

Listes

- Les propriétés CSS des listes permettent de choisir les marqueurs pour les ``, `` et ``
 - **list-style-type** a **plusieurs valeurs possibles**, les plus courantes sont : none, disc, circle, square, decimal, upper-roman, lower-roman
 - **list-style-position** : inside ou outside
 - **list-style-image** : pour définir une image qui remplace les marqueurs des listes à puce
exemple : `ul { list-style-image:url('image.gif'); }`
- **list-style** est un raccourci des 3 propriétés

Tableaux

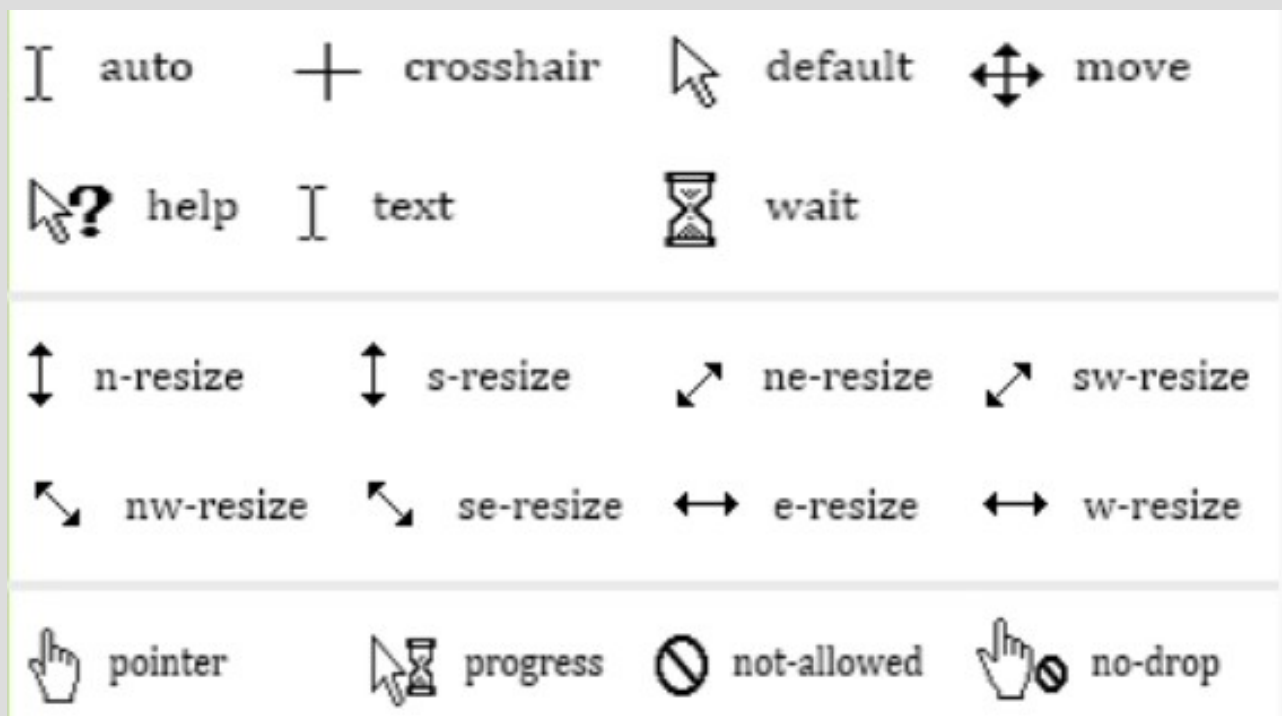
- **border-collapse** spécifie si les bordures (tableau, ligne, cellule) sont séparées ou non (valeurs : **separate** ou **collapse**). Si elles sont séparées (valeur par défaut), **border-spacing** permet de changer l'espace
- **caption-side** spécifie la position de la légende du tableau (soit **top** soit **bottom**).
- **empty-cells** spécifie s'il faut afficher les bordure et fond pour des cellules vides (**show** ou **hide**)
- **table-layout** définit la largeur (**auto** ou **fixed**)

Liens

- Les **liens** peuvent être stylés avec plusieurs propriétés CSS (font-family, background, color, text-decoration,...)
- Mais on peut donner des valeurs différentes selon l'état du lien (avec les **pseudo-classes**) :
 - **a:link** – style d'un lien pas encore visité (normal)
 - **a:visited** – style d'un lien déjà visité
 - **a:hover** – style d'un lien survolé par la souris
 - **a:active** – style d'un lien sur lequel on clique

Curseur

- Le curseur de la souris peut être modifié et il y a toute une **liste de valeurs possibles**
- Pour le définir, on a la propriété **cursor**



Pseudo-classes (CSS1/CSS 2.1)

- Les **pseudo-classes** s'associent aux sélecteurs ordinaires pour devenir des sélecteurs plus spécifiques. Elles commencent par un **simple deux-points**
- Pour les interactions souris et clavier :
:link :visited :hover :active :focus
- Et encore :
:first-child :lang

Pseudo-éléments (CSS1/CSS 2.1)

- En CSS, les **pseudo-éléments** permettent de sélectionner une partie d'un élément. Pour les distinguer des pseudo-classes, ils commencent par un **double deux-points**

::first-line

::first-letter

::before

::after

Sélecteurs (CSS3)

- Élément dont la ***valeur de l'attribut*** commence par ...

p[title^="test"] {font-style: italic}

- Élément dont la ***valeur de l'attribut*** se termine par ...

input[name\$="type"] {width:1em}

- Élément dont la ***valeur de l'attribut*** contient ...

div[class*="col"] {border:none}

- Élément qui **suit directement** un autre élément

div~p {color:blueviolet}

Pseudo-classes (CSS3)

- Pour désigner le $n^{\text{ième}}$ **enfant** de son parent
:nth-child(n)
- Pour désigner le $n^{\text{ième}}$ enfant de son parent **en partant de la fin**
:nth-last-child(n)
- Pour désigner le **premier** ou le **dernier** enfant
:first-child :last-child

Pseudo-classes (CSS3)

- Élément qui a **n** frères du **même type avant lui**
:nth-of-type(n)
- Élément qui a n-1 frères du même type **après lui**
:nth-last-of-type(n)
- Élément qui est le **premier** ou le **dernier** enfant de son type parmi les enfants de son parent
:first-of-type :last-of-type
- Élément qui est **enfant unique** ou **seul enfant d'un type donné** de son parent ou **aucun enfant**
:only-child :only-of-type :empty

Pseudo-classes (CSS3)

- Pour trouver la **racine** du document (<html>)
:root
- Pour désigner l'élément **cible d'une ancre** (#) et modifier l'aspect d'une partie de la page
:target
- Pour un **élément qui n'est pas représenté** par l'expression
:not(expression)

@-règles

- Ce sont des **règles qui étendent la syntaxe**, elles doivent être **écrites en premier** avant les commentaires. Quelques exemples :

@charset 'UTF-8'; pour définir l'encodage des CSS

@import url('style.css'); charger une CSS externe

@font-face pour définir une police externe

@page pour définir la mise en page pour imprimer

@media pour cibler un type de média

- CSS3 en a ajouté : **@viewport**, **@namespace**, **@supports**, **@counter-style**,...