

**Travaux pratiques**

# CSS 3

**Maîtrisez la conception  
de vos pages web**

**Fabrice Lemainque**

DUNOD

Toutes les marques citées dans cet ouvrage sont des marques déposées par leurs propriétaires respectifs

Mise en pages réalisée par Lumina Datamatics, Inc.

<p>Le pictogramme qui figure ci-contre mérite une explication. Son objet est d'alerter le lecteur sur la menace que représente pour l'avenir de l'écrit, particulièrement dans le domaine de l'édition technique et universitaire, le développement massif du photocopillage.</p> <p>Le Code de la propriété intellectuelle du 1<sup>er</sup> juillet 1992 interdit en effet expressément la photocopie à usage collectif sans autorisation des ayants droit. Or, cette pratique s'est généralisée dans les établissements</p>	 <p><b>DANGER</b> LE PHOTOCOPIAGE TUE LE LIVRE</p>	<p>d'enseignement supérieur, provoquant une baisse brutale des achats de livres et de revues, au point que la possibilité même pour les auteurs de créer des œuvres nouvelles et de les faire éditer correctement est aujourd'hui menacée.</p> <p>Nous rappelons donc que toute reproduction, partielle ou totale, de la présente publication est interdite sans autorisation de l'auteur, de son éditeur ou du Centre français d'exploitation du droit de copie (CFC, 20, rue des Grands-Augustins, 75006 Paris).</p>
--	--	--

© Dunod, 2018  
11 rue Paul Bert, 92240 Malakoff  
www.dunod.com  
ISBN 978-2-10-077104-2

Le Code de la propriété intellectuelle n'autorisant, aux termes de l'article L. 122-5, 2° et 3° a), d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite » (art. L. 122-4).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la propriété intellectuelle.

# Table des matières

---

Avant-propos .....	1
--------------------	---

## INTRODUCTION

Préfixes dédiés .....	3
Terminologie : les valeurs CSS .....	4
Présentation des exercices .....	6

## MISE EN ŒUVRE

 Mise en œuvre de CSS .....	8
 Rédaction d'une feuille de styles externe .....	10

## LES SÉLECTEURS

 Sélecteurs d'éléments .....	14
 Sélecteurs d'attributs .....	17
 Sélecteurs de pseudo-classes dynamiques .....	19
 Sélecteurs de pseudo-classes structurales .....	21
 Sélecteurs de pseudo-éléments .....	24
 Spécificité d'un sélecteur .....	27

## MODÈLE DE BOÎTES CSS

 Propriétés de marge .....	30
 Propriétés de bordure .....	32
 Propriétés avancées de bordure .....	34
 Propriétés d'espacement .....	37
 Propriétés de contenu .....	38
 Propriétés de positionnement .....	40
 Propriétés de couche .....	42
 Propriétés d'effet visuel .....	43

## PROPRIÉTÉS DE POLICE, DE COULEUR ET DE FOND

 Propriétés de police .....	46
 Propriétés de couleur et de fond .....	49
 Dégradés linéaires .....	53
 Dégradés radiaux .....	56
 Propriétés de texte .....	59



 TP 22	Multi-colonnage .....	62
 TP 23	Propriétés d'interface utilisateur .....	64

## LISTES ET COMPTEURS

 TP 24	Propriétés de liste .....	68
 TP 25	Propriétés de compteur et de numérotation .....	71

## CSS ET PÉRIPHÉRIQUE D'AFFICHAGE

 TP 26	Spécification de feuille de styles dépendant du média .....	74
 TP 27	Requêtes médias .....	76

## AVANCÉES CSS

 TP 28	Modèle de boîte flexible ( <i>flexbox</i> ) .....	78
 TP 29	Transformations 2D .....	82
 TP 30	Transformations 2D multiples .....	84
 TP 31	Transformations 3D .....	86
 TP 32	Transformations 3D multiples .....	88
 TP 33	Transitions .....	90
 TP 34	Animations .....	92
 TP 35	Modèle de quadrillage ( <i>grid</i> ) .....	95
 TP 36	Variables en cascade .....	101

## ANNEXES

Rappel sur les documents W3C .....	105
État d'avancement actuel CSS .....	106
Tableaux résumés .....	108
Index .....	121

LES + EN  
LIGNE



Vous trouverez sur la page de présentation de l'ouvrage du site Dunod ([www.dunod.com](http://www.dunod.com))  
**trois TP complémentaires, sur les fonctions `matrix()` et `matrix3d()`.**

# Avant-propos

---

Le langage CSS (*Cascading Style Sheet*, feuille de styles en cascade) permet la définition et la manipulation des styles appliqués à chaque élément d'un document rédigé dans un langage structurel de balisage (HTML, XHTML, XML, etc.) par la rédaction de feuilles de styles. Cela contribue à l'objectif essentiel de séparation de la structure et de la présentation et apporte de multiples avantages :

- possibilité de partager des feuilles de styles entre plusieurs documents et/ou sites ;
- indépendance entre présentation et document source structuré ;
- les navigateurs peuvent charger les feuilles de styles sélectivement, de façon à s'adapter au périphérique d'affichage (conception adaptative ou *responsive design*).

Comme généralement en informatique, c'est un langage en perpétuelle évolution, mais qui présente toutefois l'exceptionnelle caractéristique de toujours construire sur l'existant, en l'améliorant et en le complétant. En effet, CSS ne possède pas (ou plus exactement ne possède plus) de versions au sens traditionnel du terme mais propose à la place des niveaux attribués à des modules indépendants. Chaque niveau CSS complète le précédent en affinant ses définitions et en ajoutant des fonctionnalités. Tout nouveau niveau est un sur-ensemble des niveaux précédents, si bien que les fonctionnalités offertes par un niveau donné sont toujours un sous-ensemble de celles offertes par un niveau supérieur : un navigateur prenant en charge un niveau CSS élevé prendra toujours en charge tous les niveaux inférieurs.

