

## Table des matières

Mise en situation.....	3
Séquence 1 : la page d'accueil (balises de base).....	4
Objectifs.....	4
Moyens disponibles.....	4
Travail préliminaire.....	4
Contraintes.....	4
Structure.....	5
Travail demandé.....	7
Séquence 2 : la page produits (les tableaux).....	10
Objectifs.....	10
Structure.....	10
Travail demandé.....	11
Séquence 3 : la page contact (les formulaires).....	13
Objectifs.....	13
Structure.....	13
Contraintes.....	14
Travail demandé.....	15
Séquence 4 : la charte graphique (feuille de style CSS).....	16
Objectifs.....	16
Contraintes.....	16
Screenshot.....	17
Travail demandé.....	17
Séquence 5 : la page services (multimédia).....	18
Objectifs.....	18
Screenshot.....	18
Travail demandé.....	19
Annexe 1 : Mise en page des documents web.....	20
Les cadres (frames).....	20
Avantages.....	20
Inconvénients.....	21
Exemple.....	21
Les tableaux.....	22
Inconvénients.....	22
Les feuilles de style CSS (Cascading Style Sheets).....	23
Avantages.....	23
Syntaxe.....	23
Exemple.....	25
Liens pour en savoir plus.....	27
Annexe 2 : la balise object (multimédia).....	28
Introduction.....	28
L'inclusion générique : l'élément OBJECT.....	28
L'initialisation de l'objet : l'élément PARAM.....	28
Incorporer un document HTML.....	29
Incorporer un document pdf.....	29
Incorporer un document wav.....	29
Incorporer un document vidéo (mpg, avi, wmv et mov).....	29

Annexe 3 : La DTD HTML 4.01 Strict.....	31
Syntaxe DTD.....	31
L'élément HTML (racine du document).....	31
L'élément HEAD (en-tête du document).....	31
L'élément TITLE (titre du document).....	32
L'élément BODY (corps du document).....	32
L'élément P (paragraphe).....	32
L'élément BR (saut de ligne forcé).....	32
L'élément A.....	32
L'élément U (style de texte souligné) .....	33
Supplément : XHTML.....	33
Annexe 4 : Le service Web.....	34
Serveur Web (RFC HTTP 1945).....	34
Client Web (RFC HTTP 1945).....	34
Architecture Client/Serveur.....	34
Annexe 5 : quelques formats d'image.....	35
PNG (Portable Network Graphics).....	35
GIF ( Graphics Interchange Format).....	35
JPEG (JPEG File Interchange Format ou JFIF).....	35
SVG (Scalable Vector Graphics).....	35
Annexe 6 : quelques formats vidéos.....	36
MPEG (Moving Picture Experts Group).....	36
AVI (Audio Video Interleave).....	36
MOV (QuickTime).....	36
WMV (Windows Media Video).....	36
FLV (Flash Video).....	36

## Mise en situation

La société SFCV (Spécialiste Français n°1 des Caves à Vin) désire mettre en place un site web.

La planification de la réalisation du site sera la suivante :

### ◆ étape n°1 : réalisation des pages HTML

Cette étape a pour but de découvrir et de prendre en main le langage **HTML** qui est interprété côté client par le navigateur. L'objectif principal de cette étape est la réalisation des pages **statiques** du site : menu, affichage des résultats, formulaires pour une demande de catalogue, effectuer une recherche, etc ...

### ◆ étape n°2 : mise en place de l'interactivité côté client

Cette étape a pour but de découvrir et de prendre en main le langage **JavaScript** qui s'exécute côté client (donc par le navigateur). L'objectif principal de cette étape est d'assurer l'intégrité des données envoyées vers le serveur par les formulaires. En effet, il serait inutile de surcharger le serveur avec l'envoi de données manquantes, incomplètes ou incorrectes. Cette vérification peut donc se faire côté client par le langage **Javascript**. D'autre part, ce langage permet d'améliorer l'interactivité côté client.

### ◆ étape n°3 : réalisation des pages dynamiques

Cette étape a pour but de découvrir et de prendre en main le langage **PHP** qui s'exécute côté serveur. Les objectifs principaux de cette étape sont :

- de traiter les données reçues par les différents formulaires et de les insérer dans la base de données par exemple
- d'interroger la base de données et de fournir un résultat adapté aux requêtes clientes.

## Séquence 1 : la page d'accueil (balises de base)

### Objectifs

Cette première séquence a pour objectif de réaliser la première page HTML du site. Elle met en oeuvre :

- la structure des pages en utilisant quelques balises de base
- les liens vers les autres pages (menu)
- l'utilisation des listes
- l'insertion d'image pour le logo de l'entreprise

### Moyens disponibles

Un éditeur de texte pour la réalisation de la page HTML(sublimetext)

Un navigateur (**chrome** par exemple) pour visualiser des pages HTML réalisées

### Travail préliminaire

Créer un répertoire **cave** pour le stockage des fichiers du site. Dans le répertoire **cave**, créer un sous-répertoire **images** dans lequel on stockera toutes les images nécessaires pour le site. Les pages HTML porteront l'extension **.html**.

### Contraintes

A . Pour la réalisation des pages HTML de ce site, on utilisera la **HTML 5** en spécifiant la déclaration de type de document suivante :

```
<!DOCTYPE HTML >
```

B . Pour la réalisation des pages HTML de ce site, on utilisera l'encodage de caractères **UTF-8** (ou ISO-8859-15 ou éventuellement ISO-8859-1) :

```
<meta charset="UTF-8" />
```

*Remarques* : Il est important de comprendre qu'il faut obtenir une cohérence entre l'encodage utilisé pour créer le document (l'éditeur de texte) et celui pour l'interpréter (le navigateur) :

- Les outils d'édition (par exemple, un éditeur de texte) peuvent coder des documents HTML avec l'encodage de caractères de leur choix. Ces outils peuvent utiliser tout encodage commode lequel couvre la plupart des caractères contenus dans le document, pourvu que l'encodage soit correctement étiqueté. Les caractères occasionnels, qui ne sont pas contenus dans cet encodage, peuvent tout de même être représentés par des références de caractères (numérique ou entité). Le choix de l'encodage se fait généralement au moment de l'enregistrement du fichier.
- Le paramètre « charset » identifie un encodage de caractères, qui représente une méthode pour convertir une séquence d'octets en une séquence de caractères. Cette conversion s'intègre naturellement au système de l'activité du Web : les serveurs envoient des documents HTML aux agents utilisateurs sous la forme d'un flux d'octets, les agents utilisateurs les interprètent comme séquence de caractères.

Exemples de références de caractères :

- &lt; → < (permet d'éviter la confusion avec la balise ouvrante)
- &gt; → > (permet d'éviter la confusion avec la balise fermante)
- &copy; → ©

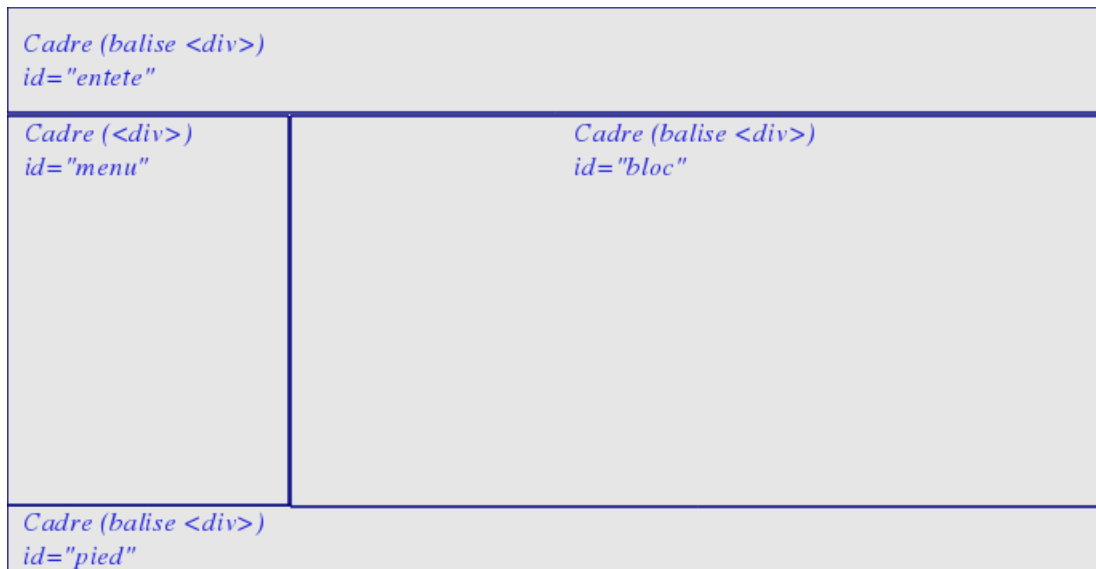
C . Respecter les noms de fichier indiqués dans le TP.

*Remarques* : dans tous les cas, ne pas utiliser d'espace, de majuscules et de caractères spéciaux (pas d'accents notamment).

## Structure

Le premier travail est de définir une structure à son document (le **fond**) et de ne pas se focaliser sur son apparence dans le navigateur (la **forme**). La mise en forme du document est une tâche qui sera réalisée plus tard (en séquence 4) notamment grâce à l'utilisation des feuilles de style et en respectant une charte graphique.

