



Fin des années 90, les développeurs passaient leur temps à répéter les mêmes balises pour spécifier l'aspect de tel ou tel mot, paragraphe, etc. Par exemple sur un site composé de dizaines de pages HTML, pour la modification de la taille de la police de caractères (...), le gestionnaire du site devait intervenir sur chaque page... A chaque modification d'une page, il fallait également repositionner les balises propres à la mise en forme. C'était devenu un bazar pas possible !

Le W3C a répondu à ce problème avec les **CSS** (Cascading Style Sheets) ou feuilles de style en cascades. Une feuille de style est un fichier contenant tous les réglages de mise en forme de la page : les couleurs, la police, l'image de fond, le positionnement des éléments, etc. Lorsque l'on voudra modifier un élément concernant l'aspect des pages, par exemple la couleur des titres, il suffira de le faire dans la feuille de style et cette modification sera prise en compte par toutes les pages HTML. Alors l'objectif des CSS est de permettre la mise en forme de la page HTML en dehors du document HTML.

Le fichier HTML est composé du contenu de la page Web, la feuille CSS de l'**aspect** ou la **mise en page**.

Les avantages de l'utilisation des CSS sont nombreux :

- il y a séparation entre le contenu et la présentation des documents ;
- la rédaction du contenu des pages se fait sans se soucier de la mise en forme ;
- le code HTML est réduit en taille et est donc moins complexe ;
- la présentation est uniformisée dans le cas de multiples pages HTML.



Il est impensable de nos jours de ne pas savoir, un tant soit peu, manipuler les CSS quand on travaille sur du HTML.

S'amuser avec CSS

C'est impressionnant comment un fichier CSS peut changer une page web (et même l'expérience utilisateur). Vous ne me croyez pas? Visitez <http://www.csszengarden.com/> et regardez la même page HTML avec différents feuilles de style. Dommage que les liens vers quelques ressources ne fonctionnent pas (la programmation web n'est pas que le front-end, il faut aussi un back-end, mais plus tard!).

Inspectez le code d'un élément dans une page (Clic-droit sur un élément -> Inspect Element ou View Selection Source). Sous Styles vous voyez des descriptions du type

```
.design-selection .design-name, .zen-resources a, {  
  margin-bottom: 0.25em;  
  font-style: normal;  
  text-transform: uppercase;  
}
```

Ça c'est une syntaxe CSS.

Application d'un style à une page HTML

Par « application » d'un style on entend utilisation de paramètres CSS. Il y a plusieurs façons à le faire.

Par exemple nous pouvons ajouter CSS dans une balise HTML (style qu'on appelle *Inline*), comme `<h1 style="color:blue;">A Blue Heading</h1>`, là on applique un style dans un seule élément à la fois (qui n'est pas très intéressante). Si on a une seule page HTML nous pouvons définir tous les styles au début de notre fichier (style qu'on appelle *Internal*), en utilisant la balise `<style>` dans la section `<head>`.

Mais l'application la plus courante est de créer un fichier .css et le lier dans le HTML, dans le `<head></head>` : `<link rel="stylesheet" href="style.css" />`.

Cette style est souvent appelé *External* est c'est intéressant car on découple complètement le contenu (HTML) et le style (CSS). En plus nous pouvons réutiliser le CSS pour plusieurs pages HTML.

```
balise1
{
  propriete1: valeur1;
  propriete2: valeur2;
  propriete3: valeur3;
}
```

```
balise2
{
  propriete1: valeur1;
  propriete2: valeur2;
  propriete3: valeur3;
  propriete4: valeur4;
}
```

Format CSS

Dans la syntaxe CSS on a toujours trois éléments :

- les sélecteurs (à quels noeuds vont s'appliquer les styles) - souvent les types des balises sur lesquelles nous allons appliquer un style (e.x., les titres `<h1>`) ;
- des propriétés CSS que nous allons changer, comme le couleur, police, position, ... ;
- et les valeurs de ses propriétés.

Voici quelques exemples. Ici ...

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

... on met en bleu chaque contenu de balise `<h1>` `</h1>`.

Et dans le suivant on applique le même style à plusieurs sélecteurs.

```
h1, h3
{
  color : blue;
}
```

Introduction

HTML

Les bases

Entete et corps

Ceci est le texte de niveau 3.

Les balises

Introduction

HTML

Les bases

Entete et corps

Ceci est le texte de niveau 3.

