

# *Les standards du Web*



*Partie 2*

Patrick VINCENT  
pvincent@erasme.org

# *Partie 1*

## *Les pages HTML* *-bases théoriques et pratiques-*



# *Principes du langage HTML*



# Définition

- HTML : **H**yper**T**ext **M**arkup **L**anguage
- Langage universel de conception de pages Web créé en 1998 par le W3C (World Wide Web Consortium)
- **Hypertexte** : permet de créer des liens hypertextes vers des ressources locales ou distantes.
- C'est un **langage de marquage** de document, il permet de décrire la structure d'un document électronique sous forme lisible par un être humain et interprétable par une machine.
- XHTML = réécriture de l'HTML visant à assurer une transition vers l'XML

# *Pas de mise en forme définitive*

- HTML est un langage qui décrit le **contenu**, la **structure logique** et la **présentation globale** d'un document mais **pas son aspect final à l'écran**, qui dépend de l'outil utilisé (écran, synthétiseur vocal, imprimante).
- Différent d'un document PDF par exemple.
- Ex : l'utilisateur peut modifier au moment de l'affichage plusieurs paramètres de présentation.
  - Dans Firefox :
    - Affichage > Taille du texte
    - Affichage > Style de la page

# Hello World

- Ouvrir un bloc-note, taper et enregistrer le fichier *page1.html* suivant :

```
<html>
  <head>
</head>
  <body>
    Hello World !
  </body>
</html>
```

- Ouvrir ce fichier avec votre navigateur Firefox
- Vérifier le code source de ce fichier

# Structuration avec des balises

- Les fichiers HTML sont constitués de texte.
- Pour définir la structure du document, on utilise des **balises** délimitées par les caractères < et >
- Balise ouvrante : **<balise>**  
Balise fermante : **</balise>**
- Les balises définissent une propriété qu'elles appliquent au contenu qu'elles encadrent.
- **Une balise ouverte doit toujours être fermée.**

```
<balise1>contenu</balise1>
```

# Exemples

- Titre de niveau 1

```
<h1>titre</h1>
```

- Mise en intonation

```
<strong>important</strong>
```

- Paragraphe

```
<p>  
texte texte texte  
texte texte texte  
</p>
```



# Les balises auto-fermantes

- Certains éléments appelés éléments vides ne marquent pas du texte.
- Il n'ont pas de contenu, les balises de début et de fin peuvent être combinées :

```
<br></br>
```

=

```
<br />
```

- Il s'agit la plupart du temps de balises nécessitant des attributs :

```

```

# Ajout d'attributs

- Des attributs peuvent être ajoutés à un élément
- Ils précisent celui-ci (apparence, comportement...) sans faire partie de son contenu
- Le nom de l'attribut apparaît après le nom d'élément dans la balise ouvrante et est suivi du signe = puis de la valeur de l'attribut entre guillemets.
- Les attributs d'un élément forment un ensemble (nom unique, pas d'ordre).

```
<a href="http://www.spip.net" title="site SPIP">Cliquer ici</a>
```

# Imbrication

- Les éléments peuvent être imbriqués les uns dans les autres :

```
<p>  
  <strong>  
    texte texte texte  
  </strong>  
</p>
```

- Il ne peut pas y avoir de chevauchement. Tout élément doit être fermé avant ceux qui le contiennent.
- Un document ainsi structuré forme donc un arbre dont les nœuds sont les éléments et les feuilles des fragments du texte.

- Compléter votre page avec :
  - une balise de titre 1 `<h1>` autour de Hello World
  - deux paragraphes de votre choix `<p>`
  - la mise en intonation `<strong>` d'une partie du texte à l'intérieur d'un paragraphe
  - l'image de l'écureuil SPIP à l'intérieur d'un paragraphe `<img />`
- Vérifier le résultat localement dans le navigateur
- Envoyer la page en ligne à la racine de votre site en FTP
  - La consulter en ligne

# Structure d'un document

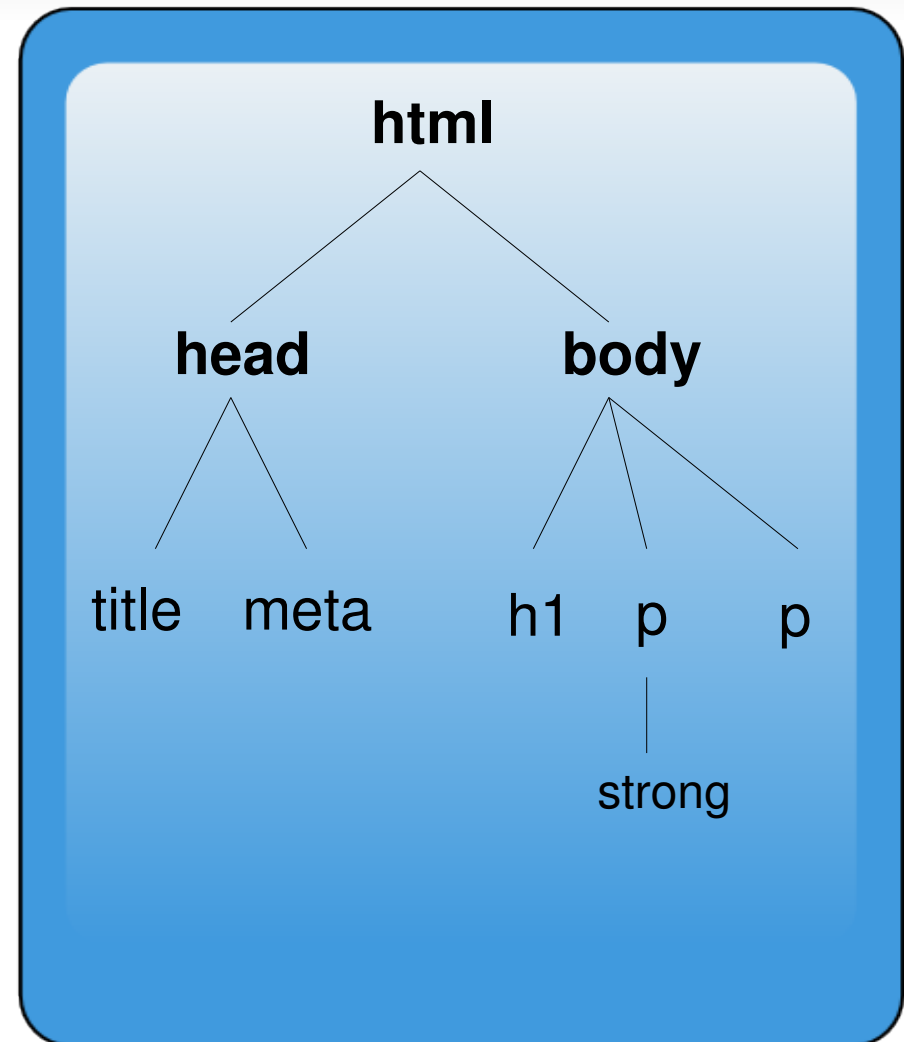
```
<html>
  <head>
    <title>Ma page</title>
    <meta name="author" content="Patrick VINCENT" />
  </head>
  <body>
    <h1>Hello World !</h1>
    <p>
      Nous avons trouvé la photo d'un animal.
      Il s'agit d'un<strong>ecureuil</strong>
      
    </p>
    <p>Pour en savoir plus cliquer
      <a href="http://www.spip.net">ici<a>
    </p>
  </body>
</html>
```

# Trois éléments de base

- **<html>...</html>** : le document complet
- **<head>...</head>** : l'en-tête
  - type de document
  - type de caractères
  - mots clés
  - titre de la fenêtre
- **<body>...</body>** : le corps du document
  - blocs
  - contenu : texte, tableaux, images, sons...
  - inclusion d'applications : flash, javascript, java

# Structure arborescente

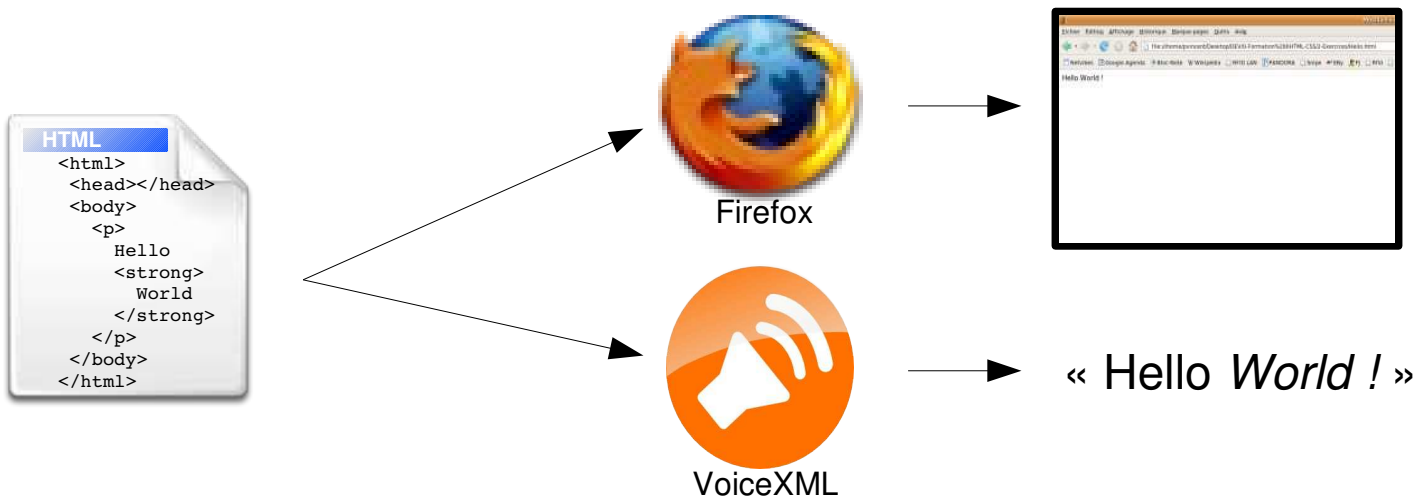
- Un document HTML peut être lu comme un arbre
- L'ordre des nœuds et leur imbrication précisent la structure du document
- Cette structure est exploitable :
  - Quels sont les titre et auteur du document ?
  - Combien y-a-t-il de paragraphes ?
  - Quel est le titre de niveau 1 qui précède le 1er paragraphe ?



Utilisé pour le  
**DOM**  
(Document Objet Model)

# Interprétation

- Le document HTML peut être visualisé dans un navigateur, la structure logique est interprétée pour fournir une présentation
- Un document peut donc être considéré à trois niveaux :
  - physique (syntaxe)
  - logique (structure d'arbre)
  - présentation : navigateur, téléphone, pda, lecteur audio...





# Règles d'édition

- Dans un nouveau fichier, commencer par copier l'ossature du fichier HTML par défaut
- Les espaces, les tabulations et les retours à la ligne ne sont pas interprétés par le navigateur
  - Pour forcer les retours à la ligne : utiliser des paragraphes `<p>` ou des césures `<br />`
  - Pour forcer l'espacement : utiliser des espaces insécables `&nbsp;`;
- Utiliser les caractères spéciaux
  - é = `&eacute;`;
  - è = `&egrave;`;
  - à = `&agrave;`;

# Pour une bonne lisibilité de la source

- Ajouter des commentaires :

<!-- -->

```
<!-- commentaire -->
```

- Aérer le document : retour à la ligne + lignes vides

```
<body>
<p>
<h1>texte</h1>
</p>
</body>
```

- Indenter les éléments

```
<body>
  <p>
    <h1>texte</h1>
  </p>
</body>
```

# Pour une bonne lisibilité de la source

**NON**

```
<html><head><title>Ma page</title></head><body><h1>Hello World !
</h1><p>Nous avons trouvé la photo d'un animal.
</p></body></html>
```

**OUI**

```
<html>

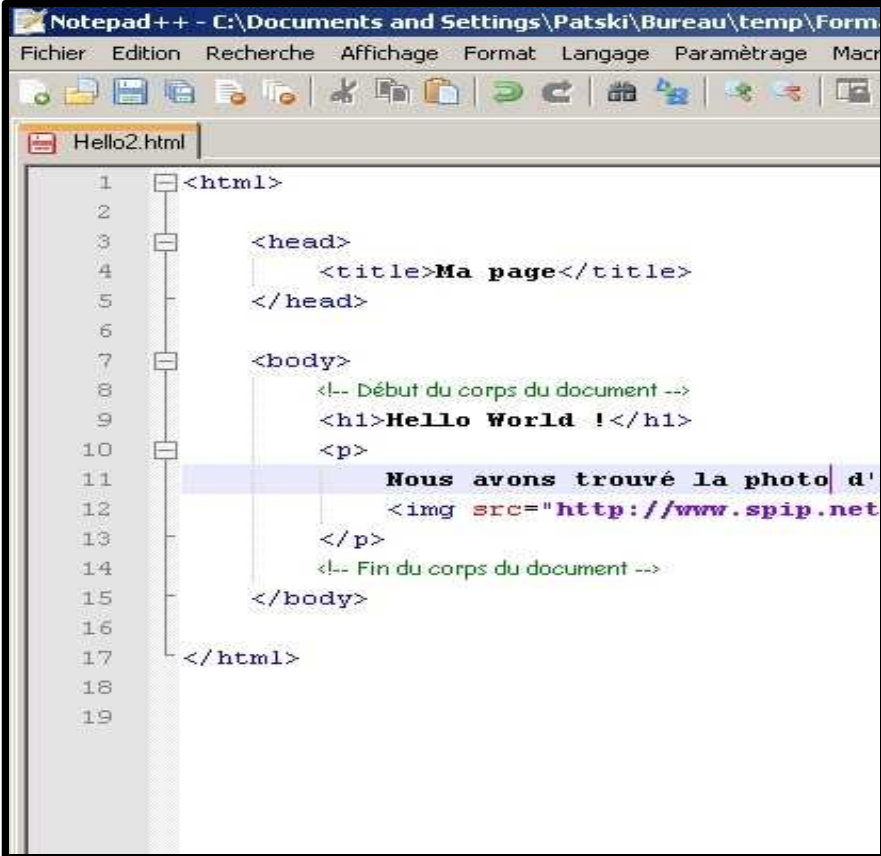
  <head>
    <title>Ma page</title>
  </head>

  <body>
    <!-- Début du corps du document -->
    <h1>Hello World !</h1>
    <p>
      Nous avons trouvé la photo d'un animal.
      
    </p>
    <!-- Fin du corps du document -->
  </body>

</html>
```

# Editeurs de texte GPL et gratuits

- notepad++ (Windows)



```
1 <html>
2
3   <head>
4     <title>Ma page</title>
5   </head>
6
7   <body>
8     <!-- Début du corps du document -->
9     <h1>Hello World !</h1>
10    <p>
11      Nous avons trouvé la photo d'
12      
11    </p>
12    <p>
13      Pour en savoir plus cliquer
14      <a href="http://www.spip.net" title="site SPIP">ici<a>
15    </p>
16  </body>
17 </html>
```

- activer le greffon *balises*
- voir aussi : Anjuta, Screem

- Si Windows installer notepad++
  - <http://notepad-plus.sourceforge.net/fr/site.htm> > Télécharger > Télécharger les binaires de Notepad++ > npp.4.0.2.Installer.exe
- Ouvrir votre fichier précédent
- Rendre la source lisible
  - ajouter des commentaires
  - l'indenter et ajouter des espaces
  - vérifier que cela n'a pas de conséquence sur l'affichage
- Ajouter
  - dans le paragraphe de votre choix, un retour à la ligne
  - puis la phrase «cet été je n'irai pas à la plage»
  - l'afficher dans Firefox en caractères cyriliques : Affichage > Encodage des caractères >UTF-8
- Mettre la page en ligne à jour et vérifier l'affichage

# *Le langage XHTML*



# Définition

- XHTML : **E**xtensible **H**yper**T**ext **M**arkup **L**anguage
- Evolution actuelle de l'HTML visant à
  - standardiser et homogénéiser le langage
  - éviter que les navigateurs utilisent leurs propres balises ou interprètent librement les marqueurs
  - assurer une transition vers un langage en arbre (XML)
- Depuis le XHTML 1.0 une page pourra évaluer son niveau de conformité aux standards du web (et indirectement aux services d'accessibilité) :
  - normes du W3C (World Wide Web Consortium) :  
<http://www.w3.org/>
  - <http://validator.w3.org/>

# Plusieurs principes de base du XHTML

- Toute balise ouvrante doit être fermée
- Les éléments doivent être correctement imbriqués
- Les balises et les attributs doivent être écrits en minuscules
- Les valeurs des attributs doivent apparaître entre apostrophes ou guillemets
- Chaque attribut doit avoir une valeur
- L'attribut **name** ne doit plus être utilisé pour l'identification des balises. A la place on utilisera l'attribut **id** (qui servait déjà pour les CSS).

```
<p>texte</p>
```

```
<p>texte <strong>important</strong></p>
```

```
<p id="par1">texte</p>
```

```

```



# Buts et pratiques

- Le XHTML vise à faire disparaître toute notion de mise en forme de la page HTML et d'en laisser la gestion aux feuilles de style.
- Les balises doivent orienter le sens sémantique donné au texte, mais pas sa mise en forme :
  - **<emphatic>** plutôt que **<italic>**
- Certaines balises et attributs sont déconseillés :
  - police : `<font face="arial"></font>`
  - gras et italique : `<b></b>` et `<i></i>`
  - centrer : `<center></center>`
  - `<body background="white" bgcolor="black"></body>`

->Il est plus judicieux d'initier un projet de design Web aujourd'hui qu'il y a 3 ans!