Vous pourrez rencontrer des références à CSS 1, 2 ou 3, mais cela correspond désormais à la seule date d'approbation : l'unique recommandation officielle (à ce jour) est CSS 2.1, mise à jour lorsque nécessaire par ajout de modules développés indépendamment (comme par exemple Sélecteurs Niveau 3 ou Couleurs Niveau 3, intégrés à la recommandation). Chaque module évolue ensuite à son rythme. En pratique et pour résumer :

- tout module préexistant dans CSS 2.1 verra ses améliorations ultérieures être de niveau 3 et supérieur ;
- tout module absent de CSS 2.1 commencera au niveau 1 lors de son intégration.

C'est ainsi que le module Multi-colonnage, initialement présenté comme CSS 3 puisque mis au point après CSS 2.1, porte en fait le nom de module Multi-colonnage Niveau 1. Nous traiterons ici tant de la recommandation de base CSS 2.1 (constituée de CSS 1 complété de CSS 2 et de quelques modules récents, souvent de niveau 3) que des modules récents nouveaux considérés comme stables et atteignant les stades finaux (CR, candidat à la recommandation, ou PR, proposé pour la recommandation).

De nombreux outils permettant de réaliser des pages web ou d'en optimiser la conception exploitent le langage CSS : par exemple Bootstrap ou WordPress, mais il existe de nombreux autres systèmes de gestion de contenu et/ou bibliothèques concernés. Les utilisateurs de ces produits ne disposent pas forcément des connaissances leur permettant de personnaliser facilement à leur guise les modèles gouvernant ces pages et d'ainsi les optimiser. Examiner la spécification n'est pas toujours suffisant tout en restant souvent fastidieux.

Ce livre s'adresse à tout public, avec pour but de mieux le familiariser pas à pas avec ce langage pour qu'il puisse par la suite en tirer le meilleur parti. Il détaille la recommandation en de nombreux travaux pratiques individuels s'appuyant sur des pages web exemples soigneusement élaborées afin de permettre de tester et de maîtriser successivement chaque aspect du langage CSS. Apprendre en pratiquant, tel est le principe directeur. Nous n'avons pas la prétention d'être exhaustifs mais avons toutefois essayé de couvrir tous les aspects actuellement exploitables de ce langage. En principe, vous devriez réaliser l'ensemble des exercices en une dizaine d'heures, chaque TP individuel demandant de

10 minutes (le plus fréquemment) à 45 minutes (pour les deux TP les plus complexes). N'hésitez toutefois pas à passer plus de temps sur chaque TP, en testant d'autres valeurs afin de voir le résultat, et d'y revenir par la suite pour mener vos propres expériences. Seule la pratique vous assurera la maîtrise de ce langage.

Il sera supposé que vous possédiez une connaissance au moins de base du HTML et de ses principaux éléments : nous nous intéressons ici uniquement à la présentation, pas à la structure, même si nous devons évidemment nous appuyer sur celle-ci. Pour réaliser les exercices, vous devrez disposer d'un éditeur de texte. Si le bloc-notes de base (notepad.exe sous Windows) peut remplir cet office, je vous recommande toutefois l'éditeur gratuit Notepad++ disponible à l'adresse <https://notepad-plus-plus.org/fr/> pour de nombreux systèmes d'exploitation. Vous devrez également disposer d'un navigateur Internet, de préférence dans sa dernière version, mais il n'est pas nécessaire d'avoir de connexion Internet : tout se déroulera en local.

Certaines conventions classiques de la programmation sont employées dans ce livre, essentiellement lors de la présentation de la syntaxe des propriétés ou fonctions :

- des crochets ([ ]) signalent un paramètre facultatif. Par exemple :  
`animation: nom durée [type] [délai] [itération] [direction]`
- un signe de pipeline (*pipe*, |) sert à séparer différentes valeurs possibles mais mutuellement exclusives. Par exemple :  
`list-style-type: disc | circle | square`
- un paramètre figurant en italique signifie qu'il devra être remplacé par une valeur lors de l'emploi de la propriété. Soit le terme mentionné est explicite en lui-même (comme dans le cas des paramètres de la propriété `animation` ci-dessus), soit il est explicité dans le texte qui suit. Le type de valeur doit être une valeur CSS adaptée ;
- les instructions figurant en gras doivent être saisies par vous-même. Si elles suivent ou sont encadrées de code non en gras, cela signifie qu'il s'agit d'un ajout à une fraction de code existant.



**LES + EN LIGNE** Vous trouverez sur la page de présentation de l'ouvrage du site Dunod ([www.dunod.com](http://www.dunod.com)) les fichiers source des exercices employés dans ce livre. Vous devrez les télécharger et les installer sur votre ordinateur. Je vous recommande de le faire dans un dossier dédié dans Documents, sous le nom TP\_CSS.

Deux fichiers correspondent à chaque TP : le fichier TP\_CSS\_XX\_début.html (où XX est le numéro du TP) est le fichier initial à partir duquel travailler, et le fichier TP\_CSS\_XX\_fini.html est le fichier à l'issue du TP, pour vous permettre de contrôler éventuellement votre travail. Il y a quelques fichiers annexes, ainsi qu'un dossier renfermant des images employées par certains exercices : ne modifiez pas le nom de ce dossier ni l'emplacement des images.

