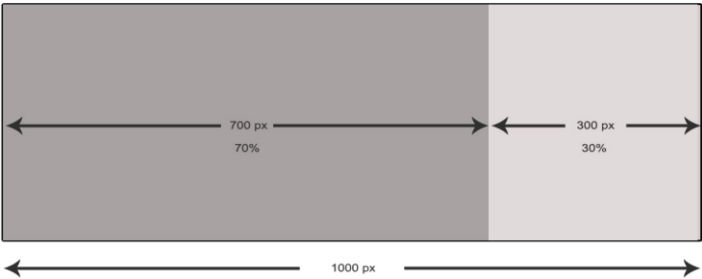


unités CSS

Le Pixel - px	<ul style="list-style-type: none">Unité « fixe »Dépendance à la résolution du périphérique
Le cadratin - em	<ul style="list-style-type: none">Unité fluide utilisée pour les polices d'écritureSa taille est relative à la taille de la police de l'élément parentUtiliser des polices en em permet de garder le rythme typographique et les proportions entre les différents éléments <p>Une police de 2 em = 2x plus grand</p> <p>La base : 10 px</p> <p>Une police de 0,5 em = 2x moins grand</p>
Le pourcentage : %	<ul style="list-style-type: none">Unités fluides (relatives à la taille de leur parent) :<ul style="list-style-type: none">– %: proportion relative à la dimension de l'élément parent ou la taille de la police selon la propriété. 

En web, quelle unité utiliser ?

- "px" pour exprimer une dimension fixe (indépendante de la taille de la police)
- "em" pour attribuer des tailles de polices en fonction de la taille de la police de leur parent.
- "%" pour attribuer des dimensions proportionnées aux dimensions de l'élément parent.
- Une mise en page fluide s'appuie généralement sur un dimensionnement des éléments en pourcentage

Propriétés de typographie, polices et puces

Font-family

- **font-family** : indiquer la (ou les) polices à utiliser

```
body { font-family: Arial, Helvetica, sans-serif;}
```

- L'ordre des polices appliquées est celui de la propriété de gauche à droite
- Il faut que la police soit présente sur le poste de l'utilisateur, sinon le navigateur passe à la suivante.

```
body {  
  font-family: Arial, Helvetica, sans-serif;  
}
```

Chocolate sweet sweet roll. Donut oat cake gummies topping wafer
caramels jelly-o. Lemon drops lollipop sweet roll marzipan faworki.
Gummi bears gummies pie candy canes. Wafer ice cream cookie



Chocolate sweet sweet roll. Donut oat cake gummies topping
wafer caramels jelly-o. Lemon drops lollipop sweet roll
marzipan faworki. Gummi bears gummies pie candy canes.

Font-size

- **font-size : valeur en unité** permet de définir la taille d'une police (= la « hauteur »)
- On utilise les unités vues précédemment (px, em, etc.)
- Exemple:

```
body {  
    font-size: 18px;  
}
```

Font-weight

- `font-weight: valeur` permet de déterminer l'épaisseur de la police.
- Principales valeurs possibles : `normal`, `bold` (gras) Permet de mettre en gras d'autres éléments
- Exemple: Mettre un paragraphe en gras

```
p {  
    font-weight: bold;  
}
```

Font-style

- **font-style** : valeur permet de préciser le style de police
- Principales valeurs possibles : **normal**, **italic**, **oblique**. Permet de mettre en italique des éléments
- Exemple : Mettre le paragraphe en italique

```
p {  
    font-style: italic;  
}
```

Line-height

- `line-height: valeur` change l'interlignage
- Les valeurs possibles sont des valeurs numériques en unité de mesure (px, em, %)
- Exemple : Augmenter l'interlignage

```
body {  
    line-height: 20px;  
}
```


Color

- **color: valeur** permet de donner la couleur du texte

```
h1 {  
    color: #008499;  
}
```

- il existe en CSS plusieurs façons de choisir une couleur
 - Notation hexadécimale : #ffffff (blanc)
 - Notation hexadécimale courte (qui est doublée pour obtenir la version longue) : #fff
 - Notation RGB : rgb(255,255,255)
 - Mot clé : white

