



CSS

FATIMATA THIAW

LES UNITÉS DE MESURES EN CSS

En CSS, les unités de mesure sont utilisées pour définir la taille, la distance et d'autres propriétés des éléments HTML. . Voici quelques-unes des unités de mesure les plus couramment utilisées :

- 01 Pixel (px)** : C'est une unité de mesure fixe qui correspond à un pixel sur l'écran.
Par exemple, **width: 200px;** définira la largeur d'un élément à 200 pixels.
- 02 Pourcentage (%)** : Cette unité de mesure est basée sur la taille de l'élément parent.
Par exemple, **width: 50%;** définira la largeur d'un élément à 50% de la largeur de son conteneur parent.
- 03 Em (em)** : Cette unité de mesure est relative à la taille de la police de l'élément parent.
Par exemple, si la taille de la police de l'élément parent est de 16 pixels, **font-size: 2em;** définira la taille de la police de l'élément à 32 pixels.
- 04 Rem (rem)** : Cette unité de mesure est similaire à l'em, mais elle est basée sur la taille de la police de l'élément racine (généralement la balise <html>). Cela le rend plus prévisible que l'em dans certains cas.

LES POSITIONS EN CSS

En CSS, il existe plusieurs valeurs pour la propriété position :

01

Static : C'est la valeur par défaut. Les éléments avec une position statique sont positionnés selon le flux normal du document.

02

Relative : Les éléments avec une position relative sont positionnés par rapport à leur position normale dans le flux du document. Cela signifie que vous pouvez utiliser les propriétés **top**, **right**, **bottom**, et **left** pour les déplacer par rapport à leur position d'origine.

03

Absolute : Les éléments avec une position absolue sont retirés du flux normal du document et sont positionnés par rapport à leur premier ancêtre positionné ou à l'élément conteneur. Si aucun ancêtre positionné n'existe, ils sont positionnés par rapport au coin supérieur gauche de la fenêtre du navigateur.

04

Fixed : Les éléments avec une position fixe sont retirés du flux normal du document et sont positionnés par rapport à la fenêtre du navigateur. Ils restent fixes même lorsque la page est défilée.

LES FLEXBOX EN CSS

01 Principe de base :

Flexbox est un modèle de disposition unidimensionnel, ce qui signifie qu'il agit le long d'un seul axe (généralement horizontal ou vertical).

Il est conçu pour organiser les éléments dans une ligne ou une colonne de manière flexible, en répartissant l'espace disponible de manière équitable entre les éléments ou en les ajustant pour s'adapter à l'espace disponible.

02 Propriétés principales :

- **display: flex;** : Cette propriété est appliquée à un conteneur pour activer le modèle Flexbox.
- **flex-direction** : Permet de spécifier la direction principale dans laquelle les éléments flexibles sont disposés (row, row-reverse, column, column-reverse).
- **justify-content** : Détermine la façon dont les éléments flexibles sont répartis le long de l'axe principal du conteneur.
- **align-items** : Contrôle l'alignement des éléments flexibles sur l'axe transversal du conteneur.
- **flex-grow, flex-shrink, flex-basis** : Propriétés permettant de contrôler la façon dont les éléments flexibles grandissent, rétrécissent et sont dimensionnés de base.

03 Utilisations :

- Mise en page d'une barre de navigation horizontale ou verticale.
- Création de mises en page responsives avec des éléments qui s'ajustent automatiquement en taille et en position en fonction de l'espace disponible.
- Alignement vertical et horizontal d'éléments à l'intérieur d'un conteneur.

LES GRID EN CSS

01 Principe de base :

- Grid est un modèle de disposition bidimensionnel qui permet de créer des mises en page complexes en définissant des grilles de lignes et de colonnes.
- Il permet de positionner les éléments dans ces grilles avec précision, en spécifiant leurs emplacements en termes de lignes et de colonnes.

02 Propriétés principales :

- **display: grid;** : Cette propriété est appliquée à un conteneur pour activer le modèle Grid.
- **grid-template-rows, grid-template-columns** : Permettent de définir la taille et le nombre de lignes et de colonnes dans la grille.
- **grid-gap** : Définit l'espace entre les lignes et les colonnes de la grille.
- **grid-row, grid-column** : Détermine la position des éléments à l'intérieur de la grille en spécifiant les lignes et les colonnes qu'ils occupent.
- **justify-items, align-items** : Contrôlent l'alignement des éléments à l'intérieur de leurs cellules de la grille.

03 Utilisations :

- Création de mises en page complexes avec des zones de contenu positionnées précisément.
- Réalisation de mises en page réactives qui s'adaptent automatiquement à différentes tailles d'écran en redimensionnant et en réorganisant les éléments.
- Création de grilles flexibles et adaptatives pour les formulaires ou les tableaux de données.

LES TRANSITIONS EN CSS

01 Définition :

Pour définir une transition sur un élément, utilisez la propriété `transition`. Vous spécifiez les propriétés CSS que vous souhaitez animer, la durée de l'animation et éventuellement d'autres paramètres comme la fonction de temporisation (timing function) et le délai avant le début de la transition.

Exemple :

```
.ma_classe {  
  transition: width 0.3s ease-in-out;  
}
```

02 Propriétés à animer :

Vous pouvez animer la plupart des propriétés CSS, telles que `width`, `height`, `color`, `background-color`, `opacity`, etc.

03 Durée de l'animation :

La durée de l'animation est spécifiée en secondes (s) ou en millisecondes (ms).

04 Fonction de temporisation (timing function) :

Elle définit l'accélération de l'animation. Les valeurs courantes sont `ease` (accélération/décélération), `linear` (vitesse constante), `ease-in` (départ lent, accélération rapide), `ease-out` (départ rapide, décélération lente), `ease-in-out` (départ lent, accélération puis décélération), et `cubic-bezier()` (fonction personnalisée).

LES TRANSITIONS EN CSS

Exemple :

```
.ma_classe {  
  transition: width 0.3s cubic-bezier(0.25, 0.1, 0.25, 1);  
}
```

05 Déclenchement de la transition :

Les transitions peuvent être déclenchées par des interactions utilisateur telles que :hover, :focus, ou en changeant dynamiquement les propriétés CSS via JavaScript.

LES TRANSITIONS EN CSS

Voici un exemple simple de transition en CSS :

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Transition en CSS</title>
<style>
    .ma_classe {
        width: 100px;
        height: 100px;
        background-color: blue;
        transition: width 0.3s ease-in-out;
    }
    .ma_classe:hover {
        width: 200px;
    }
</style>
</head>
<body>

<div class="ma_classe"></div>

</body>
</html>
```