HTML5

mémento

Rodolphe **Rimelé**

EYROLLES

Syntaxe d'HTML5

HTML (*HyperText Markup Language*) est spécifié par le W3C (*World Wide Web Consortium*, http://www.w3.org/), à l'initiative du WhatWG (http://www.whatwg.org/html), en tant que *living standard* en évolution constante, sans numéro de version.

La syntaxe d'HTML5 est plus permissive qu'en XHTML : les éléments vides ne nécessitent pas d'être fermés (ex : `<img src="image.jpg">`), la casse est variable et les guillemets facultatifs (`<INPUT type=text>`), et certaines balises fermantes sont optionnelles. Les commentaires présents dans le code source ne sont pas affichés ni interprétés par le navigateur (`<!-- Ceci est un commentaire -->`).

BONNE PRATIQUE **Gardez les bonnes habitudes XHTML**
Les conventions d'écriture XHTML sont plus constantes et claires. En cas de nombreux éléments répétés (surtout des blocs `<div>`), prenez soin d'indiquer par un commentaire la fin de chaque bloc important.

HTML5 définit des familles d'éléments plus variées que les simples types bloc et en ligne : les méta-informations, les éléments de flux, les éléments de phrasé, le contenu embarqué et le contenu interactif.

Document avec en-tête complet : des balises allégées

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>Ma page</title>
</head>
<body>
  ...
</body>
</html>
```

<html> *manifest*
La racine. Contient un élément `<head>` suivi d'un élément `<body>`. Précisez la langue du document avec l'attribut `lang`.

<head>
En-tête du document. Contient les méta-informations interprétées par le navigateur sans apparaître dans le corps du document : `<title>`, `<meta>`, `<link>`, `<style>`, `<script>`.

<body>
Corps du document. Contient toutes les balises HTML de contenu : texte, blocs, images, titres, médias, formulaires, etc.

<title>
Titre du document. Affiché par le navigateur et indexé par les moteurs de recherche. Contenu par `<head>`.

```
<title>Les kiwis à travers les âges</title>
```

<meta> *name, value, charset, http-equiv*
Méta-informations. Apporte des informations (mots-clés, description, auteur) dans l'en-tête `<head>` que les agents utilisateurs vont pouvoir exploiter.

```
<meta charset="utf-8">
<meta name="description" value="Description du document">
<meta name="keywords" value="mots, clés, énumérés">
<meta http-equiv="refresh" content="10; URL=hopla.html">
```

Autres valeurs possibles de `name` : `author`, `generator`, `application-name`.

<style> *type, media,* ***scoped***
Styles embarqués dans le document. Déclarations CSS à appliquer au document HTML.

```
<style>
h1 {
  color:red;
}
</style>
```

L'attribut `scoped` indique que les styles ne s'appliquent qu'au parent qui contient cette balise (peu pris en charge en pratique, mieux vaut toujours placer `<style>` dans `<head>`).

<script> *type, src, defer,* ***async****, charset*
Script. Contient un script (JavaScript) ou fait référence à un fichier externe avec l'attribut `src`.

```
<script src="script.js"></script>
```

L'attribut `async` indique une exécution asynchrone, dès que le script a été chargé.

<noscript>
Texte alternatif aux scripts. Lorsque les scripts sont désactivés sur le navigateur. Ignoré par les navigateurs dont les scripts sont activés.

```
<noscript>Votre navigateur ne supporte pas les scripts...</noscript>
```

<base> *href, target*
Adresse de base. L'attribut `href` indique une URL absolue de base à partir de laquelle calculer les liens du document, et `target` un contexte de navigation (`nom`, `_blank`, `_self`, `_parent` ou `_top`).

```
<base href="http://www.alsacreations.com/tutoriels/" target="_blank">
```

AVERTISSEMENT Le niveau de prise en charge de HTML, en évolution constante, varie d'un navigateur à l'autre. Consultez des tableaux de prise en charge à jour, tels que sur www.caniuse.com, l'un des plus complets. Outre ses balises, HTML5 offre de nombreuses API pilotables avec JavaScript pour créer des fonctionnalités avancées : Géolocalisation, Audio et Vidéo, Web Storage, Canvas, Microdata, Offline, Server-Sent Events, Web Workers, Web Sockets, IndexedDB, Drag&Drop, FullScreen, etc.

CONVENTIONS Les nouveaux éléments apparus avec HTML5 sont signalés par le pictogramme 5; les nouveaux attributs sont signalés en noir.

Des sections pour structurer le document

Le choix du balisage de section est laissé à la libre appréciation du développeur web.

<article>

Article. Portion de contenu indépendante, se suffisant en termes de compréhension (pouvant être syndiquée dans un flux RSS). Exemples : article de presse, fiche cinéma, réponse de forum, commentaire d'article (qui est un `<article>` imbriqué dans `<article>`).

```
<article>
  <header>
    <h1>La Vitamine C est tendance</h1>
    <time datetime="2012-04-01" pubdate>Le 1er avril 2012</time>
  </header>
  <p>...</p>
  <footer>Posté par Georges</footer>
</article>
```

<aside>

Aparté. Contenu indirectement lié au contenu principal, non nécessaire à sa compréhension : définitions, digressions, sidebar, compléments...

```
<article>
  <h1>La vitamine C est tendance</h1>
  <p>...</p>
  <aside>
    <h2>Qu'est-ce qu'une vitamine ?</h2>
    <p>Les explications...</p>
  </aside>
  <p>...</p>
</article>
```

<section>

Section. Élément générique pour une section de contenu ou d'application web, utilisé à défaut d'un autre élément de section plus sémantique tel que `article`, `nav`, `header`, `footer`, `aside`. À ne pas confondre avec `<div>` qui n'a aucune valeur sémantique. En général, `<section>` contient un titre qui la définit.

<nav>

Navigation. Regroupe la sélection des principaux liens pour naviguer dans le site, l'application ou le document.

