**CSS (Cascading Style Sheets)**

Table des matières

[Liaison à une page HTML 6](#_Toc113106608)

[Nécessaire dans toutes les feuilles de style (la base) !! 6](#_Toc113106609)

[Sélecteurs 6](#_Toc113106610)

[Pseudos-classes 7](#_Toc113106611)

[Etat de l’élément : *a:link* , *a:visited* , *a:hover* , *a:active* 7](#_Toc113106612)

[Pseudos-éléments 7](#_Toc113106613)

[Premiers caractères  *::first-line , ::first-letter* 7](#_Toc113106614)

[Insertion de contenu automatique *::before , ::after* 7](#_Toc113106615)

[Style des marqueurs de liste  *::marker* 8](#_Toc113106616)

[Style de la sélection du curseur *::selection* 8](#_Toc113106617)

[Textes 8](#_Toc113106618)

[Couleur de texte : *color* 8](#_Toc113106619)

[Alignement horizontal : *text-align, text-align-last* 8](#_Toc113106620)

[Alignement vertical : *vertical-align* 9](#_Toc113106621)

[Ecriture verticale : *writing-mode* 9](#_Toc113106622)

[Ligne de décoration de texte : *text-decoration-line* 9](#_Toc113106623)

[Style de la ligne de décoration : *text-decoration-color* , *text-decoration-style* , *text-decoration-thickness* 9](#_Toc113106624)

[Passage Majuscules/minuscules : *text-transform* 10](#_Toc113106625)

[Ajouter un alinéa : *text-indent* 10](#_Toc113106626)

[Espacement des caractères : *line-height* , *letter-spacing* , *word-spacing* 10](#_Toc113106627)

[Débordement de texte : *text-overflow* 10](#_Toc113106628)

[Polices 11](#_Toc113106629)

[Choix de la police : *font-family* 11](#_Toc113106630)

[Taille de la police : *font-size* 11](#_Toc113106631)

[Epaisseur/Style de la police : *font-weight , font-style* 11](#_Toc113106632)

[Colonnes multiples dans un texte 11](#_Toc113106633)

[Nombre de colonnes : *column-count* 11](#_Toc113106634)

[Espacement entre les colonnes : *column-gap* 12](#_Toc113106635)

[Largeur des colonnes : *column-width* 12](#_Toc113106636)

[Séparation visible des colonnes : *colomn-rule-style, column-rule-width, column-rule-color* 12](#_Toc113106637)

[Listes 12](#_Toc113106638)

[Style des marqueurs de listes : *list-style-type* 12](#_Toc113106639)

[Utiliser une image comme marqueur de liste : *list-style-image* 13](#_Toc113106640)

[Tableaux 13](#_Toc113106641)

[Surbrillance d’une ligne du tableau : *tr:hover* 13](#_Toc113106642)

[Tableau rayé : *tr:nth-child(even)* 13](#_Toc113106643)

[Arrière-plans 13](#_Toc113106644)

[Couleur d’arrière-plan : *background-color , opacity* 13](#_Toc113106645)

[Dégradé de couleurs d’arrière-plan : *linear-gradient, radial-gradient, conic-gradient* 14](#_Toc113106646)

[Dégradé style color picker rond (*conic-gradient + border-radius*) 14](#_Toc113106647)

[Image d’arrière-plan : *background-image , background-repeat* 14](#_Toc113106648)

[Superposition d’images d’arrière-plan 15](#_Toc113106649)

[Position d’arrière-plan : *background-position , background-attachment* 15](#_Toc113106650)

[Images 15](#_Toc113106651)

[Image avec taille adaptative 15](#_Toc113106652)

[Centrer une image 15](#_Toc113106653)

[Image en noir et blanc : *filter* 16](#_Toc113106654)

[Inverser une image (gauche <-> droite) : *transform* 16](#_Toc113106655)

[Masque d’image (superposition d’une image dans un logo par exemple) : *mask-image , mask-repeat* 16](#_Toc113106656)

[Masque de forme ronde : *mask-image: radial-gradient* 17](#_Toc113106657)

[Images vectorielles 17](#_Toc113106658)

[Définir la zone d’images vectorielles : *width, height* 17](#_Toc113106659)

[Propriétés communes aux images vectorielles : *fill, stroke, stroke-width* 17](#_Toc113106660)

[Propriétés propres aux images vectorielles rectangle : *rx, ry* 17](#_Toc113106661)

[Dimensions 18](#_Toc113106662)

[Hauteur et largeur : *height, width* 18](#_Toc113106663)

[Largeur maximale : *max-width* 18](#_Toc113106664)

[Débordement du texte : *overflow* 18](#_Toc113106665)

[Redimension par l’utilisateur : *resize* 18](#_Toc113106666)

[Bordures 19](#_Toc113106667)

[Style de bordure : *border-style* 19](#_Toc113106668)

[Largeur de la bordure : *border-width* 19](#_Toc113106669)

[Couleur de la bordure : *border-color* 19](#_Toc113106670)

[Arrondi de la bordure : *border-radius* 19](#_Toc113106671)

[Bordure par coté : *border-left, border-right, border-top, border-bottom* 20](#_Toc113106672)

[Marges (extérieur de la bordure) 20](#_Toc113106673)

[Marge générale : *margin* 20](#_Toc113106674)

[Marge par coté : *margin-top, margin-right, margin-bottom, margin-left* 20](#_Toc113106675)

[Marge automatique 20](#_Toc113106676)

[Espacement (intérieur de la bordure) 21](#_Toc113106677)

[Espacement général : *padding* 21](#_Toc113106678)

[Espacement par coté : *padding-top, padding-right, padding-bottom, padding-left* 21](#_Toc113106679)

[Largeur d’élément : *width + padding* 21](#_Toc113106680)

[Prise en compte des marges et padding dans la largeur : *box-sizing* 21](#_Toc113106681)

[Résumé : Modèle de boite 22](#_Toc113106682)

[Contours (extérieur de bordure) 22](#_Toc113106683)

[Style du contour : *outline-style* 23](#_Toc113106684)

[Epaisseur du contour : *outline-width* 23](#_Toc113106685)

[Couleur du contour : *outline-color* 23](#_Toc113106686)

[Raccourci d’écriture : *outline* 23](#_Toc113106687)

[Décalage du contour : *outline-offset* 24](#_Toc113106688)

[Ombres 24](#_Toc113106689)

[Ombre sur un texte : *text-shadow* 24](#_Toc113106690)

[Ombre sur une zone : *box-shadow* 24](#_Toc113106691)

[Flexbox 24](#_Toc113106692)

[Conteneur flexible : *display:flex* 24](#_Toc113106693)

[Disposition horizontale : *justify-content* 25](#_Toc113106694)

[Disposition verticale : *align-items* 25](#_Toc113106695)

[Disposition verticale d’un seul élément : *align-self* 26](#_Toc113106696)

[Direction de l’alignement : *flex-direction* 26](#_Toc113106697)

