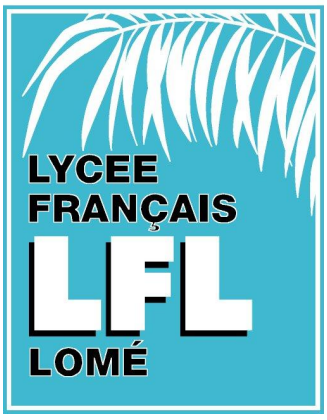
Le langage

HTML



Sommaire :

I. Le langage HTML :

Le réseau Internet fût inventé, vers 1970, par les américains pendant la guerre froide et fût l'un des premiers réseaux décentralisés permettant lors de la suppression d'un serveur de pouvoir continuer à faire fonctionner le réseau.

Le protocole de communication entre un serveur et un client est le HTTP *(HyperText Transfer Protocol)* : c'est la manière et la forme dont le client effectue sa demande au serveur et par laquelle le serveur lui communique les informations recherchées.

Comme tout ordinateur, le serveur renvoit les informations sous forme de donnée binaire : des 0 ou des 1. Ceux-ci mis bout à bout formeront des chaînes de caractères représentant la page Web demandée.

Ainsi, une chaîne de caractère peut indiquer au navigateur :

- soit d'afficher du texte :
- soit d'afficher de la couleur en arrière plan ;
- soit de chercher une image et de l'insérer dans la page affichée ;
- soit de disposer de les données sous forme d'un tableau...

Ainsi, une page HTML est composée, même si on voit apparaître des couleurs, des images, des tableaux, uniquement de caractères représentant du texte ou des directives.

Cette partie de la formation se concentre sur l'apprentissage de ce langage. Bien que des éditeurs de pages Web tels que KompoZer ou DreamWaver permettent de se dispenser de connaître ce langage, il est le chemin obligatoire pour construire des sites plus important ; l'utilisation du **JavaScript** et du PHP nécessite une bonne connaissance du langage HTML.

A. Un langage balisé :

Le langage HTML est un langage balisé ; pour formater un passage de la page en italique, on délimite cette partie à l'aide d'indicateurs : les balises.
La partie formée des balises et de son contenu s'appelle un élément HTML et sera noté dans l'ensemble de cette formation sous la forme élément

Une balise sécrit <nomDeBalise> pour la distingué du reste du texte ; ainsi, les deux caractères "<" et ">" ont une signification particulière en HTML. La plupart des élément HTML nécessitent une balise en début de partie *(appelé balise ouvrante)* et en fin de partie *(appelée balise fermante)*. Voici la forme générale de ces deux types de balises :

- ouvrante : <nomDeBalise>
- balise fermante : </nomDeBalise>

Quelques éléments HTML sont définis uniquement par une balise ouvrante : c'est le cas d'éléments n'incorporant aucun contenu tel que br définissant un saut de ligne.

Considérons le bout de code HTML suivant :

```
1 <i>Premier <b>code</b> style<br> HTML</i>
```

Et voici quelques remarques :

- L'élément `i` englobe l'intégralité du code, alors que `b` ne contient que le mot "*code*".
- Les éléments HTML peuvent s'emboîter les uns dans les autres.
- L'élément `i` représente son contenu en italique ; `b` met en gras son contenu :
  - Les mots "*Première*" et "*HTML*" sont affichés dans le navigateur en italique ;
  - Le mot "*code*" recevant l'action des deux éléments est affiché en gras et en italique.

Voici le résultat de ce code :

Premier **code** *style*
HTML

Le code suivant n'est pas en accord avec les règles syntaxiques du HTML :

```
1 <i>Premier <b>code</i> HTML</b>
```

Les éléments `i` et `b` se chevauchent alors qu'ils devraient s'emboîter ou être séparés. L'idée du programmeur était certainement d'obtenir le formatage suivant :

- "*Première*" en italique
- "*code*" en gras et en italique
- "*HTML*" en gras

Voici un code formattant la même chaîne de caractères mais juste au point de vue de la syntaxe du code HTML :

```
1 <i>Première</i> <i><b>code</b></i> <b>HTML</b>
```

Les navigateurs possèdent des correcteurs pour palier à ce genre d'erreurs, mais il est toujours préférable de taper correctement son code ; au moment de problèmes d'affichage, ce genre d'erreurs peuvent être dur à repérer.

B. Les attributs :

1. Présentation :

Chaque élément HTML a une fonction intrinsèque, nous avons déjà parlé des éléments `b` et `i` permettant de formater leurs contenus respectivement en gras et en italique.

L'élément `div` est l'un des éléments les plus utilisés, pourtant il n'apporte aucun formatage particulier. En fait, il crée un bloc s'étalant sur toute la largeur de la page et pouvant contenir plusieurs lignes *(comme un paragraphe)*.
Pour modifier, l'apparence de cet éément on lui rajoute des attributs.

L'élément font permet de modifier la fonte courante sur une partie du texte *(son contenu)*, on précise les caractéristiques de la fonte à l'aide d'attribut.

Un attribut, pour un élément HTML, est un couple $^{\text{propriété}}/_{\text{valeur}}$ s'écrivant dans la balise ouvrante de l'élément sous la forme :

propriété=*valeur*

Les valeurs sont normalement des chaînes de caractères mais si celles-ci ne présentent pas d'espace, il est possible d'oublier les guillemets.

Voici un exemple illustrant l'utilisation des attributs :

```
1 <div align="center">
2   <font size=6 face="Comic Sans MS" color="blue">
3     Bonjour
4   </font>
5 </div>
```

Etudions les attributs des différents éléments de cet exemple :

- L'élément div ne contient qu'un seul attribut :
  - L'attribut *align* pour valeur *"center"* : cet attribut a pour effet de placer le contenu de l'élément div en alignement centré.

- L'élément font possède trois attributs :
  - L'attribut *size* a la valeur *6* : le contenu de font aura une taille de 6 *(pour cette balise les tailles vont de 1 à 7).*
  - L'attribut *face* a la valeur *"Courier"* : cette police est utilisé pour afficher le contenu de l'élément font.
  - L'attribut *color* a la valeur *"blue"* : les caractères sont affichés en bleu.

Cet élément n'est plus utilisée actuellement ; on préfère les feuilles de styles CSS pour gérer le formatage d'une page.

⚠ Dans l'exemple précédent, des guillemets entourent les valeurs des attributs qui sont des chaînes de caractères. Elles peuvent être omises sauf dans le cas où la valeur contient des espaces *(comme pour la valeur "Comic Sans MS".)*

La valeur de l'attribut *face* peut inclure plusieurs valeurs : il faudra alors séparer ces différentes valeurs par des virgules.

2. *Les attributs les plus fréquents :*

id	Cet attribut donne une identification à un élément HTML de la page ; un seul élément de la page doit porter cet identifiant. Cet attribut est essentiellement utilisé par les feuilles de style et **JavaScript** pour localiser un élément	Presque tous les éléments HTML
class	Cet attribut permet de signifier l'appartenance d'un élément à un groupe d'élément. Cet attribut est essentiellement utilisé par **JavaScript** et les feuilles de styles pour agir simultanemment sur un groupe d'éément HTML	Presque tous les éléments HTML
title	Cet attribut permet de laisser des informations sur l'élément ; il est affiché à l'écran dans un bandeau jaune lorsque le client laisse son curseur dessus. Il peut être utilisé, notamment, pour laisser des informations sur une photo *(date du cliché, lieu...)*	Presque tous les éléments HTML
style	Cet attribut permet d'incorporer directement dans l'élément des déclaration de style CSS *(voir suite de la formation)* ; c'est à travers cet attribut qu'on définit les styles internes à un élément	Presque tous les éléments HTML
href	La valeur représente une URL. Sa fonctionnalité dépend de l'élément sur lequelle l'attribut est utilisé	a, area, link, base
src	La valeur de cet attribut est une URL. Elle indique, principalement, l'emplacement d'un fichier ; la fonction de cet attribut dépend de l'élément avec lequel il est utilisé	script, input, img
align	Elle permet généralement de fixer l'alignement du texte contenu dans l'élément. Ces valeurs possibles sont : *left* ; *center* ; *right*	Beaucoup d'éléments de présentation
border	Cet attribut prend pour valeur un nombre entier positif représentant l'épaisseur de la bordure entourant l'élément	table, img, object
height	Cet attribut défini la largeur de l'élément en pixel	td, img
name	Cet attribut est utilisé pour identifier un élément ; il est surtout utilisé pour définir une ancre ou pour nommmer les différents éléments d'un formulaire	img, a, input, select, textarea

size	Cet attribut définit soit la taille du texte (*font*), soit la taille du champs de texte d'un formulaire (*input*)	font, input
value	Cet attribut, pour les éléments d'un formulaire défini la valeur par défaut de ceux-ci.	input, option, param, button, li

Le reste des attributs seront présentés au fur et à mesure de la présentation du langage HTML. Une liste en annexe présente tous les attributs et les éléments auxquels ils se rapportent.

C. L'insensibilité à la casse :

Le langage HTML est insensible à la casse : cela signifie que le langage HTML ne fait aucune différence dans le nom des balises et des attributs entre les majuscules et les minuscules. Ainsi les navigateurs ne font aucune différence entre les trois balises ci-dessous ; elles représentent chacune la balise ouvrante d'un élément div :

`<DIV>` ; `<div>` ; `<DiV>`

Seul les valeurs des attributs sont parfois, mais rarement, sensibles à la casse *(c'est le cas pour la valeur de l'attribut* `script` *exécutant du code* **JavaScript***).*

D. Un langage de programmation? :

Non, le langage HTML n'est pas un langage de programmation mais plutôt un langage de description : à l'aide des balises, on décrit uniquement la mise en page du contenu et l'importation de données extérieurs *(récupération d'une image, lien hypertexte).*

Un document HTML ne possède pas de jeux d'instructions qui seront exécutées mais précise seulement aux navigateurs comment afficher son contenu.