```
<nav>
  <ul>
    <li><a href="/">Accueil</a></li>
    <li><a href="/inscription.html">Inscription</a></li>
    <li><a href="#contact">Contact</a></li>
  </ul>
</nav>
```

<header>

En-tête. S'applique au document entier, mais aussi à toute section au sens large, `<article>` y compris.

```
...
<body>
  <header>
    <img src="logo.png" alt="...">
    <h1>Titre du document</h1>
  </header>
  <p>...</p>
</body>
```

<footer>

Pied. S'applique au document entier, mais aussi à toute section au sens large, `<article>` y compris.

```
<article>
  ...
  <footer>
    <p>Tags : HTML5, audio, tutoriel, media</p>
  </footer>
</article>
```

Hiérarchisation du texte et séparations

<h1>
<h2>
<h3>
<h4>
<h5>
<h6>

Hiérarchie de titres. Six niveaux de titres pour hiérarchiser le contenu. Bien les employer selon leur niveau, et non selon leur aspect ou taille par défaut. Il revient aux CSS d'en définir le rendu.

```
<h1>Les aliments</h1>
  <h2>Les légumes</h2>
    <h3>La patate</h3>
    <h3>La tomate</h3>
  <h2>Les fruits</h2>
    <h3>Le kiwi</h3>
    <h3>La banane</h3>
    <h3>La mangue</h3>
```

<hgroup> **Groupe de titres.** Regroupe un ensemble d'éléments <h1> à <h6> quand l'en-tête est constitué de plusieurs niveaux (sous-titres, lignes additionnelles, mentions), pour représenter l'en-tête d'une section.

```
<article>
  <hgroup>
    <h2>Titre principal</h2>
    <h3>Sous-titre</h3>
  </hgroup>
</article>
```

<p> **Paragraphe.** Contient un paragraphe de texte, éventuellement accompagné d'autres balises en ligne plus sémantiques (des images, vidéos, audio), mais pas d'autre élément de type bloc (y compris <p> lui-même).

```
<p>Le gâteau est un mensonge.</p>
```

<wbr> **Indication d'opportunité de césure de mot.** Indique au navigateur qu'il est plus opportun de scinder un mot à l'emplacement de cet élément, s'il y a besoin, notamment pour éviter une césure automatique pouvant être mal interprétée.

 Saut de ligne. Force un saut de ligne : l'affichage se poursuit en début de ligne suivante.

BONNE PRATIQUE Afin d'éviter l'usage abusif de <br> pour créer de l'espacement vertical, il est préférable d'utiliser les propriétés CSS (`margin`, `padding`) pour introduire des marges sur les conteneurs et les paragraphes.

<hr> **Séparation.** Élément vide qui marque une séparation dans le contenu, typiquement au niveau des paragraphes, usuellement représentée par une barre horizontale.

Conteneurs génériques

BONNE PRATIQUE Évitez d'abuser des conteneurs génériques. Privilégiez les éléments plus discriminants qui qualifient sémantiquement (notamment les sections). N'utilisez qu'en dernier recours, si aucun autre élément plus approprié n'a été trouvé.

 Conteneur de type bloc. Élément neutre permettant de regrouper d'autres éléments de tous types (bloc ou en ligne).

 Conteneur de type en ligne. Équivalent en ligne de l'élément bloc , il n'a pas de valeur sémantique propre, mais peut être employé comme conteneur neutre.

Balises en ligne sur le statut du contenu

<ins>
cite, datetime

Contenu ajouté. Marque une portion de contenu insérée dans le document.

L'attribut `cite` peut être utilisé pour faire référence à une adresse externe expliquant la raison du changement. L'attribut `datetime` indique la date du changement.

```
<p>J'aime les kiwis <ins datetime="2012-04-01">mais aussi les cerises
</ins> et les bananes !</p>
```

<del>
cite, datetime

Contenu supprimé. Marque une portion de contenu amenée à être supprimée du document (communément affichée par du texte barré). Les attributs `cite` et `datetime` fonctionnent de façon semblable à <ins>.

<s> **Contenu obsolète.** Signifie que le contenu n'est plus exact ou pertinent dans le contexte ; usuellement représenté par un texte barré.

```
<p>Ce mémento concerne la version <s>4</s> 5 de HTML.</p>
```

Blocs sémantiques

<blockquote> *cite*

Bloc de citation. Élément destiné à regrouper plusieurs blocs (<p>, , etc.) et du texte constituant une citation. L'attribut cite optionnel peut pointer vers la source de la citation.

```
<blockquote>
  <p>Être ou ne pas être.</p>
</blockquote>
```

<address>

Informations de contact. Cet élément est prévu pour pouvoir contacter l'auteur ou l'éditeur du contenu (document, site, article, etc.), par exemple avec un e-mail.

```
<address><a href="mailto:info@email.com">Contacter le webmestre</a>
</address>
```

Si cet élément est appliqué à <body>, il constitue une information de contact pour l'ensemble du document. S'il est placé dans une section, il est relatif uniquement à cette dernière.

<pre>

Bloc de code préformaté. Destiné à être affiché avec une police à chasse fixe (caractères de même largeur), idéalement pour exposer des extraits de code source.

<code>

Extrait de code informatique. Le texte est rendu dans une police à chasse fixe.

```
<p>L'élément <code>img</code> est une image.</p>
```

<kbd>

Entrée utilisateur. Entrée à l'aide du clavier ou d'un autre moyen tel que la reconnaissance vocale. Pour marquer une touche précise, doubler cet élément.

```
<p>Entrez <kbd>www.blup.fr</kbd> dans votre navigateur.</p>
<p>Appuyez sur <kbd><kbd>Shift</kbd></kbd>.</p>
```

<var>

Variable de code. Marque une variable extraite d'un code source ou d'un algorithme.

```
<p>Il faut additionner <var>x</var> et <var>y</var>.</p>
```

<samp>

Extrait de la sortie d'un programme. Délimite un résultat renvoyé par un programme ; habituellement rendu avec une police à chasse fixe.

