

# *Les feuilles de style CSS dans la conception de pages Web*

*Bertrand LE GAL*

*Filière RSI – 2<sup>ème</sup> année*

*ENSEIRB – 2006/07*

[MCours.com](http://MCours.com)

# Plan du cours

- I. Introduction à la problématique**
- II. Le concept des feuilles de style**
- III. Définition des styles**
- IV. Et les classes dans tout cela...**
- V. Mise en oeuvre des feuilles de style**
- VI. Sélection dynamique des feuilles de style**
- VII. Conclusion**

# La Problématique de la Mise en Forme en HTML

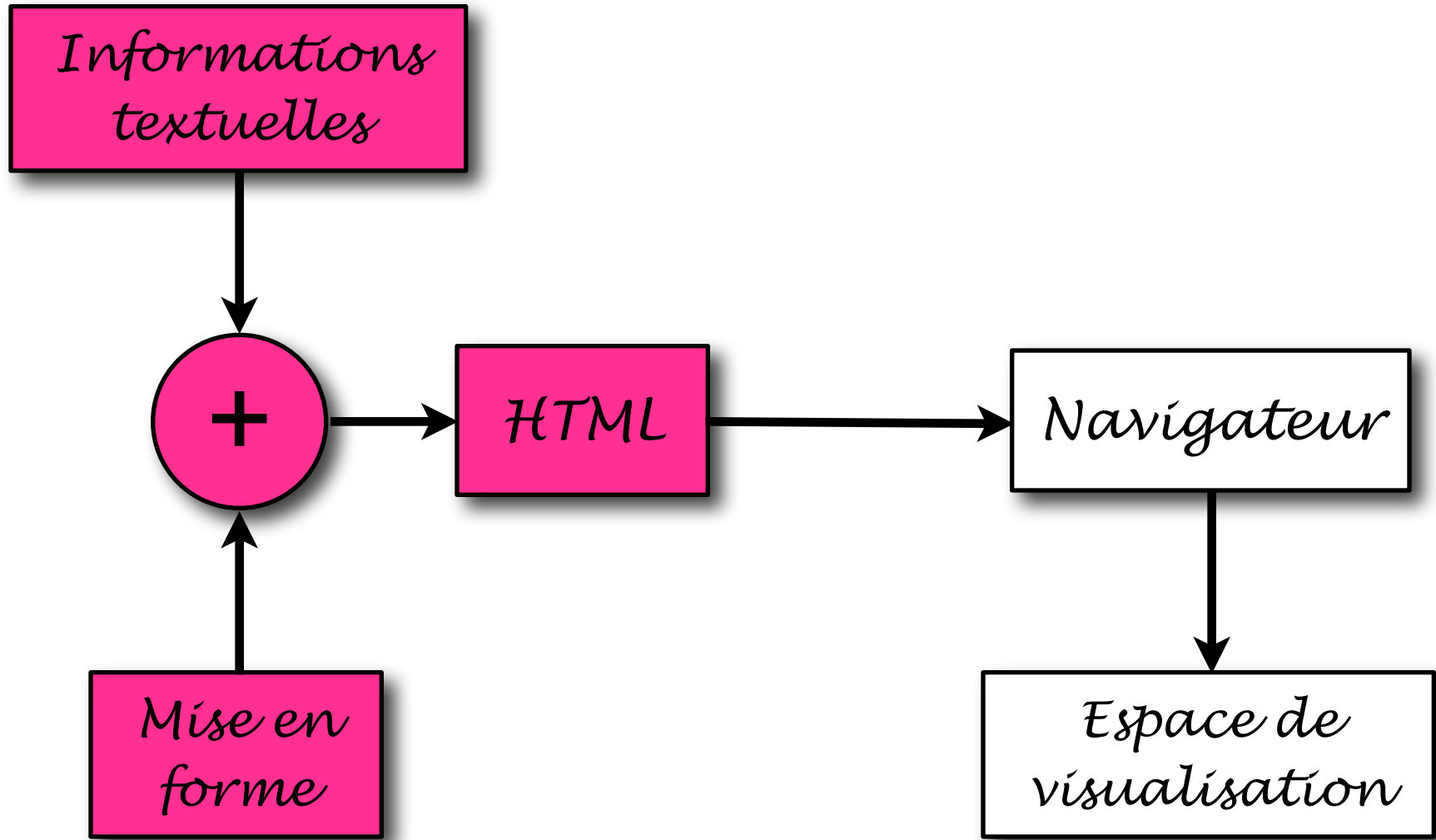
---

Bertrand LE GAL  
Filière RSI 2ème année - ENSEIRB  
2006/2007

# Problématique de la mise en forme

- Nous avons précédemment vu dans le cours dédié au langage *HTML* comment la mise en forme et le fond d'un document sont entrelacés.
- Cet entrelacement pose des problèmes pour s'adapter à différents scénarios :
  - ↪ Evolution et uniformisation de la charte graphique d'un site internet,
  - ↪ Adaptation de la mise en forme de la page aux capacités et aux spécificités des navigateurs internet (*IE, Firefox, Camino, Safari, Konqueror, Opera, etc...*),
  - ↪ Gestion des terminaux mobiles se connectant à internet (taille des écrans souvent réduite).

# La mise en forme classique en HTML



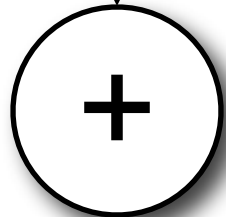
# Conclusion sur ce type d'approche

- Le fond du document (son contenu) est lié trop directement aux informations de mise en forme (les balises de présentation),
- Il est nécessaire de découper (casser) cette relation existante afin d'augmenter la flexibilité de la mise en page et ses possibilités d'évolution futures,
  - ↪ Définition de la notion de feuilles de style (CSS),
  - ↪ Les données dans un fichier / les informations de présentation dans un autre,
- Le nouveau flot permettant l'intégration de la présentation au travers de feuilles de styles et présenté sur la planche suivante.

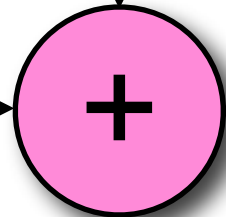
# Exploitation des feuilles de style

Informations  
textuelles

Description des  
styles



HTML



Navigateur

Mise en forme  
générique

Espace de  
visualisation

# Rapide historique du langage CSS

- 1996 : Norme **CSS 1**

