



Lyon 1



département
Informatique

Université Claude Bernard Lyon 1

Licence Math-Informatique 1^{ère} année

Partie 2

Olivier Glück

Université LYON 1 / Département Informatique

Olivier.Gluck@univ-lyon1.fr

<http://perso.univ-lyon1.fr/olivier.gluck>

Copyright

- Copyright © 2026 Olivier Glück; all rights reserved
- Ce support de cours est soumis aux droits d'auteur et n'est donc pas dans le domaine public. Sa reproduction est cependant autorisée à condition de respecter les conditions suivantes :
 - Si ce document est reproduit pour les besoins personnels du reproducteur, toute forme de reproduction (totale ou partielle) est autorisée à la condition de citer l'auteur.
 - Si ce document est reproduit dans le but d'être distribué à des tierces personnes, il devra être reproduit dans son intégralité sans aucune modification. Cette notice de copyright devra donc être présente. De plus, il ne devra pas être vendu.
 - Cependant, dans le seul cas d'un enseignement gratuit, une participation aux frais de reproduction pourra être demandée, mais elle ne pourra être supérieure au prix du papier et de l'encre composant le document.
 - Toute reproduction sortant du cadre précisé ci-dessus est interdite sans accord préalable écrit de l'auteur.

Plan du cours

- CM1 : Internet, les réseaux et le web
- CM2 : Pages HTML et feuilles de styles CSS
- CM3 : Web interactif, formulaires, pages dynamiques et PHP
- CM4 : Protocole HTTP, méthodes GET et POST
- CM5 : Les applications d'Internet
- CM6 : La couche transport : les protocoles TCP et UDP
- CM7 : Le protocole IP
- CM8 : Les protocoles Ethernet, ARP et ICMP. Synthèse des échanges entre un client et serveur Web



département
Informatique

Université Claude Bernard Lyon 1

CM2 : Pages HTML et feuilles de styles CSS

Principe du HTML, balises et attributs
Les feuilles de styles

Plan du CM2

- Principes du HTML, balises et attributs
 - Principes du HTML, le langage, structure d'un document
 - Les paragraphes, les titres et sauts de ligne
 - Formatage du texte
 - Les listes, les liens
 - Les images
 - Les tableaux
- Les feuilles de styles
 - Principes des feuilles de styles
 - Les classes et sous-classes
 - Positionner des éléments dans une page



