



# Le CSS 3



Cours tiré de : <https://openclassrooms.com/courses/apprenez-a-creer-votre-site-web-avec-html5-et-css3>

## I) La découverte du CSS :

### 1) Introduction :

Le CSS (*Cascading Style Sheets*) est un langage qui vient compléter le HTML pour mettre en forme une page web.

Il faut savoir qu'aux débuts du Web, CSS n'existait pas. En fait, il n'y avait initialement que le langage HTML. Le HTML est né en 1991 et CSS en 1996. La mise en forme de 1991 à 1996 était uniquement en HTML ! Il existe en effet des balises HTML dédiées à la mise en forme.

Cependant, les pages HTML commençaient à devenir assez complexes. Il y avait de plus en plus de balises et c'était un joyeux mélange entre le fond et la forme, qui rendait la mise à jour des pages web de plus en plus complexe. C'est pour cela que l'on a créé le langage CSS.

Cependant, le CSS n'a pas été adopté immédiatement par les webmasters. Il fallait qu'ils se défassent de certaines mauvaises habitudes et cela a pris du temps. Encore aujourd'hui, on trouve des sites web avec des balises HTML de mise en forme.

Les navigateurs web doivent lire *le* code CSS et comprendre comment afficher la page. Aucun navigateur ne connaît parfaitement toutes les fonctionnalités CSS, si le navigateur ne connaît pas une propriété CSS, il l'ignore et ne met pas en forme.

### 2) Place du CSS :

On peut écrire du code en langage CSS à trois endroits différents :

- dans un fichier .css (*méthode la plus recommandée*) ;
- dans l'en-tête <head> du fichier HTML avec la balise <style> (méthode acceptable)
- directement dans les balises du fichier HTML (*méthode non recommandée*)

### 3) Le fichier .css :

On écrit le code CSS dans un fichier spécial ayant l'extension .css.

On place cette ligne de code : <link rel="stylesheet" href="style.css" /> dans le <head> du html

Elle qui indique que ce fichier HTML est associé à un fichier appelé style.css et chargé de la mise en forme. Il faut alors créer un fichier nommé style.css que l'on placera dans le même répertoire que la page html.

## II) Appliquer un style :

### 1) Sélectionner une balise :

Dans un code CSS, on trouve trois éléments différents :

- **Des noms de balises** : on écrit les noms des balises dont on veut modifier l'apparence. Par exemple, si je veux modifier l'apparence de tous les paragraphes <p>, je dois écrire p.
- **Des propriétés CSS** : les « effets de style » de la page sont rangés dans des propriétés. Il y a par exemple la propriété color qui permet d'indiquer la couleur du texte, font-size qui permet d'indiquer la taille du texte, etc...
- **Les valeurs** : pour chaque propriété CSS, on doit indiquer une valeur. Par exemple, pour la propriété color, il faut indiquer le nom de la couleur. Pour font-size, il faut indiquer quelle taille on veut, etc...

Exemple :

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

On écrit le nom de la balise (ici p) et on ouvre des accolades pour, à l'intérieur, mettre les propriétés et valeurs que l'on souhaite. On peut mettre autant de propriétés que l'on veut à l'intérieur des accolades. Chaque propriété est suivie du symbole « deux-points » ( : ) puis de la valeur correspondante. Enfin, chaque ligne se termine par un point-virgule ( ; ).

### Remarque :

Pour faire un commentaire, il suffit de taper /\*, suivi du commentaire, puis \*/ pour terminer.

### 2) Appliquer un style à plusieurs balises :

Si les deux balises doivent avoir la même présentation. Il suffit de combiner la déclaration en séparant les noms des balises par une virgule, comme ceci :

```
h1, em
{
  color: blue;
}
```

### 3) Class et id :

Pour différencier les paragraphes et pour éviter qu'ils aient tous la même mise en forme, on utilise les attributs class et id que l'on peut mettre sur n'importe quelle balise, aussi bien titre que paragraphe, image, etc...

Exemple : sur le html : <p class="introduction">Bonjour et bienvenue sur mon site !</p>  
et dans le CSS :

```
.introduction
{
  color: blue;
}
```

### La différence entre class et id :

- 2 balises peuvent avoir le même nom avec l'attribut class.
- Un nom d'id doit en revanche être unique dans la page HTML.