Dans cette première séquence, ce travail est important car les choix faits seront réutilisés dans les autres pages. En quelque sorte, cette première page va servir de *squelette* pour les autres. On va créer quatre divisions (balise `<div>`) pour le document :



Le squelette de la page **index.html** sera donc le suivant :

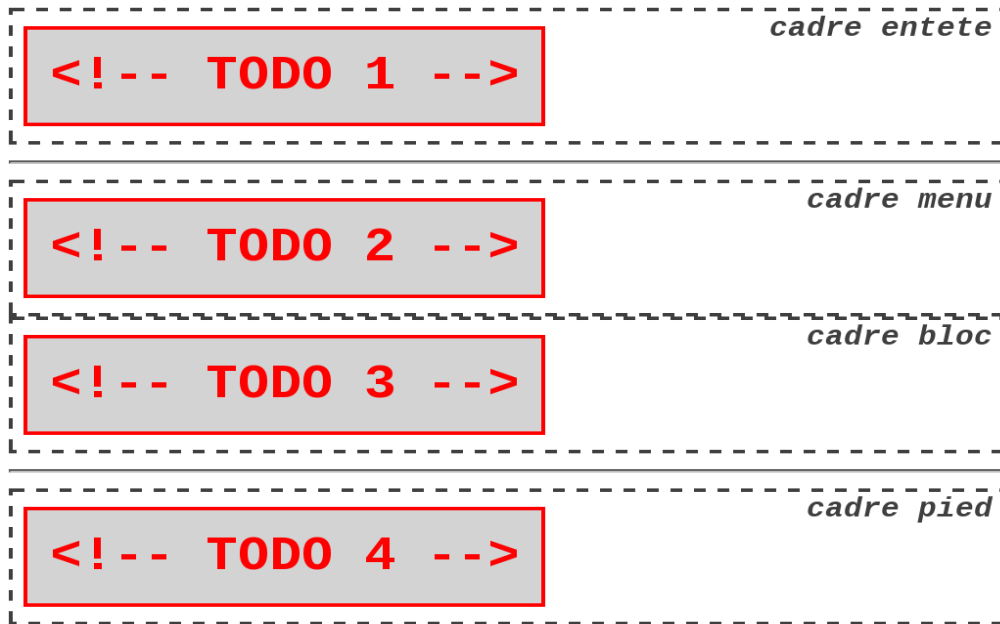
```
<!DOCTYPE HTML>

<head>
  <meta charset="UTF-8" />
  <title><!--TODO0--></title>
</head>
<body>
  <div id="entete"><!--TODO1--></div>
  <hr>
  <div id="menu"><!--TODO2--></div>
  <div id="bloc"><!--TODO3--></div>
  <hr>
  <div id="pied"><!--TODO4--></div>
</body>
</html>
```

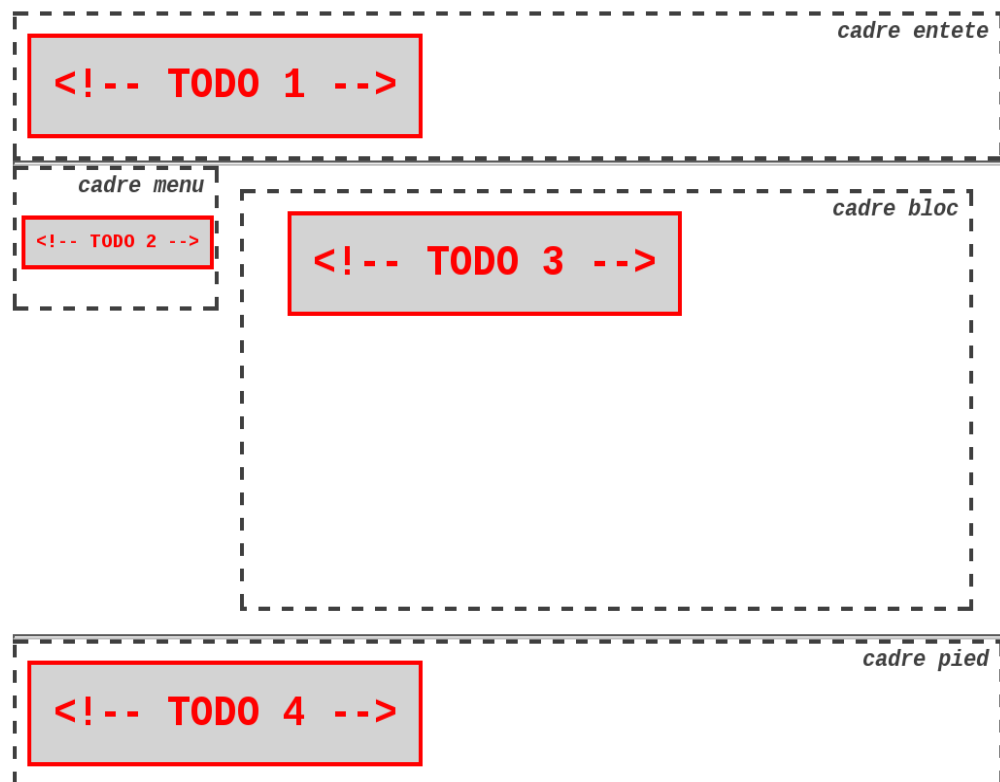
*Remarque* : Les balises `<!--TODOx-->` sont des **commentaires** en HTML. Ils indiquent ici les parties à réaliser (TODO) dans les questions du travail demandé.

**Lire l'annexe 1 sur la mise en page des sites web.**

Avant la séquence 4 et l'application de la feuille de style, on aura :



Après la séquence 4 et l'application de la feuille de style, on obtiendra :



## Travail demandé

Voici un aperçu des balises de base utilisée dans cette séquence :



cadre entete

– balise <img>

# SFCV

– balise <h1>

## N°1 des caves à vins

– balise <h3>

---

– balise <hr>

- balise <ul>
- balise <a> – balise <li>
- balise <a> – balise <li>
- balise <a> – balise <li>
- balise <a> – balise <li>

cadre menu

## Spécialiste français n°1 des caves à vin

– balise <h2>

Déclaration des droits de la bouteille

– balise <p>

- balise <ol>
- balise <li>
- balise <li>
- balise <li>
- balise <li>

Avec une gamme complète commercialisée par les plus grands distributeurs nationaux,

– balise <br>

notre société vous propose son savoir-faire incomparable pour concevoir des caves à vin de très haute qualité.

– balise <p>

## Liens

– balise <h2>

La passion du vin au féminin : [www.femivin.com](#)

– balise <a> – balise <p>

Vinosafe : [www.vinosafe.com](#)

– balise <a> – balise <p>

---

– balise <hr>

© SFCV - [Mentions légales](#)

– balise <a> – balise <p>

L'abus d'alcool est dangereux pour la santé, à consommer avec modération

– balise <p>

cadre pied

Ceci définit la **structure** de notre page web.

### 1. Titre (balise <title>)

→ Créer la page d'accueil **index.html** demandée (voir contraintes) en précisant le titre du document :

■ SFCV : Spécialiste Français n°1 des Caves à Vin

## 2 . Le bloc central : en-têtes (balises <h>), paragraphes (balises <p> et <br />) et listes (balises <ul>, <ol> et <li>)

On tiendra compte des indications suivantes :

- l'en-tête « Spécialiste français n°1 des caves à vin » est de type <h2>
- on utilisera des paragraphes (<p>) dont un utilise une liste numérotée (<ol>).

→ Modifier le document **index.html** (dans le cadre *bloc* <!--TODO3-->) qui reproduit la figure ci-dessous :

### **Spécialiste français n°1 des caves à vin**

Déclaration des droits de la bouteille

1. Article 1 : Quelle que soit l'origine, on a le droit au respect
2. Article 2 : Quelle que soit la couleur, on a le droit à un abri sûr
3. Article 3 : Quelle que soit l'âge, on a le droit à s'épanouir
4. Article 4 : Quelle que soit le lieu, on a le droit à s'intégrer

Avec une gamme complète commercialisée par les plus grands distributeurs nationaux, notre société vous propose son savoir-faire incomparable pour concevoir des caves à vin de très haute qualité.

**Attention :** vous avez tendance à « abuser » des balises <br> pour respecter la mise en page demandée. Ne faites pas ça ! Et conservez seulement les sauts de lignes nécessaires dans le texte d'un paragraphe. On vous rappelle que la présentation du document à l'écran sera réalisée à la fin par l'application d'un style (CSS). Pour l'instant, consacrez vous à la structure de vos pages plus qu'à leur visualisation.

## 3 . Les liens absolus et relatifs (balise <a>)

On tiendra compte des indications suivantes :

- l'en-tête « Liens » est de type <h2>
- on déclarera deux paragraphes pour chaque lien créé
- le lien "Mentions légales" pointerà sur un document **mentions.html**

→ Modifier le document **index.html** (dans le cadre *bloc* <!--TODO3--> et *pied* <!--TODO4-->) qui reproduit la figure ci-dessous :

### **Liens**

La passion du vin au féminin : [www.femivin.com](http://www.femivin.com)

Vinosafe : [www.vinosafe.com](http://www.vinosafe.com)

---

© SFCV - [Mentions légales](#)

L'abus d'alcool est dangereux pour la santé, à consommer avec modération

## 4 . Les images (balise <img>)

On tiendra compte des indications suivantes :

- l'en-tête pour le nom de la société sera de type <h1>
- on utilisera un autre en-tête de type <h3> pour " N°1 des ... "
- le chemin du fichier image est : **images/logo.png** (on ajoute un texte alternatif pour cette image)

→ Modifier le document **index.html** (dans le cadre *entete* <!--TODO1-->) qui reproduit la figure ci-dessous :



**SFCV**

**N°1 des caves à vins**



## 5. Le menu

On tiendra compte des indications suivantes pour créer le menu de gauche :

- on utilisera une liste non numérotée de type `<ul>` donc chaque élément de liste `<li>` contiendra un élément (lien) du menu
- le lien " Accueil " pointera sur un document **index.html**
- le lien " Produits " pointera sur un document **produits.html**
- le lien " Services " pointera sur un document **services.html**
- le lien " Contact " pointera sur un document **contact.html**

*Remarque :* ces choix correspondent au menu « classique » d'une société commerciale.

→ Modifier le document **index.html** (dans le cadre *menu* `<!--TODO2-->`) qui reproduit la figure ci-dessous :



## SFCV

N°1 des caves à vins

---

- [Accueil](#)
- [Produits](#)
- [Contact](#)

### Spécialiste français n°1 des caves à vin

Déclaration des droits de la bouteille

1. Article 1 : Quelle que soit l'origine, on a le droit au respect
2. Article 2 : Quelle que soit la couleur, on a le droit à un abri sûr
3. Article 3 : Quelle que soit l'âge, on a le droit à s'épanouir
4. Article 4 : Quelle que soit le lieu, on a le droit à s'intégrer

Avec une gamme complète commercialisée par les plus grands distributeurs nationaux, notre société vous propose son savoir-faire incomparable pour concevoir des caves à vin de très haute qualité.

