

Objectifs de l'enseignement de la matière

Développement d'Applications Web

(DAW, Licence 2, Semestre 4)

- *Présenter les systèmes d'information dans le contexte Internet.*
- *Le module initie à la programmation Web via les langages HTML, CSS, JavaScript et PHP.*
- *En plus, il initie au développement des services web.*
- *Une étude pratique (les TPs), renforce les concepts acquis.*

Recommandations :

- *Insister sur les TPs durant le module.*

Connaissances préalables recommandées :

- *Notions de base d'internet, initiation en HTML*

Le contenu de la matière se base sur le Programme établi par le Ministère de l'enseignement et de la Recherche scientifique de 2015.

Mode d'évaluation :

- *Contrôle Continu et Examen écrit*

Références

- *Concevoir des applications Web avec UML, Jim Conallen- Collection Eyrolles.*
- *J2EE, Nicolas Duminil - Dunod.*
- *EJB 2.0 Mise en œuvre. Christophe Calandreau, Alain Fauré, Nader Soukouti-*

2ème année Licence Informatique
Développement d'Applications Web
Programmation Spécifique :
Programmation Web

HTML / CSS

QUID ? « Qui » signifie « Quoi »

- APACHE : Serveur WEB
- HTML : Langage hypertexte de description de documents
- PHP : Langage de programmation interprété coté serveur
- MYSQL : Gestionnaire de base de données (non étudié ici)
- Javascript : Langage de programmation interprété coté client.

Pourquoi ces choix :

- Le prix (gratuit)
- La qualité (utilisés par les entreprises
ex : Apache = 2/3 des serveurs WEB)

Points du Module

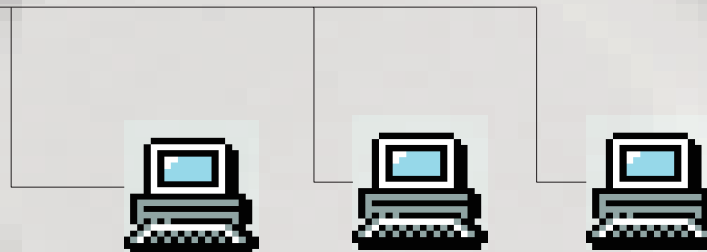
- **HTML : Hyper Text Markup Language**
 - CSS (feuilles de style Cascading Style Sheet)
 - Javascript.
 - XML EXtensible Markup Language (si possible)
- **PHP : Hypertext PreProcessor**
 - PHPde base.
 - Interaction avec des SGBDD
 - Création d'images à la volée / creation de pdf....
- **Services Web : notions de base**

Serveur WEB

- Un serveur WEB est une application capable de répondre aux requêtes de clients.
Il est connecté au clients via un réseau (souvent).
- Il communique avec le client à l'aide du protocole HTTP(**H**ypert**T**ext **T**ransfert **P**rotocol).
- Il fournit aux clients une page HTML.