Dans le CSS : l'attribut class est appelé avec un . et l'id avec un #

#### 4) Les balises universelles :

Il arrivera parfois que vous ayez besoin d'appliquer une class (ou un id) à certains mots qui, à l'origine, ne sont pas entourés par des balises. On utilise alors deux balises dites **universelles**, qui n'ont aucune signification particulière : `<span>` et `<div>`.

- `<span> </span>` : c'est une balise de type **inline**, c'est-à-dire une balise que l'on place au sein d'un paragraphe de texte, pour sélectionner certains mots uniquement. Les balises `<strong>` et `<em>` sont de la même famille. Cette balise s'utilise donc au milieu d'un paragraphe et c'est celle dont nous allons nous servir pour colorer « bienvenue ».
- `<div> </div>` : c'est une balise de type **block**, qui entoure un bloc de texte. Les balises `<p>`, `<h1>`, etc. sont de la même famille. Ces balises ont quelque chose en commun : elles créent un nouveau « bloc » dans la page et provoquent donc obligatoirement un retour à la ligne. `<div>` est une balise fréquemment utilisée dans la construction d'un design, comme nous le verrons plus tard.

#### 5) Les sélecteurs avancés :

Les sélecteurs servent à sélectionner des parties de la page HTML que l'on souhaite mettre en forme. Voici quelques sélecteurs particuliers :

##### a) Le sélecteur universel : \*

Sélectionne toutes les balises sans exception. On l'appelle le sélecteur universel.

##### b) A B : une balise contenue dans une autre :

Ex : `h3 em`

Sélectionne toutes les balises `<em>` situées à l'intérieur d'une balise `<h3>`.

(il n'y a pas de virgule entre les deux noms de balises)

##### c) A + B : une balise qui en suit une autre :

Ex : `h3 + p`

Sélectionne la première balise `<p>` située après un titre `<h3>`.

##### d) A[attribut] : une balise qui possède un attribut

Ex : `a[title]`

Sélectionne tous les liens `<a>` qui possèdent un attribut `title`.

##### e) A[attribut="Valeur"] : une balise, un attribut et une valeur exacte

Ex : `a[title="Cliquez ici"]`

Idem d), mais l'attribut doit en plus avoir exactement pour valeur « Cliquez ici ».

##### f) A[attribut\*="Valeur"] : une balise, un attribut et une valeur

Ex : `a[title*="ici"]`

Idem e), l'attribut doit cette fois contenir dans sa valeur le mot « ici » (peu importe sa position).

##### g) D'autres sélecteurs existent !

### III) La mise en forme du texte :

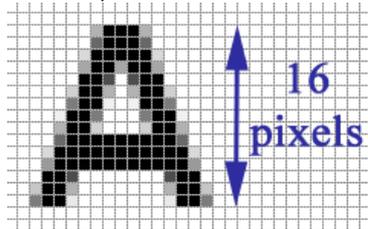
#### 1) La taille :

Pour modifier la taille du texte, on utilise la propriété CSS **font-size**.

On peut indiquer soit la taille absolue soit la taille relative :

- La **taille absolue** est souvent donnée en pixels (en centimètres cm ou millimètres mm aussi). Cette méthode est très précise mais il est conseillé de ne l'utiliser que si c'est absolument nécessaire, car on risque d'indiquer une taille trop petite pour certains lecteurs.

Ex : Pour avoir un texte de 16 pixels de hauteur, vous devez donc écrire : font-size: 16px;



- La **taille relative** est indiquée en pourcentage, « em » ou « ex », cette technique a l'avantage d'être plus souple. Elle s'adapte plus facilement aux préférences de taille des visiteurs.

Il y a plusieurs moyens d'indiquer une valeur relative : (7 tailles disponibles)

- xx-small : minuscule ;
- x-small : très petit ;
- small : petit ;
- medium : moyen ;
- large : grand ;
- x-large : très grand ;
- xx-large : gigantesque.

Exemple : font-size: small;

On peut aussi indiquer la taille en « em ».

- Si vous écrivez 1em, le texte a une taille normale.
- Si vous voulez grossir le texte, vous pouvez inscrire une valeur supérieure à 1, comme 1.3em.
- Si vous voulez réduire le texte, inscrivez une valeur inférieure à 1, comme 0.8em.

Exemple : font-size: 0.8em;

D'autres unités sont disponibles. On peut utiliser le « ex » (qui fonctionne sur le même principe que le em mais qui est plus petit de base) et le pourcentage (80%, 130%...).

