



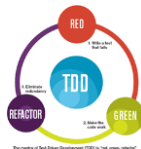
Formation Développeur Expert Java

HTTP & HTML5

- Création d'IOcean en 2001
- 2 Implantations :
 - Siège à Montpellier
 - Agence Paris
- 2 Activités :
 - ESN (SSII)
 - Editeur
- En 2016, 30 personnes pour 2,2 M€



Hudson



« Votre passeport pour l'emploi numérique »

Votre formateur



Les **projets informatiques** dont
les clients parlent avec plaisir

Sommaire

- HTTP

- Histoire
- Le protocole HTTP
- L'URL
- La requête HTTP
- La réponse HTTP
- REST

- HTML5

- Histoire
- Définition
- Le document
- Le contenu texte
- Les liens
- Les tableaux
- Les listes
- Les formulaires
- Le multimédia

HTTP

Histoire

1969, Naissance d'ARPANET premier réseau d'ordinateur, créé par la DARPA une agence de l'armée américaine

1972, Naissance du courrier électronique

1974, Naissance de TCP/IP ensemble de protocoles pour le transfert de données, adopté par ARPANET en 1983

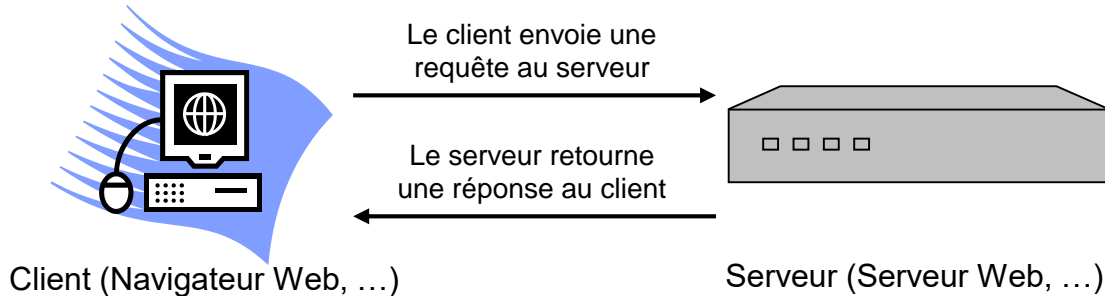
Fin 1980 - Début 1990, Naissance du Web avec les pages HTML, les URL et le protocole HTTP

Le protocole HTTP

Le protocole HTTP (pour HyperText Transfer Protocol) est le principal protocole utilisé sur Internet et permet de définir le mode d'échange entre un navigateur et un serveur sur le Web. Son but est de permettre à un client **d'envoyer des commandes** à un serveur pour qu'il les exécute et retourne une réponse au client.

Une commande permet notamment de récupérer un fichier qui peut être de différents types (HTML, CSS, JavaScript, Images, ...) dont **la localisation** est définie grâce à une chaîne de caractères appelée **URL**.

Le fonctionnement du protocole HTTP est de type client-serveur :



Le protocole HTTP

Depuis la version 1.0 du protocole, le protocole est également utilisé pour transférer des messages avec des en-têtes décrivant le contenu du message en utilisant un codage de type **MIME** (pour Multipurpose Internet Mail Extensions).

Il existe une variante du protocole HTTP, le protocole **HTTPS** (S pour « Sécurisé »).

Il fonctionne exactement de la même façon mais les données envoyées entre le client et le serveur sont chiffrées de telle sorte qu'elles ne puissent pas être lues par un tiers.

Une clé de cryptage est partagée entre le client et le serveur.

L'URL

L'URL (pour Uniform Resource Locator) permet d'identifier de manière unique une ressource sur Internet.

Ces ressources peuvent être des fichiers de différents types (HTML, CSS, JavaScript, Images, ...) mais également des adresses mails ou des « objets » applicatifs.

L'URL aussi appelée « adresse internet » se décompose ainsi :

{protocole}://[{utilisateur}:{mot de passe}]@{**adresse du serveur**}[:{port}][/{chemin d'accès ressource}][?{paramètre1=valeur1}&{paramètre2=valeur2}&{paramètre...=valeur...}][#{informati on pour le client}]

http://www.iocean.fr

protocole sous-domaine domaine TLD

L'URL

« protocole » (obligatoire) = Nom du protocole utilisé (par exemple http, https, mailto, ftp,)

« utilisateur » et « mot de passe » (facultatif) = Compte utilisateur et mot de passe pour se connecter au serveur (par exemple pour se connecter à un serveur de mail). Il est très fortement déconseillé de passer ces infos dans une URL pour des raisons évidentes de sécurité

« adresse du serveur » (obligatoire) = Nom du serveur sur le réseau ou adresse IP (par exemple www.iocean.fr, 10.124.128.2, ...)

« port » (facultatif) = Port du protocole si ce n'est pas le port par défaut (par exemple 8080 au lieu de 80 pour le protocole http)

« chemin d'accès ressource » (facultatif) = Chemin d'accès à la ressource sur le serveur (par exemple dir1/sous-dir2/fichier3.html)

« paramètrex » et « valeurx » (facultatif) = Ensemble des paramètres passé au serveur (par exemple « ?id_client=123&fg_etat=test »)

« information pour le client » (facultatif) = Information non prise en compte par le serveur mais pouvant influencer le retour de la requête par le client (par exemple le texte peut correspondre à une ancre dans la page HTML pour que le navigateur se positionne à cet endroit)

L'URL



L'URL permet donc de passer des valeurs de paramètres. Or ces valeurs peuvent contenir des caractères spéciaux ou réserver par le protocole (comme le « / » par exemple).

