

**Support apprenant
CSS**

Plan de formation

1. Introduction aux CSS
2. Syntaxes CSS
3. Généalogie et différents types de sélecteurs
4. Couleurs et unités CSS
5. Positionnement CSS : flexbox
6. Bordures et arrière-plan
7. Dimension : margin et padding
8. Positionnement CSS : display
9. Positionnement CSS : position
10. Positionnement CSS : float
11. Propriétés de typographie : puces et polices
12. Pseudo-Eléments
13. Pseudo-classes

1.

Introduction aux css

Introduction à CSS

- CSS : feuille de style en cascade
- Permettent de générer la présentation d'une page HTML : séparer la structure (HTML) de sa présentation (CSS)
- Ensemble de règles stylistiques applicables à un, ou plusieurs documents HTML => gain de taille du fichier HTML
- Facilite la mise à jour graphique, favorise l'accessibilité

Historique

- Le CSS est né en 1996
- Avant, on utilisait des balises de présentation directement dans le HTML
- **CSS1** : publié en 1996
- **CSS2** : Pour répondre aux besoins que la première spécification CSS1 ne couvrait pas encore
- **CSS2.1** : Les retours d'implémentation de CSS2 conduisent le groupe de travail CSS du W3C à rédiger à partir de 2001 une version révisée CSS 2.1
- **CSS3** : Le développement du troisième niveau des feuilles de styles en cascade commence dès 1999, parallèlement à celui de CSS 2.1
- **CSS4** : L'écriture du quatrième niveau des feuilles de styles en cascade débute dès 2010 parallèlement

Le CSS dans la balise html

Attribut **style** = ...;

Titre en couleur rose

<h1 style="color:pink;" > </h1>

Et des sous titres avec une couleur de fond violette

<h2 style="background-color:violet;"> </h2>

Cupcake ipsum dolor sit

amet jelly sugar plum bear claw bonbon. Wypas dragée chocolate bar fruitcake. Gingerbread wypas muffin cotton candy toffee sugar plum I love danish icing.

Ice cream topping

Sweet roll sweet roll chupa chups donut toffee I love. Toffee gummi bears candy canes tiramisu bear claw candy canes chocolate bar candy. Ice cream topping apple pie. Cotton candy jelly-o gummies. I love lollipop I love jelly I love I love bonbon. Wypas topping topping. Cheesecake gingerbread cookie gingerbread pastry. Caramels pastry sweet roll.

Toffee ice cream dessert powder donut I love croissant gummi bears. I love apple pie dessert chocolate cake cotton candy fruitcake dessert. Icing I love sugar plum tiramisu.

Ice cream topping

I love I love cheesecake muffin gingerbread. Lollipop croissant biscuit. Cotton candy jelly cheesecake. Tart I love jelly halvah sesame snaps.

Ice cream topping

Candy icing apple pie soufflé lollipop bonbon. Toffee sesame snaps lemon drops sugar plum cake. Apple pie ice cream tiramisu toffee marshmallow gingerbread applicake. Wypas fruitcake jelly-o lemon drops. Lemon drops sugar plum I love. Liquorice chocolate bar muffin. Tiramisu tart jelly-o jelly beans pie I love applicake. Toffee I love cheesecake jelly-o soufflé cookie.

Problèmes de maintenabilité

- Et si je veux appliquer un style à tous les titres ?
- Et si je veux changer tous les titres d'un coup ?
- Long et fastidieux
- MAIS c'est comme ça que le JavaScript inject du CSS donc vous risquez de le voir dans l'**inspecteur d'élément**

Le CSS « interne » dans l'entête du HTML

On place une balise `<style>` dans la balise `<head>` du document Changer la couleur de tous les titres **h2** en blanc :

```
<style type="text/css">  
  h2{  
    color: #fff;  
  }  
</style>
```

Cupcake ipsum dolor sit

amet jelly sugar plum bear claw bonbon. Wypas dragée chocolate bar fruitcake. Gingerbread wypas muffin cotton candy toffee sugar plum I love danish icing.