#### 2) La police :

##### a) Police classique :

La police peut poser un problème car pour qu'une police s'affiche correctement, il faut que tous les internautes l'aient. Si un internaute n'a pas la même police que vous, son navigateur prendra une police par défaut (une police standard) qui n'aura peut-être rien à voir avec ce à quoi vous vous attendiez. Par contre avec CSS 3, il est possible de faire télécharger automatiquement une police au navigateur.

La propriété CSS qui permet d'indiquer la police à utiliser est **font-family**

Exemple : font-family : arial ;

Pour éviter les problèmes si l'internaute n'a pas la même police que vous, on précise en général plusieurs noms de police, séparés par des virgules. Exemple : font-family : arial, impact, verdana ;

Remarque : si la police possède des espaces (arial black), on l'appelle entre guillemets.

Font-family : "arial black" ;

## b) Police personnalisée :

On va utiliser l'instruction **@font-face**

- Il faut que le navigateur télécharge automatiquement le fichier de la police, dont le poids peut atteindre, voire dépasser 1 Mo...
- La plupart des polices sont soumises au droit d'auteur, il n'est donc *pas légal* de les utiliser sur son site si elles ne sont pas libres de droit.
- Il existe *plusieurs formats* de fichiers de polices et ceux-ci ne fonctionnent pas sur tous les navigateurs

Voici les différents formats de fichiers de polices qui existent et qu'il faut connaître :

- .ttf : TrueType Font. Fonctionne sur IE9 et tous les autres navigateurs.
- .eot : Embedded OpenType. Fonctionne sur Internet Explorer uniquement, toutes versions. Ce format est propriétaire, produit par Microsoft.
- .otf : OpenType Font. Ne fonctionne pas sur Internet Explorer.
- .svg : SVG Font. Le seul format reconnu sur les iPhones et iPads pour le moment.
- .woff : Web Open Font Format. Nouveau format conçu pour le Web, qui fonctionne sur IE9 et tous les autres navigateurs.

```
Exemple : @font-face {  
          font-family: 'MaPolice';  
          src: url('MaPolice.eot');  
        }
```

## 3) Italique, gras, souligné... :

Pour mettre en italique, on utilise **font-style** qui peut prendre trois valeurs :

- italic : le texte sera mis en italique.
- oblique : le texte sera passé en oblique
- normal : le texte sera normal (par défaut).

Pour mettre en gras, on utilise **font-weight** qui peut prendre les valeurs suivantes :

- bold : le texte sera en gras ;
- normal : le texte sera écrit normalement (par défaut).

Pour les autres effets, on utilise **text-decoration** qui permet, entre autres, de souligner le texte, mais pas seulement. Voici les différentes valeurs qu'elle peut prendre :

- underline : souligné.
- line-through : barré.
- overline : ligne au-dessus.
- blink : clignotant. Ne fonctionne pas sur tous les navigateurs
- none : normal (par défaut).

## 4) L'alignement :

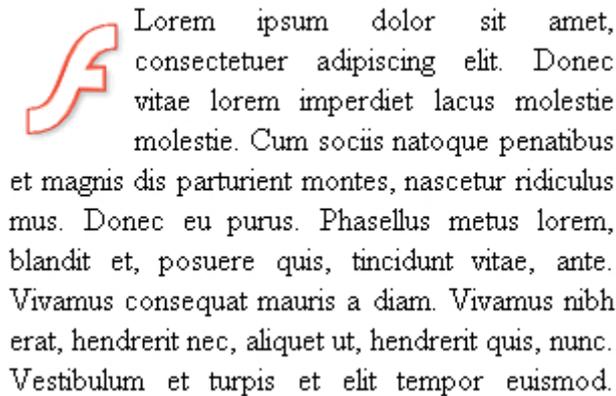
On utilise la propriété **text-align** et on indique l'alignement désiré :

- left : le texte sera aligné à gauche (c'est le réglage par défaut).
- center : le texte sera centré.
- right : le texte sera aligné à droite.
- justify : le texte sera « justifié ». Justifier le texte permet de faire en sorte qu'il prenne toute la largeur possible sans laisser d'espace blanc à la fin des lignes.

Remarque : On ne peut pas modifier l'alignement du texte d'une balise *inline* (comme <span>, <a>, <em>, <strong>...). L'alignement ne fonctionne que sur des balises de type *block* (<p>, <div>, <h1>, <h2>, ...) C'est donc en général le paragraphe entier qu'il vous faudra aligner.

