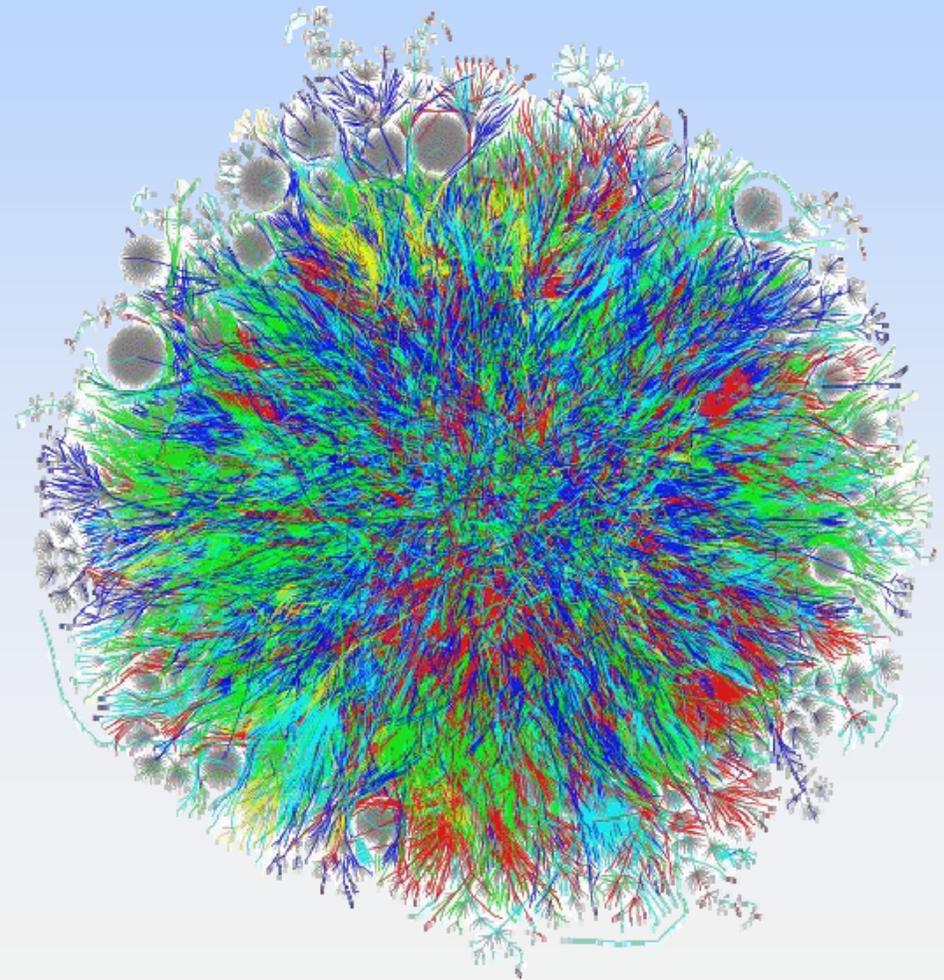


La création de pages Web

Ismael RUIZ

Ingénieur conseil

ismael.ruiz@campomar.lu



© Copyright :

Je vous informe que ce cours constitue une œuvre protégée au Luxembourg et à l'étranger par les conventions internationales en vigueur sur le droit d'auteur. La violation de l'un des droits d'auteur de l'œuvre est un délit de contrefaçon. Il est donc interdit, à titre privé ou public, de reproduire, copier, vendre, revendre ou exploiter, que ce soit dans un but commercial ou purement gratuit, ce cours.

La création de pages Web

Objectifs et prérequis

Objectifs du cours:

- ✓ Être capable d'analyser le code source d'une page web.
- ✓ Créer des pages web et maîtriser les tag les plus courants.

Les logiciels utilisés seront:

- Windows Explorer
- Notepad
- Firefox, Chrome et Internet Explorer

Les connaissances requises:

- Être à l'aise dans un environnement Windows.
- Connaissance de Windows Explorer
- Connaissance d'Internet
- Connaissance de Notepad

La création de pages Web

Environnement logistique

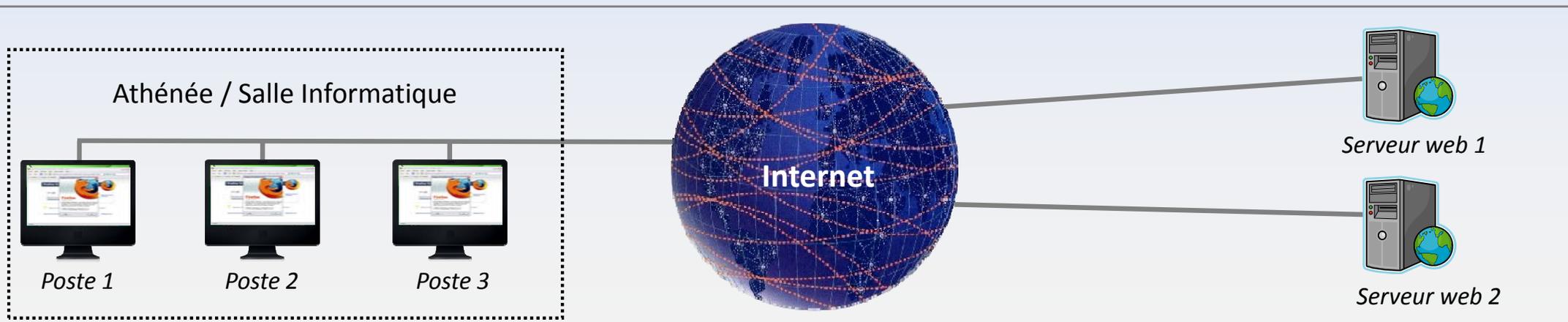
Les PC en réseau :

En général, on travaille à la maison sur un PC qui est connecté à Internet.



Dans ce cours vous travaillerez avec un PC en réseau. Cela veut dire que vous aurez besoin d'un compte + mot de passe pour vous logger dans le système.

user : a1030-userxx
password : a1030-userxx



La création de pages Web

Table des matières

1. Introduction			
1.1. Naissance et définition d'HTML	5		
1.2. Evolution et standardisation d'HTML	6		
1.3. Définition d'HTTP - Définition d'URL	8		
1.4. Méthode d'accès	10		
1.5. Concept de TAG et syntaxe	11		
1.6. La structure du document	13		
1.7. Interprétation du texte	17		
1.8. Liens absolus et liens relatifs	19		
1.9. Travaux pratiques	20		
2. Description d'HTML			
2.1. Système de numération	21		
2.2. La définition des couleurs en RGB	24		
2.3. Les liens et les ancrs	26		
2.4. L'insertion d'images	28		
2.5. l'image cliquable	32		
2.6. Les listes	35		
2.7. L'insertion de texte	37		
2.8. Les formulaires	38		
2.9. L'insertion d'audio et de vidéo	43		
2.10. Travaux pratiques	44		
3. Le tableau			
3.1. Le tableau en html	45		
3.2. Le tableau simplifié	46		
3.3. Les différents attributs du tableau	47		
3.4. Travaux pratiques	48		
4. Le formatage du texte en html			
4.1. Les titres	49		
4.2. Les éléments de formatage HTML	50		
4.2. Travaux pratiques	51		
5. Les feuilles de style (style sheet)			
5.1. Les feuilles de style css	52		
5.2. Travaux pratiques	54		
6. Le concept de couches (layer)			
6.1. Le tag div	53		
6.3. Travaux pratiques	56		
7. L'association div + css			
7.1. La relation entre layer et stylesheet	57		
7.2. Travaux pratiques	58		
8. Les navigateurs			
8.1. Les différents systèmes d'exploitation	59		
8.2. Les différents navigateurs	60		
8.3. Quels navigateurs faut-il tester	61		
9. Le projet : Création de pages web			
10. Le projet : Création de pages web			

1. Introduction

1.1. Naissance et définition d'HTML

Problématique initiale :

Echange de documents affichables sur plates-formes différentes (Windows NT, Macintosh, SUN, ...)

- ✓ **1969** Création de **GML** (**G**eneralized **M**arkup **L**anguage) par C. Goldfarb, E. Mosher et R. Lorie de IBM
- ✓ **1978** Standardisation de GML par **ANSI** (**A**merican **N**ational **S**tandards **I**nstitute) et **ISO** (**I**nternational **O**rganization for **S**tandardization).
Cela devient le **SGML** (**S**tandard **G**eneralized **M**arkup **L**anguage)
- ✓ **1990** Création de **HTML** (**H**yper**T**ext **M**arkup **L**anguage)
 - **HTML** facilite la rédaction et l'échange de documents hypertextes au sein du **World Wide Web**.
 - **HTML** est un sous-langage SGML permettant de spécifier les propriétés d'un texte grâce à des balises. (**TAGs** en anglais)
 - La **Document Type Definition (DTD)**, contient la description des TAGs en langage SGML ou XML.