### Liens

La passion du vin au féminin : [www.femivin.com](http://www.femivin.com)

Vinosafe : [www.vinosafe.com](http://www.vinosafe.com)

---

© SFCV - [Mentions légales](#)

L'abus d'alcool est dangereux pour la santé, à consommer avec modération

## 6. Validation

→ Vous devez valider la page **index.html** en utilisant le service en ligne du W3C : <http://validator.w3.org/check>. Puis, tester les liens externes.


## Séquence 2 : la page produits (les tableaux)

### Objectifs

Cette séquence a pour objectif de réaliser la page HTML qui permet de lister l'ensemble des produits commercialisés par la société SFCV. Elle met en oeuvre l'utilisation des **tableaux**. Pour naviguer plus facilement dans le document, on utilisera aussi des **liens internes** (avec des ancres).

### Structure

Évidemment, on retrouve la même structure, seul le *bloc* central droit change de contenu :





# SFCV

N°1 des caves à vins

- [Accueil](#)
- [Produits](#)
- [Contact](#)


## Caves de vieillissement

AV 175	Pratique, le thermomètre en façade et le nouveau registre de cave international pour tenir à la croix la gestion des bouteilles présentes. Confortable les clayettes collecteur réversibles à empreintes en bois massif.			
	Contenance	Système froid	Dimensions (hxlzp)	Porte
	170 (en 75cl)	Compresseur	144 x 82 x 87	Pleine
	Accessoires : 2 Clayettes réversibles et 1 Modulable (en option payante), Régulation électromécanique, Afficheur Digital de température			Noir

AV 205	Astucieux, sa lampe de lecture pour lire les étiquettes dans un environnement obscur.			
	Contenance	Système froid	Dimensions (hxlzp)	Porte
	180 (en 75cl)	Compresseur	128 x 70 x 88	Pleine
	Accessoires : Sécurité à clé, 1 Clayette coulissante (possibilité de supplémentaires en option payante), Régulation électronique, Dynamique DataDisplay			Noir

[Remonter](#)

## Caves polyvalentes

CV 182	Élégante, la poignée de porte en bois profilé.			
	Contenance	Système froid	Dimensions (hxlzp)	Porte
	170 (en 75cl)	Compresseur	125 x 70 x 87	Pleine
	Accessoires : Sécurité à clé, Clayettes bois			Brun laqué

Bientôt de nouvelles caves disponibles ...

© SFCV - [Mentions légales](#)

L'abus d'alcool est dangereux pour la santé, à consommer avec modération

On va utiliser des tableaux pour la présentation des produits de la société.

## Travail demandé

### 1. Titre (balise <title>)

→ Créer la page d'accueil **produits.html** demandée en précisant le titre du document :

SFCV : les caves à vin

→ Copier/Coller les cadres *entete*, *menu* et *pied* du fichier **index.html** réalisé à la séquence 1. Seul le cadre *bloc* sera personnalisé pour cette page.

### 2. Tableaux (balises <table>, <tr>, <th> et <td>)

a . Pour commencer simplement avec les tableaux, on va réaliser un document **tarifs.html** (qui servira plus tard à la séquence 5) qui reproduit la figure ci-dessous :


Modèle	Tarif
AV 175	<i>veuillez nous contacter</i>
AV 205	<i>veuillez nous contacter</i>
CV 183	<i>veuillez nous contacter</i>

*Remarque* : la page **tarifs.html** n'utilisera pas de cadre *entete*, *menu* et *pied*. Le tableau sera juste placé dans le corps de la page (balise body).

b . Pour la mise en page des produits, on tiendra compte des indications suivantes :

- les différentes catégories de caves seront de type <h2>
- on utilisera le fusionnement de cellules (colspan et rowspan) et on intégrera une image dans le tableau

→ Modifier le document **produits.html** qui reproduit la figure ci-dessous (images/AV175.jpg) :

Caves de vieillissement				
AV 175	Pratique, le thermomètre en façade et le nouveau registre de cave International pour tenir à la craie la gestion des bouteilles présentes. Confortable les clayettes collector réversibles à empreintes en bois massif.			
	Contenance	Système froid	Dimensions (hxlxp)	Porte
	178 (en 75cl)	Compresseur	144 x 62 x 67	Pleine
	Accessoires : 2 Clayettes réversibles et 1 Modulable (en option payante), Régulation électromécanique, Afficheur Digital de température			Noir

**Attention** : n'utilisez pas les balises <br> pour respecter la mise en page demandée. On vous rappelle que la présentation du document à l'écran sera réalisée à la fin par l'application d'un style (CSS). Pour l'instant, consacrez vous à la structure de vos pages plus qu'à leur visualisation.

Même chose pour ces deux autres produits :

- **AV 205** (images/AV205.jpg)
- Astucieux, sa lampe de lecture pour lire les étiquettes dans un environnement obscur.
- 196 (en 75cl) / Compresseur / 139 x 70 x 68 / Pleine / Noir
- Accessoires : Serrure à clé, 1 Clayette coulissante (possibilité de supplémentaires en option payante), Régulation électronique, Dynamique DataDisplay

et une cave polyvalente :

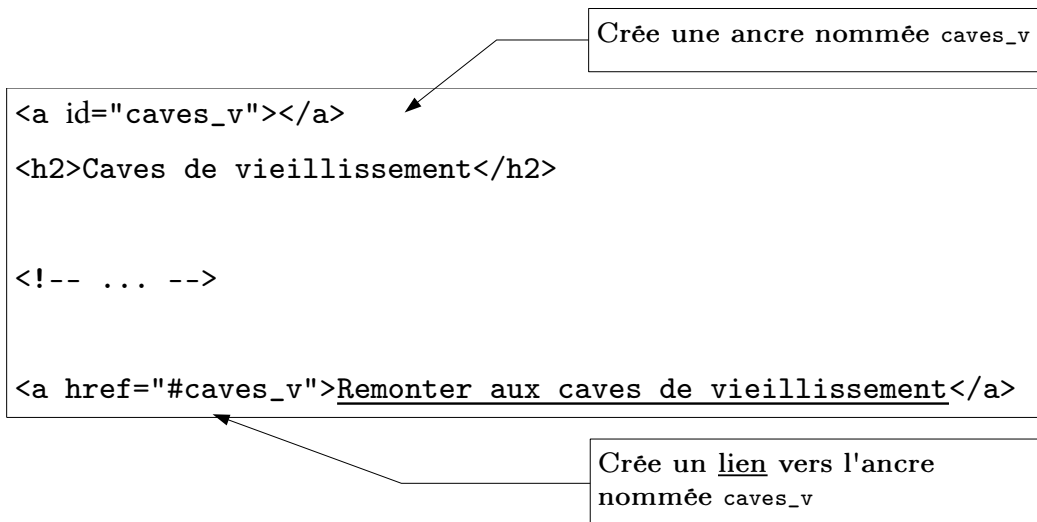
- **CV 183** (images/CV183.jpg)
- Élégante, la poignée de porte en bois profilé.
- 170 (en 75cl) / Compresseur / 125 x 70 x 67 / Pleine / Brun foncé
- Accessoires : Serrure à clé, Clayettes bois

### **3 . Les liens internes et les ancres (balise <a>)**

On réalisera les opérations suivantes :

- créer une ancre en haut du document
- placer un lien " Remonter " entre chaque catégorie et qui pointerait sur l'ancre

Exemple :



→ Modifier le document **produits.html** pour intégrer au moins deux liens internes.

### **4 . Validation et test**

→ Vous devez valider la page **produits.html** en utilisant le service en ligne du W3C :

<http://validator.w3.org/check>. Puis, tester les liens internes.


## Séquence 3 : la page contact (les formulaires)

### Objectifs

Cette séquence a pour objectif de réaliser la page HTML qui permet de demander un catalogue gratuit à la société SFCV. Elle met en oeuvre l'utilisation des **formulaires**.

### Structure

Une nouvelle fois, on retrouve la même structure, seul le *bloc* central droit change de contenu :



## SFCV

N°1 des caves à vins

---

- [Accueil](#)
- [Produits](#)
- [Contact](#)

### Demande de catalogue

Demandez-nous vite notre catalogue gratuit !  
Il vous sera livré dans les meilleurs délais.

Civilité

Nom

Prénom

Adresse

Ville  Code postal :

☒ je souhaite recevoir la newsletter - courriel :

Ce site a fait l'objet d'une déclaration auprès de la Commission Nationale Informatique et Libertés (CNIL). Conformément aux dispositions de la loi Informatiques et libertés du 6/01/1978, vous disposez d'un droit de modification, de rectification et d'opposition/suppression des données vous concernant, ce par simple demande à : [info@sfcv.com](mailto:info@sfcv.com)

N'hésitez pas à prendre contact avec l'équipe SFCV. Nous répondrons à toutes vos questions sur l'aménagement de votre cave ou le choix de votre armoire à vin. Tous les conseils de professionnels pour conserver vos meilleurs vins en toute sécurité pour le plaisir de les déguster en toute convivialité.

---

© SFCV - [Mentions légales](#)

L'abus d'alcool est dangereux pour la santé, à consommer avec modération

## Contraintes

### A. Les données du formulaire :

Les données du formulaire seront envoyées à un destinataire précisé par l'attribut **action** de la balise **<form>**. On peut indiquer, dans l'attribut **action**, soit une adresse email (action="<mailto:@email>") soit l'url d'une page (le plus souvent vers un CGI ou un script PHP).

Toutes les données des éléments du formulaire doivent être nommées (attributs **name** et **id**) afin d'être identifiables par le programme de traitement destinataire (notion de variables : le nom donné dans le formulaire correspondra au nom de la variable dans le programme serveur) ou même par le destinataire du *mail*. On n'acceptera pas les accents et on n'utilisera que les minuscules pour les attributs **name** et **id**.

