Feuilles de style pour le Web
Trucs & astuces CSS

Voir également l'index de tous les trucs et astuces.

Dans cette page:

EM, PX, PT, CM, IN...

CSS offre différentes unités pour exprimer les dimensions. Certaines proviennent de la typographie, comme le point (**pt**) et le pica (**pc**), d'autres sont connues pour leur usage quotidien, comme le centimètre (**cm**) et le pouce (**in**). Et il y a également une unité "magique" inventée spécifiquement pour CSS: le pixel **px**. Est-ce que cela signifie que différentes propriétés nécessitent différentes unités ?

Non, les unités n'ont rien à voir avec les propriétés, mais avec le média de sortie: écran ou papier.

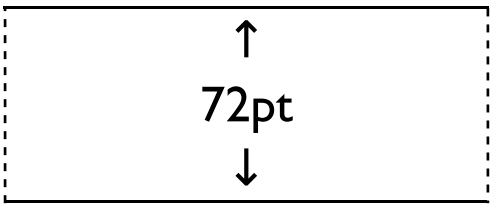
Il n'y a pas de restriction à utiliser telle unité à tel ou tel endroit. Si une propriété accepte une valeur en **px** ('margin: 5px') elle accepte également une valeur en pouces ou en centimètres ('margin: 1.2in; margin: 0.5cm') et vice-versa.

Mais généralement, vous utilisez un ensemble différent d'unités selon que vous affichez sur un écran ou que vous imprimez sur du papier. La table ci-dessous recommande les différents usages:

	Recommandé	**Usage occasionnel use**	**Non recommandé**
Écran	em, px, %	ex	pt, cm, mm, in, pc
Imprimante	em, cm, mm, in, pt, pc, %	px, ex	

La relation entre les unités absolues est la suivante: 1in = 2.54cm = 25.4mm = 72pt = 6pc

✎ Si vous avez un double-décimètre, vous pouvez vérifier la précision de votre support de sortie: voici une boîte de 1in (2.54cm) de hauteur:



Les unités appelées *absolues* (**cm**, **mm**, **in**, **pt** and **pc**) signifient la même chose en CSS que partout ailleurs, *mais uniquement si votre support de sortie a une résolution assez élevée.* Sur une imprimante laser, 1cm devrait correspondre exactement à 1 centimètre. Mais pour des supports avec une basse résolution, comme les écrans d'ordinateurs, CSS n'exige pas cela. Et effectivement, le résultat tend à être différent d'un support à l'autre et d'une implémentation de CSS à l'autre. Il est préférable de réserver ces unités pour des supports haute-résolution et en particulier pour les supports d'impression. Sur les écrans d'ordinateurs ou de portables, vous n'obtiendrez probablement pas ce que vous attendez.

✎ Auparavant, CSS exigeait que les implementations affichent les unités absolues correctement, y compris sur les écrans d'ordinateurs. Mais comme le nombre d'implementations incorrectes dépassait le nombre d'implementations correctes et que cela n'allait pas en s'améliorant, CSS a abandonné cette condition en 2011. Aujourd'hui, les unités absolues doivent fonctionner correctement sur les sorties imprimées et sur les supports haute-résolution uniquement.

CSS de définie pas ce que "haute-résolution" signifie. Mais comme actuellement les imprimantes d'entrée de gamme commencent à 300 dpi et que les écrans haut de gamme atteignent 200 dpi, la limite est probablement entre les 2.

Il y a une autre raison de ne pas employer les unités absolues pour des utilisations autres que l'impression: vous regardez des écrans différents à des distances différentes. 1cm sur un écran d'une station de travail apparaît petit. Mais le même sur un téléphone mobile juste en face de vos yeux apparaît grand. Il est préférable d'utiliser des unités relatives à la place, comme **em**.