# Structure de document XHTML 1.0

- Un document XHTML 1.0 est un document HTML 4.01 précédé d'un élément : le **doctype** (type de document)
- **Doctype XHTML 1.0 Transitional**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

- respecte les cinq principes du XHTML
- autorise la présence des balises *<font>*, *<center>* et des attributs *border*, *align*, *bgcolor*, *target*
- autorise l'emploi des *<iframe>*
- -> **permet un passage progressif des documents (et des compétences) vers l'XHTML**

# Structure de document XHTML 1.0

## ■ Doctype XHTML 1.0 Frameset

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

- identique au doctype transitional
- remplace <body> par <frameset> + <frame>
- -> permet l'utilisation de sous-sites dans un site

## ■ Doctype XHTML 1.0 Strict

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- toutes les balises et les attributs déconseillées sont exclues
- -> pour les sites validés W3C (rare!)

# Structure de la page XHTML 1.0 type

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">

<head>
  <title>Ma page</title>

  <!-- type de document et de caractères -->
  <meta http-equiv="content-type" content="text/html; charset=UTF-8" />

  <!-- meta donnees pour les moteurs de recherche -->
  <meta name="description" content="Titre" />
  <meta name="author" content="P. VINCENT" />
  <meta name="keywords" lang="fr" content="mot1, mot2" />
  <meta name="robots" content="all" />

  <!-- feuilles de style -->
  <link rel="stylesheet" type="text/css" media="screen" href="feuille.css" />
</head>

<body>
  ...
</body>

</html>
```

# Structure de document XHTML 1.1

- Un document XHTML 1.1 intègre les éléments suivants:
  - **La version de XML et le codage des caractères**

```
<?xml version="1.0" encoding="UTF-8"?>
```

- **Le Doctype XHTML 1.1 (strict strict strict)**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"  
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

- **La balise meta XHTML**

```
<meta http-equiv="content-type" content="xhtml+xml; charset=UTF-8" />
```

- A n'utiliser que dans quelques années...

# Structure de la future page XHTML 1.1

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">

<head>
  <title>Ma page</title>

  <!-- type de document et de caractères -->
  <meta http-equiv="content-type" content="xhtml+xml; charset=UTF-8" />

  <!-- meta donnees pour les moteurs de recherche -->
  <meta name="description" content="Titre" />
  <meta name="author" content="P. VINCENT" />
  <meta name="keywords" lang="fr" content="mot1, mot2" />
  <meta name="robots" content="all" />

  <!-- feuilles de style -->
  <link rel="stylesheet" type="text/css" media="screen" href="feuille.css" />
</head>

<body>
  ...
</body>

</html>
```

# *L'en-tête*

*<head></head>*



# Mentions obligatoires

- Le titre **<title></title>**
  - Il apparaît dans le haut de la fenêtre du navigateur.
  - Il est utilisé pour l'indexation par les moteurs de recherche et le stockage d'informations locales par le navigateur

```
<title>Ma page</title>
```

- Les marqueurs de meta-données **<meta />**
  - Mentions non affichées mais utiles (avec des réserves) pour les moteurs de recherche et d'indexation
    - auteur, description, tags, date

```
<meta name="description" content="pour les moteurs de recherche." />  
<meta name="author" content="P. VINCENT" />  
<meta name="keywords" lang="fr" content="mairie, informatique" />  
<meta name="date" content="2007-03-20">  
<meta name="robots" content="all" />
```



# Meta données

- Type de document et jeu de caractère **<meta />**
  - Indique au navigateur la version HTML ou XHTML
  - Donne le jeu de caractères d'encodage (défaut UTF-8)

```
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />
```

- Rechargement ou redirection de page **<meta />**
  - Permet à la page d'être redirigée au bout d'un certain temps

```
<meta http-equiv="refresh" content="5; URL=http://www.spip.net/">
```

# Inclusion des styles CSS

- Feuilles de style externes **<link />**
  - Indique l'adresse d'inclusion des feuilles de style **.css** liées à la page

```
<link rel="stylesheet" type="text/css" media="screen" href="feuille.css" />
```

- Feuilles de style locales **<style>**
  - Permet de déclarer les styles dans la page HTML elle-même

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

# Inclusion de scripts JavaScript

- Script Javascript externe **<script />**
  - Indique l'adresse d'inclusion du script **.js** lié à la page

```
<script type="text/javascript" src="menu.js" />
```

- Script Javascript local **<script>**
  - Permet de déclarer les styles dans la page HTML elle-même

```
<script type="text/javascript">  
    /* Mon script */  
</script>
```

# Divers

- Information de flux RSS **<link />**
  - Indique la présence et l'adresse du lien RSS de la page via le navigateur de l'internaute

```
<link rel="alternate" type="application/rss+xml" href="spip.php?page=backend" title="mon site" />
```

- Icône du site (favicon) **<link />**
  - Modifie l'icône affiché dans la barre d'adresse

```
<link rel="shortcut icon" type="image/png" href="/dist/favicon.png" />
```

- Ouvrir le fichier *page\_standard.html*
- Modifier :
  - le titre
  - les meta-données : auteur, mots-clefs, descriptif
- Enregistrer la page sous : ***ma\_page\_standard.html***
  - La passer en lecture seule.
  - Cette page est votre page HTML vierge par défaut pour toutes vos créations à venir.
  - Modifier le favicon
  - Ajouter l'information du flux RSS de votre site
- Pour aller plus loin, en faire une copie et :
  - Faire une redirection au bout de 5 secondes vers votre site

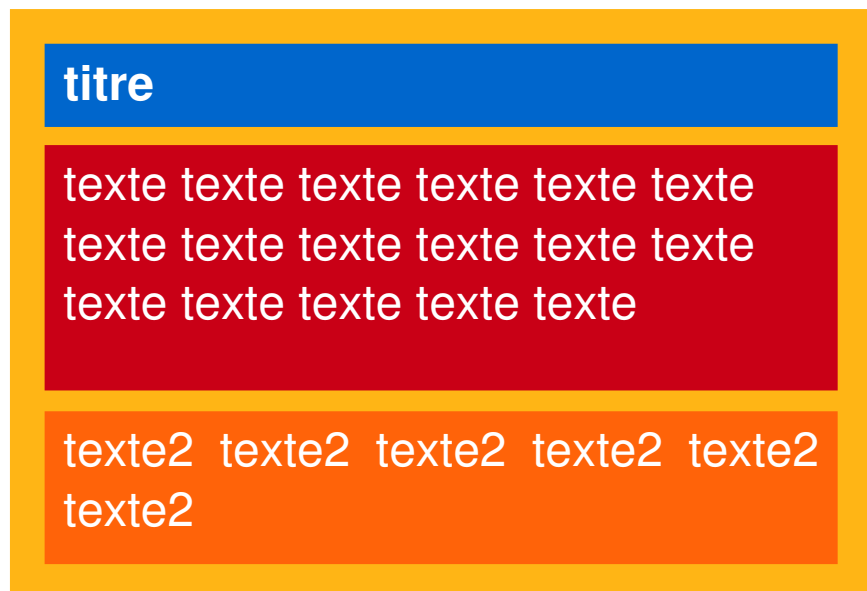
# *Structure du corps*

*<body></body>*



# Balises de type bloc

- Ce sont des «boîtes» qui peut contenir
  - d'autres éléments de type bloc
  - des éléments de type en ligne
  - des balises auto-fermantes
  - du texte
- les blocs se positionnent les uns en dessous des autres et occupent toute la largeur du parent



```
<div>  
  <h1>titre</h1>  
  <p>texte texte texte</p>  
  <p>texte2 texte2 texte2</p>  
</div>
```

# Balises de type en-ligne

- Ce sont des «boîtes» qui peut contenir
  - des éléments de type en-ligne
  - du texte
  - mais **pas de balises de type bloc**
- un élément en ligne se trouve donc aux derniers maillons de l'arbre html
- les éléments en ligne se placent à côté les uns des autres et habillent l'élément parent

texte texte texte    texte2    *texte2*

texte2 texte2 texte2 texte2 texte2

texte2    texte2



```
<div>
  <span>texte texte texte</span>
  <span>texte2 <em>texte2</em> texte2
  texte2 texte2 texte2</span>
  
</div>
```



# Conteneurs génériques

## ■ <div>

### BLOC

- élément neutre permettant de regrouper des éléments de tout type (bloc ou en-ligne) : <div>,<span>,<h1>,<img />...
- ne pas abuser des <div> et préférer des éléments avec du «sens» sémantique

```
<div>  
  <h1>Question existentielle :</h1>  
  <blockquote>To be SPIP or not to be ?</blockquote>  
</div>
```

## ■ <span>

### EN-LIGNE

- élément neutre pour regrouper des éléments en-ligne : <img />,<strong>,<em>...

```
<span><abbr>SPIP</abbr> est un outil de publication  
<strong>partagé</strong></span>
```

# Attributs de nommage

- **id** et **class** sont utilisés pour nommer les éléments HTML afin de leur appliquer des styles CSS
  - **id** désigne un élément unique dans la page
  - **class** désigne un élément qui peut apparaître plusieurs fois

```
<div id="bloc1">  
  
  <span class="texte">texte2 texte2 texte2  
  texte2 texte2 texte2</span>  
  
  <a href="page.html" class="lien">cliquer ici</a>  
  
</div>
```



# *La liste des conteneurs principaux*

page

entete

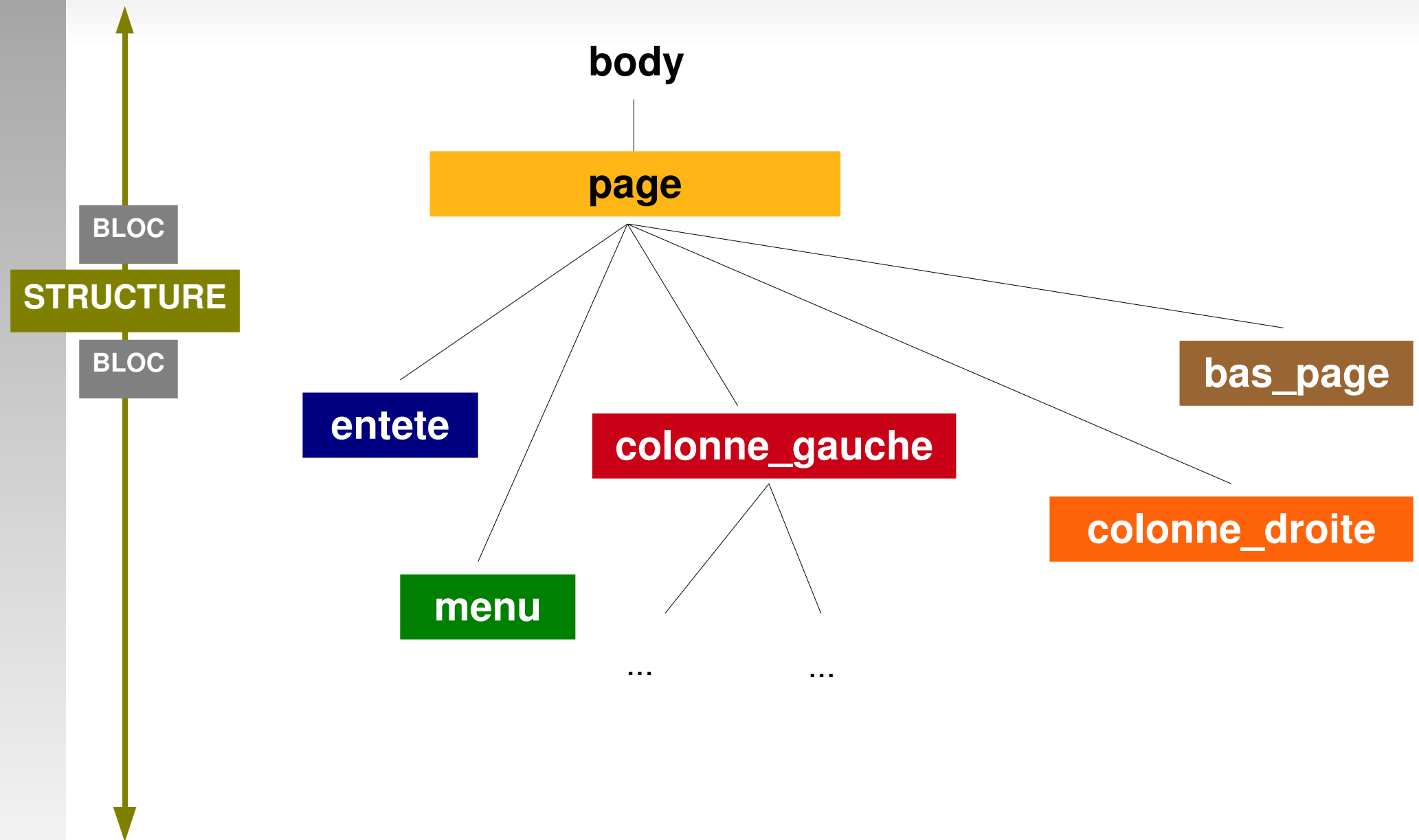
menu

colonne\_gauche

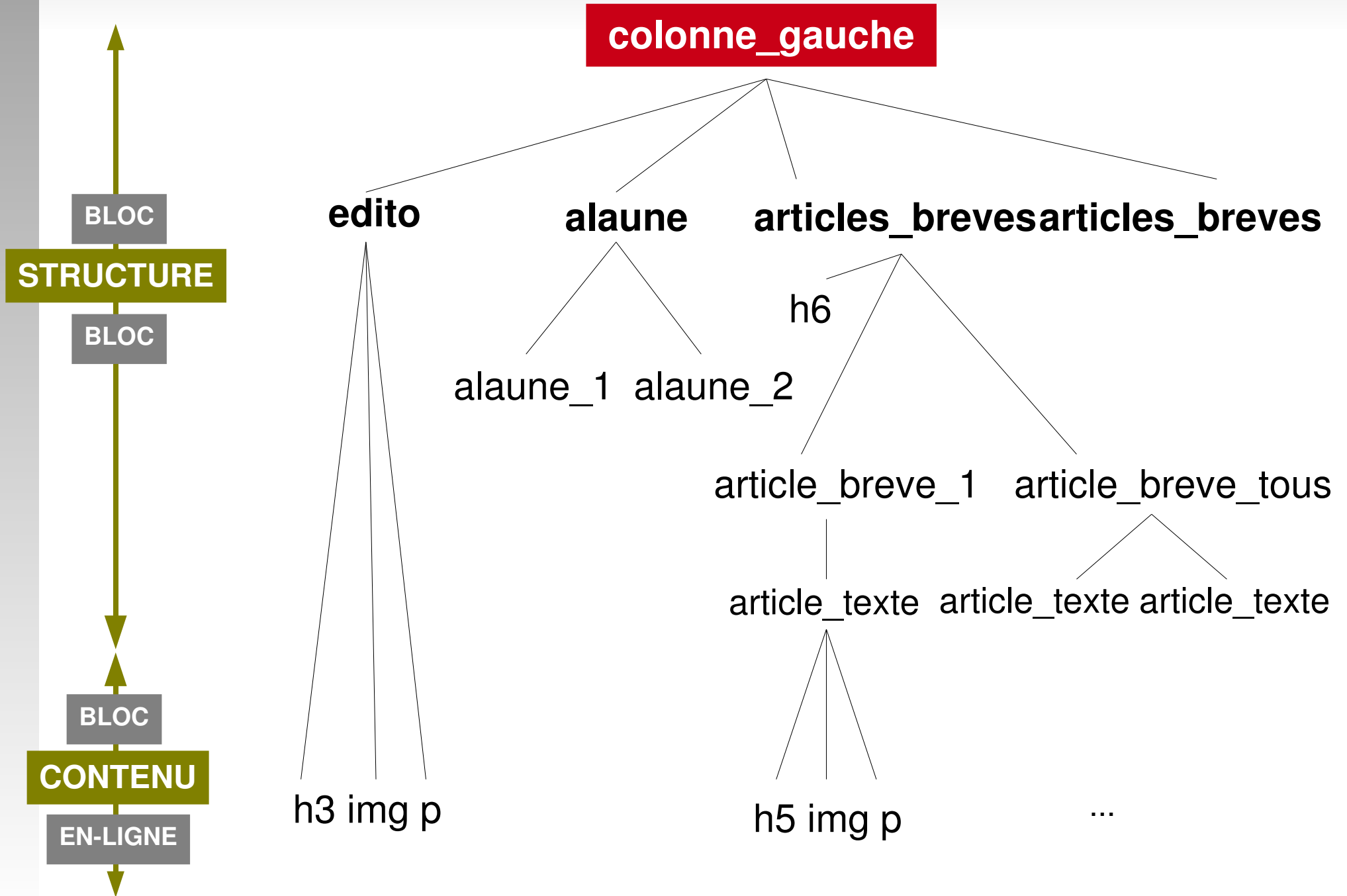
colonne\_droite

footer

# L'arbre de structure principale



# L'arbre des éléments



- Faire une copie de votre page standard : *projet1.html*
- Créer à l'intérieur du corps de la page les six conteneurs principaux à l'aide de balises **<div>**.
  - les nommer avec l'attribut **id**
  - rajouter une ligne de texte au début de chaque bloc
- A l'aide de l'arbre des éléments de la colonne gauche, créer la structure de la page