[Etalement sur plusieurs lignes/colonnes au besoin : *flex-wrap* 26](#_Toc113106698)

[Direction + étalement : *flex-flow* 26](#_Toc113106699)

[Espacement des lignes (quand il y en a plusieurs ) : *align-content* 27](#_Toc113106700)

[Ordre des éléments : *order* 27](#_Toc113106701)

[Indice d’occupation de la place disponible pour les contenus : *flex-grow* 27](#_Toc113106702)

[Indice de rétrécissement pour les contenus : *flex-shrink* 28](#_Toc113106703)

[Indice de taille initiale pour les contenus: *flex-basis* 28](#_Toc113106704)

[Grilles 28](#_Toc113106705)

[Créer une grille : *display :grid* 28](#_Toc113106706)

[Nombre de lignes/colonnes de la grille : *grid-template-columns, grid-template-rows, grid-template* 29](#_Toc113106707)

[Unité de compensation d’espace libre : *fr* 29](#_Toc113106708)

[Situer un élément dans la grille : *grid-column-start, grid-column-end, grid-column, grid-row-start, grid-row-end, grid-row, grid-area* 29](#_Toc113106709)

[Gestion des écarts entre 2 lignes/colonnes : *column-gap, row-gap* 30](#_Toc113106710)

[Ordre des éléments : *order* 30](#_Toc113106711)

[Valeur d’affichage / Visibilité : *display* , *visibility* 30](#_Toc113106712)

[Elément non visible 30](#_Toc113106713)

[Elément non visible avec conservation de l’espace 30](#_Toc113106714)

[Elément de type « bloc » sur une ligne 31](#_Toc113106715)

[Elément de type « inline » en bloc 31](#_Toc113106716)

[Elément à la fois de type « inline » et « bloc » 31](#_Toc113106717)

[Opacité d’un élément : *opacity* 31](#_Toc113106718)

[Position d’un élément (statique, mouvante…) 32](#_Toc113106719)

[Définir une position : *position* 32](#_Toc113106720)

[Chevauchement d’éléments : *z-index* 32](#_Toc113106721)

[Transformations 2D (mouvements) : *transform* 32](#_Toc113106722)

[Translation : *translate()* 32](#_Toc113106723)

[Rotation : *rotate()* 32](#_Toc113106724)

[Agrandissement/diminution : *scale() , scaleX(), scaleY()* 33](#_Toc113106725)

[Inclinaison : *skew(), skewX(), skewY()* 33](#_Toc113106726)

[Ensemble de transformations : *matrix()* 33](#_Toc113106727)

[Transformations 3D (mouvements et profondeur) : *transform* 33](#_Toc113106728)

[Rotation 3D : *rotateX(), rotateY(), rotateZ()* 33](#_Toc113106729)

[Perspective : *perspective* 34](#_Toc113106730)

[Origine de la perspective : *perspective-origin* 34](#_Toc113106731)

[Visibilité de la face arrière en perspective : *backface-visibility* 34](#_Toc113106732)

[Transitions 34](#_Toc113106733)

[Mise en place d’une transition : *transition* 34](#_Toc113106734)

[Faire plusieurs transitions sur un même élément 35](#_Toc113106735)

[Evolution de la vitesse de transition : *transition-timing-function* 35](#_Toc113106736)

[Ajouter un délai de transition : *transition-delay* 35](#_Toc113106737)

[Transition de transformation 2D ou 3D 35](#_Toc113106738)

[Animations 36](#_Toc113106739)

[Mise en place d’un animation : *@keyframes , animation-name, animation-duration* 36](#_Toc113106740)

[Animation progressive (enchaînement de différentes valeurs) 36](#_Toc113106741)

[Animations multiples 36](#_Toc113106742)

[Délai d’animation : *animation-delay* 37](#_Toc113106743)

[Répétition de l’animation : *animation-iteration-count* 37](#_Toc113106744)

[Sens de l’animation : *animation-direction* 37](#_Toc113106745)

[Evolution de la vitesse d’animation : *animation-timing-function* 37](#_Toc113106746)

[Requêtes Média 38](#_Toc113106747)

[Selon la taille de l’appareil : *@media screen and max-width/min-width* 38](#_Toc113106748)

[Selon l’orientation de l’appareil : *@media screen and orientation* 38](#_Toc113106749)

[Variables 39](#_Toc113106750)

[Utiliser des variables : *:root , var()* 39](#_Toc113106751)

[Compteurs 39](#_Toc113106752)

[Numérotation automatique de titres : *conter-reset* , *counter-increment* , *content* 39](#_Toc113106753)

[Priorité 40](#_Toc113106754)

[Instaurer une priorité : *!important* 40](#_Toc113106755)

[Fonction mathématiques 40](#_Toc113106756)

[Recherche de dimensions par fonctions mathématiques : *calc()*, *max()*, *min()* 40](#_Toc113106757)

[Curseur de souris 40](#_Toc113106758)

[Styles de curseurs : *cursor* 40](#_Toc113106759)

# Liaison à une page HTML

<head>

        <meta charset="utf-8"/>

        <title>Trop cool le HTML</title>

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

</head>

# Nécessaire dans toutes les feuilles de style (la base) !!

html{

    width: 100%;

    height: 100%;

  }

body{

    min-width: 100%;

    min-height: 100%;

    margin: 0;

    padding: 0;

  }

# Sélecteurs

p,h1,h2{         /\* sélecteurs de plusieurs éléments-balises \*/

}

#id{             /\* sélecteur d'identifiants \*/

}

.classe{         /\* sélecteur de classes \*/

}

p.classe{        /\* Union de sélecteurs (tous les paragraphes ayant la classe "classe" ici \*/

}

# Pseudos-classes

## Etat de l’élément : *a:link* , *a:visited* , *a:hover* , *a:active*

a:link{

    /\* Lien normal non visité \*/

}

a:visited{

    /\* Lien visité \*/

}

a:hover{

    /\* Lien survolé par la souris \*/

}

a:active{

    /\* Lien en train d'être cliqué \*/

}

**Remarque**: Il est important de conserver cet ordre pour les sélecteurs de lien

# Pseudos-éléments

## Premiers caractères *::first-line , ::first-letter*

p::first-line{

}

p::first-letter{

}

## Insertion de contenu automatique *::before , ::after*

p::before{                /\* insertion avant chaque paragraphe \*/

    content : url(‘image.png’);

}

p::after{           /\* insertion après chaque paragraphe \*/

    content : url(‘image.png’);

}

## Style des marqueurs de liste  *::marker*

::marker{            /\* Tous les marqueurs de listes sont rouges \*/

    color :red;

}

## Style de la sélection du curseur *::selection*

::selection{            /\* La sélection du curseur est bleue avec un fond orange \*/

    color :blue;

    background-color: orange;

}

# Textes

## Couleur de texte : *color*

p{

    color: blueviolet;

