

Lier le fichier css avec le fichier HTML avec **<link>**.

```
<!DOCTYPE html>

<html>

<head>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <p>Texte en bleu</p>

</body>

</html>
```

3. Syntaxe de base d'une règle CSS

CSS est un langage basé sur des règles.

```
sélecteur {
    propriété: valeur ;
}
```

4. Les sélecteurs CSS

Sélecteur	Description	Exemple
*	Cible tous les éléments	* { ... }
balise	Cible tous les éléments de type élément.	p { ... }
#id	Cible un élément avec un attribut id spécifique.	#T1 { ... }
.class	Cible tous les éléments avec une classe spécifique.	.btn { ... }
élément[attribut]	Cible les éléments ayant un cet attribut .	a[target] { ... }
élément[attr="value"]	Cible les éléments ayant un attribut attr égale à une valeur exacte.	input[type="text"] { ... }
élément .class	Cible les éléments de type élément avec une classe spécifique.	p.warning { ... }
élément :pseudo-classe	Cible les éléments dans un état particulier d'un élément.	a:link {....} a:visited {.....} a:hover {.....} a:active {.....}
selector1, selector2	Applique la même règle à plusieurs sélecteurs (séparés par des virgules).	h1, h2 { ... }

5. Les propriétés de mise en forme du texte

Propriété	description	Exemple
font-family	Définit la police de caractères	font-family: Arial ;
font-weight	Contrôle l'épaisseur du texte (bold, normal, lighter)	font-weight : bold ;
font-style	Définit le style	font-style : italic ;
font-size	Définit la taille de la police.	font-size : 16px ;
font	<p>Super-propriété combinant font-family, weight, style, size.</p> <p>font: [style] [weight] size family;</p> <p>➤ Les parties optionnelles : style et weight</p> <p>➤ Les parties obligatoires : size et family</p> <p>Rq :</p> <ul style="list-style-type: none"> • Sans size et family, la propriété ne fonctionne pas ! • L'ordre est obligatoire si on ajoute style et weight. 	font: italic bold 16px Arial;
text-align	Alignement horizontal du texte (left, center, right, justify)	text-align :center ;
text-shadow	<p>Ajoute une ombre au texte.</p> <p>text-shadow: décalage-x décalage-y flou couleur;</p> <p>Rq :</p> <ul style="list-style-type: none"> • Ordre obligatoire : x → y → flou → couleur • Flou est optionnel 	text-shadow: 2px 2px 4px red;
text-transform	Transforme la casse du texte (uppercase, lowercase, capitalize)	text-transform : uppercase ;
color	Définit la couleur du texte.	Color :#FF0000 ;

6. Propriétés d'arrière-plan

Propriété	Description	Exemple
background-color	Définit la couleur de fond.	background-color : #FF0000 ;
background-image	Ajoute une image de fond.	background-image : url("fond.jpg") ;
background-repeat	Contrôle la répétition de l'image (no-repeat, repeat-x, repeat-y)	background-repeat : no-repeat ;
background-size	Ajuste la taille de l'image. Valeurs principales : <ul style="list-style-type: none"> ➤ cover → l'image couvre tout le conteneur (elle peut être coupée) ➤ contain → l'image est entièrement visible (peut laisser des espaces) ➤ largeur hauteur (ex. 100px 50px) → taille personnalisée ➤ pourcentage (ex. 100% 50%) → en fonction du conteneur 	background-size : 50% 100% ;
background	Combine toutes les propriétés en une seule. Rq : <ul style="list-style-type: none"> • Ordre important : background: [background-image] [background-repeat] / [background-size]; • / est obligatoire uniquement si on définit background-size (Si on ne met pas le slash , alors les dernières valeurs sont interprétées comme background-position) 	background: url("fond.jpg") no-repeat / 50% 50% ;

7. Propriétés des bordures

Propriété	Description	Exemple
border-color	Couleur de la bordure.	border-color : red ;
border-style	Style de la bordure (solid, dashed, dotted, double)	border-style : dashed ;
border-width	Épaisseur de la bordure.	border-width :5px ;
border-radius	Arrondit les coins.	border-radius:10px ;
border	Combine toutes les propriétés en une seule Rq : <ul style="list-style-type: none"> • Ordre important : border: <border-width> <border-style> <border-color>; • Le <u>style</u> est obligatoire . 	border: 2px solid blue;