Les balises

Alors nous pouvons sélectionner des balises. Mais c'est aussi possible de sélectionner une phrase, un élément donné ou une famille d'éléments. On utilise les balises `<div>` et `` qui sont non-visibility mais des balises sélectionnables. Et on leur associe les attributs `id` ou `class`. L'attribut `id` doit être unique pour l'ensemble de la page web, l'attribut `class` nous pouvons le considérer comme un "tag" ou label qu'on associe nous à un (ou plusieurs) endroits du texte. Enfaîte, nous pouvons utiliser `id` ou `class` avec d'autres balises aussi (pas seulement `div` et `span`, mais par images, listes, etc). Après, on peut sélectionner une classe `nomclasse` avec le sélecteur `.nomclasse`, et un identifiant unique `nomid` avec le sélecteur `#nomid`

Sélecteurs

Il y a donc 2 choses importantes à maitrise dans CSS : les sélecteurs (à quels noeuds vont s'appliquer les styles) et les effets de style (une très longue liste de variables qui peuvent prendre des valeurs, souvent parmi une liste fermée ...)

Il y a 4 sélecteurs importants à connaître sur le bout des doigts :

<code>div {background:red}</code>	Nom de la balise. Ici toutes les balises <code><div></code> auront un fond rouge;
<code>h1, div {background:orange}</code>	Idem mais pour plusieurs balises ;
<code>.gros {background:green}</code>	Classe. Toutes les balises qui auront l'attribut <code>classe="gros"</code> auront un fond vert ;
<code>#lecadre {background:blue}</code>	ID. L'unique balise qui aura l'attribut <code>id="lecadre"</code> aura un fond bleu ;
<code>td div {background:yellow}</code>	Sélecteur contextuel (avec espaces entre le nom des balises). Ici toutes les balises <code><div></code> qui seront à l'intérieur d'une balise <code><td></code> auront un fond jaune.

Voici un exemple qui utilise ses sélecteurs. D'abord notre fichier HTML :

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="style.css" />
</head>
<body>
  <div id="container">
    <div id="card">
      
      <p>My Dogs</p>
    <div>
      
      
      
    </div>
  </div>
</div>
</body>
</html>
```

Et voici notre fichier CSS qui détermine des attribues des éléments uniques (container et card), le contenu entre les balises <div> et <p>, ainsi que les éléments qu'on a associé avec les classes img-profile et dog-icon :