```
<p>Le serveur renvoie <samp>404 Page Not Found</samp> ?
Inutile de persévérer.</p>
```

Balises en ligne sémantiques

<time> *datetime, pubdate*

Date et heure. Balise attribuant une signification standardisée à une date (et heure si nécessaire), compréhensible par un robot ou un programme, via l'attribut datetime. La présence de l'attribut pubdate signifie que la valeur de temps vaut pour la date de publication du plus proche ancêtre <article>.

```
<article>
  <p>Publié le <time datetime="2012-04-01T13:37:00Z">1er avril 2012
</time></p>
</article>
```

<abbr> *title*

Abréviation. Désigne une abréviation au sens large (par exemple « HTML », « CIA », « SNCF », etc.), la forme developpée étant donnée par l'attribut title qui en décrit la signification. L'attribut lang indique la langue dans laquelle l'abréviation doit être lue. Les navigateurs distinguent les abréviations en les soulignant et en affichant en infobulle le contenu de title.

```
<abbr title="HyperText Markup Language" lang="en">HTML</abbr>
```

<cite>

Mention d'une œuvre. Le titre d'une œuvre (livre, musique, émission, film, chanson, théâtre, exposition, opéra, sculpture, peinture, etc.) cité dans un texte doit être balisé par <cite>.

```
<p><cite>La Bohème</cite> est un merveilleux opéra de Giacomo
Puccini.</p>
```

<q> *cite*

Courte citation. Texte cité, au sein d'un contenu. Contenu autorisé : phrase.

```
<p>Galilée a dit <quote>Toute la nature n'est que mathématique
</quote>.</p>
```

L'attribut cite vise à faire référence à un document externe contenant la source complète de la citation.

<dfn> *title*

Définition. Le terme entouré par cet élément est accompagné d'une description ou d'une explication, contenue dans son plus proche ancêtre.

```
<p>Le <dfn>spam</dfn> est une communication électronique
non sollicitée.</p>
```

L'attribut title, ou un unique élément enfant <abbr> équipé de title peuvent remplacer le terme exact à définir.

Mise en forme de texte

<mark> (HTML5) — **Mise en valeur (surligné).** Attire l'attention sur un passage de texte particulièrement pertinent au regard d'un certain contexte. Les moteurs de recherche l'utilisent pour surligner le terme recherché (usuellement surligné par le navigateur).

<u> — **Souligné.** Souligne un texte sans lui conférer d'importance (`<strong>`) ni d'emphase (`<em>`). À Éviter sauf cas particulier (mise en évidence d'erreur synaxique ou othographique, noms propres, etc.).

<b> — **Mise en forme (gras).** Portion de texte mise en forme d'une manière particulière (par exemple, un mot-clé ou un nom de marque), sans désir de conférer d'importance (`<strong>`) ni d'emphase (`<em>`), habituellement mis en forme en gras.

<i> — **Mise en forme (italique).** Portion de texte mise en forme d'une manière particulière (par exemple, un terme technique, une pensée, un nom de navire), sans désir de conférer d'importance (`<strong>`) ni d'emphase (`<em>`), habituellement mis en forme en italique.

**** — **Forte mise en exergue.** Attribue une forte importance au contenu. Par habitude, les navigateurs affichent le texte ainsi balisé en gras.

```
<p>Faites <strong>très</strong> attention !</p>
```

**** — **Emphase.** Ajoute un effet d'emphase au texte, usuellement représenté par de l'italique (à ne pas confondre avec `<i>`).

```
<p>JavaScript est <em>vraiment</em> surpuissant.</p>
```

<small> — **Petits caractères.** Peut représenter une mention discrète, en fin de document ou de section, par exemple faisant référence à des mentions légales ou des conditions de vente. À ne pas utiliser dans le seul but de réduire la taille de la police par défaut.

<sub> — **Indice.** Affiche le texte en indice, notamment pour les formules scientifiques.

```
<abbr title="Eau">H<sub>2</sub>O</abbr>
```

<sup> — **Exposant.** Affiche le texte en exposant, notamment pour les formules mathématiques.

```
<p>ax<sup>2</sup>+bx+c</p>
```

<bdo> (HTML5) *dir* — **Modification du sens d'écriture.** Indique le sens d'affichage du texte, de gauche à droite (attribut `dir="ltr"`) ou de droite à gauche (attribut `dir="rtl"`).

<bdi> (HTML5) — **Isolation du sens d'écriture.** Balise des passages de texte susceptibles de comprendre une écriture en sens inverse du contenu principal.

Listes

Listes de définitions

<dl> — **Liste de définitions.** Associe un « terme à définir » (`<dt>`) et des « définitions » (`<dd>`) s'y rapportant. Contenu autorisé : succession de `<dt>` et `<dd>`.

<dt> — **Terme/titre de définition.** Enfant d'une liste `<dl>`, balisant le terme à définir. Contenu autorisé : contenus textuels y compris des balises de niveau intra-paragraphe chargées de contenu (*phrasing content*).

<dd> — **Contenu de la définition.** Enfant d'une liste `<dl>`, recueillant la définition du terme `<dt>` qui le précède immédiatement.

```
<dl>
  <dt>Le kiwi</dt>
  <dd>Un fruit mais aussi un animal</dd>
  <dt>Le litchi</dt>
  <dd>Un excellent fruit exotique</dd>
</dl>
```

Listes classiques

<ul> — **Liste non ordonnée.** Désigne une liste d'éléments non ordonnés. Contenu autorisé : `<li>`.

```
<ul>
  <li>premier élément</li>
  <li>deuxième élément
    <ul>
      <li>premier sous-élément</li>
      <li>second sous-élément</li>
    </ul>
  </li>
  <li>troisième élément</li>
</ul>
```

Listes

<ol> *start, reversed, type*

Liste ordonnée. Structure une liste d'éléments dont l'ordre importe, pouvant être numérotée. Contenu autorisé : <li>.
L'attribut start définit l'index à partir duquel commencer la numérotation. L'attribut reversed inverse l'ordre de la numérotation, et l'attribut type en indique le type : « 1 », «a», «A», «i», «I».

<li>

Élément de liste. Désigne un élément contenu dans une liste. Peut contenir une sous-liste <ul> ou <ol>, et ainsi de suite, ou tout autre élément de flux.