- ↳ Définition d'un standard permettant de préciser les règles de présentation des données contenues dans une page *HTML* (couleurs, polices, emplacement, etc.),

- 1998 : Norme **CSS 2**

- ↳ Extension de la norme afin de gérer l'ensemble des éléments présents dans *HTML*. Le contrôle du style est plus précis et la liste des attributs plus complète.

- 1999 : **XSL** e**X**tensible **S**tylesheet **L**anguage

- ↳ Feuille de style associée à un document *XML* et qui permet de modifier sa présentation (plus puissant que le *CSS*).

- ↳ Nous aborderons ce point dans le cours dédié à *XML/XSLT*.



# Introduction aux Feuilles de Style CSS

---

Bertrand LE GAL  
Filière RSI 2ème année - ENSEIRB  
2006/2007

# Problématique de la mise en forme

- Le concept de feuilles de style n'est pas une nouveauté à proprement parler dans le domaine de la publication des données.
  - ↳ Ces concepts existent depuis longtemps dans les traitements de textes par exemple : *LateX*, *Word*, etc.
- Elles sont apparues avec *IE 3*, mais n'étaient pas supportées par *Netscape 3*.
- Depuis, la norme *HTML 4.0* a repris le concept de feuilles de style et recommande activement son usage,
  - ↳ Langage CSS, version 1. [MCours.com](http://MCours.com)

# Concepts de feuilles de style

- A chaque bloc d'informations à mettre en forme, on va associer une étiquette pour la définir. Les étiquettes (format de mise en forme) seront quant à elles spécifiées ailleurs.
- Cela remplace la traditionnelle mise en forme durant laquelle on définit la police de caractères, la couleur, la taille, etc...
- L'ensemble des étiquettes associé au texte devra être défini par la suite afin de permettre une mise en forme de la page dans le navigateur.

# Possibilités offertes par CSS

- Caractéristiques du Texte
  - ↪ Couleur (texte, fond d'écran),
  - ↪ Police de caractères (famille, taille, style...),
  - ↪ Espacement (lettres, mots...),
  - ↪ Positionnement,
- Catégorie d'affichage
  - ↪ Dans un bloc comme un titre H1 ou un paragraphe,
  - ↪ Dans une liste(position, images, types...),
  - ↪ Dans un tableau,
- Caractéristiques liées aux différents médias
  - ↪ Styles adaptés à une restitution audio,
  - ↪ Définition de mise en page pour impression.