département
Informatique

Université Claude Bernard Lyon 1

Principes du HTML, balises et attributs

Principes du HTML, le langage, structure d'un document

Les paragraphes, les titres et sauts de ligne

Formatage du texte

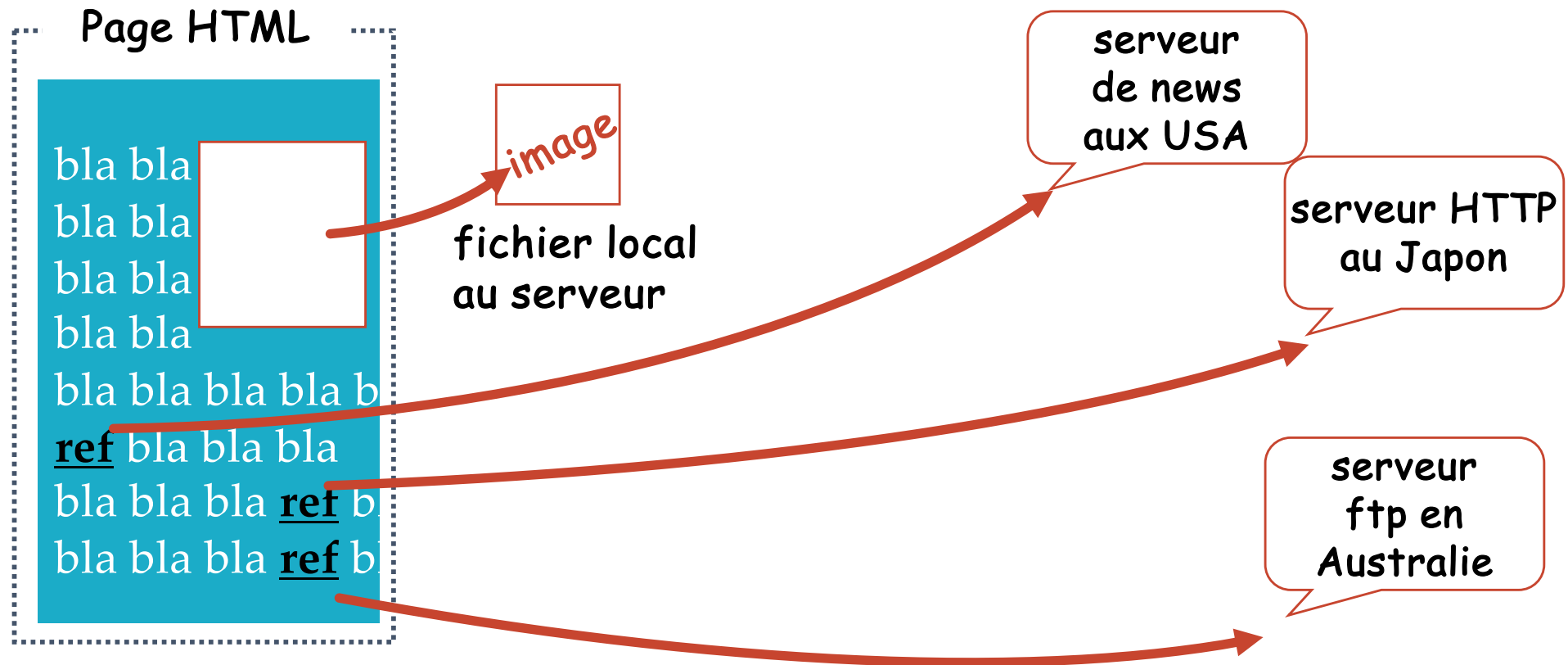
Les listes, les liens

Les images

Les tableaux

Principes du HTML (1)

Ici, 4 liens hyper-textes



Principes du HTML (2)

- Hyper Text Markup Language
- Définit la structure logique d'un document WEB
- Composé d'un ensemble de commandes de formatage
- Basé sur la notion d'**environnement** possédant un début et une fin
 - les **délimiteurs** : on parle de balises ou marqueurs
- Pour mettre en œuvre du HTML, il suffit
 - d'un éditeur de texte pour taper le code de la page,
 - d'un navigateur WEB pour afficher la page formatée.

Le langage HTML (1)

- Les balises sont définies entre `<>`
 - `<BALISE>...</BALISE>`
- La plupart des environnements peuvent être imbriqués selon des règles bien définies
 - `<H1><I>Mon titre</I></H1>`
- Il n'est pas permis de faire chevaucher des environnements
 - `<H1><I>Mon titre</H1></I>`
- Les minuscules/majuscules n'interviennent pas dans la définition des balises

Le langage HTML (2)

- Les balises dénotent des "constructions" documentaires
 - styles de paragraphe (normal, énumérations, titres...)
 - tableaux
 - styles de caractères (gras, italique, souligné...)
 - références à des images
 - références hyper-texte
 - formulaires
 - etc...
- Si aucun document n'est spécifié dans l'URL, le document par défaut est généralement **index.html**

Les règles de "bonne conception"

- **Mise en page et styles** : maintenir une certaine uniformité sur l'ensemble du site
- **Clarté** : éviter la confusion, présenter de façon simple avec des menus, éviter les textes longs... ; l'information doit être trouvée dès les premières secondes
- **Hyperliens** : ne pas en abuser, faire en sorte qu'ils indiquent clairement où ils mènent
- **Identité** : le visiteur doit savoir facilement sur quel site il est (quelle que soit la page visitée) et savoir à qui s'adresser --> logos, @mail, date de dernière mise à jour de la page...
- **Réactivité** : ne pas abuser des images, exécutions de code... (temps de chargement/exécution)
- **Navigateurs** : faire attention à leurs particularités, s'adapter à la taille de la fenêtre, au type du terminal utilisé par le client...

Structure «minimale» d'un document

<HTML>

Respectez l'indentation !

<HEAD>

<TITLE>titre-fenetre**</TITLE>**

</HEAD>

<BODY>

corps du document

</BODY>

</HTML>



HEAD : pour différencier du reste du texte, contient les titres

TITLE : affiché en haut de la fenêtre, utilisé dans les favoris (*bookmarks*)

BODY : contient le document

Les commentaires : **<!-- ne sera pas interprété -->**

Balise <P>, <Hn> et

- Balise <P>...</P>
 - Début d'un nouveau paragraphe
 - Attribut align="center/left/right/justify" (justify pas toujours interprété)
- Balises <Hn>...</Hn>, n varie de 1 à 6
 - Définissent la taille des caractères (taille décroissante)
 - Texte en caractères gras et suivi d'un saut de ligne
 - Attribut align="center/left/right"
 - <Hn> est considéré comme un paragraphe et ne doit pas être placé dans <P>...</P>
- Balise