Dans ce cas, il est nécessaire d'encoder le contenu spécial avec le caractère « % » suivi de la valeur ASCII en hexadécimal.

Exemple :

Caractère	Codage	Caractère	Codage	Caractère	Codage
Tabulation					
n	%09	+	%2B	@	%40
Espace	%20	,	%2C	[%5B
!	%21	.	%2E	\	%5C
"	%22	/	%2F]	%5D
#	%23	:	%3A	^	%5E
%	%25	;	%3B	'	%60
&	%26	<	%3C	{	%7B
(%28	=	%3D		%7C
)	%29	>	%3E	}	%7D
*	%2A	?	%3F	~	%7E

La requête HTTP

Une requête HTTP correspond à l'ensemble des lignes envoyés au serveur par le client. Elle comprend :

Une ligne de requête = Précise le type de document demandé, la méthode à appliquer et la version du protocole chacun séparé par un espace.

Exemple : GET <http://www.iocean.fr> HTTP/1.0

L'en-tête de la requête = Ensemble de lignes permettant de donner des informations supplémentaires sur le client et la requête. Chaque ligne contient un nom précisant le type d'en-tête suivi de deux points (:) et de la valeur de l'en-tête

Exemple :

Accept : text/html

User-Agent : Mozilla/4.0 (compatible; MSIE 5.0; Windows 95)

Le corps de la requête = Ensemble de lignes optionnelles devant être séparées des lignes précédentes par une ligne vide et permettant par exemple un envoi de données par une commande POST lors de l'envoi de données au serveur par un formulaire

La requête HTTP

Les principales commandes HTTP sont :

GET : Demande la ressource située à l'URL spécifiée

OPTIONS : Demande les méthodes traitées par la ressource située à l'URL spécifiée

HEAD : Demande les informations de l'en-tête de la ressource située à l'URL spécifiée

POST : Envoie de données au serveur concernant la ressource situé à l'URL spécifiée

PUT : Enregistre une nouvelle ressource situé à l'URL spécifiée

DELETE : Supprime la ressource située à l'URL spécifiée

TRACE : Retourne ce qui a été envoyé au serveur (sorte de commande echo)

La requête HTTP

Les principaux entêtes d'une requête HTTP sont :

Accept : Type de contenu accepté par le client (par exemple text/html)

Accept-Charset : Jeu de caractères attendu par le client

Accept-Encoding : Codage de données accepté par le client

Accept-Language : Langage attendu par le client

Authorization : Identification du client auprès du serveur

Content-Encoding : Type de codage du corps de la requête

Content-Language : Type de langage du corps de la requête

Content-Length : Longueur du corps de la requête

Content-Type : Type de contenu du corps de la requête (par exemple text/html)

Date : Date de début de transfert des données

Forwarded : Utilisé par les machines intermédiaires entre le client et le serveur

From : Permet de spécifier l'adresse e-mail du client

If-Modified-Since : Permet de spécifier que le document doit être envoyé s'il a été modifié depuis une certaine date

Link : Relation entre deux URL

Orig-URL : URL d'origine de la requête

Referer : URL du lien à partir duquel la requête a été effectuée

User-Agent : Chaîne donnant des informations sur le client (nom et version du navigateur, système d'exploitation)

La réponse HTTP

Une réponse HTTP correspond à l'ensemble des lignes envoyés par le serveur au client. Elle comprend :

Une ligne de statut = Précise la version du protocole utilisé et l'état du traitement de la requête à l'aide d'un code et d'un texte explicatif chacun séparé par un espace.

Exemple : HTTP/1.0 200 OK

L'en-tête de la réponse = Ensemble de lignes permettant de donner des informations supplémentaires sur le serveur et la réponse. Chaque ligne contient un nom précisant le type d'en-tête suivi de deux points (:) et de la valeur de l'en-tête

Exemple :

Date : Sat, 12 Mar 2018 15:47:00 GMT

Server : Microsoft-IIS/2.0

Le corps de la réponse = Contient le contenu demandé par la requête. Par exemple, le fichier HTML.

La réponse HTTP

Les statuts de réponse HTTP se regroupent par le premier chiffre du code :

1xx : Message d'information

2xx : Réussite (indiquent le bon déroulement de la transaction)

3xx : Redirection (indiquent que la ressource n'est plus à l'emplacement indiqué)

4xx : Erreur due au client (indiquent que la requête est incorrecte)

5xx : Erreur due au serveur (indiquent qu'il y a eu une erreur interne du serveur)

La réponse HTTP

Les principaux statuts de réponse HTTP sont :

200 OK : La requête a été accomplie correctement

201 CREATED : Elle suit une commande POST, elle indique la réussite, le corps du reste du document est sensé indiquer l'URL à laquelle le document nouvellement créé devrait se trouver.