D'autre part, le concepteur du formulaire doit souvent fixer les valeurs par défaut associées aux éléments (attribut **value**) :

- le formulaire : name="demande", id="demande"
- la civilité : name="civilité" – value="1" pour Monsieur, etc ...
- le nom : name="nom" – value="" (saisie de l'utilisateur)
- le prénom : name="prenom" – value="" (saisie de l'utilisateur)
- l'adresse : name="adresse" (saisie de l'utilisateur)
- la ville : name="ville" – value="" (saisie de l'utilisateur)
- le code postal : name="codepostal" – value="" (saisie de l'utilisateur)
- choix de recevoir la newsletter : name="newsletter" – value="oui"
- son adresse email : name="email" – value="" (saisie de l'utilisateur)
- l'envoi du formulaire : name="action" – value="Envoyer"

Remarque : l'utilisateur a la possibilité de donner son adresse *email* s'il veut recevoir la lettre d'information (*newsletter*)

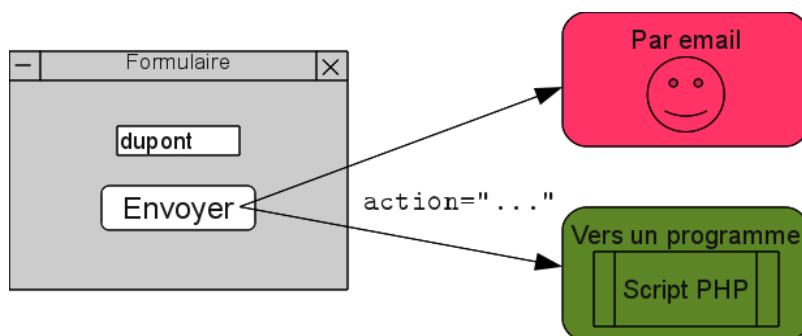
### B. L'envoi du formulaire :

En utilisant l'attribut **method**, on distingue deux méthodes (GET ou POST) d'envoi des données du formulaire :

- Avec method=**POST**, les données envoyées au serveur se trouvent dans le **corps de la requête HTTP**.
- Avec method=**GET**, les données envoyées au serveur se trouvent dans **l'URL** précisée dans l'attribut **action** (cela donnera par exemple *url?month=fevrier&day=24*).

Il faut aussi indiquer la manière avec laquelle les données du formulaire seront encodées en renseignant l'attribut **enctype** (*text/plain*, *multipart/form-data*, ...).

Il y a en général deux destinations d'envoi possibles (la plus utilisée étant celle vers un programme) :



## ***Travail demandé***

### **1 . Titre (balise <title>)**

→ Créer la page d'accueil **contact.html** demandée en précisant le titre du document :  
■ SFCV : Contact

### **2 . Formulaire (balises <form>, <input> et <textarea>)**

→ Réaliser la page demandée en respectant les contraintes énoncées ci-dessus.

### **3 . Validation et test**

→ Vous devez valider la page **contact.html** en utilisant le service en ligne du W3C :  
<http://validator.w3.org/check>.

## Séquence 4 : la charte graphique (feuille de style CSS)

### Objectifs

Cette séquence a pour objectif d'appliquer une charte graphique à l'ensemble des pages HTML réalisées. Elle met en oeuvre l'utilisation des feuilles de style CSS (*Cascading Style Sheets*). Le principe est simple : il s'agit de redéfinir le rôle d'une balise html en lui imposant de nouvelles propriétés.

### Contraintes

#### A. Fichier externe unique

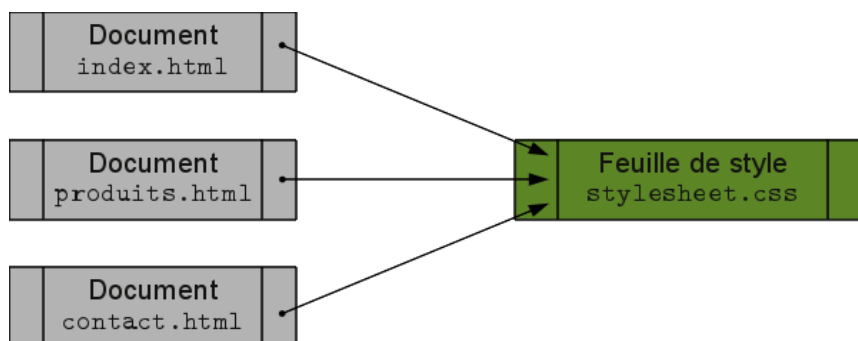
Les **feuilles de styles CSS** améliorent la qualité des mises en page sur le web (positionnement des éléments au pixel près), diminuent le temps d'affichage et factorisent le code, le rendant ainsi plus lisible et plus facile à maintenir. Le simple changement d'une feuille de styles permet de modifier la présentation du document. Dans le fichier CSS, il faut aussi faire la séparation entre structure et décoration :

- la **structure** contiendra le positionnement des éléments dans la (ou les) page(s) **ET**
- la **décoration** s'occupera notamment de la couleur et du graphisme

*Remarque* : en pratique cela pourrait donner deux fichiers CSS : un pour la structure, et un autre pour la présentation. Dans notre cas, on utilisera qu'un SEUL fichier dans lequel on placera de grandes barres de commentaires pour séparer les différentes parties (*/\* ... \*/*). Dans tous les cas, il est conseillé faire la séparation.

On utilisera donc un fichier externe unique **stylesheet.css** pour redéfinir le style des balises utilisées dans les pages HTML du site. Ce fichier CSS ne doit pas contenir de code HTML. Chaque page du site devra donc indiquer qu'elle utilise ce fichier CSS de la manière suivante :

```
<head>
  <title>Le titre de la page</title>
  <link rel="stylesheet" href="stylesheet.css" type="text/css">
</head>
```



#### B. La charte graphique

Les séquences précédentes nous ont amenés à définir une structure aux documents créés. Il reste à définir, grâce aux feuilles de style, l'interprétation visuel faite par le navigateur.

On découpera ce travail en deux parties :

- la structure visuelle des éléments : dimensions, positionnement (flottant, absolu) et leurs relations
- la décoration visuelle des éléments : police, couleur, ...

L'objectif est d'obtenir une cohérence (graphique) pour l'ensemble des pages et de faciliter les tâches de maintenance (modifications).



### B . Quelques éléments de structure à respecter :

- Taille et couleur de la police : 12px blanche (pour tout le document)
- Taille et couleur de la police : 22px grise (pour l'élément h1)
- Taille et couleur de la police : 16px blanche (pour l'élément h2)
- Taille et couleur de la police : 14px grise (pour l'élément h3)
- Taille et couleur de la police : 10px grise (pour l'élément h3 du bloc *entete*)
- Modification du texte : en majuscules (pour l'élément h3 du bloc *entete*)
- Taille de la bordure (gauche) : 2px (pour le bloc *bloc*)
- Type de la bordure (gauche) : solid (pour le bloc *bloc*)
- Couleur de la bordure (gauche) : grise (pour le bloc *bloc*)

→ Récupérer le fichier [stylesheet\\_todo.css](#) et le renommer en **stylesheet.css**.

## Screenshot

La page d'accueil **index.html** (un exemple de ce que l'on peut obtenir grâce à la structure définie avant) :



## Travail demandé

- 1 . Compléter le fichier **stylesheet.css** fourni en respectant les contraintes énoncées ci-dessus.
- 2 . Mettre au point sa feuille de style en utilisant la console d'erreurs du navigateur Firefox (Outils → Console d'erreurs)
- 3 . Valider sa feuille de style en utilisant l'outil de validation du W3C : <http://jigsaw.w3.org/css-validator/>
- 4 . Intégrer, dans toutes les pages du site, la feuille de style CSS créée.
- 5 . Pour terminer, réaliser la page **mentions.html**. Le contenu est disponible dans le fichier mentions.txt. Utiliser des ancres pour naviguer facilement dans cette page.

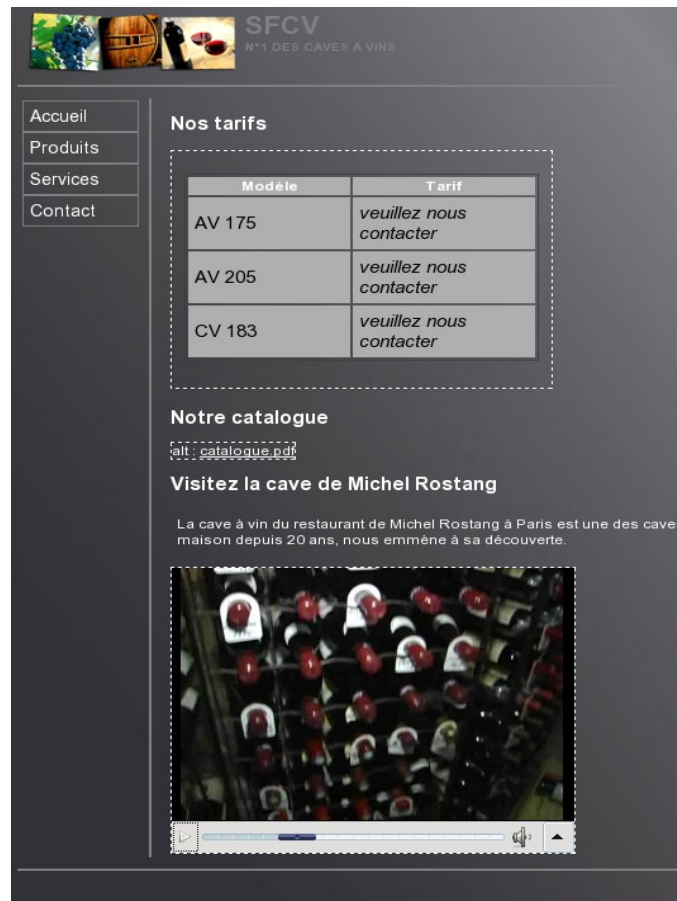
## Séquence 5 : la page services (multimédia)

### Objectifs

Cette séquence a pour objectif d'intégrer des ressources multimédias dans une page web. Elle met en oeuvre l'utilisation de la balise **object** (cf. Annexe 2).

