

## Les nouveaux éléments

Les éléments suivants ont été introduits pour une meilleure structure :

- `section` représente la section d'un document générique ou d'une application. Il peut être utilisé avec `h1-h6` pour indiquer la structure du document.
- `article` représente un bout de contenu indépendant d'un document, tel qu'une entrée d'un blog ou d'un article de journal.
- `aside` représente un bout de contenu qui n'est qu'en léger rapport avec le reste de la page.
- `header` représente l'en-tête d'une section.
- `footer` représente un pied de page pour une section et peut contenir des renseignements sur l'éditeur, sur les droits d'auteur, et cetera.
- `nav` représente une section du document à l'intention de la navigation.
- `dialog` peut être utilisé pour baliser une conversation comme suit :

```
<dialog>
  <dt> Costello
  <dd> Regarde, tu dois être le premier joueur ?
  <dt> Abbott
  <dd> Certainement.
  <dt> Costello
  <dd> Qui joue en premier ?
  <dt> Abbott
  <dd> C'est juste.
  <dt> Costello
  <dd> Quand tu rembourseres chaque mois le premier joueur, qui ramasse
l'argent ?
  <dt> Abbott
  <dd> Chaque dollar pour ça.
</dialog>
```

- `figure` peut être utilisé pour associer une légende à un contenu imbriqué, tel qu'un graphique ou une vidéo :

```
<figure>
  <video src=ogg>...</video>
  <legend>Exemple</legend>
</figure>
```

Il y a bien d'autres nouveaux éléments :

- `audio` et `video` pour du contenu multimédia. Les deux éléments fournissent une API pour que les éditeurs d'application puissent coder leur propre interface utilisateur ; mais, il y a aussi une manière de déclencher une interface utilisateur fournie par l'agent utilisateur. Les éléments `source` sont utilisés avec ces éléments s'il y a de multiples flux disponibles de différents types.
- `embed` est utilisé pour du contenu d'un plug-in.
- `m` représente une série de texte marqué.

- `meter` représente une mesure, telle que l'espace de disque.
- `time` représente une date et/ou une heure.
- `canvas` est utilisé pour afficher des graphiques bitmap dynamiques sur le moment, tels que des graphes, des jeux, et cetera.
- `command` représente une commande que l'utilisateur peut invoquer.
- `datagrid` représente une représentation interactive d'une liste arborescente ou de données tabulaires.
- `details` représente des renseignements supplémentaires ou des contrôles que l'utilisateur peut obtenir sur demande.
- `datalist`, avec le nouvel attribut `list` d'`input`, est utilisé pour créer des boîtes combinées :

```
<input list=browsers>
<datalist id=browsers>
  <option value="Safari">
  <option value="Internet Explorer">
  <option value="Opera">
  <option value="Firefox">
</datalist>
```

- Les éléments `datatemplate`, `rule` et `nest` fournissent un mécanisme de gabarit pour le HTML.
- `event-source` est utilisé pour « capturer » des événements envoyés par le serveur.
- `output` représente un type de sortie, telle que celle d'un calcul fait au moyen d'un script.
- `progress` représente une progression de tâche, telle que le téléchargement ou quand une série d'opérations lourdes est exécutée.

L'attribut `type` de l'élément `input` a maintenant les nouvelles valeurs suivantes :

- `datetime`
- `datetime-local`
- `date`
- `month`
- `week`
- `time`
- `number`
- `range`
- `email`
- `url`

L'idée de ces nouveaux types est que l'agent utilisateur peut fournir à l'utilisateur une interface, telle que la sélection d'une date dans le calendrier ou une intégration au carnet d'adresses de l'utilisateur, et soumettre au serveur un format défini. Cela donne à l'utilisateur un meilleur contact, comme sa saisie est vérifiée avant son envoi vers le serveur, ce qui veut dire moins de temps d'attente pour la réaction.

## Nouveaux attributs

HTML 5 a introduit plusieurs nouveaux attributs dans divers éléments qui faisaient déjà partie de HTML 4 :