}

## Alignement horizontal : *text-align, text-align-last*

p{

    text-align: left;     /\* le texte part de la gauche de la page \*/

    text-align: center;   /\* le texte part du centre de la page \*/

    text-align: right;    /\* le texte part de la droite de la page \*/

    text-align: justify;  /\* le texte est au format "justifié" \*/

    text-align-last: right;   /\*la dernière ligne du texte est à droite \*/

}

## Alignement vertical : *vertical-align*

 p{

    vertical-align: baseline; /\* alignement classique \*/

    vertical-align: text-top; /\* alignement au-dessus du texte \*/

    vertical-align: text-bottom; /\* alignement en dessous du texte \*/

}

## Une image contenant texte Description générée automatiquementEcriture verticale : *writing-mode*

p{

    writing-mode: vertical-rl;

}

## Ligne de décoration de texte : *text-decoration-line*

p{

Une image contenant texte

Description générée automatiquement    text-decoration: overline;

    text-decoration: underline;

    text-decoration: overline underline;

    text-decoration: line-through;

}

a{

    text-decoration: none;      /\* Utilisé pour supprimer le soulignement d'un lien \*/

}

## Style de la ligne de décoration : *text-decoration-color* , *text-decoration-style* , *text-decoration-thickness*

p{

    text-decoration-color: yellow;       /\* Change la couleur de la ligne \*/

    text-decoration-style: dotted;        /\* Change le style de la ligne \*/

    text-decoration-thickness: 5px;       /\* Change l'épaisseur de la ligne \*/

}

## Passage Majuscules/minuscules : *text-transform*

p{

    text-transform: uppercase;      /\* Tout en majuscules \*/

    text-transform: lowercase;      /\* Tout en minuscules \*/

    text-transform: capitalize;     /\* La première lettre de chaque mot en majuscules \*/

}

## Ajouter un alinéa : *text-indent*

p{

    text-indent: 50px;

}

## Espacement des caractères : *line-height* , *letter-spacing* , *word-spacing*

p{

    line-height: 0.8;       /\* Gère l'espace entre 2 lignes \*/

    letter-spacing: 5px;    /\* Gère l'espace entre 2 lettres \*/

    word-spacing: 15px;     /\* Gère l'espace entre 2 mots \*/

}

## Débordement de texte : *text-overflow*

p{

    text-overflow: ellipsis; /\* Si le texte est trop grand pour le contenu, il y a ... à la fin \*/

}

p:hover{

    text-overflow: visible ; /\* Lors du survol de la souris, le texte entier apparaît \*/

}

# Polices

## Choix de la police : *font-family*

p{

   font-family: "Times New Roman", Times, serif;

   font-family: Arial, Helvetica, sans-serif;

}

Il faut spécifier plusieurs choix pour s’assurer d’être pris en charge par le navigateur : le premier est celui à utiliser en priorité (on finit avec une famille de police plus générale, comme *serif* ou *sans-serif*)

## Taille de la police : *font-size*

p{

   font-size: 40px;

   font-size: 2.5em;

   font-size: 50%;

}

## Epaisseur/Style de la police : *font-weight , font-style*

p{

   font-weight: normal;

   font-weight: bold;         /\* Met en gras \*/

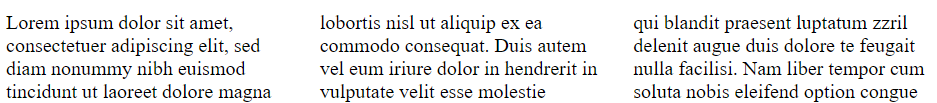
   font-style : normal;

   font-style : italic; /\* Met en italique \*/

}

# Colonnes multiples dans un texte

## Nombre de colonnes : *column-count*

div{

    column-count: 3;

}

## Espacement entre les colonnes : *column-gap*

div{

    column-gap : 40px;

}

## Largeur des colonnes : *column-width*

div{

    column-width: 100px;

}

## Séparation visible des colonnes : *colomn-rule-style, column-rule-width, column-rule-color*

div{

    column-rule-style: solid;

    column-rule-style: dashed;    /\* gère l'aspect de la séparation \*/

    column-rule-width: 1px;       /\* gère l'épaisseur de la séparation \*/

    column-rule-color: red;     /\* gère la couleur de la séparation \*/

}

# Listes

## Style des marqueurs de listes : *list-style-type*

ul{

    list-style-type: circle;

    list-style-type: square;

}

ol{

    list-style-type: upper-roman;

    list-style-type: lower-alpha;

}

## Utiliser une image comme marqueur de liste : *list-style-image*

ul{

    list-style-image: url('image.png');

}

# Tableaux

## Surbrillance d’une ligne du tableau : *tr:hover*

tr:hover{

    background-color: aquamarine;

}

La ligne du tableau survolée par la souris change de couleur

## Tableau rayé : *tr:nth-child(even)*

tr:nth-child(even){

    background-color:  #f2f2f2;

}

Une ligne sur 2 du tableau possède la couleur définie

# Arrière-plans

## Couleur d’arrière-plan : *background-color , opacity*

body{

    background-color: aquamarine;

    background-color: #E58000;

    background-color: rgb(255,140,97);

 background-color: transparent; /\* Le fond de l'élément sera identique au fond du corps de site \*/

opacity : 0.3;

}

**Attention** : il est nécessaire d’avoir une *background-color* pour appliquer une *opacity* (entre 0 🡪 entièrement transparent et 1 🡪 entièrement opaque)

## Dégradé de couleurs d’arrière-plan : *linear-gradient, radial-gradient, conic-gradient*



#grad{

    background-image: linear-gradient(red, yellow);

    background-image: linear-gradient(to right, red, yellow);  /\* de gauche à droite \*/

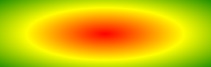
    background-image: linear-gradient(to top, red, yellow);    /\* de bas en haut \*/

  background-image: linear-gradient(to top right, red, yellow);   /\* d'en bas à gauche à l'angle en haut à droite \*/

background-image: linear-gradient(red, yellow, green);



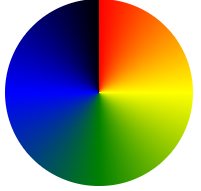
  background-image: radial-gradient(red, yellow, green);



 background-image: conic-gradient(red, yellow, green);

}

## Dégradé style color picker rond (*conic-gradient + border-radius*)

#grad {

    background-image: conic-gradient(red, yellow, green, blue, black);

    border-radius: 50%;

}

## Image d’arrière-plan : *background-image , background-repeat*

body{

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

  background-repeat: repeat-x;

   background-repeat: repeat-y;

background-repeat: no-repeat;

background-position: right top;

background-size : 100px 80px;

   background-size: contain;

}

*Background-repeat* permet de répéter l’image sur la surface de l’écran si elle n’est pas assez grande (repeat-x 🡪 horizontalement , repeat-y 🡪verticalement)

*No-repeat* permet d’afficher l’image en une seule fois.

*Background-size* avec la valeur « contain » permet de redimensionner l’image de fond afin qu’elle soit aussi grande que possible, en restant entièrement visible.