# Exemple de présentation liée aux données

```
<H1><B><FONT COLOR="blue" >Titre1 </FONT></B></H1>  
<H2><B><FONT COLOR="green">Point A</FONT></B></H2>  
<H3><B><FONT COLOR="red" >Idée a </FONT></B></H3>  
<H1><B><FONT COLOR="blue" >Titre2 </FONT></B></H1>  
<H2><B><FONT COLOR="green">Point B</FONT></B></H2>  
<H3><B><FONT COLOR="red" >Idée b </FONT></B></H3>
```

**Titre 1**

**Point A**

**Idée a**

**Titre 2**

**Point B**

**Idée b**

# Exemple de présentation liée aux données

```
<GrandTitre>Titre1 </GrandTitre>
<TitreMoyen>Point A</TitreMoyen>
<PetitTitre> Idée a </PetitTitre>
<GrandTitre>Titre2 </GrandTitre>
<TitreMoyen>Point B</TitreMoyen>
<PetitTitre> Idée b </PetitTitre>
```

```
GrandTitre =
  <H1><B><FONT COLOR="blue">
TitreMoyen =
  <H2><B><FONT COLOR="green">
PetitTitre =
  <H3><B><FONT COLOR="red">
```

```
Titre 1
Point A
Idée a
Titre 2
Point B
Idée b
```

# Avantages de cette nouvelle approche

- Séparation franche du contenu et de la mise en forme,
- Cohérence possible dans l'ensemble du site internet grâce à une feuille de style unique pour toutes les pages,
- Le code HTML devient lisible et compréhensible par tout le monde, le code est plus concis,
- Réduire le temps de chargement des pages (quand le nombre de balises de mise en forme est important),
- Possibilité de changer le style du site internet à la demande de l'utilisateur (changement du fichier CSS).

# Cascading Style Sheets - CSS

- Le *CSS* porte ce nom à cause de sa capacité à réaliser des opérations d'héritage permettant une mise en cascade des styles,
- Il existe plusieurs niveaux fichiers *CSS* présents lors le l'affichage d'une page *WEB*,
  - ↳ Le style *CSS* du navigateur internet par défaut,
  - ↳ Le style *CSS* de l'utilisateur,
  - ↳ Le style *CSS* du site internet visité,
- Chacun de ces niveaux surcharge les définitions réalisées dans les niveaux précédant.



# La définition des styles

---

Bertrand LE GAL

Filière RSI 2ème année - ENSEIRB

2006/2007

# Sémantique de définition des styles

- La définition d'un style se présente comme cela est spécifié ci-dessous :

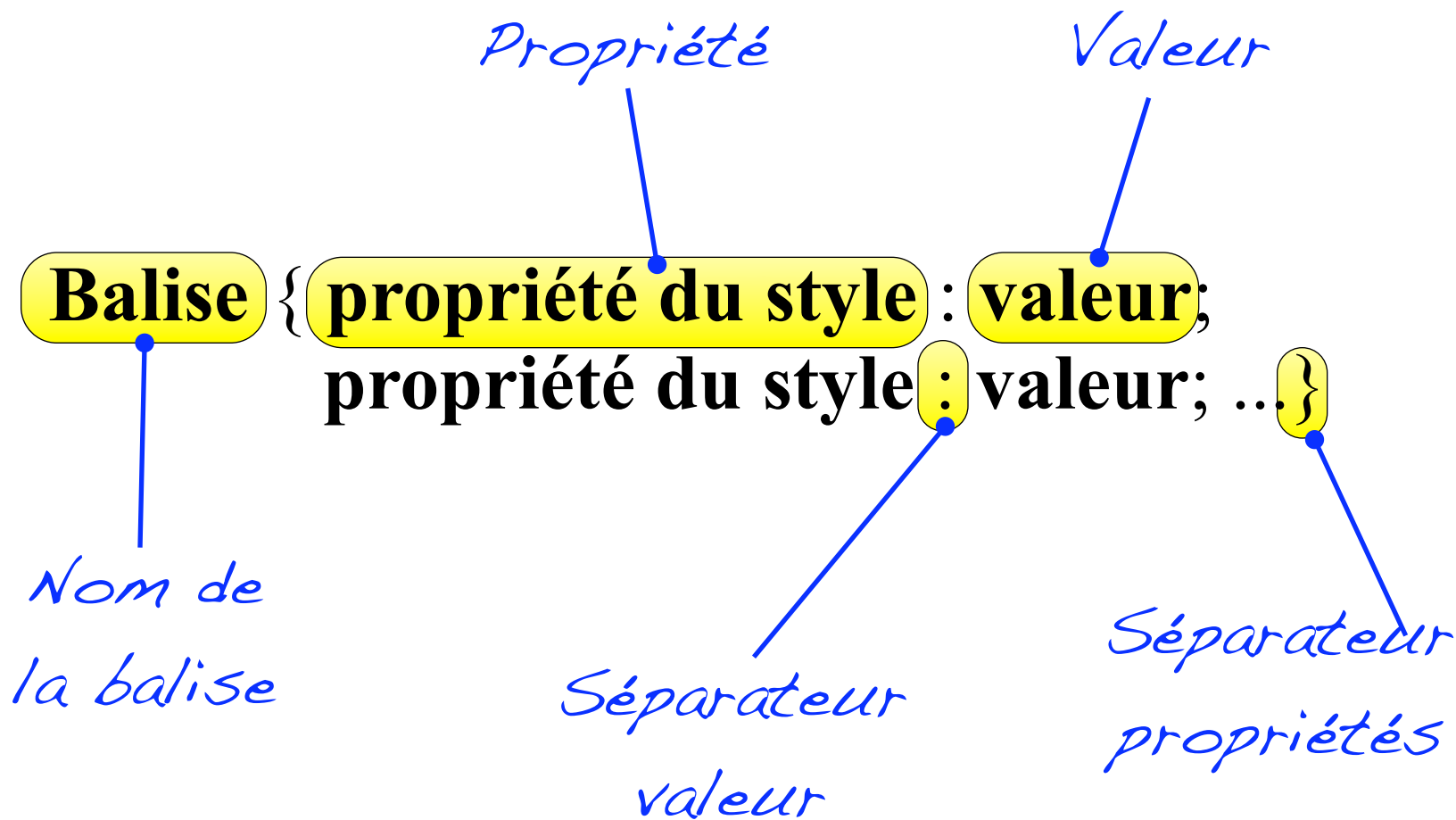
*Balise { propriété du style : valeur;  
propriété du style : valeur; ...}*