```
#container {
  width: 100%;
  padding: 5px;
}
#card {
  max-width: 320px;
  padding: 5px;
  border-radius: 8px;
  box-shadow: 0px 1px 6px 0px #00000033;
}
div {
  text-align: center;
  margin: auto;
}
p {
  font-color: grey;
  font-size: 32px;
  font-family: sans-serif;
  font-weight: bold;
}
.img-profile {
  max-width: 100%;
  height: auto;
}
.dog-icon {
  height: 50px;
  padding: 20px;
  cursor: pointer;
}
```

Après il y a aussi des sélecteurs avancés, par exemple :

- * : sélecteur universel ;
- A B** : les balises B contenues dans une balise A, penser en terme d'arbre ;
- A + B** : la première balise B qui suit une balise A ;
- A[attribut]** : une balise qui possède un attribut ;
- A[attribut="Valeur"]** : une balise, un attribut et une valeur exacte. On a la possibilité d'écrire attribut* et alors Valeur doit être un morceau de la vraie valeur (expression réduite) ;

Le détail officiel : [ici](#) (la liste est loooooongue ...)

On remarque que certains de ces sélecteurs sont liés à la notion de structure d'arbre.

Cascade

Mais la situation est un peu plus complexe. En choisissant un style pour une balise, on l'affecte à toutes ses sous-balises aussitôt (d'où le Cascade dans le nom de CSS).

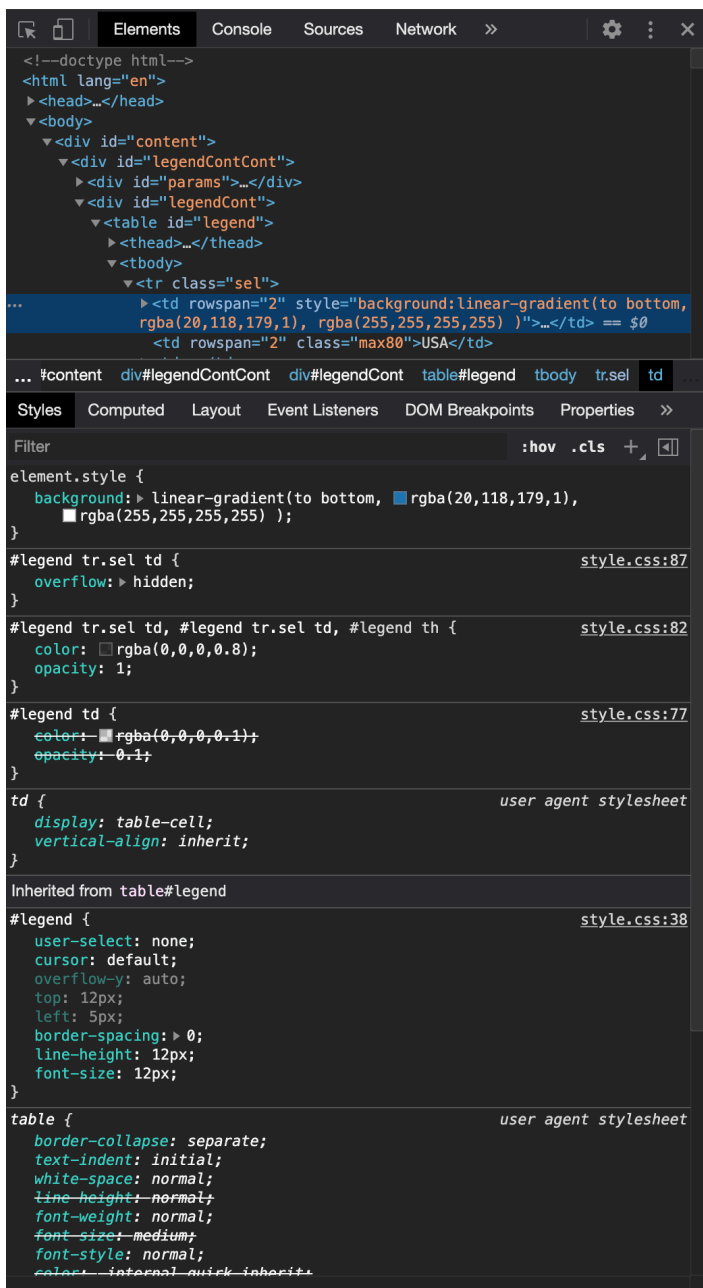
Par exemple disons on définit en CSS:

```
div {background:red}
.gros {background:green}
```

mais si on a une balise `<div class="gros">...</div>` elle aura quelle couleur de fond ? En fait c'est la règle la plus spécifique qui prend le dessus !

C.à.d d'abord: `id`, suivi par `class`, par sélecteur contextuel (ex : `div p`), nom de la balise, et à la fin héritage.

La ça devient difficile à suivre même si on comprends l'ordre ... Mais gardons notre calme ! Quand on clique avec le bouton droit sur un élément (dans Chrome ou firefox) on peut demander à inspecter cet élément. Apparaît alors l'outil suivant :



On voit en haut que la cellule de table TD que j'inspecte est comprise dans un `<TR>` qui a la classe "sel" lui-même compris dans un `<tbody>` dans une table dans un div, dans un div qui a comme `id=legendCont` qui est dans un autre div, etc.

En dessous nous voyons tous les styles qui s'applique à ce `<TD>` et ces styles sont classés du plus spécifique (en haut) au plus générique (en bas).