## Superposition d’images d’arrière-plan

body{

    background-image : url(‘image.png’), url(‘anime.gif’);

    background-position : right bottom, left top;

    background-repeat: no-repeat, repeat;

}

La première image indiquée (ici image.png) est celle apparaissant au premier plan

## Position d’arrière-plan : *background-position , background-attachment*

body{

background-position: right top;

   background-attachment: fixed;

   background-attachment: scroll;

}

*Background-position* indique la position de départ de l’image sur la page

*Background-attachment* spécifie si l’image d’arrière-plan doit défiler (scroll) avec la page, ou rester fixe (fixed), c’est-à-dire visible en permanence

# Images

## Image avec taille adaptative

img{

    max-width: 100%;

    height: auto;

}

## Centrer une image

img{

    display : block;

    margin-left: auto;

    margin-right: auto;

    width : 50%;         /\* la largeur peut varier selon la taille de l'image \*/

}

## Image en noir et blanc : *filter*

img{

    filter: grayscale(100%);

}

## Inverser une image (gauche <-> droite) : *transform*

img:hover{

    transform: scaleX(-1);

}

## Masque d’image (superposition d’une image dans un logo par exemple) : *mask-image , mask-repeat*



**+**

**=**



.masque{

    mask-image: url(‘image.png’);

    mask-repeat: no-repeat;

}

<div class="masque">

    <img src="img\_terre.jpg" alt="Cinque Terre" width="600" height="400">

</div>

Il est nécessaire de déclarer un identifiant sur le div contenant l’image, puis c’est dans cet identifiant que l’on choisit la forme du masque avec *mask-image* et de ne pas répéter le masque, sinon le masque apparait plusieurs fois en couvrant la largeur totale de l’image.

## Une image contenant texte, eau, montagne, nature Description générée automatiquementMasque de forme ronde : *mask-image: radial-gradient*

.masque{

    mask-image: radial-gradient(circle, black 50%, rgba(0, 0, 0, 0.5) 50%);

}

# Images vectorielles

## Définir la zone d’images vectorielles : *width, height*

svg{

  width: 200px;

  height: 100px;  /\* la zone mesure 200 sur 100 pixels \*/

}

## Propriétés communes aux images vectorielles : *fill, stroke, stroke-width*

rect{

  fill: red; /\* la couleur de remplissage est rouge \*/

fill: transparent;

fill-opacity: 0.8;

stroke: green; /\* le contour est en vert \*/

  stroke-width: 5px;      /\* le contour à une épaisseur de 5 pixels \*/

}

## Propriétés propres aux images vectorielles rectangle : *rx, ry*

rect{

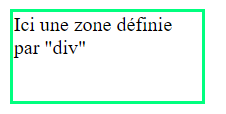
rx: 15;        /\* equivalent à border-radius : 15% pour les côtés haut et bas\*/

  ry:10;         /\* equivalent à border-radius : 10% pour les côtés gauche et droit\*/

}

# Dimensions

## Hauteur et largeur : *height, width*



width

height

div{

    height: 50px;

    width: 100px;

  height: auto;

    width: 50%;

}

“Auto” définit les dimensions par défaut selon le navigateur

Une dimension en pourcentage correspond au taux par rapport à la dimension de l’écran

## Largeur maximale : *max-width*

div{

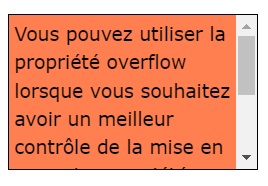
    max-width: 500px;

}

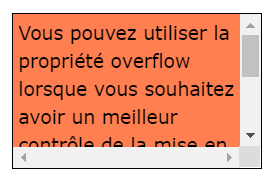
Définir une max-width permet à l’élément de s’adapter à l’écran de l’appareil : si la largeur de l’écran passe en dessous de 500 pixels, l’élément div modifiera sa taille pour un affichage plus responsive, au lieu de générer un curseur défilant horizontal

## Débordement du texte : *overflow*

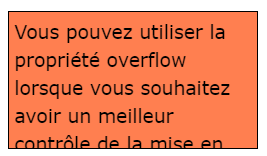
div{



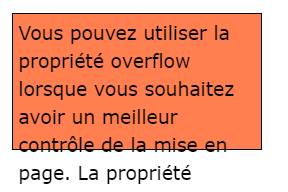
**auto**



**scroll**



**hidden**



**visible**

    overflow : visible;

    overflow: hidden;

    overflow: scroll;

    overflow : auto;

}

## Redimension par l’utilisateur : *resize*

div{

  resize : horizontal;

  resize : vertical;

  resize : both;       /\* à la fois horizontal et vertical \*/

}

# Une image contenant texte Description générée automatiquementBordures

## Style de bordure : *border-style*

h1{

   border-style: dotted;

   border-style: dashed;

   border-style: solid;

   border-style: double;

   border-style: groove;

   border-style: ridge;

   border-style: inset;

   border-style: outset;

   border-style: hidden;

   border-style: none;

}

## Largeur de la bordure : *border-width*

h1{

   border-width: 5px;

border-width: 5px 20px; /\* 5px haut et bas, 20px gauche et droite \*/

   border-width: 5px 20px 15px 10px; /\* 5px haut, 20px droite, 15px bas et 10px gauche \*/

   border-width: medium;

}

## Couleur de la bordure : *border-color*

h1{

   border-color : red;

}

## Arrondi de la bordure : *border-radius*

p{

  border-radius: 5px;

}

## Bordure par coté : *border-left, border-right, border-top, border-bottom*

p{

   border-left : 6px solid firebrick;

   border-right : 6px solid greenyellow;

   border-top : 10px solid seagreen;

   border-bottom: 10px solid blueviolet;

}

# Marges (extérieur de la bordure)

Les marges permettent de créer des espaces autour d’un l’élément

## Marge générale : *margin*

p{

    margin : 10px;

}

## Marge par coté : *margin-top, margin-right, margin-bottom, margin-left*

p{

  margin-top: 5px;

  margin-right: 10px;

  margin-bottom: 20px;

  margin-left: 15px;

  margin: 5px 10px 20px 15px;  /\* on rassemble toutes les données sur une ligne (dans cet ordre) \*/

}

## Marge automatique

div{

    margin : auto;

}

Une marge automatique permet de centrer l’élément horizontalement en mettant la même marge à gauche qu’à droite

# Espacement (intérieur de la bordure)

## Espacement général : *padding*

div{

    padding: 20px;

}

On peut également définir une valeur avec un pourcentage (ex : 50%) , ou une mesure en cm (ex : 2cm)

## Espacement par coté : *padding-top, padding-right, padding-bottom, padding-left*

div{

    padding-top : 10px;

    padding-right: 20px;

    padding-bottom: 15px;

    padding-left: 5px;

    padding: 10px 20px 15px 5px; /\* on rassemble toutes les données sur une ligne(dans cet ordre) \*/

}

## Largeur d’élément : *width + padding*

div{

    width: 300px;

    padding: 15px;