202 ACCEPTED : La requête a été acceptée, mais la procédure qui suit n'a pas été accomplie

203 PARTIAL INFORMATION : Lorsque ce code est reçu en réponse à une commande GET, cela indique que la réponse n'est pas complète.

204 NO RESPONSE : Le serveur a reçu la requête mais il n'y a pas d'information à renvoyer

205 RESET CONTENT : Le serveur indique au navigateur de supprimer le contenu des champs d'un formulaire

206 PARTIAL CONTENT : Il s'agit d'une réponse à une requête comportant l'en-tête range. Le serveur doit indiquer l'en-tête content-Range

301 MOVED : Les données demandées ont été transférées à une nouvelle adresse

302 FOUND : Les données demandées sont à une nouvelle URL, mais ont cependant peut-être été déplacées depuis...

303 METHOD : Cela implique que le client doit essayer une nouvelle adresse, en essayant de préférence une autre méthode que GET

304 NOT MODIFIED : Si le client a effectué une commande GET conditionnelle (en demandant si le document a été modifié depuis la dernière fois) et que le document n'a pas été modifié il renvoie ce code.

La réponse HTTP

Les principaux statuts de réponse HTTP sont :

400 BAD REQUEST : La syntaxe de la requête est mal formulée ou est impossible à satisfaire

401 UNAUTHORIZED : Le paramètre du message donne les spécifications des formes d'autorisation acceptables. Le client doit reformuler sa requête avec les bonnes données d'autorisation

402 PAYMENT REQUIRED : Le client doit reformuler sa demande avec les bonnes données de paiement

403 FORBIDDEN : L'accès à la ressource est tout simplement interdit

404 NOT FOUND : Le serveur n'a rien trouvé à l'adresse spécifiée

500 INTERNAL ERROR : Le serveur a rencontré une condition inattendue qui l'a empêché de donner suite à la demande

501 NOT IMPLEMENTED : Le serveur ne supporte pas le service demandé

502 BAD GATEWAY : Le serveur a reçu une réponse invalide de la part du serveur auquel il essayait d'accéder en agissant comme une passerelle ou un proxy

503 SERVICE UNAVAILABLE : Le serveur ne peut pas vous répondre à l'instant présent

504 GATEWAY TIMEOUT : La réponse du serveur a été trop longue vis-à-vis du temps pendant lequel la passerelle était préparée à l'attendre

La réponse HTTP



Les principaux entêtes d'une réponse HTTP sont :

Content-Encoding : Type de codage du corps de la réponse

Content-Language : Type de langage du corps de la réponse

Content-Length : Longueur du corps de la réponse

Content-Type : Type de contenu du corps de la réponse (par exemple text/html)

Date : Date de début de transfert des données

Expires : Date limite de consommation des données

Forwarded : Utilisé par les machines intermédiaires entre le client et le serveur

Location : Redirection vers une nouvelle URL associée au document

Server : Caractéristiques du serveur ayant envoyé la réponse

REST

REST (Representational State Transfer) est une architecture basée sur le protocole HTTP.

Elle permet de manipuler des entités d'une application et est principalement utilisée pour fournir des services sous forme d'API.

Chaque méthode HTTP **représente une action sur une ou plusieurs ressources** :

- **GET** pour récupérer une entité ou une liste d'entités
- **POST** pour ajouter une entité
- **PUT** pour modifier une entité
- **DELETE** pour supprimer une entité

Les entités sont représentées dans un format connu du client et du serveur comme JSON ou XML

REST

Quelques exemples d'une API REST

Action	URI	Résultat attendu
GET	/posts	Tous les posts
GET	/posts/1	Le post 1
GET	/posts/1/comments	Tous les commentaires du post 1
GET	/posts/1/comments?sort=date&order=desc	Tous les commentaires du post 1 triés par date antéchronologique
GET	/posts/1/comments?userId=5	Tous les commentaires de l'utilisateur 5 du post 1
POST	/posts	Créer un post
PUT	/posts/1	Modifier le post 1
DELETE	/posts/1	Supprimer le post 1
POST	/login	Obtention d'un token d'authentification
GET	/posts?token=fhGssD78	Tous les posts (accès sécurisé)

HTML5

Histoire

Le langage HTML (**HyperText Markup Language**) permet de décrire un document de présentation à l'aide de balises structurées. Les versions sont :

En 1991, création du HTML par Tim Berners-Lee

En 1993, version « initiale », version stable dite « 1.0 »

En 1995, 1ère version « officielle », HTML 2.0

En Janvier 1997, Version 3.2 (pas de version 3.0 officielle)

En Décembre 1997, Version 4.0 puis en Décembre 1999 une Version 4.01

En 2000, Naissance de XHTML 1.0 (pour Extensible HyperText Markup Language) puis en 2001, Version 1.1

Le 28 Octobre 2014, HTML 5.0 finalisé, compatible HTML et XHTML (Abandon en 2009 du XHTML 2.0)

Définition

Un document HTML contient des balises qui indiquent la nature du texte qu'elles encadrent. Une balise est un mot entouré des caractères '<' et '>'.

On distingue deux types de balises : les balises en paires et les balises orphelines.

Les balises en paires

Elles s'ouvrent, contiennent du texte, et se ferment plus loin.