Une dernière précision : CSS est un langage qui applique des propriétés de présentation à des éléments sélectionnés. Il peut donc être décomposé en deux parties : les sélecteurs et les propriétés. Il est indispensable de maîtriser au moins raisonnablement les sélecteurs avant de passer à l'étude des propriétés CSS. Il est relativement simple de retrouver la syntaxe de telle ou telle propriété, si vous vous souvenez de son existence, mais bien plus délicat de sélectionner le ou les éléments appropriés et eux seulement. Les TP 3 à 8 doivent donc impérativement être effectués et raisonnablement compris avant de pouvoir passer à la suite.

# Préfixes dédiés

La plus grande partie des derniers travaux CSS sont pris en charge par les navigateurs récents. La situation évolue en permanence, au rythme de la diffusion de nouvelles versions tant des modules CSS que des navigateurs.

Pour assurer une prise en charge des propriétés récentes avant leur intégration définitive, les navigateurs ont traditionnellement recours à des variantes dédiées accessibles à l'aide de préfixes qui leur sont propres, présentés dans le tableau suivant.

<b>Internet Explorer</b>	-ms-
<b>Firefox</b>	-moz-
<b>Chrome/Safari</b>	-webkit-
<b>Opera</b>	-o-

Ces préfixes permettent ainsi à des versions plus anciennes des navigateurs de pouvoir interpréter les propriétés CSS. Attention toutefois, la prise en charge de telle ou telle propriété par un navigateur, même en employant son préfixe, dépend de chaque propriété. Parfois, certaines valeurs uniquement ne sont pas prises en charge pour une propriété par un navigateur, même avec préfixe.

Par exemple, pour un multi-colonnage avec Firefox, il faudrait encore écrire :

```
{c-moz-column-count: 3 ;}
```

Pour garantir une fonctionnalité maximale, il conviendrait d'écrire dans le code, par sécurité :

```
{-moz-column-count: 3 ;
-ms-column-count: 3 ;
-webkit-column-count: 3 ;
-o-column-count: 3 ;
column-count: 3 ;}
```

Manifestement, une surcharge importante. Pour éviter de devoir saisir du code inutile, vous pouvez recourir à un script qui génère automatiquement ces préfixes pour les propriétés et fonctions concernées, en fonction du navigateur appelant la page. Un des plus célèbres, même s'il en existe bien d'autres, est sans doute *prefix-free* (<http://leaverou.github.io/prefixfree/>). C'est un petit fichier JavaScript (2 ko), développé par Lea Verou, dont le rôle est d'ajouter automatiquement des préfixes propriétaires.

*prefix-free* est très simple d'emploi. Ajoutez simplement la ligne

```
<script type="text/javascript"
  src="https://raw.githubusercontent.com/LeaVerou/prefixfree/master/prefixfree.min.js">
</script>
```

juste après la déclaration du CSS.

Vous pouvez également installer ce script à la racine de votre site et l'appeler depuis cet emplacement en insérant la ligne suivante juste après votre feuille de styles (ou l'appel à celle-ci) :

```
<script type="text/javascript" src="prefixfree.min.js"></script>
```

Reportez-vous si nécessaire au site officiel de *prefix-free* pour plus de détails.

Nous avons inclus ce script dans les fichiers proposés en téléchargement et l'employons systématiquement dans les fichiers d'exercice afin de garantir la meilleure compatibilité possible avec une vaste palette de navigateurs, même un peu anciens.

En cas de doute sur la prise en charge d'une propriété par tel ou tel navigateur, je vous invite à vérifier sur l'excellent site <http://caniuse.com/>.

# Terminologie : les valeurs CSS

Il est important de comprendre comment sont exprimées en CSS les valeurs et les unités. Cela est défini par le module Valeurs et unités Niveau 3 de la spécification CSS 2.1.

## Entiers et nombres

Certaines propriétés demandent comme valeur un entier ou un nombre. Celui-ci doit toujours être spécifié selon une notation décimale, avec zéro ou plus décimales, avec comme séparateur décimal un point. Par exemple :

```
left: 10.5cm
```

## Dimensions

Une dimension est un nombre immédiatement suivi d'un identifiant d'unité, facultatif si la valeur est 0. Il existe deux types d'unités de distance, relatives (la dimension est proportionnelle à la valeur d'une autre propriété de dimension) ou absolues.

Type d'unité	Identifiant	Signification
Unités relatives	em ex	Valeur de la propriété <code>font-size</code> de la police du parent Valeur de la propriété <code>x-height</code> de la police du parent
Unités absolues	in cm mm pt pc px	Pouce. 1in est égal à 2,54 cm. Centimètre Millimètre Point. Un point CSS est égal à 1/72 <sup>e</sup> de pouce. Pica. 1pc est égal à 12 pt. Unité pixel. 1px est égal à 0,75 pt.