II. Structure d'une page HTML :

Voici la structure minimale d'une page Web :

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN http://www.w3.org/TR/html4/strict.dtd">
2 <html>
3   <head>
4     Les en-têtes
5   </head>
6   <body>
7     Le corps du document
8   </body>
9 </html>
```

Commentons cette structure :

- La première ligne :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN http://www.w3.org/TR/html4/strict.dtd">
```

définit le doc-type du document.
Le HTML a connu de nombreuses versions, cette ligne précise la version du HTML qui sera utilisée pour l'interprétation et l'affichage des éléments HTML.
Le mode "*strict*" permet d'éviter de trop grandes différences dans l'affichage de notre page pour différents navigateurs.

- L'élément `html` englobe le reste du document. Sa présence est obligatoire même si son utilité est dicutée : à l'origine, il était prévu qu'une page Web contiennent du code HTML mais aussi d'autres types d'informations ; cet élément permet de baliser la partie représentant du code HTML.
Cette balise contient deux autres éléments :
  - L'élément `head` est l'entête du document, il contient des informations relatives à la page : titre de la page, table de caractères utilisée, mots-clefs caractérisant le document.
  - L'élément `body` est le corps du document, il contient l'ensemble du texte et des éléments HTML qui seront affichés dans la fenêtre du navigateur.

C'est la structure minimale qui a été présentée ici.
Mais en fait, on peut directement taper le corps du document sans se préoccuper des éléments `html` et `head` ; le serveur de l'hébergeur ou le navigateur du client rectifieront le code. Mais il reste toujours souhaitable d'écrire corectement le code : en effet, par moment l'exécution de **JavaScript** peut entraîner des difficultés si l'élément `body` n'est pas explicitement déclaré.

A. L'élément `head` :

Comme nous avons vu l'en-tête de la page Web contient des informations relatives à cette page et il est défini par l'élément `head`.

Voici un exemple assez complet, mais non-exhaustif, de l'utilisation de l'élément `head` :

```
1 <html>
2   <head>
3     <title>Mon titre</title>
4     <meta name="Author" content="David Vincent">
5     <link rel="stylesheet" type="text/css" href="↩
        maFeuille.css">
6     <style type="text/css">
7       .chapitre{font-bold:900}
8       #titre{position:absolut;top:5px}
```

```
9       </style>
10    </head>
11    <body>
12       Corps du document
13    </body>
14 </html>
```

Présentons chacune des balises présentes dans l'élément head dans l'exemple précédent :

1. Le contenu de l'élément title définit le titre de la page. Le titre est affiché dans la barre des titres du navigateur.

2. Les méta-données, définies par l'élément meta, sont des informations concernant la page à seule destination du navigateur *(elles ne seront pas affichées)* pour l'archivage, pour définir l'encodage des caractères, pour des informations personnelles...
   Seul les attributs que cet élément intégre définissent sa fonction ; voici quelques exemples d'utilisation de cet élément :

   - Cette balise précise l'auteur d'une page :
     `<meta name="Author" content="David Vincent">`
   - Il précise les mots-clefs caractérisant la page :
     `<meta name="keywords" lang="fr" content="vacances,scolaires">`
     Cet élément peut aider les moteurs de recherche pour référencer une page.
   - Il redirige automatiquement le client vers une autre page :
     `<meta name="refresh" content="3,http://www.acme.com/intro.html">`
     Cet élément redirige au bout de 3 secondes, le client vers la page d'URL :
     `http://www.acme.com/intro.html`.

   Les éléments suivants modifient les en-têtes HTTP envoyés par le serveur hébergeant la page lors de son envoi ; nous ne rentrerons pas dans l'étude du protocole HTTP mais voici quelques fonctionnalités intéressante :

   - Il récise la date d'expiration d'un document :
     `<meta http-equiv="Expires" content="Tue, 20 Aug 1996 14:25:27 GMT">`
     Cette information permet au navigateur de garder un certain temps la page Web en cache ; évitant ainsi de toujours la recharger.
   - Il précise le jeux de caractères utilisée dans ce document :
     `<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-5">`
     L'utilisation précédente assure que les caractères latin seront utilisés pour l'affichage de la page.
   - Il précise la nature des scripts utilisés dans la page :
     `<meta http-equiv="Content-Script-Type" content="text/javascript">`
     JavaScript est défini comme langage de script par défaut.

- Il précise le langage de style utilisé par défaut dans la page :
  `<meta http-equiv="Content-Style-Type" content="text/css">`

Ces deux derniers points sont optionnels car, **JavaScript** et CSS sont les modes par défaut utilisés par l'ensemble des navigateurs pour le langage de script et de style.

3. L'élément link permet de relier la page actuelle avec un fichier externe
   - Pour utiliser dans la page actuelle, une feuille de style externe, on utilisera l'élément link de la manière suivante :
     `<link rel="stylesheet" type="text/css" href="smartstyle.css">`
     On incluera une page de script externe de la même façon.
   - Si la page actuelle fait partie d'une suite ordonnée de plusieurs pages, on indiquera au moteur de recherche une telle filiation de la manière suivante :
     `<link rel="Start" title="première page" href="http://...">`
     Cette fonctionnalité n'est plus utilisée.

4. L'élément style permet de déclarer une feuille de style interne *(appelé également feuille de styles en ligne)*. Les styles CSS seront étudiés dans la prochaine partie de la formation.

B. L'élément body :

Cet élément représente l'espace d'affichage du navigateur. Tout élément qui doit être affiché dans la fenêtre du navigateur doit être présent dans cet élément.

Voici les attributs de cet élément :

- *background* prend pour valeur une URL.
  L'URL est l'adresse de l'image utilisée pour remplir le fond de la page.

- *text* prend pour valeur une couleur.
  La couleur est utilisée pour représenter la fonte courante.

- *link* prend pour valeur une couleur.
  La couleur est utilisée pour les liens non-visités.

- *vlink* prend pour valeur une couleur.
  La couleur est utilisée pour les liens déjà visités.

- *alink* prend pour valeur une couleur.
  La couleur est utilisée au moment où le client sélectionne un lien.

- *bgcolor* prend pour valeur une couleur.
  Cette couleur est utilisée pour remplir le fond de la page

Ces attributs ne sont plus utilisés actuellement, les feuilles de styles les ont remplacés ; actuellement, l'élément body est déclaré sans attribut.

Notons également les attributs *onload* et *onunload* qui sont des gestionnaires d'événements utilisés conjointement avec **JavaScript** pour lancer des actions respectivement lorsque la page vient de finir de se charger et lorsque la page est quittée.

III. Les conteneurs et les identifiants :

A. les éléments div et span :

Ces deux éléments, définis tous deux par leurs balises ouvrantes et fermantes, n'apportent aucun formatage à leur contenu. Ce sont de simples conteneurs : ils permettent de rassembler du texte et des éléments HTML dans un seul et même élément.

Leur intérêt est très lié au feuille de style CSS que nous verrons dans la prochaine partie de cette formation : en appliquant une règle de style à un de ces éléments, on répercute le style sur l'ensemble de son contenu.

Ils sont également utilisé avec les scripts **JavaScript** permettant de manipuler un ensemble de texte et d'élément comme une unique entité.

Voici la différence entre ces deux éléments :

- L'élément span est un élément en ligne : il est une considérée comme un élément faisant partie d'une ligne de la page ; il ne peut contenir de saut de ligne et prendra la taille de son contenu.
- L'élément div est un élément bloc : il prend pour largeur celle de la page et s'étale sur une ou plusieurs lignes en fonction de son contenu.
  Cet élément accepte l'attribut *align*.

L'utilisation de ces deux éléments permet de mieux structurer la page Web. Voici quelques contraintes inhérentes à leurs natures :

- Un élément span ne peut contenir d'éléments div et p qui, eux, entraînent un saut de lignes.
- Un élément div peut contenir des élément span, div et p.
  Par contre, il ne peut être contenu dans un élément p.

Les navigateurs corrigent, eux-même, la plupart de ces fautes de syntaxes. Mais pour prévenir des bugs d'affichage entre les différents navigateurs, il est impératif de suivre les règles suivantes :

- un élément p doit toujours être fermé avant l'apparition d'un élément div.
- Un élément span ne doit jamais contenir un élément p ou div.

B. Les identifiants class et id :

Presque tous les éléments HTML possèdent les attributs `class` et `id`. Ces derniers permettent respectivement de faire appartenir l'élément à un groupe d'élément *(`class`)* ou de l'identifier de manière unique *(`id`)*.

Ceci permet :

- de leur appliquer un même style CSS à l'aide d'une feuille de style CSS ;
- de les identifier lors de l'exécution d'un script **JavaScript**.

Ces deux attributs prennent pour valeur une chaîne de caractères commençant par une lettre et pouvant contenir des chiffres, des tirets "-" et des points ".". Voici des valeurs correctes pour ces attributs :

```
<div class="chapitre" id="c-01">
  <span class="titre"> ... </span>
</div>

<div class="chapitre" id="c-02">
 <span id="titre-02" class="titre">...</span>
</div>
```

Voici quelques commentaires :

- L'attribut `id` permet d'identifier l'élément de manière unique ; deux éléments d'une même page ne doivent pas partager la même valeur pour l'attribut `id` :
  Les valeurs *c-01*, *c-02* et *titre-02* sont distinctes.
- L'attribut `class` permet de regrouper les éléments dans un même groupe. Les deux éléments `div` de notre code partagent la même valeur pour l'attribut `class` ; ainsi, lorsqu'on relira une règle de styles à la valeur de cet attribut, les deux `div` recevront cette règle de style.