- Exemple de mise en forme des titres

*H1 { font-family: Arial;  
font-style: bold;  
font-color=green }*

- ↪ Cette définition permet de spécifier que tous les titre **<H1>** seront en gras et de couleur verte.

# Définition d'une balise (résumé)



# Propriétés des balises CSS

- On peut affecter plusieurs valeurs à une même propriété en CSS :

*H3 { font-family: Arial, Helvetica, sans-serif }*

- On peut associer un même style à plusieurs balises en même temps :

*H1, H2, H3 { font-family: Arial; font-style: italic }*

- On peut déclarer une balise en plusieurs fois :

*H1, H2, H3 { font-family: Arial; font-style: italic }*

*H1 { font-color: blue }*

*H2 { font-color: green }*

# Syntaxe des propriétés / valeurs

- Indifférence des propriétés à la casse des caractères,
- Pour les valeurs certains mots clef peuvent être utilisés, par exemple : *none*, *auto*, etc...
- Pour les valeurs numériques, l'unité peut être précisée :
  - ↪ Valeur absolue, Milli/Centi-mètre (*mm*, *cm*), Inch (*in*), Point (*pt*), etc...
  - ↪ Valeur relative : *em* (taille de la fonte), *px* (pixel)...
  - ↪ Pourcentage,
    - ⇒ Exemple : *H1 { font-size: 12pt; line-height: 14pt; }*

# Syntaxe des propriétés / valeurs

- Certaines propriétés regroupent un ensemble de propriétés individuelles

↪ Par exemple la propriété **font** regroupe { *font-style*, *font-weight*, *line-height* ... },

⇒ Ainsi,

*H1 { font: bold 12pt/14pt helvetica }*

⇒ Est équivalent à :

*H1 { font-weight: bold; font-size: 12pt; line-height: 14pt; font-family: helvetica }*

```
body {
  font: 8pt/16pt georgia;
  color: #555753;
  background: #fff url(blossoms.jpg) no-repeat bottom right;
  margin: 0px;
}
p {
  font: 8pt/16pt georgia;
  margin-top: 0px;
  text-align: justify;
}
h3 {
  font: italic normal 12pt georgia;
  letter-spacing: 1px;
  margin-bottom: 0px;
  color: #7D775C;
}
a:link {
  font-weight: bold;
  text-decoration: none;
  color: #B7A5DF;
}
a:visited {
  font-weight: bold;
  text-decoration: none;
  color: #D4CDDC;
}
a:hover, a:active {
  text-decoration: underline;
  color: #9685BA;
}
```

# Liste des propriétés définies en CSS

## ■ Propriétés liées au texte (1/3)

### ↪ Couleurs : **color** et **background**

- ⇒ **color** pour le premier plan (le texte en général) défini par valeur ou par code RGB
- ⇒ **background** pour le fond (couleur ou image). Pour une image, on peut préciser la position, la répétition...
- ⇒ Exemples

