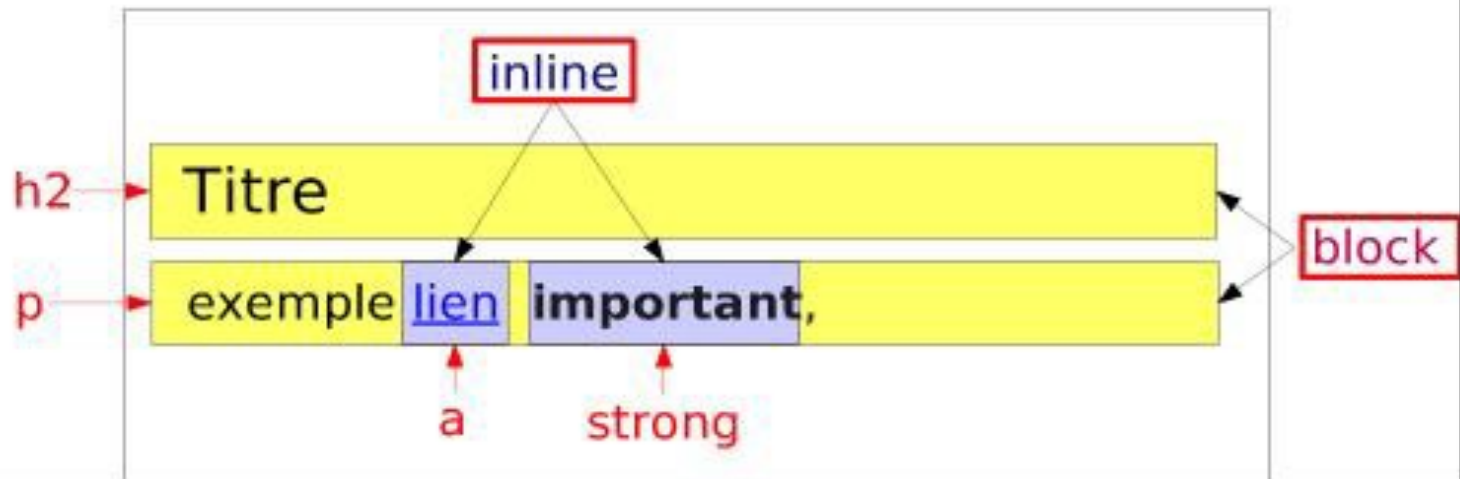
Type de boîte : display

inline: dans le flux du texte
block: occupe toute la ligne

Changer:

```
.lienMenu  
{  
  display: block;  
}
```

inline: a, img, strong, span ...
block: p, h1, div ...



Types de positions

- **Position normale** : un bloc se place dans le flux normal de la page web
- **Position relative, absolue ou fixe** : placer un élément avec un décalage en haut, en bas, à gauche, à droite.
 - ✓ **position relative** : par rapport à sa position dans le flux normal (par rapport à ses frères).
 - ✓ **position absolue** : par rapport au bloc conteneur (découpage de la page).
 - ✓ **position fixe** : par rapport à l'écran (ne bouge pas lors de défilement).

Propriété	<i>position</i>
Valeurs possibles	static (par défaut) relative absolute fixed

Types de positions

- Le positionnement utilise les décalages haut, bas, gauche et droit.

Propriété	<i>top , bottom , left , right</i>
Valeurs possibles	none (pas de décalage, valeur par défaut) relative fixe

Exemple:

```
p.Note
{
  position : relative;
  top      : 5px;
  left     : 10px;
}
```

■ Position flottante :

- placer des éléments côte à côte en fonction de la place disponible dans la fenêtre ou le conteneur.
- Chaque élément flotte vers le côté (vers droite ou vers gauche), mais est séparé de celui qui le précède ou le

Propriété	<i>float</i>
Valeurs possibles	none (valeur par défaut) left (élément décalé à gauche) right (élément décalé à droite)

- La propriété **clear** permet d'interrompre l'alignement de texte provoqué par « **float** ».

Propriété	<i>Clear</i>
Valeurs possibles	none (éléments flottants à droite et à gauche) left (pas d'éléments flottant sur la gauche) right (pas d'éléments flottant sur la droite) both (aucun élément flottant)

Mode de recouvrement « vertical »

- Les blocs s'empilent dans la page web suivant l'ordre de leur arrivé dans le code HTML.
- Cet arrangement peut être modifié grâce à la propriété **z-index**.

Propriété	<i>Z-index</i>
Valeurs possibles	auto (valeur par défaut) nombre entier (positif, nul ou négatif)

- Plus la valeur de z-index est élevée, plus le bloc se trouve en haut dans la superposition des blocs.

- Lorsqu'un contenu déborde du bloc dans lequel il est inclus, il peut être :
 - débordement **visible** : recouvre le contenu qui le suit.
 - débordement **masqué** : rend le contenu débordé invisible.
 - débordement de **défilement** : le contenu est accessible par une barre de défilement.

Propriété	<i>overflow</i>
Valeurs possibles	visible (valeur par défaut) hidden scroll auto (scroll apparaît en cas de débordement)

Le CSS 3 : quelques nouveautés

Des nouveautés.....

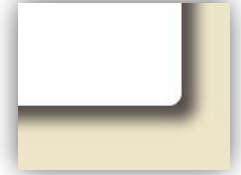
❑ border-radius

```
border-radius: 30px;
```



❑ box-shadow

```
box-shadow: 10px 10px 10px 1px ;
```



❑ text-shadow

```
text-shadow: 2px 2px 5px #444 ;
```

Preview Area

❑ Multiple Columns

```
column-count: 2;
column-gap: 10px;
```

Lorem Ipsum is simply dummy text of the printing and typesetting industry. leap into electronic typesetting, remaining essentially unchanged. It was

❑ Gradients <http://www.colorzilla.com/gradient-editor/>



```
background: linear-gradient(left,
  rgba(208,228,247,1) 0%, rgba(115,177,231,1)
  24%, rgba(10,119,213,1) 50%, rgba(83,159,225,1)
  79%, rgba(135,188,234,1) 100%);
```