IV. Texte et formatage :

Nous ne présenterons pas l'ensemble des éléments existant pour cet usage. Seul les plus utiles et quelques cas intéressants pour cette formation seront présentés ici ; reportez-vous à la documentation du W3C pour d'avantage d'information.

A. Les espaces et les sauts de lignes dans le code :

Pour bien comprendre ce qui va suivre, il faut bien faire la différence entre les deux choses suivantes :

- le code du langage saisi dans le fichier ;
- et l'affichage qui en est fait par le navigateur.

Lorsque vous tapez les trois caractères “ ” dans votre fichier, le navigateur ne les affichera pas : il remplace cet balise par un début de paragraphe.

Voici quelques interprêtations du code fait par les navigateurs et dont il faut bien avoir conscience lorsque vous tapez votre code

- Une suite d'espaces est considéré comme un simple espace.
- Un ou plusieurs saut de ligne sont également interprêtés comme un simple espace.

Ainsi les deux codes suivants auront pour résultat le même affichage

```
Premier code
Bonjour     messieurs,

comment
allez-vous?

Deuxième code
Bonjour messieurs, comment allez-vous?
```

B. Sauts de lignes et paragraphes :

On vient de voir qu'on a beau saisir plusieurs saut de ligne, rien n'y fait, le texte s'affichera sur une seule ligne. Le seul moyen d'afficher un retour à la ligne est d'utiliser les éléments HTML prévus à cet effet :

- L'élément `br`, défini par sa seule balise ouverte <br>, oblige le navigateur a effectuer un saut de ligne lors de l'affichage.
- L'élément `p`, défini par ces deux balises et , crée un paragraphe : le navigateur affichera le contenu de cet élément sur une nouvel ligne en plaçant un espace vertical au dessus et en dessous de l'élément afin de le séparer du reste du document.
  L'élément `p` accepte l'attribut *align*.

⚠ L'élément `p` ne peut pas contenir d'éléments `div` et `p`. Si un tel cas arrive, les navigateurs gèrent ce défaut de syntaxe en fermant automatique l'élément `p` : ceci peut entraîner un affichage inattendu.

La balise fermante est optionnelle, mais FireFox et Internet Explorer ne réagissent pas de la même façon : il est donc conseillé de l'écrire explicitement. Essayez le code suivant avec les deux navigateurs.

```
<p>Premier paragraphe
<div>Second paragraphe</div>
```

Cette différence resulte d'une veille guerre lors de la conception du langage entre Microsoft et Netscape.

C. D'autres éléments pour le formatage :

- L'élément pre, défini par ses deux balises, permet d'afficher son contenu en respectant les règles suivantes :
  - Tous les espaces présents dans le contenu de l'élément sont affichés tel quel.
  - Les sauts de lignes du contenu sont également respectés lors de l'affichage.

  L'utilisation de cet élément provoque également un changement de fonte.

- Les éléments em, strong, cite, dfn, code, samp, kdb, var, abbr, acronym *(définis chacun par leurs deux balises)* offre un formatage pré-établi afin que tous les bouts de codes, les citations, les abbrévations aient la même apparence d'un bout à l'autre d'un site Web. Les feuilles de styles CSS ont rendu ces éléments obsolètes.

- Les éléments h1, h2, h3, h4, h5, h6 *(définis chacun par leurs deux balises)* permettent de les titres de pages, de chapitres, de paragraphes, de sous-paragraphes... Ils sont également de moins en moins utilisés.

V. Les listes :

A. Les éléments HTML :

L'élément ul, défini par ses deux balises <ul> et , permet de créer des listes non-numérotés *(non-ordinales)* de la forme suivante :

- liste 1
- liste 2
- liste 3

L'élément ol, défini par ses deux balises, <ol> et permet de créer des listes numérotés *(listes ordinales)* de la forme suivante :

1. liste 1
2. liste 2
3. liste 3

Dans les deux cas, chaque élément de la liste est défini par l'élément li : cet élément est défini par ces deux balises, la balise fermante est néanmoins optionnelle.

Voici le codage des deux listes précédentes :

```
1 Liste non-ordinale
```

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

 Liste ordinale
<ol>
   <li> liste 1
   <li> liste 2</li>
   <li> liste 3
</ol>
```

B. Les attributs des listes :

On peut contrôler l'aspect des listes par l'utilisation d'un certain nombre d'attributs présentés ci-dessous :

- L'attribut ***type*** permet de définir le symbole marquant le début d'un nouvel élément de la liste :
  - Pour l'élément ul :

	Affichage
type="*disc*"	•
type="*square*"	■
type="*circle*"	○

  - Pour l'élément ol :

	Affichage
type="*1*"	1, 2, 3...
type="*a*"	a, b, c...
type="*A*"	A, b, C...
type="*i*"	i, ii, iii...
type="*I*"	I, II, III...

- L'élément ol admet l'attribut ***start*** permettant de modifier le rang par lequel commence la numérotation des éléments de sa liste *(par défaut, cette valeur vaut 1)*.

- L'élément li, définissant un élément de la liste, admet les deux attributs suivants :
  - L'attribut ***type*** permet de modifier localement le symbole utilisé pour marquer l'élément courant de la liste *(voir les valeurs dans les tableaux ci-dessus)*.
  - Un élément d'une liste ordinale *(définie par ol)* peut voir sa numérotation changée à l'aide de l'attribut ***value***.

VI. Les tableaux :

A. Les éléments HTML :

Voici les éléments essentiel pour construire des tableaux :

- L'élément table est défini par ses deux balises <table> et </table>. Son contenu définit le corps du tableau. L'insertion d'un tableau entraîne le passage à une nouvelle ligne *(les paragraphes courant seront implicitement fermés)*.
- L'élément tr est défini par ses deux balises ; la balise fermante est optionnelle. Son contenu définit une ligne du tableau.
- L'élément td est défini par ses deux balises ; la balise fermante est optionnelle. Son contenu définit le contenu d'une cellule du tableau.

Ainsi, un tableau est construit ligne par ligne.

Voici d'autres éléments pour la construction de tableau mais leur utilisation est plutôt rare, ils ne seront pas discutés dans cette formation.

- L'élément caption représente le titre du tableau et il est affiché au dessus du tableau.
- Les éléments thead , tbody et tbody contiennent une ou plusieurs lignes. Ces éléments permettent de structurer un tableau, ils représentent respectivement l'entête, le corps et le bas de tableau. Aucun formatage particulier n'est apporté mais elles permettent de mieux structurer un tableau et l'utilisation des feuilles de styles peut alors faciliter leur mise en page.
  Plusieurs éléments tbody peuvent être présent dans un même tableau.
- L'élément th permet de définir les labels pour les colonnes d'un tableau.
- Les éléments colgroup et col , dans leur utilisation cojointe, permettent d'appliquer un même formatage à un groupement de plusieurs colones.

B. Les attributs de table , tr , td :

Seul les attributs réellement les plus utiles sont présentés ici : de nombreux attributs sont devenus obsolètes avec l'arrivée des feuilles de styles. Vous trouverez l'ensemble des attributs de ces trois éléments en annexe.

- Pour l'élément table :
  - ***align*** définit l'alignement du tableau dans la page courante : les valeurs possibles sont *left*, *center*, *right*.
  - ***width*** est un entier représentant la largeur en pixels du tableau ou un pourcentage représentant la largeur du tableau relativement à celle de la page courante.
  - ***cellspacing*** est un entier représentant l'espacement entre les cellules. Sa valeur sera une longueur.

- ***cellpadding*** est un entier définissant la marge intérieure des cellules : c'est à dire l'espace séparant le contenu d'une cellule de sa bordure.
- ***border*** est un entier définissant l'épaisseur de la bordure entourant le tableau et ses cellules. Par défaut, sa valeur est nulle.

- Pour l'élément `tr` :
  - ***align*** définit l'alignement du contenu des cellules de la ligne courante. Ses valeurs possibles sont *left*, *center*, *right*.
  - ***valign*** définit l'alignement vertical des cellules de cette ligne *(L'effet ne sera visible si au moins une cellule de cette ligne s'affiche sur plus d'une ligne)*. Il accepte pour valeur : *top*, *middle*, *bottom*, *baseline*.

- Pour l'élément `td` :
  - ***colspan*** fusionne la cellule courante avec les cellules se trouvant à sa droite. Sa valeur est un entier et représente le nombre de cellules fusionnées.
  - ***rowspan*** fusionne la cellule courante avec les cellules se trouvant sous elle. Sa valeur est un entier et représente le nombre de cellules fusionnées.
  - Pour respecter l'espace horizontal disponible ou imposé par l'attribut ***width***, certaines cellules du tableau peuvent s'afficher sur plusieurs lignes ; un retour à la ligne est inséré automatiquement par le navigateur. L'attribut ***nowrap*** interdit au navigateur d'insérer de tels retour à la ligne *(cet attribut ne prend pas de valeur)* : il est alors fréquent que la largeur du tableau dépasse celle de la page.
  - Les attributs ***align*** et ***valign*** s'appliquent également aux cellules.

C. Un exemple :

Voici un bout de code illustrant l'utilisation de ces éléments et de ces attributs et sa représentation est la tableau de droite :