 - Saut de ligne

Balises de titres

```
<HTML><HEAD>
```

```
  <TITLE>titre-fenetre</TITLE>
```

```
</HEAD><BODY>
```

```
  <H1>Titre "H1"</H1>
```

```
  <P>Texte sous le titre.</P>
```

```
  <H2>Titre "H2"</H2>
```

```
  <P>Texte sous le titre.</P>
```

```
  <H3>Titre "H3"</H3>
```

```
  <P>Texte sous le titre.</P>
```

```
</BODY></HTML>
```



Balises de formatage de texte (1)

- Balise `<PRE>...</PRE>`
 - Texte pré-formaté
 - Permet de conserver le formatage du texte tel qu'il a été saisi dans le fichier source HTML (respect de la mise en page précise : espacement, saut de ligne...)
 - Utile par exemple pour faire un tableau « manuellement » ou pour afficher un code indenté
- Styles physiques indépendants du navigateur utilisé et de sa configuration (certains sont dépréciés, utiliser les styles)
 - `...` : gras
 - `<I>...</I>` : italique
 - `<TT>...</TT>` : machine à écrire (police largeur fixée)
 - `<STRIKE>...</STRIKE>` : texte barré
 - `<U>...</U>` : souligné
 - `_{...}` : indice
 - `^{...}` : exposant
 - `<SMALL>...</SMALL>` : petite police
 - `<BIG>...</BIG>` : grande police

Balises de formatage de texte (2)

- Styles logiques qui peuvent dépendre du navigateur utilisé et de sa configuration
 - `` : gras
 - `` : mettre un texte en valeur (italique)
 - `<DFN>` : définition
 - `<CITE>` : citation bibliographique
 - `<CODE>` : programme informatique
 - `<BLOCKQUOTE>` : tabulation + espaces paragraphes
 - ...

Les caractères spéciaux

- Pour être sûr qu'ils soient correctement interprétés par tous les navigateurs :

- caractères accentués : **&<lettre><accent>;**

accent : *acute* (aigu), *grave* (grave), *circ* (^), *uml* (¨)

exemples : *é;* (é), *Î;* (Î), *ü;* (ü)

- autres caractères :

ñ; (ñ), *ç;* (ç), *ß;* (ß), *©;* (©)

æ; (æ), *&;* (&), * ;* (espace), *";* (")

>; (>), *<;* (<)

Balises : les listes

■ Non numérotées

 : Unnumbered List

... : List Item

- attribut `type="square/circle/disc"` pour ou

■ Numérotées

...

- attribut `type="1/A/a/I/i"` pour
- attribut `start="valeur"` pour (valeur de départ)
- attribut `value="valeur"` pour (réinitialise le séquençement à la nouvelle valeur)

Balises pour les liens hyper-texte (1)

```
<HTML><HEAD>
```

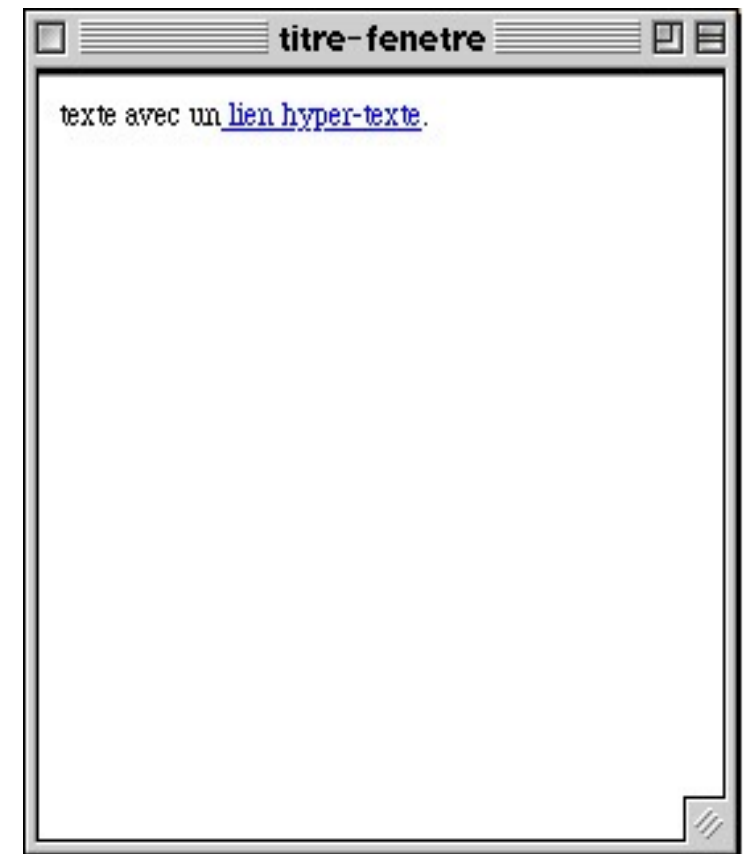
```
  <TITLE>titre-fenetre</TITLE>
```

```
</HEAD><BODY>
```

```
<P>texte avec un <A href="cible.html">
```

```
lien hyper-texte</A>.</P>
```

```
</BODY></HTML>
```