<h1>*Ceci est un titre***</h1>**

On distingue une balise ouvrante **<h1>** et une balise fermante **</h1>** qui indique que le titre se termine. La balise fermante porte toujours le même nom que la balise ouvrante avec un '/' devant.

Les balises orphelines

Ce sont des balises qui servent le plus souvent à insérer un élément à un endroit précis,

**
**

le '/' de fin n'est pas obligatoire. On pourrait écrire seulement **
**. Néanmoins, il est recommandé de le rajouter pour une question de lisibilité.

Définition

Une balise peut contenir des attributs qui peuvent être vus comme des options qui les complètent la balise.

Les attributs se placent après le nom de la balise ouvrante (et seulement dans la balise ouvrante) et ont souvent une valeur.

Exemples :

```

```

```
<p id="parag1" class="moyen">Ceci est un paragraphe</p>
```

Les balises en paire peuvent contenir d'autres balises mais elles doivent être fermées dans le sens inverses de leur ouverture.

Exemples :

```
<div>
  <p id="parag1">
    Ceci est correct
  </p>
</div>
```

```
<div>
  <p id="parag1">
    Ceci n'est pas correct
  </div>
</p>
```


Définition

Une balise peut également servir à afficher des commentaires.

Il commence par les caractères '`<!--`' et se termine par les caractères '`-->`'.

Exemples :

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

```
<!--  
commentaire sur  
plusieurs lignes  
-->
```

Les commentaires ne sont pas affichés par le navigateur et peuvent vous servir à vous repérer dans votre code.

ATTENTION !!! Il faut bien garder à l'esprit que tout le code HTML est envoyé au navigateur et peut donc être consulté par l'utilisateur en faisant « Afficher le code source de la page ». Il est donc fortement conseillé de ne pas mettre n'importe quoi dans le code HTML ou les commentaires

Le document

Certaines balises sont obligatoires pour un document HTML. Ce sont :

<!DOCTYPE>

Cette balise indique que le document est une page HTML. Elle est positionnée à toute première ligne et commence par un '!'. Pour HTML5, le doctype est « html ».

<!DOCTYPE html>

<html>

C'est la balise principale du code. Elle englobe tout le contenu de la page. Une page html est constituée de deux parties : L'en-tête **<head>** et le corps **<body>**.

<head>

C'est la balise de l'en-tête de la page. Elle donne des informations générales sur la page comme son titre, les fichiers css et javascript à charger, etc... Les informations que contient l'en-tête ne sont en général pas affichées directement sur la page. Elle contient au minimum une balise **<title>** pour préciser le titre de la page et normalement une balise **<meta>** pour préciser le jeu de caractère de la page (en général « utf-8 »).

<body>

C'est la balise du corps de la page, c'est-à-dire le contenu qui sera affiché.

Le document

Un exemple de document HTML vide est donc :

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title>Titre</title>  
  </head>  
  <body>  
  
    </body>  
</html>
```

Vous remarquerez que les balises sont indentées. Ceci n'est pas du tout nécessaire pour que le code HTML s'affiche correctement dans le navigateur mais totalement indispensable pour qu'il soit lisible par un développeur.

Le document

Balises d'en-tête

Ces balises sont toutes situées dans l'en-tête de la page web entre `<head>` et `</head>` :

Balise	Description
<code><link /></code>	Liaison avec une feuille de style
<code><meta /></code>	Métadonnées de la page web (charset, mots-clés, etc.)
<code><script></code>	Code JavaScript
<code><style></code>	Code CSS
<code><title></code>	Titre de la page

Le document

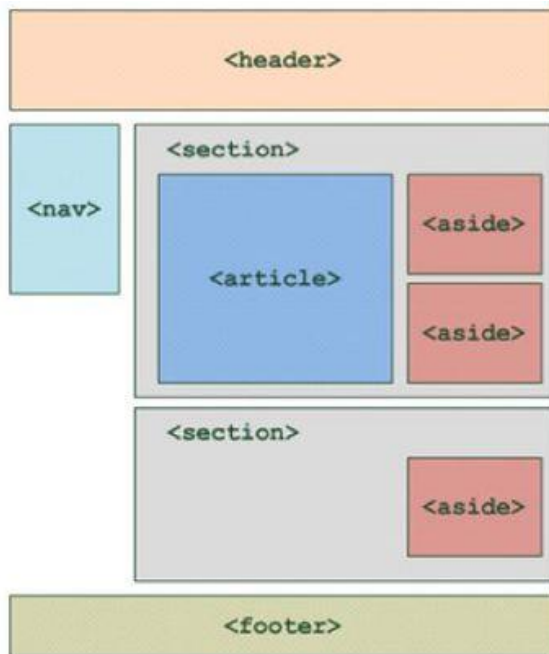
Balises structurales

Ces balises permettent de construire le squelette de notre site web :

Balise	Description
<header>	En-tête
<nav>	Liens principaux de navigation
<footer>	Pied de page
<section>	Section de page
<article>	Article (contenu autonome)
<aside>	Informations complémentaires

Le document

Avec les balises structurales, un affichage de notre site pourrait être celui-là :



Le document

Balises génériques ou balises universelles

Les balises génériques sont des balises qui n'ont pas de sens sémantique particulier :

Balise	Description
	Balise générique de type inline
<div>	Balise générique de type block

La plupart du temps, ces balises sont utilisées pour définir le design du site

Le contenu texte

Dans une page HTML, le contenu texte est affiché sans prendre en compte le formatage du texte, c'est-à-dire les retours chariot, les tabulations ou les espaces surnuméraires.

Exemples :

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <meta charset="utf-8">
```

```
  <title>Titre</title>
```

```
</head>
```

```
<body>
```

```
  Test de texte :
```

```
    + Tabulation
```

```
    + Espace
```

```
</body>
```

```
</html>
```