- Les éléments `a` et `area` ont maintenant un attribut `media` en cohérence avec l'élément `link`. C'est purement à titre indicatif.
- Les éléments `a` et `area` ont un nouvel attribut appelé `ping` qui spécifie une liste d'URIs séparées par une espace qui doivent être épinglées quand l'hyperlien est suivi. Actuellement, le suivi de l'utilisateur est essentiellement effectué au moyen de redirections. Cet attribut permet à l'agent utilisateur d'informer les utilisateurs de quels URIs vont être épinglées, en plus de donner aux utilisateurs soucieux de confidentialité une manière de le désactiver.
- L'élément `area`, par cohérence, a maintenant les attributs `hreflang` et `rel`.
- L'élément `base` peut maintenant avoir un attribut `target`, aussi bien principalement par cohérence avec l'élément `a` que parce qu'il était déjà largement supporté. Aussi, l'attribut `target` pour les éléments `a` et `area` n'est-il plus désapprouvé, comme il est utile dans des applications Web, par exemple conjointement avec `iframe`.
- L'attribut `value` pour l'élément `li` n'est plus désapprouvé, comme il n'est pas un attribut de présentation. Il en va de même pour l'attribut `start` de l'élément `ol`.
- L'élément `meta` a maintenant un attribut `charset`, comme il était déjà supporté et qu'il fournit une façon plus sympathique de spécifier le [codage des caractères](#) du document.
- Un nouvel attribut `autofocus` peut être spécifié dans les éléments `input` (sauf quand l'attribut `type` est `hidden`), `select`, `textarea` et `button`. Il fournit une manière déclarative de cibler un champ de contrôle de formulaire durant le chargement de la page. L'utilisation de cette caractéristique devrait augmenter l'expérience de l'utilisateur, comme ce dernier peut le désactiver s'il ne l'aime pas, par exemple.
- Le nouvel attribut `form` des éléments `input`, `output`, `select`, `textarea`, `button` et `fieldset` permet aux champs de contrôle d'être associés à plus d'un formulaire.
- Les éléments `input`, `button` et `form` ont un nouvel attribut `replace`, qui affecte ce qui sera fait du document après qu'un formulaire a été soumis.
- Les éléments `form` et `select` (ainsi que l'élément `datalist`) ont un attribut `data`, qui tient compte du préremplissage automatique des champs de contrôle de formulaire, dans le cas de `form`, ou du champ de contrôle de formulaire, dans le cas de `select` et de `datalist`, avec des données provenant du serveur.
- Le nouvel attribut `required` s'applique à `input` (sauf quand l'attribut `type` est `hidden`, `image` ou un type de bouton tel que `submit`) et à `textarea`. Il indique que l'utilisateur doit renseigner une valeur afin de soumettre le formulaire.
- Les éléments `input` et `textarea` ont un nouvel attribut appelé `inputmode`, qui laisse entendre à l'interface utilisateur quel genre de saisie est attendu.
- Vous pouvez maintenant désactiver un `fieldset` entier en y utilisant l'attribut `disabled`.

Ce n'était pas possible auparavant.

- L'élément `input` a plusieurs nouveaux attributs pour spécifier des contraintes : `autocomplete`, `min`, `max`, `pattern` et `step`. Comme mentionné auparavant, il a aussi un nouvel attribut `list`, qui peut être utilisé avec les éléments `datalist` et `select`.
- `input` et `button` ont aussi un nouvel attribut `template`, qui peut être utilisé pour des gabarits de répétition.
- L'élément `menu` a trois nouveaux attributs : `type`, `label` et `autosubmit`. Ils permettent à l'élément de se transformer en un menu tel qu'il est trouvé dans des interfaces utilisateur typiques, ainsi que de pourvoir à des menus contextuels conjointement avec l'attribut global `contextmenu`.
- L'élément `style` a un nouvel attribut `scoped`, qui peut être utilisé pour activer des feuilles de style portées. Les règles de style dans un tel élément `style` ne s'appliquent qu'à l'arborescence locale.
- L'élément `script` a un nouvel attribut appelé `async`, qui influe sur le chargement et l'exécution du script.
- L'élément `html` a un nouvel attribut appelé `manifest`, qui pointe une liste de cache d'application utilisée conjointement avec l'API pour des applications Web hors connexion.

Plusieurs attributs de HTML 4 s'appliquent maintenant à tous les éléments. Ils sont appelés attributs globaux : `class`, `dir`, `id`, `lang`, `tabindex` et `title`.

Il y a aussi plusieurs nouveaux attributs globaux :

- L'attribut `contenteditable` indique que l'élément est une zone éditable. L'utilisateur peut en changer le contenu et manipuler le balisage.
- L'attribut `contextmenu` peut être utilisé pour pointer un menu contextuel fourni par l'éditeur.
- L'attribut `draggable` peut être utilisé conjointement avec la nouvelle API « glisser-déposer ».
- L'attribut `irrelevant` indique qu'un élément n'est pas encore, ou n'est plus, pertinent.

Ce qui suit, ce sont les attributs pour le modèle de répétition. Ce sont des attributs globaux et, en tant que tels, peuvent être utilisés sur tous les éléments HTML, ou sur tout élément dans tout autre espace de nom, dont les attributs sont dans l'espace de nom <http://www.w3.org/1999/xhtml> :

- `repeat`
- `repeat-start`
- `repeat-min`
- `repeat-max`

HTML 5 fait aussi de tous les attributs gestionnaires d'événements de HTML 4 qui prennent la forme *onnom de l'événement* des attributs globaux et en ajoute plusieurs nouveaux pour de nouveaux événements qu'il définit, tels que l'attribut `onmessage`, qui peut être utilisé avec le nouvel élément `event-source` et l'API de messagerie de documents multiples.