```
1 <table border=1 cellspacing=↩
    3 cellpadding=10>
2 <tr><td rowspan=2 valign="↩
    middle">1-1<td>1-2<td>1-3
3 <tr><td>2-2<td>2-3
4 <tr><td>3-1<td colspan=2 ↩
    align="center">3-2
5 </table>
```

1-1	1-2	1-3
	2-2	2-3
3-1		3-2

VII. Les liens :

Les liens hyper-texte permettent au client de passer d'une page à l'autre en cliquant sur ceux-ci. Ce fut le grand apport des pages Web dans les années 90.

C'est l'élément `a` qui définit la plupart des liens ; d'autres redirections peuvent être définies par un formulaire ou un script **JavaScript**. Lorsque le client clique sur le contenu de l'éément `a`, le navigateur se dirige vers une nouvelle page qu'on appelle la *page cible* du lien.

A. Les adresses :

Pour rediriger un client vers une autre page, il faut pouvoir indiquer l'emplacement du fichier cible, son adresse. Les URL sont les "*Uniform Resource Location*" et représentent l'adresse des fichiers sur le Web. Il existe de deux types d'URL :

- Les *URL absolues* :
  On indique entièrement l'emplacement des fichiers. L'URL a l'allure suivante :
  `http://audacity.sourceforge.net/download/linux/index.html`
  `http` est le protocole de communication ; `audacity.source.net` est le domaine ; le fichier actuellement visualisé se trouve à l'emplacement :
  `download/linux/index.html`
  en partant de la racine du site.

- Les *URL relatives* :
  On indique uniquement le chemin a parcourir du fichier source vers le fichier cible ; c'est à dire qu'on indique les dossiers qu'on traverse, ceux qu'on remonte pour partir du fichier contenant le lien et se diriger vers le fichier cible
  Dans l'écriture de telles URL, le double point "**..**" représente le dossier parent du dossier courant. Les URL relatives ont déjà étudiées dans KompoZer et seront revus dans la partie exercice.

La maîtrise des URL relatives est essentiel pour la construction d'un site Web. En effet, construisant le site sur vos ordinateurs l'adresse représentant un fichier peut s'écrire ainsi :

- Pour Window :
  `C:\monSite\Actualite\essaie.html`

- Pour Linux :
  `/utilisateurs/home/profs/castanet/public_html/Actualite/essaie.html`

Mais une fois transféré chez un hébergeur, ce fichier sera consultable par votre navigateur par une adresse de la forme :

`http://www.monHebergeur.fr/Actualite/essaie.html`

Tous les liens de votre site ciblant un fichier de votre propre site devront être écris à l'aide des URL relatives.

B. L'élément `a` :

L'élément a permet de définir un lien mais voici les attributs qui permettront de définir le comportement du lien :

- La valeur de l'attribut ***href*** est une chaîne de caractère qui désigne l'URL *(relative ou absolue)* de la page cible..
  Si sa valeur débute par la séquence *mailto*, par exemple :
  "mailto:thomas@hotmail.com"
  alors le logiciel de messagerie *(s'il en existe un)* ouvrira un message à destination de *thomas@hotmail.com*.

- L'attribut ***target*** permet de dirigier l'ouverture du lien vers une autre fenêtre. Par défaut, l'action d'un lien change la page courante. Voici les valeurs possibles de cet attribut :
  - avec la valeur *_blank*, à chaque action sur le lien, une nouvelle fenêtre s'ouvrira.
  - avec une chaîne quelconque, le lien s'ouvrira pour la première fois dans une nouvelle fenêtre mais les prochains liens actionnés possédant le même attribut ***target*** s'ouvriront dans cette fenêtre *(limitant ainsi le nombre de fenêtres ouvertes)*.

  Lorsque le site est construit à l'aide de cadres *(technique ayant tendance à disparaître)*, les valeurs *_self*, *_parent*, *_top* permettaient d'indiquer dans quel cadre le lien s'ouvrait.

- L'attribut ***name*** prend pour valeur une chaîne de caractères. Avec sa présence, l'élément devient également une ancre.

C. Les ancres :

Lorsqu'une page atteint une taille consécante, il peut être utile de placer des liens à l'intérieur de la page pour se déplacer plus facilement d'un endroit à un autre de la même page.

Les ancres peuvent être définis :

- Par tout élément possédant l'attribut ***id***. La valeur de cet attribut est alors l'identifiant de cet ancre.

- Par tout élément a ayant ***name*** pour attribut. Sa valeur est alors l'identifiant de l'ancre.

Pour utiliser un lien se dirigeant directement à l'emplacement d'une ancre, on utilise un élément a où la valeur de l'attribut ***href*** est l'URL désignant le fichier cible sauf qu'à la fin de cette chaîne de caractères, on rajoute le symbole "#" suivi du nom de l'ancre.

Voici un exemple :

Supposons que la page `http://monSite.com/index.html` contiennent une ancre nommée "*sommaire*". Voici un lien ciblant ce fichier et se positionnant directement sur l'ancre :

...

VIII. Les images et les objets :

A. Les images :

Pour insérer des images dans une page Web, on se sert de l'élément img. Cet élément est défini seulement par sa balise ouverte <img>.

Ce sont ses attributs qui précisent la localisation du fichier contenant l'image mais permettent également de préciser quelques paramètrages d'affichage :

- L'attribut *src* contient l'URL du fichier contenant l'image. La présence de cet attribut est obligatoire pour tout élément img.
- L'attribut *align* permet de définir la position de l'image relativement à son conteneur :
  - Les valeurs *bottom*, *middle*, *top* définissent son alignement vertical relativement à la ligne de base courante.
  - Les valeurs *left*, *right* donne à l'image le status d'objet flottant : l'image est placé à gauche ou à droite de la page, le reste de la page s'affiche en entourant l'image.
- L'attribut *border*, prenant un entier pour valeur, définit l'épaisseur de la bordure entourant l'image. Par défaut, sa valeur vaut 0.
- Les attributs *usemap* et *ismap* permettent de définir une imageMap permettant de rendre cliquable certaines zones de l'image. Cette attribut ne sera pas discutée ici.

B. Les objets :

Tous les navigateurs, actuellement, prennent en charge *(quitte à télécharger un plugin)* l'affichage de vidéo, d'animation Flash...

Tous ces objets sont inclus dans les pages Web à l'aide de l'élément object défini par ses deux balises. On donne les paramètres d'affichage au plugin à l'aide de l'élément param.

Voici des exemples d'inclusion :

```
1  Voici le code pour appeler des fichiers avi et mpeg
2  <object type="application/x-mplayer2">
3  <param name="autostart" value="true">
4  <param name="showcontrols" value="true">
5  <param name="filename" value="*.mpg">
6  </object>
7
8  Voici le code pour appeler une vidéo QuickTime
9  <object classid="clsid:02BF25D5-8C17-4B23-BC80-↩
       D3488ABDDC6B" codebase="http://www.apple.↩
       com/qtactivex/qtplugin.cab" width="" height="">
10 <param name="src" value="*.mov">
11 <param name="controller" value="true">
```

```
<object  width="" height="" class="mov" type="video/quicktime" data="/images/peint.mov">
<param name="controller" value="true">
</object>
</object>