*BODY : { background : url(/images/fondhiver.jpg) blue }*

*P.attention { color: red ; background:yellow }*



# Liste des propriétés définies en CSS

## ■ Propriétés liées au texte (2/3)

MCours.com

### ↳ Polices : **font**

- ⇒ **font-family** définit la famille. Il existe 5 classes de polices génériques : *serif*, *sans-serif*, *monospace*, *cursive* et *fantasy*,
- ⇒ **font-style** définit le style : *normal*, italique (*italic*) ou *oblique*,
- ⇒ **font-variant** définit aussi le style : *normal* ou petites capitales (*small-caps*),
- ⇒ **font-size** définit la taille en absolu ou en relatif par rapport à la taille courante,
- ⇒ **font-weight** définit l'épaisseur (*bold*, *bloder* ou *100*, *200...900*).

# Liste des propriétés définies en CSS

## ■ Propriétés liées au texte (3/3)

↳ Texte :

- ⇒ Définition de l'espacement inter-lettres (*letter-spacing*),
- ⇒ Définition de l'espacement inter-mots (*word-spacing*),
- ⇒ de l'alignement (*text-align*), de l'indentation (*text-indent*),
- ⇒ de la hauteur des lignes (*line-height*), etc.

# Liste des propriétés définies en CSS

## ■ Propriétés de positionnement

- ↪ Le modèle propose la définition de rectangles ("*boxes*") ayant une marge avec à l'intérieur une bordure optionnelle, un "*bourrage*" optionnel aussi et enfin le contenu de l'élément (par exemple le texte)
  - ⇒ **Définition des marges** : *margin*, *margin-bottom*, *margin-left*, *margin-right*,
  - ⇒ **Définition du "bourrage" interne** : *padding*, *padding-bottom*, *padding-left*, *padding-right*,
  - ⇒ **Définition des bordures** : *border*, *border-bottom*, *border-left*, *border-right*,
  - ⇒ **Définition de la hauteur** : *height* et de la largeur : *width* du rectangle
  - ⇒ **Définition du "flottement"** d'un élément par rapport au texte : *float*

# Autres propriétés ...

- Catégorie d'affichage d'un élément :
  - ↪ **table** : dans une table et on peut préciser le titre (*table-caption*), une ligne (*table-row*)...
  - ↪ **none** : pas d'affichage
- Caractéristiques des listes :
  - ↪ *list-style-type* définit le types des items (*disc*, *circle*, *upper-latin*, *decimal*...),
  - ↪ *list-style-image* définit des images pour les items,
  - ↪ *list-style-position* définit la position (*inside*, *outside*).
- Caractéristiques liées aux différents médias
  - ↪ Définition de styles adaptés à une restitution audio,
  - ↪ Définition de mise en page pour impression.

# Mise en oeuvre des styles

---

Bertrand LE GAL

Filière RSI 2ème année - ENSEIRB

2006/2007

# Que faire une fois les styles définis

- Il faut maintenant incorporer la définition des styles dans la page HTML afin de pouvoir les utiliser,
- Il existe plusieurs manières de procéder afin d'utiliser un style dans une page HTML :
  - ↳ *Définition locale* : les styles CSS sont définis directement dans la page HTML avant d'être utilisés,
  - ↳ *Définition externe* : les styles CSS sont définis dans un fichier externe qui est "chargé" par le fichier HTML lors de sa visualisation par un navigateur,
- La seconde méthode est à privilégier, mais nous allons aborder tout de même les 2 approches.

# Les définitions locales

- La méthode la plus simple est d'inclure les balises *CSS* directement dans le corps de la page *HTML*,
- Les informations liées au style des données sont alors incluses dans les balises d'en-tête de la page *HTML*,
  - ↪ Cela permet de bien différencier les informations de leur mise en forme,
  - ↪ Chaque page peut alors posséder sa propre mise en page,
- Pour ajouter des styles dans une page *HTML*, il faut utiliser la balise **<STYLE>**.

# Exemple de présentation liée aux données

*Indique que  
l'on va définir  
des styles*

```
<HTML>
  <HEAD>
    <STYLE type="text/css">
      <!--
        La ou les feuille(s) de style
      -->
    </STYLE>
  </HEAD>
</BODY>
```

*On définit l'ensemble des  
styles que l'on va utiliser*



# Exemple de présentation liée aux données

- L'attribut **“text/css”** spécifie au navigateur que les données qui suivent sont de type texte et qu'elles correspondent à des données de type *CSS*,
- Les balises de commentaire `<!-- -->` sont ajoutées afin de s'assurer que le navigateur n'essayera pas d'interpréter tout ou partie du code *HTML*,
- Les commentaires dans les feuilles *CSS* doivent être encadrés par les tags **“/\*”** et **“\*/”** comme en *JAVA/C++*.