}

La largeur totale de l’élément *div* est 330px (300 + 2\*15)

## Prise en compte des marges et padding dans la largeur : *box-sizing*

div{ div{

    width: 45%; width: 45%;

    padding: 30px; padding: 30px;

    box-sizing: border-box;  background-color: aquamarine;

background-color: aquamarine }

}

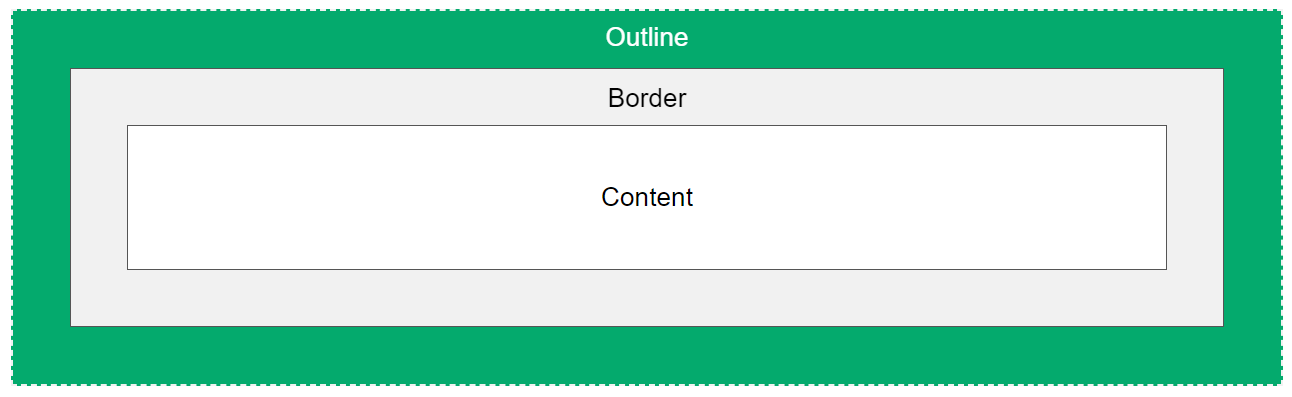
Dans le cas de gauche, le padding de la balise est compris dans sa longueur (la balise occupera donc bien 45% de son contenant)

Dans le cas de droite, le padding de la balise s’ajoute à la largeur paramétrée (la balise occupera 45% de l’espace + son padding)

**Remarque :** cela fonctionne de manière identique pour les marges.

# Une image contenant table Description générée automatiquementRésumé : Modèle de boite

# Contours (extérieur de bordure)

Un contour est une ligne tracée à l’extérieur de la bordure de l’élément

## Une image contenant texte Description générée automatiquementStyle du contour : *outline-style*

div{

    outline-style: dotted;

    outline-style: dashed;

    outline-style: solid;

    outline-style: double;

    outline-style: groove;

    outline-style: ridge;

  outline-style: inset;

    outline-style: outset;

}

## Epaisseur du contour : *outline-width*

div{

    outline-width: thin; /\*Trait fin\*/

    outline-width: medium; /\*Trait moyen\*/

    outline-width: thick; /\*Trait épais\*/

    outline-width: 5px;

}

## Couleur du contour : *outline-color*

div{ div{ div{

    outline-style: solid; outline-style: dotted; outline-style: outset;

    outline-color: red; outline-color: blue; outline-color: green;

} } }

## Raccourci d’écriture : *outline*

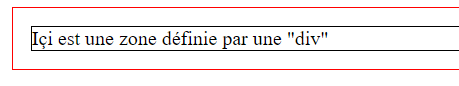
div{

    outline: 5px solid yellow;  /\* epaisseur/style/couleur \*/

    outline : thick dashed white;

}

## Décalage du contour : *outline-offset*

div{

    border: 1px solid black;

    outline: 1px solid red;

    outline-offset: 15px;

}

# Ombres

## Ombre sur un texte : *text-shadow*

h1{

    text-shadow: 5px 2px;     /\* 5 pixels d'ombre horizontale, 2 pixels d'ombre verticale \*/

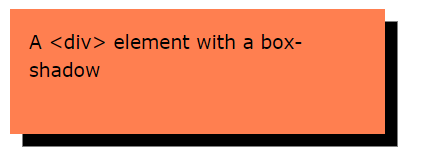
text-shadow: 2px 2px 5px red; /\* Recommandé \*/

    text-shadow: 5px 2px red;     /\* On ajoute une couleur rouge à l'ombre \*/

}

## Ombre sur une zone : *box-shadow*

div {



    box-shadow : 10px 10px;

box-shadow: 10px 10px 5px grey;   /\* Recommandé \*/

}

# Flexbox

Une flexbox est une zone dans laquelle on peut organiser l’agencement de différents éléments dans une seule dimension (horizontalement ou verticalement).

## Conteneur flexible : *display:flex*

.flexcontainer{

    display : flex;                  /\* nécéssaire pour rendre flexible \*/

}

<div class="flex-container"> /\* la flexbox possède ici 3 éléments \*/

     <div id="flex-element1"></div>

     <div id="flex-element2"></div>

     <div id="flex-element3"></div>

</div>

Il est nécessaire d’indiquer cette propriété dans toute flexbox pour s’assurer que son contenu est devenu flexible.

**Remarque :** la disposition des différents éléments de la flexbox est gérée directement depuis son sélecteur (ici la classe *flexcontainer*).

## Disposition horizontale : *justify-content*

.flexcontainer{

      display : flex;

      justify-content: flex-start;       /\* Tous les éléments se collent en partant du début du conteneur \*/

      justify-content: flex-end;         /\* Tous les éléments se collent en partant de la fin du conteneur \*/

      justify-content: center;           /\* Tous les éléments se collent au centre du conteneur \*/

      justify-content: space-between;    /\* Les éléments s'affichent avec un espace égal entre eux (ceux aux extrémités sont collés aux limites du conteneur) \*/

      justify-content: space-around;     /\* Les éléments s'affichent avec un espace égal entre eux, et un espace entre ceux aux extrémités et les limites du conteneur \*/

}

## Disposition verticale : *align-items*

.flexcontainer{

    display : flex;

    align-items: flex-start;    /\* Tous les éléments s'alignent en haut du conteneur \*/

    align-items: flex-end;     /\* Tous les éléments s'alignent en bas du conteneur \*/

    align-items: center;       /\* Tous les éléments s'alignent au centre du conteneur \*/

    align-items: baseline;   /\* Les éléments s'alignent afin que leur première ligne soient toutes au même niveau horizontal \*/

    align-items: stretch;       /\* Les éléments sont verticalement étirés autant que possible dans le conteneur \*/

}

## Disposition verticale d’un seul élément : *align-self*

.flexcontainer{

    display : flex;

    align-items: flex-start;      /\* tous les éléments sont alignés au niveau du début du conteneur...\*/

}

#flex-element2{

    align-self: flex-end;         /\* ...sauf l'élément 2 qui est aligné avec la fin du conteneur \*/