### Screenshot

La page d'accueil **services.html** :



Lire l'annexe 2 sur la balise **object** et les exemples fournis.

## **Travail demandé**

### **1 . Titre (balise <title>)**

→ Créer la page d'accueil **services.html** demandée en précisant le titre du document :  
■ SFCV : les services

### **2 . Contenu multimédia (balise <object>)**

→ Réaliser la page demandée (voir *screenshot*).

La page (le cadre *bloc*) contiendra trois parties :

- Nos tarifs (titre h2) : incorporation dans la page du fichier **tarifs.html** (réalisé à la séquence 2)
- Notre catalogue (titre h2) : incorporation du catalogue au format PDF (**catalogue.pdf**)
- Visite d'une cave (titre h2) : incorporation d'une vidéo au format AVI (**visite.avi**)

### **3 . Validation et test**

→ Vous devez valider la page **services.html** en utilisant le service en ligne du W3C :  
<http://validator.w3.org/check>.

→ Tester la visualisation de la vidéo.

**Il est conseillé de lire l'annexe 5 sur les formats d'image et l'annexe 6 sur les formats vidéo.**

## **TRAVAIL FINAL**

**Retouchez le fichier CSS pour personnaliser votre site**

## Annexe 1 : Mise en page des documents web

Tout d'abord, il faut savoir que le langage HTML d'origine ne prévoyait pas véritablement d'éléments pour assurer sa mise en page. Donc différentes techniques utilisées par les *webmasters* se sont succédées :

- les cadres (*frames*)
- les tableaux
- et maintenant les **feuilles de styles CSS** (et la balise conteneur `<div>`)

### Les cadres (*frames*)

Les cadres découpent la fenêtre principale en autant de petits cadres, chacun d'eux jouant le même rôle qu'une fenêtre : un document HTML propre avec barres de défilement indépendantes...

Ces cadres peuvent être chargés de manière totalement indépendante : un cadre sera modifié et l'autre maintenu. Ils peuvent interargir entre eux : un choix dans un premier cadre pourra provoquer le chargement d'une nouvelle page dans un deuxième.

Les cadres permettent donc de créer rapidement une mise en page simple pour un site. L'utilisation classique consiste ainsi à présenter sur un cadre à gauche une liste de liens vers les différentes parties d'un site, qui s'affichent dans la fenêtre principale au centre. Cela facilite de plus la maintenance des différentes pages.

Cependant, les moteurs de recherche indexent les pages à l'intérieur des cadres autant que les autres. Il y a donc un risque pour qu'une page indexée de la sorte soit "orpheline", et ne donne pas accès au reste du site. Plus généralement, une telle page ne se suffit souvent pas à elle-même (il y manque parfois les outils de navigation à l'intérieur du site). Enfin, la présence de cadres s'accompagne souvent de celles de barres de défilement horizontales.

Il faut savoir qu'au début des *frames* sur le Web, leur utilisation posait toutes sortes de problèmes. Par exemple, il n'existait aucun moyen de masquer les bordures séparant les différentes zones dans la fenêtre du navigateur, ce qui fait que l'écran était rempli de traits gris et de barres de défilement. Dans la mesure où la place est limitée sur une page Web, on comprend que l'utilisateur ne se réjouisse pas de voir la fenêtre du navigateur ainsi remplie. C'est l'une des raisons pour lesquelles les *frames* ont (eu) mauvaise réputation. Aujourd'hui pourtant, il est possible de rendre invisibles les bordures entre *frames*.

**Remarque:** les responsables de sites ont constaté que la plupart des internautes choisissent la version sans *frame* lorsqu'elle est proposée.

### Avantages

Pour le *webmaster*: gestion et maintenance des pages plus simple et plus rapide

Pour l'internaute: sommaire toujours présent à l'écran, permettant de naviguer plus rapidement d'une page à l'autre sans revenir en arrière.

## Inconvénients

Pour le *webmaster*:

- ♦ Pages orphelines (chaque page HTML composant un *frameset* peut être appelée directement par son adresse URL, indépendamment de son environnement *frameset*).
- ♦ Se retrouver avec des *frames* dans les *frames* (un site à base de *frames* peut appeler un autre site à base de *frames* etc ...)

Pour l'internaute:

- ♦ Sauvegarde ou impression d'une page problématique
- ♦ Non accessibilité à l'information pour les aveugles et mal-voyants.  
(<http://www.braillenet.jussieu.fr/education/livreblanc/>)
- ♦ Espace réduit pour le contenu
- ♦ Complexité de la navigation: les *frames* peuvent constituer un véritable labyrinthe, notamment quand le *webmaster* a concocté un *frameset* encapsulant des sites externes, ou des sommaires en cascades.

## Exemple

```
<HTML>
  <HEAD>
    <TITLE>Un document avec deux cadres verticaux</TITLE>
  </HEAD>
  <FRAMESET cols="25%,75%" frameborder="0">
    <FRAME src="menu.html" name="menu">
    <FRAME src="accueil.html" name="principal">
  </FRAMESET>
</HTML>
```

## **Les tableaux**

Originellement prévus dans le but de mettre en forme des données, les tableaux ont rapidement été détournés par les développeurs web pour combler un vide qui faisait cruellement défaut au HTML à l'époque : la mise en page.

En effet, dans ce qu'il convient d'appeler la préhistoire du web, point de CSS ni de séparation entre le contenu et la présentation. Les développeurs pensaient avoir trouvé la parade avec les tableaux, et ont donc commencé à utiliser massivement ces derniers pour du positionnement et autres mises en page complexes.

Les tableaux sont franchement inadaptés aux exigences d'aujourd'hui, à cause de leurs limites notamment en termes d'accessibilité, de facilité d'utilisation et de maintenance.

**Remarque :** il est important de noter que les tableaux remplissent parfaitement leur rôle quand il s'agit de structurer des données.

## **Inconvénients**

- ◆ Lenteur d'affichage : un tableau est un élément complexe à traiter, notamment pour définir les tailles de chaque cellule et ligne, et nécessite d'analyser toutes les cellules ... surtout si les tableaux sont imbriqués
- ◆ Page illisible pour les navigateurs non-graphique (ne pas oublier l'existence de navigateurs spécialisés pour les handicapés, avec plage braille ou synthétiseur vocal, qui ne peuvent pas restituer les informations graphiques)
- ◆ Impression problématique : changer complètement le *design* d'une page pour permettre une meilleure impression est impossible ce qui amène souvent à faire une page séparée pour l'impression, et avec les coûts associés à sa création et à sa maintenance, et cela se transforme en un gâchis d'énergie et de ressources considérables.
- ◆ Complexe et coûteux à produire (même avec des éditeurs WYSIWYG)
- ◆ Lourdeur des pages : beaucoup de balises pour peu de contenu !

## Les feuilles de style CSS (Cascading Style Sheets)

Les feuilles de style CSS permettent la mise en forme du contenu des pages Web.

### Avantages

Les **feuilles de styles CSS** améliorent la qualité des mises en page sur le web (positionnement des éléments au pixel près), diminuent le temps d'affichage et factorisent le code, le rendant ainsi plus lisible et plus facile à maintenir. Le simple changement d'une feuille de styles permet de modifier la présentation du document. Une seule feuille de style permet d'appliquer la présentation d'un site web.

### Syntaxe

Cet exemple basique devrait vous permettre de comprendre l'utilisation des feuilles de style.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta content="text/html; charset=UTF-8" http-equiv="content-type">
    <title>Exemples</title>
    <link rel="stylesheet" href="style.css" type="text/css">
  </head>
  <body>
    <h3>Application d'un style global aux balises body, h1, h2 et p :</h3>
    <div><code>body { color: #4B0082; }<br>
    h1 { font-family: Arial; font-size: 22px; color: blue; }<br>
    h2 { font-family: Arial; font-size: 16px; color: green; }<br>
    p { font-family: Arial; font-size: 12px; color: red; }<br>
    </code></div>
    <h1>&lt;h1&gt;Je suis un titre de niveau 1&lt;/h1&gt;</h1>
    <h2>&lt;h2&gt;Je suis un titre de niveau 2&lt;/h2&gt;</h2>
    <p>&lt;p&gt;Je suis un paragraphe&lt;/p&gt;</p>
    <h1>&lt;h1&gt;Je suis un autre titre de niveau 1&lt;/h1&gt;</h1>
    <hr>
    <h3>Application d'un style individuel (id) aux balises h1, h2 et p :</h3>
    <div><code>
    #id_h1 { font-family: Courier; color: black; } /* on ne touche pas à la taille */<br>
    #id_h2 { font-family: Courier; color: black; }<br>
    #id_p { font-family: Courier; color: black; }<br>
    </code></div>
    <h1 id="id_h1">&lt;h1 id="id_h1"&gt;Je suis un titre de niveau 1 (avec un id)
&lt;/h1&gt;</h1>
    <h2 id="id_h2">&lt;h2 id="id_h2"&gt;Je suis un titre de niveau 2 (avec un id)
&lt;/h2&gt;</h2>
    <p id="id_p">&lt;p id="id_p"&gt;Je suis un paragraphe (avec un id)&lt;/p&gt;</p>
    <h1>&lt;h1&gt;Je suis un autre titre de niveau 1 (sans id)&lt;/h1&gt;</h1>
    <div><em>Remarque : h1, h2 et p ont hérité de la taille (font-size) définie globalement pour ces
balises.<br>Attention, l'id doit être défini et utilisé qu'une fois, car un id(entifier) est toujours
unique.</em></div>
    <hr>
    <h3>Application d'une classe de style (class) commune aux balises h1, h2 et p :</h3>
    <div><code>
    .my_class { font-family: Tahoma; font-size: 16px; } /* on ne touche pas à la couleur */<br>
    </code></div>
    <h1 class="my_class">&lt;h1 class="my_class"&gt;Je suis un titre de niveau 1 (avec
my_class)&lt;/h1&gt;</h1>
    <h2 class="my_class">&lt;h2 class="my_class"&gt;Je suis un titre de niveau 2 (avec
my_class)&lt;/h2&gt;</h2>
    <p class="my_class">&lt;p class="my_class"&gt;Je suis un paragraphe (avec my_class)
&lt;/p&gt;</p>
    <h1>&lt;h1&gt;Je suis un autre titre de niveau 1 (sans my_class)&lt;/h1&gt;</h1>
    <div><em>Remarque : h1, h2 et p ont hérité de leur couleur (color) définie globalement pour ces
balises.</em></div>
    <hr>
  </body>
</html>
```

La feuille de style **style.css** :

