Les variables CSS

1. Introduction

Les variables CSS, aussi appelées **Custom Properties**, permettent de stocker des valeurs réutilisables dans votre code. Elles offrent une solution puissante pour centraliser et uniformiser les styles, facilitant ainsi la maintenance et les mises à jour.

1.1. Pourquoi utiliser les variables CSS ?

- **Facilité de maintenance :** Modifiez une seule valeur pour affecter plusieurs endroits dans une feuille de styles.
- **Réutilisabilité :** Évitez les répétitions dans votre code.
- **Flexibilité :** Ajustez les valeurs dynamiquement via JavaScript si nécessaire.

2. Déclaration et utilisation des variables CSS

2.1. Déclarer une variable CSS

Les variables CSS sont définies à l'aide du préfixe `--`, suivi d'un nom. Elles sont habituellement déclarées dans un sélecteur `:root` pour une portée globale.

Exemple de déclaration globale :

```css
:root {
  --main-color: #3498db;
  --secondary-color: #2ecc71;
  --font-size-default: 16px;
}
```

Vous pouvez également déclarer des variables à portée locale, c'est-à-dire au sein d'un sélecteur spécifique.

Exemple de déclaration locale :

```css
.box {
  --border-color: #e74c3c;
}
```

2.2. Utiliser une variable CSS

Utilisez la fonction `var()` pour accéder à une variable :

Exemple :

```css
h1 {
  color: var(--main-color);
  font-size: var(--font-size-default);
}

.btn {
  background-color: var(--secondary-color);
  border: 2px solid var(--border-color, black); /* Valeur par défaut */
}
```

2.3. Valeurs par défaut avec `var()`

Vous pouvez fournir une **valeur de secours** dans `var()` au cas où la variable n'est pas définie ou n'est pas accessible.

Syntaxe :

```css
color: var(--non-existent, #000); /* Utilisera #000 comme valeur par défaut */
```

3. Portée des variables CSS

1. **Portée globale :** Les variables définies dans `:root` sont accessibles partout dans le document.

```css
:root {
  --global-color: #f39c12;
}

p {
  color: var(--global-color);
}
```

2. **Portée locale :** Les variables définies dans un élément ou une classe ne s'appliquent qu'à ce contexte spécifique et ses descendants.

```css
.card {
  --card-bg-color: #ecf0f1;
}
```

```css
.card-header {
  background-color: var(--card-bg-color);
}
```

4. Exemple pratique

4.1. HTML

```html
<div class="container">
 <h1>Bienvenue</h1>
 <p>Ceci est un exemple utilisant des variables en CSS.</p>
 <button class="btn">Cliquez ici</button>
</div>
```

4.2. CSS

```css
:root {
 --main-bg-color: #f0f8ff;
 --text-color: #333;
 --btn-color: #2ecc71;
 --btn-color-hover: #27ae60;
 --padding-default: 10px;
}

.container {
 background-color: var(--main-bg-color);
 color: var(--text-color);
 padding: var(--padding-default);
}

.btn {
 background-color: var(--btn-color);
 color: white;
 padding: var(--padding-default);
 border: none;
 border-radius: 5px;
}

.btn:hover {
 background-color: var(--btn-color-hover);
}
```

5. À RETENIR

Concept	Description
Déclaration	Définir une variable avec –nom-variable dans un sélecteur.
Utilisation	Récupérer la valeur avec var(–nom-variable).
Portée globale	Déclaration dans :root, accessible partout.
Portée locale	Déclaration dans un sélecteur (cible uniquement cet élément et ses descendants).
Valeur par défaut	Fournir une valeur alternative dans var(…) en cas d'absence de la variable.
Avantages	Centralise les styles, facilite les modifications, et améliore la lisibilité.