Balises pour les liens hyper-texte (2)

- Définition d'un hyperlien `texte`
 - **href="nom"** Le nom du document à lier peut être :
 - une URL standard : `http://www.univ-lyon1.fr`
 - une adresse mail : `mailto:toto@zozo.fr` (mais spam)
 - un chemin relatif : `Cours/index2.html`
 - un chemin absolu : `/` est la racine du serveur Web !
 - une étiquette : `#LABEL`
 - une combinaison URL, chemin relatif, étiquette
 - **texte** : texte utilisé pour décrire/représenter le lien dans le document
- Définition d'une étiquette
 - `...` (attribut NAME obsolète)
 - le texte entre `<A>` et `` n'est pas une zone cliquable

Balise <BODY> : paramètres de la page

- Fond d'écran

`<BODY background="nom_image">`

- répétition d'une image sur l'écran du navigateur

- Couleurs

`<BODY`

`bgcolor="#000000"` : couleur de fond

`text="#000000"` : couleur du texte

`link="#000000"` : couleur des liens non encore visités

`vlink="#000000"` : couleur des liens déjà visités

`alink="#000000"` : couleur des liens lors du clic

- Noms des couleurs

- en clair : aqua, black, gray, olive...

- en notation hexadécimale RGB

Balise : insertion d'images (1)

```
<HTML><HEAD>
```

```
  <TITLE>titre-fenetre</TITLE>
```

```
</HEAD><BODY>
```

```
<P>Texte sur l'image.</P>
```

```
<P><IMG src="logo.jpg"></P>
```

```
<P>Texte sous l'image.</P>
```

```
</BODY></HTML>
```



Balise : insertion d'images (2)

- Insertion d'une image dans un document :

<IMG src="nom"

- nom local ou URL distante
- extensions : jpeg, gif, xbm ...

<IMG height="hauteur" width="largeur"

- informe le navigateur de la hauteur et la largeur de l'image en pixels (évite au navigateur de les calculer)
 - peut aussi permettre de redimensionner l'image
- Pour gérer l'alignement ou la bordure d'une image, utiliser les styles

Balise : insertion d'images (3)

<IMG vspace="nb_pixels" hspace="nb_pixels"

- tailles des marges (espacement image/texte)

<IMG alt="ceci est une info-bulle"

- définit un commentaire pour l'image
- affiché à la place de l'image pendant le téléchargement
- affiche une info-bulle quand la souris passe sur l'image (certains navigateurs seulement)
- donne des infos aux indexeurs

L'attribut **alt** est aussi utilisable dans d'autres balises

Balises <TABLE>, <TR>, <TH> et <TD>

- Les tableaux permettent de positionner précisément les objets dans le navigateur, de gérer la mise en page
- Les cases (ou cellules) peuvent contenir des données multimédia (textes, images, liens...)
- **<TABLE>...</TABLE>** définit un tableau dans un nouveau paragraphe
- Spécification d'une ligne

<TR>...</TR>

marquent le début et la fin d'une ligne du tableau

- Spécification des cellules
 - Cellules d'en-tête (centré, en gras)

<TH>...</TH>

- Cellules de données

<TD>...</TD>

Les attributs d'un tableau (1)

- Accélérer l'affichage des tableaux
 - Fixer la dimension du tableau
 - par défaut, la dimension est calculée automatiquement par le navigateur en fonction du contenu des cellules et de la taille de la fenêtre
 - la largeur (**width**) et la hauteur (height) peuvent être fixés grâce aux styles

Les attributs d'un tableau (2)

- Utiliser les styles pour gérer
 - le positionnement du tableau dans la page
 - l'alignement du contenu des cellules
 - les bordures du tableau
 - les espacements et marges entre cellules, à l'intérieur des cellules
 - les couleurs du fond et des bords
 - l'image d'arrière plan
 - les ombrages
 - ...