Pour une animation Flash
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=7,0..." width="200" height="20" id="dewplayer" align="middle"><param name="allowScriptAccess" value="sameDomain">
<param name="movie" value="dewplayer.swf?mp3=techno.mp3&amp;bgcolor=FFFFFF">
<param name="quality" value="high" /><param name="bgcolor" value="FFFFFF">
</object>
```

Voici différentes remarques relatives aux éléments et aux attributs présents dans cet exemple :

- Pour l'élément object :
  - L'attribut *type* représente le MIME type de l'objet à insérer. Ce code permet au navigateur de connaître le type de fichier envoyé et de lancer automatiquement le lecteur à ce type de fichier. Voici quelques exemples :
    - Vidéo : *"application/x-mplayer2"*
    - Animation ou vidéo Flash : *"application/x-shockwave-flash"*
    - Vidéo RealPlayer : *"audio/x-pn-realaudio-plugin"*

Les deux attributs suivants sont spécifiques à Internet Explorer, ils lui permettent de déterminer quelle application utilisée pour afficher l'objet introduit par object :

- L'attribut *classid* donne une clé associée de manière unique avec le logiciel à utiliser pour afficher object.
- L'attribut *codebase* indique généralement l'emplacement du logiciel pour le télécharger s'il n'est pas présent sur l'ordinateur du client.

- Pour l'élément param :
  - L'attribut *name* indique le nom du paramètre à fournir au logiciel de visualisation.
  - L'attribut *value* détermine la valeur de ce paramètre

Anciennement, Internet Explorer utilisait l'élément embed permettant d'actionner des

activeX notamment pour l'inclusion de tels fichiers ; cet élément ne fait pas partie du standard du W3C.

Un manque de standardisation entoure l'inclusion d'objet dans une page Web ; ceci vient du combat entre Netscape et Microsoft mais également car, par exemple, il existe plusieurs lecteurs permettant d'afficher une vidéo dans une page Web *(Media player, Real player...)* et il n'existe pas non plus de standardisation sur les paramètres utilisés par chacun de ces lecteurs.

Une seule solution pour être sûr que votre page est valide : testez votre page sur différents navigateurs et systèmes d'exploitation.

IX. Les formulaires :

A. Introduction :

Un formulaire est un espace de la page Web dans lequel le client peut cocher des cases, remplir des champs de texte... afin de fournir des informations.
Celles-ci sont ensuite envoyées vers le serveur hébergeant le site pour traitement ; ces données peuvent être enregistrées dans une base de donnée ou alors peuvent permettre de renvoyer au client des informations en fonctions du formulaire qu'il vient de remplir.

Pour traiter ses données sur le serveur, il faud un langage de programmation tel que PHP ou ASP. Il n'est pas question, à ce niveau, d'apprendre la programmation sur de tels langages *(plus tard?!)*

Une fois les données saisies, le client appuye sur un bouton pour envoyer les données du formulaire : cette action s'appelle la *soumission du formulaire* au serveur. Les données sont transférées jusqu'au serveur sous la forme $^{\text{nom}}/_{\text{valeur}}$. Lors de la soumission du formulaire, le navigateur demande au serveur une nouvelle page : c'est en intégrant les données à cette demande que les données sont transférées jusqu'au serveur.

Il existe deux modes d'envoi des données :

- La méthode *GET* :
  les couples de données $^{\text{nom}}/_{\text{valeur}}$ sont transférées jusqu'au serveur via l'URL ; en voici un exemple :
  `http://www.monHebergeur.fr/traitement.php?age=19&prenom=thomas&choix=4`
  Dans cette URL, on repère les couples suivant de données $^{\text{age}}/_{19}$, $^{\text{prenom}}/_{\text{thomas}}$ et $^{\text{choix}}/_{4}$.

  Cette méthode présente l'inconvénient de présenter dans l'URL les données saisies par l'utilisateur *(manque de confidentialité)* et une limitation en taille *(l'URL est limité suivant les navigateurs à 255 caractères)*.

- La méthode *POST* :

Les couples de données sont transmises dans l'entête de la demande, ce point est plus délicat est fait partie du protocole HTTP et n'a pas lieu d'être cité ici mais même si les données ne sont pas cryptées, elle reste difficilement lisible pour un simple utilisateur.

B. L'élément form :

L'élément form, à l'aide de ses deux balises, définit le corps du formulaire : tous les contrôles d'un même formulaire doivent être contenu dans le même élément form.

Voici les attributs de cet élément :

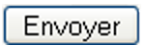
- L'attribut ***action*** a pour valeur l'URL vers la page vers laquel le client va être redirigé, cette page doit inclure le code de programmation traitant les données envoyées par le formulaire avant de renvoyer un page de confirmation *(par exemple)* au client.
- L'attribut ***method*** est une chaîne de caractère indiquant la méthode d'envoi des données : ses deux valeurs peuvent être *get* ou *post (voir Introduction)*.
- L'attribut ***onsubmit*** permet juste avant la soumission d'un formulaire d'exécuter un script **JavaScript** ; ainsi, on peut vérifier l'intégrité des données saisies par le client dans le formulaire.
- L'attribut ***target***, comme pour l'élément a, spécifie dans quelle fenêtre est redirigée le client.
- Lors de l'envoie de fichiers par un formulaire, un traitement des données avant l'envoie doit être précisé. L'attribut ***enctype*** doit alors posséder la valeur *"multipart/form-data"*.

C. L'élément input :

L'élément le plus fréquent dans un formulaire est l'élément input ; cet élément n'est défini que par sa balise ouvrante, sa balise fermante n'existe pas. La nature de ce contrôle n'est déterminée que par l'attribut ***type***. Passons directement à l'étude de ses attributs :

- L'attribut ***type*** détermine la nature du contrôle :

Valeur de ***type***	Commentaire	Représentation
text	Propose un champ de texte sur une ligne dans lequel le client peut saisir du texte	texte
password	Propose un champ de texte mais lorsque le client tape seul s'affiche le caractère * préservant les données saisies des curieux ; mais attention les données ne sont pas codées pour autant.	******

Valeur de *type*	Commentaire	Représentation
checkbox	Propose une case à cocher. Si la case est coché le couple *nom/valeur* est transmise au serveur lors de la soumission du formulaire, dans le cas contraire le couple $^{nom}/_{valeur}$ est ignoré.	
radio	Cet élément est également une case à cocher mais parmi tous ces éléments ayant également une valeur commune de l'attribut ***name***, seul l'un d'eux peut être sélectionné. La sélection de l'un entraîne la déselection de l'autres.	
submit	Propose un bouton ayant pour fonction la soumission du formulaire au serveur.	
reset	Propose un bouton réinitialisant les contrôles à leurs valeurs par défaut.	
file	Propose au client d'indiquer un fichier présent localement sur le disque dur du client pour l'envoyer en pièce jointe au serveur.	
hidden	Propose un contrôle ne s'affichant pas et dont la valeur ne peut pas être modifiée par le client ; parfois utile en programmation.	
image	Propose une image cliquable entraînant la soumission du formulaire. Deux couples de valeurs supplémentaires seront envoyés relative à la position de l'action sur l'image. On peut coupler cette image à une imageMap permettant de découper celle-ci en plusieurs zones cliquables.	
button	Propose un bouton dont l'action n'est pas définie : on se servira de ceci dans les scripts.	

Voici les attributs disponibles pour le contrôle input. Certains sont accessibles suivant la valeur de l'attribut ***type*** :

- L'attribut ***name*** défini le nom du contrôle et servira, lors de l'envoi du formulaire, à la transmition des données sous la forme $^{nom}/_{valeur}$.

- L'attribut ***value*** définit :
  - Pour les contrôles de type *text*, *password*, *file*, elle représente la valeur par défaut du contrôle ; elle peut être modifiable par le client.