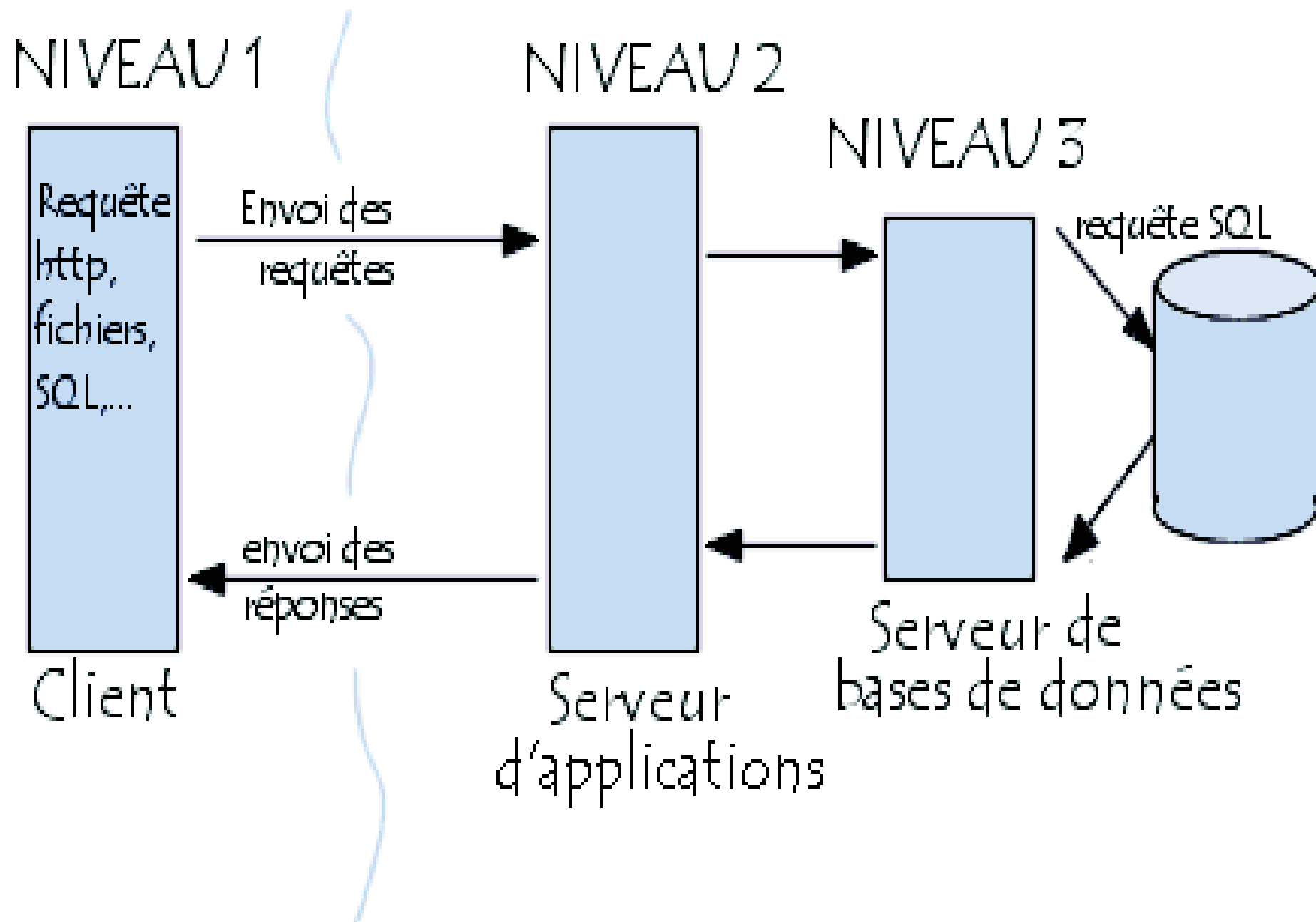
Serveur

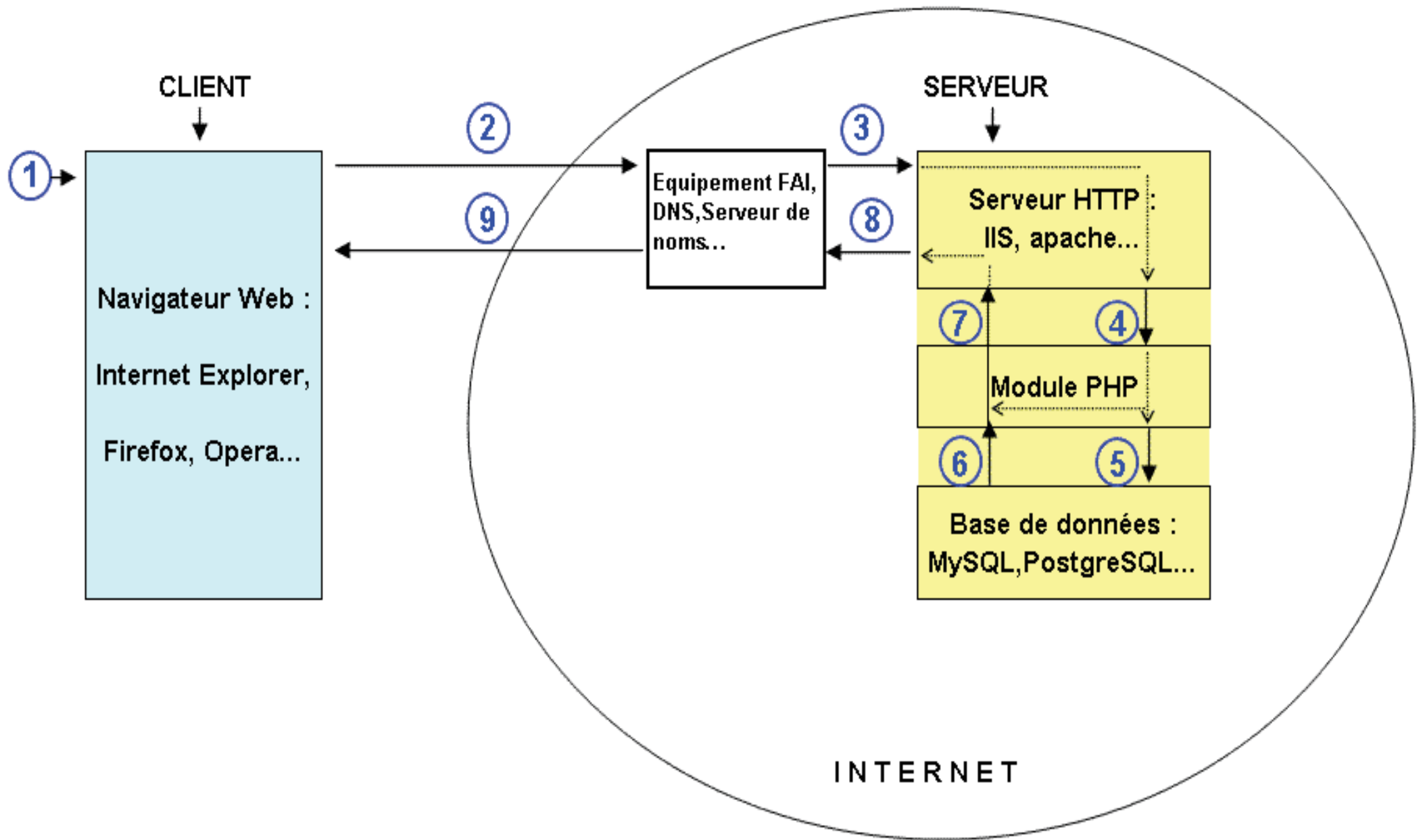


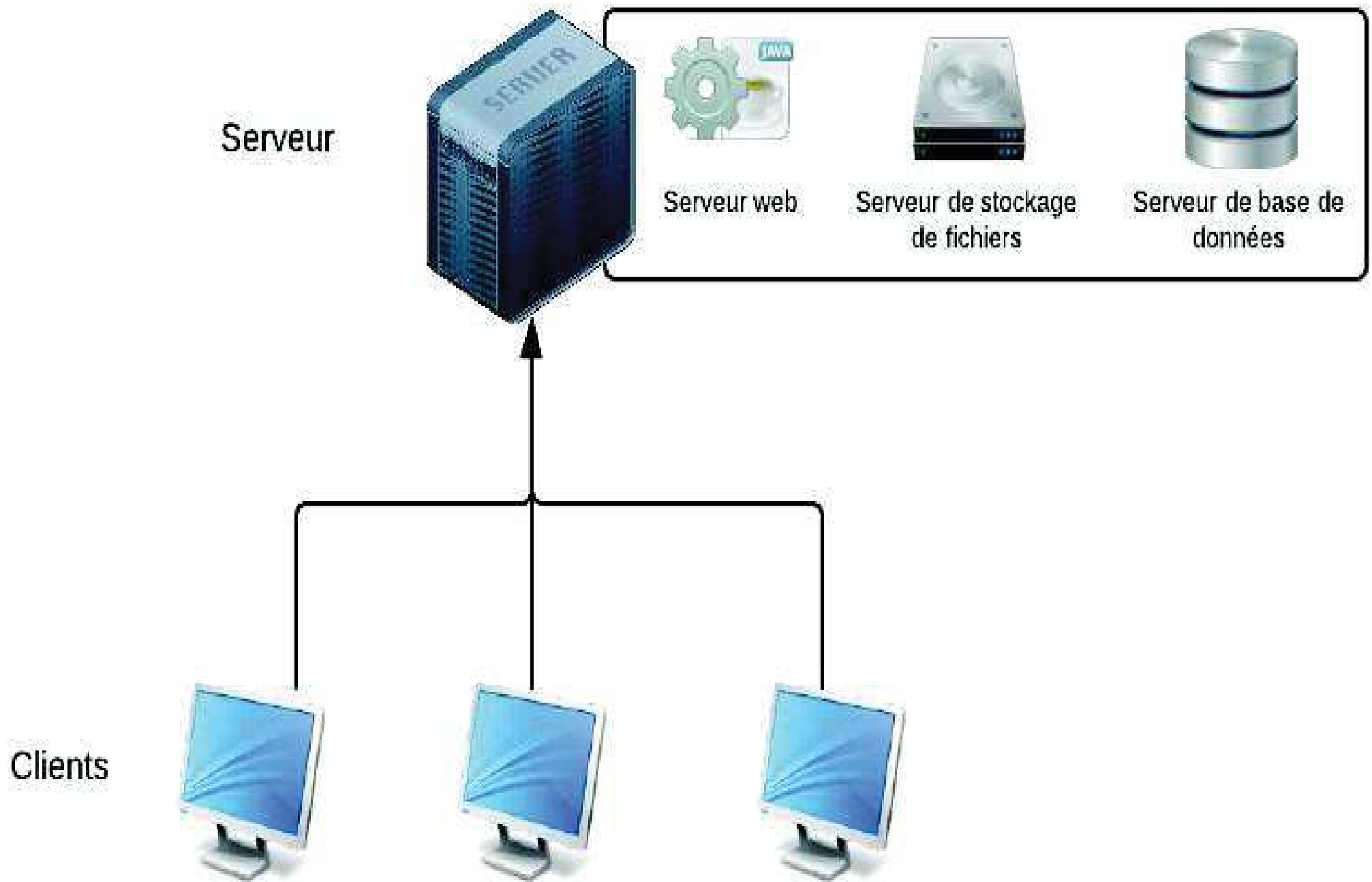
Clients

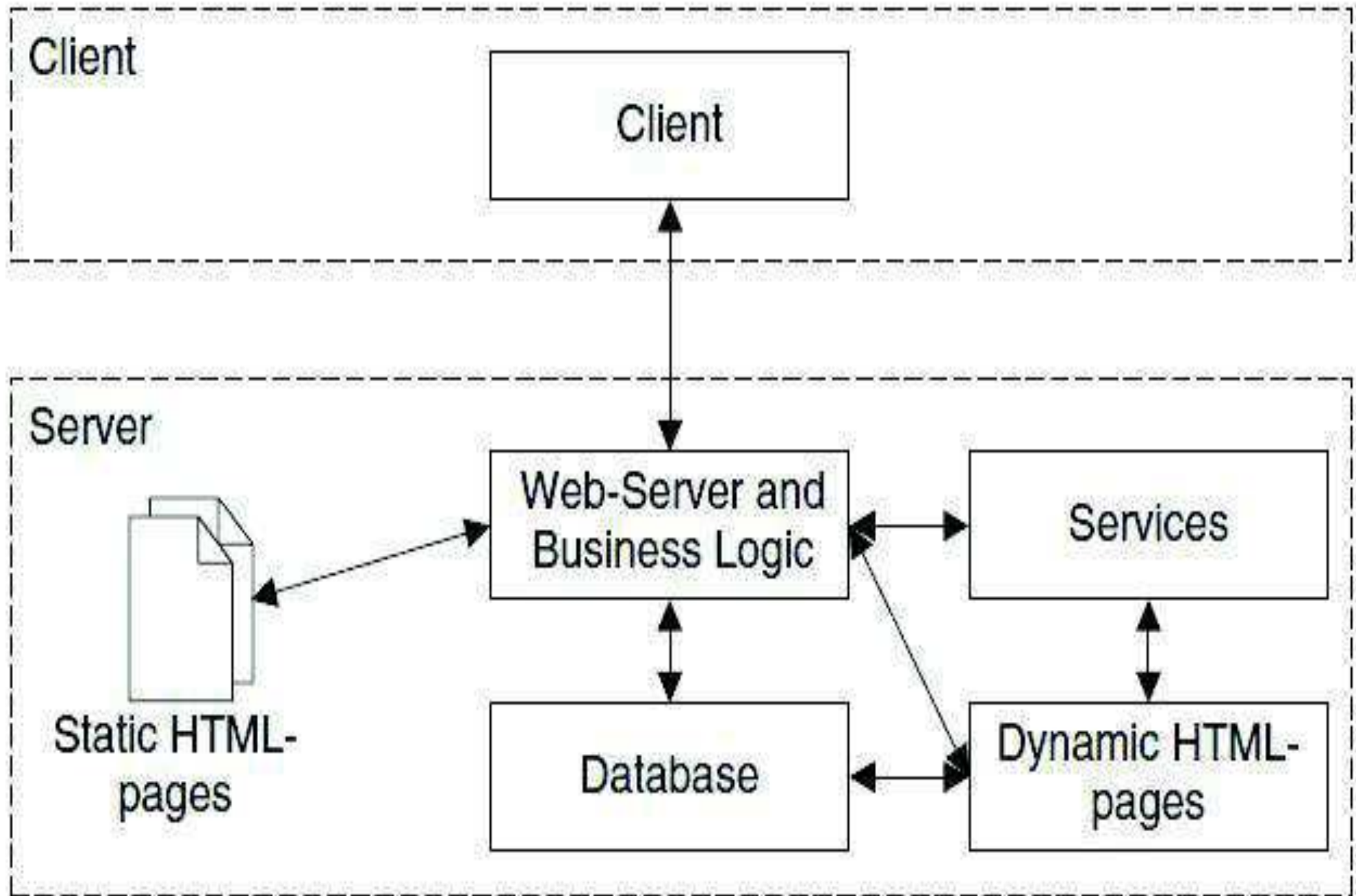


This is what's displayed through your web browser.









Types de Sites

On distingue trois types de sites :

- Site **Statique** : les requêtes portent sur des fichiers **html**.
- Site **Dynamique** : les requêtes déclenche sur le serveur l'exécution d'un script (**php**). Le script produit une page HTML.
- Site "Sapin de Noël" : les requêtes portent sur des fichiers html contenant des scripts coté client.

On peut avoir un site dynamique et sapin de noel...

DEFINITIONS

Le WWW(World Wide Web)?

► WEB ?

Du réseau Internet au Web

Si on a un réseau et des documents :

► **Web** = réseau + documents hypertextes

- réseau = machines connectées

- documents hypertextes = documents connectés

► **DONC :**

Web = documents connectés sur machines connectées

► Réseau international Internet

→ **World Wide**

► **Donc le WEB est une application/utilisation particulière des possibilités offertes par Internet**

Hypertexte ?

- ▶ Un **système hypertexte** est un système contenant des nœuds liés entre eux par des hyperliens permettant de passer automatiquement d'un nœud à un autre.
- ▶ Un **document hypertexte** est donc un document qui contient des hyperliens et des nœuds.
- ▶ Un **nœud** est une « unité minimale d'information », notion assez floue qui signifie simplement que l'information d'un nœud sera toujours présentée *entière*.
- ▶ *Les **liens** entre les parties du texte sont gérés par ordinateur et permettent d'accéder à l'information d'une manière associative ou, tout au moins, d'une façon de naviguer personnalisée, de manière non linéaire, au gré de l'utilisateur.*
- ▶ *La notion d'hypertexte a trouvé sa plus grande réalisation dans le World Wide Web.*
- ▶ Lorsque les nœuds ne sont pas uniquement textuels, mais aussi audiovisuels, on peut parler de système et de documents hypermédias.

HTML

Langage hypertexte de description de documents

- Hypertexte : permet une navigation non-linéaire
- Interprété par un client http :
 - adaptation au client (taille, polices...)
- Fournit une description simple d'un document.
- Documents composés de texte, images.

➤ **LANGAGE STATIQUE: AFFICHER DES PAGES WEB**

>>>> Nous utiliserons : HTML5

HTML

Architecture d'un document

- Format d'un document html : texte simple.
- Ce qui est dans le fichier est affiché, sauf instructions html.
- Instructions sous formes de balises (ou « tags ») (< >).