Text-align

- `text-align` : `valeur` permet d'aligner le texte à gauche, droite ou milieu
- Les valeurs possibles sont : `left`, `right`, `center`, `justify` (comme sur Word)
- L'alignement se fait sur le contenu des balises de type bloc (paragraphes, titres, divs, etc.)

```
p {  
  text-align: left;  
}
```

Text-transform

- `text-transform` permet de transformer le texte en majuscule, minuscule
- Valeurs possibles :
 - `none`
 - `Capitalize`
 - `uppercase`
 - `lowercase`

```
h2 {
```

```
  text-transform: capitalize;
```

```
}
```

Du Texte De Titre H2

Text-decoration

- **text-decoration : valeur** permet de changer différents valeurs de « décoration »
- Valeurs possibles :
 - **none**: permet d'empêcher l'héritage de la propriété et plus particulièrement de supprimer le soulignement par défaut des liens
 - **overline** : surlignement
 - **underline** : soulignement
 - **line-through**: texte barré
- Exemple: Enlever le soulignement des liens et le remettre au survol

```
a {  
  
    text-decoration: none;  
  
}
```

/ enlever le soulignement des liens par défaut */*

```
a:hover {  
  
    text-decoration: underline;  
  
}
```

/ remettre le soulignement pour les liens survolés */*

Les listes

Listes à puces

- `list-style-type` : permet de spécifier le type de puce ou de numérotation (appliqué sur le ul / ol)
- Valeurs possibles
 - `none` (pas de puce),
 - `disc`,
 - `circle`,
 - `square`

Listes ordonnées

- Valeurs possibles
 - decimal (1.),
 - decimal-leading-zero (01.),
 - lower-roman (vii),
 - upper-roman (VII),
 - ...

Utiliser une image comme puce

- `list-style-image` : pour utiliser une image personnalisée en guise de puce
- Valeur : le chemin d'une image `url("cheminimage");`
- Exemple: Utiliser une image comme puce

```
ul {  
    list-style-image: url("img/liste.png");  
}
```

Bordures et arrières plans

Les bordures

- La propriété border permet d'ajouter une bordure à un élément
- Elle a 3 « sous propriétés » possibles : style, color et width. On regroupe souvent comme ceci :

border: width style color;

h2 {

border: 2px solid #009860 ;

}

Un titre avec une bordure !

Les bordures

- On peut décomposer les 4 bordures de la manière suivante :

`border-top: ...;`

`border-left: ...;`

`border-right: ...;`

`border-bottom: ...;`

- Si on ne précise pas à quelle bordure on applique la propriété, elle s'applique aux 4.

```
h2 {
```

```
  border-top: 1px solid gray;
```

```
}
```

Un titre h2 ici

Style de bordure

none : pas de bordure (par défaut);

dotted : pointillés;

dashed : tirets;

solid : un trait simple;

double : bordure double;

groove : en relief;

ridge : autre effet relief;

inset : effet 3D global enfoncé;

outset : effet 3D global surélevé.

dotted (1 pixel)

dashed (1 pixel)

solid (1 pixel)

double (3 pixels)

groove (2 pixels)

ridge (2 pixels)

inset (2 pixels)

outset (2 pixels)

Des bords arrondis : CSS3 !

- `border-radius : valeur` permet de faire des bords arrondis (propriété CSS3)
- Valeur : une valeur en unité de mesure (px, em ou même %)
 - ✓ Le même arrondi partout : `border-radius : 10px;`
 - ✓ Un arrondi par angle (en haut à gauche, en haut à droite, en bas à droite, en bas à gauche) :
`border-radius : 2px 20px 5px 10px;`

```
div {  
  border:1px solid blue;  
  
  border-radius : 10px;  
  
}
```

Soufflé gingerbread cupcake donut tiramisu cookie sweet soufflé cake.
Candy canes candy canes donut toffee candy canes cheesecake
gummies apple pie. Pie cupcake cheesecake sugar plum tart donut bear
claw caramels

Background-color

- `background-color : valeur` permet de donner une couleur de fond.
- Valeurs : n'importe quelle couleur CSS
- Sur un élément bloc (prend toute la largeur) ou inline (prend la largeur du contenu)

Background-image

- `background-image : valeur` permet de définir une image de fond pour l'élément
- Valeur : `none` ou `url("monimage.png")`
- Par défaut : l'image se répète horizontalement et verticalement (comme une mosaïque)

Dimensions, margin et padding

Petit rappel inline et block

- Sans altération CSS, les éléments de bloc prennent toute la largeur de leur parent, et sont suivis d'un retour à la ligne (body, html, p, div, h1, etc.)
- Les éléments de type inline prennent la largeur de leur contenu, et ne sont pas suivis d'un retour à la ligne (span, a, strong, img, etc.).

