

# XML et les feuilles de style

Courte présentation de CSS

# Introduction

- ▶ XML s'applique à la structure des données mais pas à leur présentation
- ▶ Comment faire pour présenter les données contenues dans un fichier XML ?

# Les styles en HTML

- ▶ Attribut style

```
<H1 style="font-weight: bold; text-align: center">
  Bienvenue
</H1>
```

- ▶ L'attribut style ne s'applique qu'aux éléments auxquels il a été ajouté

- ▶ Élément <style>

```
<style>
  H1 {font-weight: bold; text-align: center}
</style>
<H1>Bienvenue</H1>
```

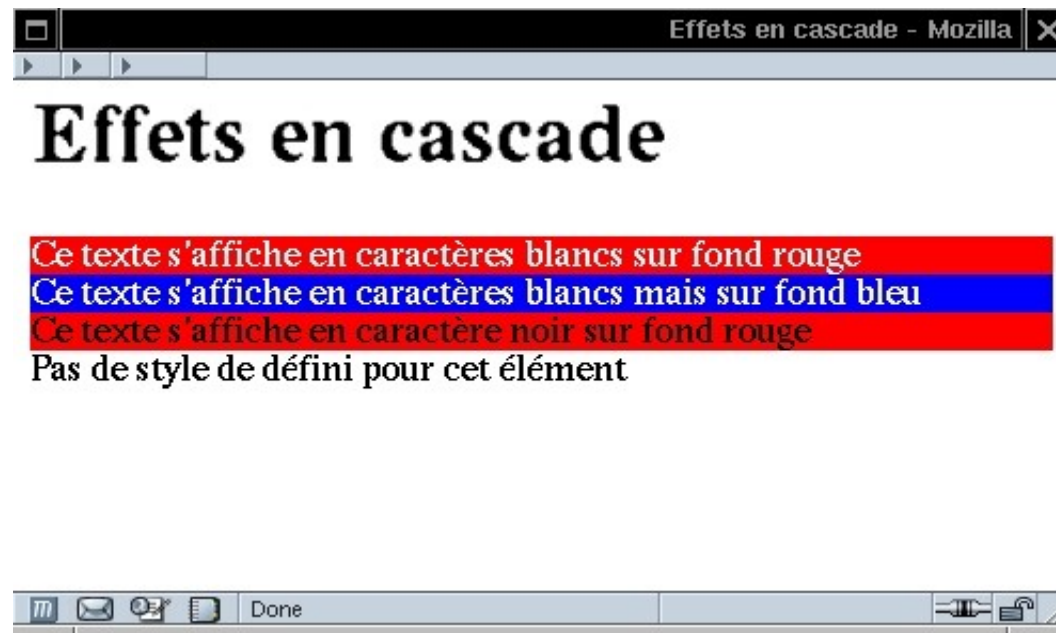
- ▶ La déclaration de style CSS pour chaque type de balise s'appelle **règle**
- ▶ La collection de toutes les règles s'appelle **feuille de style**

# Style en cascade

- ▶ Quand un style est appliqué à un élément, il se répercute en cascade sur tous les éléments enfants

```
<html> <head> <title>Effets en cascade</title> </head>
<body>
  <h1>Effets en cascade</h1>
  <div style="background-color:red;color:white">
    Ce texte s'affiche en caractères blancs sur fond rouge
  <div style="background-color:blue">
    Ce texte s'affiche en caractères blancs mais sur fond bleu
  </div>
  <div style="color:black">
    Ce texte s'affiche en caractères noirs sur fond rouge
  </div>
</div>
<div>
  Pas de style de définie pour cet élément
</div>
</body>
</html>
```

# Style en cascade



# Les classes en HTML

- ▶ Supposons que nous avons à rédiger du texte sur des fonds de couleurs différentes en fonction de l'importance de l'information
  - ▶ Une sur fond rouge
  - ▶ Une sur fond jaune
  - ▶ Une sur fond vert
- ▶ Classiquement, nous allons faire :

```
<div style="background-color:red" >....</div>  
<div style="background-color:yellow" >....</div>  
<div style="background-color:green" >....</div>
```
- ▶ Cette écriture est lourde.
- ▶ Dans les feuilles de style CSS, il existe une solution pour simplifier cela : les classes

# Les classes en HTML (suite)

- ▶ La solution :

```
<style>
    .urgent {background-color : red; }
    .important {background-color : yellow; }
    .normal {background-color : green; }
</style>
```

- ▶ Utilisation des classes

```
<div class="urgent"> Demain, changement de salle</div>
<div class="important"> La semaine prochaine, programmer
    un cours</div>
```

# Exemple plus complexe

```
<body>
  <style>
    .note {border:solid 3px black; background-color:lightBlue;}
    .astuce { background-color:yellow;}
    .avertissement { background-color:red;}
    P.note {border:dashed 3px black; background-color:lightBlue;}
  </style>
  <div class="astuce">Sur fond jaune</div>
  <div class="note astuce">Avec bord et fond rouge</div>
  <p class="note">Avec bord en pointillé et fond bleu
clair</p>
</body>
```

Sur fond jaune

Avec bord et fond rouge

Avec bord en pointillé et fond bleu clair



# Les pseudo-classes

- ▶ Elles permettent de définir des styles d'affichage de certains états d'un marqueur
- ▶ Attaché au nom de l'élément par : au lieu du .
- ▶ Elles possèdent des noms prédéfinis. Il en existe 7 avec CSS2
  - ▶ 3 associées à la balise <a>
    - ▶ a:link lien non visitée
    - ▶ a:active lien activé
    - ▶ a:visited lien visité