- 2 grandes parties dans un document :
 - Doctype : indique quel version du langage est utilisée
 - Entête (head) : informations générales sur la page
 - Corps (body) : Ce qui sera affiché sur le client.

HTML

Fonctionnement des balises

- Une balise est délimitée par des crochets < >
- Toutes doivent être ouvertes et fermées < > , </ >
- Elles peuvent avoir des attributs (options).

<p>

Du texte ici

Puis encore du texte.

</p>

HTML

Exemple

Première page HTML typique :

```
<!doctype .....>
<html>
  <head>
    <title> Ma premi&egrave;re page</title>
  </head>
  <body>
    <p>Le texte dans ma premi&eagrave;re page. </p>
  </body>
</html>
```

Les caractères particuliers pour

Car	HTML	En français
â	â	a minuscule accent circonflexe
à	à	a minuscule accent grave
é	é	e minuscule accent aigu
ê	ê	e minuscule accent circonflexe
è	è	e minuscule accent grave
ë	ë	e minuscule tréma
î	î	i minuscule accent circonflexe
ï	ï	i minuscule tréma
ô	ô	o minuscule accent circonflexe
œ	œ	o e minuscule liés
û	û	u minuscule accent circonflexe
ù	ù	u minuscule accent grave
ü	ü	u minuscule tréma
ç	ç	c cedille minuscule
<	<	inférieur à
>	>	supérieur à

HTML

Balises de structuration du document

- `<doctype>` : renseigne sur la version du langage utilisé
- `<html></html>` : il s'agit du langage HTML
- `<head></head>` : entête.
- `<body></body>` : corps du document.

HTML

Balises de mise en forme des pages

Pour décrire le découpage des pages en zones :

- `<div> </div>` : une unité qui fait sens pour le site (ex : une entête, un menu,)

Pour ce découpage, il y eut (obsolète) :

- `<frameset> <frame>`
- `<table> <tr> <td>`

HTML

Balises de mise en forme de texte :

- `<p></p>` : un paragraphe.
- ` ` : accentué (gras visuellement)
- ` ` : mis en évidence (italique visuellement)
- `` : exposant. (superscript).
- `` : indice (subscript).

HTML

Balises de mise en forme de texte 2

- `<xmp></xmp>` : non interprété (affiché tel quel).
- `<h1></h1>` : premier niveau de titre. (aussi `<h2>...`
- `<!-- test -->` : commentaire.
- `` : liste numérotée.
- `` : élément de liste.
- `` : liste à puces.

HTML

Balises de mise en forme de texte :
les forbidden : Ne pas utiliser !

- `
` : passage à la ligne.
- `<hr>` : ligne horizontale.
- `<i></i>` : italique.
- `` : gras (bold).

HTML

Balises utiles

- `` : insertion d'images.
- `<a>` : ancre. Notamment pour les liens.
- `<table></table>` : tableau.
 - `<tr></tr>` : ligne d'un tableau.
 - `<td></td>` : case d'un tableau.
- Attribut : `colspan/rowspan` : multi colonne/ligne.