## 5) Les flottants :

Le CSS nous permet de faire flotter un élément autour du texte. On dit aussi qu'on fait un « habillage ». Exemple :



On utilise l'instruction **float**. Cette propriété peut prendre deux valeurs très simples :

- left : l'élément flottera à gauche.
- right : l'élément flottera... à droite.

On peut aussi bien utiliser la propriété float sur des balises *block* que sur des balises *inline*

Appliquer un float à une balise.

Puis continuer à écrire du texte à la suite normalement

Exemple :

En html :

```
<p> Ceci est un texte normal.</p>
```

En CSS :

```
.imageflottante
{
  float: left;
}
```

Quand on met en place un flottant, le texte autour l'habille. Mais il faut qu'après le texte continue *en dessous* du flottant.

Il existe en fait une propriété CSS qui permet de dire : « *Stop, ce texte doit être en-dessous du flottant et non plus à côté* ». C'est la propriété clear, qui peut prendre ces trois valeurs :

- left : le texte se poursuit en-dessous après un float: left;
- right : le texte se poursuit en-dessous après un float: right;
- both : le texte se poursuit en-dessous, que ce soit après un float: left; ou après un float: right;.

Pour simplifier, on utilise tout le temps le clear: both, qui marche après un flottant à gauche et après un flottant à droite.

Exemple : en html : 

```
<p></p>
<p>Ce texte est écrit à côté de l'image flottante.</p>
<p class="dessous">Ce texte est écrit sous l'image flottante.</p>
```

en CSS :

```
.imageflottante
{ float: left; }
.dessous
{clear: both;}
```

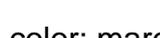
#### IV) La couleur et le fond :

##### 1) La couleur du texte :

La propriété qui permet de modifier la couleur du texte est **color**.

##### a) Le nom :

La méthode la plus simple et la plus pratique pour choisir une couleur consiste à taper son nom (*in english*) mais il n'existe que seize couleurs dites « standard » :

white	
silver	
gray	
black	
red	
maroon	
lime	
green	
yellow	
olive	
blue	
navy	
fuchsia	
purple	
aqua	
teal	

Exemple : `color: maroon;`

##### b) Hexadécimal :

La plupart des écrans peuvent afficher seize *millions* de couleurs. Il existe en CSS plusieurs façons de choisir une couleur parmi toutes celles qui existent. On peut donner une couleur grâce à la notation hexadécimale. Un nom de couleur en hexadécimal ressemble à : `#FF5A28`. On commence par un `#` puis les deux suivants indiquent une quantité de rouge, les deux suivants une quantité de vert et les deux derniers une quantité de bleu. En mélangeant ces quantités (qui sont les composantes Rouge-Vert-Bleu de la couleur) on peut obtenir la couleur qu'on veut.

Ainsi, `#000000` correspond à la couleur noire et `#FFFFFF` à la couleur blanche.

Certains logiciels de dessin comme Photoshop et Gimp vous indiquent les couleurs en hexadécimal.

Exemple : `color: #FFFFFF;`

##### c) Le RGB (Red Green Blue):

Comme avec la notation hexadécimale, pour choisir une couleur, on doit définir une quantité de rouge, de vert et de bleu. Il faut taper `rgb` en plaçant les nombres correspondants compris entre 0 et 255.

Exemple : `color: rgb(240,96,204);`

Avec le logiciel de dessin Paint, on peut trouver la couleur voulue.

**Remarque :** On peut trouver tous les codes sur <http://www.colorpicker.com>

## 2) Le fond :

### a) Couleur :

Pour indiquer une couleur de fond, on utilise la propriété CSS **background-color**. Elle s'utilise de la même manière que la propriété color, c'est-à-dire qu'on peut taper le nom d'une couleur, l'écrire en notation hexadécimale ou encore utiliser la méthode RGB.

Pour indiquer la couleur de fond de la page web, il faut travailler sur la balise <body> qui correspond à l'ensemble de la page web, c'est donc en modifiant sa couleur de fond que l'on changera la couleur d'arrière-plan de la page.

**Remarque** : Attention ! En CSS, si vous appliquez un style à une balise, toutes les balises qui se trouvent à l'intérieur prendront le même style !

### b) Image :

La propriété permettant d'indiquer une image de fond est **background-image**. Comme valeur, on doit renseigner url("nom\_de\_l\_image.png").