Les unités **em** et **ex** dépendent de la police de caractères et peuvent être différentes pour chaque élément du document. L'unité **em** est simplement la taille de la police de caractères. Pour un élément dont la police est de taille 2in, 1em signifie donc 2in. Exprimer des tailles, comme les *margins* et les *paddings*, en **em** signifie qu'elles sont relatives à la taille de la police, et si l'utilisateur a une large police de caractères (sur un grand écran par exemple) ou au contraire une petite police (sur un smartphone), les tailles seront proportionnelles. Les déclarations telles que **text-indent: 1.5em** et **margin: 1em** sont très courantes en CSS.

L' unité **ex** est rarement utilisée. Son but est d'exprimer des tailles relatives à *l'x-height* de la police. Le *x-height* est, schématiquement, la hauteur des plus petites lettres comme le *a, c, m,* or *o.* Les polices qui ont la même taille (donc le même **em**) peuvent varier considérablement concernant la taille de leurs plus petites lettres, et lorsqu'il est important que certaines images par exemple aient le même x-height, l'unité **ex** est disponible.

L'unité **px** est l'unité magique en CSS. Elle n'est pas relative à la police courante ni relative aux unités absolues. L'unité **px** est définie pour être 'petite mais visible', et de telle façon qu'une ligne horizontale de 1px de largeur peut être affichée de façon nette, sans 'arête' (ni anti-aliasing). Ce qui est net, petit et visible dépend du support et de la façon dont il est utilisé: est-ce que vous le tenez près des yeux, comme un téléphone mobile, à la distance d'un bras, comme un écran d'ordinateur, ou à une distance intermédiaire, comme une liseuse? Le **px** n'est donc pas défini comme une

longueur constante, mais comme quelque chose qui dépend du type de matériel et de son usage spécifique.

Pour avoir une idée de l'apparence du **px**, prenez un écran cathodique des années 90: le plus petit point qu'il peut afficher mesure environ 1/100e de pouce (0.25mm) ou un peu plus. L'unité **px** tire son nom de ces pixels d'écrans.

De nos jours, il y a des supports qui peuvent en principe afficher des points nets plus petits (bien que vous puissiez avoir besoin d'une loupe pour les voir). Mais les documents du siècle dernier qui utilisaient des **px** dans CSS sont rendus de la même façon, quelque soit le support. Les imprimantes en particulier peuvent afficher des lignes très nettes avec des détails bien inférieurs à 1px, mais même sur les imprimantes, une ligne de 1px est rendue de la même façon qu'elle l'aurait été sur un écran d'ordinateur. Les supports changent, mais le **px** a toujours la même apparence visuelle.

✎ En fait, CSS requiert que **1px** soit exactement 1/96e de pouce sur toutes les sortie imprimées. CSS considère que les imprimantes, contrairement aux écrans, n'ont pas besoin d'avoir différentes tailles pour **px** pour être en mesure d'imprimer des lignes nettes. Dans un média imprimé, un **px** à donc non seulement la même apparence visuelle d'un média à un autre, mais il est en plus de la même taille (de la même manière que pour **cm**, **pt**, **mm**, **in** et **pc**, comme expliqué ci-dessus).

CSS définie également que les images matricielles (comme les photos) sont affichées par défaut avec 1 pixel d'image correspondant à 1px. Ainsi, une photo avec une résolution 600 x 400 aura 600px de large et 400px de haut. Les pixels de la photo ne correspondant donc pas aux pixels du support de sortie (lequel peut être très petit), mais correspondent aux unités **px**. Cela rend possible le fait d'aligner exactement des images avec d'autres éléments du document, tant que vous utilisez l'unité **px** dans votre feuille de style, et non **pt**, **cm**, etc.

UTILISER EM OU PX POUR LES TAILLES DES POLICES

CSS a héritée de la typographie les unités **pt** (point) et **pc** (pica). Les imprimantes les ont traditionnellement utilisées (ainsi que les unités similaires) de préférence à **cm** ou **in**. Dans CSS il n'y a aucune raison d'utiliser **pt**, utilisez l'unité que vous préférez. Mais il y a une bonne raison de ne pas utiliser **pt** ni aucune autre unité absolue et d'utiliser seulement **em** et **px**.