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

```
<html>
<head>
<title>titre</title>
</head>
```

```
<body>
<p>Ceci est un paragraphe</p>
<hr/>
```

```
<h1>bonjour!</h1>
<h3>moyenjour.</h3>
<h6>mauvaisjour.</h6>
<p>je vais aller
```

```
<a href="http://www.via.ecp.fr">
rendre visite à via</a>. via c'est bien
.</p>
```

```
<ul>
<li>un élément</li>
<li>un autre élément</li>
</ul>
```

```
<ol>
<li>premier élément</li>
<li>deuxième élément</li>
</ol>
```

```
</body>
```

```
</html>
```



les tables :

1ère ligne, 1ère colonne	1ère ligne, 2ème colonne
2ème ligne, 1ère colonne	2ème ligne, 2ème colonne

```
<table>
```

```
  <tr>
```

```
    <td>1ère ligne, 1ère colonne</td>
```

```
    <td>1ère ligne, 2ème colonne</td>
```

```
  </tr>
```

```
  <tr>
```

```
    <td>2ème ligne, 1ère colonne</td>
```

```
    <td>2ème ligne, 2ème colonne</td>
```

```
  </tr>
```

```
</table>
```

```
<h1>une table un peu plus  
compliquée...</h1>
```

```
<table border="1">  
  <tr>  
    <td colspan="3">  
      1ère ligne, 1ère colonne  
    </td>  
    <td>1ère ligne, 4ème colonne</td>  
  </tr>  
  <tr>  
    <td>2ème ligne, 1ère colonne</td>  
    <td>2ème ligne, 2ème colonne</td>  
    <td>2ème ligne, 3ème colonne</td>  
    <td rowspan="2">  
      2ème ligne, 4ème colonne  
    </td>  
  </tr>  
  <tr>  
    <td colspan="2" rowspan="2">  
      3ème ligne, 1ère col  
    </td>  
    <td>3ème ligne 3ème colonne</td>  
  </tr>  
  <tr>  
    <td>4ème ligne, 3ème colonne</td>  
    <td>4ème ligne, 4ème colonne</td>  
  </tr>  
</table>
```



HTML

Balises blocs et inline

Deux types de balises :

- blocs (des parties entières de texte)

ex : <p> <div> <h1>

- en ligne (inline) : enrichit une portion de texte

ex :

- Les blocs se suivent (par défaut) l'un en dessous de l'autre.
- Les inline se suivent (par défaut) l'un a coté de l'autre.
- Une inline ne peut contenir que des inline (et du texte)
- une bloc peut contenir des blocs, des inline, du texte.

HTML

Les feuilles de styles

On les appelle aussi **CSS** pour « **Cascading Style Sheets** ».

- Objectif : Définir le style d'un document html indépendamment de son contenu.
- Intérêt : Votre site est beau, bien rempli...vous voulez changer l'image de fond de toutes les pages...
- Mise en oeuvre : 3 méthodes dont par exemple: les styles sont définis dans un fichier, référencé dans le document html (voir suite).

*On utilisera la version 3
CSS3*

La feuille de **style** est un **modèle** permettant une mise en forme **automatisée** des différentes parties d'une page web. Elle s'applique sur les balises telles que les titres, les paragraphes, les divisions, les liens, les listes.

Il existe 3 méthodes:

- 1/ Utiliser la balise <style> à l'intérieur du programme html, elle s'applique sur les <p > , les <div>, les titres <h1> et des liens.
- 2/Utiliser un fichier externe : lien externe dans lequel on va enregistrer tous les styles que l'on veut pour les paragraphes, les titres, ...etc, et qu'on appellera P.css.
- 3/ Solution locale : on utilisera à l'intérieur du <Body> un style comme on veut, au niveau des titres (header), des divisions et des paragraphes, ...etc.

1/ Balise style à l'intérieur du programme html.	2/ Les styles sont dans un Fichier externe P.css, utiliser un Link.	3/ Solution locale : les styles dans définis le body.
<pre> <html> <head> <style> </head> <body> <p> <div> <h2> <a href (liens)> </body> </html> </pre>	<pre> {.....} P.css P.html : <html> <head> <Link..... </head> <body> </body> </html> </pre>	<pre> <html> <head> </head> <body> <h2 style= <div style = <p style = </body> </html> </pre>

1/ Utiliser la balise <style> à l'intérieur du programme html, elle s'applique sur les <p> , les <div>, les titres <h1> et des liens:

```
<html>  
  <head>  
    <style>  
  </head>  
  
  <body>  
    <p>  
    <div>  
    <h2>  
    <a href (liens)>  
  
  </body>  
</html>
```

2/Utiliser un fichier externe : lien externe dans lequel on va enregistrer tous les styles que l'on veut pour les paragraphes, les titres, ...etc, et qu'on appellera P.css.

On le réutilisera à l'intérieur du programme html grâce à un **<Link ...** :

{.....} P.css

<HTML>

<HEAD>

<Link.....

</HEAD>

P.html

<BODY>

</BODY>

</HTML>

3/ Solution locale : on utilisera à l'intérieur du <Body> un style comme on veut, au niveau des titres, des divisions et des paragraphes, ...etc:

```
<html>
```

```
<head>
```

```
</head>
```

```
  <body>
```

```
    <h2 style=
```

```
    <div style =
```

```
    <p style =
```

```
      </body>
```

```
</html>
```

Quand aux **attributs**, ils se trouvent sous cette forme:

```
{ font-size: 34 pt ; (hauteur d'un caractère de 14 à 34 pt  
possibles)
```

```
font-family:arial; (famille de caractères)
```

```
text-align: center;
```

```
color: blue;
```

```
background: yellow;
```

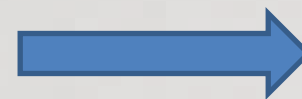
```
font-style: oblique;}
```



CSS

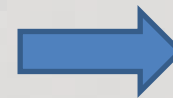
Comme ce programme:

```
<html>
<head>
<style>
{ font-size: 34 pt ;
  font-family:arial;
  text-align: center;
  color: blue;
  background: yellow;
  font-style: oblique;}
</style>
</head>
<body>
  <p class="titre">css4</p>
</body> </html>
```



Exemple 1:

```
<html>
<head>
  <style>
    . titre { color: 00ff00;
              font-size: 20 pt ;
              background: gold;
              text-align: center;
            }
  </style>
</head>
<body>
  <h1 class="titre">css1 - hello</h1>
</body>
</html>
```



CSS1-Hello

Exemple 2: liens externes

css2.css

```
.titre {color:00ff00;....background:gold;}
```

css2.html

```
<html>
```

```
<head>
```

```
<link ref = stylesheet  
      href="css2.css"  
      type="text/css">
```

```
</head>
```

```
<body>
```

```
<h1 class="titre">CSS1-Hello</h1>
```

```
</body>
```

```
</html>
```



CSS1-Hello

Nous obtenons comme Exemple1!

Exemple3: Liens locaux qui se trouvent au niveau du body: On affiche 2 titres CSS3-red, CSS3-green:

css3.html

```
<html>
```

```
<head> ..... </head>
```

```
<body>
```

```
  <h2 style="color-red">CSS3-red </h2>
```

```
  <h2 style="color-green">CSS3-green</h2>
```

```
</body>
```

```
</html>
```



CSS3-red

CSS3-green

HTML

Fonctionnement des feuilles de styles

Le document html indique l'utilisation de feuilles de styles avec la ligne suivante dans l'entête :

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

La feuille de style indique ensuite des règles pour le formatage.



```
/* Definitions des titres niveau 2 */
```

```
h2 {color:#0000ca; text-align:left; font-size:large; font-style:italic}
```

HTML

Limiter les problèmes de polices grâce aux css

Voici comment préciser trois polices possibles dans l'ordre de préférence, en fonction de leur disponibilité.

```
h3
{
font-family: verdana, arial, sans-serif; font-size: 14pt; color: red;
background-color: #ffffff;
}
```

HTML

surcharger des tags grâce aux css

Ici, on utilise la balise `<p>` pour laquelle on définit une nouvelle classe : petit

```
p.petit
{ font-family: times, serif; font-size: 10pt; color:black;
font-style:italic; background-color : #ffffff;
}
```

```
<p class="petit"> ceci va etre ecrit en petit </p>
```

HTML

utiliser au mieux l'hérédité dans les css

Vous pouvez créer des sous-classes applicables à toute balise.

```
.colored{ color:red; }  
.marge { margin-left:30px}
```

Ceci est bien pratique pour faire des choses comme :

```
<h1 class="colored"> Un gros titre rouge </h1>  
<b class="colored"> du texte gras et rouge </b>  
<p class="marge"> un paragraphe avec une marge </p>
```

HTML

utiliser au mieux l'hérédité dans les css 2

La balise `` est utilisée en CSS pour modifier l'aspect d'une zone particulière de données.