- Afin de visualiser les blocs ajouter à l'en-tête de votre document :

```
<style type="text/css">
  div
  {
    padding: 10px;
    margin: 10px;
    border: black solid 1px;
  }
</style>
```

- Depuis le fichier *projet1-complet.html* ajouter les éléments de contenu suivants :
  - Bloc Edito
    - L'image *images/arton1.gif*
    - Un titre <h3> : *Carte d'identité de la commune*
    - Un paragraphe <p> avec du contenu mise en forme : *Description de la commune...*
  - Bloc `article_breve_1`
    - L'image *images/rubon3-15c52.gif*
    - Un titre <h5> *Titre* avec un lien <a>
    - La date en mise en forme emphatique <em>
    - Un paragraphe <p> avec du contenu mise en forme : *Texte...*
    - Un lien : *lire la suite*



# *Détail des balises du corps*

*<body></body>*



# Les balises de type bloc

Balise	Signification
<code>&lt;body&gt;</code>	Balise bloc mère du corps de la page, c'est le conteneur principal
<code>&lt;div&gt;</code>	Conteneur de blocs
<code>&lt;p&gt;</code>	Paragraphe
<code>&lt;h1&gt; ... &lt;h6&gt;</code>	Titre de niveau 1 à 6
<code>&lt;blockquote&gt;</code>	Marque une citation
<code>&lt;pre&gt;</code>	<u>Affiche exactement ce qu'on y mets</u> (tabulations,sauts)
<code>&lt;ul&gt;,&lt;ol&gt;,&lt;li&gt;</code>	Liste énumérative et liste numérotée
<code>&lt;dl&gt;,&lt;dd&gt;,&lt;dt&gt;</code>	Liste de définition
<code>&lt;table&gt;,&lt;th&gt;,&lt;tr&gt;,&lt;td&gt;</code>	Tableaux
<code>&lt;tbody&gt;,&lt;thead&gt;,&lt;tfoot&gt;</code>	Sous partie d'un tableau
<code>&lt;fieldset&gt;,&lt;form&gt;</code>	Formulaire interactif
<code>&lt;noscript&gt;</code>	Texte alternatif aux scripts
<code>&lt;map&gt;</code>	image réactive (nécessite <code>&lt;area&gt;</code> )
<code>&lt;hr /&gt;</code>	Séparation horizontale

- Note : Il s'agit essentiellement d'éléments de structure

# Liste des balises de type en-ligne

Balise	Signification
<code>&lt;span&gt;</code>	Conteneur en-ligne
<code>&lt;a&gt;</code>	Liens
<code>&lt;img /&gt;</code>	Images
<code>&lt;br /&gt;</code>	Saut de ligne
<code>&lt;em&gt;</code> , <code>&lt;strong&gt;</code>	Mise en intonation du texte
<code>&lt;abbr&gt;</code> , <code>&lt;acronym&gt;</code> , <code>&lt;cite&gt;</code> , <code>&lt;q&gt;</code> , <code>&lt;dfn&gt;</code>	Sigle, citation et définition
<code>&lt;del&gt;</code> , <code>&lt;ins&gt;</code>	Suppression / ajout de texte
<code>&lt;kbd&gt;</code> , <code>&lt;code&gt;</code> , <code>&lt;samp&gt;</code> , <code>&lt;var&gt;</code>	Instruction et code
<code>&lt;sub&gt;</code> , <code>&lt;sup&gt;</code>	Indices et exposants
<code>&lt;object&gt;</code>	Animation multimédia embarquée

- Note : Il s'agit essentiellement d'éléments de contenu / mise en forme

- `<h1>...<h6>` **BLOC**
  - permet de hiérarchiser les titres en 6 niveaux
  - personnalisables via les CSS
  - ne peuvent pas inclure de balises de type BLOC

```
<div>
  <h1>Liste des Outils Internet</h1>

  <h2>Navigateurs</h2>
    <h3>IE</h3>
    <h3>Firefox</h3>

  <h2>CMS</h2>
    <h3>SPIP</h3>
    <h3>Joomla</h3>
</div>
```

# Les balises de paragraphe

- Ne peuvent pas contenir d'autres balises de type Bloc
- `<p>` , `<hr />` , `<pre>` , `<blockquote>` **BLOC**

```
<div>
  <p>To be SPIP or not to be ?</p>
  <pre>
    Nul ne le sait
      mais
        il convient d'y réfléchir
          à l'aide de cette citation :
  </pre>
  <hr />
  <blockquote lang="fr" cite="De l'âme, Aristote">
    « C'est aussi pourquoi il n'y a pas
      à rechercher si l'âme et le corps sont une
      seule chose, pas plus qu'on ne le fait pour
      la cire et l'empreinte, ni d'une manière générale
      pour la matière d'une chose quelconque et
      ce dont elle est la matière. »
  </blockquote>
  <hr />
</div>
```

- **<a>** **EN-LIGNE**
  - permet de naviguer au sein de la page ou vers une autre page
  - **attributs**
    - **href** : destination (obligatoire)
    - **title** : infobulle
    - **name** : pour nommer une ancre

```
<a href="http://www.spip.net" title="site SPIP">Cliquer ici</a>
```

```
<br />
```

```
<a href="#en_bas">Aller jusqu'à l'ancre en bas</a>
```

```
<br /><br /><br /><br /><br /><br /><br /><br /><br /><br />
```

```
<br /><br /><br /><br /><br /><br /><br /><br /><br /><br />
```

```
<a href="#" name="en_bas">ici</a>
```

## ■ `<img />` EN-LIGNE

- intègre l'image dans la page HTML
- **attributs**
  - **src** : source (obligatoire)
  - **alt** : texte alternatif pour autres moteurs de rendu ou pb (obligatoire)
  - **title** : info-bulle
  - **width** et **height** : taille

```

```

## ■ `<map>` BLOC

```
<img src ="planets.gif" width="145" height="126" alt="Planets"
usemap ="#planetmap" />
<map id ="planetmap" name="planetmap">
  <area shape="rect" coords="0,0,82,126" href="sun.htm" alt="Sun" />
  <area shape="circle" coords="90,58,3" href="mercur.htm" alt="Mercury" />
</map>
```

- **<table>, <tr>, <th>, <td>** **BLOC**
  - permet de intégrer un tableau de données dans la page
  - **attributs** (disparaîtront à terme)
    - **width** et **height** : taille
    - **border** : taille de la bordure

```
<table border="1" width="90%" height="300px">
  <tr>
    <th>Poisson</th>
    <td>20 euros</td>
    <td>500 grammes</td>
  </tr>
  <tr>
    <th>Viande</th>
    <td>15 euros</td>
    <td>900 grammes</td>
  </tr>
</table>
```



## ▪ `<ul>`, `<ol>`, `<li>` **BLOC**

```
<p>Voici une liste non-ordonnée :</p>
```

```
<ul>
```

```
  <li>Le XHTML est accessible</li>
```

```
  <li>Le XHTML est bien structuré</li>
```

```
  <li>Le XHTML est donne un sens à vos pages web</li>
```

```
</ul>
```

```
<p>Voici une liste ordonnée :</p>
```

```
<ol>
```

```
  <li>Le XHTML est accessible</li>
```

```
  <li>Le XHTML est bien structurer</li>
```

```
  <li>Le XHTML est donne un sens à vos page web :</li>
```

```
<ol>
```

```
  <li>C'est bon pour les internautes</li>
```

```
  <li>C'est bon pour les moteurs de recherche</li>
```

```
</ol>
```

```
</ol>
```

- `<dl>`, `<dt>`, `<dd>` **BLOC**
  - Comme les listes ne peut pas contenir d'autres éléments de type bloc que ceux notés ici

```
<p>Voici une liste de termes et de définitions :</p>
```

```
<dl>
```

```
  <dt>Développeur</dt>
```

```
  <dd>Un développeur est un informaticien qui  
    programme des logiciels.</dd>
```

```
  <dt>Web</dt>
```

```
  <dd>Réseau maillé de serveurs  
    d'informations formant une toile d'araignée.</dd>
```

```
</dl>
```

# Les balises de mise en forme

- **<em>**, **<strong>**, **<br />**, **<del>**, **<ins>**, **<sub>**, **<sup>**
  - mise en forme
- **<abbr>**, **<cite>**, **<q>**, **<dfn>**, **<code>**
  - abbréviations, définitions et citation de code

EN-LIGNE

```

<div>
  Une <em>information</em> <strong>importante</strong> doit être partagée.
  <br /> <br />
  Voici des <del>informations</del> <ins>propositions</ins> de réflexions
  <sub>basses</sub> et <sup>élevées</sup> :
  <br /> <br />
  <abbr title="Système de publication pour l'internet partagé"
  lang="fr">SPIP</abbr>
  <br /> <br />
  <q>To be or not to be.</q>
  <cite>Shakespear</cite>
  <br /> <br />
  <dfn>Un kiwi</dfn> est un fruit exotique mais aussi un petit animal.
  <br /> <br />
  En PHP, la variable langage sera <code>$langage = "XHTML";</code>
</div>

```

# Les objets embarqués

- **<object>**, **<param />** **EN-LIGNE**
  - permet d'inclure des animations multimédia à l'intérieur de la page
  - Flash, Active X, Applet Java...

```
<div>
  <object type="application/x-shockwave-flash"
  data="http://www.erasme.org/libre/divers/animations/calculatrice.swf"
  width="831" height="143">
    <param name="movie"
    value="http://www.erasme.org/libre/divers/animations/calculatrice.swf" />
    <param name="quality" value="high" />
  </object>
</div>
```

# Divers

- Les extensions **htm, html, xhtml, php**
  - peuvent contenir du code html
  - sont interprétées par les navigateurs
- Les tailles
  - peuvent être spécifiées en pixels ou pourcentage
  - seront préférentiellement placées dans les CSS
- Les caractères spéciaux
  - taper «caractères spéciaux» dans un moteur de recherche
- Pour en savoir plus :
  - <http://xhtml.le-developpeur-web.com/>
  - <http://www.aliasdmc.fr/courshtml/>
  - <http://www.fil.univ-lille1.fr/technoweb/docs/selfhtml/html/reference/elements.htm>

- Depuis le fichier *projet1.html* ajouter les éléments de contenu suivants :
  - Bloc Edito
    - L'image *images/arton1.gif*
    - Un titre `<h3>` : *Carte d'identité de la commune*
    - Un paragraphe `<p>` avec du contenu mise en forme : *Description de la commune...*
  - Bloc Article1
    - L'image *images/rubon3-15c52.gif*
    - Un titre `<h5>` *Titre* avec un lien `<a>`
    - La date en mise en forme emphatique `<em>`
    - Un paragraphe `<p>` avec du contenu mise en forme : *Texte...*
    - Un lien : *lire la suite*
- toute autre modification vous semblant intéressante

## *Partie 2*

# *Les feuilles de style*

*-CSS 2-*



# *Présentation des feuilles de style*





# Définition

- **Cascading Style Sheets = Feuilles de Style en Cascade**
  - Langage utilisé pour décrire la présentation d'un document structuré en (X)HTML
  - Le standard est dicté par le World Wide Web Consortium ([www.w3c.org](http://www.w3c.org)), les navigateurs essayent de le respecter
  - Versions : CSS 2 (>1998) et CSS3 (en cours)
- **Séparation du fond de la forme**
  - Les pages XHTML décrivent la structure et le contenu
  - Les CSS décrivent la présentation pour l'affichage sur un média donné (screen, palm, imprimante...)

# *Par la pratique*

- **CSSZenGarden**

- 1 page HTML <-> n feuilles de styles
  - <http://www.csszengarden.com/tr/francais/>
  - <http://www.mezzoblue.com/zengarden/alldesigns/official/>

- **Désactiver les styles via FireFox**

- Affichage > Style de la page > Aucun style
  - <http://fr.wikipedia.org/wiki/Accueil>
  - <http://fr.yahoo.com/>

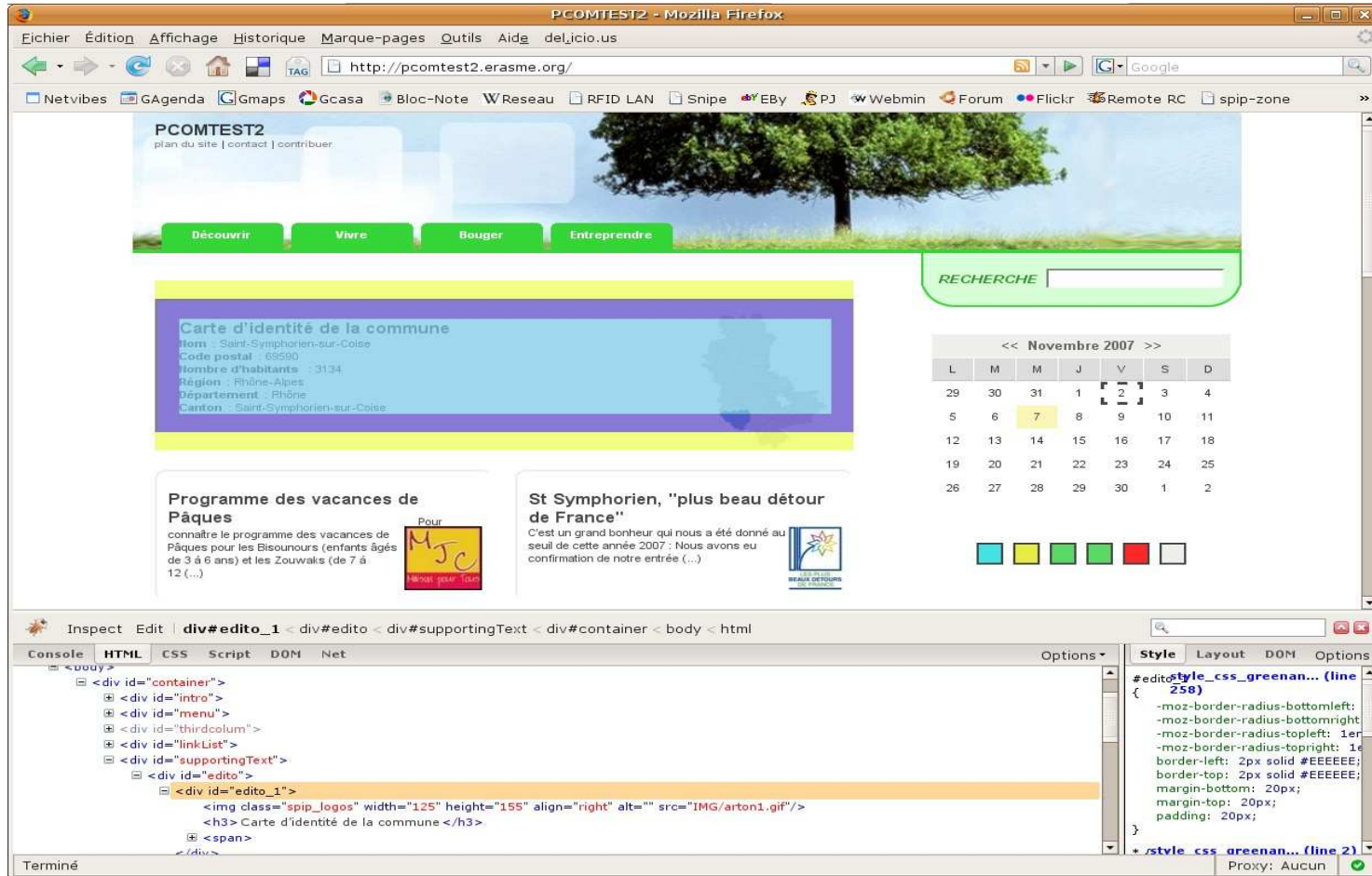
- **Chaque balise XHTML a un rendu par défaut dans le navigateur**