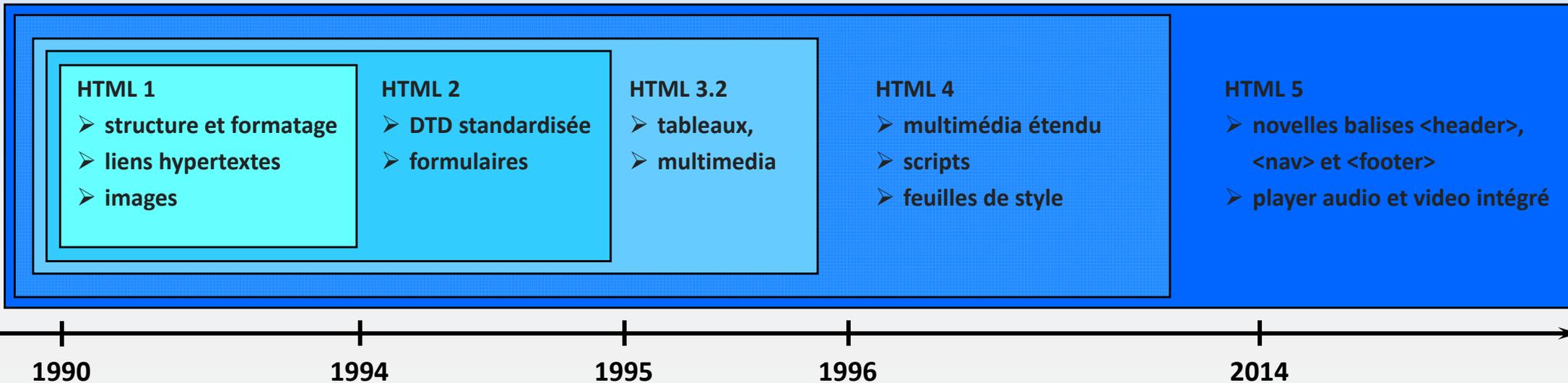
1. Introduction

1.2. Evolution et standardisation d'HTML

- HTML 1.0 : obsolète (*CERN, début des années 90*)
- HTML + : successeur de HTML 1.0, obsolète
- HTML 2.0 : première standardisation officielle (*IETF : RFC 1866, novembre 1995*)
- HTML 3.0 : défini en mars 95, abandonné
- HTML 3.2 : seconde recommandation officielle (*W3C : REC-html32, janvier 97*)

- HTML 4.0 : troisième recommandation officielle (*W3C : REC-html40, décembre 97*)

- ✓ **HTML 5 :** **recommandation officielle actuelle**



1. Introduction

1.2. Evolution et standardisation d'HTML

L'Internet, le World Wide Web et, entre autres, HTML sont standardisés par deux organismes :

➤ Internet Engineering Task Force (**IETF**) : www.ietf.org

- Communauté internationale, créée en 1986
- Rassemble dans des **Workgroups** les personnes qui se sentent concernées par l'évolution et le bon fonctionnement de l'Internet
- Standards : **Request For Comments (RFC)**

➤ World Wide Web Consortium (**W3C**) : www.w3.org

- Consortium industriel international, créé en 1994
- Propose une base de documentation sur le World Wide Web
- Développe des applications et des prototypes afin de présenter et tester les nouveaux standards
- Standards : **RECommendation (REC)**

✓ Web Accessibility Initiative (**WAI**) : www.w3.org/WAI/

La principale mission du WAI est de proposer des solutions techniques pour rendre le World Wide Web accessible aux handicapés et d'une manière plus générale à toute personne sans nécessiter de prérequis particulier.

Les actions de WAI se situent dans cinq domaines :

- les technologies du Web
- le développement de recommandations
- le développement d'outils
- l'information et la formation
- la recherche et le développement

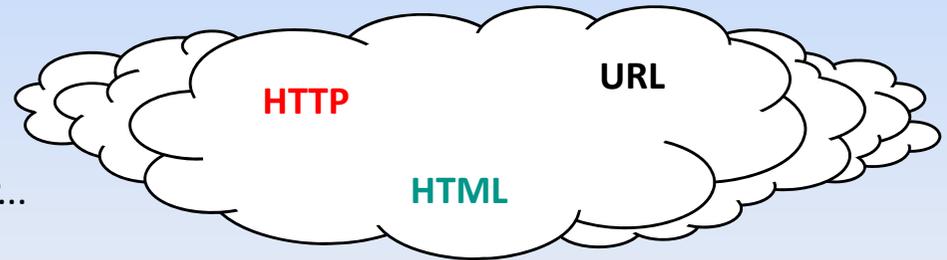
1. Introduction

1.3. Définition d'HTTP - Définition d'URL

➤ L'HyperText Transfer Protocol, **HTTP** (IETF : RFC 2068) est le protocole du World Wide Web permettant le transfert de ressources : texte, images, sons, video...

➤ **URL** : Uniform Resource Locator (IETF : RFC 738 et RFC 1808)

- Chaîne de caractères formatée qui identifie et localise de manière unique une ressource (texte, image ou son) au sein de l'Internet
- **Une URL se compose de six parties :**
 - le protocole ou la méthode d'accès
 - le nom DNS ou l'adresse IP du serveur WWW, FTP, NNTP...
 - [le port TCP/IP (80 par défaut pour HTTP)]
 - [le chemin d'accès à la ressource]
 - [le nom de la ressource]
 - [# suivi d'un nom d'une "cible" interne]
 - [? suivi de paramètres pour un moteur de recherche ou un script]



Remarque : la portion de l'URL qui identifie le fichier HTML est sensible à la casse

1. Introduction

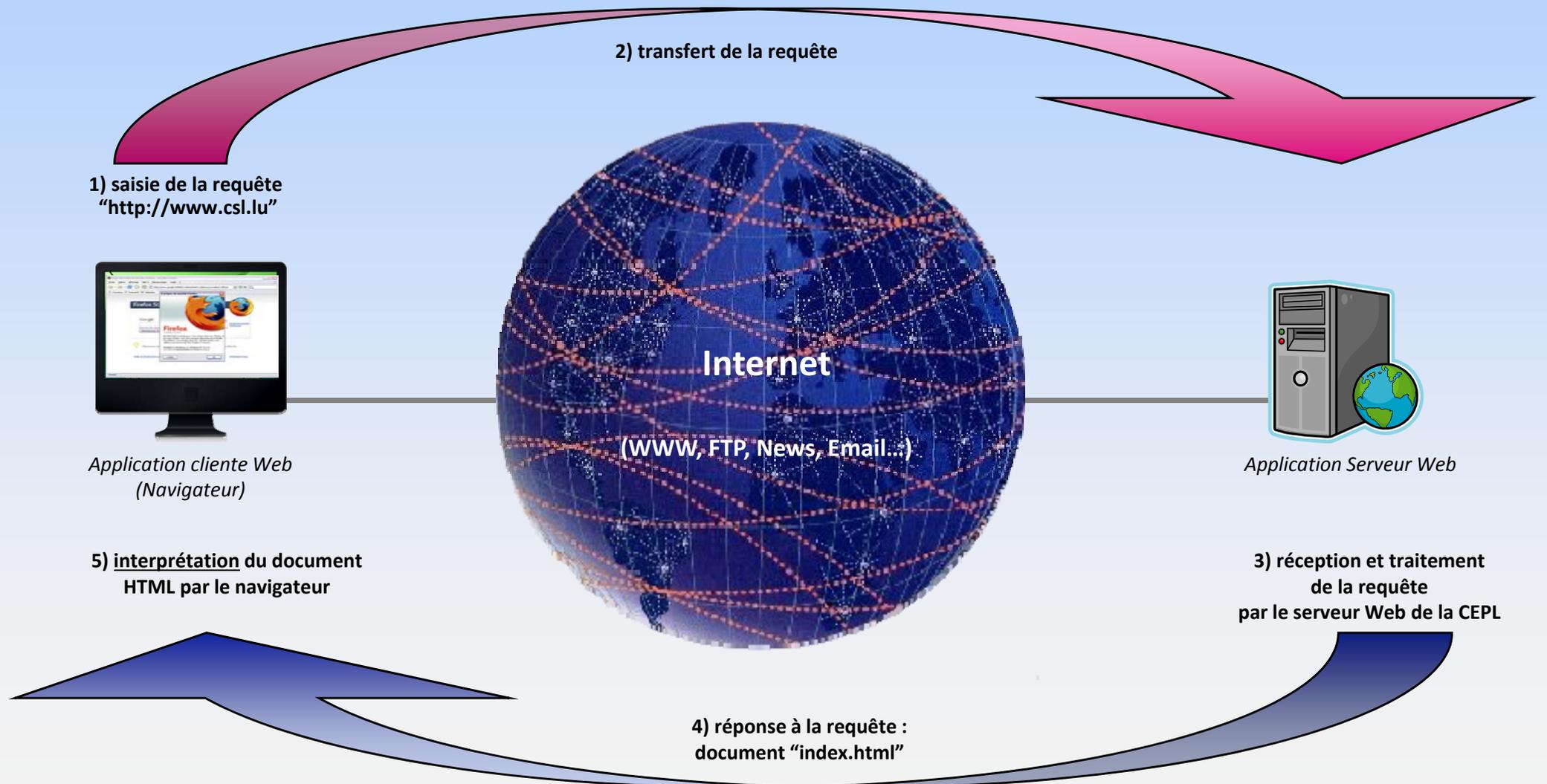
1.3. Définition d'HTTP - Définition d'URL