Test de texte : + Tabulation + Espace

Il est donc nécessaire d'utiliser des balises pour préciser le sens de l'on souhaite donner à une partie du texte. Ce sens pourra ensuite être « converti » en effet visuel par défaut ou en appliquant des feuilles de styles.

Le contenu texte

Balises de structuration du texte

Ces balises permettent de donner un sens au contenu que l'on souhaite afficher :

Balise	Description
<abbr>	Abréviation
<blockquote>	Citation (longue)
<cite>	Citation du titre d'une œuvre ou d'un évènement
<q>	Citation (courte)
<sup>	Exposant
<sub>	Indice
	Mise en valeur forte
	Mise en valeur normale
<mark>	Mise en valeur visuelle
<h1> à <h6>	Titre de niveau 1 à 6

Le contenu texte

Balise	Description
 	Retour à la ligne
<p>	Paragraphe
<hr />	Ligne de séparation horizontale
<address>	Adresse de contact
	Texte supprimé
<ins>	Texte inséré
<dfn>	Définition
<kbd>	Saisie clavier
<pre>	Affichage formaté (pour les codes sources)
<progress>	Barre de progression
<time>	Date ou heure

Le contenu texte



Exemples :

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <meta charset="utf-8">
```

```
  <title>Titre</title>
```

```
</head>
```

```
<body>
```

```
  <p>Test de texte :</p>
```

```
  + Tabulation<br />
```

```
  + Espace<br />
```

```
  <del>+ Autre</del>
```

```
</body>
```

```
</html>
```



Test de texte :

- + Tabulation
- + Espace
- + Autre

Les liens

Les liens permettent de changer de page ou de se repositionner dans une page. Ils servent également pour envoyer un mail ou télécharger un fichier :

Balise	Description
<code><a></code>	Lien hypertexte

L'attribut **href** permet d'indiquer l'adresse du lien (autre page, lien interne, fichier...).

L'attribut **target** permet de préciser que l'on souhaite que le lien s'ouvre dans une autre fenêtre du navigateur.

L'attribut **title** permet de préciser le texte que l'on souhaite afficher lorsque la souris vient sur le lien dans le navigateur.

Les liens permettent aussi d'amener vers d'autres endroits sur la même page. Il faut créer une ancre avec l'attribut id d'une balise quelconque pour « marquer » un endroit dans la page, puis faire un lien vers l'ancre comme ceci : ``.

Les liens



Exemples de liens :

`` : Lien vers un autre site (www.iocean.fr)

`` : Lien vers une page (page2.html) dans le même répertoire que la page actuelle

`` : Lien vers une page (page3.html) dans un sous-répertoire (dir)

`` : Lien vers une page (page4.html) dans un répertoire parent (parent_dir)

`` : Se positionne dans la même page sur la balise dont l'attribut id="*ancree1*"

`` : Lien vers une page (page5.html) et positionnement sur la balise dont l'attribut id="*ancree2*"

`` : Lien pour ouvrir l'outil de messagerie pour écrire un mail à cchampion@iocean.fr

`` : Lien pour télécharger un fichier (rapport.zip)

Les tableaux

Balises de tableau

Ces balises permettent de définir un vrai tableau que l'on souhaite afficher comme tel avec des lignes et des colonnes :

Balise	Description
<table>	Tableau
<caption>	Titre du tableau
<tr>	Ligne de tableau
<th>	Cellule d'en-tête
<td>	Cellule
<thead>	Section de l'en-tête du tableau
<tbody>	Section du corps du tableau
<tfoot>	Section du pied du tableau

Les tableaux

Historiquement, les tableaux en HTML servait à designer les pages web. Aujourd'hui, il est très fortement déconseillé de se service des tableaux dans ce sens.

Les tableaux ne doivent servir qu'à afficher des informations sous forme de tableaux de données.

Les balises minimum à la construction d'un tableau sont :

`<table>` **`</table>`** : indique le début et la fin d'un tableau (balise de type bloc).

`<tr>` **`</tr>`** : indique le début et la fin d'une ligne du tableau.

`<td>` **`</td>`** : indique le début et la fin du contenu d'une cellule.

Un tableau se construit ligne par ligne. Dans chaque ligne (`<tr>`), on indique le contenu des différentes cellules (`<td>`).

Il est également possible d'identifier une ligne d'entête en utilisant la balise **`<th>`** à la place de **`<td>`** :

`<th>` **`</th>`** : indique le début et la fin du contenu d'une cellule d'entête

Les tableaux

```
<table>  
  <tr>  
    <td>Cell 1</td>  
    <td>Cell 2</td>  
  </tr>  
  <tr>  
    <td>Cell 3</td>  
    <td>Cell 4</td>  
  </tr>  
</table>
```



Cell 1	Cell 2
Cell 3	Cell 4