8. Propriétés des listes

propriété	Description	Exemple
list-style-type	Définit le type de marqueur (circle, square, upper-roman, lower-alpha)	list-style-type : circle ;
list-style-position	Position du marqueur par rapport au texte : outside (par défaut) Ou inside	list-style-position : inside ;
list-style-image	Remplace le marqueur par une image.	list-style-image :url("puce.png") ;
list-style	Combine les 3 propriétés ci-dessus en une seule. Rq : <ul style="list-style-type: none">•Ordre important : list-style: [type] [position] [image];• Les valeurs sont optionnelles, mais l'ordre doit être correct si elles sont présentes• Si list-style-image est défini avec une image valide c'est cette image qui sera utilisée comme puce.	list-style: square inside url("puce.png");

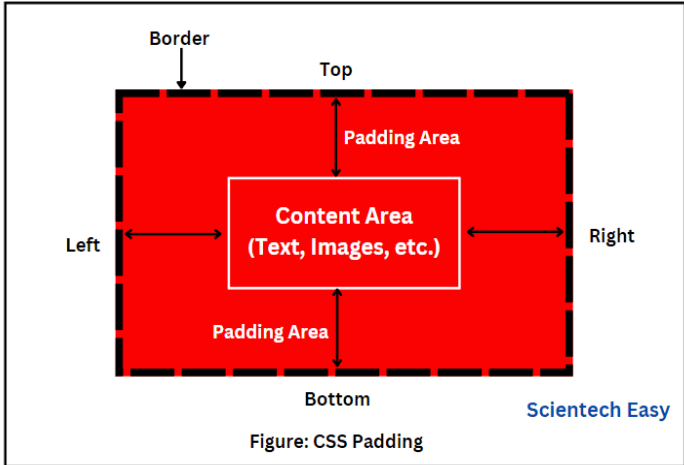
9. Propriétés des tableaux


Propriété	Description	Exemple
table-layout	Contrôle l'algorithme de disposition du tableau ✓ table-layout: auto; (valeur par défaut) <ul style="list-style-type: none">• Le navigateur calcule la largeur des colonnes en fonction du contenu.• Les colonnes s'adaptent à ce qu'il y a dedans.• Si une cellule contient un long mot, la colonne peut s'élargir. ✓ table-layout: fixed; <ul style="list-style-type: none">• Le navigateur n'attend pas de lire tout le contenu pour poser la largeur des colonnes.• Les largeurs sont déterminées à partir du premier row (ou bien des width données en CSS).• Le contenu peut être coupé ou aller à la ligne si ça ne rentre pas.	<pre>table { table-layout: fixed; } td { width: 25%; /* 4 colonnes égales */ }</pre>
border-collapse	Définit si les bordures des cellules sont fusionnées (collapse) ou séparées (separate).	border-collapse :collapse ;

10. Propriétés des images

Filtre	Description	Exemple
filter : blur()	Applique un flou gaussien à l'élément.	filter: blur(5px);
filter : grayscale()	Convertit en niveaux de gris <ul style="list-style-type: none"> ➤ grayscale(0) ou grayscale(0%) → pas de gris, image normale. ➤ grayscale(1) ou grayscale(100%) → complètement noir et blanc. ➤ Entre 0 et 1 (ou 0% et 100%) → effet partiel. 	filter: grayscale(1);
filter : invert()	Inverse les couleurs <ul style="list-style-type: none"> ➤ invert(0) → aucune inversion (image normale). ➤ invert(1) ou invert(100%) → inversion complète. ➤ Valeurs intermédiaires (0.2, 0.5, etc.) → mélange entre normal et inversé. 	filter: invert(1);