Exemples :

- <http://dept.uni.edu:8002/docs/div/file.html>
- <http://www.crpcu.lu/rapact/fr/credi.htm#GED>
- <http://search.yahoo.com/bin/search?p=html>
- https://www.google.lu/?gws_rd=ssl
- <ftp://ftp.dept.uni.edu/directory/file.ps.Z>
- <news://news.uni-stuttgart.de/alt.hypertext>
- <file://localhost/C:/Temp/test/demo.html>
- <file:///C:/Temp/test/demo.html>
- <mailto:formation@cepl.lu>

1. Introduction

1.4. Méthode d'accès



1. Introduction

1.5. Concept de TAG et syntaxe

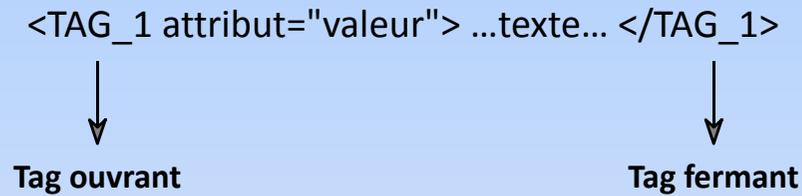
- Les TAGs permettent de définir :
 - les **propriétés logiques** d'un document : titres, sous-titres, paragraphes, listes...
 - les **propriétés physiques** d'un document : gras, italique, alignements, couleurs...
- Un TAG est un mot-clé encadré par les signes “inférieur à” et “supérieur à” : **<nom_du_tag>**
 - Exemple : **** pour un formatage du texte en **bold**

❖ Remarques importantes :

- Les TAGs vont, le plus souvent, par paires : un TAG ouvrant et un TAG fermant
- les TAGs sont insensibles à la casse
- le TAG ouvrant peut contenir des attributs qui en modifient le comportement par défaut
- les attributs sont optionnels et peuvent être spécifiés dans n'importe quel ordre
- les TAGs peuvent être imbriqués ce qui permet d'en cumuler les effets

1. Introduction

1.5. Concept de TAG et syntaxe



Pour valider la syntaxe de votre code, utilisez cette adresse :

<http://validator.w3.org>

1. Introduction

1.6. La structure du document

Tout document HTML conforme à la spécification HTML doit comporter :

- la **spécification du type de document**
- la **déclaration du document** même avec :
 - une **entête** qui contient des informations relatives au document
 - un **corps** qui contient le texte formaté à afficher

Spécification du type de document

La Document Type Declaration (DTD) correspond à un commentaire placé en début du document HTML.

Ce commentaire indique :

- que le document est un document HTML
- que l'HTML utilisé se conforme à une DTD (grammaire) particulière

Cette déclaration est facultative mais conseillée.

Exemples :

HTML4 : `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`

HTML5 : `<!DOCTYPE html>`

1. Introduction

1.6. La structure du document

Déclaration du document

<HTML>

<HEAD>

<TITLE>

</TITLE>

</HEAD>

<BODY>

</BODY>

</HTML>

entête du document HTML

corps du document HTML

<HTML>	Ceci est le début d'un document de type HTML.
</HTML>	Ceci est la fin d'un document de type HTML.
<HEAD>	Ceci est le début de la zone d'en-tête (prologue au document proprement dit contenant des informations destinées au navigateur).
</HEAD>	Ceci est la fin de la zone d'en-tête.
<TITLE>	Ceci est le début du Titre de la page.
</TITLE>	Ceci est la fin du Titre de la page.
<BODY>	Ceci est le début du document proprement dit.
</ BODY>	Ceci est la fin du document proprement dit.

Remarque :

C'est en théorie par le TAG <HTML> que l'interpréteur devrait reconnaître le document HTML...
en réalité, c'est par l'extension du document (html, htm, asp...).

1. Introduction

1.6. La structure du document

Entête du document

- Section contenant de l'information sur le document et son contexte
- Délimitée par les TAGs : <HEAD>...</HEAD>
- Doit contenir le titre du document : <TITLE>...**titre du document**...</TITLE>
- Peut contenir les TAGs : <LINK>, <META>
 - <LINK> :
Rattachement des feuilles de style externes
Exemple : `<link type="text/css" rel="stylesheet" href="..../css/mystyles.css">`
 - <META> :
Définition des informations pour le document courant : informations sur l'auteur, date d'expiration du document, affichage automatique d'un autre document, liste de mots-clés en différentes langues, informations propriétaires uniquement interprétables par certains navigateurs

1. Introduction

1.6. La structure du document

Corps du document

- Délimité par les TAGs : <BODY>...</BODY>
- Peut contenir : du texte, des images, des tableaux, des liens hypermédias, des listes, des formulaires ...
- Quelques exemples d'attributs :
 - **alt:** description d'une image
 - **class:** spécifie une classe
 - **href:** emplacement pour un lien
 - **id:** spécifie un identifiant
 - **src:** emplacement pour une image
 - **style:** insertion d'une règle css

1. Introduction

1.7. Interprétation du texte

Généralités :

- L'interpréteur HTML traduit les espaces multiples, les tabulations et retours chariots contenus dans un document HTML par un espace simple
- Un tag mal orthographié, ou inconnu, sera ignoré par l'interpréteur mais pas le texte qu'il marque
- Des caractères spéciaux inconnus seront affichés comme du texte standard

Encodage de quelques caractères spéciaux et étendus :

 espace insécable			
<	<	é	é
>	>	ê	ê
"	"	è	&egrav;
&	&	È	È
©	©	ß	ß
®	®	Æ	Ø
±	±	æ	æ

1. Introduction

1.7. Interprétation du texte

Encodage des caractères spéciaux et étendus :

Les caractères étendus contenus dans un document HTML sont quand même interprétés par les navigateurs récents :

- car la table de caractères (charset) est souvent spécifiée dans la section HEAD :