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

Ceci est bien pratique pour faire des choses comme :

```
<span style= "font-size: 16pt; color: fuchsia; background-color: #FFFFFF; " >un truc très occasionnel</span>
```

```
<span class="red">juste mettre en rouge.</span>
```



Ne pas en abuser...

HTML

utiliser au mieux l'hérédité dans les css 3

La balise <div> est utilisée en CSS pour modifier l'aspect de paragraphes

```
.marge{ margin-left:30px }
```

Ceci est bien pratique pour faire des choses comme :

```
<div class= "marge" >tout un paragraphe avec sa marge. Mais c'est  
comme <p> alors ? Ben non, on verra ca plus loin...</div>
```

HTML

Les liens dans les css

Les pseudos classes liées aux liens (balise <a>) : link : lien normal.

visited : lien deja visité.

hover : on passe dessus avec la souris. active : on a cliqué dessus.

```
a:link { color:blue; text-decoration:underline; background-color: #ffffff;
        } a:visited { color:purple; text-decoration:underline ;
        background-color:
        #ffffff; }
a:hover { color:red; text-decoration:underline; background-color:
        #ffffff; }
a:active { color: red; font-weight: bold; background-color: #ffffff; }
```


HTML

Positionnement dans les css

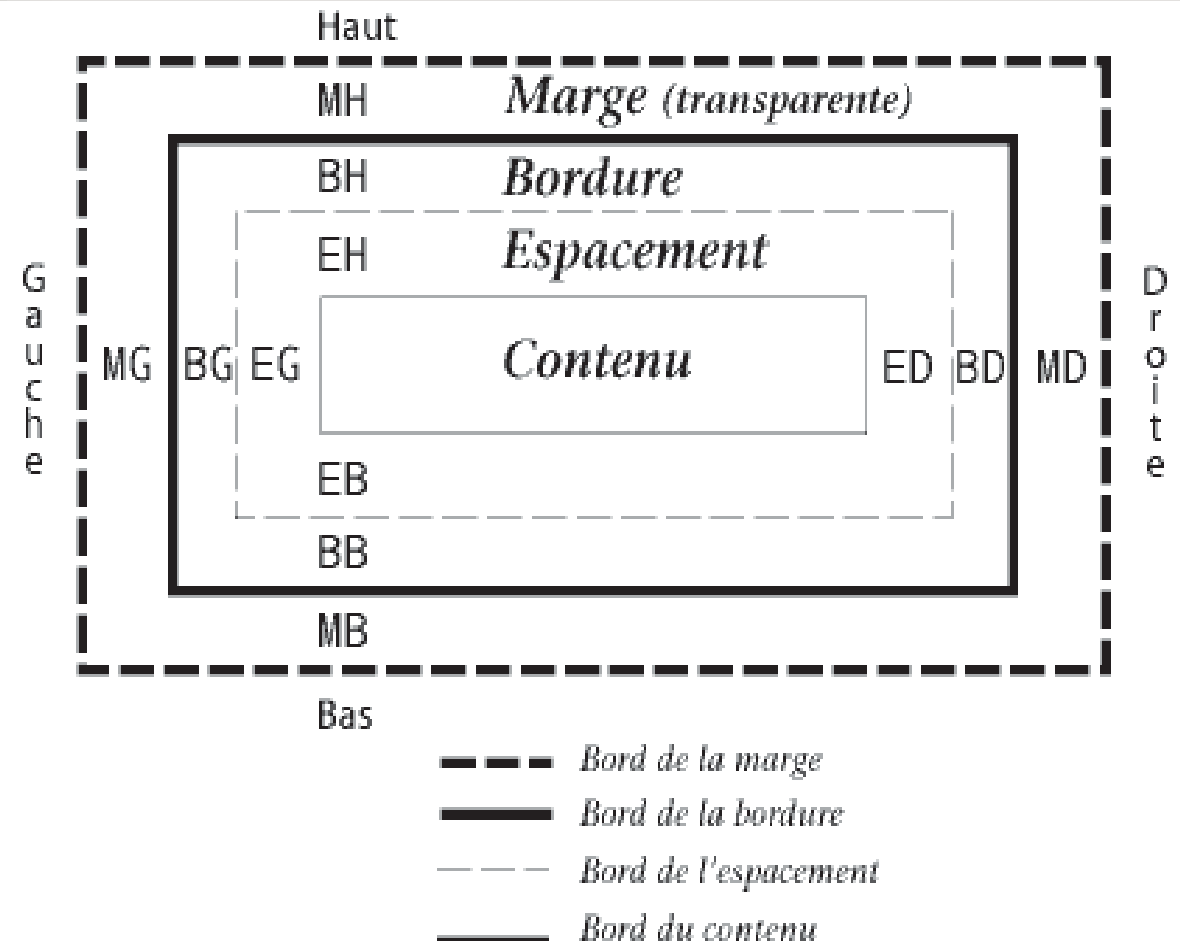
Le modèle de boîte : vaut pour toutes les balises bloc :

En anglais (et en css) :

margin : marge externe.

border : bordure

padding : marge interne



Positionnement et CSS

Positionnement dans le flux

Le positionnement normal : dans le flux.

Les éléments les uns en dessous des autres, puis par rapport :

- aux marges internes du conteneur.
- aux marges externes des éléments.

Avec ceci, on peut "déplacer" un élément dans son conteneur, et le contenu d'un élément au sein de cet élément.

Mais on ne peut pas placer deux blocs l'un à côté de l'autre !