- pour les contrôles de type *checkbox*, *radio*, *hidden*, *submit*, la valeur de **`value`** est la valeur associée au contrôle : c'est celle-ci qui sera renvoyée au serveur si le contrôle est sélectionné lors de la soumission.

- Pour les contrôles *text*, *password*, *file*, la valeur de **`size`** représente le nombre de caractères que peut afficher le champs texte associé à ce contrôle.

- Pour les contrôles *text* ou *password*, la valeur de l'attribut **`maxlength`** est le nombre maximal de caractères que peut saisir le client.

- Pour les contrôles *checkbox* et *radio*, la présence de l'attribut **`checked`** *(attribut ne prenant aucune valeur)* indique si le contrôle est coché par défaut.

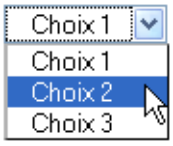
- Pour les contrôles de type *image*, la valeur de **`src`** est une URL indiquant la direction de l'image représentant le bouton.

Les deux attributs suivants sont généralement utilisés conjointement à l'éxécution de code **JavaScript**. Ils ne prennent aucune valeur.

- Pour les contrôles de type *text*, *password*, *file*, la présence de l'attribut **`readonly`** empêche le client d'en modifier la valeur.
  Sa valeur est quand même transmise au serveur lors de la soumission du formulaire.

- Pour tous les contrôles issus de l'élément `input`, l'attribut **`disabled`** désactive le contrôle. Le client ne peut en modifier la valeur et le couple de donnée associé à ce contrôle ne sera pas envoyé lors de la soumission du formulaire.

D. Les listes d'options :



L'élément `select` proposent au client un contrôle présentant plusieurs valeurs généralement à l'aide d'un menu déroulant ; le client sélectionne un *(ou plusieurs)* choix parmi les choix proposés.

L'élément `select` est défini par ses deux balises ; chaque choix proposé est défini à l'aide de l'élément `option` *(la balise fermante est optionnelle)*.

Voici les différents attributs de ces deux éléments :

- Pour l'élément `select` :
  - **`name`** est obligatoire et détermine le nom du contrôle. Sa valeur est le nom utilisé lors du transfert du couple $^{\text{nom}}/_{\text{valeur}}$ vers le serveur.
  - L'attribut **`size`**, prennant pour valeur un nombre entier, représente le nombre de choix visible à l'écran : si la valeur de **`size`** est supérieur ou égal au nombre de choix, ce contrôle apparaîtra sous la forme d'une liste de choix *(et plus un menu déroulant)*.
  - L'attribut **`disabled`**, ne prenant pas de valeur, permet de désactiver le contrôle *(utile conjointement à* **JavaScript***)*.

- L'attribut ***multiple***, ne prenant pas de valeur, permet au client de sélectionner plusieurs valeurs dans le menu déroulant.

- Pour l'élément option :
  - L'attribut ***value*** représente la valeur qui se transmise au serveur dans le cas où ce choix est sélectionné lors de la soumission du formulaire.
  - L'attribut ***selected***, ne prenant pas de valeur, sélectionne le choix par défaut lors du chargement du formulaire.
  - L'attribut ***disabled***, ne prennant pas de valeur, désactive la ligne correspondante : le client ne pourra pas sélectionner ce choix.

E. L'élément textarea :

L'élément textarea définit un champ de texte sur plusieurs lignes permettant au client de saisir des commentaires. Voici les attributs disponible pour cette commande :

- L'attribut ***name*** détermine le nom du contrôle ; il sert lors du transfert du couple $^{\text{nom}}/_{\text{valeur}}$ vers le serveur.
- L'attribut ***row*** prend une valeur entière et il détermine le nombre de lignes constituant le champ de texte affiché.
- L'attribut ***cols***, prenant une valeur entière, détermine le nombre de colonnes constituant le champ de texte affiché.
- L'attribut ***disabled***, ne prenant pas de valeur, désactive le contrôle et le couple *(sera utile par utilisation conjointe de JavaScrip)*.
- L'attribut ***readonly***, ne prenant pas de valeur, empêche le client de modifier la valeur du contrôle ; le couple $^{\text{nom}}/_{\text{valeur}}$ sera quand même transféré au serveur lors de la soumission du serveur.

X. Un peu plus loin :

A. Les styles CSS :

Les styles CSS est le standard utilisé actuellement pour mettre en forme une page Web. Les styles CSS seront étudiés dans la prochaine partie de la formation.

Signalons seulement qu'une feuille de style interne est défini par l'élément style, défini par ses deux balises. Elle contient des règles de styles qui seront utilisées par les éléments de la page.

B. Les scripts :

Les scripts sont des bouts de programme inclus dans le code HTML. Actuellement, le standard des scripts embarqués dans une page Web est le **JavaScript**.

On ne rentrera pas dans le détail du **JavaScript** ici, car il fera l'objet d'une attention particulière dans la suite de la formation. On signale au passage qu'il est défini par l'élément script défini par ses deux balises `<script>...</script>`.

XI. Les couleurs du Web :

A. Définition :

Les pages Web affichent les couleurs codées au format RGB sur 24 bits :

- le format RGB *(Red Green Blue)* est la représentation d'une couleur dans un système additif où on mélange une quantité de rouge, de vert et de rouge.

- Chaque couleur de pixels sera codée sur 24 bits. Chaque composante *(le rouge, le vert, le bleu)* de la couleur sera représentée sur un octet *(un octet est une représentation de 8 bits)*. Hors un octet représente un nombre compris :
  - 0 et 255 en base décimal,
  - 00 et FF en notation héxadécimale.

Ainsi, les couleurs seront codées par une chaîne de caractères de la forme suivante :
"#RRGGBB" Où *RR*, *GG*, *BB* représentent l'octet représentant respectivement la quantité de rouge, de vert, de bleu composant la couleur.

Vous trouverez au cours des exercices une page Web "*Couleurs.html*" vous permettant de choisir facilement le code de la couleur désirée. Vous trouverez dans le paragraphe suivant quelques exemples de couleurs et leurs codes correspondants.

Ce format s'appelle communément le *standart RGB (sRGB)*.

Dans le Cd de la formation, vous trouverez une page Web à l'emplacement :

d-exerciceHtmlCss ⇝ CouleursWebs.html

Cette page présente une palette de couleurs ; en laissant votre curseur sur une couleur, vous apercevrez le code correspondant. Cette page facilitera le choix de votre couleur.

B. Des couleurs et des noms :

Le langage HTML propose seize noms, dans le standard W3C, associés à des couleurs ; cela permet, en autre, d'avoir la même couleur à des endroits différents.

	Nom	code
	black	#000000
	silver	#C0C0C0
	gray	#808080
	white	#FFFFFF
	maroon	#800000
	red	#FF0000
	purple	#800080
	fuchsia	#008080

	Nom	code
	green	#008000
	lime	#00FF00
	olive	#808000
	yellow	#FFFF00
	navy	#000080
	blue	#0000FF
	teal	#008080
	aqua	#00FFFF

Puis, apparue une palette plus large de couleurs :

	Nom	Code
	AliceBlue	#F0F8FF
	Aquamarine	#7FFFD4
	Bisque	#FFE4C4
	Blue	#0000FF
	BurlyWood	#DEB887
	Chocolate	#D2691E
	Cornsilk	#FFF8DC
	DarkBlue	#00008B
	DarkGray	#A9A9A9
	DarkMagenta	#8B008B
	DarkOrchid	#9932CC
	DarkSeaGreen	#8FBC8F
	DarkTurquoise	#00CED1
	DeepSkyBlue	#00BFFF
	Feldspar	#D19275
	ForestGreen	#228B22
	GhostWhite	#F8F8FF
	Gray	#808080
	HoneyDew	#F0FFF0
	Indigo	#4B0082
	Lavender	#E6E6FA
	LemonChiffon	#FFFACD
	LightCyan	#E0FFFF
	LightGreen	#90EE90
	LightSeaGreen	#20B2AA
	LightSlateGray	#778899

	Nom	Code
	AntiqueWhite	#FAEBD7
	Azure	#F0FFFF
	Black	#000000
	BlueViolet	#8A2BE2
	CadetBlue	#5F9EA0
	Coral	#FF7F50
	Crimson	#DC143C
	DarkCyan	#008B8B
	DarkGreen	#006400
	DarkOliveGreen	#556B2F
	DarkRed	#8B0000
	DarkSlateBlue	#483D8B
	DarkViolet	#9400D3
	DimGray	#696969
	FireBrick	#B22222
	Fuchsia	#FF00FF
	Gold	#FFD700
	Green	#008000
	HotPink	#FF69B4
	Ivory	#FFFFF0
	LavenderBlush	#FFF0F5
	LightBlue	#ADD8E6
	LightGoldenRodYellow	#FAFAD2
	LightPink	#FFB6C1
	LightSkyBlue	#87CEFA
	LightSteelBlue	#B0C4DE

	Nom	Code
	Aqua	#00FFFF
	Beige	#F5F5DC
	BlanchedAlmond	#FFEBCD
	Brown	#A52A2A
	Chartreuse	#7FFF00
	CornflowerBlue	#6495ED
	Cyan	#00FFFF
	DarkGoldenRod	#B8860B
	DarkKhaki	#BDB76B
	Darkorange	#FF8C00
	DarkSalmon	#E9967A
	DarkSlateGray	#2F4F4F
	DeepPink	#FF1493
	DodgerBlue	#1E90FF
	FloralWhite	#FFFAF0
	Gainsboro	#DCDCDC