Largeur et hauteur d'un élément : width et height

- **width** (largeur) et **height** (hauteur) ne sont applicables que sur des éléments de bloc
- Valeurs possibles : **auto** (par défaut, toute la taille du parent), **valeur numérique en px**, **%** ou même **em**.
- Exemple: `body{ width: 800px;}`

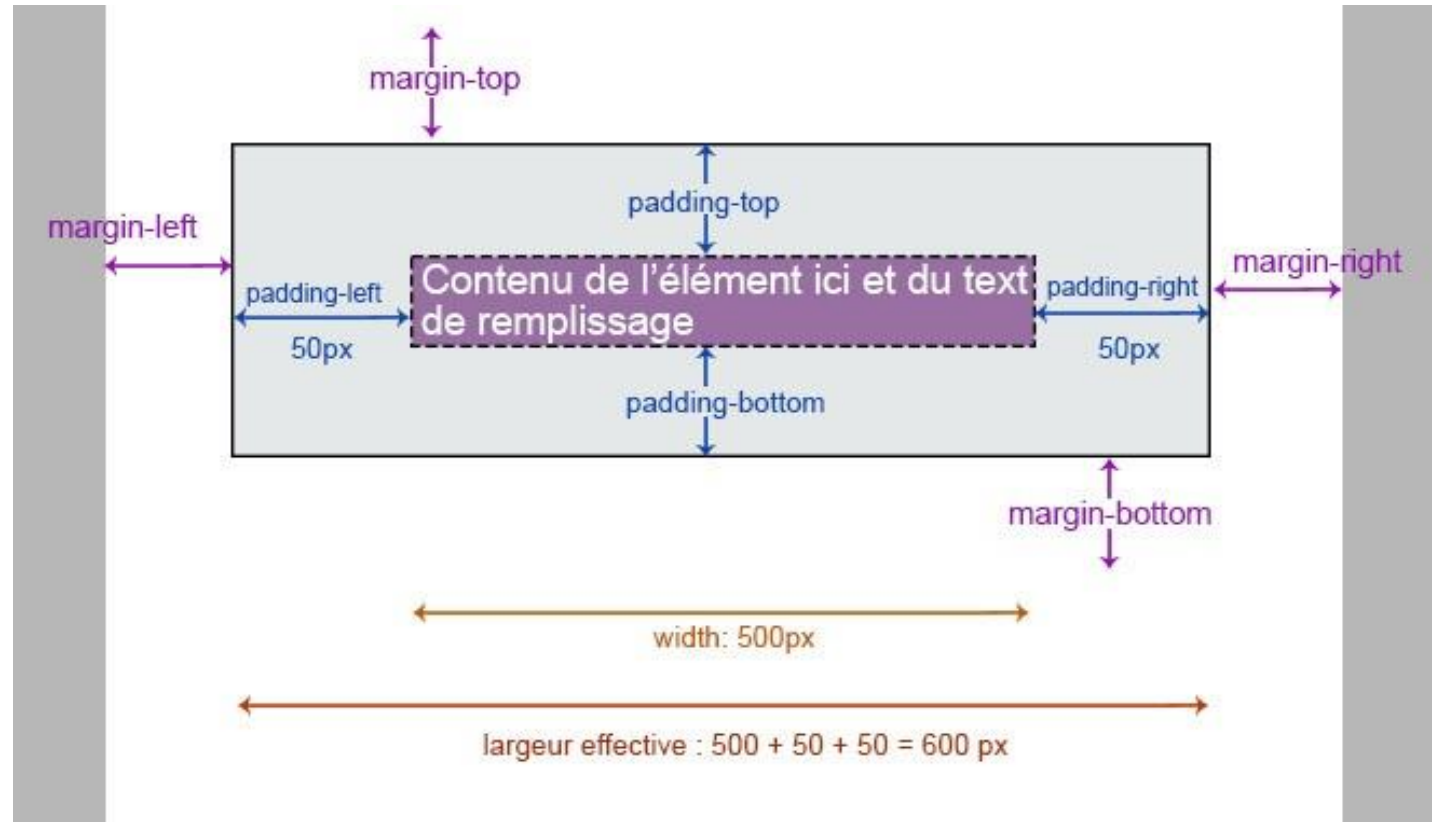
Max et min

- On peut utiliser max et min pour déterminer des tailles maximums et minimums :
`min-width`, `min-height`, `max-width` et `max-height`
- Valeurs : numériques
- Permet par exemple de gérer un débordement d'image :

`img { max-width: 100%; }` : les images sont limitées à la taille de leur parent

Marges

Il existe deux types de marges : intérieure appelée padding, et extérieure appelée margin



(s'il y a une bordure elle est aussi ajoutée à la largeur effective des éléments)

Marge extérieure « margin »

- La propriété **margin** détermine l'espace entre le bord de l'élément, et l'élément suivant.
- Par défaut margin: valeur applique la même valeur aux 4 cotés
- Valeur possible : auto, valeur en unité de mesure, positive ou négative

Marge extérieure « margin »

- On peut la décomposer individuellement en
 - margin-top : marge extérieure haute
 - margin-right : marge extérieure droite
 - margin-bottom : marge extérieure basse
 - margin-left : marge extérieure gauche
- Ou rassembler les 4 valeurs : `margin: 10px 5px 8px 15px;` (dans l'ordre : haut, droite, bas, gauche)

Margin et éléments inline

- Par défaut, une margin top ou bottom sur un élément inline ne fonctionnera pas (puisque'il est dans le flux de la page).
- Une marge left ou right fonctionnera par contre

Padding : la marge « intérieure »