}

## Direction de l’alignement : *flex-direction*

.flexcontainer{

     display : flex;

     flex-direction : row;               /\* alignement horizontal en partant du début du conteneur \*/

     flex-direction : row-reverse;       /\* alignement horizontal en partant de la fin du conteneur \*/

    flex-direction : column;            /\* alignement vertical en partant du début du conteneur \*/

     flex-direction : column-reverse;    /\* alignement vertical en partant de la fin du conteneur \*/

}

## Etalement sur plusieurs lignes/colonnes au besoin : *flex-wrap*

.flexcontainer{

  display : flex;

  flex-wrap: nowrap;        /\* Tous les éléments doivent être alignés sur une seule ligne/colonne \*/

  flex-wrap: wrap;          /\* Les éléments s'étalent sur plusieurs lignes/colonnes si besoin \*/

  flex-wrap: wrap-reverse; /\* Les éléments s'étalent sur plusieurs lignes/colonnes en partant de la fin du conteneur si besoin \*/

}

## Direction + étalement : *flex-flow*

.flexcontainer{

        display : flex;

        flex-flow: column wrap;                    /\* flex-direction + flex-wrap \*/

        flex-flow: row-reverse wrap-reverse;       /\* flex-direction + flex-wrap \*/

}

## Espacement des lignes (quand il y en a plusieurs ) : *align-content*

.flexcontainer{

    display : flex;

     align-content: flex-start;

     align-content: flex-end;

     align-content: center;

     align-content: space-between;

     align-content: space-around;

     align-content: stretch;

}

**Remarque** : *align-content* défini l’espacement entre lignes d’éléments, et pas entre les éléments eux-mêmes (il y a *justify-content* pour ça).

## Ordre des éléments : *order*

.flexcontainer{

        display : flex;

    }

    #flex-element2{

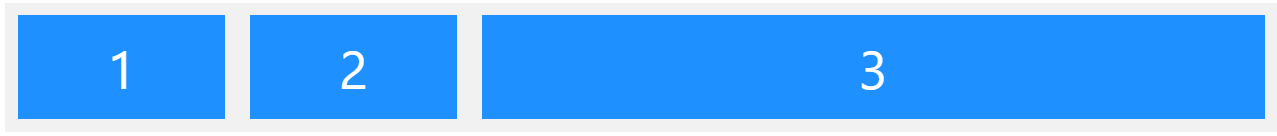
        order: -1;        /\* l'élément 2 est décalé vers le début du conteneur \*/

        order: +1;        /\* l'élément 2 est décalé vers la fin du conteneur\*/

    }

**Remarque** : Tous les éléments ont par défaut l’indice d’ordre 0, donc si on change l’indice d’ordre d’un seul élément, il se retrouvera directement à l’extrémité du début (ordre négatif), ou à l’extrémité de la fin (ordre positif).

## Indice d’occupation de la place disponible pour les contenus : *flex-grow*

.flexcontainer{

    display : flex;

}

#flex-element3{

    flex-grow: 1;  /\* l'élément 3 va chercher à prendre toute la place libre restante dans le conteneur, car les 2 autres éléments possèdent l'indice par défaut qui est 0 \*/

}

## Indice de rétrécissement pour les contenus : *flex-shrink*

.flexcontainer{

    display : flex;

}

#flex-element3{

    flex-shrink: 0;  /\* l'élément 3 rétrécit moins vite que les autres, car son indice est inférieur aux autres qui ont l’indice par défaut (qui est 1 pour cette propriété) \*/

}

## Indice de taille initiale pour les contenus: *flex-basis*

.flexcontainer{

    display : flex;

}

#flex-element3{

    flex-basis: 200px;       /\* l'élément 3 aura une largeur initiale de 200px, avant répartition de l'espace restant (il ne se contente pas de faire la taille de son contenu, contrairement aux autres éléments qui le font par défaut) \*/

}

# Grilles

Une grille repose sur le même principe qu’une flexbox, sauf qu’elle permet de gérer plusieurs dimensions en même temps (horizontalement et verticalement).

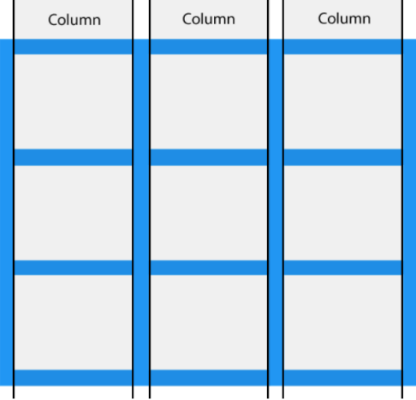
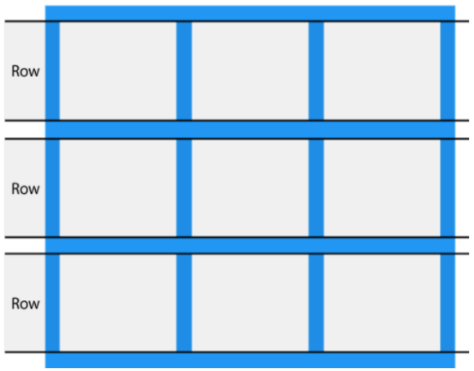
## Créer une grille : *display :grid*

.conteneur-grille{

  display: grid;     /\* on indique que les éléments de cette classe sont des conteneurs de grilles \*/

}

## Nombre de lignes/colonnes de la grille : *grid-template-columns, grid-template-rows, grid-template*



.conteneur-grille{

     display: grid;

     grid-template-columns: 20% 20% 20% 20% 20%;   /\*5 colonnes occupant chacune 20% de la place \*/

     grid-template-rows: 10% 40% 200px 7em;         /\*4 lignes avec des hauteurs diverses \*/

  grid-template: 20% 20% 20% 20% 20% / 10% 40% 200px 7em;  /\* grid-template-columns, grid-template-rows \*/

}

## Unité de compensation d’espace libre : *fr*

.conteneur-grille{

        display: grid;

        grid-template-columns: 100px 1fr 5fr;

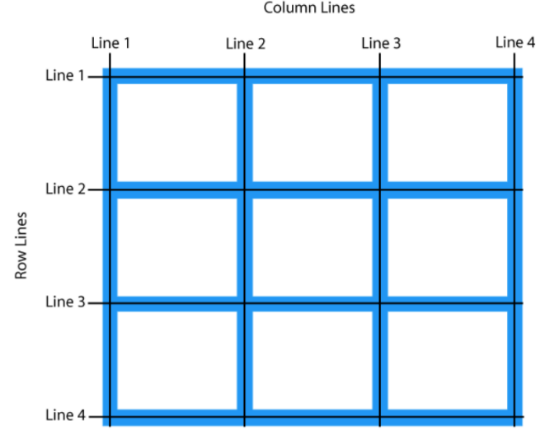
}

Ici, la première colonne mesure 100 pixels, la deuxième occupe 1/6 de l’espace restant et la troisième 5/6 de l’espace restant.