	GoldenRod	#DAA520
	GreenYellow	#ADFF2F
	IndianRed	#CD5C5C
	Khaki	#F0E68C
	LawnGreen	#7CFC00
	LightCoral	#F08080
	LightGrey	#D3D3D3
	LightSalmon	#FFA07A
	LightSlateBlue	#8470FF
	LightYellow	#FFFFE0

	Lime	#00FF00
	Magenta	#FF00FF
	MediumBlue	#0000CD
	MediumSeaGreen	#3CB371
	MediumTurquoise	#48D1CC
	MintCream	#F5FFFA
	NavajoWhite	#FFDEAD
	Olive	#808000
	OrangeRed	#FF4500
	PaleGreen	#98FB98
	PapayaWhip	#FFEFD5
	Pink	#FFC0CB
	Purple	#800080
	RoyalBlue	#4169E1
	SandyBrown	#F4A460
	Sienna	#A0522D
	SlateBlue	#6A5ACD
	SpringGreen	#00FF7F
	Teal	#008080
	Turquoise	#40E0D0
	Wheat	#F5DEB3
	Yellow	#FFFF00

	LimeGreen	#32CD32
	Maroon	#800000
	MediumOrchid	#BA55D3
	MediumSlateBlue	#7B68EE
	MediumVioletRed	#C71585
	MistyRose	#FFE4E1
	Navy	#000080
	OliveDrab	#6B8E23
	Orchid	#DA70D6
	PaleTurquoise	#AFEEEE
	PeachPuff	#FFDAB9
	Plum	#DDA0DD
	Red	#FF0000
	SaddleBrown	#8B4513
	SeaGreen	#2E8B57
	Silver	#C0C0C0
	SlateGray	#708090
	SteelBlue	#4682B4
	Thistle	#D8BFD8
	Violet	#EE82EE
	White	#FFFFFF
	YellowGreen	#9ACD32

	Linen	#FAF0E6
	MediumAquaMarine	#66CDAA
	MediumPurple	#9370D8
	MediumSpringGreen	#00FA9A
	MidnightBlue	#191970
	Moccasin	#FFE4B5
	OldLace	#FDF5E6
	Orange	#FFA500
	PaleGoldenRod	#EEE8AA
	PaleVioletRed	#D87093
	Peru	#CD853F
	PowderBlue	#B0E0E6
	RosyBrown	#BC8F8F
	Salmon	#FA8072
	SeaShell	#FFF5EE
	SkyBlue	#87CEEB
	Snow	#FFFAFA
	Tan	#D2B48C
	Tomato	#FF6347
	VioletRed	#D02090
	WhiteSmoke	#F5F5F5

XI. Les caractères HTML et ASCII :

Une guerre de technologie a toujours eu lieu pour imposer un encodage de caractère.

Toutes données en informatique sont des nombres. Alors comment garder en mémoire des caractères. Il suffit de créer une table de correspondance entre nombres et caractères. En voici un exemple tout simple :

$$A \leftrightsquigarrow 1,\ B \leftrightsquigarrow 2,\ C \leftrightsquigarrow 3 \quad \ldots \quad Z \leftrightsquigarrow 26$$

L'informatique a connu son essort sur le continent américain et ils créèrent une table de correspondance appelé table de correspondance ASCII *(American Standard code for Information Interchange)*. Mais la première version de cette table ne comportait pas les caractères accentués, une seconde table ASCII vit alors le jour. Puis, il fallu intégrer ensuite les caractères asiatiques et arabes, le format Unicode vit alors le jour.

Sans rentré dans les détails, nous utiliserons *(sans s'en rendre compte)* de la norme ISO-8859-1 aussi appelé "*latin-1*".

&#00; à &#08;		Non utilisé	
	Saut de ligne	

		Tabulation	
&#11; à		Non utilisé	

&#13;		Retour chariot	
&#32;		Espace	
&#34;	"	Guillemet droit	
&#36;	$	Signe dollar	
&#38;	&	Esperluette	
&#40;		Parenthèse ouvrante	
&#42;	*	Astérisque	
&#44;	,	Virgule	
&#46;	.	Point	
0 à 9	Les chiffres de 0 à 9		
&#59;	;	Point-virgule	
&#61;	=	Signe "égal à"	
&#63;	?	Point d'interrogation	
A à Z	Les lettres majuscules de A à Z		
&#92;	\	Barre oblique inverse	
&#94;	^	Accent circonflexe	
&#96;	`	Accent grave- Apostrophe inverse	
&#123;	{	Accolade ouvrante	
&#125;	}	Accolade fermante	
&#127; à Ÿ		Non utilisé	
&#161;	¡	Point d'exclamation renversé	&iexcl;
&#163;	£	Signe livre sterling	&pound;
&#165;	¥	Signe yen	&yen;
&#167;	§	Signe de section	&sect;
&#169;	©	Signe Copyright	&copy;
&#171;	«	Guillemet ouvrant	&laquo;
&#173;	-	Trait d'union insécable	&shy;
&#175;	¯	Accent long	&macr;
&#177	±	Signe "Plus ou moins"	&plusmn;
&#179;	³	Puissance 3	&sup3;
&#181;	μ	Signe micron	&micro;
&#183;	·	Point médian	&middot;
&#185;	¹	Puissance 1	&sup1;
&#187;	»	Guillemet fermant	&raquo;
&#189;	½	Fraction un demi	&frac12;
&#191;	¿	Point d'interrogation renversé	&iquest;
&#193;	Á	A majuscule avec accent aigu	&Aacute;
&#195;	Ã	A majuscule avec tilde	&Atilde;

&#14; à &#31;		Non utilisé	
&#33;	!	Point d'exclamation	
&#35;	#	Signe dièse	
&#37;	%	Signe pourcentage	
&#39;	'	Apostrophe	
&#41;		Parenthèse fermante	
&#43;	+	Signe d'addition	
&#45;	-	Tiret	
&#47;	/	Barre oblique	
&#58;	:	Deux-points	
&#60;	<	Signe "inférieur à"	
&#62;	>	Signe "supérieur à"	
&#64;	@	Arobas	
&#91;	[	Crochet ouvrant	
&#93;	]	Crochet fermant	
&#95;	_	Tiret bas - Trait de soulignement	
a à z	Les lettres minuscules de a à z		
&#124;	|	Barre verticale	
&#126;	~	Tilde	
		Espace insécable	
&#162;	¢	Signe cent	&cent;
&#164;	※	Signe monnaie	&curren;
&#166;	¦	Barre verticale courte	&brvbar;
&#168;	¨	Umlaut - Tréma	&uml;
&#170;	ª	Caractère ordinal, marque du féminin	ª]
&#172;	¬	Trait d'union conditionnel	&not;
&#174;	®	Signe marque déposée	&reg;
&#176;	º	Signe degré	&deg;
&#178;	²	Puissance 2	&sup2;
&#180;	´	Accent aigu	&acute;
&#182;	¶	Signe de paragraphe	&para;
&#184;	¸	Cédille	&cedil;
&#186;	º	Caractère ordinal, marque du masculin	&ordm;
&#188;	¼	Fraction un quart	&frac14;
&#190;	¾	Fraction trois quarts	&frac34;
&#192;	À	A majuscule avec accent grave	&Agrave;
&#194;	Â	A majuscule avec accent circonflexe	&Acirc;
&#196;	Ä	A majuscule avec tréma ou Umlaut	&Auml;

&#197;	Å	A majuscule avec accent circulaire	&Aring;
&#199;	Ç	C majuscule avec cédille	&Ccedil;
&#201;	É	E majuscule avec accent aigu	&Eacute;
&#203;	Ë	E majuscule avec tréma ou Umlaut	&Euml;
&#205;	Í	I majuscule avec accent aigu	&Iacute;
&#207;	Ï	I majuscule avec tréma ou Umlaut	&Iuml;
&#209;	Ñ	N majuscule avec tilde	&Ntilde;
&#211;	Ó	O majuscule avec accent aigu	&Oacute;
&#213;	Õ	O majuscule avec tilde	&Otilde;
&#215;	×	Signe de multiplication	&times
&#217;	Ù	U majuscule avec accent grave	&Ugrave;
&#219;	Û	U majuscule avec accent circonflexe	&Ucirc;
&#221;	Ý	Y majuscule avec accent aigu	&Yacute;
&#223;	ß	double s minuscule, caractère allemand, ligature sz	&szlig;
&#225;	á	a minuscule avec accent aigu	&aacute;
&#227;	ã	a minuscule avec tilde	&atilde;
&#229;	å	a minuscule avec accent circulaire	&aring;
&#231;	ç	c minuscule avec cédille	&ccedil;
&#233;	é	e minuscule avec accent aigu	&eacute‘;
&#235;	ë	e minuscule avec tréma ou Umlaut	&euml;
&#237;	í	i minuscule avec accent aigu	&iacute;
&#239;	ï	i minuscule avec tréma ou Umlaut	&iuml;
&#241;	ñ	n minuscule avec tilde	&ntilde;
&#243;	ó	o minuscule avec accent aigu	&oacute;
&#245;	õ	o minuscule avec tilde	&otilde;
&#247;	÷	Signe de division	&divide
&#249;	ù	u minuscule avec accent grave	&ugrave;
&#251;	û	u minuscule avec accent circonflexe	&ucirc;
&#253;	ý	y minuscule avec accent aigu	&yacute;
&#255;	ÿ	minuscule avec tréma ou Umlaut	&yuml;

&#198;	Æ	Diphtongue AE majuscule, ligature	&AElig;
&#200;	È	E majuscule avec accent grave	&Egrave;
&#202;	Ê	E majuscule avec accent circonflexe	&Ecirc;
&#204;	Ì	I majuscule avec accent grave	&Igrave;
&#206;	Î	I majuscule avec accent circonflexe	&Icirc;
&#208;	ð	ETH majuscule, caractère islandais	&ETH;
&#210;	Ò	O majuscule avec accent grave	&Ograve;
&#212;	Ô	O majuscule avec accent circonflexe	&Ocirc;
&#214;	Ö	O majuscule avec tréma ou Umlaut	&Ouml;
&#216;	Ø	O majuscule barré	&Oslash;
&#218;	Ú	U majuscule avec accent aigu	&Uacute;
&#220;	Ü	U majuscule avec tréma ou Umlaut	&Uuml;
&#222;	Þ	THORN majuscule, caractère islandais	&THORN;
&#224;	à	a minuscule avec accent grave	&agrave;
&#226;	â	a minuscule avec accent circonflexe	&acirc;
&#228;	ä	a minuscule avec tréma ou Umlaut	&auml;
&#230;	æ	Diphtongue ae minuscule, ligature	&aelig;
&#232;	è	e minuscule avec accent grave	&egrave;
&#234;	ê	e minuscule avec accent circonflexe	&ecirc;
&#236;	ì	i minuscule avec accent grave	&igrave;
&#238;	î	i minuscule accent circonflexe	&icirc;
&#240;	ð	eth minuscule, caractère islandais	&eth;
&#242;	ò	o minuscule avec accent grave	&ograve;
&#244;	ô	o minuscule avec accent circonflexe	&ocirc;
&#246;	ö	o minuscule avec tréma ou Umlaut	&ouml;
&#248;	ø	o minuscule barré	&oslash;
&#250;	ú	u minuscule avec accent aigu	&uacute;
&#252;	ü	u minuscule avec tréma ou Umlaut	&uuml;
&#254;	þ	thorn minuscule, caractère islandais	&thorn;

XII. Les événements HTML :

A. Présentation :

Un événement est un message envoyé par un objet à un programme afin que ce dernier puisse réagir :

- Le clic sur un bouton entraînera le lancement d'un programme ou d'une routine.

- Le passage de la souris au-dessus d'un lien peut le faire changer de couleurs.

Ainsi, le code suivant :

