



Cours - les unités CSS

Note importante : Ce document est conçu comme un guide de référence, mais n'est pas exhaustif. La capacité à rechercher des informations complémentaires et à résoudre des problèmes de manière autonome est une compétence essentielle pour tout développeur. Si vous rencontrez des difficultés ou si certaines étapes ne fonctionnent pas comme prévu, n'hésitez pas à consulter d'autres ressources (documentation officielle, tutoriels YouTube, forums spécialisés comme Stack Overflow). Apprendre à trouver l'information manquante et à déboguer par vous-même est probablement la compétence la plus précieuse que vous développerez

Introduction

Le choix des bonnes unités en CSS est essentiel pour créer des designs responsives, accessibles et cohérents. Chaque unité a ses avantages, inconvénients et cas d'usage spécifiques.

W3Schools.com

W3Schools offers free online tutorials, references and exercises in all the major languages of the web. Covering popular subjects like HTML, CSS, JavaScript, Python, SQL, https://www.w3schools.com/cssref/css_units.php



Les types d'unités CSS

1. Unités absolues

Ces unités ont une taille fixe qui ne change pas selon le contexte.

Unité	Description	Cas d'usage idéal
px (pixel)	Unité la plus basique, représente un point sur l'écran	Bordures, ombres, éléments qui doivent rester de taille constante
pt (point)	Utilisé principalement pour l'impression (1pt = 1/72 pouce)	Documents destinés à l'impression
cm , mm , in	Centimètres, millimètres, pouces	Rarement utilisés en web, sauf pour l'impression

Quand les utiliser :

- Quand la taille doit rester absolument identique quelle que soit la taille d'écran
- Pour les éléments fins comme les bordures (1px, 2px)
- Pour les tailles minimales de sécurité (min-width: 320px)

2. Unités relatives à la police

Ces unités changent en fonction de la taille de la police.

Unité	Description	Cas d'usage idéal
em	Relative à la taille de police de l'élément parent	Espacements intérieurs et marges liés au texte
rem	Relative à la taille de police de l'élément racine (html)	Structure générale, composants réutilisables
ex	Hauteur de la lettre "x"	Rare, utilisé pour des alignements très précis
ch	Largeur du caractère "0"	Largeur des blocs de texte, comme des paragraphes

Quand les utiliser :

- **em** : Pour les éléments dont la taille doit être proportionnelle à la taille du texte de leur parent
- **rem** : Pour la cohérence globale du site, indépendamment des niveaux d'imbrication
- Pour créer des designs qui s'adaptent aux préférences d'accessibilité des utilisateurs

3. Unités relatives au viewport

Ces unités sont relatives à la taille de la fenêtre du navigateur.

Unité	Description	Cas d'usage idéal
<code>vw</code>	1% de la largeur du viewport	Éléments qui s'adaptent à la largeur de l'écran
<code>vh</code>	1% de la hauteur du viewport	Sections pleine hauteur, héros
<code>vmin</code>	1% de la plus petite dimension (largeur ou hauteur)	Responsive entre orientations portrait/paysage
<code>vmax</code>	1% de la plus grande dimension (largeur ou hauteur)	Effets spéciaux, cas particuliers

Quand les utiliser :

- Pour les sections qui doivent occuper un pourcentage spécifique de l'écran
- Pour les mises en page immersives qui remplissent l'écran
- Pour les polices qui doivent s'adapter à la taille de l'écran

4. Unités relatives au parent

Ces unités sont relatives à l'élément parent.

Unité	Description	Cas d'usage idéal
<code>%</code>	Pourcentage relatif à l'élément parent	Largeurs et hauteurs relatives
<code>fr</code>	Fraction de l'espace disponible (uniquement en Grid)	Mise en page avec CSS Grid

Quand les utiliser :

- Pour les largeurs qui doivent être proportionnelles à leur conteneur
- Pour répartir l'espace disponible (avec `fr` dans Grid)
- Pour créer des mises en page fluides

Guide de sélection des unités

Pour la typographie

- **Taille de base** : `rem` pour maintenir la cohérence globale
- **Ajustements locaux** : `em` pour les variations relatives au contexte
- **Taille responsive** : combinaison de `rem` et `vw` (`calc(1rem + 1vw)`)

Pour les layouts

- **Structure principale** : `%`, `vw`, `vh`
- **Grilles** : `fr`, `%`
- **Marges et espacements** : `rem` pour cohérence globale
- **Paddings liés au texte** : `em`

Pour les éléments UI

- **Bordures et ombres** : `px`
- **Boutons et composants** : `rem` pour taille, `em` pour padding interne
- **Radius et détails** : `px` ou `em` selon le contexte