Les attributs d'une ligne/cellule (1)

- Attributs communs aux lignes et cellules
 - alignement horizontal du contenu dans une cellule ou une ligne
 - `<TR align="left | center | right">...</TR>`
 - `<TD align="left | center | right">...</TD>`
 - alignement vertical du contenu dans une cellule ou une ligne
 - `<TR valign="top | middle | bottom">...</TR>`
 - `<TD valign="top | middle | bottom">...</TD>`
 - Couleurs d'arrière plan d'une cellule ou d'une ligne
 - `<TR bgcolor="couleur ou #RRVVBB">...</TR>`
 - `<TD bgcolor="couleur ou #RRVVBB">...</TD>`

Les attributs d'une ligne/cellule (2)

- Largeur et hauteur d'une cellule
 - La hauteur pour une cellule est appliquée à toutes les cellules de la ligne
 - La largeur d'une cellule est appliquée à toutes les cellules de la colonne
 - Il suffit de formater une seule cellule
 - Les tailles sont prises en compte si non contraires à la taille du tableau, sinon le navigateur essaie d'harmoniser
- `<TH width="nb_pixels | %" height="nb_pixels | %">`
- Pour empêcher le passage automatique à la ligne dans la cellule
- `<TD nowrap>`

Les attributs d'une ligne/cellule (3)

- Fusions de cellules

- Pour définir le nombre de colonnes sur lesquelles s'étend une cellule :

`<TH colspan="nombre">...</TH>` : nombre de cellules à fusionner sur une ligne

- Pour définir le nombre de lignes sur lesquelles s'étend une cellule :

`<TD rowspan="nombre">...</TD>` : nombre de cellules à fusionner sur une colonne

- Ces fusions doivent être cohérentes par rapport aux lignes et colonnes du tableau

- Balise `<CAPTION align="left|center|right">`

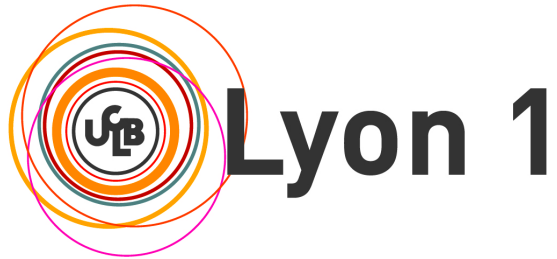
- Permet de donner un titre au tableau

Les autres balises HTML

- Il y a plein d'autres balises HTML !

Consultez par exemple le site de Mozilla

<https://developer.mozilla.org/fr/docs/Web/HTML/Element>



Lyon 1



département
Informatique

Université Claude Bernard Lyon 1

Les feuilles de styles

Principes des feuilles de styles

Les classes et sous-classes

Positionner des éléments dans une page

Principe des feuilles de styles

- Même idée que les styles des traitements de texte
 - Distinction entre la structuration du texte et les propriétés typographiques que l'on applique au texte
 - Les styles permettent d'appliquer certaines propriétés à des balises HTML
 - Plusieurs documents peuvent partager les mêmes instructions typographiques
 - Homogénéité de typographie et de présentation de toutes les pages auxquelles s'appliquent la feuille de styles
 - L'apparence des documents peut facilement être modifiée en utilisant les styles définis dans la feuille de styles

Principe des feuilles de styles

- Les styles peuvent être définis
 - directement dans la page HTML
 - dans un fichier séparé : c'est la feuille de styles !
 - Un style défini directement dans la page HTML est prioritaire sur un style défini dans un fichier externe
- *CSS - Cascading Style Sheets*
 - Mise en cascade des styles
 - Langage déclaratif type HTML qui permet de définir les styles appliqués aux balises dans un fichier externe **.css**

La balise <STYLE>

- Entre <head>...</head>
- L'attribut TYPE
 - `text/css` pour la norme CSS
 - `text/javascript` pour JASS (description JavaScript)
- Exemple (attention à la casse des mots-clés)

```
<html><head>  
<style type="text/css">  
  P {font-size: 16pt; color: blue;}  
</style>  
</head>
```

```
<html><head>  
<style type="text/javascript">  
  tags.P.fontSize=16;  
  tags.P.color="blue";  
</style>  
</head>
```