Par exemple :

```
font-size: 1.5em
```

Un pourcentage s'exprime par un nombre immédiatement suivi du signe %. Il est toujours relatif à une autre valeur. Par exemple :

```
width: 50%
```

## Couleurs

Une couleur CSS peut être définie de diverses façons, selon le module Couleur Niveau 3 :

- Trois valeurs RGB, exprimées à l'aide de la fonction `rgb()` sous forme décimale, hexadécimale ou comme pourcentages.
- Quatre valeurs RGBA à l'aide de la fonction `rgba()`. Les valeurs hexadécimales ne sont pas autorisées en RGBA. Le quatrième chiffre, compris entre 0 et 1, détermine l'opacité de la couleur. Les deux déclarations ci-dessous définissent une opacité explicite de 1 (la valeur par défaut) :

```
color: rgba(255,0,0,1)
```

```
color: rgba(100%,0%,0%,1)
```

- Trois valeurs HSL (*hue-saturation-lightness*, teinte-saturation-luminosité) à l'aide de la fonction `hsl()`. La teinte est la couleur exprimée en degrés d'un cercle chromatique (0-360) et la saturation et la luminosité sont exprimées en pourcentages. Soit par exemple `color: hsl(0, 100%, 50%)` pour du rouge ou `color: hsl(120, 100%, 50%)` pour du vert sombre.
- Trois valeurs HSLA à l'aide de la fonction `hsla()`. C'est la contrepartie de la notation RGBA :  
`color: hsla(120, 100%, 50%, 1)`
- Des noms de couleur étendus, comme `blue`, `red`, `olive`, `lightgrey`, etc. La valeur `transparent` est équivalente à la définition d'un noir transparent `rgba(0,0,0,0)`.

## URI (*Uniform Resource Identifier*)

Ces valeurs sont notées `url()` (une survivance de l'époque où une URI se nommait une URL). Entre les deux parenthèses se trouve l'URI elle-même, facultativement encadrée d'apostrophes ou de guillemets, le tout facultativement encadré d'espaces vierges. Les trois exemples ci-dessous sont donc identiques :

```
url("http://www.monsite.fr/redpoint.png")
```

```
url( http://www.monsite.fr/redpoint.png )
```

```
url(http://www.monsite.fr/redpoint.png)
```

# Présentation des exercices

## BLOC-NOTES

À chaque début d'exercice un bloc-notes vous indique le numéro du TP. Ce numéro est utile car il sert de référence pour d'autres exercices.

## INTRODUCTION

L'introduction procure des indications sur l'exercice à venir et contient un encadré précisant les TP éventuellement prérequis et les fichiers d'exercice employés (à télécharger sur le site des éditions Dunod, [www.dunod.com](http://www.dunod.com)). Le temps de réalisation est une estimation du nombre de minutes à accorder par exercice.

## CAPTURE

Les copies d'écran présentées dans ce livre sont réalisées sur un ordinateur exécutant Windows avec comme navigateur Firefox 54. Si vous possédez un autre système d'exploitation ou un autre navigateur (ou même une version différente de Firefox) ou si la fenêtre de votre programme est de taille différente, vous pourriez remarquer certaines différences.



## Sélecteurs de pseudo-classes dynamiques

CSS introduit le concept de pseudo-éléments et de pseudo-classes pour permettre une mise en forme fondée sur des informations extérieures à l'arborescence du document. Les pseudo-classes classifient des éléments selon certaines de leurs caractéristiques qui ne peuvent être déduites de l'arborescence du document, plutôt que d'après leur nom, leurs attributs ou leur contenu. Les pseudo-classes peuvent être **dynamiques**, dans le sens où une caractéristique peut être modifiée et l'élément perdre ou acquérir une pseudo-classe selon les interactions de l'utilisateur.

**Prérequis :** TP 3 à 4  
**Fichier exemple :** TP\_CSS\_05\_début.html  
**Durée nécessaire :** 10 minutes

**1** Ouvrez dans votre éditeur de texte le fichier TP\_CSS\_05\_début.html, puis enregistrez-le sous le nom TP\_CSS\_05.html. Affichez ce fichier dans votre navigateur.

**2** Certains liens (les éléments A en HTML) font référence à un emplacement au sein d'une ressource. Ce type d'URI se termine par un signe dièse (#) suivi d'un identifiant d'ancre, nommé identificateur de fragment, et constitue un lien vers un élément du document nommé élément cible (*target*), comme dans le fichier exemple (dans le premier élément p) :

```
<a href="#cible">
```

Un élément cible peut être représenté par la pseudo-classe :target. Saisissez :

```
#cible:target {background-color: pink}
```

Enregistrez le fichier, rafraîchissez votre navigateur. Cliquez sur le lien : le tableau, cible de l'ancre, s'affiche avec un arrière-plan rose.

### Fichier d'exercice TP CSS 05 Sélecteurs de pseudo-classes dynamiques

Je suis un élément p racine. Allez-voir [là-bas](#) si j'y suis.

Je suis un élément div racine contenant un formulaire.

Un formulaire...  
 Ceci  
 ou cela.

Je suis un élément p racine.

Je suis un div racine avec dedans un tableau.

Colonne 1	Colonne 2	Colonne 3
C1L1	C2L1	C3L1
C1L2	C2L2	C3L2
C1L3	C2L3	C3L3
C1L4	C2L4	C3L4
C1L5	C2L5	C3L5

**3** Les pseudo-classes :link et :visited s'appliquent respectivement à des liens non visités et visités. Ces deux pseudo-classes sont mutuellement exclusives. En HTML, ces pseudo-classes s'appliquent exclusivement aux éléments a dotés d'un attribut href. Les deux déclarations suivantes sont équivalentes :

```
a:link {color: red ;}  
:link {color: red ;}
```

Et pour un lien visité :

```
:visited {color: green ;}
```

**PUCE**

Les puces des pas à pas vous aident à suivre l'ordre de lecture de l'exercice.

## TP 5 ● Sélecteurs de pseudo-classes dynamiques

Saisissez une des règles pour `:link`, puis la règle pour `:visited`. Enregistrez le fichier, rafraîchissez votre navigateur (désormais, nous ne rappellerons plus ces étapes, mais vous devez les respecter après chaque modification du fichier html). Comme nous avons déjà visité le lien, il s'affiche en vert.