## Situer un élément dans la grille : *grid-column-start, grid-column-end, grid-column, grid-row-start, grid-row-end, grid-row, grid-area*

.conteneur-grille{

     display: grid;



*Element-grille*

     grid-template-columns: 20% 20% 20% 20% 20%;

     grid-template-rows: 20% 20% 20% 20% 20%;

}

#element-grille{

     grid-column-start: 2;

     grid-column-end: 4;

     grid-column: 2/4;       /\* grid-column-start, grid-column-end \*/

grid-column: 2/ span 2;   /\* span permet de préciser que la largeur est de 2 cases \*/

     grid-row-start: 1;

     grid-row-end: 2;

     grid-row: 1/2;           /\* grid-row-start, grid-row-end \*/

     grid-area: 1/2/2/4;     /\* grid-row-start, grid-column-start, grid-row-end, grid-column-end \*/

}

## Une image contenant shoji, bâtiment Description générée automatiquementGestion des écarts entre 2 lignes/colonnes : *column-gap, row-gap*

.conteneur-grille{

  row-gap: 5px;

  column-gap: 10px;

  gap: 5px 10px; /\* row-gap , column-gap \*/

}

## Ordre des éléments : *order*

.conteneur-grille{

        display: grid;

        grid-template-columns: 20% 20% 20% 20% 20%;

        grid-template-rows: 20% 20% 20% 20% 20%;

    }

    #element-grille{

        order: +1;       /\* l'élément est décalé vers la droite, car tous les autres ont un indice d'ordre par défaut qui est 0 \*/

        order: -1;       /\* l'élément est décalé vers la gauche, car tous les autres ont un indice d'ordre par défaut qui est 0 \*/

    }

# Valeur d’affichage / Visibilité : *display* , *visibility*

## Elément non visible

img{

    display: none;

}

L’état de l’élément peut être changé avec du JavaScript, mais il est actuellement non visible

## Elément non visible avec conservation de l’espace

p{

    visibility : hidden;

}

L’élément n’est pas visible, mais l’espace qu’il occupe reste

## Elément de type « bloc » sur une ligne

Une image contenant texte

Description générée automatiquementli{

    display : inline;

}

Ex : Ici la liste s’affiche horizontalement et non verticalement

## Elément de type « inline » en bloc

a{

    display: block;

}

## Elément à la fois de type « inline » et « bloc »

span{

    display : inline-block;

}

## Opacité d’un élément : *opacity*

Une image contenant texte, arbre, route, voyageant

Description générée automatiquementimg{

    opacity : 0.2;

    opacity : 0.5;

}

img:hover{

    opacity : 1; /\* L'image n'est plus opaque lors du survol \*/

}

# Position d’un élément (statique, mouvante…)

## Définir une position : *position*

div{

    position : static;     /\* l'élément occupe une place statique sur la page \*/

    position : relative;   /\* l'élément occupe une place lui permettant de contenir des éléments en position absolue \*/

    position : absolute;   /\* l'élément est positionné selon l'élément (en position relative) qui le contient \*/

  position : fixed;      /\* L'élément ne suit pas le défilement de la page, reste toujours à l'écran \*/

    position : sticky;     /\* L'élément ne bouge pas de sa place, sauf quand il quitte l'écran \*/

}

**Remarque** : Il faut au moins définir une position entre top, bottom, left ou right pour que l’élément en position *absolute* se repère à son contenant.

## Chevauchement d’éléments : *z-index*

div{

    z-index : 1;       /\* L'indice est positif, donc s'affichera devant \*/

    z-index : -2;      /\* L'indice est négatif, donc s'affichera derrière \*/

}

**Remarque** : *z-index* ne marche que sur les éléments positionnés avec l’attribut *position*

# Transformations 2D (mouvements) : *transform*

## Translation : *translate()*

div{

    transform : translate(50px, 100px);  /\* L'élément se déplace de 50 pixels vers la droite et 100 pixels vers le bas \*/

}

## Rotation : *rotate()*

div{

    transform : rotate(20deg);  /\* l'élément tourne de 20 degrés dans le sens des aiguilles d'une montre \*/

    transform : rotate(-20deg); /\* Sens inverse \*/

}

## Agrandissement/diminution : *scale() , scaleX(), scaleY()*

div{

    transform: scaleX(2);   /\* Augmente de 2 fois la largeur d'origine \*/

    transform: scaleY(3);   /\* Augmente de 3 fois la hauteur d'origine \*/

transform: scaleX(-1); /\* Inversion de la gauche et la droite \*/

transform: scale(2,3);

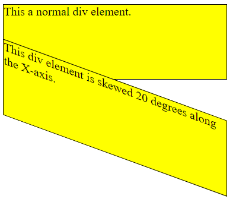
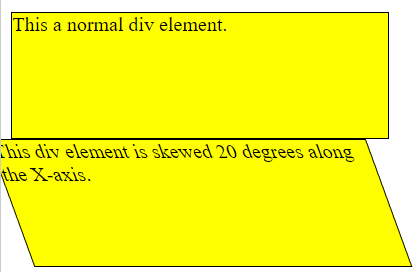
    transform: scale(0.5,0.8); /\* Diminue de moitié fois la largeur d'origine, et de 20% la hauteur d'origine \*/

}

## Inclinaison : *skew(), skewX(), skewY()*

div{

    transform: skewX(20deg);      /\* Incline de 20 degrés le long de l'axe X \*/



X

Y

    transform: skewY(30deg);      /\* Incline de 30 degrés le long de l'axe Y \*/

    transform: skew(20deg, 30deg)

}

## Ensemble de transformations : *matrix()*

div{

    transform: matrix(1, -0.3, 0, 1, 0, 0);

    /\* matrix(scaleX(),skewY(),skewX(),scaleY(),translateX(),translateY()) \*/

}

# Transformations 3D (mouvements et profondeur) : *transform*

Les transformations 3D peuvent également modifier l’axe Z d’un élément.

## Rotation 3D : *rotateX(), rotateY(), rotateZ()*

div{

    transform: rotateX(150deg);     /\* tourne de 150 degrés selon l'axe vertical \*/

    transform: rotateY(-150deg);    /\* tourne de 150 degrés selon l'axe horizontal, dans le sens inverse \*/

    transform: rotateZ(150deg);     /\* tourne de 150 degrés selon l'axe de profondeur \*/

}

## Perspective : *perspective*

div{

    perspective : 100px;   /\* l'élément possède une petite perspective \*/

    perspective : 800px;   /\* l'élément possède une grande perspective \*/

}

## Origine de la perspective : *perspective-origin*

div{

    perspective-origin: left;          /\* La perspective part de gauche \*/

    perspective-origin: center;        /\* La perspective part du centre \*/

    perspective-origin: top left;      /\* La perspective part d'en haut à gauche \*/

    perspective-origin: bottom right;  /\* La perspective part d'en bas à droite \*/

}

## Visibilité de la face arrière en perspective : *backface-visibility*

div{

    backface-visibility: visible;    /\* l'arrière est visible \*/

    backface-visibility: hidden;    /\* l'arrière n'est pas visible \*/

}

# Transitions

## Mise en place d’une transition : *transition*

div{

width : 100px;

    transition: width 2s;    /\* on prépare la transition en indiquant ce qui change, et la durée \*/