- le style CSS relatif à la balise vient **surcharger** ce rendu

# Firebug

- Les extensions Firefox

- WebDeveloper, FireBug, IETab
- <https://addons.mozilla.org/fr/firefox/browse/type:1/cat:4/sort:popular>



# *Pourquoi les CSS ?*

- Les CSS assure la cohérence graphique d'un site.
- La modification de la charte graphique d'un site web est facilitée par l'utilisation des CSS : seule la feuille de style est modifiée.
- Le code XHTML de la page est allégé, il y a donc une diminution du poids et par conséquent du temps de chargement.
- La compatibilité est améliorée
  - ...même si le navigateur le plus utilisé n'affiche que 60% de compatibilité aux CSS2

# Implémentation des styles

- **Dans les balises**

- à éviter car on perd l'aspect global du style

```
<body>
  <p style="/* Mes styles */">texte</p>
</body>
```

- **Dans la page**

- pour des maquettes et des feuilles simples à tester

```
<head>
  <style type="text/css">
    /* Mes styles */
  </style>
</head>
```

# Implémentation des styles (2)

- **Dans une feuille de style extérieure**
  - solution à retenir pour externaliser et mutualiser la feuille

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

OU

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

# Implémentation des styles (3)

- possibilité de spécifier le **média** dédié au style
  - **all**, **screen**, **print**, projection, braille, embossed, aural, speech...
  - seuls pour l'instant les medias d'affichage fonctionnent correctement
- possibilité de spécifier des feuilles alternatives avec **title**
  - Firefox permet de y accéder dans >Affichage>Styles

```
<head>
  <style type="text/css" media="screen" title="style1">
    /* Mon style1 */
  </style>
  <style type="text/css" media="screen" title="style2">
    /* Mon style2 */
  </style>
  <style type="text/css" media="print">
    /* Style impression */
  </style>
</head>
```

# *Syntaxe*





# Déclaration des styles

```
balise
{
  propriete: valeur;
  propriete: valeur;
  propriete: valeur;
}
```

sélecteur      propriété      valeur

↓                   ↓                   ↓

```
p { font-family: Verdana; }
```

↑

déclaration

- balise : nom de la balise dont on modifie l'apparence
  - <em>, <b>, <h1>
- propriété : catégories d'effet de style
  - color, font-size, font-family
- valeur : valeur associée à la propriété
  - red, 10px, Verdana

# Syntaxe

- On peut ajouter des **commentaires** `/*...*/`
- Une déclaration ouverte doit toujours être fermée `{...}`
- Une définition de valeur doit se terminer par `;`  
On peut enchaîner plusieurs définitions de valeurs au sein d'une même déclaration.
- Les espaces et les tabulations ne sont pas interprétés.

```
balise
{
    /* Première déclaration */
    propriete: valeur;
}
```

# Premiers exemples

```
body
{
  font-family: Verdana;
  font-size: 12px;
  color: #484747;
  background-color: #EEEEEE;
}
```

```
div
{
  padding: 10px;
  margin: 10px;
  border: black solid 1px;
}
```

```
h1
{
  font-size: 250%;
  color: gray;
  font-weight: bold;
}
```

```
a,a:link,a:visited
{
  color: #484747;
  text-decoration: none;
}

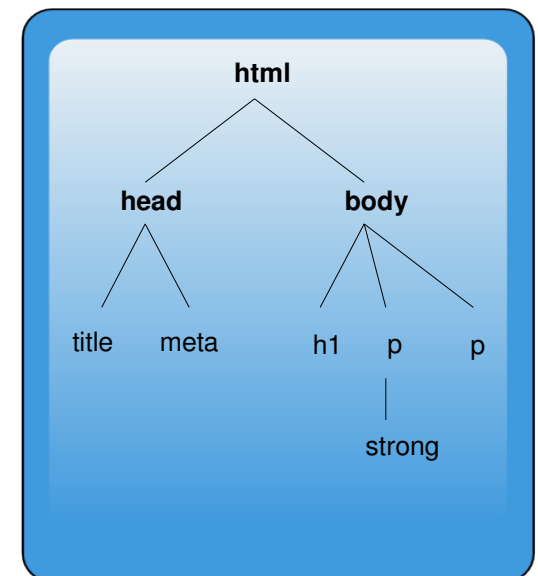
a:hover
{
  color: red;
  text-decoration: underline;
}
```

# Feuilles en cascade

- Les styles sont appliqués successivement dans l'ordre suivant (par ordre croissant d'importance) :
  - 1. Styles extérieurs
  - 2. Styles de la page
  - 3. Style dans la balise elle-même
- En cas de surcharge, le dernier l'emporte

- Les propriétés de chaque balise sont héritées des éléments parents dans l'arbre du document

transmission  
des propriétés



# Couleurs

- Une couleur peut être définie par
  - son nom (si il existe) : **green**
  - son code hexadécimal long : **#008080**
  - son code hexadécimal court : **#FFF**
  - son code RGB : **rgb(0,0,255)** ou **rgb(10%,10%,30%)**
  - **transparent**
- Les 16 couleurs de base du Web (compatibles W3C)

#00FFFF Aqua	#000000 Black	#0000FF Blue	#FF00FF Fuchsia
#000080 Navy	#808000 Olive	#800080 Purple	#FF0000 Red
#808080 Gray	#008000 Green	#00FF00 Lime	#800000 Maroon
#C0C0C0 Silver	#008080 Teal	#FFFFFF White	#FFFF00 Yellow

- <http://www.webmaster-hub.com/outils/color.html>

# Unités de valeur

- La taille et la position des éléments doivent comporter une unité :
  - **font-size** : 12px;
  - **width**: 90%;
  - **margin**: 0;
- Rappports
  - **%** (pourcentage) : la taille / position de l'élément est calculée de façon relative par rapport à son parent
  - **em** : 1em = 100%, 1.2em = 120%, etc.
- Mesures
  - **px** : pixels, plus petite unité de l'écran
  - **pt** : points, plus petite unité de l'imprimerie

- A partir de votre dernière version de *projet2.html* ou depuis *projet3.html*
  - créer un espace pour votre style **<style>** dans la page avec les attributs **media** et **title**
  - ajouter les styles vus en exemple pour les éléments
    - body
    - a et a:hover
    - h1
    - div
  - tester le résultat dans le navigateur à chaque étape
  - faire quelques essais de couleur via le sélecteur de couleur en ligne

# Les sélecteurs

*bal* remplace n'importe quelle balise

## \* {} sélecteur universel

- désigne tous les éléments

## bal {} sélecteur de balise

- désigne toutes les balises *<bal>* de la page

## .nom\_classe {} sélecteur de classe

- désigne les éléments ayant l'attribut *class="nom\_classe"*

## bal.nom\_classe {} sélecteur de classe

- désigne toutes les balises *bal* ayant l'attribut *class="nom\_classe"*



# Les sélecteurs (2)

*bal* remplace n'importe quelle balise

## **#nom\_id { }** sélecteur d'identification

- désigne l'élément ayant reçu `id="nom_id"`

## **bal#nom\_id { }** sélecteur d'identification

- désigne la balise `<bal>` ayant reçu `id="nom_id"`

## **bal1 bal2 bal3 { }** hiérarchie

- désigne les éléments `<bal3>` contenus dans une balise `<bal2>` lui-même contenu dans une balise `<bal1>`

## **bal1, bal2, bal3 { }** sélecteur collectif

- s'applique aux balises `<bal1>`, `<bal2>` et `<bal3>`

# Les sélecteurs (3)

*bal* remplace n'importe quelle balise

## **bal1 + bal2 { }**      **Sélecteur adjacent**

- désigne les *<bal2>* directement placés derrière un *<bal1>*

## **bal1 > bal2 { }**      **Sélecteur d'enfants**

- désigne les *<bal2>* directement placés dans un *<bal1>*

# Pseudo classes et éléments

Note : les balises sont citées à titre d'exemple

**p**:first-letter|first-line|after { }

**pseudo-élément**

- désigne un sous ensemble de l'élément `<p>`
  - first-letter : première lettre
  - first-line : première ligne
  - after : ajout après l'élément

**img**:active|hover|focus|visited { }

**pseudo-classes dyn.**

- désigne l'élément `<img>` dans un état en temps réel
  - hover : l'élément est survolé
  - active : souris pressée et non relâchée
  - focus : l'élément est activé (un formulaire par exemple)
  - link : lien non visité (`<a>` uniquement)
  - visited : lien visité (`<a>` uniquement)

# Comment écrire les sélecteurs qui s'appliquent au texte en vert ?

```
<strong>c'est important</strong>
```

```
<div id="principal">  
  Mon texte  
</div>
```

```
<a href="#" class="liens">  
  cliquer ici  
</a>
```

```
<div class="home">  
  <strong>c'est important</strong>  
</div>
```

# Comment écrire les sélecteurs qui s'appliquent au texte en vert ?

```
<strong>c'est important</strong>
```

**strong { ... }**

```
<div id="principal">  
  Mon texte  
</div>
```

**#principal { ... }**

```
<a href="#" class="liens">  
  cliquer ici  
</a>
```

**a, a:hover { ... }**

ou

**a.liens { ... }**

**a.liens:hover { ... }**

```
<div class="home">  
  <strong>c'est important</strong>  
</div>
```

**div.home strong { ... }**

- A partir de votre dernière version de *projet4.html*
  - ajouter les styles suivants au bloc edito
    - la couleur de fond de votre choix (ex : #cc3300) et la propriété :  
*-moz-border-radius:10px;*
    - titre de niveau 3 contenu dans l'edito, les propriétés suivantes :  
*font-size: 17px;*  
*font-weight: bold;*
    - images contenu dans l'edito, les propriétés suivantes :  
*float:right; /\*alignement à droite\*/*
    - images contenu dans l'edito, lors du survol l'ajout d'une bordure
  - augmenter la taille de la première lettre de tous les paragraphes de la page
  - tester le résultat dans le navigateur

# *Mise en forme*



# Typographie

- Polices

- Elles ne sont pas toutes présentes sur tous les ordinateurs
- Elles se déclarent dans un ordre de préférence
- On termine par les polices standard : sans-serif, serif, monospace, cursive.

<b>Arial Black</b>	Palatino
Verdana <b>Impact</b>	Georgia Times
Trebuchet MS	Garamond
Helvetica Geneva	<b>COPPERPLATE</b>
<small>POLICES SANS-SERIF</small>	<small>POLICES SERIF</small>

Courier	Comic Sans MS
Courier New	<i>Zapf Chancery</i>
Monaco	<i>Brush Script</i>
<b>Lucida</b>	
<small>MONOSPACE</small>	<small>CURSIVE (MANUSCRIT)</small>