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; CHARSET=utf-8">
```

- car les navigateurs actuels permettent de spécifier une table de caractères par défaut

1. Introduction

1.8. Liens absolus et liens relatifs

Lien absolu:

Le lien absolu se fait à partir de la racine, définie pour chaque site web.

Liens relatifs:

Le lien relatif se fait à partir de la page sur laquelle on se trouve. Le chemin d'accès vers un répertoire supérieur se fait avec deux points qui se suivent (..).

Exemples à partir de l'illustration:

1. Comment se ferait le lien à partir de la page1.html vers la page page2.html ?
2. Comment se ferait le lien à partir de la page1.html vers l'image Sunset.jpg ?

Liens absolus:

1. Mon_site_web/html/page2.html
2. Mon_site_web/html/images/Sunset.jpg

Liens relatifs:

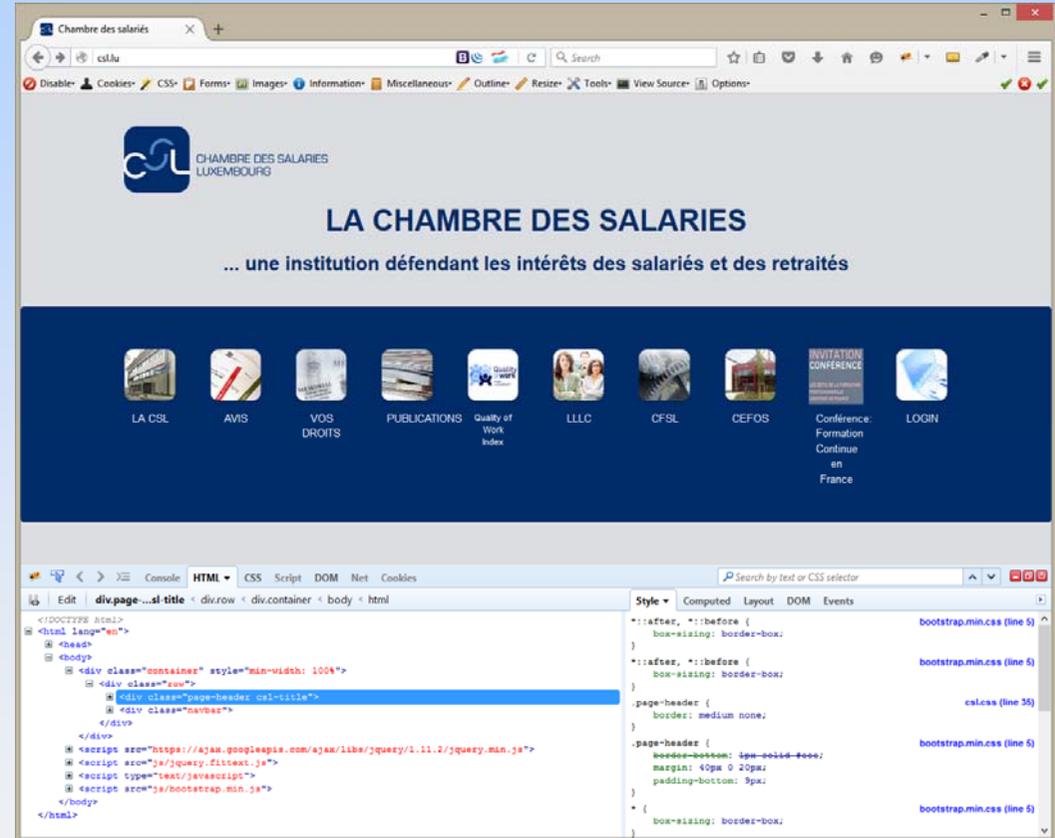
1. page2.html
2. ../images/Sunset.jpg



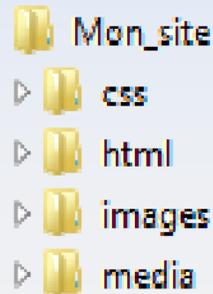
1. Introduction

1.9. Travaux pratiques

1. Installation de 'Firebug' dans le navigateur Firefox. Naviguez sur « www.csl.lu » et analysez le code source.



2. Création de cette arborescence sur le disque partagé et copie sur le disque usb.



3. Création de votre première page html, validation de la syntaxe et visualisation dans un navigateur.

2. Description d'HTML

2.1. Système de numération

Le système décimal (base 10)

Ce système est utilisé dans nos activités quotidiennes. Il est basé sur 10 chiffres:

0 1 2 3 4 5 6 7 8 9

L'addition de dernier chiffre et du deuxième se fait de la manière suivante: $9 +$

$$\begin{array}{r} 1 \\ \hline 10 \end{array}$$

Le chiffre décimal 1971 quant à lui se compose de la façon suivante:

$$1971 = 1000 + 900 + 70 + 1 \text{ ou } 1 \cdot 10^3 + 9 \cdot 10^2 + 7 \cdot 10^1 + 1 \cdot 10^0$$

On parle alors (de la droite vers la gauche) d'unités (10^0), de dizaines (10^1), de centaines (10^2), de milliers (10^3),

2. Description d'HTML

2.1. Système de numération

Un système binaire (base 2)

Il est à la base du système de numération utilisé en informatique (1 le courant passe, 0 le courant ne passe pas). Il est basé sur 2 chiffres:

0 1

L'addition de dernier chiffre et du deuxième se fait de la manière suivante:

$$\begin{array}{r} 1 + \\ 1 \\ \hline 10 \end{array}$$

Pour trouver la valeur décimale d'un chiffre binaire il faut le décomposer de la même façon qu'un chiffre décimal. On ne travaille cependant pas avec un base de 10, mais avec une base de 2.

Le chiffre binaire 1101 se compose donc de la façon suivante:

$$1101 = 1*2^3 + 1*2^2 + 0*2^1 + 1*2^0 = 1*8 + 1*4 + 0*2 + 1*1 = 13$$

Le chiffre binaire 1101 a donc comme valeur décimale 13.

2. Description d'HTML

2.1. Système de numération

Un système hexadécimale (base 16)

Le système hexadécimal est utilisé notamment en électronique numérique et en informatique. En effet, chaque chiffre hexadécimal correspond exactement à quatre chiffres binaires (ou bits), rendant les conversions très simples et fournissant une écriture plus compacte.. Il est basé sur 16 chiffres:

0 1 2 3 4 5 6 7 8 9 A B C D E F

L'addition de dernier chiffre et du deuxième se fait de la manière suivante:

$$\begin{array}{r} F + \\ \hline 1 \\ 10 \end{array}$$

Pour trouver la valeur décimale d'un chiffre hexadécimal il faut le décomposer de la même façon qu'un chiffre décimal. On ne travaille cependant pas avec un base de 10, mais avec une base de 16.

Le chiffre hexadécimal A1B4 se compose donc de la façon suivante:

$$A1B4 = 10*16^3 + 1*16^2 + 11*16^1 + 4*16^0 = 10*4096 + 1*256 + 11*16 + 4*1 = 40960 + 256 + 176 + 4 = 41396$$

Le chiffre hexadécimal A1B4 a donc comme valeur décimale 41396.

Le chiffre hexadécimal EC se compose de la façon suivante:

$$EC = 14*16^1 + 12*16^0 = 14*16 + 12*1 = 224 + 12 = 236$$

Le chiffre hexadécimal EC a donc comme valeur décimale 236.

2. Description d'HTML

2.2. La définition des couleurs RGB

Comment sont affichés les couleurs sur un écran?

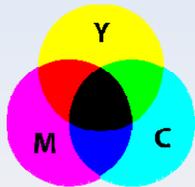
L'écran RGB construit toutes les couleurs avec ces trois couleurs primaires:

- Red (Rouge)
- Green (Vert)
- Blue (Bleu)

Pour faire du jaune l'écran mélange le rouge et le vert.

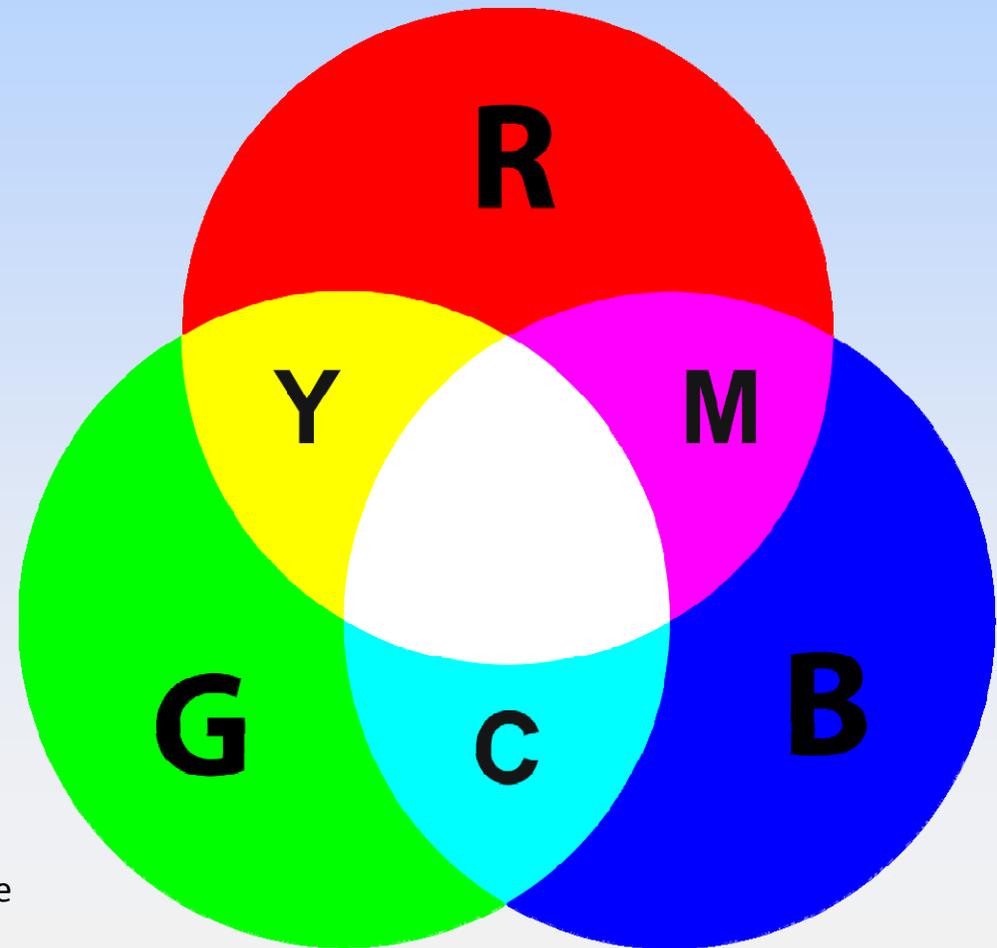
Votre imprimante par contre travaille avec le trois couleurs suivantes.

- Cyan
- Magenta
- Yellow



Pour faire du rouge l'imprimante mélange le magenta et le jaune.

C'est pour cela que les couleurs sur l'écran ne sont pas identiques à celles de votre impression!



2. Description d'HTML

2.2. La définition des couleurs RGB

Toutes les couleurs sont ainsi composés de rouge, vert et bleu. Chaque intensité est représenté par deux chiffres hexadécimaux. Voici un tableau montrant les couleur à mémoriser:

Couleur	Nom	Valeur hex	Valeur déc	Code html	Commentaires
	blanc	ff ff ff	255 255 255	#ffffff	La couleur blanche, par exemple, est faite par chacune des trois couleurs primaires à leur pleine intensité
	rouge	ff 00 00	255 0 0	#ff0000	
	vert	00 ff 00	0 255 0	#00ff00	
	bleu	00 00 ff	0 0 255	#0000ff	
	jaune	ff ff 00	255 255 0	#ffff00	Le jaune résulte du mélange de rouge et de vert.
	magenta	ff 00 ff	255 0 255	#ff00ff	Le magenta résulte du mélange de rouge et de bleu.
	cyan	00 ff ff	0 255 255	#00ffff	Le cyan résulte du mélange de vert et de bleu.
	gris clair	c8 c8 c8	200 200 200	#c8c8c8	Les niveaux de gris se composent d'intensités égales de chaque couleur. Plus il y a d'intensité, plus on se rapproche de blanc (gris clair). Moins il y a d'intensité, plus on se rapproche de noir (gris foncé).
	gris foncé	3c 3c 3c	34 34 34	#3c3c3c	
	noir	00 00 00	0 0 0	#000000	Noir, c'est l'absence de toute couleur sur un écran

2. Description d'HTML

2.3. Les liens et les ancres

Le tag utilisé : `<a>` ``

Ce tag peut être utilisé de deux façons:

- Comme lien vers un autre fichier ou pointeur (départ)
- Comme pointeur (destination)

Voici les attributs les plus importants:

href	URI de la ressource reliée
name	extrémité du lien nommée
target	restituer dans ce cadre
shape	à utiliser avec les images cliquables côté client

2. Description d'HTML

2.3. Les liens et les ancres

Quelques exemples:

1. `Just an anchor named "anchor3."`
2. `A link to navigate to "anchor3" in the same document.`
3. `Go from here (anchor 3) to home page.`

2. Description d'HTML

2.4. L'insertion d'images

Quels formats d'images peut-on utiliser sur Internet ?

1. Format gif
2. Format jpg
3. Format png

Si vous vous occupez personnellement de la création ou de l'optimisation des images de vos pages Web, vous serez amené à choisir l'un ou l'autre de ces formats. Lequel? La réponse dépend des différents facteurs, les deux principaux étant la taille (pixels) du fichier résultant et la qualité souhaitée.

Le format gif

Le format gif est le premier format spécifiquement destiné au web. Il a initialement été mis au point par CompuServe. Ses caractéristiques essentielles sont les suivantes:

- *maximum de 256 couleurs*
- *compression sans perte de qualité pouvant atteindre un rapport 1:3, voire beaucoup plus selon les images*
- *possibilité de transparence (un seul niveau de transparence)*
- *possibilité de gif entrelacé*

2. Description d'HTML

2.4. L'insertion d'images

Le format jpg

Le format jpg est destiné à la compression d'images photographique, qui comprennent donc de nombreuses variations de tons. Ses principales caractéristiques sont les suivantes :

- millions de couleurs
- compression avec perte de qualité variable selon les images, et qui peut être paramétré lors de l'enregistrement; le taux de compression peut atteindre 1:10, voire beaucoup plus.
- possibilité de jpg progressif

Le format png

Le format png , comme le format gif, a été spécialement développé pour le web (png signifie portable network graphics). Il ressemble au format gif par certains aspects, mais ses possibilités sont plus étendues. Ses caractéristiques sont les suivantes :

- millions de couleurs, mais possibilité de limiter le nombre de couleurs à 256
- compression sans perte de qualité de même type que le format gif, avec des taux de compression légèrement supérieures
- possibilité de 256 niveaux de transparence
- possibilité de png entrelacé

L'inconvénient de ce format est qu'il est incompatible avec les navigateurs plus anciens.

2. Description d'HTML

2.4. L'insertion d'images

Le tag utilisé : ` `

- En HTML, l'image ne fait pas partie de votre document. Le browser va la chercher à l'adresse indiquée.
- Presque en conséquence logique de ceci, le fait d'utiliser la même image à plusieurs reprises dans un fichier HTML ne modifie en rien sa taille.
- Prévoir un texte pour les browsers n'ayant pas l'option image activée, permet au lecteur de ne pas perdre le fil de l'exposé et peut-être d'activer cette option pour découvrir votre oeuvre.
- Il est important pour la fluidité de l'affichage de préciser la taille en hauteur et largeur de l'image car le browser peut ainsi, connaissant l'emplacement à réserver pour celle-ci, continuer à afficher le texte..
- Nous pouvons très bien faire un lien sur une image (``)

2. Description d'HTML

2.4. L'insertion d'images

Voici les attributs les plus importants:

src	URI de l'image à incorporer
alt	brève description
border	épaisseur de la bordure du lien
height	surclasser la hauteur
width	surclasser la largeur
name	nom de l'image pour les scripts
usemap	utiliser une image cliquable côté client

Quelques exemples:

1. ``

2. `` (image et lien)

2. Description d'HTML

2.5. L'image cliquable

Le tag utilisé : `<map> </map>` + `<area> </area>` + attribut `usemap` dans le tag ``

Pour créer une image cliquable il faut faire deux choses:

- Ajouter l'attribut `usemap` dans le tag `` de l'image en question
- Créer les zones sensibles dans le tag `<map> </map>`

Voici les attributs les plus importants:

IMG	
usemap	utiliser une image cliquable côté client

MAP	
name	pour appel par l'attribut <code>usemap</code>

AREA	
href	URI de la ressource reliée
alt	brève description
coords	liste de longueurs séparées par des virgules
shape	contrôle l'interprétation des coordonnées
target	restituer dans ce cadre

2. Description d'HTML

2.5. L'image cliquable

Vous avez trois possibilités de créer des zones:

rectangulaire : shape="rect"	on définit les coordonnées du point en haut à gauche en x puis en y, puis le point en bas à droite en x puis en y. Ce qui donne : coords="x1, y1, x2, y2"
cercle : shape="circle"	on définit le centre du cercle puis le rayon en pixels. On obtient donc : coords="x1, y1, rayon"
polygone : shape="polygon"	on définit les coordonnées en x et y pour chaque point du polygone (comme pour le rectangle, mais il y a plus de points).

2. Description d'HTML

2.5. L'image cliquable

Exemple:

```

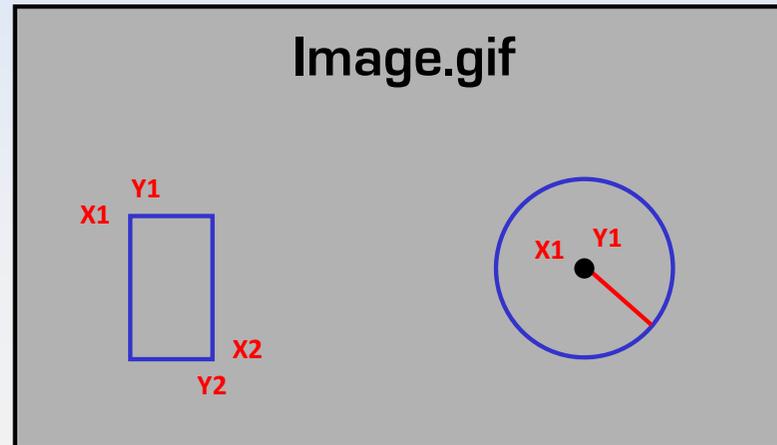
```

```
<map name="nom">
```

```
  <area shape=rect coords="X1,Y1,X2,Y2" href="URL" alt= "URL" >
```

```
  <area shape=circle coords="X1,Y1,rayon" href="URL" alt= "URL" >
```

```
</map>
```



2. Description d'HTML

2.6. Les listes

Les tag utilisés : `` `` ou `` ``

Les listes numérotées

Une liste numérotée est délimitée par les balises `` et `` entre lesquelles chaque élément de la liste est précédé de la balise ``.

Les listes à puces

Une liste non numérotée est définie de la même façon, sauf qu'elle est encadrée par les balises `` et ``.

Voici les attributs:

UL	
compact	espacement inter-items réduit
type	style de l'item de liste
OL	
compact	espacement inter-items réduit
start	numéro commençant la séquence
type	style de numérotation

LI	
type	style de puce
value	réinitialise le numéro dans la séquence

2. Description d'HTML

2.6. Les listes

Exemples:

```
<ul>
  <li>1</li>
  <li>2</li>
  <li>3 </li>
</ul>
```

```
<ol>
  <li>a</li>
  <li>b</li>
  <li>c</li>
</ol>
```

```
<ul>
  <li>aaa
    <ol>
      <li>a</li>
      <li>b</li>
      <li>c</li>
    </ol>
  </li>
  <li>bbb</li>
  <li>ccc</li>
</ul>
```

<code><ul type="disc "></code>	Puces rondes pleine
<code><ul type="circle"></code>	Puces rondes creuses
<code><ul type="square"></code>	Puces carrées
<code><ol type=1 ></code>	(1,2,3)
<code><ol type=a ></code>	(a,b,c)
<code><ol type=A ></code>	(A,B,C)
<code><ol type=i ></code>	(i,ii,iii)
<code><ol type=I ></code>	(I,II,III)

2. Description d'HTML

2.7. L'insertion de texte

Le retour à la ligne.

Tags utilisé : `
`

Ce tag n'a pas de tag fermant!