Positionnement et CSS

Positionnement

fichier html :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
    <title> des tests </title>
    <link rel="stylesheet" href="flux.css" type="text/css" />
  </head>
  <body>
    <div id="entete">      entete      </div>
    <div id="navigation">  barre de navigation  </div>
    <div id="milieu">      Ici le milieu
      <div id="menu">      le menu      </div>
      <div id="corps">    le corps du texte</div>
    </div>
    <div id="footer">      un pied de page  </div>
  </body>
</html>
```

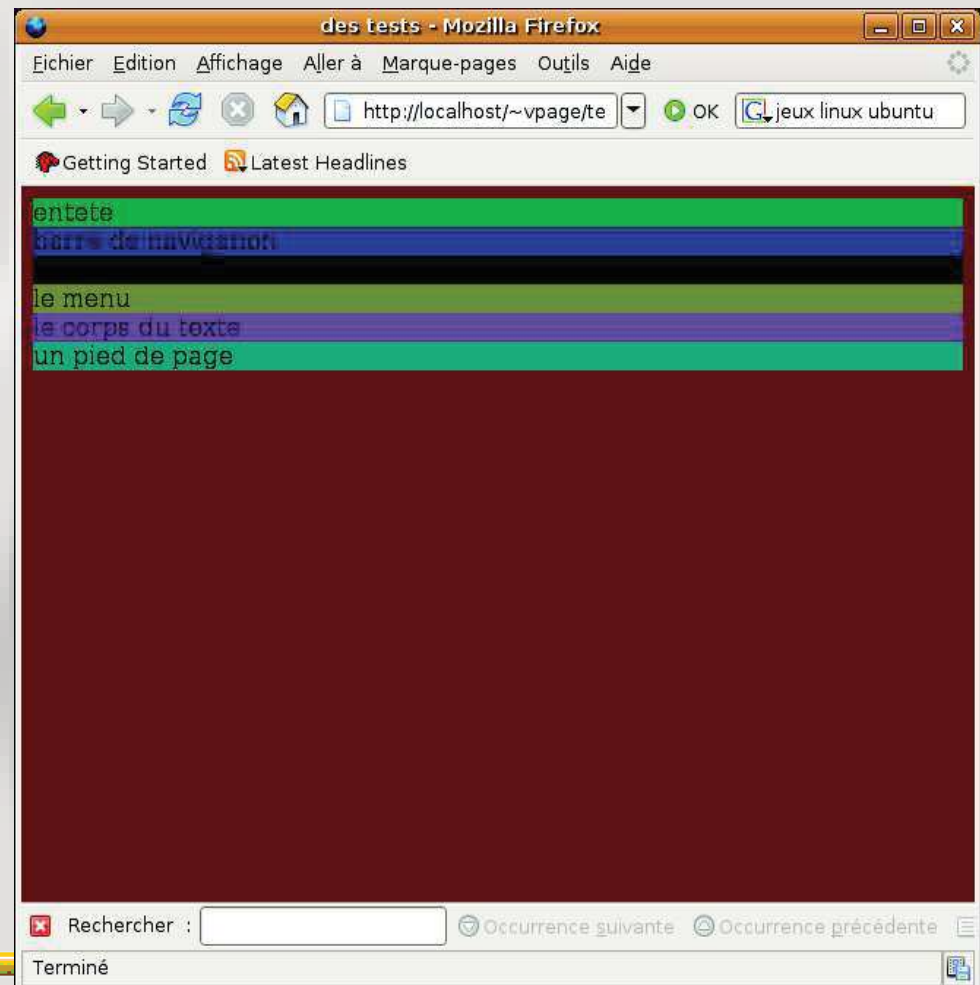
Positionnement et CSS

Positionnement

Et une css quasi vide :

```
body{  
    color: #000;  
    background-color: #800;  
}  
  
#entete{    background-color : #080; }  
  
#navigation{ background-color : #008; }  
  
#milieu{    background-color : #444; }  
  
#menu{      background-color : #880; }  
  
#corps{     background-color : #808; }  
  
#footer{    background-color : #088; }
```

Pour ce résultat :



Positionnement et CSS

Positionnement : marges et centrage

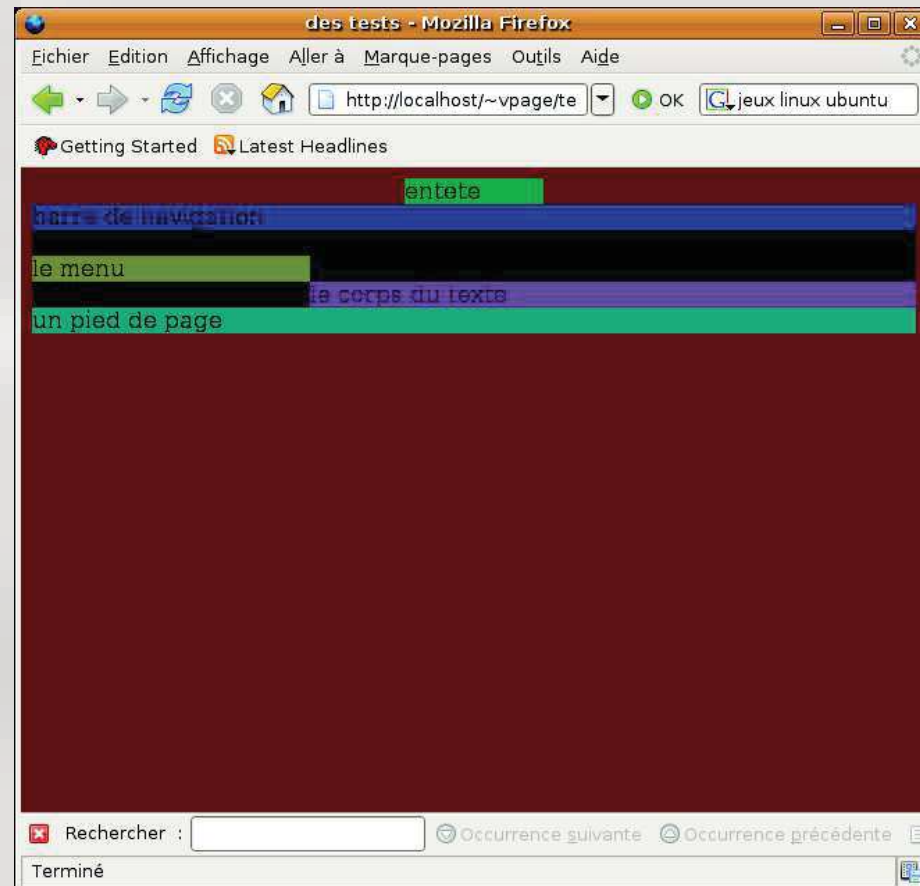
on rajoute les lignes suivantes

```
#entete{      width : 100px;
              margin : auto;
}

#menu{       width : 200px;}

#corps{      margin-left:200px;}
```

Pour ce résultat :



Positionnement et CSS

Positionnement flottant

Déplacer un élément à gauche ou à droite dans son parent.

Le reste du contenu du parent se positionne autour de l'élément déplacé.

Attention : Comme le float sort du flux courant, il n'est pas compté dans le calcul de la hauteur du conteneur, si celle-ci n'est pas spécifiée. Il peut donc "dépasser" en hauteur.

Récapitulatif :

** L'élément est d'abord placé normalement dans le flux. Donc si un autre élément bloc est placé avant lui, l'élément flottant se positionnera en dessous; et si un élément bloc le suit, ce dernier sera placé dessous. Bref, le comportement classique du positionnement en flux*

** Ensuite l'élément qui a reçu la propriété float est "poussé" à gauche ou à droite de son conteneur.*

** Les éléments qui le suivent au sein du conteneur se placent alors autour de lui.*

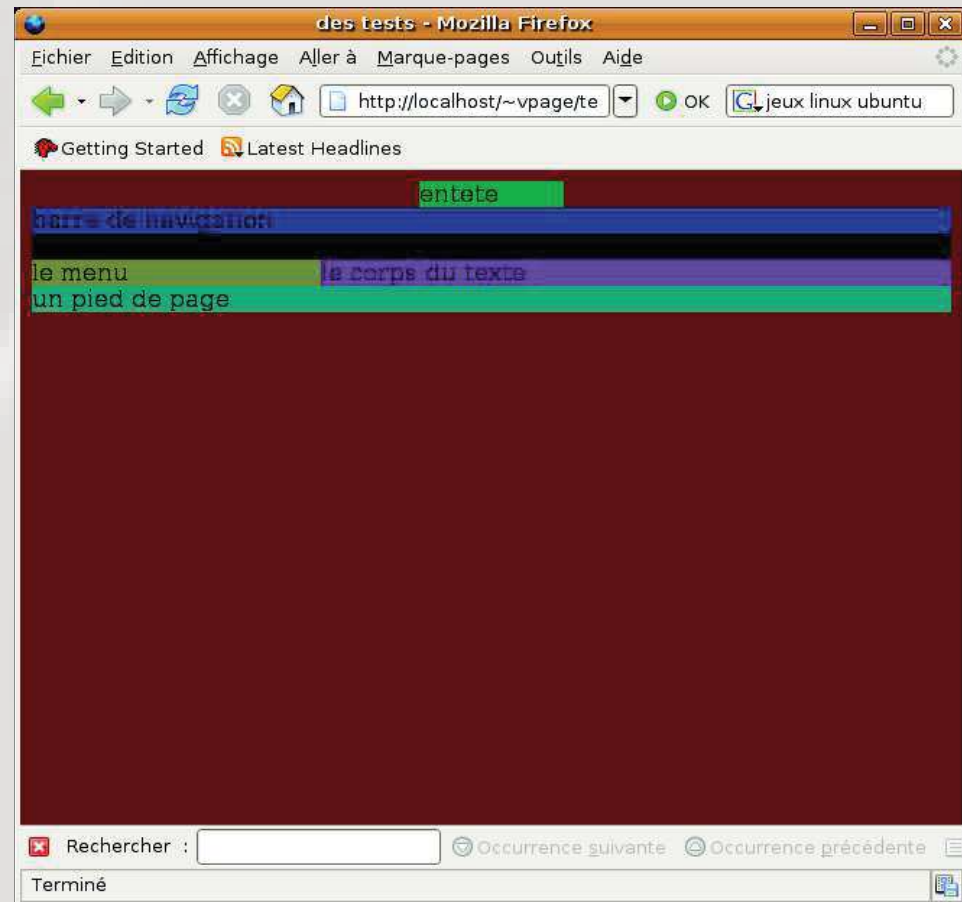
Positionnement et CSS

Positionnement float

on rajoute les lignes suivantes