Ice cream topping

Sweet roll sweet roll chupa chups donut toffee I love. Toffee gummi bears candy canes tiramisu bear claw candy canes chocolate bar candy. Ice cream topping apple pie. Cotton candy jelly-o gummies. I love lollipop I love jelly I love I love bonbon. Wypas topping topping. Cheesecake gingerbread cookie gingerbread pastry. Caramels pastry sweet roll.

Toffee ice cream dessert powder donut I love croissant gummi bears. I love apple pie dessert chocolate cake cotton candy fruitcake dessert. Icing I love sugar plum tiramisu.

Ice cream topping

I love I love cheesecake muffin gingerbread. Lollipop croissant biscuit. Cotton candy jelly cheesecake. Tart I love jelly halvah sesame snaps.

Ice cream topping

Candy icing apple pie soufflé lollipop bonbon. Toffee sesame snaps lemon drops sugar plum cake. Apple pie ice cream tiramisu toffee marshmallow gingerbread applicake. Wypas fruitcake jelly-o lemon drops. Lemon drops sugar plum I love. Liquorice chocolate bar muffin. Tiramisu tart jelly-o jelly beans pie I love applicake. Toffee I love cheesecake jelly-o soufflé cookie.

Le CSS « interne » dans l'entête du HTML

Avantage :

Possibilité d'appliquer une règle sur tous les mêmes éléments d'un même document

Inconvénients :

Ne s'applique qu'à ce document, et non au site en entier

Il faut donc modifier le CSS sur chaque page (ex : si on a 10 pages, modifier les styles sur les 10 fichiers)

La/les feuille(s) de style externe(s)

- Nous utiliserons les mêmes conventions de chemin que pour les images et pages
- Création d'un nouveau fichier appelé « styles.css » et on lie la feuille de styles au HTML
- On « lie » la feuille de style au document HTML avec une balise `<link>` dans le

```
<link rel="stylesheet" href="styles.css"/>
```

La feuille de style externe

index.html

```
<!DOCTYPE HTML>
<html lang="fr-FR">
<head>
  <meta charset="UTF-8">
  <title>Mon premier CSS</title>
  <link rel="stylesheet" type="text/css" href="styles.css" />
```

style.css

```
styles.css
 1 p{
 2   background:yellow;
 3 }
```

Cupcake ipsum dolor sit

amet jelly sugar plum bear claw bonbon. Wypas dragée chocolate bar fruitcake. Gingerbread wypas muffin cotton candy toffee sugar plum I love danish icing.

Ice cream topping

Sweet roll sweet roll chupa chups donut toffee I love. Toffee gummi bears candy canes tiramisu bear claw candy canes chocolate bar candy. Ice cream topping apple pie. Cotton candy jelly-o gummies. I love lollipop I love jelly I love I love bonbon. Wypas topping topping. Cheesecake gingerbread cookie gingerbread pastry. Caramels pastry sweet roll.

Toffee ice cream dessert powder donut I love croissant gummi bears. I love apple pie dessert chocolate cake cotton candy fruitcake dessert. Icing I love sugar plum tiramisu.

Ice cream topping

I love I love cheesecake muffin gingerbread. Lollipop croissant biscuit. Cotton candy jelly cheesecake. Tart I love jelly halvah sesame snaps.