Les sélecteurs et classes

- Mettre en forme des éléments HTML avec un sélecteur
 - Un sélecteur associé à un élément HTML particulier
 - P** {font-size: 16pt; color: red;}
 - s'applique à tous les paragraphes
- Définir une classe pour un sous-ensemble d'éléments
 - Une classe particulière pour n'importe quel élément
 - .vert** {color: green;}
 - s'applique à toute balise revendiquant le sélecteur "vert"
 - Une classe particulière pour un élément particulier (<h1>, <p>, <cite>, ...)
 - P.bleu** {color: blue;}
 - P.i** {font-style: italic;}
 - s'applique aux paragraphes qui se revendiquent du sélecteur "bleu" ou "i »
- Utiliser attribut **class** pour appliquer les styles d'une classe

Un premier exemple

```
<!-- 1.html -->
<html><head>
  <style type="text/css">
    .vert {color: green;}
    P {font-size: 16pt; color: red;}
    P.bleu {color: blue;}
    P.italic {font-style: italic;}
    H1.titre1 {border: solid; border-width: 1;
              text-align: center; color: cyan;}
  </style>
</head><body>
  <h1 class="vert">h1 standard classe vert</h1>
  <h1 class="titre1">h1 classe titre1</h1>
  <p>redéfinition du standard P</p>
  <p class="bleu">P classe bleu</p>
  <p class="italic">P classe italic</p>
</body></html>
```



Les sous-classes de style : l'attribut ID

- Permet de faire varier certains paramètres d'une classe
- Exemple : une classe définit un style général de paragraphe mais on souhaite pouvoir changer uniquement la définition de la couleur
 - première possibilité : faire une deuxième classe (la duplication de code est toujours une mauvaise chose)
 - mieux : définir un modificateur qui n'agit que sur les paramètres souhaités
- # est le caractère permettant la définition de sous-classes

Les sous-classes de style : l'attribut ID

- Exemple :

```
<style type="text/css">  
  .standard {color: green; font-size: 10pt;}  
  #cyan {color: cyan;}  
  P {font-size: 16pt; color: yellow;}  
  P.special {font-style: italic; color: red;}  
  P#bleu {color: blue;}  
</style>
```

- L'attribut ID

- permet de réaliser une exception dans une classe ou être utilisé seul (alors équivalent à CLASS)

Un deuxième exemple

```
<!-- 2.html -->
<html><head>
  <style type="text/css">
    .standard {color: green; font-size: 10pt;}
    #cyan {color: cyan;}
    P {font-size: 16pt; color: yellow;}
    P.special {font-style: italic; color: red;}
    P#bleu {color: blue;}
  </style>
</head><body>
  <p>redéfinition du standard P</p>
  <p class="standard">P classe standard</p>
  <p class="special">P classe special</p>
  <p class="standard" id="cyan">P classe standard mais cyan</p>
  <p class="standard" id="bleu">P classe standard mais bleu</p>
  <p class="special" id="bleu">P classe special mais bleu</p>
  <h1 class="standard" id="cyan">h1 classe standard mais cyan</h1>
  ID devient une classe...
  <h1 id="cyan">h1 mais cyan</h1>
  Impossible...
  <h1 class="standard" id="bleu">h1 classe standard mais bleu</h1>
</body></html>
```



Les feuilles de styles

- La définition des styles se fait dans un ou plusieurs fichiers séparés
- Permet d'appliquer les styles à un ensemble de documents
- Inclusion de commentaires comme en C `/* */` avec CSS
- Aucune règle concernant le nommage du fichier mais il est préférable de mettre `.css`
- Inclusion d'une feuille de styles dans un document avec la balise `<LINK>`

La balise <LINK>

- Permet de référencer des fichiers externes
- Entre <head>...</head>
- Trois attributs
 - **REL="stylesheet"** indique que le fichier inclus est une feuille de styles (d'autres valeurs dans le futur...)
 - **TYPE="text/css"** ou "text/javascript" - norme de styles utilisée dans le fichier
 - **HREF="mystyles.css"** permet d'indiquer l'URL de la feuille de styles
- Il est possible de lier plusieurs feuilles de styles tout en continuant d'utiliser des définitions internes
- Définition ponctuelle de style
`<P STYLE="font-size=35; color=blue;">P en bleu 35</P>`

Utilisation de feuilles de styles

```
/* styles CSS pour balise P */
```

```
/* pstyle.css */
```

```
P {font-size: 16pt; color: yellow;}
```

```
P.special {font-style: italic; font-size: 30pt; color: red;}
```

```
P#bleu {color: blue;}
```

```
/* styles CSS généraux */
```

```
/* allstyle.css */
```

```
.standard {color: green; font-size: 10pt;}
```

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

```
.noirsurblanc {background-color: white; color: black;}
```

```
.blancsurnoir {background-color: black; color: white;}
```

```
#rouge {color: red;}
```