```
#menu{ float:left;}
```

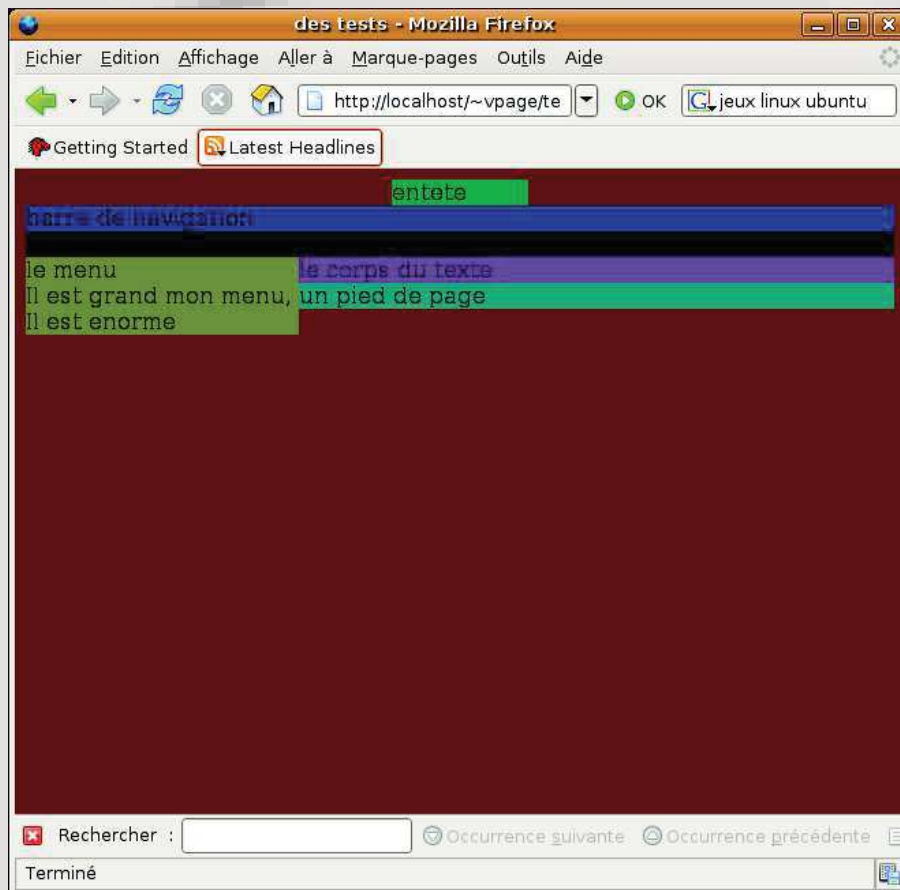
Pour ce résultat :



Positionnement et CSS

Positionnement : float

Mais si on allonge le menu, on a ceci :



La taille de la div de menu n'est pas prise en compte pour le calcul de la taille de la div du milieu...

Notez que le pied de page n'appartient pas à la div milieu....

Positionnement et CSS

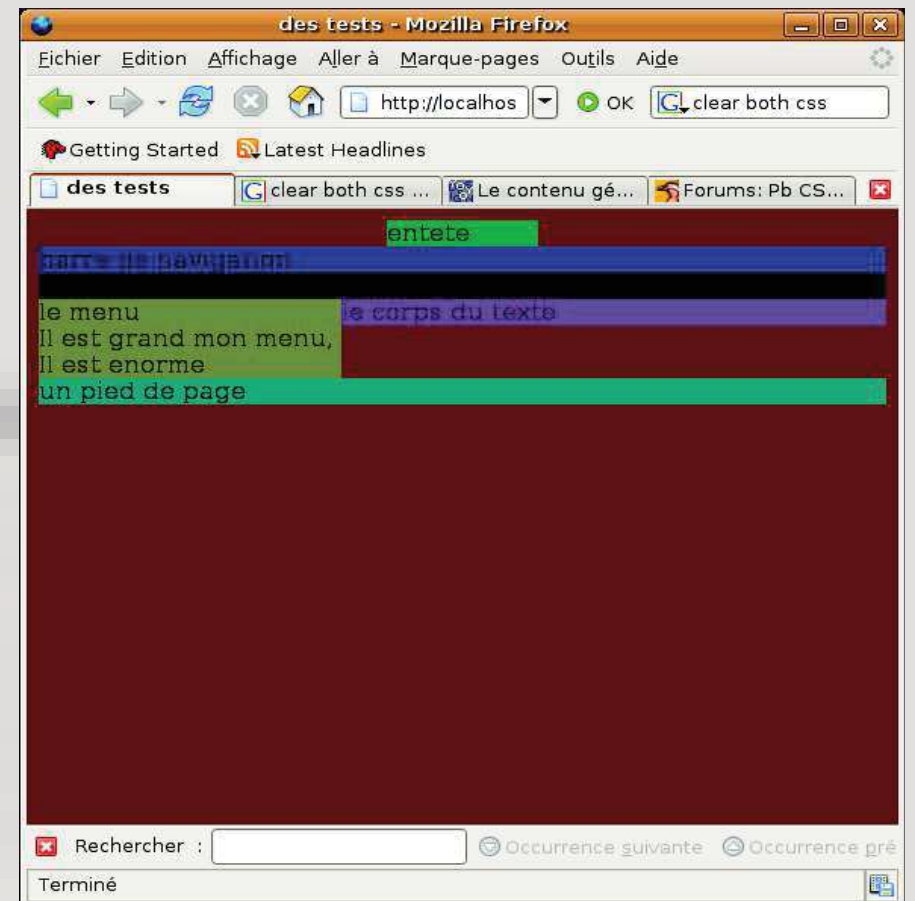
Positionnement : float

Le clear : oblige la boîte à avoir un côté libre. La boîte se place alors sous les autres boîtes flottantes floats du côté spécifié.

on rajoute les lignes suivantes
`#footer{ clear:left;}`

Notez le vide rouge entre le corps de texte et le footer : la div milieu ne s'étend pas jusqu'au pied de page !

On peut avoir clear:right ou clear:both



Positionnement et CSS

Positionnement absolu

Positionner un élément "en dur".

Pour spécifier la position d'un élément, on donne la position de son coin supérieur gauche. On peut aussi donner l'indiquer comme suit :

- haut et droite (+ largeur)

L'élément est retiré du flux. Son positionnement est donc le même quel que soit sa position dans le fichier html...

Positionnement et CSS

Positionnement absolu

Ici on repositionne quelques éléments en absolu.

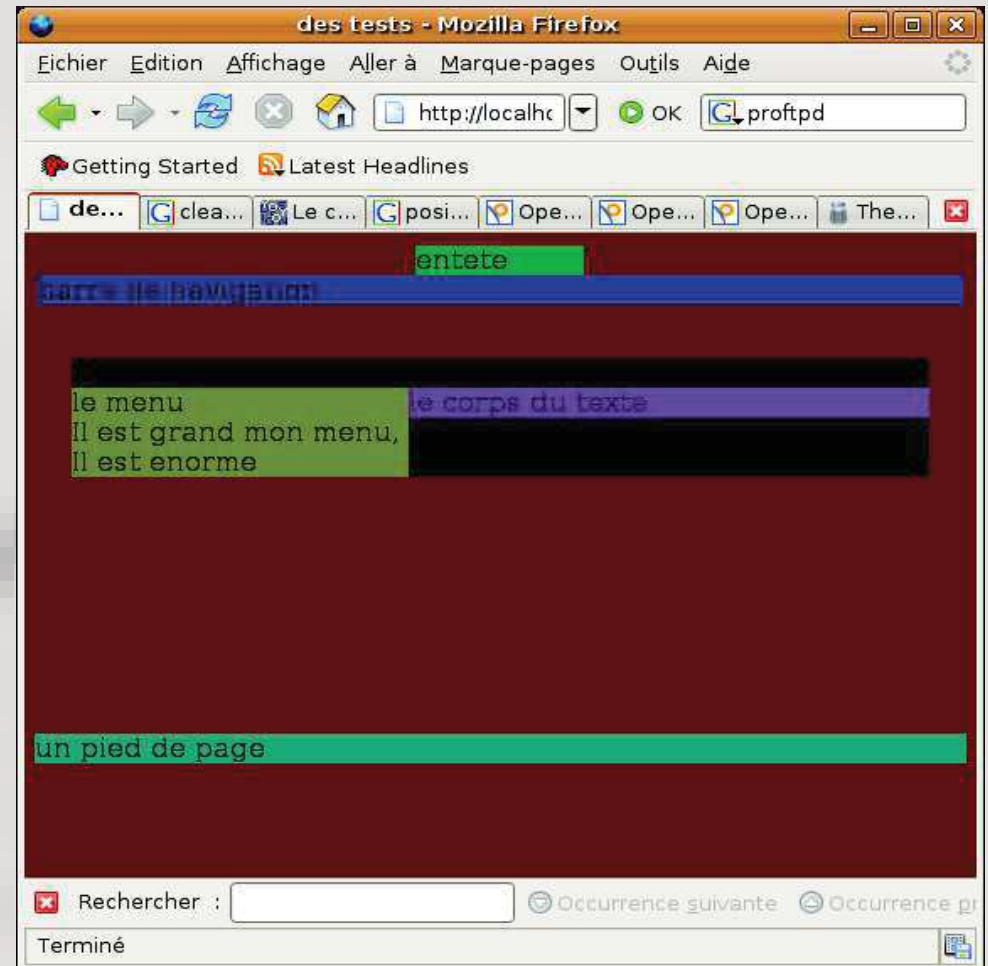
on rajoute les lignes suivantes