```
<p>Classement de la compétition :</p>
<ol>
  <li>Philippe Dukiwi</li>
  <li>Rodolphe Delabanane</li>
  <li>Raphaël Quetsche</li>
</ol>
```

Embarquer du multimédia

<figure> HTML5

Unité de contenu, avec légende optionnelle <figcaption>, qui regroupe habituellement une ou plusieurs illustrations, pouvant être mentionnées depuis le contenu principal et déplacées à l'extérieur du flux, **sans affecter la signification du document**.

<figcaption> HTML5

Légende de figure. Fonctionne de pair avec <figure> qui est son unique parent autorisé, pour attribuer une légende à l'ensemble du bloc, qu'il contienne une ou plusieurs illustrations.

```
<p>[...] le chocolat ne pourrait exister sans le cacaoyer.</p>
<figure>
  <img src="feves.jpg" alt="Fèves de cacao">
  <figcaption>La récolte des fèves précède la torréfaction, Brésil, photo de <a href="http://www...">John McGeek</a></figcaption>
</figure>
```

<details> HTML5 *open*

Informations ou contrôles additionnels. Représente un ensemble de contenus ou de contrôles qui n'est pas présenté par défaut à l'utilisateur et qui peut être déployé sur demande, usuellement par un clic sur l'intitulé défini avec <summary>. L'attribut open définit l'état (déployé) par défaut.

<summary> HTML5

Résumé, légende ou intitulé. Fournit un intitulé, indiqué à l'utilisateur, pour son parent <details>. Cet intitulé doit résumer le reste du contenu qui peut être déployé.

```
<details>
  <summary>Statistiques du téléchargement</summary>
  <p>...</p>
</details>
```

Images

Les images présentes dans le document HTML (hors images de fond définies avec CSS) sont à considérer comme des éléments à part entière devant apporter des informations à l'utilisateur. Les formats PNG, JPEG, GIF sont désormais universellement reconnus.

<img> *src, alt, height, width, usemap, ismap*

Image. Insère une image de contenu dans le document. L'attribut src spécifie l'adresse vers le fichier image. L'attribut alt indique un texte alternatif à l'image. Les dimensions sont définies en pixels par width (largeur) et height (hauteur). L'attribut ismap envoie les coordonnées cliquées au serveur.

```
<p><img src="kiwis.jpg" alt="Poêlée de kiwis" width="300"
height="100">La poêlée de kiwis est une spécialité...</p>
```

<map> *name*

Carte cliquable pour image. Catalogue les zones sensibles <area> d'une image utilisée en guise d'image réactive. Si l'attribut id est présent, il doit avoir la même valeur que name, qui permet de lier <map> et <img> avec son attribut usemap.

BONNE PRATIQUE **alt ou title ?**

L'attribut alt sert d'alternative textuelle aux éléments graphiques. Son contenu est utilisé par les navigateurs non graphiques ou les lecteurs vocaux pour les malvoyants pour qui cette information est essentielle. Il est régulièrement utilisé à contre-emploi, pour créer une infobulle au survol de la souris sur un élément. Or, cette fonction est assignée à l'attribut title. Utilisez un attribut alt="" vide pour les images qui sont purement décoratives.

<area> *shape, coords, **rel, media, hreflang,** type, href, target, alt*

Zone cliquable. Zone sensible d'image réactive `<map>`. Sa forme est définie par l'attribut `shape` qui prend les valeurs `circle` (cercle), `poly` (polygone), `rect` (rectangle) et `default` (toute l'image), ainsi que par les coordonnées de la forme choisie avec l'attribut `coords`.

```
<img src="carte.png" usemap="#atlas">
<map name="atlas">
  <area shape="rect" coords="0,0,100,90" href="amerique-nord.html" alt="Amérique du Nord">
  <area shape="circle" coords="200,75,50" href="oceanie.html" alt="Océanie">
</map>
```

Audio et vidéo

Nouveaux venus avec HTML5, les éléments de média permettent d'agrémenter un document ou une application avec sons et vidéos, sans utiliser d'extension propriétaire... Cependant, chaque navigateur reconnaît des formats différents (parmi MP3, Ogg Vorbis, WAV, AAC pour l'audio, et MP4/H.264, WebM, Ogg Theora pour la vidéo...).

<audio> 5

Audio. Son ou élément audio, interprété dans le navigateur, dont la source est définie par l'attribut `src`, ou par une succession d'enfants `<source>` (voir plus bas) proposant des formats alternatifs.

```
<audio src="musique.mp3" controls>
  Ce navigateur ne supporte pas HTML5 Audio
</audio>
```

<video> 5

Vidéo. Vidéo affichée et interprétée dans le navigateur, dont la source est définie par l'attribut `src`, ou par une succession d'enfants `<source>` (voir plus bas) proposant des formats alternatifs.

```
<video src="film.mp4" controls poster="apercu.jpg" width="640" height="480">
  Ce navigateur ne supporte pas HTML5 Video
</video>
```

<source> 5

Source alternative audio ou vidéo. Placé dans `<audio>` ou `<video>`, vient en remplacement de l'attribut `src` pour préciser plusieurs sources possibles, dont le format est indiqué par `type`, l'adresse par `src` et le média destination par `media`.