}

div:hover{

    width : 300px;           /\* Lors du survol, la transition s'effectue et la taille change \*/

}

## Faire plusieurs transitions sur un même élément

div{

    transition: width 2s, length 4s;    /\* la longueur et la largeur vont changer \*/

}

## Evolution de la vitesse de transition : *transition-timing-function*

div{

    transition-timing-function: linear;       /\* même vitesse tout le temps \*/

    transition-timing-function: ease;         /\* lent, puis rapide, puis lent \*/

    transition-timing-function: ease-in;      /\* démarrage lent \*/

    transition-timing-function: ease-out;     /\* fin lente \*/

    transition-timing-function: ease-in-out;  /\* lent, puis moyen, puis lent \*/

}

## Ajouter un délai de transition : *transition-delay*

div{

    transition-delay: 1s;   /\* La transition mettra 1 seconde à démarrer \*/

}

## Transition de transformation 2D ou 3D

div{

    transition: transform 2s;

}

div:hover{

    transform: rotate(180deg);

}

## Transition efficace pour le responsive

#titre{

    transition: all 1s ease;

}

# Animations

Une animation s’effectue d’elle-même, sans avoir besoin d ‘évènement déclencheur comparé aux transitions.

## Mise en place d’un animation : *@keyframes , animation-name, animation-duration*

@keyframes exemple {       /\* on définit l'animation par son nom \*/

    from {background-color: red;} to {background-color: yellow;}

}

div{

    background-color: green;       /\* une fois l'animation finie, le fond devient vert \*/

    animation-name: exemple;

    animation-duration: 4s;           /\* l'animation dure 4 secondes après le chargement de la page \*/

}

## Animation progressive (enchaînement de différentes valeurs)

@keyframes exemple {

    0% {background-color: red;}

    25% {background-color: yellow;}     /\* quart de l'animation \*/

    50% {background-color: green;}      /\* moitié de l'animation \*/

    100% {background-color: blue;}

}

## Animations multiples

@keyframes exemple {

    0% {background-color: red; left:20px;}

    25% {background-color: yellow; bottom:50px;}

    50% { opacity: 0.4; text-align: justify;}

}

## Délai d’animation : *animation-delay*

div{

    animation-delay : 2s;   /\* l'animation mettra 2 secondes à se lancer apres le chargement de la page \*/

    animation-delay : -3s;   /\* l'animation commencera 3 secondes avant le chargement de la page \*/

}

## Répétition de l’animation : *animation-iteration-count*

div{

    animation-iteration-count: 3;          /\* l'animation se répétera 3 fois \*/

    animation-iteration-count: infinite;   /\* l'animation se répétera indéfiniment\*/

}

## Sens de l’animation : *animation-direction*

div{

    animation-direction: normal;               /\* l'animation se joue normalement \*/

    animation-direction: reverse;              /\* l'animation se joue dans le sens inverse \*/

    animation-direction: alternate;            /\* l'animation se joue normalement, puis dans le sens inverse \*/

    animation-direction: alternate-reverse;    /\* l'animation se joue dans le sens inverse, puis normalement\*/

}

## Evolution de la vitesse d’animation : *animation-timing-function*

div{

    animation-timing-function: linear;       /\* même vitesse tout le temps \*/

    animation-timing-function: ease;         /\* lent, puis rapide, puis lent \*/

animation-timing-function: ease-in;      /\* démarrage lent \*/

    animation-timing-function: ease-out;     /\* fin lente \*/

    animation-timing-function: ease-in-out;  /\* lent, puis moyen, puis lent \*/

}

# Requêtes Média

Les requêtes Média se basent sur les dimensions de la fenêtre, de l’appareil , la résolution ou encore son orientation pour définir un style différent

## Selon la taille de l’appareil : *@media screen and max-width/min-width*

@media screen and (min-width: 600px) {

    body{

        background-color: brown;  /\* si l'écran mesure plus de 600 pixels, le fond est marron \*/

    }

}

@media screen and (max-width: 992px) {

    body{

        background-color: orange;  /\* si l'écran mesure moins de 992 pixels, le fond est orange \*/

    }

}

@media screen and (min-width: 600px) and (max-width: 992px){

    body{

        background-color: blue;  /\* si l'écran mesure entre 600 et 992 pixels, le fond est bleu \*/

    }

## Selon l’orientation de l’appareil : *@media screen and orientation*

@media screen and (orientation: landscape) {     /\* format paysage \*/

    body{

        color :red;

    }

}

@media screen and (orientation : portrait) {     /\* format portrait \*/

    body{

        color : purple;

    }

}

# Variables

## Utiliser des variables : *:root , var()*

:root{                   /\* on déclare les variables dans la racine de la feuille de style \*/

    --bleu : #1e90ff;

    --blanc : #ffffff;

}

body{

    background-color: var(--white);     /\* on peut rappeler ces variables n'importe où \*/

}

h1{

    color : var(--bleu);

}

# Compteurs

## Une image contenant texte Description générée automatiquementNumérotation automatique de titres : *conter-reset* , *counter-increment* , *content*

body{

    counter-reset : section;

}

h1::before {

    counter-increment: section;

    content: "Section " counter(section) ": ";

}

**Remarque :** Il est nécessaire de créer dans le body un compteur (appelé ici « section ») avec *counter-reset*, que l’on incrémente avec *counter-increment* à chaque balise <h1> et fait apparaître dans le format défini grâce à *content*

# Priorité

## Instaurer une priorité : *!important*

.paragraphe{

    background-color: aliceblue;

}

p{

    background-color: aquamarine !important;

}

Ici, un paragraphe de la classe « paragraphe » est censé être de la couleur *aliceblue* pour une question de priorité (les identifiants sont prioritaires sur les classes qui sont prioritaires sur les sélecteurs de balise), mais si l’on indique que la couleur *aquamarine* est plus importante pour tous les paragraphes, c’est elle qui sera utilisée.

# Fonction mathématiques

## Recherche de dimensions par fonctions mathématiques : *calc()*, *max()*, *min()*

div{

    width: calc(100% - 100px);  /\* La longueur de l'élément est la taille de l'écran - 100 pixels \*/

    width: max(50%, 300px);     /\* La longueur de l'élément suivra les dimensions du choix le plus grand \*/

    width: min(50%, 300px);     /\* La longueur de l'élément suivra les dimensions du choix le plus petit \*/

}

# Curseur de souris

## Styles de curseurs : *cursor*

.clicinterdit{

    cursor: help;

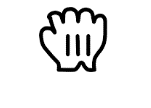
    cursor: wait;

    cursor: not-allowed;



Zoom-in

    cursor: zoom-in;



Grab

    cursor: grab;

}