```
#milieu{  
position:absolute; top:5em;  
left:5%; right:5%;  
}
```

```
#footer{  
position:absolute;  
top:20em; left:1%; right:1%;  
}
```

Notez la taille de la div milieu.

Attention aux chevauchements...



Positionnement et CSS

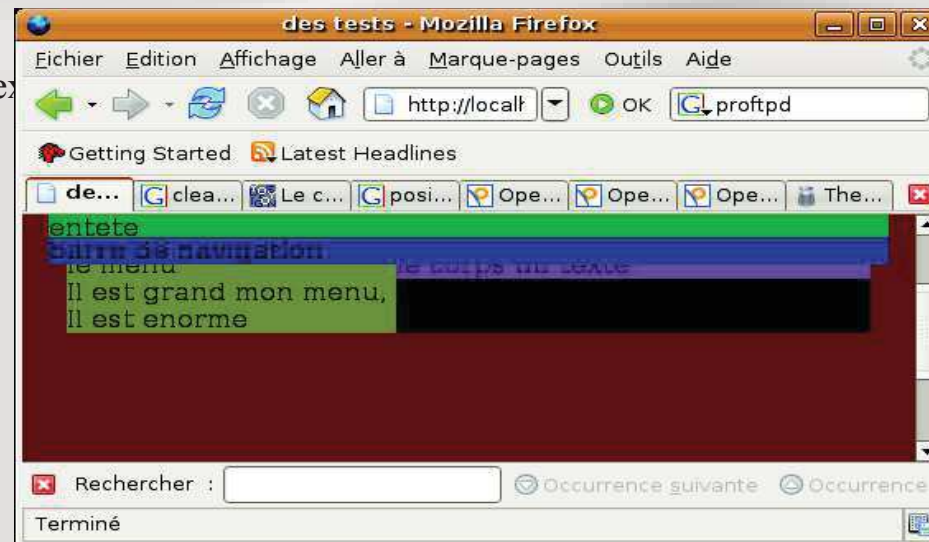
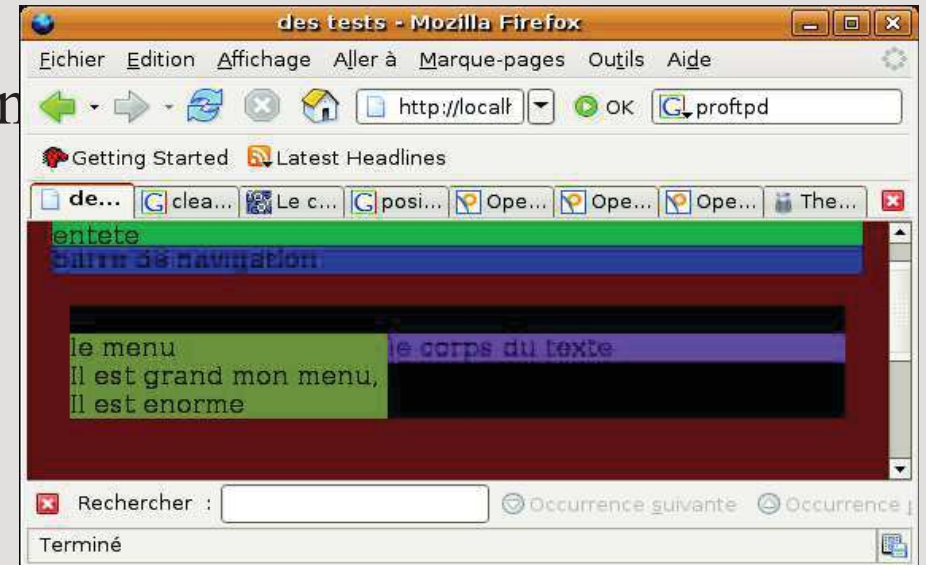
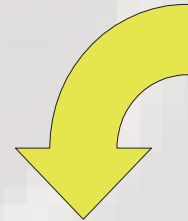
Positionnement fixe

Les éléments fixes ne bougent pas quelque soit la position du défilement dans la fenêtre du navigateur...

Ici, la barre et l'entête sont fixes.

```
#entete#entete{  
position:fixed; top:0; left:1em; right:1em;  
z-index:2; margin : auto;  
background-color : #080;  
}
```

```
#navigation{  
position:fixed;  
top:1em; left:1em; right:1em; z-index:  
background-color : #008;  
}
```



Positionnement et CSS

Positionnement Relatif

Non abordé ici !

Positionnement et CSS

Quelques remarques

- Une balise bloc peut être traitée comme une balise inline et inversement avec la propriété display (:inline ou :bloc)
- Attention à la fusion de marges.
- Certains navigateurs sont récalcitrants :
 - évaluation des tailles, comportement fixed, floats, ils n'interprètent pas les choses de la même façon.
 - Testez vos sites avec plusieurs navigateurs, plusieurs versions des navigateurs.....