

Traduction à revoir

Il existe quatre balises pour la manipulation de tableau de grande échelle. Toutes doivent être placées au début de la ligne pour fonctionner correctement.

### **(:table (attr...):)**

termine n'importe quel tableau précédent (équivalent à (:tableend:)), et génère une nouvelle balise HTML `<table>` avec les paramètres passés avec `attr`. Les attributs et valeurs possibles sont:

- `border` (un entier positif; c'est la largeur des bordures)
- `bordercolor` (un nom de couleur ou son code hex; la couleur des bordures; ne s'affiche pas dans tous les navigateurs)
- `cellspacing` (un entier positif déterminant l'espace en pixel entre les cellules du tableau)
- `cellpadding` (un entier positif déterminant l'espace en pixel entre le contenu d'une cellule et son bord)
- `width` (un entier positif ou un pourcentage pour déterminer la largeur du tableau)
- `bgcolor` (un nom de couleur ou son code hex; la couleur de fond)
- `align` (*left, center ou right*; alignement du texte)
- `summary` (n'apparaît pas à l'affichage; utilisé principalement pour faciliter la navigation des personnes non voyantes)

### **(:cell (attr...):)**

ferme n'importe quelle cellule précédente si elle existe et génère une nouvelle cellule avec les paramètres donnés par `attr`. En HTML, une nouvelle balise "`<td attr>`" est créée ( et éventuellement d'autres balises `<table>`, `<tr>`, et `</td>` si elles sont nécessaires à la construction d'un tableau HTML valide).

- `align` (*left, center ou right*)
- `valign` (*top, middle ou bottom*)
- `colspan` (un entier positif)
- `rowspan` (un entier positif)
- `bgcolor` ((un nom de couleur ou son code hex)
- `width` (un entier positif ou un pourcentage)

### **(:cellnr (attr..):)**

ferme n'importe quelle cellule précédente si elle existe et génère une nouvelle cellule au début de la prochaine colonne du tableau. En HTML, des nouvelles balises "`<tr><td attr>`" sont créées ( et éventuellement d'autres balises `<table>`, `</td>`, et `</tr>` si elles sont nécessaires pour produire du HTML valide).

- `align` (*left, center ou right*)
- `valign` (*top, middle ou bottom*)
- `colspan` (un entier positif)
- `rowspan` (un entier positif)

- bgcolor ((un nom de couleur ou son code hex)
- width (un entier positif ou un pourcentage)

## (:tableend:)

ferme la cellule de tableau précédente et termine tout tableau. Génère des balises HTML `</td>`, `</tr>`, et `</table>` suivant leur nécessité.

Pour les balises `(:table:)`, `(:cell:)`, et `(:cellnr:)` l'auteur peut spécifier n'importe quel attribut qui serait valide dans une balise HTML `<table>` ou `<td>`. Ainsi, en utilisant `rowspan`, `colspan`, etc, il est permis de construire des tableaux complexes. Cependant, il est impossible d'emboîter un `(:table:)` à l'intérieur de `(:cell:)` ou `(:cellnr:)` -- le paragraphe suivant explique pourquoi.

Beaucoup se demandent pourquoi n'utilisons nous pas juste les balises HTML standard (`<table>`, `<tr>`, `<td>`, `<th>`) au lieu d'en créer de nouvelles, offrant la possibilité d'emboîter les tableaux. Il y a deux réponses à ça: premièrement, l'utilisation de balises HTML pour créer un tableau effraie les auteurs/utilisateurs ordinaires (voir [Audiences](#) et [Philosophie de PmWiki](#)), et deuxièmement, il est très facile de concevoir des tableaux HTML défailants qui s'affichent incorrectement (voir pas du tout) sur certains navigateurs. Même des professionnels expérimentés du Web se retrouvent parfois avec des agencements de balises faux, aussi est-il peu réaliste d'exiger de l'utilisateur moyen de toujours y parvenir ou d'être capable de relire le code HTML d'un tableau complexe créé par quelqu'un d'autre.

*Commentaire courant:* L'utilisateur moyen ou débutant n'écrira pas du HTML directement, mais utilisera certainement un outil, tel que FrontPage, ou même MSWord, pour générer son code HTML. Cela est beaucoup plus facile que d'apprendre même le plus simple des langages de balises PmWiki.

*La réponse de Pm:* Et une fois que le code HTML a été généré puis posté, comment quelqu'un d'autre fait-il pour éditer ou modifier le tableau si il n'a pas le fichier original FrontPage ou MSWord utilisé pour le créer ? N'oubliez pas l'aspect "collaboration d'auteurs" de ce projet. Le code HTML que ces outils génèrent est parmi les plus difficiles à lire et à modifier.

Il est difficile d'écrire le code nécessaire à PmWiki pour comprendre et réparer le code de tableaux complexes, c'est pourquoi PmWiki utilise la façon simplifiée décrite ci-dessus. Elle permet néanmoins de répondre à la plupart des besoins courants en matière de tableau (avec la possibilité d'exceptions permettant l'emboîtement de tableau).

Et, cela ne veut pas dire que les tableaux emboîtés sont impossibles dans PmWiki--ils ne peuvent juste pas être créés facilement par un auteur utilisant le code wiki par défaut. Un [administrateur du Wiki](#) pourra bien sûr créer des entêtes, des pieds de page et d'autres configurations locales qui utiliseront les tableaux emboîtés.

### **Exemple 1. Un tableau utilisant cette technique**

```
(:table border=1 cellpadding=5 cellspacing=0:) (:cell:) a1 (:cell:) b1 (:cell:)
c1 (:cell:) d1 (:cellnr:) a2 (:cell:) b2 (:cell:) c2 (:cell:) d2 (:tableend:)
```

produit ...

a1	b1	c1	d1
a2	b2	c2	d2

qui crée la même sortie que:

```
<table border='1' cellpadding='5' cellspacing='0'> <tr> <td>a1</td> <td>b1</td>
<td>c1</td> <td>d1</td> </tr> <tr> <td>a2</td> <td>b2</td> <td>c2</td>
<td>d2</td> </tr> </table>
```

---

<< [Tableaux](#) | [Index documentation](#) | [Styles Wiki](#) >>