

HTML 5

**Une référence
pour le développeur web**

Rodolphe Rimelé

Préface de Raphaël Goetter

© Groupe Eyrolles, 2011, ISBN : 978-2-212-12982-3

EYROLLES



Fonctionnalités modifiées et obsolètes



Différences HTML 5 par rapport à HTML 4

Une publication du W3C recense les nouveaux éléments et attributs, ceux qui ont été supprimés, et ceux qui ont été modifiés. D'autres différences y sont mises en évidence notamment pour l'API et la syntaxe en général.

RESSOURCE Spécification W3C

HTML 5 differences from HTML 4

▶ <http://www.w3.org/TR/html5-diff/>

Fonctionnalités obsolètes

Ces pratiques ne généreront pas d'erreur au validateur, mais des avertissements :

- attribut `http-equiv` de l'élément `meta` : l'attribut `lang` doit être utilisé en remplacement ;
- attribut `border` sur l'élément `img` : CSS doit être utilisé en remplacement ;
- attribut `language` sur l'élément `script` : il doit être omis ou remplacé par l'attribut `type` avec la valeur `text/javascript` ;
- attribut `name` sur l'élément `a` : il doit être omis ou remplacé par l'attribut `id`. Une exception cependant : s'il est présent, il ne doit pas être vide, mais posséder la même valeur que l'`id`.

Fonctionnalités obsolètes non conformes

Éléments

Ces éléments sont entièrement obsolètes et **ne doivent plus être utilisés** :

- `applet` : remplacé par `embed` ou `object` ;
- `acronym` : remplacé par `abbr` ;
- `bgsound` : remplacé par `audio` ;
- `dir` : remplacé par `ul` ;
- `frame`, `frameset`, `noframes` : remplacé par `iframe` et CSS, ou par les langages interprétés côté serveur pour générer des pages complètes ;
- `isindex` : utiliser un formulaire `form` combiné à un champ texte `input` ;
- `listing`, `xmp` : remplacé par `pre` et `code` ;
- `noembed` : remplacé par `object` quand une alternative est nécessaire ;
- `plaintext` : utiliser le type `MIMEtext/plain` ;
- `strike` : remplacé par `del` ;
- `basefont`, `big`, `blink`, `center`, `font`, `marquee`, `multicol`, `nobr`, `spacer`, `tt`, `u` : remplacé par les propriétés CSS équivalentes.

Précisions au sujet de `tt` :

- utiliser `kbd` pour baliser une entrée au clavier ;
- utiliser `var` pour baliser une variable ;
- utiliser `code` pour baliser un code source ;
- utiliser `samp` pour baliser une sortie de données.

Précisions au sujet de `u` :

- utiliser `em` pour baliser une emphase ;
- utiliser `b` pour baliser des mots-clés ;
- utiliser `mark` pour baliser un texte surligné ou marquant une référence.

Attributs

Ces attributs sont obsolètes, bien qu'appartenant à des éléments qui font encore partie du langage et **ne doivent plus être utilisés** :

- `charset` sur `a` et `link` : utiliser un en-tête HTTP Content-Type ;
- `coords` et `shape` sur `a` : utiliser l'élément `area` au lieu de `a` ;
- `methods` sur `a` et `link` : utiliser HTTP OPTIONS ;
- `name` sur `a`, `embed`, `img` et `option` : utiliser l'attribut `id` ;

- `rev` sur `a` et `link` : utiliser l'attribut `rel` ;
- `urn` sur `a` et `link` : utiliser l'attribut `href` ;
- `nohref` sur `area` : facultatif, ne plus utiliser ;
- `profile` sur `head` : facultatif, ne plus utiliser ;
- `version` sur `html` : facultatif, ne plus utiliser ;
- `usemap` sur `input` : utiliser l'élément `img` au lieu de `input` ;
- `longdesc` sur `iframe` et `img` : utiliser un élément `a` ou une image map ;
- `lowsrc` sur `img` : utiliser une image compressée progressive (présente dans `src`) ;
- `target` sur `link` : facultatif, ne plus utiliser ;
- `archive`, `classid`, `code`, `codebase`, `codetype` sur `object` : utiliser `data` et `type` pour invoquer des extensions et l'élément `param` pour les paramètres ;
- `declare` sur `object` : répéter l'élément `object` complètement ;
- `standby` sur `object` : optimiser la ressource pour un chargement incrémental ;
- `type` et `valuetype` sur `param` : utiliser `name` et `value` sans déclarer le type de valeur ;
- `event` et `for` sur `script` : utiliser les événements DOM ;
- `datapagesize` sur `table` : facultatif, ne plus utiliser ;
- `abbr` sur `td` et `th` : utiliser un texte explicite ou préciser avec `title` ;
- `axis` sur `td` et `th` : utiliser l'attribut `scope` ;
- `datasrc`, `datafld` et `dataformatas` : utiliser XMLHttpRequest.

Pour les attributs suivants, **utiliser les propriétés CSS en remplacement** :

- `alink`, `link`, `marginbottom`, `marginheight`, `marginleft`, `marginright`, `marginintop`, `marginwidth`, `text`, `vlink` sur `body` ;
- `bgcolor` sur `body`, `table`, `td`, `th`, `tr` ;
- `background` sur `body`, `table`, `thead`, `tbody`, `tfoot`, `tr`, `td`, `th` ;
- `clear` sur `br` ;
- `align` sur `caption`, `div`, `hr`, `h1` à `h6`, `iframe`, `input`, `img`, `legend`, `col`, `embed`, `input`, `img`, `legend`, `object`, `p`, `table`, `tbody`, `thead`, `tfoot`, `td`, `th`, `tr` ;
- `char`, `charoff`, `valign`, `width` sur `col`, `tbody`, `thead`, `tfoot`, `td`, `th`, `tr` ;
- `compact` sur `dl` ;
- `hspace`, `vspace` sur `embed`, `input`, `img`, `legend`, `object` ;
- `color`, `noshade`, `size`, `width` sur `hr` ;
- `allowtransparency`, `frameborder`, `marginheight`, `marginwidth`, `scrolling`, sur `iframe` ;
- `border` sur `img`, `object`, `table` ;

- `type` sur `ul`, `li` ;
- `compact` sur `menu`, `ol`, `ul` ;
- `width` sur `pre`, `table` ;
- `height`, `nowrap` sur `td`, `th` ;
- `cellspacing`, `cellpadding`, `frame`, `rules` sur `table`.