## Éléments ayant changé

Ces éléments ont une signification légèrement modifiée en HTML 5 pour mieux refléter la manière dont ils sont utilisés sur le Web ou pour les rendre plus utiles :

- L'élément `a` sans attribut `href` représente maintenant un « lien fictif ».
- L'élément `address` est maintenant porté par le nouveau concept de sectionnement.
- L'élément `b` représente maintenant une portée de texte à déporter stylistiquement de la prose normale sans transmettre d'importance en plus, telle que les mots clés dans un résumé de document, des noms de produits dans un article, ou d'autres portées de texte dont la présentation typographique typique est en gras.
- L'élément `hr` représente maintenant une coupure thématique au niveau du paragraphe.
- L'élément `i` représente maintenant une portée de texte dans une voix ou un mode alterné, ou à défaut déportée de la prose normale, telle qu'une désignation taxonomique, un terme technique, une expression idiomatique d'une autre langue, une pensée, un nom de navire, ou une autre prose dont la présentation typographique typique est en italique. L'usage varie largement selon la langue.
- Pour l'élément `label`, le navigateur ne devrait plus déplacer la cible de l'étiquette vers le champ de contrôle, à moins qu'un tel comportement ne soit standard pour l'interface utilisateur de la plateforme de base.
- L'élément `menu` est redéfini pour être utile à de vrais menus.
- L'élément `small` représente maintenant un petit caractère (pour des commentaires secondaires et des mentions légales).
- L'élément `strong` représente maintenant l'importance plutôt que la forte emphase.

## Éléments absents

Les éléments mentionnés dans cette section ne sont pas à utiliser par les éditeurs. Les agents utilisateurs devront, pourtant, les supporter et HTML 5 obtiendra une section d'affichage en temps voulu qui dira exactement comment. (L'élément `isindex`, par exemple, est déjà supporté par l'analyseur.)

Les éléments suivants ne figurent pas en HTML 5 parce que leur effet est de pure présentation et, par conséquent, mieux pris en charge par CSS :

- `basefont`
- `big`
- `center`
- `font`, bien qu'il soit autorisé quand un éditeur WYSIWYG l'insère, à cause des limitations à l'état de l'art de l'interface utilisateur pour ces éditeurs.
- `s`
- `strike`
- `tt`

- u

Les éléments suivants ne figurent pas en HTML 5 parce que leur usage affectait négativement l'utilisabilité et l'accessibilité pour l'utilisateur final :

- frame
- frameset
- noframes

Les éléments suivants ne sont pas inclus parce qu'ils n'ont pas été souvent utilisés, ont créé de la confusion ou peuvent être pris en charge par d'autres éléments :

- acronym n'est pas inclus parce qu'il a créé beaucoup de confusions. Les éditeurs doivent utiliser abbr pour les abréviations.
- applet a été rendu obsolète au profit d'object.
- L'usage d'isindex peut être remplacé par celui de champs de contrôle de formulaire.
- dir a été rendu obsolète au profit d'ul.

Finalement, le noscript n'est conforme qu'à la syntaxe HTML. Il n'est pas inclus dans la syntaxe XML, comme son usage se fie à un analyseur HTML.

## Attributs absents

Quelques attributs de HTML 4 ne sont plus autorisés en HTML 5. S'ils ont besoin d'avoir un impact sur les agents utilisateurs pour des raisons de compatibilité, la manière dont ils devraient fonctionner dans ces scénarios est définie.

- L'attribut accesskey dans a, area, button, input, label, legend et textarea.
- Les attributs rev et charset dans link et a.
- Les attributs shape et coords dans a.
- L'attribut longdesc dans img et iframe.
- L'attribut target dans link.
- L'attribut nohref dans area.
- L'attribut profile dans head.
- L'attribut version dans html.
- L'attribut name dans map, img, object, form, iframe, a (à la place, utilisez id).
- L'attribut scheme dans meta.
- Les attributs archive, classid, codebase, codetype, declare et standby dans object.
- Les attributs valuetype et type dans param.
- Les attributs charset et language dans script.
- L'attribut summary dans table.
- Les attributs headers, axis et abbr dans td et th.
- L'attribut scope dans td.

En outre, HTML 5 n'a aucun des attributs de présentation qui figuraient en HTML 4, comme ils sont mieux pris en charge par CSS :

- L'attribut align dans caption, iframe, img, input, object, legend, table, hr,

div, h1, h2, h3, h4, h5, h6, p, col, colgroup, tbody, td, tfoot, th, thead, tr et body.

- Les attributs alink, link, text et vlink dans body.
- L'attribut background dans body.
- L'attribut bgcolor dans table, tr, td, th et body.
- L'attribut border dans table, img et object.
- Les attributs cellpadding et cellspacing dans table.
- Les attributs char et charoff dans col, colgroup, tbody, td, tfoot, th, thead et tr.
- L'attribut clear dans br.
- L'attribut compact dans dl, menu, ol et ul.
- L'attribut frame dans table.
- L'attribut frameborder dans iframe.
- L'attribut height dans iframe, td et th.
- Les attributs hspace et vspace dans img et object.
- Les attributs marginheight et marginwidth dans iframe.
- L'attribut noshade dans hr.
- L'attribut nowrap dans td et th.
- L'attribut rules dans table.
- L'attribut scrolling dans iframe.
- L'attribut size dans hr, input et select.
- L'attribut style dans tous les éléments, à l'exception de font.
- L'attribut type dans li, ol et ul.
- L'attribut valign dans col, colgroup, tbody, td, tfoot, th, thead et tr.
- L'attribut width dans hr, table, td, th, col, colgroup, iframe et pre.