Erreurs courantes et bonnes pratiques

1. Les marges en pourcentage (%) - généralement à éviter

Problème : Les marges en pourcentage (`margin: 3%`) sont **généralement à éviter** car :

- Les marges verticales en % (top/bottom) sont calculées par rapport à la **largeur** du parent, pas la hauteur
- Elles créent des espacements incohérents qui varient selon la taille de l'écran
- Elles peuvent poser des problèmes d'accessibilité lors du zoom
- Elles rendent le débogage plus difficile car leurs valeurs réelles changent constamment

Alternatives recommandées:

- Utilisez `rem` pour les espacements entre composants (cohérence globale)
- Utilisez `em` pour les espacements liés au texte
- Utilisez `px` pour les espacements minimaux fixes

Exemples:

```
/* À éviter */
.card {
  margin: 3%; /* Varie selon la largeur et crée des incohérences */
}
```

```

}

/* Préférer */
.card {
  margin: 1.5rem; /* Cohérent avec le système de design */
}

```

Cas limités où les marges en % peuvent être appropriées:

- Mise en page nécessitant des proportions très spécifiques
- Certains cas de marges horizontales uniquement (`margin-left/right`)
- Dans des formules `calc()` pour des calculs avancés

2. Centrage avec marges fixes - à éviter

Problème : Utiliser des valeurs spécifiques comme `margin-left: 25em` pour centrer des éléments est:

- Fragile (change si le contenu ou le conteneur change)
- Non responsive
- Difficile à maintenir

Meilleures pratiques pour centrer:

- Pour centrer horizontalement un bloc:

```

margin-left: auto;
margin-right: auto;
margin: 0 auto; /*Equivalent*/

```

- Pour centrer tout le contenu:

```

display: flex;
justify-content: center;
align-items: center;

```

- Pour centrer avec Grid:

```

display: grid;

```

```
place-items: center;
```

Bonnes pratiques générales

1. Utilisez **rem** comme unité de base pour la structure et la typographie
2. Préférez les unités relatives (**%** , **em** , **rem** , **vh** , **vw**) pour la plupart des cas
3. Réservez **px** pour les détails fins qui ne doivent pas changer
4. Utilisez **fr** dans les grilles pour une répartition flexible
5. Combinez les unités avec **calc()** pour des calculs avancés

Pièges à éviter

1. L'effet de cascade avec **em** : Les **em** s'accumulent dans les éléments imbriqués
2. Dépendance excessive aux **px** : Rend le design moins accessible et moins responsive
3. Mauvais mélange d'unités : Peut créer des incohérences dans le design
4. Oublier le zoom : Testez votre site avec différents niveaux de zoom pour l'accessibilité

Tableau complet des unités recommandées par contexte

Élément	Unité recommandée	Alternative	À éviter	Pourquoi
Taille de police de base	rem ou px	/	em pour la base	Base cohérente pour tout le site
Tailles de police secondaires	rem ou em	%	px	S'adapte aux préférences d'accessibilité
Largeur de conteneur	% , rem	vw	px fixe	Responsive et adaptatif
Hauteur de section	vh , rem	%	px fixe	S'adapte à la taille d'écran

Bordures	px	rem (fins)	em	Précision visuelle nécessaire
Marges entre composants	rem	%	px (sauf petites)	Cohérence du rythme vertical
Padding interne	em ou rem	%	px (sauf petites)	S'adapte au contenu
Espacement texte-icône	em	rem	px	Proportionnel au texte associé
Grilles	fr, %	rem	px	Distribution flexible de l'espace
Media queries	em, rem	px	%	Cohérence avec le système d'espacement
Espace entre sections	rem	vh	%	Rythme vertical cohérent
Indentation	em	rem	px	Proportionnelle au texte
Largeur de bouton	auto, em	rem	px fixe	S'adapte au contenu
Espacement interne bouton	em	rem	%	Proportionnel au texte du bouton
Largeur colonne grille	fr	%	px	Répartition égale ou proportionnelle
Hauteur d'en-tête	vh, rem	%	px	S'adapte à la taille d'écran

Protection intellectuelle : Ce cours, est une création originale de **MANJAL Younes** et est protégé par le droit d'auteur conformément au Code de la propriété intellectuelle français (article L.111-1). Toute reproduction, modification, diffusion ou utilisation, même partielle, sans autorisation

écrite préalable est strictement interdite. L'intelligence artificielle a été utilisée uniquement comme outil d'assistance pour la correction orthographique et la reformulation de certaines phrases, sous ma supervision et validation. Le contenu pédagogique et intellectuel reste entièrement ma création personnelle.