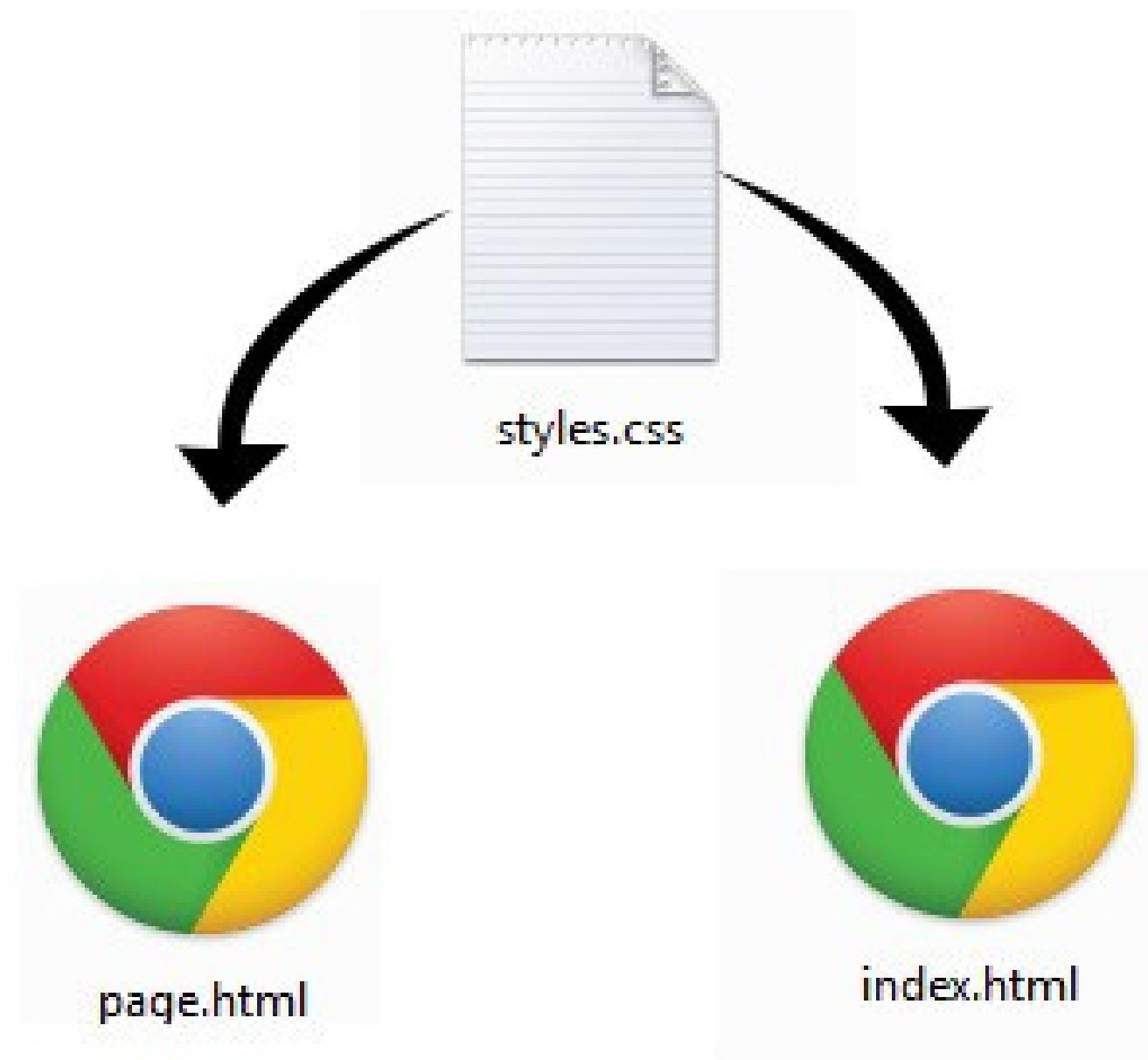
Ice cream topping

Candy icing apple pie soufflé lollipop bonbon. Toffee sesame snaps lemon drops sugar plum cake. Apple pie ice cream tiramisu toffee marshmallow gingerbread applicake. Wypas fruitcake jelly-o lemon drops. Lemon drops sugar plum I love. Liquorice chocolate bar muffin. Tiramisu tart jelly-o jelly

La feuille de style externe

Avantages :

- Séparation totale de la structure et de la présentation : possibilité d'avoir des présentations alternatives
- Possibilité d'appliquer la même feuille de style sur plusieurs pages (donc le site en entier) sans devoir la dupliquer
- Cohérence de la présentation sur



2.