# Les définitions globales

- Afin de profiter pleinement des styles CSS définis pour mettre en forme l'ensemble du site internet, il est possible de réaliser une ou des inclusions.

```
<HTML>
  <HEAD>
    <LINK rel=stylesheet type="text/css" href="styles.css">
  </HEAD>
<BODY>
```

*Fichier contenant les styles  
que l'on souhaite utiliser*

# Explication des paramètres utilisés

- La balise **<LINK>** précise au navigateur qu’il va rencontrer un lien sur un fichier externe (qu’il va devoir télécharger),
  - ↳ L’attribut **“rel”** précise le type de contenu pointé par le lien,
  - ↳ L’attribut **“type”** précise comme tout à l’heure que l’on va avoir à faire à du *CSS*,
  - ↳ L’attribut **“href”**, précise pour sa part la localisation du fichier contenant les styles *CSS*.

# Et les classes dans tout cela...



Bertrand LE GAL  
Filière RSI 2ème année - ENSEIRB  
2006/2007

# Qu'est ce qu'une classe ?

- Les styles sont des éléments pratiques qui permettent s'assurer de la cohérence du style graphique du site,
- Mais la définition des styles pose un problème de taille, comment faire pour changer les attributs d'une partie des éléments ?
  - ↳ *La première solution* est de créer un type de balise par type de donnée,
  - ↳ *La seconde solution* est proposée par la norme CSS avec *l'introduction des classes*,
- Ces dernières permettent d'associer à un type de balise, plusieurs styles en fonction des besoins.

# Sémantique de définition des styles

- La définition d'un style se présentait ainsi :

*Balise { propriété du style : valeur;  
propriété du style : valeur; ...}*

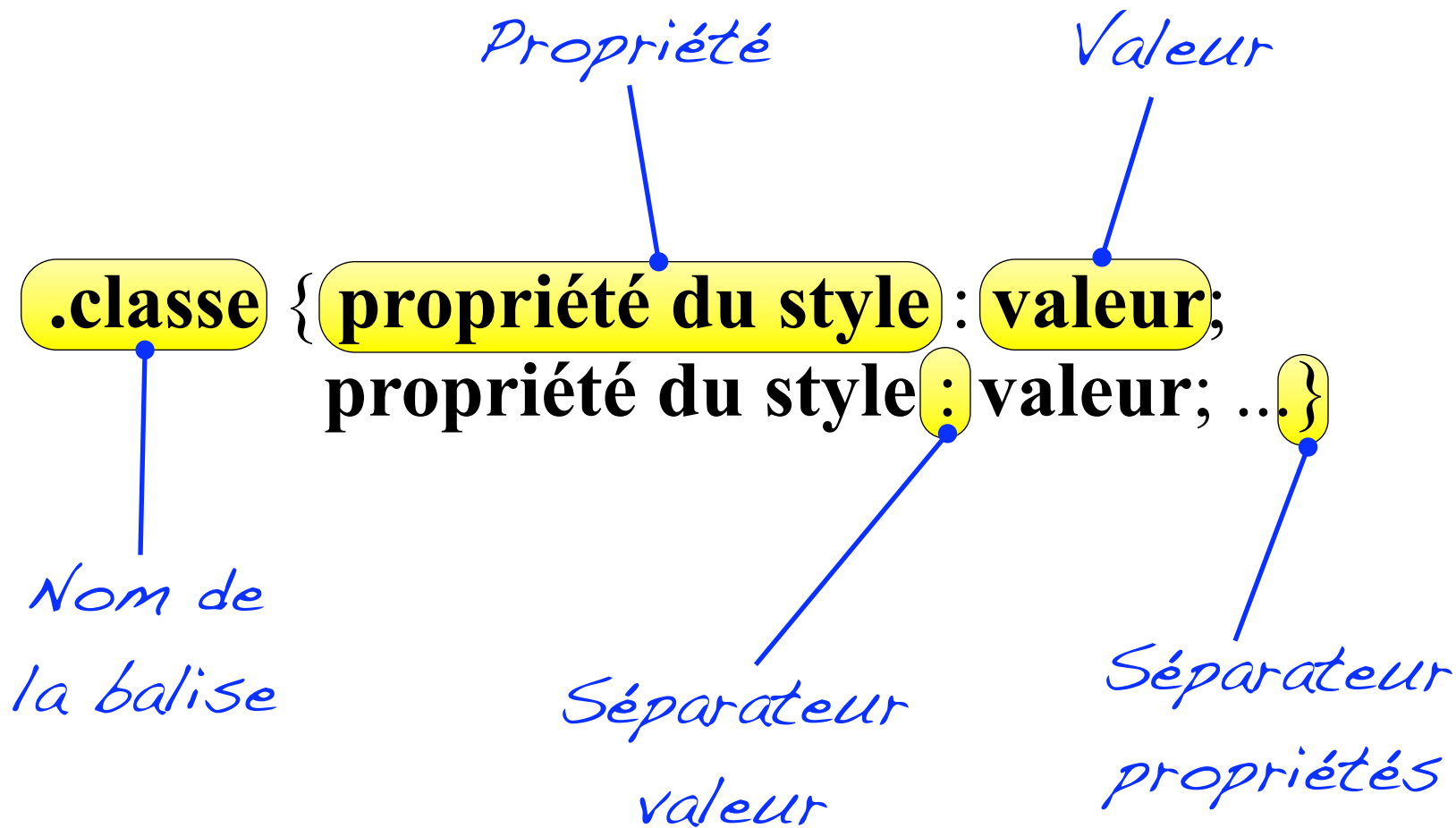
- Elle a maintenant évoluée comme décrit ci-dessous :