```
<video controls width="640" height="480">
  <source src="film.mp4" type="video/mp4">
  <source src="film.webm" type="video/webm">
  <source src="film.ogv" type="video/ogg">
  <source src="film-mobile.mp4" type="video/mp4" media="handheld">
</video>
```

<track> 5 *kind, src, srclang, label, default*

Piste de texte. Lie un fichier de piste texte pour un élément média. L'attribut `kind` indique le type de piste : `subtitles` (sous-titres), `captions` (légendes), `descriptions` (audiodescription), `chapters` (chapitrage), `metadata` (méta-informations). Le format préconisé est le WebVTT. L'attribut `label` définit l'étiquette présentée à l'utilisateur, et `srclang` la langue du fichier source.

Attributs spécifiques à l'audio et à la vidéo

autoplay	**Lecture automatique.** Active la lecture dès le chargement de la page.
preload	**Préchargement.** Précharge le fichier média, dès le chargement de la page.
controls	**Affichage des contrôles.** La présence de l'attribut `controls` affiche les contrôles natifs du navigateur (lancement de la lecture, arrêt, progression, volume, etc.).
loop	**Lecture en boucle.** Active la lecture en boucle.
poster	**Aperçu par défaut.** Indique l'adresse de l'image d'aperçu à charger pour occuper la zone d'affichage lorsque la vidéo n'est pas jouée.
width, height	**Dimensions.** Spécifie les dimensions d'affichage de la zone vidéo.
mediagroup	**Groupe d'éléments audio et vidéo.** Permet de regrouper différents éléments média (audio, vidéo), par exemple dans une liste de lecture gérée par JavaScript.
muted	**Silencieux.** Active le mode silencieux (volume nul).
src	**Source.** Indique l'adresse de la source du média (si `<source>` n'est pas utilisé).

Autres contenus embarqués : canvas, iframe et object

<canvas> *width, height*

Zone de dessin 2D ou 3D. Surface de dessin transparente, manipulable avec JavaScript. Pour la syntaxe complète, se référer aux ouvrages détaillant toutes les fonctions de dessin.

```
<canvas width="640" height="480" id="dessin">Texte alternatif</canvas>
<script>
var canvas = document.getElementById('dessin');
var ctx = canvas.getContext('2d');
ctx.fillRect(0,0,50,50);
</script>
```

<iframe> *src, srcdoc, name, width, height,* ***sandbox, seamless***

Cadre (contexte) de navigation imbriqué. Embarque dans le document une navigation sur un autre document externe, à l'adresse `src`, en allouant des dimensions définies par `width` (largeur) et `height` (hauteur).

L'attribut `seamless` dit au navigateur de rendre le contenu comme si le fichier faisait partie intégrante du document dans lequel il est appelé (par exemple, pour l'application des styles CSS).

L'attribut `sandbox` précise les règles de sécurité à autoriser : `allow-same-origin`, `allow-forms`, `allow-scripts`, `allow-top-navigation`. Sa seule présence sans valeur désactive toutes les autorisations.

```
<iframe src="pub.html" width="300" height="150">
</iframe>
```

<object> *data, type, name, width, height, form, usemap*

Objet externe. Permet de faire appel à un objet externe, souvent interprété via une extension (plug-in), par exemple pour Flash ou Java. Le type MIME de l'objet est précisé par l'attribut `type` et les dimensions par `width` (largeur) et `height` (hauteur).

```
<object type="application/x-java-applet">
  <param name="code" value="MaClasse">
  <param name="archive" value="Applet.jar" />
  <p>Texte alternatif : Java n'est pas installé.</p>
</object>
```

<param>

Paramètre d'objet. Définit les paramètres d'exécution pour `<object>`.

```
<object type="application/x-shockwave-flash">
  <param name="movie" value="anim.swf">
</object>
```

<embed> *src, type, height, width*

Intégration de contenu externe. Officialisé, cet élément (vide) est proche de `<object>` pour embarquer du contenu à exécuter avec des extensions ; tous ses paramètres sont définis cependant par des attributs « libres » et non par des enfants `<param>`.

```
<embed src="animation.swf" quality="high">
```

Menus et commandes

<menu> *type, label*

Menu. Définit un menu contextuel (`type="context"`) ou une barre d'outils (`type="toolbar"`), regroupant des commandes `<command>`, principalement à destination des applications web.

```
<menu type="toolbar">
  <li>
    <menu label="Fichier">
      <command onclick="save()">Sauver</command>
      <command onclick="quit()">Quitter</command>
    </menu>
  </li>
  <li>
    <menu label="Edition">
    ...
    </menu>
  </li>
</menu>
```

Dans le premier cas, le menu contextuel est lié à un autre élément HTML via l'attribut `contextmenu`.

```
<img contextmenu="menuctx">
<menu type="context" id="menuctx">
  <command label="Copier" onclick="copier()">
  <command label="Supprimer" onclick="suppr()">
</menu>
```

<command> *type, label, icon, disabled*

Commande. Commande de menu, déclenchant habituellement un script.

L'attribut `type` peut prendre la valeur `command` (par défaut), `radio` et `checkbox`.

Tableaux

Les tableaux de données sont conçus pour structurer des informations en lignes et en colonnes. Ils ne faut pas les utiliser pour la mise en page du document, qui doit être assurée par des CSS.

BONNE PRATIQUE L'élément `<table>` n'est pas un outil de mise en page !
Rappelons que les tableaux servent à structurer l'information, sous la forme de lignes et de colonnes. La mise en forme doit toujours être déléguée à une feuille de style CSS.

<table>

Tableau de données. Structure un tableau de données, épaulé par les éléments `<tr>` (ligne du tableau), `<th>` (cellule d'en-tête) et `<td>` (cellule). Les éléments complémentaires sont `<caption>` (légende du tableau), `<thead>` (groupe de cellules d'en-tête), `<tbody>` (groupe de cellules de corps) et `<tfoot>` (groupe de cellules de pied). Pour des raisons d'accessibilité, l'attribut `<summary>` (résumé) est conseillé.