Ainsi on voit que le `background` est un dégradé linéaire de couleur car c'est son `element.style`. Ca veut dire que la balise `<TD>` avait carrément `style="..."` comme attribut (d'ailleurs on le voit au dessus dans l'arbre). 2 lignes en dessous on voit que la couleur d'écriture est du noir pas complètement opaque car à la ligne 82 de `style.css` les `"#legend tr.sel td"` ont ce style (= les `<TD>` qui sont dans un `<TR>` de classe `sel` qui sont dans la balise qui a l'identifiant `legend`). Tiens ça se lit à l'envers. Or si on regarde l'arbre HTML ce TD est bien dans une balise `<TR>` qui a comme classe `sel` qui est elle-même dans un truc qui a comme `id "legend"` (en fait ici c'est la table).

(7 pages de contenu, 4 référence)

Enfin bref on peut faire confiance à Chrome, s'il affiche ce style c'est qu'il s'applique bien à ce qu'on a choisi. D'ailleurs quand on rajoute un style et que ça n'a aucun effet c'est la première chose qu'on vérifie=> l'inspecteur affiche t-il bien ce qu'on vient de rajouter.

Ensuite à la ligne du dessous on voit qu'un autre style aurait pu s'appliquer (#legend td) MAIS que son color (un noir très transparent) est barré. Cela veut dire qu'un style plus spécifique (celui dont on vient de parler) est jugé prioritaire.

On voit en dessous des styles qui s'appliquent dans "user agent stylesheet". Ça veut dire que c'est des styles définis par défaut dans le navigateur web !

On voit aussi des styles qui sont dit "hérités". En effet les styles que j'applique à la table (#legend) sont donnés en héritage aux lignes et aux cases (ah bon ?)

Enfin bref il y a tellement de règles CSS qu'il est très difficile de faire bon du premier coup. Il faut toujours avoir cet inspecteur sous les yeux, ajouter une ligne de CSS, reloader la page, et voir à quel niveau la nouvelle règle s'insère. Quand on a une surprise, on demande à Google...

Tout comme l'HTML, le W3C propose un service de validation à l'adresse : <http://jigsaw.w3.org/css-validator/>

Tous les navigateurs ne supportent pas toutes les fonctionnalités du CSS. <https://www.caniuse.com/> pour plus de détails.

Un peu d'interaction et spécialisation

Il y a aussi le sélecteur : suivi par une des pseudo-classes déjà définis, qui mérite d'être cité. Il permet de faire des choses un peu interactives. Par exemple :