```
<table>  
  <tr>  
    <th>Entête 1</th>  
    <th>Entête 2</th>  
  </tr>  
  <tr>  
    <td>Cell 1</td>  
    <td>Cell 2</td>  
  </tr>  
  <tr>  
    <td>Cell 3</td>  
    <td>Cell 4</td>  
  </tr>  
</table>
```



Entête 1	Entête 2
Cell 1	Cell 2
Cell 3	Cell 4

Les tableaux

Pour des tableaux conséquents, il est possible d'utiliser des balises permettant de structurer un tableau :

<caption> </caption> : indique le titre du tableau.

<thead> </thead> : indique l'en-tête du tableau (en haut).

<tfoot> </tfoot> : indique le pied du tableau (en bas).

<tbody> </tbody> : indique le corps du tableau (au centre)

Il est conseillé d'écrire les blocs du tableau dans cet ordre : **<thead>** , **<tfoot>** et **<tbody>**

Ces balises ne sont pas obligatoires et sont plutôt conseillées pour les gros tableaux.

Les tableaux

`<table>`

`<caption>Titre</caption>`

`<thead>`

`<tr>`

`<th>Entête 1</th>`

`<th>Entête 2</th>`

`</tr>`

`</thead>`

`<tfoot>`

`<tr>`

`<th>Pied 1</th>`

`<th>Pied 2</th>`

`</tr>`

`</tfoot>`

`<tbody>`

`<tr>`

`<td>Cell 1</td>`

`<td>Cell 2</td>`

`</tr>`

`<tr>`

`<td>Cell 3</td>`

`<td>Cell 4</td>`

`</tr>`

`</tbody>`

`</table>`



Entête 1

Cell 1

Cell 3

Pied 1

Titre

Entête 2

Cell 2

Cell 4

Pied 2

Les tableaux

Il est également possible de fusionner plusieurs cellules dans un tableau grâce aux attributs **colspan** et **rowspan** de la balise **<td>** :

Exemples pour fusionner 2 colonnes :

<table>

<tr>

<td>Cell 1</td>

<td>Cell 2</td>

<td>Cell 3</td>

</tr>

<tr>

<td colspan="2">2 Colonnes </td>

<td>Cell 4</td>

</tr>

</table>



Cell 1	Cell 2	Cell 3
2 Colonnes		Cell 4

Les tableaux

Exemples pour fusionner 2 lignes :

<table>

<tr>

<td rowspan="2">2 lignes</td>

<td>Cell 1</td>

<td>Cell 2</td>

</tr>

<tr>

<td>Cell 3</td>

<td>Cell 4</td>

</tr>

</table>



2 lignes	Cell 1	Cell 2
	Cell 3	Cell 4

Les listes

Balises de listes

Ces balises permettent de créer des listes (listes à puces, listes numérotées, listes de définitions...) :

Balise	Description
	Liste à puces, non numérotée
	Liste numérotée
	Élément de la liste à puces
<dl>	Liste de définitions
<dt>	Terme à définir
<dd>	Définition du terme

Les listes



Exemples de liste à puces :

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



- Element 1
- Element 2
- Element 3

Exemples de liste numérotés :

```
<ol>  
  <li>Element 1</li>  
  <li>Element 2</li>  
  <li>Element 3</li>  
</ol>
```



1. Element 1
2. Element 2
3. Element 3

Exemples de liste de définitions :

```
<dl>  
  <dt>terme 1</dt>  
  <dd>Le terme 1 est ...</dd>  
  <dt>terme 2</dt>  
  <dd>Le terme 2 est ...</dd>  
</dl>
```



terme 1
 Le terme 1 est ...
terme 2
 Le terme 2 est ...

Les formulaires

Balises de formulaire

Ces balises permettent de définir un formulaire de saisie :

Balise	Description
<form>	Formulaire
<label>	Libellé d'un champ
<input />	Champ de formulaire (texte, mot de passe, case à cocher, bouton, etc.)
<textarea>	Zone de saisie multiligne
<select>	Liste déroulante
<option>	Élément d'une liste déroulante
<optgroup>	Groupe d'éléments d'une liste déroulante
<fieldset>	Groupe de champs
<legend>	Titre d'un groupe de champs

Les formulaires

Un formulaire ajoute de l'interactivité dans votre site web en permettant à l'utilisateur de saisir des données qui seront ensuite envoyées vers le serveur pour être traitées.

Pour ce faire, on utilise la balise `<form>` `</form>`. Cette balise permet également de préciser l'URL du serveur qui doit être appelée pour traiter les informations saisies par l'utilisateur ainsi que la méthode pour envoyer les données.

Il est important de comprendre que la page HTML ne traite pas les données saisies, elle peut au mieux les mettre en forme et les vérifier avec du code JavaScript par exemple. Pour que ces données soient traitées (par exemple, enregistrer dans une base de données), il est nécessaire de les envoyer vers un serveur (Back-End) qui pourra faire le traitement nécessaire.

L'attribut **action** permet de préciser l'URL du serveur à appeler pour traiter les données saisies. Cette URL est construite sur le même principe que pour les liens.

L'attribut **method** indique la méthode du protocole HTTP à utiliser pour envoyer les données. Il existe deux possibilités :

- **POST** : la méthode la plus utilisée pour les formulaires car elle permet d'envoyer un grand nombre de données directement dans le corps de la requête HTTP
- **GET** : une méthode peu adaptée car elle est limitée à 255 caractères. Les données sont envoyées à la suite de l'URL

Les formulaires

Exemple d'un formulaire de saisie :

```
<form action="connexion.jsp" method="post">
```

...