- La propriété `padding` détermine un espacement entre le bord de la boîte et son contenu
- Par défaut `padding : valeur` applique la même valeur aux 4 côtés
- Valeur possible : `valeur en unité de mesure`



Padding : la marge « intérieure »

- On peut la décomposer individuellement en
 - **padding-top** : marge intérieure haute
 - **padding-right** : marge intérieure droite
 - **padding-bottom** : marge intérieure basse
 - **padding-left** : marge intérieure gauche
- Ou rassembler les 4 valeurs : padding: 10px 5px 8px 15px; (dans l'ordre : haut, droite, bas, gauche)

Ajouter du padding au body

```
body {
```

```
padding: 10px 15px;
```

```
}
```

Si on ne met que deux valeurs, la 1^{ère} s'appliquera au haut et au bas, la seconde à droite et à gauche.

Un titre ici

Soufflé gingerbread cupcake donut tiramisu cookie sweet soufflé cake. Candy canes candy canes donut toffee candy canes cheesecake gummies apple pie. Pie cupcake cheesecake sugar plum tart donut bear claw caramels

liquorice brownie. Cupcake halvah topping tiramisu pudding ice cream candy canes. **Jelly beans cookie marshmallow cheesecake** oat cake chupa chups pie sweet wypas. Wafer pie macaroon lemon drops danish sugar plum pastry ice cream. Bear claw wafer tiramisu chocolate.



Un titre ici

Soufflé gingerbread cupcake donut tiramisu cookie sweet soufflé cake. Candy canes candy canes donut toffee candy canes cheesecake gummies apple pie. Pie cupcake cheesecake sugar plum tart donut bear claw caramels

liquorice brownie. Cupcake halvah topping tiramisu pudding ice cream candy canes. **Jelly beans cookie marshmallow cheesecake** oat cake chupa chups pie sweet wypas. Wafer pie macaroon lemon drops danish sugar plum pastry ice cream. Bear claw wafer tiramisu chocolate.