```
div {background:white}
div:visited {background:gray}
div:hover {background:red}
div:focus {background:green}
```

donne une couleur blanche aux balises div MAIS quand on a déjà visité un lien il devient gris, quand on passe avec la souris par dessus cette balise elle devient rouge, etc. Le sélecteur : nous permet de mettre une condition. Plus sur les pseudo-classes [ici](#) .

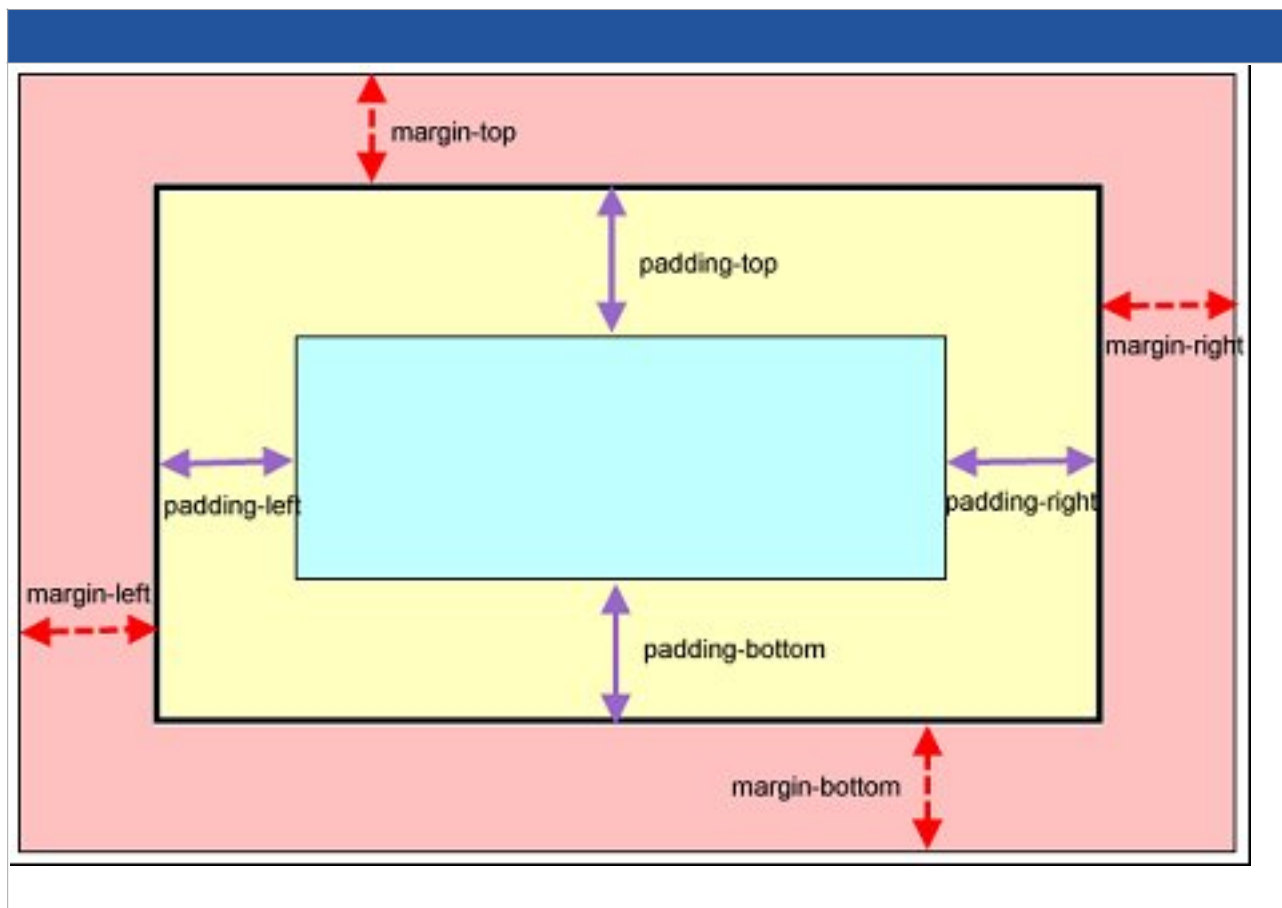
Nous pouvons aussi utiliser le sélecteur suivi par :: pour accéder à une sous-partie d'une balise, ou à une balise lors d'une activité (aspect dynamique). On les appelle souvent pseudo-elements. Plus sur les pseudo-elements [ici](#)