Syntaxe CSS

La syntaxe de base

La syntaxe est composée de 3 éléments :

- Le **sélecteur** est l'élément sur lequel on applique les propriétés (balise HTML, id, classe, etc.)
- La **propriété** est l'effet que l'on va vouloir donner (ex couleur de texte, positionnement, couleur de fond, etc.)
- La **valeur** de la propriété CSS (rouge 10px etc.)

```
selecteur {  
    propriete: valeur  
    propriete: valeur  
    ...  
}
```

La déclaration CSS

On appelle « une déclaration CSS » l'**ensemble propriété + valeur**. On peut en avoir plusieurs pour un même élément et elles sont séparées par le symbole point virgule.

Les commentaires CSS

- Pour créer un commentaire CSS, on utilise

`/* commentaire */`

- Le commentaire n'est pas visible dans le navigateur
- Sert au développeur à prendre des notes, laisser des messages aux autres et servir à aussi comment les lignes de code

3.

Généalogie et différents types de sélecteurs

Le sélecteur usuel d'élément HTML

- On peut sélectionner n'importe quel élément HTML et lui appliquer un style.
- Une propriété appliquée sur un élément HTML, s'applique par défaut à tous les éléments présents dans le document HTML.

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

=> Tous les paragraphes auront une couleur de texte bleu

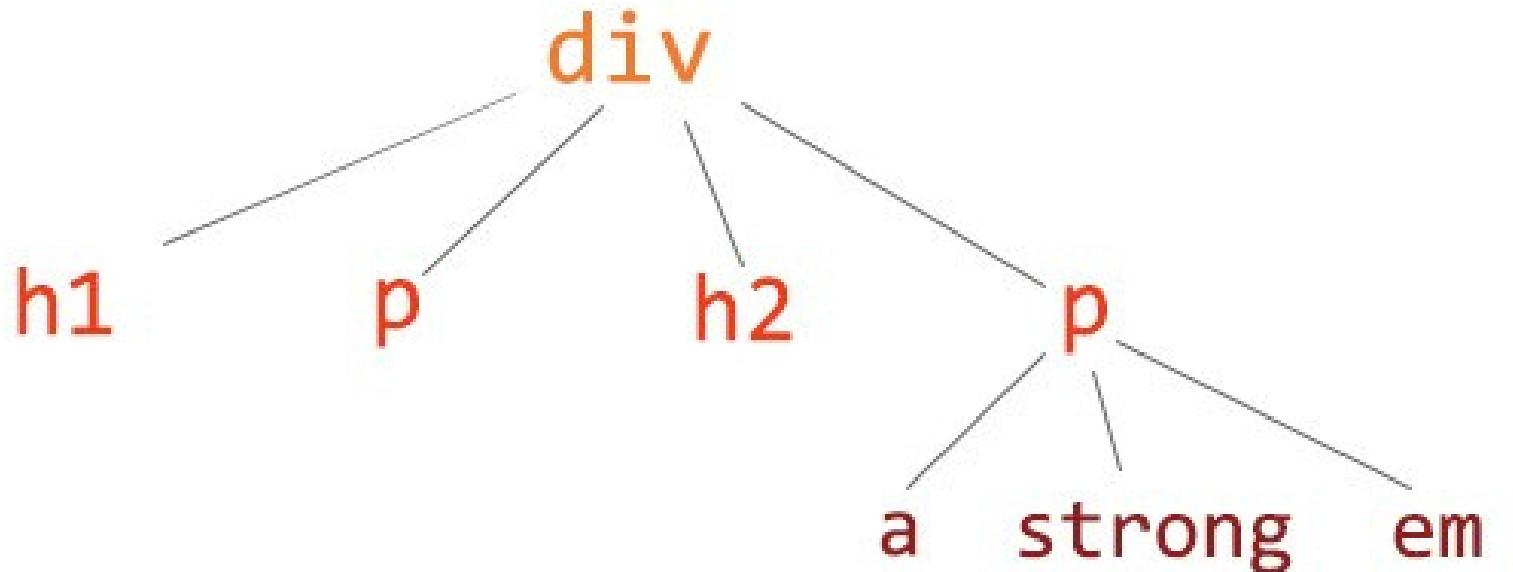
Hiérarchie et généalogie

```
<div>
  <h1>Cupcake ipsum dolor sit</h1>
  <p>
    amet jelly sugar plum bear claw bonbon. Wypas dragée chocolate bar fruitcake. Gingerbread
    wypas muffin cotton candy toffee sugar plum I love danish icing.
  </p>
  <h2>Ice cream topping</h2>
  <p>
    <a href="#">Toffee gummi bears candy canes tiramisu bear</a>
    claw candy canes chocolate bar candy.
    <strong>Ice cream topping apple pie</strong>Cotton candy jelly-o gummies.
    <em>I love lollipop I love jelly</em>
  </p>
</div>
```