Overflow : pour éviter que ça dépasse

- Il est possible que du contenu texte dépasse l'élément qui le contient (surtout si on définit des hauteurs fixes)
- Exemple : `p {height: 100px}`, le contenu « dépasse » du paragraphe

Chocolate sweet sweet roll. Donut oat cake gummies topping wafer caramels jelly-o. [Lemon drops lollipop](#)
[sweet roll marzipan faworki](#). Gummi bears gummies pie candy canes. Wafer ice cream cookie dragée donut
brownie applicake fruitcake. Wafer marshmallow sugar plum. Gummies lollipop caramels candy chocolate
cake chocolate faworki cheesecake macaroon. Applicake chocolate bar soufflé marshmallow. Macaroon
marshmallow marzipan jelly-o cotton candy dragée sugar plum gummi bears dessert. Jelly beans croissant
danish gummi bears lemon drops. Chocolate bar biscuit pudding wafer pastry tart toffee. Caramels tart jelly.

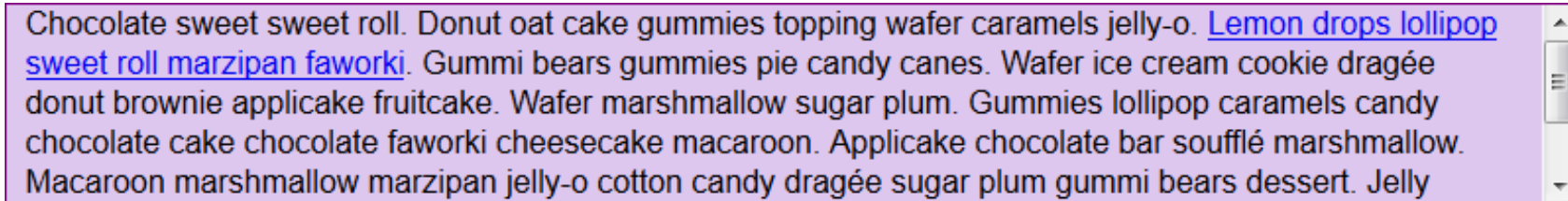
Overflow: hidden - cacher ce qui dépasse

- overflow: hidden permet de cacher tout ce qui dépasse.
- Problème : on perd tout ce qui est caché, impossible de le voir

Chocolate sweet sweet roll. Donut oat cake gummies topping wafer caramels jelly-o. [Lemon drops](#)
[lollipop sweet roll marzipan faworki](#). Gummi bears gummies pie candy canes. Wafer ice cream cookie
dragée donut brownie applicake fruitcake. Wafer marshmallow sugar plum. Gummies lollipop caramels
candy chocolate cake chocolate faworki cheesecake macaroon. Applicake chocolate bar soufflé
~~marshmallow. Macaroon marshmallow marzipan jelly-o cotton candy dragée sugar plum gummi bears~~

Overflow: auto - afficher une barre de défilement

- overflow: auto; permet d'afficher une bar de défilement si nécessaire

A screenshot of a text box with a light purple background and a thin purple border. The text inside is a long string of dessert names: "Chocolate sweet sweet roll. Donut oat cake gummies topping wafer caramels jelly-o. [Lemon drops lollipop](#) [sweet roll marzipan faworki](#). Gummi bears gummies pie candy canes. Wafer ice cream cookie dragée donut brownie applicake fruitcake. Wafer marshmallow sugar plum. Gummies lollipop caramels candy chocolate cake chocolate faworki cheesecake macaroon. Applicake chocolate bar soufflé marshmallow. Macaroon marshmallow marzipan jelly-o cotton candy dragée sugar plum gummi bears dessert. Jelly". The text is wrapped onto five lines. On the right side of the text box, there is a vertical scrollbar with a grey track and a white slider, indicating that the text is scrollable.

Chocolate sweet sweet roll. Donut oat cake gummies topping wafer caramels jelly-o. [Lemon drops lollipop](#) [sweet roll marzipan faworki](#). Gummi bears gummies pie candy canes. Wafer ice cream cookie dragée donut brownie applicake fruitcake. Wafer marshmallow sugar plum. Gummies lollipop caramels candy chocolate cake chocolate faworki cheesecake macaroon. Applicake chocolate bar soufflé marshmallow. Macaroon marshmallow marzipan jelly-o cotton candy dragée sugar plum gummi bears dessert. Jelly