Exemple : background-image: url("fond.png");

On peut compléter la propriété background-image par plusieurs autres propriétés qui permettent de changer le comportement de l'image de fond :

- La propriété CSS background-attachment permet de « fixer » le fond. L'effet obtenu est qu'on voit alors le texte « glisser » par-dessus le fond. Deux valeurs sont disponibles :
  - fixed : l'image de fond reste fixe ;
  - scroll : l'image de fond défile avec le texte (par défaut).
- Par défaut, l'image de fond est répétée en mosaïque. On peut changer cela avec la propriété background-repeat :
  - no-repeat : le fond ne sera pas répété. L'image sera donc unique sur la page.
  - repeat-x : le fond sera répété uniquement sur la première ligne, horizontalement.
  - repeat-y : le fond sera répété uniquement sur la première colonne, verticalement.
  - repeat : le fond sera répété en mosaïque (par défaut).
- On peut indiquer où doit se trouver l'image de fond avec background-position. Cette propriété n'est intéressante que si elle est combinée avec background-repeat: no-repeat; (un fond qui ne se répète pas).

Vous devez donner à background-position deux valeurs en pixels pour indiquer la position du fond par rapport au coin supérieur gauche de la page (ou du paragraphe, si vous appliquez le fond à un paragraphe). Exemple : background-position: 30px 50px;

Il est aussi possible d'utiliser ces valeurs :

- top : en haut ;
- bottom : en bas ;
- left : à gauche ;
- center : centré ;
- right : à droite.

Il est possible de combiner ces mots. Par exemple : background-position: top right;

- Quand on utilise beaucoup de propriétés en rapport avec le fond, on peut utiliser la propriété **background** dont la valeur peut combiner plusieurs propriétés : background-image, background-repeat, background-attachment et background-position.

Exemple : background: url("soleil.png") fixed no-repeat top right;

- Il est possible de donner plusieurs images de fond à un élément. Pour cela, il suffit de séparer les déclarations par une virgule.

Exemple : `background: url("soleil.png") fixed no-repeat top right, url("neige.png") fixed;`  
La première image de cette liste sera placée par-dessus les autres. Attention donc, l'ordre de déclaration des images a son importance : si on inverse le soleil et la neige dans le code CSS précédent, on ne voit plus plus le soleil !

### 3) La transparence :

On peut faire varier les niveaux de transparence des éléments. On utilise la propriété **opacity** et la notation RGBa.

La propriété opacity permet d'indiquer le niveau d'opacité (c'est l'inverse de la transparence).

- Avec une valeur de 1, l'élément sera totalement opaque : c'est le comportement par défaut.
- Avec une valeur de 0, l'élément sera totalement transparent.

Il faut donc choisir une valeur comprise entre 0 et 1. Ainsi, avec une valeur de 0.6, votre élément sera opaque à 60%... et on verra donc à travers.

Exemple : `opacity: 0.6;`

Remarque :

Si on applique la propriété opacity à un élément de la page, *tout* le contenu de cet élément sera rendu transparent (même les images, les autres blocs à l'intérieur, etc.). Si on veut juste rendre la couleur de fond transparente, il faudra utiliser plutôt la notation RGBa.

La notation RGBa est la notation RGB avec un quatrième paramètre : le niveau de transparence (appelé « canal alpha »). De la même façon que précédemment, avec une valeur de 1, le fond est complètement opaque. Avec une valeur inférieure à 1, il est transparent.

Exemple : `background-color: rgba(255, 0, 0, 0.5); /* Fond rouge à moitié transparent */`

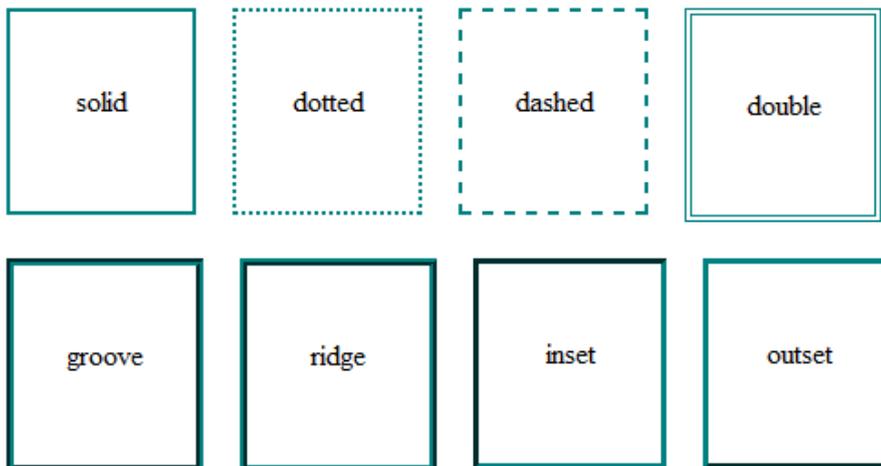
## V) Les bordures et les ombres :

### 1) Les Bordures :

De nombreuses propriétés CSS vous permettent de modifier l'apparence de vos bordures : `border-width`, `border-color`, `border-style` mais on va utiliser la propriété **border** qui regroupe l'ensemble de ces propriétés.