**Sélecteurs de pseudo-classes dynamiques**

Je suis un élément p racine. Allez-voir [là-bas](#) si j'y suis.

**4**

La pseudo-classe `:hover` s'applique lorsque l'utilisateur place son pointeur sur un élément sans l'activer. La pseudo-classe `:active` s'applique lorsqu'un élément est activé par l'utilisateur, par exemple entre le moment où l'utilisateur appuie sur le bouton de la souris et celui où il le relâche. La pseudo-classe `:focus` s'applique lorsque l'élément possède le focus. Un élément peut posséder simultanément plusieurs de ces pseudo-classes. Saisissez :

```
a:hover {font-size: 2em ;}
a:active {font-size: 3em ;
          color: blue}
```

Lorsque vous placez le pointeur de la souris sur le lien, sa police double de taille. Si vous cliquez sur le lien et maintenez enfoncé le bouton de la souris, il s'agrandit encore et devient bleu.

Je suis un élément p racine. Allez-voir [là-bas](#) si j'y suis.

Je suis un élément p racine. Allez-voir [là-bas](#) si j'y suis.

**5**

Dans une page, l'utilisateur peut cocher une case ou une option. Certains éléments de menu sont cochés lorsque l'utilisateur les sélectionne. En ce cas, la pseudo-classe `:checked` s'applique. Cette pseudo-classe est dynamique par essence mais peut être fondée sur la présence d'attributs dans le document (comme les attributs HTML `selected` et `checked`). Saisissez :

```
:checked {width: 100px ;}
```

Je suis un élément div racine contenant un formulaire.

Un formulaire...

Ceci  ou cela.

Lorsque vous cochez une des cases des options, l'élément s'agrandit.

Fermez le fichier TP\_CSS\_05.html dans l'éditeur et le navigateur.

20

**ZOOM**

Le plus souvent, les captures d'écran sont recadrées sur la zone pertinente.

**SAISISSEZ**

Les TP vous demanderont de saisir de nouvelles règles CSS. Presque toujours, les fichiers d'exercices possèdent déjà une partie de feuille de styles remplie : dans l'élément `style`, des règles CSS sont encadrées par les commentaires `/*` et `*/` ne pas toucher à ces règles `/*` et `*/` saisir les nouvelles règles à partir d'ici `*/`. Ignorez ces règles : vous pourrez toujours y revenir plus tard. Elles servent à mettre en forme de façon optimale l'exercice qui vous est proposé. Vous apprendrez peu à peu ce qu'elles signifient. Saisissez les nouvelles règles après le dernier commentaire.



# Mise en œuvre de CSS

Vous pouvez appliquer un ou plusieurs styles à votre page de plusieurs façons : dans le corps du document, dans un élément, à l'aide de l'attribut `style` ou dans l'élément `head`, à l'aide d'un élément `style` ou d'un élément `link`.

**Prérequis :** néant

**Fichier exemple :** TP\_CSS\_01\_début.html

**Durée nécessaire :** 10 minutes

**1** Ouvrez dans votre éditeur de texte le fichier TP\_CSS\_01\_début.html, puis enregistrez-le sous le nom TP\_CSS\_01.html. Affichez ce fichier dans votre navigateur (avec Notepad++, choisissez *Execution*, puis sélectionnez votre navigateur dans la liste des propositions *Launch in...*).

**2** La première méthode consiste à saisir des styles dans la balise d'ouverture d'un élément lui-même, à l'aide de son attribut `style`. Cette méthode est hautement déconseillée. Dans le fichier exemple, ajoutez dans le premier élément `p` l'attribut et sa valeur figurant en gras ci-dessous :

```
<body>
  <h1>Ce titre est en bleu</h1>
  <p style="color:green">
    ce paragraphe est en vert.
  </p>
</body>
```

Enregistrez votre fichier, puis rafraîchissez votre navigateur. Le paragraphe s'affiche en vert.

**3** À l'intérieur de l'élément `head`, avec un élément `style`. Cette méthode est normalement réservée à la mise au point provisoire d'une feuille de styles ou pour une page unique. C'est pour cette dernière raison que nous l'emploierons par facilité dans ce livre. Saisissez le code en gras ci-dessous à l'intérieur de l'élément `head`, avant sa balise de fermeture :

```
<head>
  (...)
  <style type="text/css">
    h1 {color: blue}
  </style>
</head>
```

Enregistrez votre fichier, puis rafraîchissez votre navigateur. Le titre s'affiche en bleu.

## Ce titre est en bleu

ce paragraphe est en vert

Pour faire un site web il faut :

- Un éditeur de texte
- Connaître le HTML
- Connaître le CSS

## Ce titre est en bleu

ce paragraphe est en vert

Pour faire un site web il faut :

- Un éditeur de texte
- Connaître le HTML
- Connaître le CSS

**4** À l'intérieur de l'élément `head`, avec un élément `link` vers une feuille de styles externe. C'est la méthode fortement recommandée. Ajoutez le code en gras ci-dessous à l'intérieur de l'élément `head`, avant l'élément `style` :

```
<head>
  (...)
  <link href="style01.css"
        rel="stylesheet">
  <style>
  (...)
</head>
```

Enregistrez votre fichier, puis rafraîchissez votre navigateur. Vous voyez de nouvelles mises en forme s'appliquer au document.