Utilisation de feuilles de style

```
<!-- 4.html -->
<html><head>
    <link rel="stylesheet" type="text/css" href="allstyle.css">
    <link rel="stylesheet" type="text/css" href="pstyle.css">
</head><body>
    <span class="noirsurblanc">
        HTML<br />
        <span class="blancsurnoir" ID="rouge">
            et<br />
        </span>
        <span class="blancsurnoir">
            STYLES<br />
        </span>
        c'est facile !<br />
    </span>
    <p class="special" id="vert">P special vert (2 feuilles)</p>
    <p style="color: red; background-color=blue;">
        mais je prefere utiliser ce style local<br />
        en surcharge du style général de P
    </p>
</body></html>
```



Compléments sur les styles

- Groupement de balises

H1, H2, H3 {color: red;}

tous les titres H1, H2, H3 seront en rouge

- Sélection contextuelle d'application d'un style

H3 A {color: black;} : les liens définis dans `<H3>...</H3>` seront noirs

H3 H3 A {color: red;} : les liens dans

`<H3><H3><A>...</H3></H3>` seront rouges



- Héritage des styles

`<H1 class="titre1"><cite>HTML</cite></H1>`

des balises imbriquées dans des balises de niveau supérieur héritent du style défini dans ces balises

Compléments sur les styles

font-size: 14pt | +1pt Taille de la police

font-family: Arial | Times | ... Type de police

font-weight: normal | bold | 100...900 Poids de la police

font-style: normal | italic | oblique Style de la police

font-variant: normal | small-caps Petites majuscules

text-transform: none | uppercase | lowercase Maj., Min.

text-decoration: none | underline | overline | blink | line-through

text-align: left | center | right | justify Alignement du texte

text-indent: 1pt Retrait de la première ligne

vertical-align: top | middle | bottom | ... Alignement vertical

letter-spacing: 1pt Espacement des lettres

word-spacing: 3pt Espacement des mots

Compléments sur les styles

line-height: 1pt Hauteur de la ligne

color: rgb(0, 255, 0) Couleur de la police

background-image: url Image d'arrière-plan

background-color: transparent | rgb(0,255,0) Couleur d'arrière-plan

background-position: Position de l'arrière-plan

margin (ou -left, -right, -top, -bottom): 3px Marges

border: 1px Epaisseur de la bordure

border-style: solid | double | groove | ridge | dotted | dashed | ...

border-style-left (-top | -right | -left | bottom): solid | ...

border-color: blue Couleur de la bordure

padding (-top | -right | -left | bottom): 1px Remplissage interne

display: block | inline | none à la ligne suivante | à la suite | caché

Compléments sur les styles

■ Propriétés concernant les listes

list-style-type:

- pour les listes non numérotées : **square** | **circle** | **disc**
- pour les listes numérotées :

decimal 1, 2, 3, ...

decimal-leading-zero 01, 02, 03, ...

lower-roman i, ii, iii, ...

upper-roman I, II, III, ...

lower-alpha a, b, c, ...

upper-alpha A, B, C, ...

list-style-image: <images/puce.gif>;

- pour utiliser une image comme puce

list-style-position: outside | inside;

- pour positionner la puce dans le paragraphe ou non

Compléments sur les styles

- Appliquer des styles à certains états : les pseudo-classes et pseudo-éléments

`a:link {color: red;}`

`a:visited {color: blue;}`

`a:active {color: green;} // la souris clique sur le lien`

`a:hover {color: yellow;} // la souris passe sur le lien`

`p:first-line {color: red;}`

`p:first-letter {font-size: 200%;}`

`input:focus, textarea:focus {font-weight: bold;}`

- Autre méthode pour définir une couleur

`color: rgb(0, 255, 0);`

`color: rgb(25%, 50%, 25%);`

Les balises et <DIV>

- Idée : attribuer des définitions de styles à un ensemble d'objets (texte, image, formulaire...)
- La balise permet de définir un bloc "in-line" de texte (ou **conteneur**) auquel s'appliquera le style référencé par l'attribut CLASS et/ou ID
- La balise <DIV> permet aussi de définir un bloc ou conteneur mais
 - les blocs <DIV> se placent les uns en dessous des autres et ont une dimension
 - les blocs se placent les uns à côté des autres et ont la dimension qu'occupe leur contenu
- Attributs **class** pour CSS, **id** pour Javascript