- Les espacements

- Peuvent être fixés par les attributs line-height et letter-spacing

letter spacing \_\_\_\_\_

letter spacing \_\_\_\_\_

letter spacing \_\_\_\_\_

line height \_\_\_\_\_

line height \_\_\_\_\_

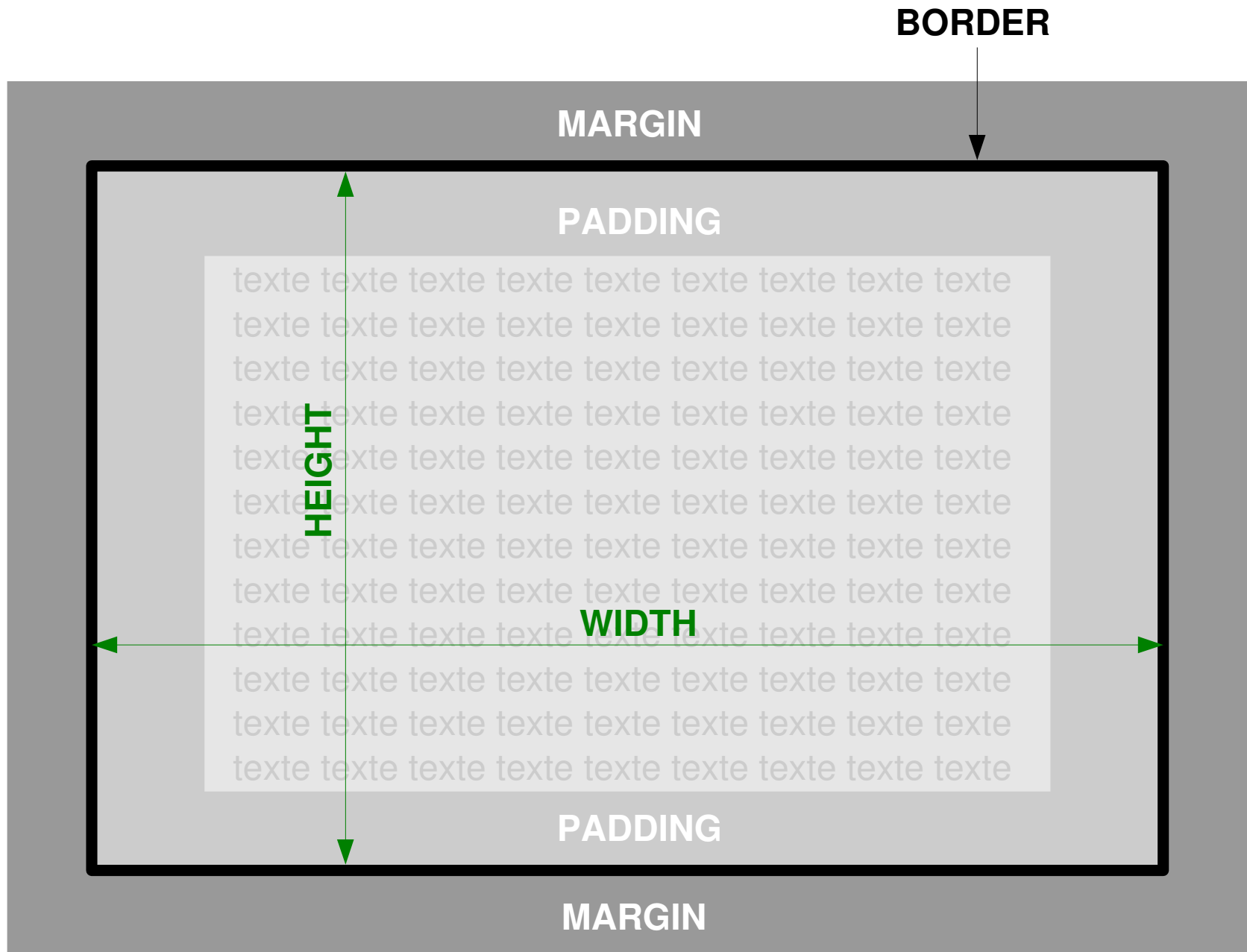
HAUTEUR DE LIGNE



# Typographie

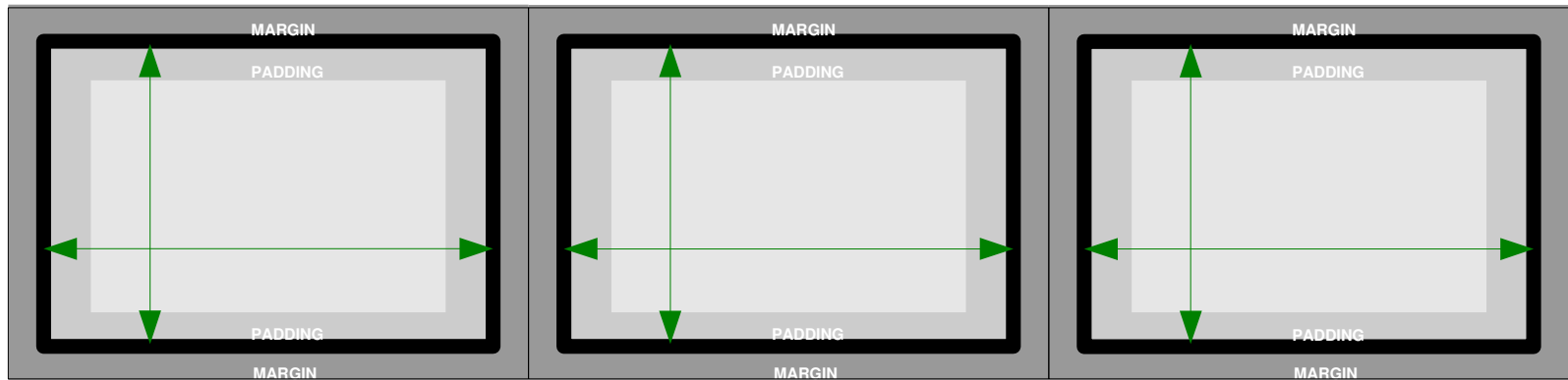
Action	Propriété	Valeur	Description
Choix de la police	<b>font-family :</b>	<b>"Trebuchet MS", arial, sans-serif;</b>	applique les polices par ordre décroissant ; mettre les noms avec espace entre ""
Taille de la police	<b>font-size :</b>	<b>10px; 1.2em; 90%;</b>	préférer em et % pour hériter des valeurs des éléments parents
Couleur	<b>color :</b>	<b>silver; #0033FF; transparent;</b>	
Orientation de la police	<b>font-style :</b>	<b>normal; italic; oblique;</b>	
Epaisseur	<b>font-weight :</b>	<b>bold; bolder; normal;</b>	
Soulignement	<b>text-decoration :</b>	<b>none; underline; overline; line-through;</b>	
Mise en majuscule	<b>text-transform :</b>	<b>none; capitalize; uppercase; lowercase;</b>	capitalize: première lettre uniquement.
Alignement du texte	<b>text-align :</b>	<b>left; right; center; justify;</b>	
Hauteur de ligne	<b>line-height :</b>	<b>normal; 1.2em; 90%; 10px;</b>	
Espacement entre les lettres	<b>letter-spacing :</b>	<b>normal; -2px;</b>	
Espacement entre les mots	<b>word-spacing :</b>	<b>normal; 3px;</b>	
Raccourci	<b>font :</b>	<b>font-style, font-variant, font-weight, fontsize/line-height, font-family</b>	<b>p { font: bold 12px/24px verdana }</b>

# Présentation des blocs



# Compléments

- Plusieurs blocs : les margins s'ajoutent



- Quelles est la taille d'un bloc total (boîte) ?
  - Selon le standard : **boîte = margin+border+width**
  - Pour IE : **boîte = width**
  - -> Privilégier les padding aux margin...

# Présentation des blocs

- Tout élément en bloc possède 4 attributs :
  - taille propre : *width* et *height*
  - bordure : *border*
  - marge intérieure : *padding*
  - marge extérieure : *margin*

Action	Propriété	Valeur	Notes
Largeur	<b>width :</b>	<b>30%; 250px; auto;</b>	auto calcule automatiquement la taille actuelle (image par exemple)
Hauteur	<b>height :</b>	<b>30%; 250px; auto;</b>	

# Marges et espacements

Description	Propriété	Exemple
Marge supérieure	<code>margin-top :</code>	<code>5px;</code>
Marge de droite	<code>margin-right :</code>	<code>0.5em;</code>
Marge inférieure	<code>margin-bottom :</code>	<code>2pt;</code>
Marge de gauche	<code>margin-left :</code>	<code>0;</code>
Raccourci pour les marges	<code>margin :</code>	<code>-5px 0.5em 2pt 0;</code> <code>auto;</code> (alignement centré du bloc)
Espace intérieur entre l'élément et la bordure supérieure	<code>padding-top :</code>	<code>3px;</code>
Espace intérieur entre l'élément et la bordure droite	<code>padding-right :</code>	<code>0.25em;</code>
Espace intérieur entre l'élément et la bordure inférieure	<code>padding-bottom :</code>	<code>0;</code>
Espace intérieur entre l'élément et la bordure gauche	<code>padding-left :</code>	<code>2pt;</code>
Raccourci vers l'ensemble des propriétés d'espace intérieur	<code>padding :</code>	<code>3px 0.25em 0 2pt;</code>

# Arrières plan

Action	Propriété	Valeur	Notes
Couleur d'arrière-plan	<b>background-color :</b>	"#RRGGBB";	
Image d'arrière-plan	<b>background-image :</b>	url( <a href="http://url">http://url</a> );	
Répétition de l'arrière-plan	<b>background-repeat :</b>	repeat; repeat-x; repeat-y; no-repeat;	
Spécifie si l'image reste fixe avec les déplacements de l'écran	<b>background-attachment :</b>	scroll; fixed;	
Position de l'image par rapport au coin supérieur gauche	<b>background-position :</b>	top; middle; bottom; left; center; right;	possibilité d'indiquer des valeurs en pixels
Raccourci global vers les propriétés des AP	<b>background :</b>	#000000 url(test.jpg) no-repeat scroll center top;	

```
body
{
    background: transparent url(images/siteon0.jpg)
    no-repeat scroll center top;
}
```

# Bordures

Description	Propriété	Valeur
Epaisseur de la bordure	<b>border[-top -left -bottom -right]-width :</b>	<b>10px; 2em;</b>
Epaisseur de la bordure	<b>border-width :</b>	<b>10px 15px 15px 10px; (HDBG)</b>
Couleur de la bordure	<b>border[-top -left -bottom -right]-color :</b>	<b>#RRGGBB;</b>
Style de la bordure	<b>border[-top -left -bottom -right]-style :</b>	<b><i>solid;</i> (pleine) <i>dashed;</i> (en tirets) <i>dotted;</i> (en pointillés) <i>double;</i> (double) <i>ridge;</i> <i>inset;</i> <i>outset;</i> (en 3D)</b>
Effet arrondi (Mozilla)	<b>-moz-border-radius :</b>	<b>10px; 2em; 10px 10px 10px 10px; (HDBG)</b>
Raccourci global les propriétés de bordure	<b>border :</b>	<b>border: 1px 0 0 2px dotted green;</b>

```
body
{
    border: 1px 0 0 2px dotted green;
}
```

- A partir de votre dernière version de *projet5.html* définir la présentation des blocs suivants et le tester dans le nav :
  - **page**
    - largeur : 1000 pixels
    - marge : automatique
    - couleur de fond : de votre choix (ex : #FFFFFF) ou logo de saint symphorien : images/logo\_saint\_sym.gif
  - **entete**
    - image de fond : images/siteon0.jpg
  - **edito**
    - bordures haute et gauche : #eee 2px solid
    - bordures basse et droite : 0px
    - bords arrondis : 10 px
  - harmoniser les couleurs : polices, background body et page



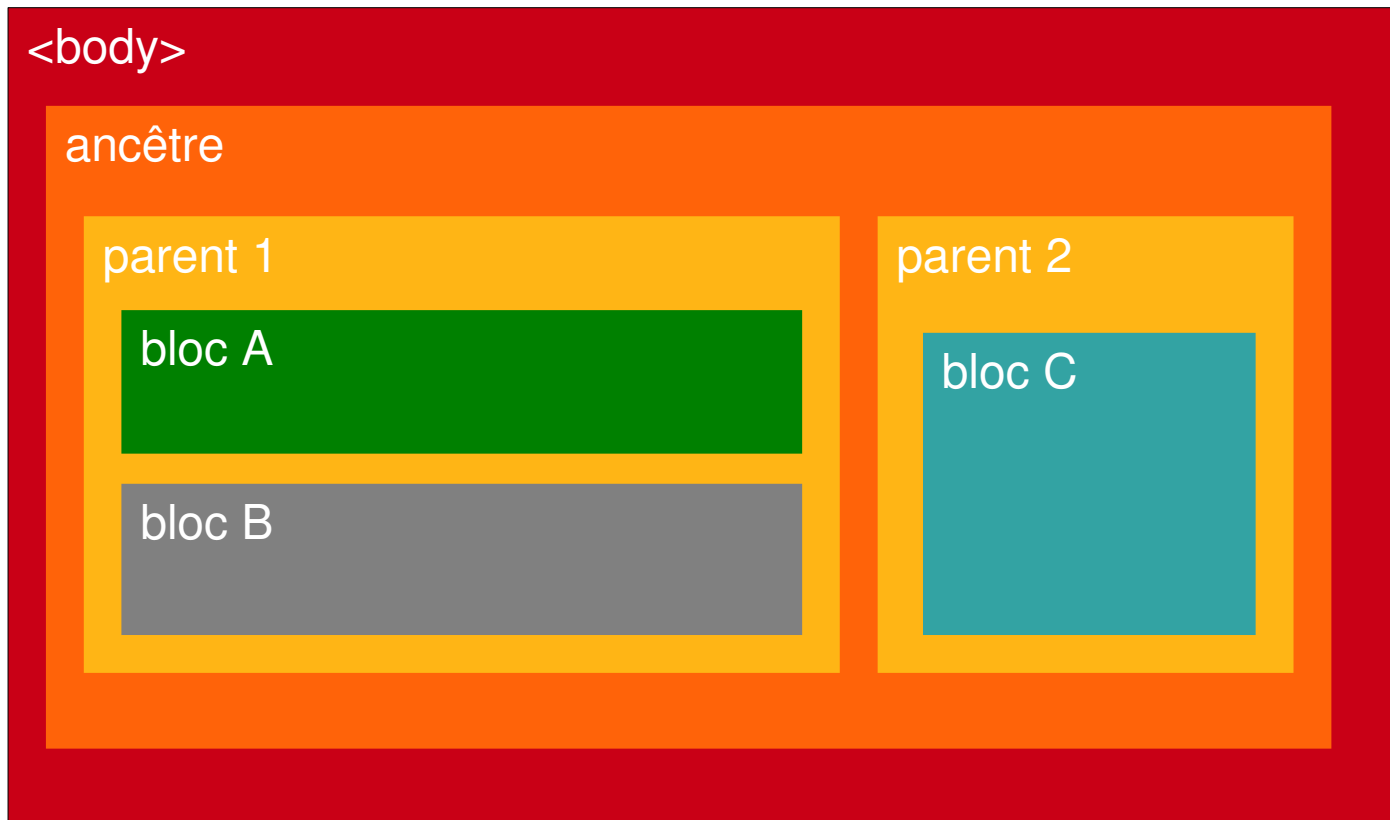
# *Positionnement*



# Propriétés de position

Description	Propriété	Valeur
Propriétés	<b>display :</b>	<i>none;</i> (n'affiche pas le bloc) <i>block;</i> (force la propriété bloc) <i>inline;</i> (force la propriété en-ligne) <i>list-item;</i> (force la propriété en-ligne) <i>table;</i> (force la propriété tableau)
Affichage	<b>visibility :</b>	<b>visible;</b> <b>hidden;</b> (cache l'élément mais réserve un espace pour son affichage)
Position	<b>position :</b>	<b>static;</b> (default - dans le flux) <b>relative;</b> (décalage - dans le flux) <b>absolute;</b> (fixe - hors du flux) <b>fixed;</b> (fixe - hors du flux - noscroll)
Distance au parent	<b>top :</b> / <b>right :</b> / <b>bottom :</b> / <b>left :</b>	<b>-10px;</b> <b>30%;</b> <b>2em;</b> (pas pour static)
Profondeur d'affichage (calque)	<b>z-index :</b>	<b>auto;</b> <b>1000;</b>
Positionnement flottant	<b>float :</b>	<b>left;</b> <b>right;</b> <b>none;</b> (default)
Efface le flottement	<b>clear :</b>	<b>left;</b> <b>right;</b> <b>both;</b> <b>none;</b> (default)

# Hiérarchie des éléments



- A et B sont frères
- A, B et C ont le même ancêtre
- Tous les éléments sont contenus dans `<body>`

- De quelle nature sont bloc A et bloc B ? (bloc ou en ligne)
- Parent 1 et Parent 2 ?

# Le flux de document

position par défaut

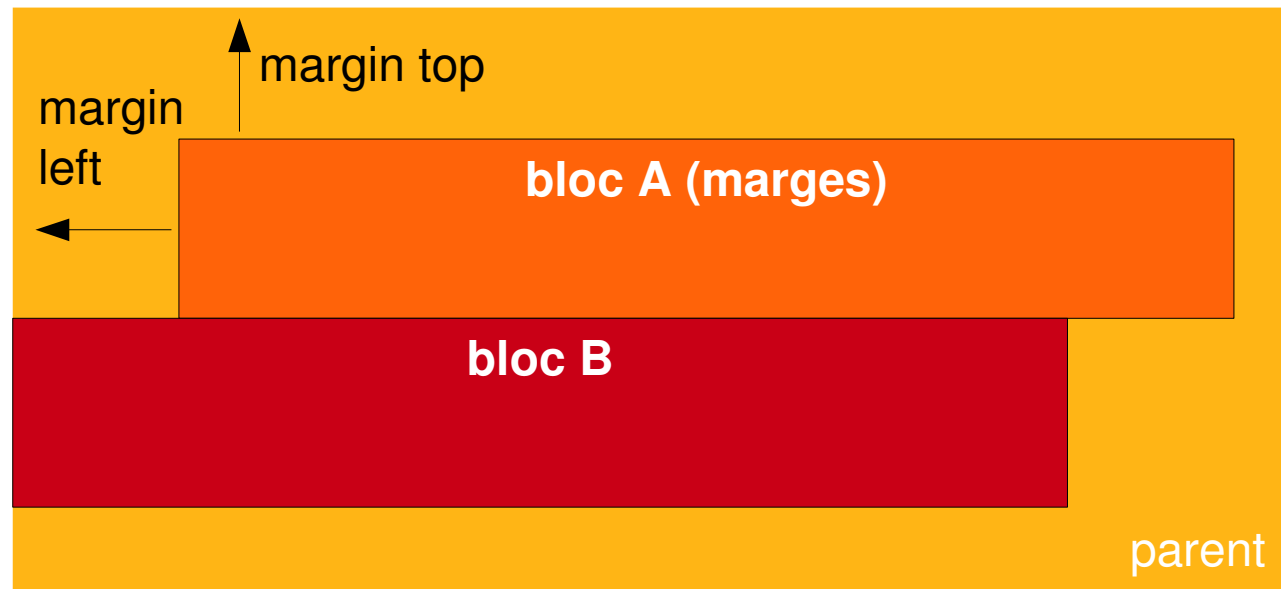
position : relative;

- Par défaut les éléments se placent en haut à gauche de leur parent et suivent le flux de document :
  - 1. **l'ordre** dans lequel ils sont inscrits dans le code HTML
  - 2. **leur nature** (bloc ou en-ligne)
    - bloc : en dessous du précédent
    - en-ligne : à côté du précédent