```
1 <body onload="...">
2   <span onmouseover="...">      </span>
```

```
3    <form  onchange="...">
4
5    </form>
6  </body>
```

Détermine la reaction à trois événements :

- La page finissant de se charger lancera un script (onload)

- L'élément span réagit lorsque le curseur passera au dessus de lui (onmouseover).

- Le formulaire form lance un script chaque fois qu'un de ses contrôle subira un changement de valeur (onchange)

B. Liste des événements HTML :

Evénements des fenêtres - appliquable qu'à l'élément body

Attributs	Désignation
onload	Exécutera le script à la fin du chargement de la page
onunload	Exécutera le script à la fermeture de la page

Evénements des formulaires

onchange	Exécutera le script lorsque le contrôle associé à l'élément changera de valeur (*input*, *select*, *textarea*)
onsubmit	Exécutera le script lors de la soumission du formulaire (*form*)
onreset	Exécutera le script lors de la soumission du formulaire (*form*)
onselect	Exécutera le script lorsque l'élément est sélectionné (*input*, *élementtextarea*)
onfocus	Exécutera le script lorsque l'élément prend le focus (*input*, *select*, *select*, *textarea*, *button*)
onblur	Exécutera le script lorsque l'élément perd le focus *(agit sur les même éléments que onfocus)*

Evénements liés au clavier

onkeydown	Exécutera le script lorsque une touche du clavier sera enfoncée
onkeypress	Exécutera le script lorsqu'une touche du clavier sera enfoncé et relâché
onkeyup	Exécutera le script lorsque la touche sera relachée

Evénements liés à la souris

onclick	Exécutera le script lorsqu'on cliquera sur l'élément
ondbclick	Exécutera le script lorsqu'on effectuera un double clic sur l'élément
onmousedown	Exécutera le script lorsqu'on laissera enfoncer le bouton de la souris
onmousemove	Exécutera le script lorsque la souris bougera au dessus de l'élément
onmouseover	Exécutera le script lorsque la souris survolera l'élément
onmouseout	Exécutera le script lorsque le curseur sortira de lélément
onmouseup	Exécutera le script lorsque le bouton de la souris sera relâché

XIII. Toutes les éléments HTML :

A

A accesskey, charset, class, coords, dir, href, hreflang, id, lang, name, onblur, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, rel, rev, shape, style, tabindex, target, title, type	ABBR class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title
ACRONYM class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	ADDRESS class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title
APPLET align, alt, archive, class, code, codebase, height, hspace, id, name, object, style, title, vspace, width	AREA accesskey, alt, class, coords, dir, href, id, lang, nohref, onblur, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, shape, style, tabindex, target, title

B

B	BASE
class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	href, target
BASEFONT	**BDO**
class, color, face, id, lang, size	class, dir, id, lang, style, title
BIG	**BLOCKQUOTE**
class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	cite, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title
BODY	**BR**
alink, background, bgcolor, class, dir, id, lang, link, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onload, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, onunload, style, text, title, vlink	class, clear, id, style, title
BUTTON	
accesskey, class, dir, disabled, id, lang, name, onblur, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, tabindex, title, type, value	

C

CAPTION align, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	CENTER class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title
CITE class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	CODE class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title
COL align, char, charoff, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, span, style, title, valign, width	COLGROUP align, char, charoff, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, span, style, title, valign, width

D

DD class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	DEL cite, class, datetime, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title
DFN class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	DIR class, compact, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

DIV

align, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

DL

class, compact, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

DT

class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

E

EM

class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

F

FIELDSET

class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

FONT

class, color, dir, face, id, lang, size, style, title

FORM

accept-charset, accept, action, class, dir, enctype, id, lang, method, name, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, onreset, onsubmit, style, target, title

FRAME

class, frameborder, id, longdesc, marginheight, marginwidth, name, noresize, scrolling, src, style, title

FRAMES
onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup

FRAMESET
class, cols, dir, id, onload, onunload, rows, style, title

H

H1
align, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

H2
align, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

H3
align, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

H4
align, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

H5
align, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

H6
align, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

HEAD
dir, lang, profile

HR
align, class, dir, id, lang, noshade, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, size, style, title, width

HTML
dir, lang, version

I

I	IFRAME
class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	align, class, frameborder, height, id, longdesc, marginheight, marginwidth, name, scrolling, src, style, title, width
IMG	INPUT
align, alt, border, class, dir, height, hspace, id, ismap, lang, longdesc, name, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, src, style, title, usemap, vspace, width	accept, accesskey, align, alt, checked, class, dir, disabled, id, ismap, lang, maxlength, name, onblur, onchange, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, onselect, readonly, size, src, style, tabindex, title, type, usemap, value
INS	ISINDEX
cite, class, datetime, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	class, dir, id, lang, prompt, style, title

K

KBD
class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

L

LABEL

accesskey, class, dir, for, id, lang, onblur, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

LEGEND

accesskey, align, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

LI

class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title, type, value

LINK

charset, class, dir, href, hreflang, id, lang, media, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, rel, rev, style, target, title, type

M

MAP

class, dir, id, lang, name, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

MENU

class, compact, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

META

content, dir, http-equiv, lang, name, scheme

N

NOFRAMES

class, dir, id, lang, style, title

NOSCRIPT

class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

O

OBJECT

align, archive, border, class, classid, codebase, codetype, data, declare, dir, height, hspace, id, lang, name, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, standby, style, tabindex, title, type, usemap, vspace, width

OL

class, compact, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, start, style, title, type

OPTGROUP

class, dir, disabled, id, label, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

OPTION

class, dir, disabled, id, label, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, selected, style, title, value

P

P

align, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

PARAM

id, name, type, value, valuetype

PRE

class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title, width

Q

Q

cite, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

S

S class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	SAMP class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title
SCRIPT charset, defer, language, src, type	SELECT class, dir, disabled, id, lang, multiple, name, onblur, onchange, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, size, style, tabindex, title
SMALL class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	SPAN class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title
STRIKE class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	STRONG class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title
STYLE dir, lang, media, title, type	SUB class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title
SUP class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	

T

TABLE align, bgcolor, border, cellpadding, cellspacing, class, dir, frame, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, rules, style, summary, title, width	TBODY align, char, charoff, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title, valign
TD abbr, align, axis, bgcolor, char, charoff, class, colspan, dir, headers, height, id, lang, nowrap, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, rowspan, scope, style, title, valign, width	TEXTAREA accesskey, class, cols, dir, disabled, id, lang, name, onblur, onchange, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, onselect, readonly, rows, style, tabindex, title
TFOOT align, char, charoff, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title, valign	TH abbr, align, axis, bgcolor, char, charoff, class, colspan, dir, headers, height, id, lang, nowrap, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, rowspan, scope, style, title, valign, width
THEAD align, char, charoff, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title, valign	TITLE dir, lang

TR
align, bgcolor, char, charoff, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title, valign

TT
class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

U

U
class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

UL
class, compact, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title, type

V

VAR
class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title

XIV. Tous les attributs des éléments :

A

abbr	TD, TH
accept	FORM, INPUT
action	FORM
alink	BODY
archive	APPLET, OBJECT

accept-charset	FORM
accesskey	A, AREA, BUTTON, INPUT, LABEL, LEGEND, TEXTAREA
align	CAPTION, APPLET, IFRAME, IMG, INPUT, OBJECT, LEGEND, TABLE, HR, DIV, H1, H2, H3, H4, H5, H6, P, COL, COLGROUP, TBODY, TD, TFOOT, TH, THEAD, TR
alt	APPLET, AREA, IMG, INPUT
axis	TD, TH

B

background	BODY
border	TABLE, IMG, OBJECT

bgcolor	TABLE, TR, TD, TH, BODY

C

cellpadding	TABLE
char	COL, COLGROUP, TBODY, TD, TFOOT, TH, THEAD, TR
charset	A, LINK, SCRIPT
cite	BLOCKQUOTE, Q, DEL, INS

cellspacing	TABLE
charoff	COL, COLGROUP, TBODY, TD, TFOOT, TH, THEAD, TR
checked	INPUT
class	Tous les éléments HTML excepté BASE, BASEFONT, HEAD, HTML, META, PARAM, SCRIPT, STYLE, TITLE

classid	OBJECT
code	APPLET
codetype	OBJECT
cols	FRAMESET, TEXTAREA
compact	DIR, DL, MENU, OL, UL
coords	AREA, A

clear	BR
codebase	OBJECT, APPLET
color	BASEFONT, FONT
colspan	TD, TH
content	META

D

data	OBJECT
declare	OBJECT
dir	Tous les éléments HTML excepté APPLET, BASE, BASEFONT, BDO, BR, FRAME, FRAMESET, IFRAME, PARAM, SCRIPT, BDO

datetime	DEL, INS
defer	SCRIPT
disabled	BUTTON, INPUT, OPTGROUP, OPTION, SELECT, TEXTAREA

E

enctype	FORM

F

face	BASEFONT, FONT
frame	TABLE

for	LABEL
frameborder	FRAME, IFRAME

H

headers	TD, TH
href	A, AREA, LINK, BASE
hspace	APPLET, IMG, OBJECT

height	IFRAME, TD, TH, IMG, OBJECT, APPLET
hreflang	A, LINK
http-equiv	META

I

id	Tous les éléments HTML excepté BASE, HEAD, HTML, META, SCRIPT, STYLE, TITLE

ismap	IMG, INPUT

L

label	OPTION, OPTGROUP
language	SCRIPT
longdesc	IMG, FRAME, IFRAME

lang	Tous les éléments HTML excepté APPLET, BASE, BASEFONT, BR, FRAME, FRAMESET, IFRAME, PARAM, SCRIPT
link	BODY

M

marginheight	FRAME, IFRAME
maxlength	INPUT
method	FORM

marginwidth	FRAME, IFRAME
media	STYLE, LINK
multiple	SELECT

N

name	BUTTON, TEXTAREA, APPLET, SELECT, FORM, FRAME, IFRAME, IMG, A, INPUT, OBJECT, MAP, PARAM, META
noresize	FRAME
nowrap	TD, TH

nohref	AREA
noshade	HR

O

object	APPLET
onchange	INPUT, SELECT, TEXTAREA
onfocus	A, AREA, BUTTON, INPUT, LABEL, SELECT, TEXTAREA
onblur	A, AREA, BUTTON, INPUT, LABEL, SELECT, TEXTAREA
onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup	Tous les éléments HTML excepté APPLET, BASE, BASEFONT, BDO, BR, FONT, FRAME, FRAMESET, HEAD, HTML, IFRAME, ISINDEX, META, PARAM, SCRIPT, STYLE, TITLE
onload	FRAMESET,BODY

onreset	FORM
onsubmit	FORM

onselect	INPUT, TEXTAREA
onunload	FRAMESET, BODY

P

profile	HEAD

prompt	ISINDEX

R

readonly	TEXTAREA, INPUT
rev	A, LINK
rowspan	TD, TH

rel	A, LINK
rows	FRAMESET, TEXTAREA
rules	TABLE

S

scheme	META
scrolling	FRAME, IFRAME
shape	AREA, A
span	COL,COLGROUP
standby	OBJECT
style	Tous les éléments HTML excepté BASE, BASEFONT, HEAD, HTML, META, PARAM, SCRIPT, STYLE, TITLE

scope	TD, TH
selected	OPTION
size	HR, FONT, INPUT, BASEFONT, SELECT
src	SCRIPT, INPUT, FRAME, IFRAME, IMG
start	OL
summary	TABLE

T

tabindex	A, AREA, BUTTON, INPUT, OBJECT, SELECT, TEXTAREA
text	BODY

target	A, AREA, BASE, FORM, LINK
title	Tous les éléments HTML excepté BASE, BASEFONT, HEAD, HTML, META, PARAM, SCRIPT, TITLE

type	A, LINK, OBJECT, PARAM, SCRIPT, STYLE, INPUT, LI, OL, UL, BUTTON

U

usemap	IMG, INPUT, OBJECT

V

valign	COL, COLGROUP, TBODY, TD, TFOOT, TH, THEAD, TR
valuetype	PARAM
vlink	BODY

value	INPUT, OPTION, PARAM, BUTTON, LI
version	HTML
vspace	APPLET, IMG, OBJECT

W

width	HR, IFRAME, IMG, OBJECT, TABLE, TD, TH, APPLET, COL, COLGROUP, PRE

Index