11. Propriétés des boîtes

Propriété	Description	Exemple
width et heigth	<ul style="list-style-type: none"> ➤ Largeur et hauteur de l'élément ➤ Unités : <ul style="list-style-type: none"> • px → pixels fixes • % → proportion par rapport au parent (ça ne marche que si le parent a lui-même une hauteur définie) ➤ Fonctionne seulement sur des éléments de type bloque 	width : 100px ; height : 200px ;
padding	<p>Espace intérieur (haut/bas gauche/droite) entre le contenu et la bordure.</p> 	padding: 20px; /* haut, droite, bas, gauche */ padding: 10px 30px; /* haut et bas, gauche et droite */ padding: 5px 15px 10px; /* haut , gauche et droite , bas */ padding: 5px 10px 15px 20px; /*haut, droite, bas, gauche (sens horaire)*/

margin	<p>Espace extérieur (haut/bas gauche/droite) entre la bordure et les autres éléments qui l'entourent.</p>  <p>Centrer un élément horizontalement : margin : auto ; 👉 L'élément se centre dans son parent .</p>	<p>margin: 20px; /* haut, droite, bas, gauche */</p> <p>margin: 10px 30px; /* haut et bas , gauche et droite */</p> <p>margin: 5px 15px 10px; /* haut, gauche et droite, bas */</p> <p>margin: 5px 10px 15px 20px; /* haut, droite, bas, gauche */</p>
box-shadow	<p>Ajouter une ombre portée à une boîte (comme un div, un bouton, une image, etc.). box-shadow: [inset] offset-x offset-y [blur] [spread] [color];</p> <ul style="list-style-type: none"> ▪ inset → optionnel, fait l'ombre à l'intérieur (sinon c'est une ombre externe par défaut). ▪ offset-x → obligatoire, déplacement horizontal. ▪ offset-y → obligatoire, déplacement vertical. ▪ blur → optionnel (flou). ▪ spread → optionnel (taille de l'ombre). ▪ color → optionnel (par défaut = couleur du texte). 	<p>box-shadow :2px 2px 5px grey ;</p>
overflow	<p>Contrôle le comportement du contenu qui dépasse la taille d'un élément</p> <ul style="list-style-type: none"> ▪ visible (par défaut) : Le contenu déborde sans restriction. ▪ hidden : Le contenu qui dépasse est coupé et invisible. ▪ Scroll : Des barres de défilement apparaissent toujours, même si ce n'est pas nécessaire. ▪ Auto : Des barres de défilement apparaissent uniquement si nécessaire. 	<p>overflow : auto ;</p>
opacity	<p>contrôle la transparence d'un élément (Valeur : de 0 à 1) 0 → complètement transparent (invisible) 1 → complètement opaque (normal) Exemple : 0.5 → 50% transparent</p>	<p>opacity : 0.5 ;</p>

12.transform

Fonction CSS	Description	Exemple
translate(X, Y)	Déplace un élément sur l'axe X et Y	transform: translate(50px, 20px);
translateX(X) / translateY(Y)	Déplacement sur un seul axe	transform: translateX(30px);
rotate(angle)	Fait tourner un élément autour de son centre	transform: rotate(45deg);
scale(X, Y)	Change la taille d'un élément (X et Y)	transform: scale(2, 1.5);
scaleX(X) / scaleY(Y)	Change la taille sur un seul axe	transform: scaleY(0.8);
skew(Xdeg, Ydeg)	Incline un élément sur X et Y	transform: skew(20deg, 10deg);
skewX(Xdeg) / skewY(Ydeg)	Incline sur un seul axe	transform: skewX(15deg);
translate(...) rotate(...) scale(...)	Combine plusieurs transformations en une seule	transform: translate(50px, 20px) rotate(30deg) scale(1.5);

13.Les pseudo-classe des liens

C'est un mot-clé ajouté aux sélecteurs qui permet de cibler un état particulier d'un élément

Pseudo-classe	Description	Exemple
:link	Sur un lien non encore visité (première fois qu'on le voit)	a:link { color: blue; }
:visited	Sur un lien déjà visité (le navigateur l'a enregistré dans l'historique)	a:visited { color: green; }
:active	Quand on clique sur le lien (pendant l'appui)	a:active { color: orange; }
:hover	Quand la souris passe au-dessus du lien	a:hover { color: red; }

Remarque :

- Certaines pseudo-classes comme **:link** et **:visited** sont réservées uniquement aux liens hypertexte
- D'autres, comme **:hover** et **:active**, s'appliquent à n'importe quel élément HTML (pas seulement aux liens).