```
</form>
```

Un formulaire de saisie doit contenir des zones de saisie. Pour cela, on peut utiliser les balises **<input />**, **<textarea> </textarea>** et **<select> </select>** en fonction du type de saisie que l'on souhaite faire.

Il est également recommandé d'utiliser la balise **<label> </label>** pour préciser le libellé du champ de saisie.

Les formulaires

Les zones de saisie de texte « simple » utilise la balise **<input />**.

Cette balise propose les attributs principaux suivants :

name : nom de la zone de saisie (correspond au nom qui sera utilisé pour envoyer la valeur au serveur);

type : type de saisie pour la zone (cf page suivante)

id : identifiant unique de la zone utilisé notamment pour les labels (en général, on met à la même chose que le name)

value : valeur par défaut de la zone

size : taille de la zone dans le formulaire

maxlength : nombre de caractères maximum pour la zone

placeholder : texte à afficher quand aucun valeur n'est définie

La balise **<label> </label>** permet d'associer un libellé à une zone de saisie. Cette association se fait par l'attribut **for** qui doit contenir l'id de la zone de saisie associée

Les formulaires

Les principales zone de saisie de type texte « simple » sont :

checkbox	une case à cocher permettant de sélectionner/désélectionner une valeur.
color	un contrôle permettant de choisir une couleur.
date	un contrôle qui permet de saisir une date (sans heure).
datetime-local	un contrôle qui permet de saisir une date et une heure, sans fuseau horaire associé.
email	un champ qui permet d'éditer une adresse email.
file	un contrôle qui permet à l'utilisateur de sélectionner un fichier. L'attribut accept peut être utilisé pour définir les types de fichier que le contrôle permet de sélectionner.
hidden	un contrôle qui n'est pas affiché mais dont la valeur est envoyée au serveur.
number	un contrôle qui permet de saisir un nombre.
password	un champ de saisie sur une seule ligne dont la valeur est masquée. Ce champ est destiné à la saisie d'un mot de passe.
radio	un bouton radio. Pour un même nom de champ, seul un bouton radio peut être sélectionné à un instant donné.
search	un champ de texte sur une seule ligne qui permet de saisir des requêtes de recherche. Les sauts de ligne sont automatiquement supprimés de la valeur saisie.
tel	un contrôle qui permet de saisir un numéro de téléphone.
text	un champ de texte sur une seule ligne. Les sauts de ligne sont automatiquement retirés de la valeur saisie.
time	contrôle qui permet de saisir une valeur pour une heure, sans fuseau horaire associé.
url	un champ qui permet d'éditer une URL.

Les formulaires

Exemple d'un formulaire de saisie sans label :

```
<form action="connexion.jsp" method="post">  
    Login : <input type="text" name="login" />  
    Mot de passe : <input type="password" name="password" />  
    <input type="submit" value="Envoyer le formulaire">  
</form>
```

Exemple d'un formulaire de saisie avec label :

```
<form action="connexion.jsp" method="post">  
    <label for="login" >Login</label> :  
    <input type="text" name="login" id="login" />  
    <label for="password">Mot de passe</label> :  
    <input type="password" name="password" id="password" />  
    <input type="submit" value="Envoyer le formulaire">  
</form>
```

Les formulaires

Les zones de saisie de texte type commentaire utilise la balise `<textarea>` `</textarea>`.

Cette balise propose les attributs principaux suivants :

name : nom de la zone de saisie (correspond au nom qui sera utilisé pour envoyer la valeur au serveur);

id : identifiant unique de la zone utilisé notamment pour les labels (en général, on met à la même chose que le name)

rows : nombre de ligne de la zone

cols : nombre de colonne de la zone

Le texte éventuellement contenu dans la balise définit le texte par défaut de la zone.

Comme pour les inputs, la balise `<label>` `</label>` permet d'associer un libellé à cette zone de saisie.

Les formulaires

Exemple d'un formulaire de saisie d'un commentaire :

```
<form action="commentaire.jsp" method="post">  
  <label for="commentaire">Commentaire</label> :  
  <textarea name="commentaire" id="commentaire" rows="3" cols="40">  
    Ceci est le commentaire par défaut  
  </textarea>  
  <input type="submit" value="Envoyer le formulaire">  
</form>
```

Les formulaires

Les zones de saisie de texte type liste déroulante utilise la balise `<select> </select>`. Elle propose les attributs principaux suivants :

name : nom de la zone de saisie (correspond au nom qui sera utilisé pour envoyer la valeur au serveur);

id : identifiant unique de la zone utilisé notamment pour les labels (en général, on met à la même chose que le name)

Cette balise contient elle-même plusieurs balises `<option> </option>` qui contiennent chacune une ligne de la liste déroulante. Elle contient l'attribut **value** qui permet de définir la valeur de la ligne (qui peut donc être différente du libellé affiché dans la liste).

Comme pour les input, la balise `<label> </label>` permet d'associer un libellé à cette zone de saisie.

Les formulaires

Exemple d'un formulaire de saisie d'une liste déroulante :