# *Le flux de document (2)*

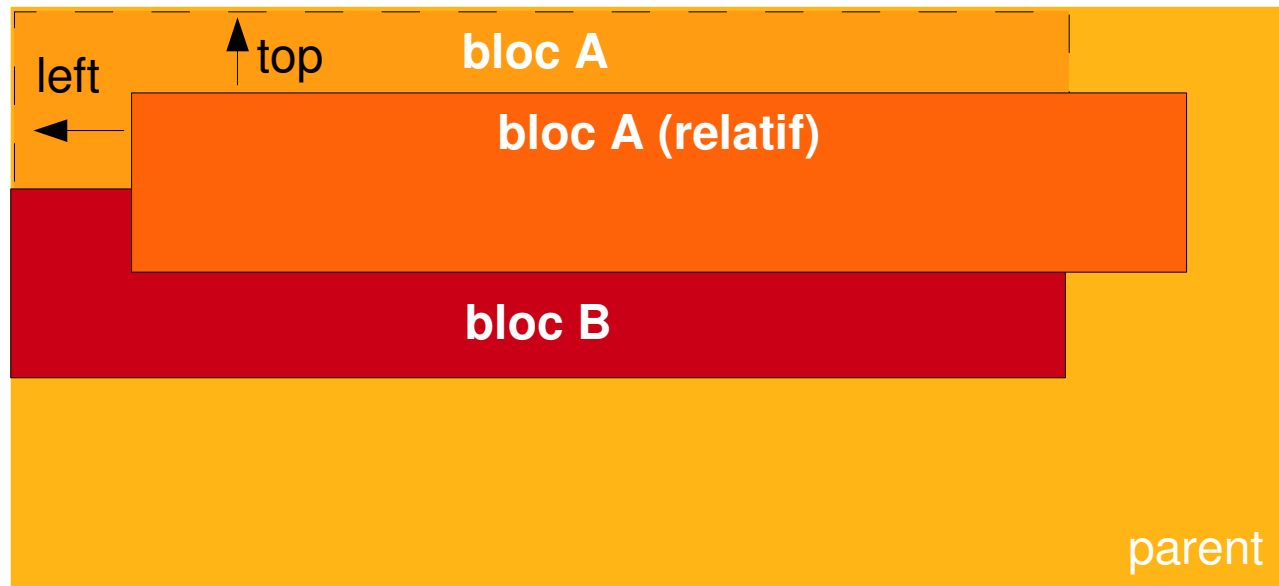
- 3. **leurs marges** internes (padding) et externes (margin)



note : il s'agit bien des marges externes (margin) du bloc et non des marges internes (padding) du parent

## Le flux de document (3)

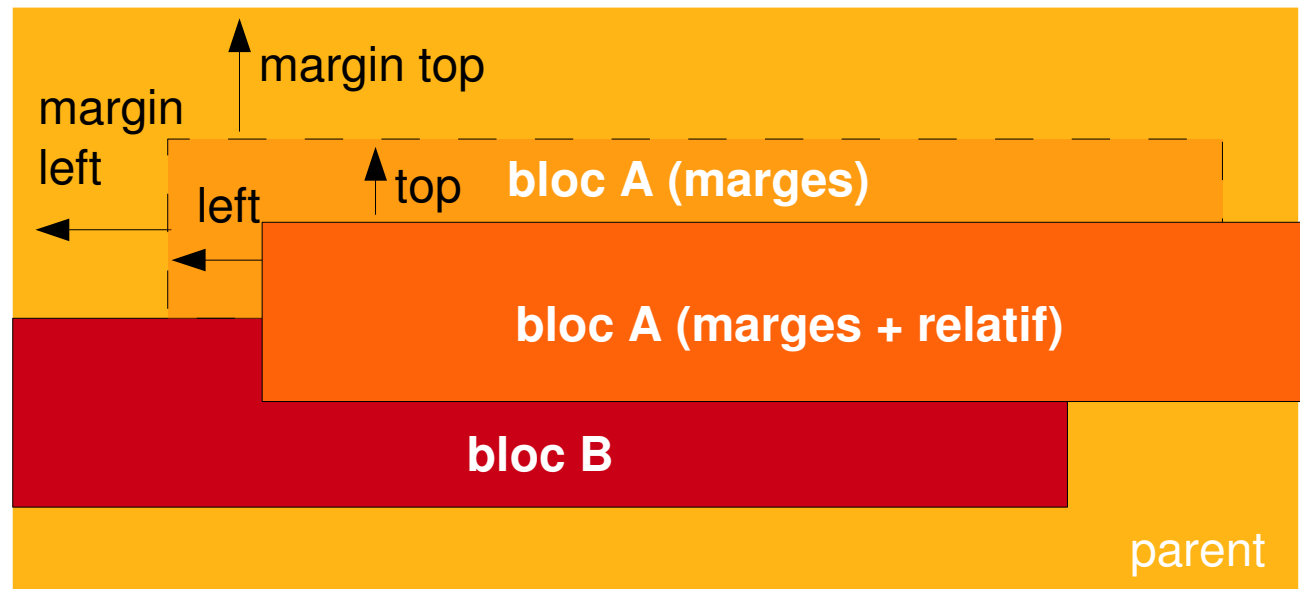
- 4. **les décalages** (top, right, left, bottom) lorsque les blocs sont en position relative



- les éléments frères ne sont pas influencés par le décalage : ils se positionnent comme si le bloc A était dans le flux
- cet exemple n'a pas de marges en plus des décalages
- le z-index est utile dans de tels cas

# Positionnement en flux (exemple)

- marges + décalage



- la taille du parent est influencée par celle des enfants si :
  - ils utilisent des marges
  - le parent possède une taille non fixe (% , em , pas de taille)
- les décalages n'ont pas d'influence sur la taille du parent

# Positionnement flottant

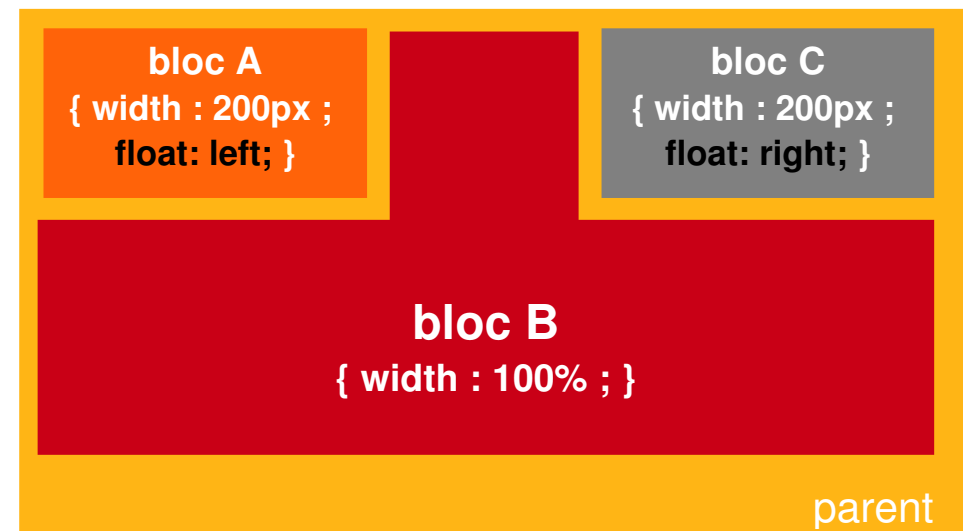
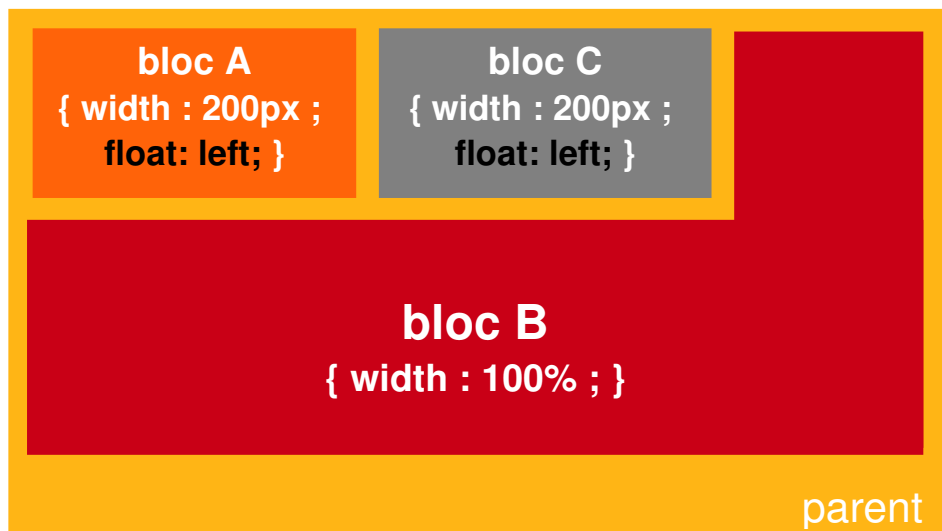
position par défaut

position : relative;

- Propriété **float** : `float: right;` `float: left;`
- Un élément flottant sort du flux normal pour prendre place à gauche ou à droite du bloc qui le contient.
- L'élément qui le suit s'écoulera dans l'espace laissé libre en épousant sa forme.
- Notes
  - L'avantage est de pouvoir placer des blocs côte à côte
  - PIEGE: les flottants sortent du flux normal ce qui implique qu'ils n'influencent plus la hauteur du conteneur. Si l'on prend un paragraphe flottant plus gros que les autres, il débordera du conteneur.



# Positionnement flottant

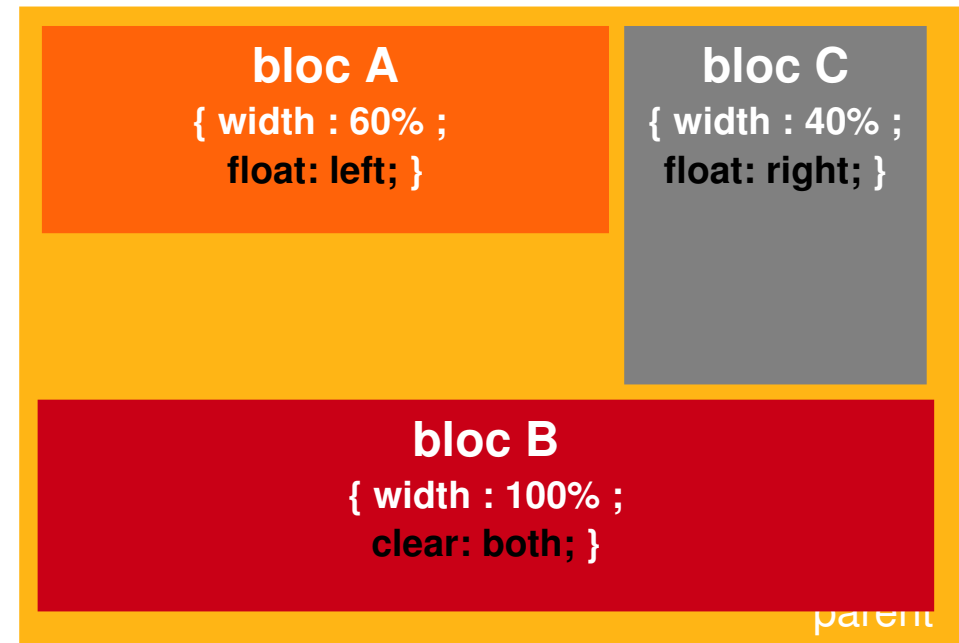
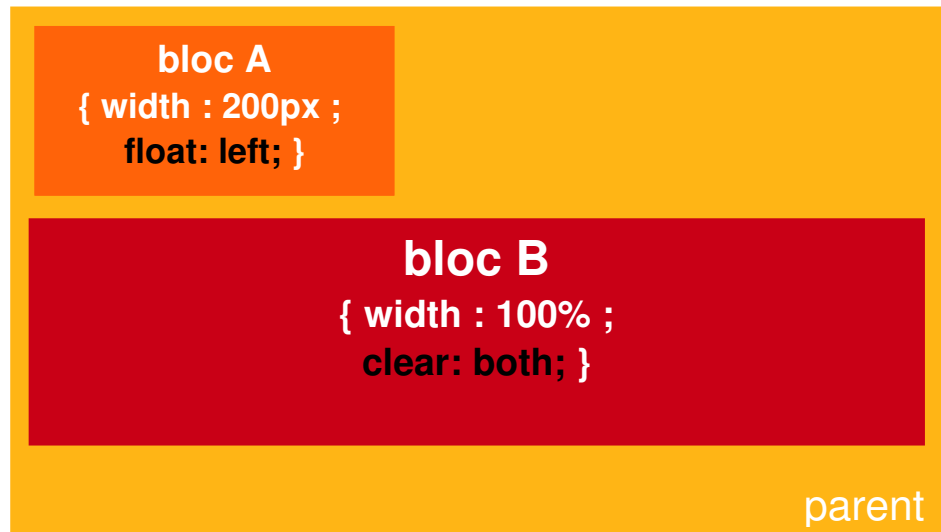


# Positionnement flottant

position par défaut

position : relative;

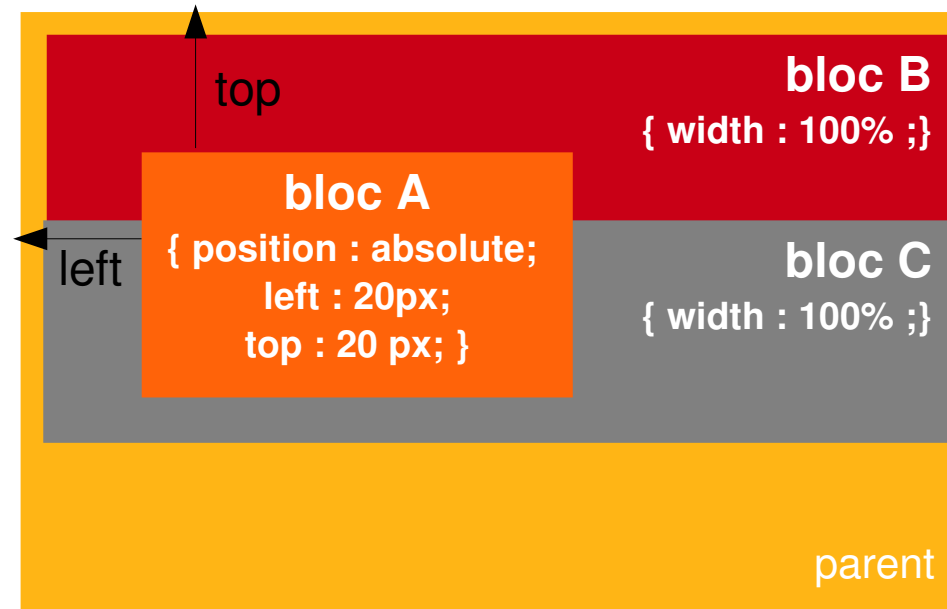
- Propriété **clear** : `clear: both;`
- Empêche un élément du flux de se trouver sur la même ligne qu'un élément flottant.
- Typiquement utilisé pour les pieds de page.



# Positionnement absolu

position : absolute;

- `position: absolute; top: 2em; left: 0;`
  - l'élément se place par rapport au dernier parent ayant une position absolue. Par défaut : `<body>`
  - l'élément sort du flux et n'affecte pas les autres blocs
  - on utilise les attributs `top`, `left`, `right`, `bottom` pour le placer
- Pour des éléments indépendants de l'environnement

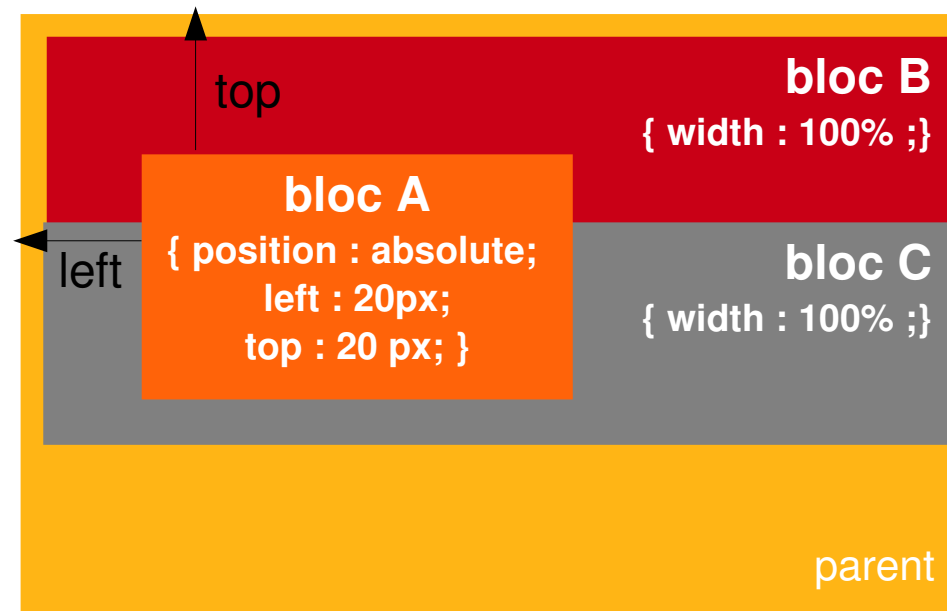


# Positionnement fixe

position : fixed;

- ```
position: fixed; top: 2em; left: 0;
```

  - Identique à la position absolue
  - Différence : l'élément ne bouge pas lorsque la page défile avec l'ascenseur
- Pour des effets de style. Ne pas en abuser.