```
<table summary="Résumé du contenu du tableau">
<caption>Légende à afficher</caption>
  <thead>
    <tr>
     <th>En-tête de colonne</th>
      ...
    </tr>
  </thead>
  <tfoot>
    <tr>
     <th>Pied de colonne</th>
      ...
    </tr>
  </tfoot>
  <tbody>
    <tr>
     <th>Cellule d'en-tête</th>
     <td>Cellule</td>
      ...
    </tr>
  </tbody>
</table>
```

Légende

En-tête 1	En-tête 2
Cellule d'en-tête	Cellule
Cellule d'en-tête	Cellule
Cellule d'en-tête	Cellule
Pied de colonne 1	Pied de colonne 2

<caption>

Légende d'un tableau de données. Légende associée à un tableau de données, qui doit se situer immédiatement après la balise ouvrante `<table>`.

<tr>

Rangée/ligne de tableau. Regroupe les cellules (`<td>` ou `<th>`) d'une même ligne dans un tableau.

<th>

scope, colspan, rowspan, headers

Cellule d'en-tête. Désigne une cellule d'en-tête (intitulé de colonne ou de ligne) d'un tableau de données. Les attributs `colspan` et `rowspan` définissent respectivement le nombre de cellules fusionnées en colonnes et en lignes. L'attribut `scope` spécifie sur quelles cellules s'applique l'en-tête (`row` : ligne, `col` : colonne, `rowgroup`, `colgroup`). L'attribut `headers` indique une liste d'identifiants pour lesquels s'applique l'en-tête.

<td>

colspan, rowspan, headers

Cellule de tableau. Regroupe le contenu d'une cellule de tableau. Les attributs `colspan` et `rowspan` définissent respectivement le nombre de cellules fusionnées en colonnes et en lignes.

<thead>

En-tête de tableau. Regroupe les cellules d'en-tête d'un tableau de données. Les éléments `<thead>` et `<tfoot>` doivent être déclarés avant le(s) `<tbody>`.

<tbody>

Corps de tableau. Regroupe les cellules du corps d'un tableau de données.

<tfoot>

Pied de tableau. Regroupe les cellules du pied d'un tableau de données.

<col-group>

span

Groupe de colonnes. Regroupe une ou plusieurs colonnes définies par `<col>`, avant tout élément `<thead>`, `<tbody>`, `<tfoot>` et `<tr>`. L'attribut `span` définit le nombre de colonnes sur lesquelles cet élément doit s'étendre.

<col>

span

Colonne(s) d'un groupe. Permet d'appliquer des propriétés communes à une colonne dans un tableau de données, par exemple des styles CSS.

```
<table>
  <colgroup>
    <col span="2" style="color:red" />
    <col style="background:green" />
  </colgroup>
  <tr>
    <td>Cellule 1</td>
    <td>Cellule 2</td>
    <td>Cellule 3</td>
  </tr>
</table>
```

Liens

L'essence même du Web : le lien permet de naviguer d'une page à l'autre.

<a>
*href, target, rel, hreflang, **media**, type*

Lien hypertexte. Crée un lien hypertexte permettant de naviguer au sein de la page ou vers une autre page HTML. Ses attributs courants sont `href` (destination du lien), `id` (pour créer une ancre) et `hreflang` (langue de la destination). On peut préciser `media` (type de média pour lequel le lien est prévu) et `type` (type MIME de la ressource destination).

```
<a href="http://html5.blup.fr/">Découvrir HTML5</a>
```

Les éléments contenus deviennent cliquables (par exemple, des images). Avec HTML5, il est autorisé d'y placer des éléments de type bloc.

```
<a href="http://html5.blup.fr/" title="Découvrir HTML5">
<img src="go.png" alt="">
</a>
```

Les relations existant en complément de `<link>` sont `bookmark`, `nofollow`, `noreferrer`, `tag`.

<link>
*href, rel, hreflang, **media**, type, **sizes***

Relation externe. Déclare un lien vers des ressources externes, couramment employé pour lier une feuille de style CSS au document HTML avec le type `stylesheet` ou une icône avec le type `icon`.

```
<link rel="stylesheet" type="text/css" href="styles.css"
  media="screen">
```

D'autres types de relations : `alternate`, `author`, `help`, `license`, `next`, `prefetch`, `prev`, `search`.

RAPPEL **Les types MIME**
Les types MIME précisent, sous forme d'un type et d'un sous-type, les formats de fichiers rencontrés sur le Web. Ils sont envoyés par le serveur web au navigateur pour lui indiquer comment interpréter le contenu qui lui est délivré. Exemples : `text/html`, `text/javascript`, `image/png`, `application/xml`, `video/mp4`.

Formulaires

<form>
*action, method, enctype, **novalidate**, **autocomplete***

Formulaire. Regroupe un ensemble de champs interactifs permettant de recueillir des informations auprès de l'utilisateur. Le formulaire est généralement validé à l'aide d'un élément `<input type="submit">`. `<form>` ne peut contenir directement que des éléments de type bloc et ne peut pas contenir d'autres éléments `<form>`.

```
<form action="..." method="post">
  <label for="nom">Nom</label>
  <input name="nom" id="nom" type="text">
  <input value="envoyer" name="envoi" type="submit" />
</form>
```

L'attribut `novalidate` indique que le contenu du formulaire ne doit pas être validé (avec `pattern`, `required`, et types `email`, `url`, etc.) avant la soumission.

<fieldset>
Groupe de champs. Regroupe des informations de même thématique dans un formulaire pour en faciliter la compréhension, et doit contenir directement un élément `<legend>`.

```
<form action="..." method="post">
  <fieldset>
    <legend>Informations personnelles</legend>
    <label for="nom">Nom</label>
    <input id="nom" name="nom" type="text">
    <label for="prenom">Prénom</label>
    <input id="prenom" name="prenom" type="text">
  </fieldset>
</form>
```

<legend>
Légende d'un regroupement de champs. Ajoute une légende à un groupe `<fieldset>`, auquel elle est obligatoirement associée. La légende améliore l'accessibilité quand l'élément `<fieldset>` est restitué de manière non visuelle.

Champs de formulaires

<label>
for

Étiquette. Associe une étiquette à un champ de formulaire, où l'attribut `for` se rapportera à l'attribut `id` du champ en question. L'élément `<label>` est utile pour aider à la compréhension du formulaire et accroît l'accessibilité.

```
<label for="nom">Votre nom</label>
<input id="nom" name="nom_utilisateur">
```

Formulaires

<input> *type, name, value,* ***accept, list, pattern, required, placeholder, readonly...***

Champ d'entrée. Un élément interactif (champ) dans un formulaire, défini par son `type` : `text` (texte), `password` (mot de passe), `checkbox` (case à cocher multiple), `radio` (case à cocher unique), `submit` (validation), `reset` (effacement des champs), `file` (fichier), `hidden` (champ masqué), `image` (soumission avec une image), `button` (bouton générique).