```
<form action="liste.jsp" method="post">  
  <label for="pays" > Pays</label> :  
  <select name="pays" id="pays" >  
    <option value="FR">France</option>  
    <option value="ES">Espagne</option>  
    <option value="US">États-Unis</option>  
  </select >  
</form>
```


Le multimédia

Balises pour le multimédia

Ces balises permettent d'afficher du contenu multimédia (image, vidéo, son) :

Balise	Description
	Image
<figure>	Figure (image, code, etc.)
<figcaption>	Description de la figure
<audio>	Son
<video>	Vidéo
<source>	Format source pour les balises <audio> et <video>

Le multimédia

On peut insérer des images dans une page HTML.

Pour cela, on utilise la balise ****. Cette balise doit toujours se trouver dans une balise **<p> </p>**.

Elle propose les attributs principaux suivants :

src : précise l'URL où se trouve l'image à afficher

alt : permet de définir le texte qui sera affiché si le navigateur ne peut pas télécharger l'image.

title : permet de définir le texte de l'infobulle sur l'image

Les différents formats d'image qu'il est recommandé d'utiliser en fonction du type d'image sont :

Une photo	JPEG
N'importe quel graphique avec peu de couleurs (moins de 256)	PNG 8 bits ou éventuellement GIF
N'importe quel graphique avec beaucoup de couleurs	PNG 24 bits
Une image animée	GIF animé

Le multimédia

Exemple d'affichage d'une image :

<p>

</p>

Le multimédia

On peut également insérer des sons dans une page HTML.

Pour cela, on utilise la balise **<audio> </audio>**. Elle propose les attributs principaux suivants :

controls : pour ajouter les boutons « Lecture », « Pause » et la barre de défilement.

width : précise la largeur de l'outil de lecture audio.

loop : la musique sera jouée en boucle.

autoplay : la musique sera jouée dès le chargement de la page.

preload : indique si la musique peut être préchargée dès le chargement de la page ou non :

auto(par défaut) : le navigateur décide ce qu'il doit précharger.

metadata : charge uniquement les métadonnées (durée, etc.).

none : pas de préchargement pour économiser de la bande passante.

Les différents formats de musique que l'on peut télécharger dépendent du navigateur.
Ce sont : MP3, AAC, OGG et WAV.

Si l'on veut s'assurer qu'un maximum de navigateurs sera capable de lire le son, il est possible de préciser plusieurs fichiers avec des formats différents grâce à la balise **<source>**

Le multimédia

Exemple d'affichage d'un son :

<audio controls>

<source src="music.mp3">

<source src="music.ogg">

</audio>

Le multimédia

Enfin, on peut insérer des vidéos dans une page HTML.

Pour cela, on utilise la balise **<video>** **</video>**. Elle propose les attributs principaux suivants :

controls : pour ajouter les boutons « Lecture », « Pause » et la barre de défilement.

width : précise la largeur de la vidéo.

height : précise la hauteur de la vidéo.

loop : la vidéo sera jouée en boucle.

autoplay : la vidéo sera jouée dès le chargement de la page.

preload : indique si la vidéo peut être préchargée dès le chargement de la page ou non :

auto(par défaut) : le navigateur décide ce qu'il doit précharger.

metadata : charge uniquement les métadonnées (durée, etc.).

none : pas de préchargement pour économiser de la bande passante.

Les différents formats de vidéo que l'on peut télécharger dépendent du navigateur.

Ce sont : H.264, Ogg Theora et WebM

Si l'on veut s'assurer qu'un maximum de navigateurs sera capable de lire le son, il est possible de préciser plusieurs fichiers avec des formats différents grâce à la balise **<source>**

Le multimédia

Exemple d'affichage d'une vidéo :

<video controls>

<source src="video.avi">

<source src="video.mp4">

</video >

Catégorisation des contenus



Il existe plusieurs catégories de contenu qui régissent les règles d'affichage, d'imbrication ou plus généralement d'utilisation des balises au sein d'un document HTML.

Connaitre ces règles est très utile pour créer des structures cohérentes et valides.

Les principales catégories de contenu sont les suivantes :

- Le contenu de méta-données
- Le contenu de flux
- Le contenu sectionnant
- Le contenu de titre
- Le contenu phrasé
- Le contenu intégré
- Le contenu interactif
- Le contenu associé au formulaire

A noter qu'une même balise peut être associée à une ou plusieurs catégories.



Plus de détails sur le site de Mozilla [MDN](https://developer.mozilla.org/)

Quelques ressources utiles

<https://www.w3.org/TR/html5/syntax.html>

<https://www.w3schools.com/html/default.asp>

<https://developer.mozilla.org/fr/docs/Web/HTML>

Pense-bête pratique

<http://overapi.com/html>

<http://htmlcheatsheet.com/>



Restons en contact.

DIGINAMIC

Lionel Cabon, Directeur
contact@diginamic.fr

N° Déclaration OF : 91 34 08867 34

Nantes, Paris, Montpellier

www.diginamic.fr