- A partir de votre dernière version de *projet6.html* définir le positionnement des blocs suivants et les tester :
  - **colonne\_gauche**
    - width: 650px;
    - float: left;
  - **colonne\_droite**
    - width: 250px;
    - float: right;
  - **footer**
    - clear: both;

# *Compléments*



# Curseur

- Détermine l'apparence du curseur au survol
- Valeurs
  - **default** : Curseur par défaut
  - **pointer** : Main
  - **crosshair** : Viseur
  - **help** : Point d'interrogation
  - **wait** : Attente
  - **text** : Texte
  - **move** : Objet déplaçable
  - **...-resize** : Changer la taille en fonction de la direction

```
élément  
{  
    cursor : apparence;  
}
```

# Listes

| Action                                           | Propriété                          | Valeur                                                                                    |
|--------------------------------------------------|------------------------------------|-------------------------------------------------------------------------------------------|
| Type de puces et de numérotation                 | <code>list-style-type :</code>     | <code>decimal, upper-roman, lower-latin, disc, circle, square</code> ou <code>none</code> |
| Permet de personnaliser les puces avec une image | <code>list-style-image :</code>    | <code>url(image.png);</code>                                                              |
| Spécifie le retrait des puces                    | <code>list-style-position :</code> | <code>inside; outside;</code>                                                             |
| Raccourci global vers les propriétés des listes  | <code>list-style :</code>          | <code>type position url();</code>                                                         |

- Des menus avec des listes
  - <http://css.alsacreations.com/Galleries-de-menus-en-CSS>
  - <http://www.alsacreations.com/livre/?/Exemples/exdouze>
  - <http://www.cssplay.co.uk/menus/visitedmenu.html>



```
<div id="menu">
  <ul>
    <li><a href="#">Vie municipale</a></li>
    <li><a href="#">Vie culturelle</a></li>
    <li><a href="#">Economie</a></li>
    <li><a href="#">Actualités</a></li>
  </ul>
</div>
```

- Créer votre menu horizontal à l'aide des styles suivants :

```
#menu ul
{
  padding-left:20px;
  list-style-type:none;
}

#menu li
{
  float:left;
  width:112px;
  margin-left:5px;
}

#menu a, #menu a:visited
{
  display:block;
  font-size:11px;
  font-weight: bold ;
  text-decoration:none;
  color: #fff;
  width:112px;
  height:30px;
  background-image:url(images/corner.gif);
  text-align:center;
  line-height:29px;
}

#menu a:hover, #menu ul ul a:hover
{
  color:#249526;
  background-image:url(images/cornerb.gif);
}
```

# Compatibilité entre les navigateurs

- <http://www.webdevout.net/browser-support-summary>
  - Gecko (Firefox, Mozilla...) ainsi que le moteur Khtml (Safari, Konqueror) sont de très bonnes valeurs.
  - Un seul navigateur est obsolète: Internet Explorer (IE)... qui représente 80% des navigateurs utilisés sur Internet.
- Le dilemme du développeur :
  - On ne peut pas développer un site accessible sans tenir compte de IE
  - On ne peut pas développer un site uniquement pour IE
    - Ne respecte pas les standards
    - Solution non pérenne
    - Solution non accessible
    - Solution propriétaire

# Accessibilité

- Accessiweb, créé par l'association BrailleNet, a établi un ensemble de règles pour qu'un site respecte les recommandations du W3C (WAI)
  - <http://www.accessiweb.org/>
- Recommandations :
  - Utiliser les standards
  - Concevoir une interface cohérente
  - Concevoir une présentation claire et aérée
  - Les liens doivent guider efficacement l'internaute
  - Proposer des alternatives

# *Outils de validation*

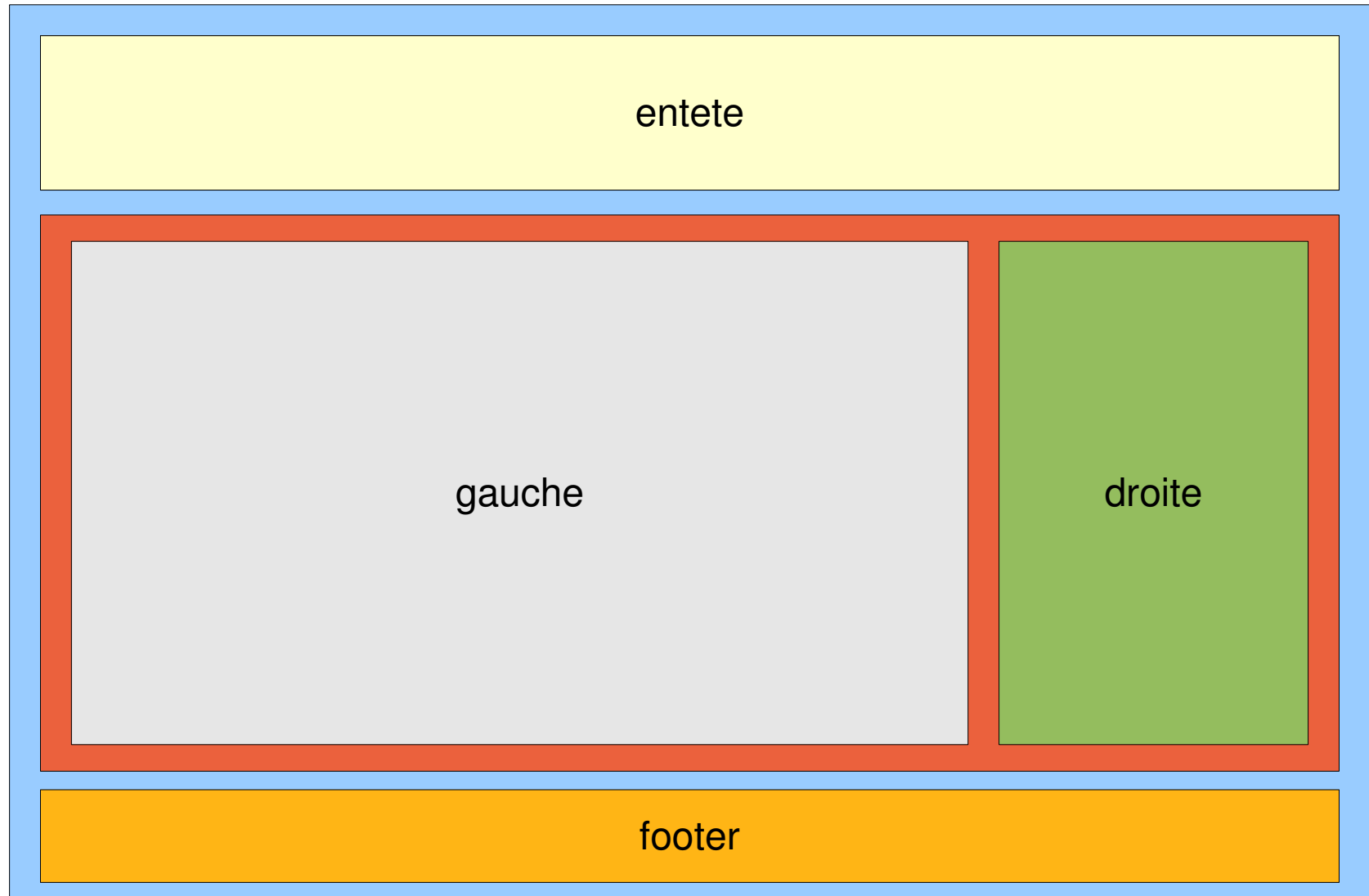
Les outils de validation sont des partenaires précieux :

- Outil de validation pour XHTML:
  - <http://validator.w3.org/>
- Outil de validation pour CSS2:
  - <http://jigsaw.w3.org/css-validator/>
- Outils de validation pour l'accessibilité d'un document:
  - WebXact: <http://webxact.watchfire.com/>
  - Wave 3: <http://www.wave.webaim.org/>
  - Accès-pour-tous: <http://www.acces-pour-tous.net/>

# Notion de template

- Template = canevas = modèle de mise en forme
- Un template est composé de deux éléments
  - Une page de structure et de données en **XHTML**
  - Une feuille de style de présentation en **CSS**
- La création de template est l'étape préalable à la création des squelettes
  - -> il ne restera qu'à remplacer les données statiques par des données dynamiques grâce au le langage de boucle de SPIP

# *Le canevas de page classique : une entête, deux colonnes*



# *D'autres templates*

- Des templates XHTML+CSS en ligne :
  - <http://www.free-css-templates.com>
  - <http://www.oswd.org/designs/search/advanced/>
  - [http://www.zymic.com/show\\_templates.php?id=1](http://www.zymic.com/show_templates.php?id=1)
  - <http://themes.wordpress.net>
- Des squelettes SPIP (XHTML+CSS+SPIP) en ligne :
  - <http://www.spip-contrib.net/-Squelettes->

# *Pour en savoir plus sur les CSS*

- Merci aux auteurs de ces contributions :
  - <http://www.pompage.net/pompe/cssdezero-2/>
  - <http://mammouthland.free.fr/cours/css/>
  - <http://openweb.eu.org/css/>
  - <http://css.alsacreations.com>
  - <http://fr.selfhtml.org/css/proprietes/index.htm>
  - <http://css.alsacreations.com/Bases-et-indispensables/Comprendre-le-positionnement-des-balises-en-C>



# Compléments XHTML / CSS

- Des éditeurs CSS / XHTML avec interfaces graphiques
  - CSS Editor : <http://www.framasoft.net/article1783.html>
  - Nvu : <http://www.framasoft.net/article2243.html>  
<http://frenchmozilla.sourceforge.net/nvu/>
- QUIZZs
  - <http://quiz.alsacreations.com/>
  - <http://quiz.alsacreations.com/?quiz=1>
  - <http://quiz.alsacreations.com/?quiz=4>

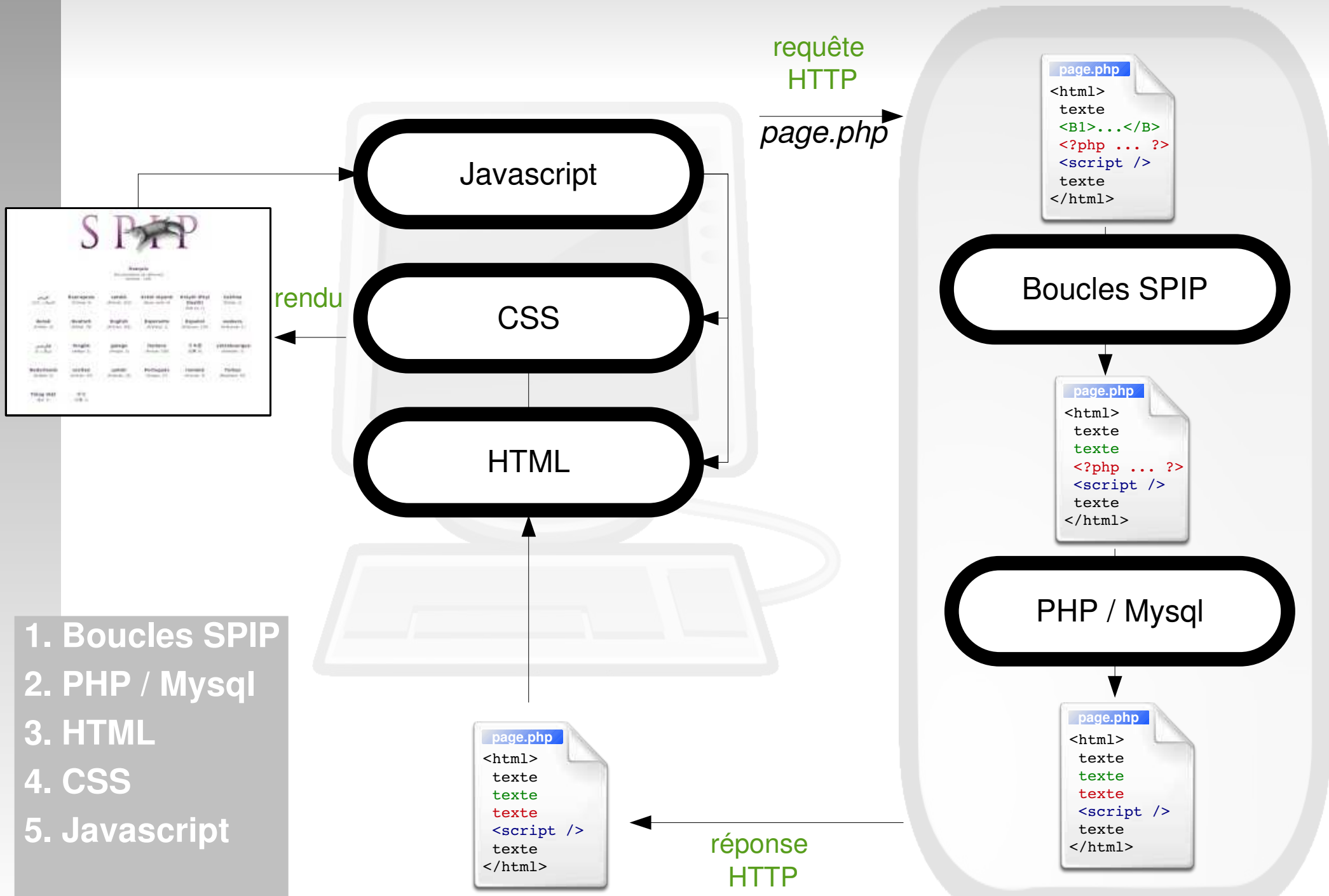
## *Partie 3*

# *Javascript*

*-notions-*



# Ordre d'exécution des instructions



# Javascript

- Langage de script
  - ne nécessite pas de compilateur
- Code dans la page HTML
  - directement interprété par le navigateur
- Orienté objet
  - méthodes : *history.back()*;
  - propriétés : *navigator.appName*;
- Utilisé pour rendre les pages Web interactives
  - permet de **modifier les pages** (contenu et mise en forme) sans les recharger
  - support actuel d'**Ajax** pour dialoguer avec le serveur sans recharger la page

# Java et Javascript

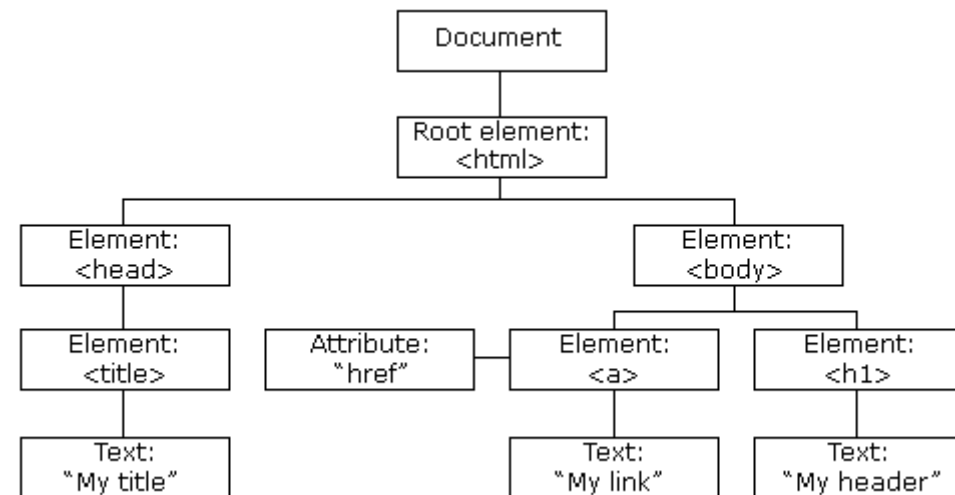
Javascript	Java
Langage interprété	Langage pseudo-compilé (chargement d'une machine virtuelle)
Code intégré au HTML	Code (applet) à part du document HTML, appelé à partir de la page
Langage peu typé	Langage fortement typé (déclaration du type de variable)
Liaisons dynamiques: les références des objets sont vérifiées au chargement	Liaisons statiques: les objets doivent exister au chargement (compilation)
Accessibilité du code	Confidentialité du code
Sûr: ne peut pas écrire sur le disque dur	Sûr: ne peut pas écrire sur le disque dur

# Contexte

- Historique
  - Développé par Netscape et Sun en 1995 (Live script)
  - Suivi par Microsoft (Jscript)
  - Standardisé par l'ECMA 262 / W3C -> Javascript
- Versions actuelles
  - 1.7 (Firefox 2.0), 1.5 (Opera et Safari) et Jscript 5.6 (IE7)
- Lié au DOM 2.0 (document model object)
  - Recommandation du W3C pour le Web visant à permettant à un langage de script de s'interfacer avec
    - le navigateur : navigation, évènements, historique
    - les documents interprétés : contenu XHTML, feuille de style CSS, graphiques SVG...

