

Mémento CSS

Première et terminale NSI

① PRINCIPE ET SYNTAXE

```
/* sélecteur { propriété: valeur; } */
h1 {
  color: #0b598a;
  font-size: 2em;
}
p, li { /* plusieurs cibles */
  line-height: 1.5;
}
```

Trois façons de relier le CSS à la page, de la meilleure à la moins bonne :

```
<link rel="stylesheet" href="style.css">
<style> p { color: gray; } </style>
<p style="color: gray;">En ligne</p>
```

② SÉLECTEURS

Sélecteur	Cible
p	tous les éléments <p>
.note	éléments de classe note
#menu	élément d'identifiant menu
*	tous les éléments
nav a	les <a> descendants d'un <nav>
ul > li	les enfants directs
h2 + p	le <p> qui suit immédiatement un <h2>
p.note	les <p> de classe note
a[target]	les <a> ayant l'attribut
a:hover	lien survolé
li:first-child	premier de sa liste
tr:nth-child(2n)	lignes paires
input:focus	champ actif
p::first-letter	première lettre
h2::before	contenu inséré avant

Spécificité. Quand plusieurs règles s'appliquent, la plus précise l'emporte : style en ligne, puis #id, puis .classe et pseudo-classes, puis élément. À précision égale, la dernière règle écrite gagne. !important force une règle; on l'évite. Certaines propriétés (couleur, police) s'héritent du parent, d'autres non (marges, bordures).

③ COULEURS ET UNITÉS

```
color: red; /* nom */
color: #ff0000; /* hexadécimal */

color: #f00; /* forme courte */
color: rgb(255, 0, 0);
color: rgba(255, 0, 0, 0.5); /* transparence */
color: hsl(0, 100%, 50%);
```

Unité	Sens
px	pixel, taille fixe
em	relatif à la taille de police de l'élément
rem	relatif à la taille de police de la racine
%	relatif au parent
vw, vh	pourcentage de la largeur, de la hauteur de la fenêtre

④ TEXTE ET POLICE

```
body {
  font-family: "Source Sans 3", sans-serif;
  font-size: 16px;
  color: #222;
}
h1 { font-weight: bold; text-align: center; }
em { font-style: italic; }
a { text-decoration: none; }
p { line-height: 1.6; text-indent: 1em; }
.majuscules { text-transform: uppercase; }
```

⑤ MODÈLE DE BOÎTE

Chaque élément est une boîte : le contenu, puis padding (marge intérieure), puis border (bordure), puis margin (marge extérieure).

```
.carte {
  width: 300px;
  padding: 10px 20px; /* vertical, horizontal */
  border: 2px solid #0b598a;
  border-radius: 8px;
  margin: 1em auto; /* auto : centre */
  box-sizing: border-box; /* width tout compris */
  background-color: #f5f8f8;
  box-shadow: 0 2px 6px rgba(0, 0, 0, 0.2);
}
```

Raccourcis : margin: 10px (quatre côtés), margin: 10px 5px (haut-bas puis gauche-droite), margin: 1px 2px 3px

4px (haut, droite, bas, gauche). `margin-top`, `padding-left`, `border-bottom` ciblent un côté.

⑥ AFFICHAGE ET POSITIONNEMENT

display	Effet
block	occupe toute la largeur, retour à la ligne (<code>div</code> , <code>p</code> , <code>h1</code>)
inline	dans le flux du texte, sans dimensions (<code>span</code> , <code>a</code>)
inline-block	en ligne mais avec largeur et hauteur
none	élément retiré de l'affichage
flex	conteneur flexible (voir ci-dessous)
grid	conteneur en grille

```
.bandeau { position: fixed; top: 0; left: 0; }
.bulle { position: absolute; top: 10px; }
.parent { position: relative; } /* référence */
.dessus { z-index: 10; }
.defile { overflow: auto; height: 200px; }
```

`position: absolute` se place par rapport au premier ancêtre positionné (`relative`, `absolute` ou `fixed`); `fixed` reste visible au défilement.

⑦ FLEXBOX

```
.ligne {
  display: flex;
  flex-direction: row; /* ou column */
  justify-content: space-between;
  align-items: center;
  flex-wrap: wrap;
  gap: 1rem;
}
.ligne > .large { flex: 2; } /* 2 parts */
```

`justify-content` répartit sur l'axe principal (`flex-start`, `center`, `space-between`, `space-around`), `align-items` sur l'axe secondaire (`stretch`, `center`, `flex-start`).

```
.grille {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  gap: 12px;
}
```

⑧ ADAPTATION À L'ÉCRAN

```
img { max-width: 100%; height: auto; }
@media (max-width: 600px) {
  .ligne { flex-direction: column; }
```

```
  nav { display: none; }
}
```

Nécessite dans le `<head>` : `<meta name="viewport" content="width=device-width, initial-scale=1">`.

⑨ VARIABLES CSS

```
:root { --accent: #0b598a; --marge: 12px; }
h1 { color: var(--accent); }
.carte { padding: var(--marge); }
```

Une valeur définie une seule fois, réutilisée partout : changer la couleur du site revient à changer une ligne.

⑩ INTERACTIONS ET TRANSITIONS

```
button {
  cursor: pointer;
  transition: background-color 0.3s;
}
button:hover { background-color: #0b598a; }
.cache { opacity: 0; }
```

⑪ EXEMPLE COMPLET

```
<nav class="menu">
  <a href="#" class="actif">Accueil</a>
  <a href="#">Cours</a>
</nav>
```

```
.menu {
  display: flex;
  gap: 8px;
  padding: 8px;
  background-color: #1e293b;
}
.menu a {
  color: white;
  padding: 6px 12px;
  border-radius: 4px;
  text-decoration: none;
}
.menu a:hover, .menu a.actif {
  background-color: #0b598a;
}
```

12 BONNES PRATIQUES

- Un fichier `style.css` séparé, des classes aux noms parlants (`.carte`, `.menu`), jamais de style en ligne.
- Une règle par ligne, une indentation par bloc, un point-virgule après chaque déclaration.
- Vérifier le rendu dans l'inspecteur du navigateur (F12), onglet Éléments : les règles appliquées et barrées s'y lisent.
- Référence complète : developer.mozilla.org/fr/docs/W