**5** Ouvrez dans l'éditeur le fichier `style01.css` pour en examiner grossièrement le code (sans entrer dans le détail). Le mécanisme de cascade permet au navigateur de classer les règles selon leur spécificité pour déterminer celle qui est applicable.

Fondamentalement, tout élément hérite du style de son parent (l'élément qui le contient), sauf si la feuille de styles précise le contraire. La théorie de l'héritage est donc très simple : vous partez de l'information la plus externe (la feuille de styles externe) et progressez vers l'intérieur, comme suit :

- propriétés de feuille de styles externe de l'élément courant ;
- propriétés de feuille de styles locale de l'élément courant ;
- styles de ligne de l'élément courant.

C'est pourquoi dans notre cas, même si la feuille de styles externe définit une couleur pour le titre et les paragraphes, les styles définis en interne (dans le `head` pour le titre et dans l'attribut `style` pour le `p`) prennent le dessus.

Remarquez toutefois que si l'élément `link` avait été situé après l'élément `style`, dans le `head`, la feuille de styles externe aurait pris le dessus sur les règles de l'élément `style`, car venant après. Ce comportement semble toutefois parfois dépendre du navigateur.

Fermez les fichiers dans l'éditeur et le navigateur.



# Rédaction d'une feuille de styles externe

Une feuille de styles externe est un fichier texte ordinaire, créé dans un éditeur de texte quelconque et enregistré avec une extension `.css`. Elle est composée de règles CSS et de commentaires. Une règle CSS se compose d'un sélecteur suivi d'un bloc contenant une ou plusieurs déclarations.

**Prérequis :** néant

**Fichier exemple :** TP\_CSS\_02\_début.html

**Durée nécessaire :** 5 minutes

**1** Ouvrez dans votre éditeur de texte le fichier `TP_CSS_02_début.html`, puis enregistrez-le sous le nom `TP_CSS_02.html`. Dans ce fichier, ajoutez dans l'élément `head` la ligne en gras ci-dessous :

```
<head>
(...)
  <link href="style02.css"
        rel="stylesheet">
</head>
```

Enregistrez le fichier puis exécutez-le dans votre navigateur.

## Titre du TP CSS 02

Ceci est un paragraphe de premier niveau

Ceci est un élément div de premier niveau, qui contient deux paragraphes :

Un premier paragraphe...

et un second.

**2** Ouvrez dans votre éditeur de texte un nouveau fichier, puis enregistrez-le sous le nom `style02.css`.

**3** Dans le fichier `style02.css`, saisissez une première ligne :

```
/* Feuille de styles style02.css du TP
CSS 02 */
```

C'est un commentaire : il débute par les caractères `/*` et finit par `*/`. Il peut comporter plusieurs lignes et se situer n'importe où dans la feuille de styles. Son contenu est sans effet sur le rendu. Il est impossible d'imbriquer des commentaires.

Veillez à toujours commenter votre code CSS : un commentaire initial précisant le but de la feuille de styles est indispensable pour vous y retrouver par la suite.

**4** Saisissez en dessous votre première règle CSS :

```
h1 {font-style: italic}
```

Une règle CSS se compose d'un sélecteur suivi d'un bloc de déclaration(s) :

- Un sélecteur se compose de tout ce qui précède la première accolade ouvrante (`{`). Ici c'est `h1`, faisant référence à l'élément HTML `h1` de titre de premier niveau.
- Un bloc de déclaration débute par une accolade ouvrante (`{`) et se termine par une accolade fermante (`}`). Des espaces à l'intérieur des accolades sont facultatifs.
- Une déclaration est vide ou se compose du nom d'une propriété (ici `font-style`) suivi d'un signe deux-points (`:`) et d'une valeur (ici `italic`). Des espaces autour du signe deux-points sont facultatifs.

Enregistrez le fichier, puis rafraîchissez le navigateur : le titre apparaît en italique.

## Titre du TP CSS 02

Ceci est un paragraphe de premier niveau

Ceci est un élément div de premier niveau, qui contient deux paragraphes :

Un premier paragraphe...

et un second.

**5** Saisissez maintenant à la suite les règles ci-dessous. Ne cherchez pas à quoi correspondent les sélecteurs ou les déclarations employés, nous y reviendrons largement : regardez simplement la structure.

```
h1 {font-weight: bold}
h1 {font-size: 2em}
```

Plusieurs déclarations peuvent être rassemblées en un bloc, sous forme de liste séparée par des points-virgules (;). Les trois règles précédentes peuvent être rassemblées en procédant comme montré ci-dessous : ajoutez les points-virgules bleus et remplacez ce qui est rouge barré par des espaces.

```
h1 {font-style: italic ;⚡
h1{font-weight: bold ;⚡
h1{font-size: 2em ;}
```

N'oubliez surtout pas le point-virgule à la fin de chaque déclaration. Il est d'ailleurs conseillé de le mettre même lorsqu'il n'y a qu'une déclaration, cela évitant toute erreur potentielle en cas d'ajout de déclaration. Et remarquez comme les indentations clarifient le code.

Enregistrez le fichier, puis rafraîchissez le navigateur.

**6** Saisissez à la suite les règles ci-dessous :

```
div {background-color: grey ;
      color: white ;}
p {background-color:
    rgba(255,255,255,.6) ;
   color: blue ;}
```

Enregistrez le fichier, puis rafraîchissez le navigateur.

## Titre du TP CSS 02

Ceci est un paragraphe de premier niveau

Ceci est un élément div de premier niveau, qui contient deux paragraphes :

Un premier paragraphe...

et un second.