Les balises et <DIV>

```
<!-- 3.html -->
<html><head>
  <style type="text/css">
    .noirsurblanc {background-color: white; color: black;}
    .blancsurnoir {background-color: black; color: white;}
    #rouge {color: red;}
  </style>
</head><body>
  <span class="noirsurblanc">
    HTML<br>
    <span class="blancsurnoir" ID="rouge">
      et<br>
    </span>
    <span class="blancsurnoir">
      STYLES<br>
    </span>
    c'est facile !<br>
  </span>
</body></html>
```



Positionner un bloc en CSS

- Un bloc se positionne par rapport à son conteneur (le conteneur initial est défini par <body>)

- en précisant des marges

```
<!-- 6.html -->          /* height: automatic; width: 80%; */
<html><head>
  <style type="text/css">
    div.conteneur {
      background-color: blue;
      height: 100px; width: 100px;}
    div.bloc1 {
      background-color: yellow;
      height: 50px; width: 50px;
      margin-left: 20px; margin-top: 40px;}
  </style>
</head><body>
  <div class="conteneur" id="conteneur">
    <div class="bloc1" id="bloc1">bloc1...</div>
  </div>
</body></html>
```



Positionner un bloc en CSS

- La propriété **float** permet de positionner un bloc à droite ou à gauche dans son conteneur, **le reste du conteneur occupera l'espace restant**

```
<!-- 7.html -->
<html><head>
  <style type="text/css">
    div.conteneur {
      background-color: yellow;
      height: 200px; width: 200px;
    }
    img.bloc1 {
      float: left;
    }
  </style>
</head><body>
  <div class="conteneur" id="conteneur">
    
    bla bla bla bla bla bla bla bla bla
    bla bla bla bla bla bla bla bla bla
    bla bla bla bla bla bla bla bla bla
    bla bla bla bla bla bla bla bla bla
  </div></body></html>
```




Positionner un bloc en CSS

- En utilisant des positions absolues/relatives
 - **position: relative;** --> positionnement par rapport à l'élément précédent
 - **position: absolute;** --> positionnement par rapport au coin supérieur gauche du conteneur
 - top: Npx;** --> distance depuis le bord supérieur
 - left: Npx;** --> distance depuis le bord gauche
 - **position: fixed;** --> positionnement pour fixer le bloc dans la fenêtre et le garder visible lors d'un défilement de la page
- Autres unités de mesures que `px`: `mm`, `cm`, `pt`, ...
- Autres propriétés possibles après `position`:
 - rendre le bloc visible ou invisible
 - visibility: visible | hidden;** (ne s'applique qu'aux blocs)
 - superposer des blocs
 - z-index: 1 | 2 | 3...;**

Positionner un bloc en CSS

```
<!-- 8.html -->
<html><head>
  <style type="text/css">
    div.conteneur {
      background-color: blue;
      height: 200px; width: 200px;
      position: absolute; top: 20px; left:40px;
    }
    div.bloc1 {
      background-color: yellow;
      height: 50px; width: 50px;
      position: absolute; top: 50px; left:100px;
    }
  </style>
</head><body>
  <div class="conteneur" id="conteneur">
    bla bla bla bla bla bla bla bla bla bla bla bla
    <div class="bloc1" id="bloc1">bloc 1</div>
  </div>
</body></html>
```



Positionner un bloc en CSS

```
<!-- 8bis.html -->
<html><head>
  <style type="text/css">
    div.conteneur {
      background-color: blue;
      height: 200px; width: 200px;
      position: absolute; top: 20px; left: 40px;
    }
    div.bloc1 {
      background-color: yellow;
      height: 50px; width: 50px;
      position: relative; display: inline;
    }
  </style>
</head><body>
  <div class="conteneur" id="conteneur">
    bla bla bla bla bla bla bla bla bla bla bla
    <div class="bloc1" id="bloc1">bloc 1</div>
  </div>
</body></html>
```



Positionner un bloc en CSS

```
<!-- 8ter.html -->
```

```
<html><head>
```

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

```
    div.bloc1 {
```

```
      background-color: blue;
```

```
      height: 200px; width: 200px;
```

```
      position: absolute; top: 20px; left: 40px; z-index: 2;}
```

```
    div.bloc2 {
```

```
      background-color: yellow;
```

```
      height: 50px; width: 50px;
```

```
      position: absolute; top: 20px; left: 40px; z-index: 1;}
```

```
  </style>
```

```
</head><body>
```

```
  <div class="bloc1" id="bloc1">bloc 1</div>
```

```
  <div class="bloc2" id="bloc2">bloc 2</div>
```

```
</body></html>
```