```
<input name="nom" type="text" value="">
<input type="submit" value="Valider">
<input type="radio" checked="checked">
```

Avec HTML5, les nouveaux types : `datetime` (date et heure), `datetime-local` (localisées avec fuseau horaire), `date` (date), `month` (mois), `time` (heure), `week` (semaine), `number` (valeur numérique), `range` (intervalle), `email` (e-mail), `url` (adresse), `search` (recherche), `tel` (numéro téléphone), `color` (code couleur).

Avec `type="file"`, l'attribut `accept` précise les types MIME des fichiers acceptés.

```
<input type="file" accept="image/*">
```

<button>

Bouton. Crée un bouton étiqueté dans un formulaire. En cliquant sur ce bouton, une action particulière peut être déclenchée avec JavaScript.

```
<button>Cliquez-moi</button>
```

<textarea> *rows, cols,* ***wrap***

Zone d'entrée texte multiligne. Une zone de texte de plusieurs lignes et de plusieurs colonnes pour la saisie de textes de longueurs variables.

```
<textarea cols="60" rows="10">Le texte</textarea>
```

L'attribut `wrap` accepte les valeurs `hard` (insertion de retours à la ligne dans la valeur soumise pour ne pas excéder la valeur de `cols`) et `soft` (pas de retour à la ligne).

<select> *multiple*

Liste de choix. Renferme une série d'options qui seront présentées à l'utilisateur dans une liste déroulante. Ne peut contenir directement que des éléments `<optgroup>` ou `<option>`.

```
<select name="fruits">
  <option>fruits du marché</option>
  <option>fruits exotiques</option>
  <option>fruits avariés</option>
</select>
```

<option> *label, value, selected, disabled*

Option de liste de choix. Choix offert dans une liste `<select>` (ou `<optgroup>`) ou `<datalist>`. L'attribut `label` est l'étiquette présentée à l'utilisateur, et l'attribut `value` porte la valeur transmise à la soumission du formulaire.

<optgroup> *label, disabled*

Groupe d'options. Regroupe des options `<option>` sous une même étiquette définie par l'attribut obligatoire `label`.

Autres

<output> 5 *name,* ***for, form***

Résultat de calcul. Balise un contenu, résultat d'une opération dans un formulaire, (par exemple, une somme calculée par JavaScript dans un panier de commande). L'attribut `form` définit à quel formulaire se rapporte cette sortie (si `<output>` est placé en dehors de `<form>`). L'attribut `for` reçoit une liste d'`id`, séparés par des espaces, des éléments entrant en jeu pour le calcul.

```
<form onsubmit="return false;" oninput="out.value = parseInt
(a.value)
+ parseInt(b.value);">
  <input name="a" id="a" type="number"> +
  <input name="b" id="b" type="number"> =
  <output name="out" for="a b"></output>
</form>
```

<progress> 5 ***value, max***

Progression. Représente une barre de progression (par exemple, pour une tâche nécessitant du temps, telle qu'un envoi de fichier), manipulable avec JavaScript.

```
<progress max="100" value="30" id="pg"><span>30</span>%</progress>
<script>document.getElementById('pg').value=50;</script>
```

<meter> 5 ***min, max, low, high, optimum, value***

Jauge. Représente une jauge, reflétant un état mesuré, avec un intervalle connu (un minimum `min` et un maximum `max`, et optionnellement, des valeurs basses `low`, haute `high` et optimale `optimum`).

```
Billets vendus : <meter min="0" max="50" value="40">40 sur 50
</meter>
```

<datalist> 5 **Liste de suggestions.** Regroupe un ensemble de zéro ou plusieurs éléments `<option>` qui représentent des choix prédéfinis pour d'autres contrôles d'entrée, par exemple un champ de texte ou de recherche, y faisant référence grâce à l'attribut `list`.

```
<input list="choix" name="pays">
<datalist id="choix">
  <option label="France" value="FR">
  <option label="Suisse" value="CH">
  <option label="Canada" value="CA">
</datalist>
```

<keygen> 5 *challenge, keytype* **Générateur de clés.** Contrôle avancé générant une paire de clés (publique/privée) de type `keytype` (par exemple, `rsa`).

Attributs de formulaire

Ces attributs s'appliquent à la quasi-totalité des éléments de formulaire.

name **Nom du champ pour la validation.** Donne un nom à l'élément d'entrée, pour la réception côté serveur dans les valeurs GET/POST, voire la manipulation avec JavaScript. Ne pas confondre avec `id`.

```
<label for="pr">Prénom</label>
<input name="prenom" type="text" id="pr">
```

disabled **Désactivation.** Désactive le contrôle de l'élément (grisé).

form **Formulaire lié.** Permet d'associer explicitement un champ avec un conteneur `<form>` (via son identifiant) qui ne serait pas son ancêtre naturel.

maxlength **Longueur maximale du contenu.** Fixe la longueur maximale du contenu qui peut être entré dans le champ (nombre de caractères).

size **Taille par défaut.** Indique la largeur visible (nombre de caractères) d'un champ de formulaire recevant du texte.

value **Valeur par défaut.** Valeur transmise au serveur en GET/POST, affichée dans le champ à l'utilisateur si l'attribut est présent dès le chargement du document.

readonly **Lecture seule.** Avec la présence de cet attribut, la valeur ne peut être modifiée par l'utilisateur.

placeholder **Indication ou exemple de saisie.** La valeur de cet attribut donne une indication de saisie à l'utilisateur. Se masque dès le début de la saisie ou à la réception du focus.

```
<input placeholder="Ville, pays, code..." name="geo">
```

required **Champ requis.** Le champ doit être complété par l'utilisateur pour permettre la validation du formulaire.

pattern **Masque de saisie.** Spécifie quelle forme doit prendre la valeur entrée pour être valide. Pour la syntaxe, se référer aux expressions régulières.