```
/* toutes les balises body, h1, h2 et p auront ce style */
body { color: #4B0082; }
h1 { font-family: Arial; font-size: 22px; color: blue; }
h2 { font-family: Arial; font-size: 16px; color: green; }
p { font-family: Arial; font-size: 12px; color: red; }
code { font-family: Courier; }

/* les balises identifient leur style individuel et unique (id → #) */
#id_h1 { font-family: Courier; color: black; } /* on ne touche pas à la taille */
#id_h2 { font-family: Courier; color: black; }
#id_p { font-family: Courier; color: black; }

/* pour toutes les balises utilisant cette classe de style (class → .) */
.my_class { font-family: Tahoma; font-size: 16px; } /*on ne touche pas à la couleur*/
```

Ce qui donne :

Application d'un style global aux balises body, h1, h2 et p :

```
body { color: #4B0082; }
h1 { font-family: Arial; font-size: 22px; color: blue; }
h2 { font-family: Arial; font-size: 16px; color: green; }
p { font-family: Arial; font-size: 12px; color: red; }
```

**<h1>Je suis un titre de niveau 1</h1>**

**<h2>Je suis un titre de niveau 2</h2>**

**<p>Je suis un paragraphe</p>**

**<h1>Je suis un autre titre de niveau 1</h1>**

---

Application d'un style individuel (id) aux balises h1, h2 et p :

```
#id_h1 { font-family: Courier; color: black; } /* on ne touche pas à la taille */
#id_h2 { font-family: Courier; color: black; }
#id_p { font-family: Courier; color: black; }
```

**<h1 id="id\_h1">Je suis un titre de niveau 1 (avec un id)</h1>**

**<h2 id="id\_h2">Je suis un titre de niveau 2 (avec un id)</h2>**

**<p id="id\_p">Je suis un paragraphe (avec un id)</p>**

**<h1>Je suis un autre titre de niveau 1 (sans id)</h1>**

*Remarque : h1, h2 et p ont hérité de la taille (font-size) définie globalement pour ces balises.  
Attention, l'id doit être défini et utilisé qu'une fois, car un id(entifier) est toujours unique.*

---

Application d'une classe de style (class) commune aux balises h1, h2 et p :

```
.my_class { font-family: Tahoma; font-size: 16px; } /* on ne touche pas à la couleur */
```

**<h1 class="my\_class">Je suis un titre de niveau 1 (avec my\_class)</h1>**

**<h2 class="my\_class">Je suis un titre de niveau 2 (avec my\_class)</h2>**

**<p class="my\_class">Je suis un paragraphe (avec my\_class)</p>**

**<h1>Je suis un autre titre de niveau 1 (sans my\_class)</h1>**

*Remarque : h1, h2 et p ont hérité de leur couleur (color) définie globalement pour ces balises.*

---



## Exemple

```
<html>
<head>
  <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type">
  <title>Exemple - CSS</title>
  <link rel="stylesheet" href="./style.css" type="text/css">
</head>
<body>
  <div id="header">
    L'en-t&ecirc;te ...
  </div>

  <div id="left">
    Le c&ocirc;t&eacute; gauche ...
  </div>

  <div id="right">
    Le c&ocirc;t&eacute; droit ...
  </div>

  <div id="middle">
    Le milieu ...
  </div>

  <div id="footer">
    Le pied de page ...
  </div>
</body>
</html>
```

La feuille de style **style.css** :

```
/* Structure */

body
{
  margin: 10px;
  padding: 0px;
}

div#header
{
  margin: 0px;
  padding: 1%;
  clear: both;
  height: 5%;
}

div#left
{
  margin: 0px;
  padding: 1%;
  float: left;
  width: 20%;
  height: auto;
}
```

```
div#right
{
    margin: 0px;
    padding: 1%;
    float: right;
    height: 80%;
    width: 10%;
}

div#middle
{
    float: left;
    width: 60%;
    margin: 1%;
    padding: 1%;
}

div#footer
{
    margin: 0px;
    padding: 1%;
    height: auto;
    clear: both;
}

/* Décoration */

div#header
{
    border: 2px dashed #FF8822;
    color: #FF8822;
}

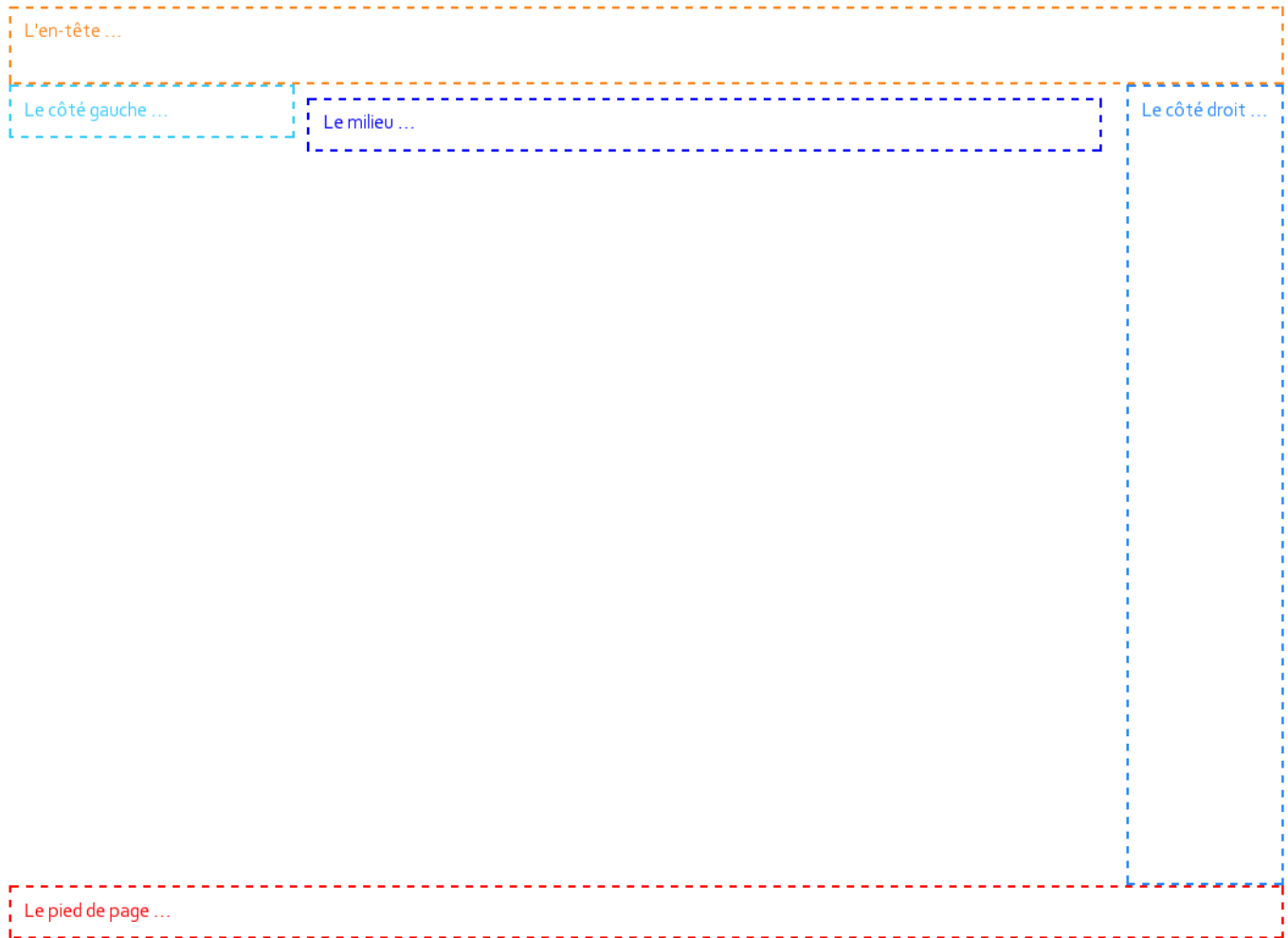
div#left
{
    border: 2px dashed rgb(51, 204, 255);
    color: rgb(51, 204, 255);
}