*Balise.**nom\_de\_classe** { propriété du style : valeur;  
propriété du style : valeur; ...}*

- La balise est facultative et la déclaration peut devenir :

***.nom\_de\_classe** { propriété du style : valeur;  
propriété du style : valeur; ...}*

# Définition d'une balise (résumé)



# Définition d'une balise (résumé)

```
.ma_classe { font-weight: bold;  
             font-color: #000080 }  
.important { font-weight: italic;  
            font-color: #008000 }
```

```
<H1 CLASS=ma_classe>Le Titre</H1>
```

```
<P CLASS=important>
```

Voici un bref résumé du document que vous allez lire...

```
</P>
```

```
<P CLASS=ma_classe>
```

La classe peut être réutilisée !

```
</P>
```



# Les IDs dans la norme CSS

- Afin d'utiliser les classes *CSS* dans les programmes *Javascripts* qui utilisent aussi les “.” pour gérer les attributs, il a fallu trouver une parade.
- Les IDs fonctionnent de manière équivalente aux classes, la syntaxe devient alors :

```
#nom_de_ID { propriété du style : valeur;  
                propriété du style : valeur; ... }
```

- Pour l'utiliser, il suffit d'écrire :

```
<balise id="nom_de_ID"> ... </balise>
```

# Les IDs dans la norme CSS

- Notons qu'il ne sera possible d'effectuer qu'un seul appel à `#nom_de_ID` par page internet :
- Pour `#ma_classe { ... }`  
`<P id=ma_classe>` est valide, [MCours.com](http://MCours.com)
- Mais si, dans la même page, on rencontre une nouvelle utilisation de “`ma_classe`”, tel :  
`<H3 id=ma_classe>` cette déclaration est incorrecte,
- En cas d'absence d'utilisation des classes dans du JavaScript, les IDs doivent être oubliés.

# Modification du champ d'application

- Les sélecteurs permettent de spécifier plus finement les champs d'application des styles définis à l'aide du langage *CSS*,
- Des balises *HTML*
  - Les balises H1 : *H1 {...}*
- Des balises *HTML* ayant certains attributs
  - Les balises HR de largeur égale à 50%de la fenêtre :  
*HR [WIDTH="50%"] {...}*
- Des balises *HTML* d'une classe donnée
  - Les balises H1 de classe intro (<H1 class=intro>) : *H1.intro {...}*

# Modification du champ d'application

- Une balise spécifique pour traiter un cas particulier d'un ensemble,
  - ↳ La balise *H1* d'identité plan ( $\langle H1 ID=plan \rangle$ )  
*H1#plan {..}* ou *#plan {..}*
- Utilisation des *Jokers* (\*) en CSS
  - ↳ Toutes les balises ayant l'attribut *LANG* positionné,  
*\*[LANG] { ... }*
  - ↳ Toutes les balises *HTML* de classe “important”,  
*\*.important {...}*
- Des pseudo-classes qui prennent en compte certains aspects dynamiques (clicks...).