```
<!DOCTYPE html>
<html>
<body>

<p>This is<br>a para<br>graph with line breaks</p>

</body>
</html>
```

This is
a para
graph with line breaks

Créer un paragraphe.

Tag utilisé : `<p> </p>`

```
<!DOCTYPE html>
<html>
<body>

<p>
This paragraph
contains a lot of spaces
in the source code,
but the browser
ignores it.
</p>

<p>
The number of lines in a paragraph depends on the size of the browser window. If
you resize the browser window, the number of lines in this paragraph will change.
</p>

</body>
</html>
```

This paragraph contains a lot of lines in the source code, but the browser ignores it.

This paragraph contains a lot of spaces in the source code, but the browser ignores it.

The number of lines in a paragraph depends on the size of the browser window. If you resize the browser window, the number of lines in this paragraph will change.

Créer un texte préformaté:

Tag utilisé : `<pre> </pre>`

```
<!DOCTYPE html>
<html>
<body>

<p>The pre tag preserves both spaces and line breaks:</p>

<pre>
  My Bonnie lies over the ocean.

  My Bonnie lies over the sea.

  My Bonnie lies over the ocean.

  Oh, bring back my Bonnie to me.
</pre>

</body>
</html>
```

The pre tag preserves both spaces and line breaks:

My Bonnie lies over the ocean.
My Bonnie lies over the sea.
My Bonnie lies over the ocean.
Oh, bring back my Bonnie to me.

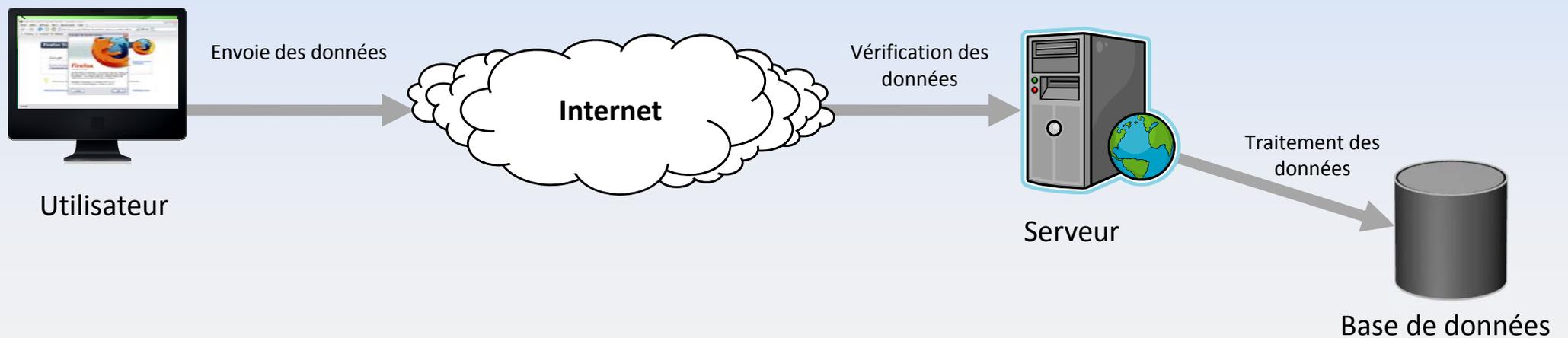
2. Description d'HTML

2.8. Les formulaires

A quoi sert un formulaire ?

Les formulaires permettent de récolter des information sur les utilisateurs.

Il y à donc une interaction entre le client et le serveur. Le formulaire concerne la partie client. Le traitement des données se fait du coté serveur via script. En général les données sont sauvegardés dans une base de données.



2. Description d'HTML

2.8. Les formulaires

Le tag utilisé : `<form> </form>`

Avant de pouvoir utiliser les différentes sortes de formulaires (ligne de texte, liste déroulante, cases à cocher...), il faut déclarer au browser qu'il devra gérer des formulaires et ce qu'il devra en faire.

Voici les attributs:

accept-charset	liste des jeux de caractères reconnus
accept	liste de types MIME pour chargement d'un fichier sur le serveur
action	gestionnaire de formulaire côté serveur
method	méthode HTTP utilisée pour soumettre le formulaire
name	nom du formulaire pour les scripts
onreset	le formulaire a été réinitialisé
onsubmit	le formulaire a été soumis
target	restituer dans ce cadre

2. Description d'HTML

2.8. Les formulaires

Les différents champs qui peuvent être utilisés dans un formulaire:

- Ligne de texte: `INPUT type="text"` → indique un champ de saisie d'une seule ligne.

```
<FORM>  
<INPUT type="text" name="nom" size="50">  
</FORM>
```

- Zone de saisie: La balise `<TEXTAREA>...</TEXTAREA>` introduit une zone de texte multilignes.

```
<FORM>  
<TEXTAREA name="nom" rows=4 cols=40>Valeur par défaut</TEXTAREA>  
</FORM>
```

- Liste déroulante: La balise `<SELECT></SELECT>` indique au browser l'usage d'une liste déroulante.
 - Les éléments de la liste sont introduits par la balise `<OPTION> ... (</OPTION>` facultatif).

```
<FORM>  
<SELECT name="nom" size="1">  
<OPTION Value="valeur1">Valeur 1  
<OPTION Value="valeur2">Valeur 2  
</SELECT>  
</FORM>
```

2. Description d'HTML

2.8. Les formulaires

- Bouton d'option → Les boutons d'option, aussi appelés boutons radio, ont comme particularité qu'une seule option à la fois peut être activée (le "ou" exclusif).

```
<FORM>  
<INPUT type="radio" name="nom du groupe" value="valeur du bouton">  
</FORM>
```

- Case à cocher → La philosophie des cases à cocher [**checkbox**] est assez similaire aux boîtes d'option. Ici, cependant, plusieurs choix simultanés peuvent être réalisés.

```
<FORM>  
<INPUT type="checkbox" name="nom" value="valeur attachée au bouton">  
</FORM>
```

- Bouton de commande → Le bouton de commande sert à placer un bouton, auquel on assignera plus tard une tâche particulière.

```
<FORM>  
<INPUT type="button" name="nom" value="texte du bouton">  
</FORM>
```

2. Description d'HTML

2.8. Les formulaires

- Bouton "Submit " → Le bouton Submit a la tâche spécifique de transmettre toutes les informations contenues dans le formulaire à l'URL désignée dans les attributs **ACTION** et **METHOD** du tag **<FORM>**.

```
<FORM>  
<INPUT TYPE="submit" NAME="nom" VALUE="texte du bouton">  
</FORM>
```

- Bouton "Reset " → Le bouton Reset permet d'annuler les modifications apportées aux contrôles d'un formulaire et de restaurer les valeurs par défaut.

```
<FORM>  
<INPUT TYPE="reset" NAME="nom" VALUE="texte du bouton">  
</FORM>
```

2. Description d'HTML

2.9. L'insertion d'audio et de vidéo

Insertion d'audio

Tag utilize : `<audio>` `</audio>`

```
<audio controls>
```

```
  <source src="horse.ogg" type="audio/ogg">
```

```
  <source src="horse.mp3" type="audio/mpeg">
```

```
  Your browser does not support the audio tag.
```

```
</audio>
```

Insertion de vidéo

Tag utilisé : `<video>` `</video>`

```
<video width="320" height="240" controls>
```

```
  <source src="movie.mp4" type="video/mp4">
```

```
  <source src="movie.ogg" type="video/ogg">
```

```
  Your browser does not support the video tag.
```

```
</video>
```

2. Description d'HTML

2.10. Travaux pratiques

1. Créer page1.html avec une image et lien sur page2.html avec une image et vice et versa.



Observe cette l'image.....



...et tente de trouver les
10 erreurs sur celle-ci.

2. Faire la liste suivante:

- Item 1
- Item 2
 1. Sous-item 2.1
 2. Sous-item 2.2
 3. Sous-item 3.2
- Item 3
- Item 4
 - Sous-item 4.1
 - Sous-item 4.2
- Item 5

3. Le tableau

3.1. Le tableau en html

Le tableau en html comporte trois zones (head, body et foot). Elle sont toujours interprétés dans cet ordre là. Cette syntaxe permet d'expliquer à l'internaute qu'il s'agit précisément d'un tableau.

Syntaxe:

```
<table border = "1">  
<tfoot>  
<tr>  
<td>This text is in the TFOOT</td>  
</tr>  
</tfoot>  
<thead>  
<tr>  
<td>This text is in the THEAD</td>  
</tr>  
</thead>  
<tbody>  
<tr>  
<td> This text is in the TBODY</td>  
</tr>  
</tbody>  
</table>
```

This text is in the THEAD
This text is in the TBODY
This text is in the TFOOT

3. Le tableau

3.2. Le tableau simplifié

Le tableau simplifié est celui qui est beaucoup utilisé. Celui-ci interprète ligne par ligne dans l'ordre de la syntaxe.

Syntaxe:

```
<table border = "1">  
<tr>  
<td>This text is in the TFOOT</td>  
</tr>  
<tr>  
<td>This text is in the THEAD</td>  
</tr>  
<tr>  
<td> This text is in the TBODY</td>  
</tr>  
</table>
```

This text is in the TFOOT
This text is in the THEAD
This text is in the TBODY

3. Le tableau

3.3. Les différents attributs du tableau

Voici les attributs :

TABLE	
align	position de la table relativement à la fenêtre
bgcolor	couleur d'arrière-plan des cellules
border	épaisseur du contour de la table
cellpadding	espacement à l'intérieur des cellules
cellspacing	espacement entre les cellules
width	largeur de la table
TR	
align	alignement du texte
bgcolor	couleur d'arrière-plan des rangées
valign	alignement vertical dans les cellules
TH	
align	alignement du texte
bgcolor	couleur d'arrière-plan des rangées
valign	alignement vertical dans les cellules

TD	
align	alignement du texte
bgcolor	couleur d'arrière-plan des cellules
colspan	nombre de colonnes couvertes par la cellule
height	hauteur de la cellule
nowrap	suppression de la césure
rowspan	nombre de rangées couvertes par la cellule
valign	alignement vertical dans les cellules

3. Le tableau

3.4.Travaux pratiques

1. Création d'un tableau comme suit :

Le bord du tableau est visible. Sa largeur est de 700px et il est centré sur la page. Les colonnes du tableau ont toutes la même largeur. Les marges intérieures des cellules est de 2px. Les marges entre les cellules est de 5px.

1	2	3	4	5	6	7
2	A			B	C	D
3	A			E	F	G
4				G		I
5	J	K	L	M	N	
6	O	P	Q	R	S	T
7	U	V	W	X	Y	Z
8	I	II	III	IV	V	VI
9	VII				VIII	IX
10	X	XI	XII	XIII	XIV	XV

4. Le formatage du texte en html

4.1. Les titres

Les titres sont définis de <h1> à <h6>.

<h1> définit le titre le plus important. <h6> définit le titre le moins important.

Les titres en html sont importants:

- Utilisez exclusivement les titres HTML pour définir les titres et les sous-titres. Ne les utiliser pas pour rendre le texte grand ou gras.
- Les moteurs de recherche utilisent vos titres pour indexer de la structure et le contenu de vos pages Web.
- Les internautes utilisent les titres pour parcourir vos pages. Il est important d'utiliser des rubriques pour montrer la structure du document.
- h1 rubriques devraient être le titre principal, suivis par les sous-titres h2, h3 sous-titre moins important, et ainsi de suite.

Le format de <h1> à <h6> sont prédéfinis dans tous les navigateurs.

4. Le formatage du texte en html

4.2. Les éléments de formatage HTML

HTML définit des éléments spéciaux pour définir le texte avec une signification particulière.

HTML utilise des éléments comme `` et `<i>` pour formater le texte, comme le texte en gras ou en italique.

Les éléments de mise en forme ont été conçus pour afficher des types de texte particuliers:

Texte en gras	<code></code>
Texte important	<code></code>
Texte italique	<code><i></code>
Texte emphasis	<code></code>
Le texte marqué	<code><mark></code>
Petit texte	<code><small></code>
Le texte supprimé	<code></code>
Texte souligné	<code><ins></code>
Text _{indices}	<code><sub></code>
Text ^{Exposants}	<code><sup></code>

4. Le formatage du texte en html

4.2. Travaux pratiques

Créez le texte suivant sur une page html:

Polynômes et équations du second degré

Savoir-faire :

Mettre un polynôme du second degré sous forme canonique

Déterminer les racines d'un polynôme du second degré

Etudier le signe d'un polynôme du second degré

Utiliser la forme la plus adaptée (développée, canonique, factorisée) d'un polynôme du second degré en vue de résoudre un problème

1. Polynômes du second degré

Définition

On appelle **polynôme (ou trinôme) du second degré** toute expression pouvant se mettre sous la forme :

$$P(x)=ax^2+bx+c$$

où a , b et c sont des réels avec $a \neq 0$

Exemples

$P(x)=2x^2+3x-5$ est un polynôme du second degré.

$P(x)=x^2-1$ est un polynôme du second degré avec $b=0$ mais $Q(x)=x-1$ n'en est pas un car a n'est pas différent de zéro. (C'est un polynôme du premier degré-ou une fonction affine)

$P(x)=5(x-1)(3-2x)$ est un polynôme du second degré car en développant on obtient une expression du type souhaité.

2. Equations du second degré

Définition

On appelle **racine** d'un polynôme P une solution de l'équation $P(x)=0$

Remarque

Ne pas confondre les mots "*racine*" et "*racine carrée*" !

5. Les feuilles de styles (stylesheet)

5.1. Les feuilles de styles CSC

Les feuilles de styles sont la meilleure façon d'appliquer le formatage du texte et des balises, car tout ce qui concerne le formatage se fera dans un fichier séparé. Cela permet d'avoir un code html 'propre'.

Il faudra alors rattacher la page html avec le fichier csc (feuille de style).

Tag utilisé : `<link> </link>`

Voici les attributs :

href	URI de la ressource reliée
rel	types de lien régressif
type	type de contenu consultatif

Exemple:

```
<link href="../../1030/Mon_site/css/style.css" rel="stylesheet" type="text/css" />
```

5. Les feuilles de styles (stylesheet)

5.1. Les feuilles de styles CSC

Une fois crée le lien avec le fichier css on peut définir dans celui-ci des nouvelles classes ou bien redéfinir toutes les balises existantes. Ceci est très performant ,car tout définition de formatage ne se fait plus dans le code source de la page html.

La définition des nouvelles classes comme celles des balises se fait comme ceci:

```
body
{
text-align: center;
background-color: #0000ff;
font-size: 12px;
}
```

```
.titre
{
text-align: center;
color: #ff00ff;
font-size: 10px;
}
```

5. Les feuilles de styles (stylesheet)

5.1. Travaux pratiques

- Créer un répertoire css.
- Créer un fichier texte avec l'extension css.
- Créer une page html dans laquelle vous allez définir le lien vers le fichier css.
- Expérimentez les commandes css via l'adresse web de w3schools (<http://www.w3schools.com/css/>).
 - margin et padding
 - border
 - color
 - background
 - width et height
 - font

6. Le concept de couches (layers)

6.1. Le tag div

Le concept de couches donne la possibilité des créer différentes zones sur une page html. Ces zones ont leur propre contenu.

Tag utilisé : `<div> </div>`

Voici l'attribut le plus important :

style	Définit les attributs du layer
--------------	--------------------------------

Cet attribut nous permettra d'insérer des commandes css dans le code html

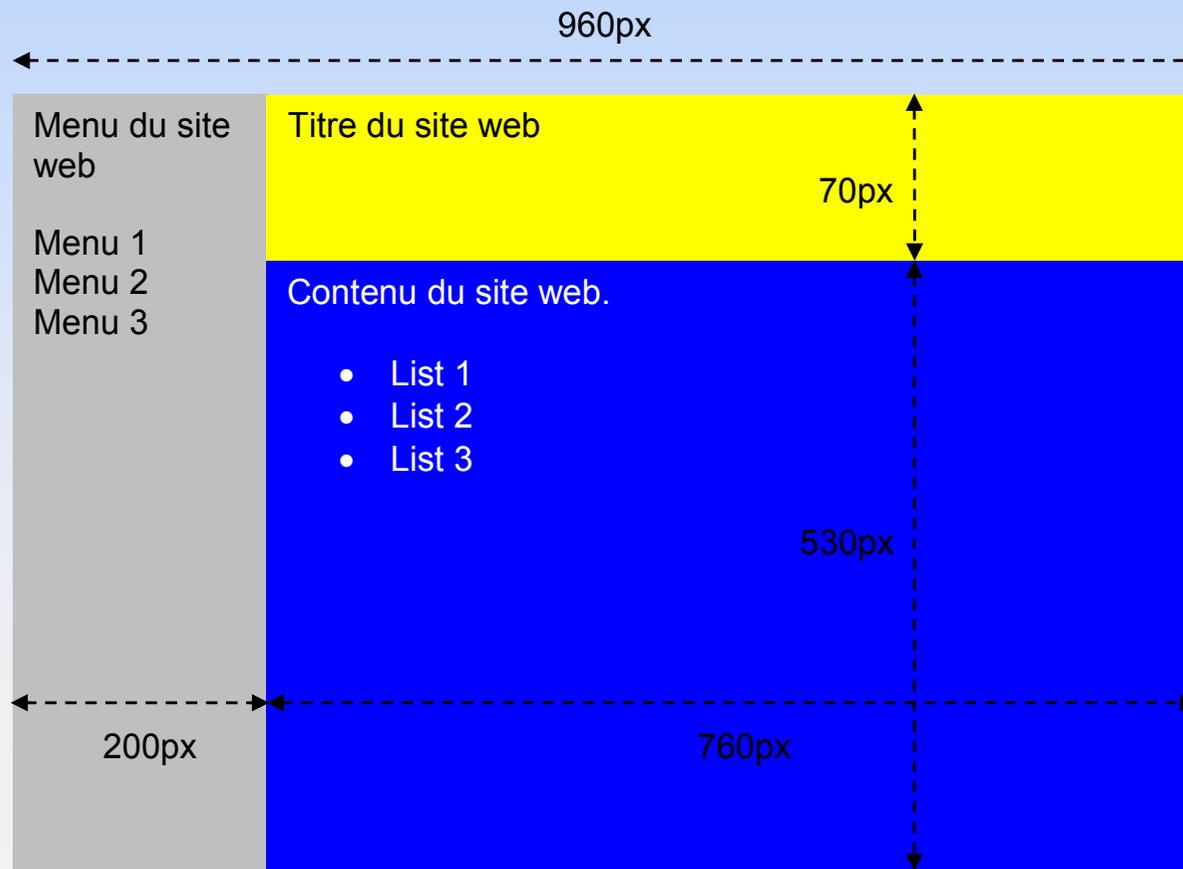
Exemple:

```
<div style=" width:150px; height:150px; border: 1px solid #FF0000;"></div>
```

6. Le concept de couches (layers)

6.2. Travaux pratiques

Créez une page html (page7.html) avec des layers comme l'illustration ci-dessous ! Les quatre layers sont définis dans une feuille de style (style7.css). Le quatrième layer sert à centrer le site web dans le browser. Les trois autres layers ont une marge intérieure de 5px.



7. L'association div + css

7.1. Relation entre layer et stylesheet

Comme pour le formatage classique, on peut définir tous les aspects du layer dans un fichier css. La relation entre le fichier html et le fichier css est la même.

Rappel:

Tag utilisé : `<link> </link>`

Exemple:

```
<link href="../CEPL/1030/Mon_site/css/layer.css" rel="stylesheet" type="text/css" />
```

Les zones dans le fichier html seront définies avec de identifiants (id)

7. L'association div + css

7.2. Travaux pratiques

Créez une page html (page_pomme.html) avec des layers comme l'illustration ci-dessous !

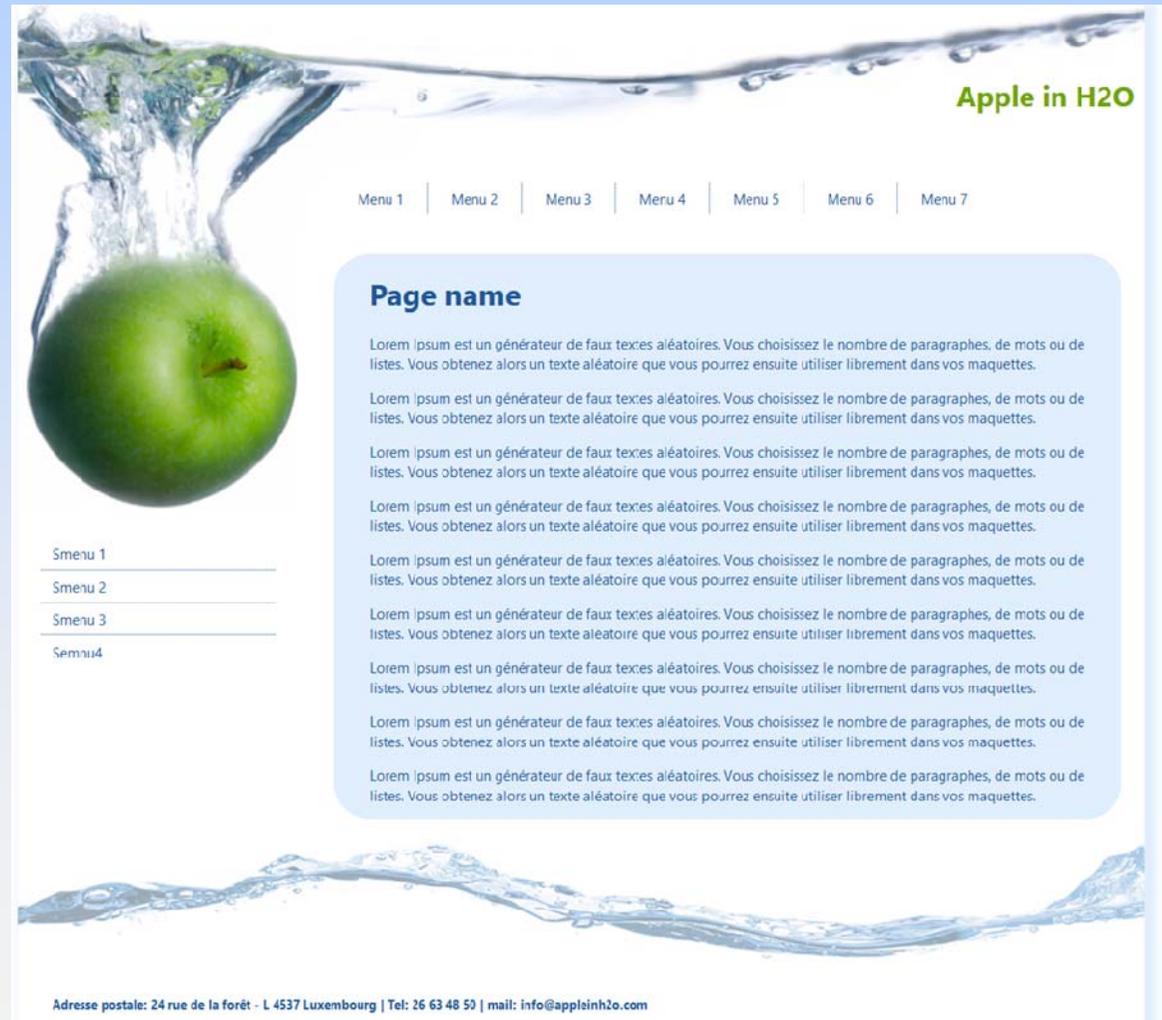
Tous les layers sont définis dans une feuille de style (style_pomme.css).

Vous trouverez les images dans le répertoire 'images_pomme'.

Le nom de l'image vous indiquera dans quel layer il faudra la placer:

- apple_water_header.png
- apple_water_menu.png
- apple_water_smenu.png
- apple_water_footer.png

Un layer sert à centrer le site web dans le browser !s



8. Les navigateurs

8.1. Les différents systèmes d'exploitation

Aujourd'hui il faut faire la différence entre un dispositif fixe (desktop, labtop, ...) et un dispositif mobile (tablet, smartphone, ...).

Systèmes d'exploitation des dispositifs fixes:

- MS Windows (85%)
- Mac OS (10%)
- Linux (5%)

Systèmes d'exploitation des dispositifs mobiles:

- Android (62%)
- iOP (26%)
- MS Windows (8%)

Voici deux sites web pour consulter des statistiques web:

- <http://www.w3schools.com/browsers/default.asp>
- <http://gs.statcounter.com>

8. Les navigateurs

8.2. Les différents navigateurs

Quels navigateurs sont utilisés dans les différents systèmes d'exploitations?

Dispositif fixe			Dispositif mobile		
MS Windows	Mac IOS	Linux	Android	iOS	MS Windows
IE 7, 8, 9, 10, 11 Firefox Chrome Safari Opera	Safari Firefox Chrome	Firefox Chrome	Firefox Chrome	Safari	IE

Dispositifs fixes:

- Chrome est de loin le navigateur le plus utilisé.
- Concernant Firefox est le logiciel du développeur web. Il existe une multitude de modules complémentaires qu'on peut installer gratuitement. Voici le nom de quelques modules pour le développement de pages web:
 - Firebug
 - Wappalyzer
 - Measureit
 - Colorzilla

Dispositifs mobiles:

- Les navigateurs natifs sont pratiquement utilisés exclusivement

8. Les navigateurs

8.3. Quels navigateurs faut-il tester

Comment tester les pages web développées?

Il existe deux possibilités.

1. Version peu onéreuse. Installer un PC avec les navigateurs suivants:

- Internet Explorer 7, 8, 9, 10, 11
- Firefox
- Chrome
- Safari

2. Version pro. Installer un PC MS Windows une machine avec Mac OS et un PC avec linux et tester les navigateurs suivants:

- MS Windows → Internet Explorer 7, 8, 9, 10, 11, + Firefox + Chrome
- Mac OS → Safari + Firefox + Chrome
- Linux → Firefox + Chrome