div#right
{
    border: 2px dashed #2288FF;
    color: #2288FF;
}

div#middle
{
    border: 2px dashed blue;
    color: blue;
}

div#footer
{
    border: 2px dashed red;
    color: red;
}
```

On obtient :



### ***Liens pour en savoir plus***

Spécification DOM: <http://xmlfr.org/w3c/TR/REC-DOM-Level-1/>

Spécification HTML 4: <http://www.la-grange.net/w3c/html4.01/>

Spécification XHTML: <http://www.la-grange.net/w3c/xhtml1/>

Spécification CSS2: <http://www.yoyodesign.org/doc/w3c/css2/cover.html>

Les standards du Web: <http://www.openweb.eu.org/>

CSS pratique: <http://pompage.net/pompe/csspratique/>

## Annexe 2 : la balise object (multimédia)

### Introduction

Les fonctionnalités multimédias de HTML permettent aux auteurs d'inclure dans leurs pages des images, des applets (des programmes qui sont automatiquement chargés puis lancés sur la machine de l'utilisateur), des séquences vidéo et d'autres documents HTML.

HTML 4 introduit l'élément **OBJECT**, qui offre une solution générale aux inclusions d'objets génériques. L'élément OBJECT permet aux auteurs HTML de spécifier tout ce que l'objet requiert pour sa présentation par un agent utilisateur : le code source, les valeurs initiales et les données d'exécution.

Le nouvel élément OBJECT prend donc en charge quelques unes des tâches effectuées par les éléments existants. Considérons le comparatif des fonctionnalités suivant :

Type d'inclusion	Élément spécifique	Élément générique
Image	IMG	OBJECT
Applet	APPLET (déconseillé)	OBJECT
Un autre document HTML	IFRAME	OBJECT

Par exemple, pour inclure des images, les auteurs peuvent utiliser l'élément OBJECT ou bien l'élément IMG.

Le comparatif indique que chaque type d'inclusion possède une solution spécifique et une solution générique. L'élément générique OBJECT servira de solution pour l'implémentation des types de média futurs.

### L'inclusion générique : l'élément OBJECT

Définition des attributs :

<b>classid</b>	-- identifie l'implémentation --
<b>codebase</b>	-- URI de base pour classid, data, archive --
<b>data</b>	-- référence aux données de l'objet --
<b>type</b>	-- type de contenu des données --
<b>codetype</b>	-- type de contenu pour code code --
<b>archive</b>	-- liste d'URI séparés par des espaces --
<b>standby</b>	-- message à afficher pendant le chargement --
<b>height</b>	-- surclasse la hauteur --
<b>width</b>	-- surclasse la largeur --
<b>usemap</b>	-- utiliser une image cliquable côté client --
<b>name</b>	-- soumettre comme partie d'un formulaire --
<b>tabindex</b>	-- position dans l'ordre de tabulation --

### L'initialisation de l'objet : l'élément PARAM

Les éléments PARAM spécifient l'ensemble des valeurs qui peuvent être requises à l'exécution par un objet. Il peut y avoir un nombre quelconque d'éléments PARAM dans le contenu d'un élément OBJECT, ou APPLET, dans n'importe quel ordre, mais ceux-ci doivent se placer au début du contenu de l'élément englobant OBJECT, ou APPLET.

Définition des attributs :

id	-- identifiant unique dans le document --
name	-- nom de propriété --
value	-- valeur de propriété --
valuetype	-- comment interpréter la valeur --
type	-- type de contenu pour la valeur quand valuetype="ref"

Extrait de la spécification :

<http://www.la-grange.net/w3c/html4.01/struct/objects.html>

## ***Incorporer un document HTML***

```
<object data="datas/test.html" type="text/html" width="300" height="200">
  alt : <a href="datas/test.html">test.html</a>
</object>
```

## ***Incorporer un document pdf***

```
<object data="datas/test.pdf" type="application/pdf" width="300" height="200">
  alt : <a href="datas/test.pdf">test.pdf</a>
</object>
```

## ***Incorporer un document wav***

```
<object type="audio/x-wav" data="datas/test.wav" width="200" height="20">
  <param name="src" value="datas/test.wav">
  <param name="autoplay" value="false">
  <param name="autoStart" value="0">
  alt : <a href="datas/test.wav">test.wav</a>
</object>
```

*Remarques :*

Vous pouvez spécifier des paramètres relatifs au document par l'intermédiaire de la balise param. Internet Explorer a parfois besoin du paramètre src pour localiser correctement la source.

Le paramètre "autoplay" est compréhensible par QuickTime, "autoStart" par Windows Media Player et Real Audio.

## ***Incorporer un document vidéo (mpg, avi, wmv et mov)***

```
<object data="datas/test.mpg" type="video/mpeg" width="320" height="255">
  <param name="src" value="datas/test.mpg">
  <param name="autoplay" value="false">
  <param name="autoStart" value="0">
  alt : <a href="datas/test.mpg">test.mpg</a>
</object>
```

```
<object data="datas/test.avi" type="video/x-msvideo" width="320" height="255">
  <param name="src" value="datas/test.avi">
  <param name="autoStart" value="0">
  alt : <a href="datas/test.avi">test.avi</a>
</object>
```

```
<object type="video/x-ms-wmv" data="datas/test.wmv" width="320" height="255">
  <param name="src" value="datas/test.wmv">
  <param name="autoStart" value="0">
  alt : <a href="datas/test.wmv">test.wmv</a>
</object>
```

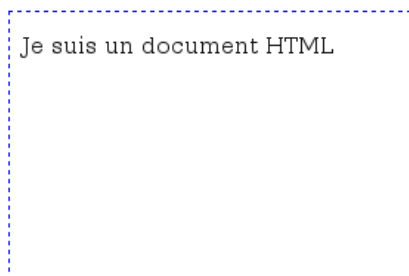
```
<object classid="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B"
  codebase="http://www.apple.com/qtactivex/qtplugin.cab"
  width="320" height="255">
  <param name="src" value="datas/test.mov" >
  <param name="controller" value="true" >
  <param name="autoplay" value="false">
  <object type="video/quicktime" data="datas/test.mov" width="320" height="255">
    <param name="controller" value="true" >
    <param name="autoplay" value="false">
    alt : <a href="datas/test.mov">test.mov</a>
  </object>
</object>
```

Ces exemples sont extraits du site : <http://joliclic.free.fr/html/object-tag/>

Ces exemples sont aussi disponibles sur : <http://tvaira.free.fr/web/src/objects.html>

## Exemples d'utilisation de la balise <OBJECT>

Incorporer un document HTML :



Incorporer un document pdf :

alt : [tpHTML.pdf](#)

Incorporer un document wav :



Incorporer un document vidéo (mpg, avi, wmv et mov) :



## Annexe 3 : La DTD HTML 4.01 Strict

Le DTD HTML 4.01 Strict **exclut** les attributs et les éléments de présentation que le W3C entend supprimer graduellement avec la gestion croissante des feuilles de style. Les définitions des éléments et attributs signalent clairement ceux qui sont déconseillés. Les auteurs devraient employer le DTD strict autant que possible, mais ils peuvent employer le DTD transitoire si la reconnaissance des attributs et éléments de présentation est exigée. En général, les auteurs devraient utiliser les feuilles de style, pour obtenir des effets de style et de mise en forme, plutôt que les attributs de présentation HTML. Les attributs de présentation HTML sont déconseillés quand il existe une alternative par les feuilles de style (CSS).

### Syntaxe DTD

Opérateur	Signification	Exemple
	L'élément A est obligatoirement présent	A
+	L'élément doit être présent au minimum une fois	A+
*	L'élément peut être présent plusieurs fois (ou aucune)	A*
?	L'élément peut être optionnellement présent	A?
	L'élément A ou l'élément B peuvent être présents	A B
,	L'élément A doit être présent et suivi de l'élément B	A,B
()	Les parenthèses permettent de regrouper des éléments afin de leur appliquer les autres opérateurs	(A,B)+

#### Balise ouvrante et fermante :

- - : Les deux tirets entre le nom de l'élément et la première parenthèse '(' indiquent que l'élément doit obligatoirement avoir une balise d'ouverture ET une balise de fermeture.

- O : Un tiret et un O (" - O "), la balise de fermeture n'est pas obligatoire (optionnelle)

O O : Deux O à la place des deux tirets (" O O "), les deux balises ne sont pas obligatoires (optionnelles).

#### Exemple :

<!ELEMENT TR - O (TH|TD)+ -- rangée de table →

Ce qui donne : Balise ouvrante : obligatoire, balise fermante : optionnelle

L'élément TR se comporte comme un conteneur pour une rangée de cellules de la table. La balise fermante peut être omise. L'élément TR devra contenir au moins un (ou plusieurs) élément(s) TH (cellule de rubrique d'une table) ou TD (cellule de données d'une table).

### L'élément HTML (racine du document)

<!ELEMENT HTML O O (HEAD, BODY) -- élément racine du document -->

### L'élément HEAD (en-tête du document)

Balise ouvrante : optionnelle, balise fermante : optionnelle

L'élément HEAD contient des informations sur le document courant, tels que son titre, des mots-clés que les moteurs de recherche peuvent exploiter et d'autres données qui ne sont pas considérées comme faisant partie du contenu du document.

## ***L'élément TITLE (titre du document)***

Balise ouvrante : obligatoire, balise fermante : obligatoire

Chaque document HTML doit avoir un élément TITLE dans la section HEAD. Il faut exactement un seul titre par document. L'élément TITLE n'est pas considéré faire partie du flux de texte. Il devrait, par exemple, être affiché comme titre de la page ou de la fenêtre.

## ***L'élément BODY (corps du document)***

Balise ouvrante : optionnelle, balise fermante : optionnelle

Le corps du document représente le contenu du document. Le contenu peut être présenté par un agent utilisateur de diverses façons. Par exemple, pour les navigateur visuels, on peut assimiler le corps à un canevas dans lequel s'inscrit le contenu : le texte, les images, les couleurs, les graphiques, etc.

Voici un extrait de la déclaration de l'élément BODY dans la DTD Strict :