```
<input name="codepays" pattern="[A-Z]{3}">
<input type="tel" pattern="^0[0-9]+$">
```

autocomplete **Autocomplétion.** Active (`on`) ou désactive (`off`) l'autocomplétion.

autofocus **Autofocus.** Donne le focus (par exemple, le contrôle au clavier) à l'élément dès le chargement de la page.

list **Liste de suggestions.** Associe une liste `<datalist>` à l'élément en faisant référence à l'attribut `id` de celle-ci.

multiple **Valeurs multiples.** Permet la sélection ou l'entrée de plusieurs valeurs pour les types `email` et `file`.

dirname **Direction de la saisie.** Attribue un nom complémentaire pour la réception côté serveur de l'information de direction de la saisie pour ce champ (valeur `rtl` ou `ltr`).

```
<input type="text" name="commentaire" dirname="commentaire.dir">
```

Éléments obsolètes

Certains éléments présents dans HTML4 et dans les précédentes versions sont désormais obsolètes, car remplacés avantageusement par les feuilles de style CSS ou par d'autres fonctionnalités mieux pensées. Il s'agit de `<acronym>`, `<applet>`, `<basefont>`, `<big>`, `<center>`, `<dir>`, `<font>`, `<frame>`, `<frameset>`, `<isindex>`, `<noframes>`, `<strike>`, `<tt>`, `<u>`, `<xmp>`.

Attributs globaux applicables à tout élément HTML5

contenteditable	**Contenu éditable dynamiquement.** L'élément devient modifiable dans le navigateur.
contextmenu	**Menu contextuel.** Lie un menu contextuel déclaré par `<menu>` grâce à son `id`.
draggable	**Élément déplaçable.** Active (`true`) ou désactive (`false`) la possibilité de déplacer un élément, notamment dans le cadre du Drag & Drop.
dropzone	**Zone de dépôt.** Active l'élément comme zone de dépôt pour le Drag & Drop. Les valeurs possibles sont `copy` : copie des données déposées, `move` : déplacement des données déposées, `link` : création d'un lien vers les données d'origine, `string:text/plain` acceptation de texte Unicode, `file:type` acceptation de types de fichiers. `<div dropzone="copy file:image/png file:image/gif file:image/jpeg"></div>`
hidden	**Contenu inaccessible.** Si cet attribut est présent, l'élément n'est plus, ou n'est pas encore, pertinent (par exemple, une étape supplémentaire dans un processus d'inscription en ligne). Le navigateur ne doit pas l'afficher. À ne pas confondre avec `display:none` en CSS qui relève purement de la présentation. Retirer l'attribut `hidden` avec JavaScript réactive l'accès à l'élément et à son contenu.
spellcheck	**Correction orthographique.** Active (valeur `on`) ou désactive (valeur `false`) la vérification orthographique interne au navigateur. `<input type="text" name="nomfamille" spellcheck="off">`

accesskey	**Touche d'accès rapide.** Méthode d'accès direct au clavier (par combinaison de touches variées selon le système) aux éléments recevant le focus.
	`<a href="/html" accesskey="h">HTML</a>`
	`<a href="/css" accesskey="c">CSS</a>`

class	**Classe.** Confère une classe à un ou plusieurs éléments HTML, qui peut être utilisée dans la feuille de style CSS pour attribuer des propriétés communes à plusieurs de ces éléments, ou avec JavaScript et la méthode `document.getElementsByClassName("classe")`.
id	**Identifiant unique.** Attribue un identifiant unique à l'élément HTML pour le désigner tant avec CSS `#id` qu'avec JavaScript `document.getElementById("id")`.
dir	**Direction du texte.** Indique la direction du texte contenu, avec pour valeurs `auto`, `ltr` (*left to right*) : gauche à droite, `rtl` (*right to left*) : droite à gauche.
lang	**Langue.** Indique par un code langue celle dans laquelle le contenu texte est rédigé.
style	**Styles spécifiques.** Attribue une ou plusieurs propriétés de style CSS directement à l'élément, en général prioritaires sur la feuille de style. `<p style="color:red">Rouge</p>`
tabindex	**Ordre de tabulation.** Définit un ordre de tabulation (différent de l'ordre naturel) pour la navigation au clavier, parmi les éléments qui peuvent recevoir le focus, avec pour valeurs possibles : un entier négatif : peut recevoir le focus mais n'est pas accessible dans la séquence de navigation ; zéro : ordre automatique ; un entier positif : ordre déterminé par la séquence.
title	**Titre.** Fournit des informations à propos d'un élément, souvent affichées par le navigateur sous forme d'infobulle lorsque le curseur survole cet élément.

Chez le même éditeur…

HTML 5. Une référence pour le développeur web, R. Rimelé
CSS avancées : vers HTML 5 et CSS 3, R. Goetter
CSS 2 : pratique du design web, 3e éd., R. Goetter
Mémento XHTML, 2e éd., R. Goetter
Mémento Ergonomie web, 2e éd., A. Boucher
Mémento Sites web : les bonnes pratiques, 3e éd., É. Sloïm
Mémento MySQL 5, 2e éd., R. Rimelé
Mémento PHP 5 et SQL, 2e éd., C. Pierre de Geyer et G. Ponçon

Code éditeur : G13420
ISBN : 978-2-212-13420-9

Conception : Nord Compo

Merci d'avoir acheté ce mémento.

Vous pouvez le retrouver ainsi que toute la collection en version papier dans toutes les bonnes librairies !

Découvrez toute la collection mémento **sur :** www.eyrolles.com

Raphaël Goetter
Mémento XHTML
978-2-212-12541-2

Amélie Boucher
Mémento - Ergonomie web
978-2-212-12698-3

Elie Sloïm , Opquast
Mémento Sites web - Les bonnes pratiques
978-2-212-12802-4

Rodolphe Rimelé
Mémento MySQL 5
978-2-212-13419-3

Chez le même éditeur :

Raphaël Goetter
CSS avancées
978-2-212-13405-6

Rodolphe Rimelé
HTML5
978-2-212-12982-3

Tous les livres informatiques sont sur
www.editions-eyrolles.com

Et pour toutes les nouveautés numériques, retrouvez-nous sur Twitter et Facebook

@ebookEyrolles
EbooksEyrolles