# Le Javascript est lié au DOM

- DOM = Document Model Object
  - arborescence d'objets représentant la structure du document interprété par le navigateur
- Recommandation du W3C pour le Web visant à permettre à tout langage de s'interfacer facilement avec
  - le contenu et la structure du document (XHTML)
  - le style (CSS)
  - indirectement le navigateur : navigation, évènements, historique
- Ex : le DOM HTML



# Généralités

- Les déclarations de fonctions sont dans l'en-tête
- Les scripts se place dans le corps de la page (body)  
les instructions se terminent par ;
- Appel à un script extérieur
- Commentaires
- Lancement d'une fonction au chargement

```
<head>  
  <script type="text/javascript">  
    function traite() {...}  
  </script>  
</head>
```

```
<body>  
  <script language="javascript">  
    document.write('bla');  
  </script>  
</body>
```

```
<script language="javascript"  
src="url/fichier.js" />
```

```
/* commentaire */
```

```
<body onLoad="traite()">
```



# Premier script

alert.html

- Créer le fichier et lancer le fichier suivant :

```
(...)  
<body>  
  <p>texte1</p>  
  <script language="javascript">  
    alert('ceci est une alerte');  
    document.write('<h1>Hello World!</h1>');  
  </script>  
  <p>texte3</p>  
</body>  
-> | (...)
```

séquentiellement

- Remarques sur le code source

# Les variables

- Objet repéré par son nom, contenant des données, qui pourront être modifiées lors de l'exécution du programme
- Types
  - Nombre entier ou à virgule : **3** ou **3.1416**
  - Chaîne de caractère : **"ma chaîne de caractère"**
  - Booléen : **true** ou **false**
  - Pas de valeur : **null**
- Déclaration
  - Globale : **texte = "bonjour";** (partout)
  - Locale : **var texte = "bonjour";** (à l'intérieur de la fonction)
- Appel
  - Exemple : **document.write(texte);**

# Les opérateurs

Opérateur	Dénomination	Effet	Exemple	Résultat (avec x valant 7)
+	opérateur d'addition	Ajoute deux valeurs	$x+3$	10
-	opérateur de soustraction	Soustrait deux valeurs	$x-3$	4
*	opérateur de multiplication	Multiplie deux valeurs	$x*3$	21
/	plus: opérateur de division	Divise deux valeurs	$x/3$	2.3333333
=	opérateur d'affectation	Affecte une valeur à une variable	$x=3$	Met la valeur 3 dans la variable x
%	opérateur modulo	Retourne le reste de la division entière	$x \% 2$	1
++	Incrémentatation	Augmente d'une unité la variable	$x++$	8
--	Décrémentatation	Diminue d'une unité la variable	$x--$	6
$+=$ , $-=$ , $*=$ , $/=$	ajoute l'opérande de gauche par l'opérande de droite		$x+=3$	10

- Créer le script suivant :

```
<script langage="javascript">
  var x=3.14159, u = Math.PI, s="3.14159";

  var a=x+u;
  alert(a);

  var b=x+s;
  alert(b);
</script>
```

- Que renvoient les alertes ?

# Comparaisons et tests

Opérateur	Dénomination	Effet	Exemple	Résultat (avec x valant 7)
==	opérateur d'égalité	Compare deux valeurs et vérifie leur égalité	x==3	Retourne True si X est égal à 3, sinon False
===	opérateur d'identité	Vérifie l'identité de valeur et de type de deux valeurs	a===b	Retourne True si a est égal à b et est de même type, sinon False
!=	opérateur de différence	Vérifie qu'une variable est différente d'une valeur	x!=3	Retourne 1 si X est différent de 3, sinon 0
!==	opérateur de non identité	de type de deux valeurs, c'est-à-dire si les deux valeurs n'ont pas la même valeur ou bien sont de types	a!==b	Retourne True si a est différent de b ou bien est de type différent, sinon False
<	opérateur d'infériorité stricte	Vérifie qu'une variable est strictement inférieure à une valeur	x<3	Retourne True si X est inférieur à 3, sinon False
<=	opérateur d'infériorité	Vérifie qu'une variable est inférieure ou égale à une valeur	x<=3	Retourne True si X est inférieur ou égale à 3, sinon False
>	opérateur de supériorité stricte	Vérifie qu'une variable est strictement supérieure à une valeur	x>3	Retourne True si X est supérieur à 3, sinon False
>=	opérateur de supériorité	Vérifie qu'une variable est supérieure ou égale à une valeur	x>=3	Retourne True si X est supérieur ou égal à 3, sinon False

```
if (x==2) document.write("X vaut 2");  
  
if (x==2)  
{ //liste d'instructions }  
else  
{ //autre série d'instructions }
```

- Créer un script
  - qui stocke dans une variable le nom du navigateur
    - *navigator.appName*
  - qui stocke dans une variable la version du navigateur
    - *navigator.appVersion*
  - si le navigateur est "Netscape" ou "Microsoft Internet Explorer" de version < 4
    - invite l'utilisateur à changer de navigateur

- Créer le script suivant :

```
<script langage="javascript">
  var t = Array();
  for(var i=0;i<=10;i++)
  {
    t[i] = i*i;
  }
  alert(t);
</script>
```

- Quel est le rôle de ce script ?

# Les fonctions

- Créer la fonction suivante :
  - Dans le <head>

```
<script language="javascript">
  function carre(nombre)
  {
    var carre = nombre*nombre;
    return carre;
  }
```

- </script>
- Dans le <body>

```
<script language="javascript">
  for(var i=0;i<=10;i++)
  {
    document.write(carre(i)+"<br />");
  }
</script>
```



# Les évènements

- `<balise onEvenement="code_javascript;" >`  
`<balise onEvenement="fonction()" >`

```
<a onClick="alert('vous avez cliqué');" href="#" >Test</a>
```

## ▪ Liste des évènements

- `onAbort` (en cas d'interruption)
- `onBlur` (en quittant)
- `onChange` (après modification réussie)
- **`onClick`** (en cliquant)
- **`onDbIcIck`** (en double-cliquant)
- `onError` (en cas d'erreur)
- `onFocus` (en activant)
- `onKeyDown` (en appuyant sur une touche)
- `onKeyPress` (en maintenant une touche appuyée)
- `onKeyUp` (en relâchant la touche)
- **`onLoad`** (en chargeant un fichier)
- `onMouseDown` (touche de souris appuyée)
- **`onMouseMove`** (en bougeant la souris)
- **`onMouseout`** (en quittant l'élément avec la souris)
- **`onMouseover`** (en passant sur l'élément avec la souris)
- `onMouseUp` (en relâchant la touche de souris)
- `onReset` (en initialisant le formulaire)
- `onSelect` (en sélectionnant du texte)
- `onSubmit` (en envoyant le formulaire)
- `onUnload` (en quittant le fichier)

# La notion d'objets

- Exemple : **jardin.arbre.nid.hauteur=20;**

-> De la même façon la page web est divisée en objets afin de les manipuler : Hiérarchie

- Fenêtre > Document HTML > Elements HTML
- Ex : **(window).document.form.radio[0]**

- Un objet possède des :

## - propriétés

- nom\_de\_l'objet.nom\_de\_la\_propriété
- Ex : **document.form.radio[1].checked=false;**

## - méthodes

- nom\_de\_l'objet.methode();
- Ex : **document.form.reset();**

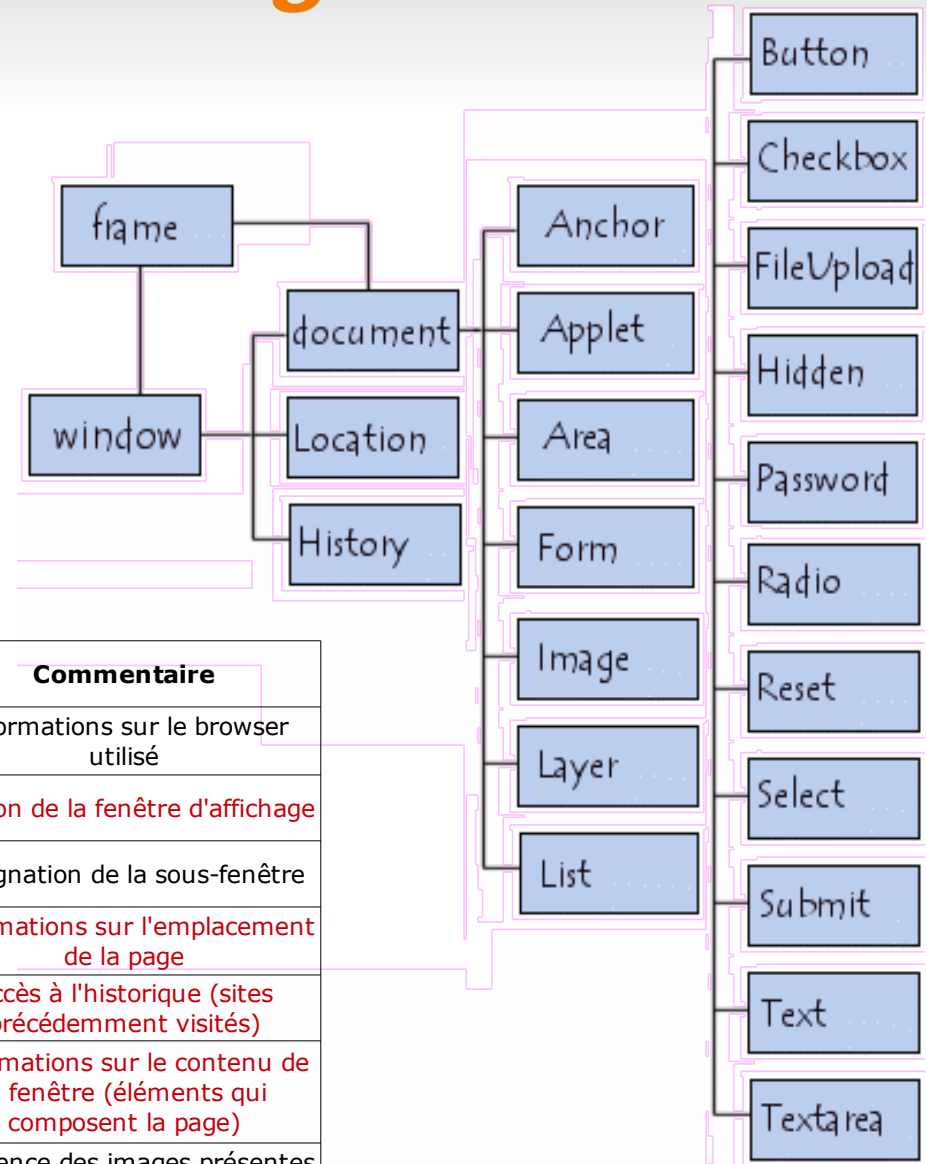
```
jardin
  arbre
    branche
      feuille
        nid
          largeur: 20
          couleur: jaune
          hauteur: 4
      tronc
    racine
```



```
<form action="">
  <input type="radio">
  Français
  <input type="radio">
  Anglais
  <input type="button"
  value="Déterminer"/>
  <input type="text"
  value="" name="Text1"/>
</form>
```

# Les objets du navigateur

- On utilise surtout document, location et history en direct



Niveau1	Niveau2	Niveau3	Commentaire	
navigator			Informations sur le browser utilisé	
window			Gestion de la fenêtre d'affichage	
	parent, frames, self, top		Désignation de la sous-fenêtre	
	location		Informations sur l'emplacement de la page	
	history		Accès à l'historique (sites précédemment visités)	
	document			Informations sur le contenu de la fenêtre (éléments qui composent la page)
		images		Référence des images présentes dans la page
		forms		Référence des formulaires présents dans la page
		links		Référence des liens présents dans la page
	anchors		Référence des ancrages présents dans la page	

- Propriétés
  - **navigator.appName** : Netscape
  - **navigator.appCodeName** : Mozilla
  - **navigator.appVersion** : 5.0 (X11; fr)
  - **navigator.language** : fr
  - **navigator.cookieEnabled** : yes
- Sous-objets
  - **navigator.mimeTypes** : gestion des types de fichiers
  - **navigator.plugins** : plugins installés

- Propriétés
  - **window.parent** : fenêtre mère
  - **window.name** : nom de la fenêtre
- Méthodes
  - **window.open()**
  - **window.close()**
  - **window.resizeTo()**
- Sous-objets
  - **window.document**
  - **window.history**
  - **window.location**

# *L'objet history*

location\*.html

- Méthodes
  - **history.back()**
  - **history.forward()**
  - **history.go(-3)**

## *L'objet location*

- Propriétés
  - **location.href**
- Méthodes
  - **location.reload()**
  - **location.replace(url)**
  - **history.go(-3)**

# *L'objet document*

- Contient des sous-objets : les éléments HTML
- Pour accéder à ces éléments :
  - `document.getElementById("nom")`
  - `document.getElementsByTagName("img")`
- Pour accéder aux propriétés de ces éléments :
  - `document.élément.getAttribute(name);`
- Pour modifier les propriétés de ces éléments :
  - `document.élément.setAttribute(name,value);`
  - `document.élément.style.property="value";`
- Pour accéder et modifier le contenu de ces éléments :
  - `document.élément.innerHTML (= "value");`

# Exemple

document\_object\_setattribute\_align.html

- modifier un attribut interactivement : *align*
  - fonctions

```
function aligner(comment)
{
    document.getElementById("indecis").setAttribute('align',comment);
}

function affiche_alignement()
{
    alert(document.getElementById("indecis").getAttribute('align'));
}
▪ corps
```

```
<h1 id="indecis" onclick="affiche_alignement()">
    Où est donc ma place?</h1>
<a href="javascript:aligner('left')">à gauche?</a><br>
<a href="javascript:aligner('center')">au centre?</a><br>
<a href="javascript:aligner('right')">à droite?</a><br>
```



- Faire la même chose en utilisant la propriété *.style.textAlign*
- Créer un nouveau document
  - permettant de modifier interactivement la taille d'une image
- Créer un nouveau document
  - permettant de cacher / afficher une zone de texte lorsque l'on clique sur un texte
- Créer un nouveau document
  - qui modifie le contenu d'un titre lorsque l'on clique sur texte

# *d'autres exemples*

- [http://www.w3schools.com/html/dom/dom\\_examples.asp](http://www.w3schools.com/html/dom/dom_examples.asp)
- <http://fr.selfhtml.org/javascript/exemples/index.htm>
  - Compter les visites (cookies)
  - Calculatrice
  - Diaporama
  - Vérifier formulaire
- Des scripts, des scripts, des scripts...
  - <http://www.editeurjavascript.com/>
  - <http://www.allhtml.com/>
  - <http://www.dynamicdrive.com/>
  - <http://www.javascriptfr.com/>

# Crédits

- Patrick VINCENT
  - [pvincent@erasme.org](mailto:pvincent@erasme.org)
  - [pvincent4@gmail.com](mailto:pvincent4@gmail.com)
- Contenu
  - non garanti exempt d'erreurs ;)
  - sous licence **FDL**
- Merci !