```
<!ELEMENT BODY O O (P|H1|H2|H3|H4|H5|H6|UL|OL|PRE|DL|DIV|NOSCRIPT|BLOCKQUOTE|FORM|HR|TABLE|FIELDSET|ADDRESS|SCRIPT)+ -- corps du document -->
```

## ***L'élément P (paragraphe)***

Balise ouvrante : obligatoire, balise fermante : optionnelle

L'élément P correspond à un paragraphe. Il ne peut pas contenir d'éléments de type bloc (même pas un autre élément P). On déconseille vivement aux auteurs d'utiliser des éléments P vides. Les agents utilisateurs devraient ignorer de tels éléments P vides.

## ***L'élément BR (saut de ligne forcé)***

Balise ouvrante : obligatoire, balise fermante : interdite

L'élément BR coupe (interrompt) la ligne de texte courante.

## ***L'élément A***

Balise ouvrante : obligatoire, balise fermante : obligatoire

Remarque : Aucun attribut obligatoire

Définition des attributs (extrait)

- name = Cet attribut nomme l'ancre courante de sorte que celle-ci puisse être la destination d'un autre lien. La valeur de cet attribut doit être un nom d'ancre unique. La portée de ce nom a pour cadre le document courant. Remarquez que cet attribut partage le même espace de nom que l'attribut id.
- href = Cet attribut spécifie la localisation d'une ressource Web, définissant ainsi un lien entre l'élément courant (l'ancre source) et l'ancre destination définie par cet attribut.

Chaque élément A définit une ancre :

- 1 . le contenu de l'élément A définit la position de l'ancre ;
- 2 . l'attribut name nomme l'ancre, de sorte que celle-ci puisse être la destination de zéro ou plusieurs liens ;
- 3 . l'attribut href spécifie la localisation d'une ressource Web.

Les auteurs peuvent également créer un élément A qui ne spécifie aucune ancre, i.e., qui ne spécifie aucun attribut href, name ou id. Les liens et les ancres définis par l'élément A ne doivent pas être imbriqués ; un élément A ne doit pas contenir d'autres éléments A.



## ***L'élément U (style de texte souligné)***

Balise ouvrante : obligatoire, balise fermante : obligatoire

L'élément U restitue un texte souligné. Déconseillé.

## ***Supplément : XHTML***

Etant donné que XHTML est une application XML, quelques habitudes qui étaient parfaitement légales dans le HTML 4 doivent être changées.

En résumé, les balises et les attributs suivent quelques règles simples :

- les noms des balises doivent être écrits en minuscules ;
- toute balise ouverte est fermée (par exemple <html>...</html> ou <br />) ;
- les noms des attributs doivent être écrits en minuscules ;
- les valeurs des attributs doivent être entre guillemets doubles.

## Annexe 4 : Le service Web

### Serveur Web (RFC HTTP 1945)

Le serveur Web est un programme applicatif acceptant des connexions dans le but de traiter des requêtes HTTP en délivrant une réponse HTTP. Il peut être considéré comme serveur de fichiers ou de documents.

Les deux principaux serveurs Web du marché sont : *Microsoft IIS* et *Apache* (serveur libre multi-plateforme représentant à lui seul environ 60% du marché)

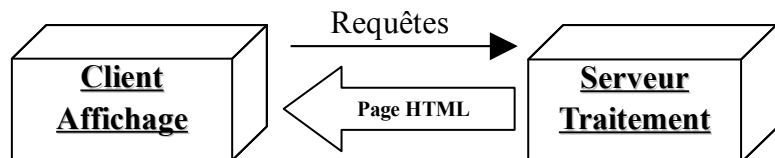
### Client Web (RFC HTTP 1945)

Un client est un programme applicatif qui permet d'émettre des requêtes HTTP et interpréter les réponses HTTP afin d'extraire les données reçues (HTML, Javascript, etc ...). Le plus souvent, l'applicatif client est un **navigateur** (en anglais *browser*, en québécois *fureteur*).

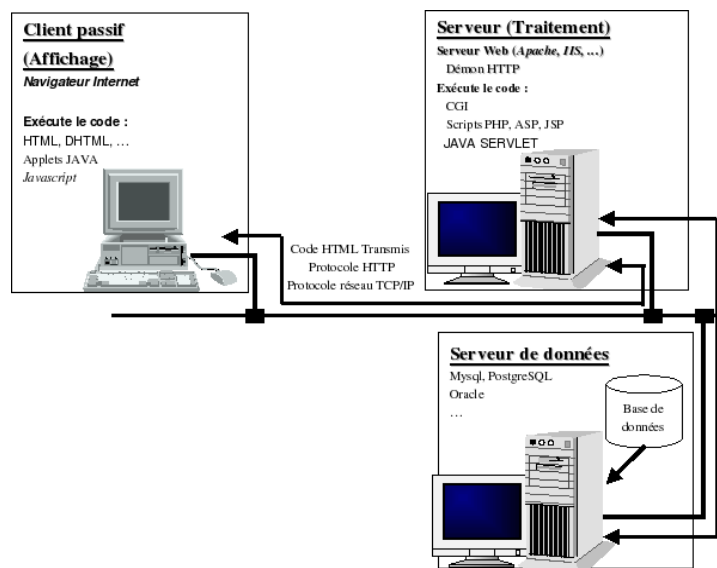
Les plus utilisés sont : Internet Explorer, Firefox, Safari, Chrome, Opéra, Lynx, Konqueror, ...

### Architecture Client/Serveur

C'est la topologie la plus courante actuellement. Le serveur est en attente des demandes en provenance du client : c'est le fournisseur du service.



L'architecture client/serveur Web plus détaillée peut être schématisée de la manière suivante :



#### Remarques:

- Le serveur de données peut, éventuellement, être installé dans la même machine que le serveur Web, si ces performances le permettent.
- Les documents à base de langages s'exécutant côté client ne nécessitent pas forcément la présence d'un serveur Web. De manière générale, le serveur permet l'accès distant et la centralisation des documents web à partager.

## **Annexe 5 : quelques formats d'image**

### ***PNG (Portable Network Graphics)***

C'est un format ouvert d'images numériques, qui a été créé pour remplacer le format GIF, à l'époque propriétaire et dont la compression était soumise à un brevet. Le PNG est un format non destructeur spécialement adapté pour publier des images simples. Il a été normalisé par l'ISO.

PNG est une spécification pour Internet et l'objet d'une Recommandation W3C et d'une RFC. Il a été créé pour contourner la licence existante sur le format GIF, le plus en vogue à la fin des années 1990, CompuServe refusant de laisser utiliser son format gratuitement. PNG a alors été défini mais en augmentant les capacités de GIF.

### ***GIF ( Graphics Interchange Format)***

C'est un format d'image numérique couramment utilisé sur le web.

GIF a été mis au point par CompuServe en 1987 pour permettre le téléchargement d'images en couleur. Ce format utilise l'algorithme de compression sans perte. En 1989, le format GIF a été étendu pour permettre le stockage de plusieurs images dans un fichier. Ceci permet de créer des diaporamas ou des animations.

### ***JPEG (JPEG File Interchange Format ou JFIF)***

C'est le format de fichier le plus utilisé pour contenir les images enregistrées avec la compression JPEG. L'extension de fichier .jpeg ou .jpg indique un fichier au format JFIF.

Le format JFIF est complémentaire du GIF et du PNG pour la publication d'image sur le World Wide Web. Bien que plus intéressante pour des photographies ou des images lourdes, la compression JPEG provoque une perte d'information (algorithme de compression destructif) pouvant aboutir à une perte de qualité visible si l'utilisateur privilégie un taux de compression élevé, particulièrement lorsque l'image contient des changements nets de couleur ou peu de couleurs (par exemple des logos, captures d'écran, diagrammes, ...). Le format JFIF ne gère ni les animations ni la transparence.

Actuellement une société détentrice de certains brevets intente une grande vague de procès contre les éditeurs de logiciels qui utilisent la compression JPEG.

### ***SVG (Scalable Vector Graphics)***

SVG est un format de données conçu pour décrire des ensembles de graphiques vectoriels et basé sur XML. Ce format est spécifié par le World Wide Web Consortium.

Le SVG est très utilisé dans le monde de la cartographie et du téléphone portable.

Lire : <http://fr.wikipedia.org/wiki/Svg>

## **Annexe 6 : quelques formats vidéos**

### ***MPEG (Moving Picture Experts Group)***

C'est un groupe de travail du comité technique de l'ISO et de la CEI pour les technologies de l'information. Ce groupe d'experts est chargé du développement de normes internationales pour la compression, la décompression, le traitement et le codage de la vidéo, de l'audio et de leur combinaison, de façon à satisfaire une large gamme d'applications.

MPEG a développé notamment les normes suivantes :

- MPEG-1 : première norme audio et vidéo utilisé plus tard pour les Vidéo CD. Elle comprend le populaire format audio MPEG-1 partie 3 audio couche 3 (MP3).
- MPEG-2 : norme applicable au codage de l'audio et la vidéo, ainsi que leur transport pour la télévision numérique : télévision numérique par satellite, télévision numérique par câble, télévision numérique terrestre, et (avec quelques restrictions) pour les vidéo-disques DVD ou SVCD. C'est notamment le format utilisé jusqu'à présent pour la TV sur ADSL.
- MPEG-4 : norme qui permet, entre autres, de coder des objets vidéo/audio, le contenu 3D et inclut le DRM. La partie 2 de MPEG-4 a connu du succès grâce à la mise en application DivX ainsi que dans les téléphones mobiles.

### ***AVI (Audio Video Interleave)***

C'est un format de fichier conçu pour stocker des données audio et vidéo. Il a été présenté par Microsoft en novembre 1992, en tant qu'élément de la vidéo pour la technologie de Windows.

Dans un fichier AVI, chaque composante audio ou vidéo peut être compressée par n'importe quel codec.

Le format DivX est souvent utilisé comme codec vidéo, et le format mp3 comme codec audio, mais d'autres codecs peuvent également être utilisés, par exemple XviD ou MPEG pour la vidéo, et mp2, WAV etc. pour l'audio.

### ***MOV (QuickTime)***

Un fichier QuickTime contient une ou plusieurs pistes, chacune comporte un type de données particuliers : audio, vidéo, effet ou texte (pour des sous-titres par exemple). Ces fichiers ont le plus souvent l'extension .mov.

QuickTime est un framework multimédia développé par Apple. QuickTime est utilisé par treize millions de personnes pour la lecture en streaming, ce qui en fait un des logiciels majeurs pour cet usage.

### ***WMV (Windows Media Video)***

C'est le nom d'une famille de codecs vidéo propriétaires développé par Microsoft. Sur Internet, il est fréquent de rencontrer ce type de fichiers vidéo, que ce soit en téléchargement ou en streaming.

### ***FLV (Flash Video)***

C'est un format de fichier utilisé sur Internet pour diffuser des vidéos via le lecteur Adobe Flash Player versions 6, 7, 8, 9 et 10, de manière à pouvoir l'incorporer aux animations Flash. Ce format est notamment utilisé par les sites de partage de vidéos sur Internet (youtube par exemple).