## Titre du TP CSS 02

Ceci est un paragraphe de premier niveau

Ceci est un élément div de premier niveau, qui contient deux paragraphes :

Un premier paragraphe...

et un second.

**7** Pour attribuer une même règle à plusieurs éléments, séparez les sélecteurs par une virgule. Saisissez la règle suivante :

```
div, p {margin: 10px ;}
```

## Titre du TP CSS 02

Ceci est un paragraphe de premier niveau

Ceci est un élément div de premier niveau, qui contient deux paragraphes :

Un premier paragraphe...

et un second.

Même si elle reste rudimentaire, vous avez construit ici une première feuille de styles. C'est ainsi qu'il faudrait rédiger votre CSS, que ce soit avec une feuille externe ou dans un élément `style` à l'intérieur de l'élément `head`. Dans ce livre, nous placerons toujours les déclarations CSS à l'intérieur de l'élément `head`, dans un élément `style`, comme exposé à l'étape 3 du TP précédent, afin d'éviter une inutile multiplication de fichiers.

Fermez le fichier `TP_CSS_02.html` dans l'éditeur et le navigateur, ainsi que le fichier CSS dans l'éditeur.



# Les sélecteurs

Fondamentalement, CSS se décompose en deux parties : les sélecteurs, pour sélectionner le ou les éléments concernés, et les propriétés appliquées à ou aux éléments sélectionnés. Il est indispensable de maîtriser au moins raisonnablement les sélecteurs avant de passer à l'étude des propriétés CSS. Nous ne le rappellerons pas dans les prérequis tellement cela est fondamental.

Un sélecteur CSS détermine quels éléments sont identifiés dans le document : autrement dit, à quoi vont s'appliquer le ou les styles définis dans le bloc de déclaration qui le suit. Ce peut être un simple nom d'élément ou une chaîne complexe de sélecteurs séparés par des opérateurs de combinaison. Le module Sélecteurs Niveau 3 a été intégré à la spécification CSS 2.1.

*Je suis un élément p racine.*

*je suis un div racine*

*Elément p contenu dans un élément div*

*Je suis une liste à puces (ul) enfant d'un div.*

- Premier élément li de la liste à puces
- Deuxième élément li de la liste à puces
- Troisième élément de liste, avec attribut REL fixé à 'un'
- Quatrième élément de liste de classe 'red cadre'
- Cinquième élément de liste avec comme attribut id 'imp'
- Sixième et dernier élément li de la liste à puces, de classe 'red'

```
<style>
* {font-style: italic;}
li {font-style:normal;}
ul li {color:blue;}
ul li + li {color:green;}
ul *[REL="un"] {color:red;}
ul li + li.red {background-color:rgba(255,150,150,.9);}
li.red.cadre { border:4px solid green;}
#imp {font-weight:bold;}
#imp:not(div) { background-color:rgb(220,220,200);}
</style>
```



# Sélecteurs d'éléments

La première catégorie de sélecteurs sélectionne des éléments directement d'après leur nom ou leur situation directe dans le document.

**Prérequis :** néant

**Fichier exemple :** TP\_CSS\_03\_début.html

**Durée nécessaire :** 10 minutes

**1** Ouvrez dans votre éditeur de texte le fichier TP\_CSS\_03\_début.html, puis enregistrez-le sous le nom TP\_CSS\_03.html. Affichez ce fichier dans votre navigateur.

**2** Le sélecteur universel \* correspond à tous les types d'éléments de l'arborescence du document, donc à tous les éléments. Un sélecteur de type, plus restrictif, identifie toute instance d'un unique type d'élément dans le document. Saisissez dans l'élément style de l'élément head :

```
p {color: blue ;}
```

Enregistrez le fichier, rafraîchissez votre navigateur. Tous les éléments p du document sont affichés en bleu.

**3** Il est parfois souhaitable de sélectionner des éléments qui sont des descendants d'un autre élément. Vous créez un sélecteur de descendant en combinant deux sélecteurs séparés par un espace vierge. Ajoutez une nouvelle règle dans l'élément style :

```
div p {color: red ;}
```

Enregistrez le fichier, rafraîchissez votre navigateur. Tout élément p descendant d'un élément div est affiché en rouge.

## Fichier d'exercice TP CSS 03

Je suis un élément p racine.

Je suis un élément div racine.

Je suis un élément div enfant d'un élément div racine.

Je suis un élément p, enfant d'un élément div lui-même enfant d'un élément div.

je suis la suite de l'élément div enfant d'un élément div.

Je suis un élément p racine.

je suis un div racine

Je suis une liste ordonnée (ol) enfant d'un div.

1. Premier élément li de la liste ordonnée

Element p contenu dans le premier élément de liste

2. Deuxième élément de liste
3. Troisième élément de liste

Elément p contenu dans un élément div

Element div racine

Dernier élément div racine du document.

## Fichier d'exercice TP CSS 03

Je suis un élément p racine.

Je suis un élément div racine.

Je suis un élément div enfant d'un élément div racine.

Je suis un élément p, enfant d'un élément div lui-même enfant d'un élément div.

je suis la suite de l'élément div enfant d'un élément div.

Je suis un élément p racine.

je suis un div racine

Je suis une liste ordonnée (ol) enfant d'un div.

1. Premier élément li de la liste ordonnée

Element p contenu dans le premier élément de liste

2. Deuxième élément de liste
3. Troisième élément de liste

Elément p contenu dans un élément div

Element div racine

Dernier élément div racine du document.