Pour border on peut utiliser jusqu'à trois valeurs pour modifier l'apparence de la bordure :

- La largeur : indiquez la largeur de votre bordure. Mettez une valeur en pixels (comme 2px).
- La couleur : c'est la couleur de votre bordure.
- Le type de bordure :
  - none : pas de bordure (par défaut) ;
  - solid : un trait simple ;
  - dotted : pointillés ;
  - dashed : tirets ;
  - double : bordure double ;
  - groove : en relief ;
  - ridge : autre effet relief ;
  - inset : effet 3D global enfoncé ;
  - outset : effet 3D global surélevé.



Exemple : `border: 3px blue dashed;`

Si on veut mettre des bordures différentes en fonction du côté (haut, bas, gauche ou droite), on utilise ces quatre propriétés :

- `border-top` : bordure du haut ;
- `border-bottom` : bordure du bas ;
- `border-left` : bordure de gauche ;
- `border-right` : bordure de droite.

Exemple : `border-left: 2px solid black;`

La propriété `border-radius` permet d'arrondir facilement les angles de n'importe quel élément. Il suffit d'indiquer la taille (« l'importance ») de l'arrondi en pixels :

Exemple : `border-radius: 10px;`

On peut aussi préciser la forme de l'arrondi pour chaque coin : `border-radius: 10px 5px 10px 5px;`

Il est possible d'affiner l'arrondi des angles en créant des courbes elliptiques. Dans ce cas, il faut indiquer deux valeurs séparées par une barre oblique (*slash*, caractère /).

Exemple : `border-radius: 20px / 10px;`

## 2) Les Ombres :

Il existe deux types d'ombres :

- les ombres des boîtes ;
- les ombres du texte.

La propriété **box-shadow** s'applique à tout le bloc et prend quatre valeurs dans l'ordre suivant :

1. le décalage horizontal de l'ombre ;
2. le décalage vertical de l'ombre ;
3. l'adoucissement du dégradé ;
4. la couleur de l'ombre.

Exemple, pour une ombre noire de 6 pixels, sans adoucissement : `box-shadow: 6px 6px 0px black;`  
On peut aussi rajouter une cinquième valeur facultative : `inset`. Dans ce cas, l'ombre sera placée à l'intérieur du bloc, pour donner un effet enfoncé.

Avec **text-shadow**, on peut ajouter une ombre directement sur les lettres de votre texte. Les valeurs fonctionnent exactement de la même façon que `box-shadow` : décalage, adoucissement et couleur.

## VI) Les apparences dynamiques :

Nous verrons dans ce chapitre comment changer l'apparence :

- au survol ;
- lors du clic ;
- lors du focus (élément sélectionné) ;
- lorsqu'un lien a été consulté.

### 1) Au survol :

**hover** signifie « survoler ». `a: hover` peut donc se traduire par : « Quand la souris est sur le lien » (quand on pointe dessus). Il faut alors définir l'apparence que doivent avoir les liens lorsqu'on pointe dessus.

On a défini ici deux versions des styles pour les liens :

- pour les liens par défaut (non survolés) ; exemple : `a { }`
- pour les liens au survol. `a : hover { }`

### 2) Au clic et lors de la sélection :

On peut changer l'apparence des éléments lorsque l'on clique dessus et lorsqu'ils sont sélectionnés. Le pseudo-format **:active** permet d'appliquer un style particulier *au moment du clic*. En pratique, il n'est utilisé que sur les liens.

Le lien gardera cette apparence très peu de temps : en fait, le changement intervient lorsque le bouton de la souris est enfoncé Exemple : `a: active { }`

**:focus** : lorsque l'élément est sélectionné. Le pseudo-format `:focus` applique un style *lorsque l'élément est sélectionné*.

### 3) Lorsque le lien a déjà été consulté :

Il est possible d'appliquer un style à un lien vers une page qui a déjà été vue. Par défaut, le navigateur colore le lien en violet. On peut changer cette apparence avec **:visited** (qui signifie « visité »).