14. Contrôle de l’affichage et du placement des éléments

Propriété CSS	Effet principal	Retiré du flux normal	Exemple
display : inline	L’élément reste sur la même ligne, largeur/hauteur non modifiable directement	Non	<code>span { display: inline; }</code>
display : block	L’élément occupe toute la largeur disponible, commence sur une nouvelle ligne	Non	<code>div { display: block; }</code>
display : inline-block	Sur la même ligne mais largeur/hauteur modifiables	Non	<code>div { display: inline-block; }</code>
float :left/right	L’élément flotte à gauche ou à droite du conteneur	Oui	<code>img { float: left; }</code>
position :static	Position par défaut dans le flux normal	Non	<code>div { position: static; }</code>
position :relative	Déplacement relatif à sa position normale	Non	<code>div { position: relative; top: 10px; }</code>
Position :absolute	Retiré du flux, positionné par rapport au parent positionné (c’est-à-dire qui a position: relative, absolute, ou fixed)	Oui	<code>div { position: absolute; top: 0; left: 0; }</code>
position :fixed	Fixé par rapport à la fenêtre, même lors du scroll	Oui	<code>div { position: fixed; bottom: 0; right: 0; }</code>
transform : translate(X,Y)	Déplace visuellement l’élément sans affecter le flux	Non	<code>div { transform: translate(50px, 20px); }</code>

NB :

Retiré du flux normal :

- L’élément n’occupe plus sa place habituelle dans le document.
- Les autres éléments qui suivent agissent comme s’il n’était pas là

15. Les animations

Pour créer une animation on peut utiliser soit la propriété **transition** ou la règle **keyframes**

	Transition	@keyframes
Déclenchement	Sur un événement (hover, focus, changement de classe)	Automatique dès qu'on applique l'animation
Étapes	De l'état initial à l'état final (2 états seulement)	Plusieurs étapes possibles (0%, 50%, 100%)
Répétition	Une seule fois (sauf si on redéclenche l'événement)	Peut-être infinie ou multiple (animation-iteration-count)
Complexité	Simple, idéal pour des changements simples (couleur, taille, opacité)	Complexe, permet des mouvements, rotations, boucles, changements multiples simultanés

a) Les transitions

Une **transition CSS** permet de **modifier progressivement les propriétés d'un élément** lorsqu'un événement se produit

Propriété	Description	Exemple
transition-property	La ou les propriétés CSS à animer : background-color, width, all, etc (Obligatoire)	transition-property: background-color;
transition-duration	Durée de la transition (Obligatoire)	transition-duration: 0.5s;
transition-delay	Délai avant le début de la transition (facultatif)	transition-delay: 0.2s;
transition	Raccourci pour définir toutes les propriétés ci-dessus en une seule ligne Toujours mettre l'ordre classique : propriété → durée → délai.	transition: all 0.5s 0s;

b) Les règles @keyframes

Propriété	Description	Exemple
@keyframes	Définit les étapes de l'animation.	@keyframes Nom_Animation { 0% {} . . 100% {} }
animation-name	Nom de l'animation (défini dans @keyframes).	animation-name :Nom_Animation ;
animation-duration	Durée totale de l'animation.	animation-duration : 2s ;
animation-delay	Délai avant le début de l'animation .	animation-delay : 0.5s ;
animation-iteration-count	Nombre de répétitions. (ex: 1, 10, infinie)	animation-iteration-count :infinite ;
animation-direction	Sens de l'animation. <ul style="list-style-type: none"> • normal (par défaut) :L'animation se joue de 0% à 100% à chaque boucle, toujours dans le même sens • reverse :L'animation se joue de 100% à 0% à chaque boucle, sens inverse à chaque boucle • alternate : Mouvement de va-et-vient : 1ère boucle de 0%→100%, 2ème boucle 100%→0%, etc. • alternate-reverse : Va-et-vient mais commence à l'envers 100%→0%, puis 0%→100%, etc. 	animation-direction : alternate ;
animation	Super-propriété combinant toutes les valeurs ci-dessus. Pour éviter toute confusion, mieux vaut respecter l'ordre standard ci-dessous : animation: animation-name animation-duration [animation-delay] [animation-iteration-count] [animation-direction] ;	animation: monAnimation 2s infinite;