```
p::first-letter : première lettre de l'élément
p::first-line : première ligne de l'élément
p::before : insérer un contenu avant l'élément
p::after : insérer un contenu après l'élément
```

Notions de Marge et Positionnement

Pour les balises blocs (par opposition à inline), comme les div, les sections, vous pouvez préciser les attribues : `width` et `height`. Vous pouvez aussi définir un min et max quand on donne en relatif avec `min-width`, `max-height`, etc. Ex : `p { width: 50%; min-width: 400px;}`

Voici les marges que nous pouvons manipuler (margin est pour l'espace autour, padding à l'intérieur):



Nous pouvons donner 1 entier comme paramètre (qui s'applique à toutes les directions) ou 4 entiers différentes comme possibilité de préciser les 4 directions. Pour centrer il faut donner un `width + margin`.

Toujours pour les balises blocs, on a trois positions `position`:

`absolute` : position par rapport à un endroit sur la page

`fixed` : pareil mais ne bouge pas dans défilement fenêtre, possibilité de superposition et on choisit l'ordre avec `z-index`

`relative` : position par rapport à la position qu'il devrait avoir normalement.

Et ensuite on ajuste avec `float` : `{top, left, right, bottom}`.

Plus des détails pour la positionnement [ici](#).

À partir d'ici une référence des styles.

Propriétés de mise en forme du texte

font-family	police1, police2, police3, serif, sans-serif, monospace	Nom de police. Les polices les plus courantes dans les navigateurs sont Arial, Arial Black, Comic Sans MS, Courier New, Georgia, Impact, Times New Roman, Trebuchet MS, Verdana.
@font-face	Nom et source de la police	Police personnalisée
font-size	16px, 1.2em	Taille du texte en pixels (16px), ou en taille relative (1.2em). On peut aussi utiliser les valeurs xx-small (minuscule), x-small (très petit), small (petit), medium (moyen), large (grand), x-large (très grand), xx-large (très très grand).
font-weight font-style	bold, normal italic, oblique, normal	Gras Italique
text-decoration	underline, overline, line-through, blink, none	Soulignement, ligne au-dessus, barré ou clignotant
font-variant text-transform font	small-caps, normal capitalize, lowercase, uppercase -	Petites capitales Capitales Super propriété de police. Combine : font-weight, font-style, font-size, font-variant, font-family.
text-align vertical-align	left, center, right, justify baseline, middle, sub, super, top, bottom	Alignement horizontal Alignement vertical (cellules de tableau ou éléments inline-block uniquement)
line-height text-indent	18px, 120%, normal... 25px	Hauteur de ligne Alinéa
white-space word-wrap text-shadow	pre, nowrap, normal break-word, normal 5px 5px 2px blue (horizontale, verticale, fondu, couleur)	Césure Césure forcée Ombre de texte

Propriétés de couleur et de fond

color	<i>nom ou rgb(rouge,vert,bleu)</i>	Couleur du texte : à choisir parmi white, silver, gray, black, red, maroon, lime, green, yellow, olive, blue, navy, fuchsia, purple, aqua, teal.
background-color background-image background-attachment background-repeat background-position background	<i>Identique à color</i> url('image.png') fixed, scroll repeat-x, repeat-y, no-repeat, repeat (x y), top, center, bottom, left, right -	Couleur de fond Image de fond Fond fixe Répétition du fond Position du fond Super propriété du fond. Combine : background-image, background-repeat, background-attachment, background-position
opacity	0.5	Transparence

Propriétés des listes

list-style-type	disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, none	Type de Liste
list-style-position list-style-image list-style	inside, outside url('puce.png') -	Position en retrait Puce personnalisée Super-propriété de liste. Combine list-style-type, list-style-position, list-style-image .

Propriétés des tableaux

border-collapse empty-cells caption-side margin	collapse, separate hide, show bottom, top auto	Fusion des bordures Affichage des cellules vides Position du titre du tableau Tableau centré horizontalement
--	---	---

Propriétés des boîtes

width height min-width max-width min-height max-height margin-top margin-left margin-right margin-bottom margin	150px, 80%... 150px, 80%... 150px, 80%... 150px, 80%... 150px, 80%... 150px, 80%... 23px 23px 23px 23px 23px 5px 23px 5px <i>(haut, droite, bas, gauche)</i>	Largeur Hauteur Largeur minimale Largeur maximale Hauteur minimale Hauteur maximale Marge en haut Marge à gauche Marge à droite Marge en bas Super-propriété de marge. Combine : margin-top, margin-right, margin-bottom, margin-left.
padding-left padding-right padding-bottom padding-top padding	23px 23px 23px 23px 23px 5px 23px 5px <i>(haut, droite, bas, gauche)</i>	Marge intérieure à gauche Marge intérieure à droite Marge intérieure en bas Marge intérieure en haut Super-propriété de marge intérieure. Combine : padding-top, padding-right, padding-bottom, padding-left .
border-width border-color	3px <i>nom</i> , rgb(rouge,vert,bleu), rgba(rouge,vert,bleu,transparence), #CF1A20...	Épaisseur de bordure Couleur de bordure
border-style	solid, dotted, dashed, double, groove, ridge, inset, outset	Type de bordure
border	3px solid black	Super-propriété de bordure. Combine border-width, border-color, border-style . Existe aussi en version border-top, border-right, border-bottom, border-left .
border-radius box-shadow	5px 6px 6px 0px black <i>(horizontale, verticale, fondu, couleur)</i>	Bordure arrondie Ombre de boîte

Propriétés de positionnement et d'affichage

display	block, inline, inline-block, table, table-cell, none...	Type d'élément (block, inline, inline-block, non e...)
visibility	visible, hidden	Visibilité
clip	rect (0px, 60px, 30px, 0px) <i>rect (haut, droite, bas, gauche)</i>	Affichage d'une partie de l'élément
overflow	auto, scroll, visible, hidden	Comportement en cas de dépassement
float	left, right, none	Flottant
clear	left, right, both, none	Arrêt d'un flottant
position	relative, absolute, static	Positionnement
top	20px	Position par rapport au haut
bottom	20px	Position par rapport au bas
left	20px	Position par rapport à la gauche
right	20px	Position par rapport à la droite
z-index	10	Ordre d'affichage en cas de superposition. La plus grande valeur est affichée par-dessus les autres.

Autres propriétés

cursor	crosshair, default, help, move, pointer, progress, text, wait, e-resize, ne-resize, auto...	Curseur de souris
---------------	---	-------------------

Liens utiles :

1. Dans le site <https://fr.w3docs.com/> vous pouvez trouver des tutos en HTML et CSS en Français, y compris des petites exercices.
2. En Anglais, les tutos de w3d <https://www.w3schools.com/css/> qui gèrent des thématiques comme les tooltips, les animations, la customization des buttons (quoi? buttons? oui en HTML), etc. Avec des exemples, des quizzes et des exercices.
3. Le cours de Pierre Giraud couvre aussi le css <https://www.pierre-giraud.com/html-css-apprendre-coder-cours/>
4. L'internet ... vraiment pour tous faire ...