✎ Voici quelques lignes de différentes épaisseurs. Toutes ou partie peuvent être nettes (mais au minimum les lignes 1px et 2px doivent être visibles et nettes):

0.5pt, 1px, 1pt, 1.5px, 2px

Si les 4 premières lignes apparaissent identiques (ou si la ligne 0.5pt n'est pas affichée), vous avez probablement un écran qui ne peut afficher des points inférieurs à 1px. Si les lignes semblent de plus en plus épaisses, vous regardez probablement cette page sur un écran en haute résolution ou sur du papier. Et si 1pt semble plus épais que 1.5px, vous avez probablement un smartphone.

L'unité magique en CSS, le **px**, est souvent très adaptée, en particulier si le style nécessite l'alignement d'un texte sur des images, ou tout simplement parce que tout ce qui a une largeur de 1px ou d'un multiple de 1px est assuré d'avoir un rendu net.

Mais pour les tailles de police, il est préférable d'utiliser **em**. L'idée est (1) de ne pas spécifier la taille de la police de l'élément BODY (en HTML), mais d'utiliser la taille par défaut du support, parce que c'est une taille que l'utilisateur peut lire confortablement; (2) exprimer les tailles de police des autres éléments en **em**: 'H1 {font-size: 2.5em}' pour rendre le H1 2½ fois plus grand que le contenu normal du corps du document.

Le seul endroit ou vous pourriez utiliser **pt** (ou **cm** ou **in**) pour définir une taille de police est dans une feuille de style pour l'impression, si vous voulez être sûr que la police imprimée aura exactement une certaine taille. Mais même dans ce cas, utiliser la taille par défaut de la police est généralement plus indiqué.

✎ L'unité **px** vous évite donc à devoir connaître la résolution du support. Que le sortie soit en 96 dpi, 100 dpi, 220 dpi ou 1800 dpi, une dimension exprimée avec un nombre entier de **px** sera toujours rendue correctement et de façon très similaire sur tous les supports. Mais qu'en est-il si vous voulez vraiment connaître la résolution du support, par exemple, savoir si on peut utiliser une ligne en **0.5px** ? La réponse consiste à vérifier la résolution avec Media Queries. Expliquer Media Queries sort du cadre de cet article, néanmoins voici un petit exemple:

```
div.mybox { border: 2px solid }
@media (min-resolution: 2dppx) {
  /* Media with 2 or more dots per px */
  div.mybox { border: 1.5px solid }
}
```

NOUVELLES D'UNITÉS EN CSS

Pour rendre encore plus facile le fait d'écrire des règles de style qui ne dépendent que de la taille de la police par défaut, CSS a introduit une nouvelle unité depuis 2013: Le **rem**. Le **rem** (pour "root em") est la taille de l'élément racine du document dans la police de caractères. Contrairement au **em**, qui peut être différent pour chaque élément, le **rem** est constant tout au long du document. Par exemple, pour définir la même marge gauche pour les éléments P et H1, comparez cette feuille de style antérieure à 2013:

```
p { margin-left: 1em }
h1 { font-size: 3em; margin-left: 0.333em }
```

et la nouvelle version:

```
p { margin-left: 1rem }
h1 { font-size: 3em; margin-left: 1rem }
```

D'autres nouvelles unités rendent possible le fait de spécifier des tailles relatives à la fenêtre du lecteur, ce sont les unités **vw** and **vh**. Le **vw** équivaut à 1/100e de la largeur de la fenêtre et le **vh** équivaut à 1/100e de sa hauteur. Il y a également **vmin**, qui sélectionne la plus petite valeur entre **vw** et **vh**. Et **vmax**. (vous devinerez ce qu'elle signifie.)

Comme elles sont très récentes, elles ne fonctionnent pas encore partout. Mais en ce début 2015, plusieurs navigateurs les supportent déjà.

Unités CSS: em, px, pt, cm, in…

Bert Bos, *coordinateur de l'activité style*

Created 12 Jan 2010;
Last updated mer. 21 sept. 2016 05:20:30 UTC

LANGUES

- Deutsch
- English
- Français
- Polski
- Português
- Русский
- Українська

À propos des traductions