# Les pseudo-classes en CSS

- Caractérisées par un mot-clef séparé par deux points “:” des balises *HTML* :

↪ *Balise\_HTML:Mot\_Clef*

↪ Exemple de la balise `<a>` pour les liens,

- Exemples de pseudo-classes,

↪ Survol d'une zone (active) par la souris : *A:hover {...}*

↪ Sélection d'une information : *A:active {...}*

↪ Activation d'un champ de saisie : *INPUT:focus {...}*

↪ Test de l'attribut *LANG* (fonction) : *BODY:lang(fr) {...}*

# La cas particulier de la balise <A>

- La balise lien est composée de plusieurs pseudo-classes par défaut reflétant l'état du lien :
  - ↪ **Le lien** : dans son état par défaut (non visité et non actif) *{A:Link}*,
  - ↪ **Visité** : Le site internet a déjà été visité dans le passé par le navigateur *{A:Visited}*,
  - ↪ **Actif** : Le site internet est actuellement actif (chargé dans une autre fenêtre/onglet) *{A:Active}*,
  - ↪ **MouseOver** : La souris se trouve à l'heure actuelle au dessus du lien *{A:Hover}*.

# Exemple de définition de la balise <a>

```
A:link { font-family: Arial  
          font-weight: bold;  
          font-size: 12px }
```

```
A:visited { font-family: Arial  
             font-weight: bold;  
             font-size: 12px;  
             font-color: blue }
```

```
A:active { font-color: red }
```

```
A:hover { font-family: Arial  
           font-weight: italic;  
           font-color: yellow }
```

# Utilité des balises <DIV> et <SPAN> avec CSS

---

Bertrand LE GAL  
Filière RSI 2ème année - ENSEIRB  
2006/2007



# Problématique de la mise en forme

- Les balises **<DIV>** et **<SPAN>** peu utiles en *HTML* pur permettent de découper des parties d'une même page et/ou d'un même paragraphe.
- Ces balises vont permettre de découper une page en différents blocs logiques tels :
  - ↳ Un résumé,
  - ↳ Des paragraphes,
  - ↳ Une conclusion,
- Chaque sous partie du document va être englobé dans une balise **<DIV CLASS>** adéquate.

# Mise en oeuvre des balises

- La balise **<DIV>** est utilisée pour remplacer la balise **<P>** des paragraphes,

**<DIV CLASS="résumé">**Voici le résumé du document que vous allez lire... blabla**</DIV>**

**<DIV CLASS="texte">**Voici maintenant le document dont le résumé parlait...**</DIV>**

- La balise **<SPAN>** a été développée afin de permettre la mise en forme de petit blocs de texte, contenus au sein même d'un paragraphe,

Disons que dans mon résumé, certains mots sont plus **<SPAN CLASS=important>**importants**</SPAN>** que les autres ?!

# Exemples de CSS (Effets simples et Complexes)

---

Bertrand LE GAL  
Filière RSI 2ème année - ENSEIRB  
2006/2007

# Sites proposant des exemples

- Pour des exemples de mise en oeuvre avancée du langage CSS, je vous conseille les sites internet suivants :

➤ <http://www.csszengarden.com/tr/francais/>

➤ <http://batraciens.net/css-astuces/index.htm>

➤ <http://matchab.free.fr/index.php>

➤ <http://www.publishtogether.com>

# Conclusion

---

Bertrand LE GAL  
Filière RSI 2ème année - ENSEIRB  
2006/2007

# Conclusion sur le langage CSS

- L'utilisation des feuilles de style *CSS* est devenue indispensable pour gérer une mise en page uniforme d'un site internet (charte graphique),
- Possibilité de mettre en oeuvre des effets graphiques sur les menus, le texte, les liens, les images, les tableaux ...
- Le langage est limité tout de même à la mise en forme "simple" des données vis-à-vis de *XSL* ...
  - ↳ Pas de transformation possible des données à afficher...

- **The World Wide Web Consortium (W3C)**, Le site de toutes les normes, <http://www.w3.org/>.
- **Maitriser le langage HTML**, Van Lancker Luc, Aout 2000, Partie dédiée au CSS, <http://www.ccim.be/ccim328/htmlplus/idxstyle.htm>
- **Cours en ligne de CSS**, Gilles Chagnon, Septembre 2006, <http://gilles.chagnon.free.fr/cours/html/>
- **Les feuilles de style CSS**, Dominique Bouillet, Avril 2001, <http://etna.int-evry.fr/COURS/INTERNET/css/>
- **Liste des propriétés CSS**, <http://stylescss.free.fr/>