**4** Il est possible de créer des sélecteurs de descendants plus complexes. Saisissez :

```
div * p {color: green ;}
```

Un sélecteur complexe se lit généralement mieux en allant de la droite vers la gauche : vous sélectionnez ici les éléments `p` descendants de n'importe quel élément (le sélecteur universel `*`) lui-même descendant d'un élément `div`. Enregistrez le fichier, rafraîchissez votre navigateur.

### Fichier d'exercice TP CSS 03

*Je suis un élément p racine.*

Je suis un élément div racine.

Je suis un élément div enfant d'un élément div racine.

*Je suis un élément p, enfant d'un élément div lui-même enfant d'un élément div.*

je suis la suite de l'élément div enfant d'un élément div.

*Je suis un élément p racine.*

je suis un div racine

Je suis une liste ordonnée (ol) enfant d'un div.

1. Premier élément li de la liste ordonnée

*Element p contenu dans le premier élément de liste*

2. Deuxième élément de liste
3. Troisième élément de liste

*Elément p contenu dans un élément div*

Element div racine

Dernier élément div racine du document.

**5** Un sélecteur d'enfant identifie un élément qui est l'enfant direct d'un autre élément. Le combinateur employé est le signe supérieur (`>`). Cette règle va s'appliquer à tous les éléments `p` enfants directs de l'élément `body` (mais donc pas à tous les `p`, comme la première règle que nous avons écrite) :

```
body > p {font-style: italic}
```

Enregistrez le fichier, rafraîchissez votre navigateur.

### Fichier d'exercice TP CSS 03

*Je suis un élément p racine.*

Je suis un élément div racine.

Je suis un élément div enfant d'un élément div racine.

*Je suis un élément p, enfant d'un élément div lui-même enfant d'un élément div.*

je suis la suite de l'élément div enfant d'un élément div.

*Je suis un élément p racine.*

je suis un div racine

Je suis une liste ordonnée (ol) enfant d'un div.

1. Premier élément li de la liste ordonnée

*Element p contenu dans le premier élément de liste*

2. Deuxième élément de liste
3. Troisième élément de liste

*Elément p contenu dans un élément div*

Element div racine

Dernier élément div racine du document.

**6** Vous pouvez combiner des sélecteurs de descendants et d'enfants. Ajoutez la règle :

```
div ol>li p {color: purple}
```

Vous identifiez ici un élément `p` descendant d'un élément `li` (ligne de liste), lui-même un enfant direct d'un élément `ol` (liste numérotée), lui-même descendant d'un élément `div`. Remarquez que les espaces vierges facultatifs autour du combinateur `>` sont ici absents, ce qui nuit un peu à la lisibilité. Mieux vaut toujours les ajouter. Enregistrez le fichier, rafraîchissez votre navigateur.

### Fichier d'exercice TP CSS 03

*Je suis un élément p racine.*

Je suis un élément div racine.

Je suis un élément div enfant d'un élément div racine.

*Je suis un élément p, enfant d'un élément div lui-même enfant d'un élément div.*

je suis la suite de l'élément div enfant d'un élément div.

*Je suis un élément p racine.*

je suis un div racine

Je suis une liste ordonnée (ol) enfant d'un div.

1. Premier élément li de la liste ordonnée

*Element p contenu dans le premier élément de liste*

2. Deuxième élément de liste
3. Troisième élément de liste

*Elément p contenu dans un élément div*

Element div racine

Dernier élément div racine du document.

**7** Un sélecteur de parent adjacent identifie un élément frère d'un autre élément et situé immédiatement après ce dernier. Il est créé à l'aide du combinateur +.

```
li + li {color: olive}
```

Enregistrez le fichier, rafraîchissez votre navigateur. Tout élément de liste sauf le premier possédera une police olive (les éléments suivants ont toujours un élément `li` comme frère précédent).

**8** Vous pouvez sélectionner un élément frère d'un autre élément, situé n'importe où après lui à l'aide du sélecteur de frère, créé à l'aide du combinateur ~.

```
div ~ ul {margin-left: 10px}
```

Enregistrez le fichier, rafraîchissez votre navigateur. Toute liste à puces `ul` située après (mais pas forcément immédiatement) un élément `div` possède une marge gauche de 10 pixels (il n'y en a qu'une dans notre fichier exemple).

### Fichier d'exercice TP CSS 03

*Je suis un élément p racine.*

Je suis un élément div racine.

Je suis un élément div enfant d'un élément div racine.

*Je suis un élément p, enfant d'un élément div lui-même enfant d'un élément div.*

je suis la suite de l'élément div enfant d'un élément div.

*Je suis un élément p racine.*

je suis un div racine

Je suis une liste ordonnée (ol) enfant d'un div.

1. Premier élément li de la liste ordonnée

*Element p contenu dans le premier élément de liste*

2. Deuxième élément de liste

3. Troisième élément de liste

*Elément p contenu dans un élément div*

Element div racine

Dernier élément div racine du document.

### Fichier d'exercice TP CSS 03

*Je suis un élément p racine.*

Je suis un élément div racine.

Je suis un élément div enfant d'un élément div racine.

*Je suis un élément p, enfant d'un élément div lui-même enfant d'un élément div.*

je suis la suite de l'élément div enfant d'un élément div.

*Je suis un élément p racine.*

je suis un div racine

Je suis une liste ordonnée (ol) enfant d'un div.

1. Premier élément li de la liste ordonnée

*Element p contenu dans le premier élément de liste*

2. Deuxième élément de liste

3. Troisième élément de liste

*Elément p contenu dans un élément div*

Element div racine

Dernier élément div racine du document.

Fermez le fichier `TP_CSS_03.html` dans l'éditeur et le navigateur.