# Les pseudo-classes (suite)

## ▶ Depuis CSS2

- ▶ 2 pseudo-classes dynamiques associées à la balise <a>
  - ▶ a:hover lorsque que le pointeur de la souris est au-dessus du lien
  - ▶ a:focus lorsque que l'élément devient l'objet d'attention (par la touche TAB ou bien par clique avec la souris)
- ▶ :lang(..) pour définir un style en fonction de la langue définit en attribut dans une balise
- ▶ :first-child Permet de spécifier la façon dont l'élément doit être affiché lorsqu'il est l'enfant de l'élément conteneur

# Exemple d'utilisation de :first-child

```

<style>
h1, body {font-size:11pt}
h1 em, p strong { color: red}
p em:first-child {color:blue}
p>em:first-child {color:green}
</style>
<h1>Sans balise em dans le h1 </h1>
<h1><em>Avec em dans le h1</em></h1>
<h1><p>Test dans un paragraphe <it><em>Ici commence la balise
em</em></it> et se <em>termine</em> maintenant</p></h1>
<p>Ici <em>premier enfant et directement</em> en-dessous de la
balise <em> mais pas cela</em></p>

```

Sans balise em dans le h1

*Avec em dans le h1*

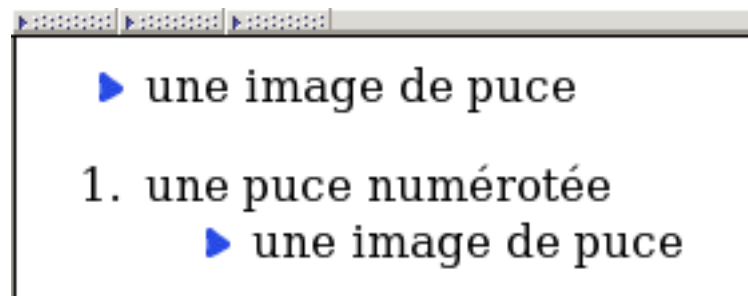
Test dans un paragraphe *Ici commence la balise em* et se *termine* maintenant

Ici *premier enfant et directement* en-dessous de la balise *mais pas cela*

# Héritage de sélecteurs

- ▶ Exemple : changeons les puces des listes non ordonnées
  - ▶ Balise <LI> mais elle est aussi utilisée dans les listes ordonnées
  - ▶ Héritage de sélecteurs (*fichier csscascade.html*)

```
<STYLE>
  UL LI {list-style-image:url(puce.png);}
</STYLE>
<UL>
  <LI>une image de puce
</UL>
<OL>
  <LI>une puce numérotée
  <UL>
    <LI>une image de puce
  </UL>
</OL>
```



# Sélecteurs contextuels

▶ Exemple :

```
h1 em, p strong {color: red }
```

- ▶ Signifie que si une balise `<em>` se trouve dans un élément `<h1></h1>` ou bien qu'une balise `<strong>` se trouve dans un élément `<P></P>`, alors le contenu sera affiché en rouge
- ▶ De même si `<em>` est dans un élément `<ul>`, lui même dans un élément `<h1>`

# Propriétés des polices

## ▶ font-family

- ▶ Se rapporte au style de la font
- ▶ Polices génériques définies dans les spécifications CSS2 : serif, sans-serif, cursive, fantasy et monospace
- ▶ Possibilité de mettre une liste de police

```
h1 {font-family: Helvetica, sans-serif }
```

## ▶ font-size

- ▶ Possibilité d'utiliser des valeurs de taille en valeur absolue, relative, pourcentage ou par un mot clé
  - ▶ Les mots clés : xx-small, x-small, small, medium, large, x-large, xx-large
  - ▶ Dans ce cas, utilise les tailles prédéfinies par le navigateur

## Propriétés des polices (2)

- ▶ font-style
  - ▶ normal, italic ou oblique
- ▶ font-weight
  - ▶ normal, bold pour les valeurs absolues
  - ▶ bolder ou lighter pour les valeurs relatives
  - ▶ Ou bien des valeurs, multiples de 100
    - ▶ 100 pour le plus maigre
    - ▶ 900 pour le plus gras
    - ▶ 400 correspond à normal et 700 à bold

# Styles externes

- ▶ En HTML

```
<LINK type="text/css" rel="stylesheet" href="fichierstyle.css">
```

- ▶ En XML

```
<?xml-stylesheet type="text/css" href="fichierstyle.css">
```



# Exemple CSS et XML

```
<?xml-stylesheet type="text/css" href="cssxml.css" ?>
<document>
<titre>Bienvenue</titre>
<important>test</important>
<note>Ceci est une note</note>
</document>
```

```
important {background-color: red;}
titre {
  font-family: serif;
  font-size: 20pt;
  font-weight: bold;
}
note {
  font-style: italic;
}
```



**Bienvenue** test *Ceci est une note*

# Exemple CSS et XML : les attributs en CSS 2

```
<?xml-stylesheet type="text/css" href="cssxml2.css" ?>
<document>
<titre>Bienvenue</titre>
<note type="important">test</note>
<note>Ceci est une note</note>
</document>
```

```
note[type="important"] {background-color: red;}
titre {
  font-family: serif;
  font-size: 20pt;
  font-weight: bold;
}
note {
  font-style: italic;
}
```



**Bienvenue** test Ceci est une note