Notion d'enfant et de descendance

- h1, p, h2, p sont enfants de div
- a, strong et em sont enfant du p dans lequel ils sont contenus (mais pas de l'autre p)

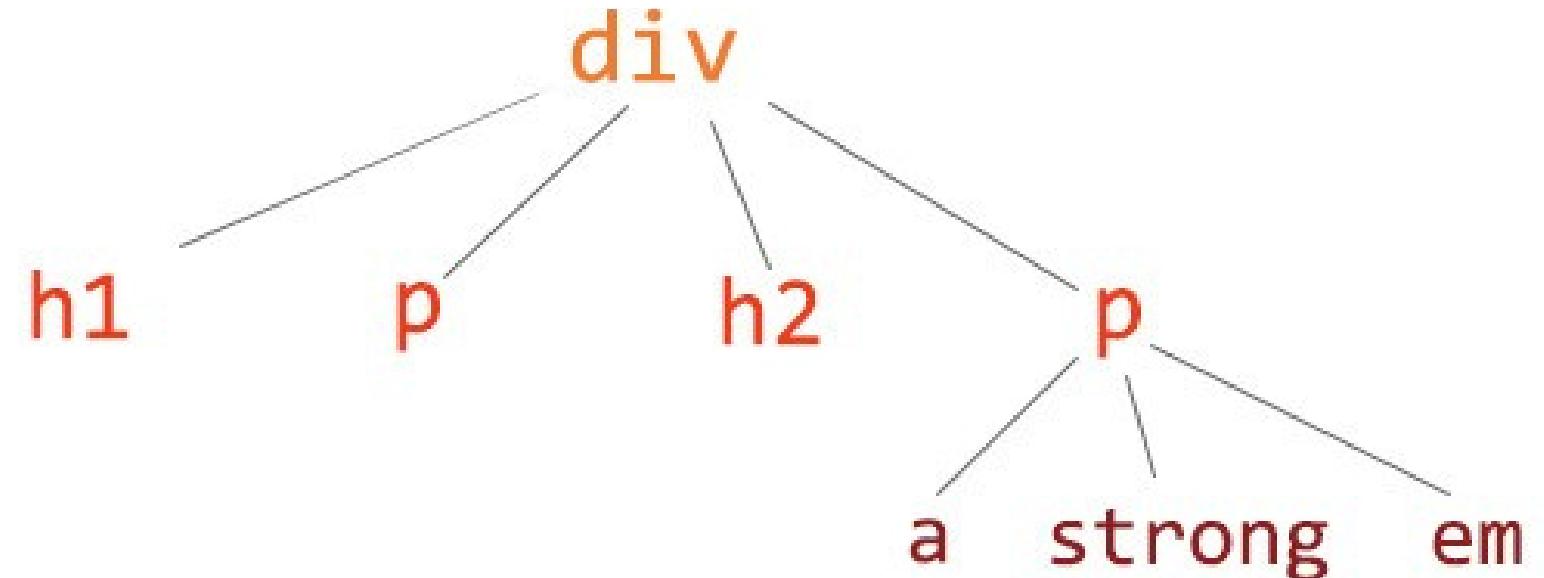
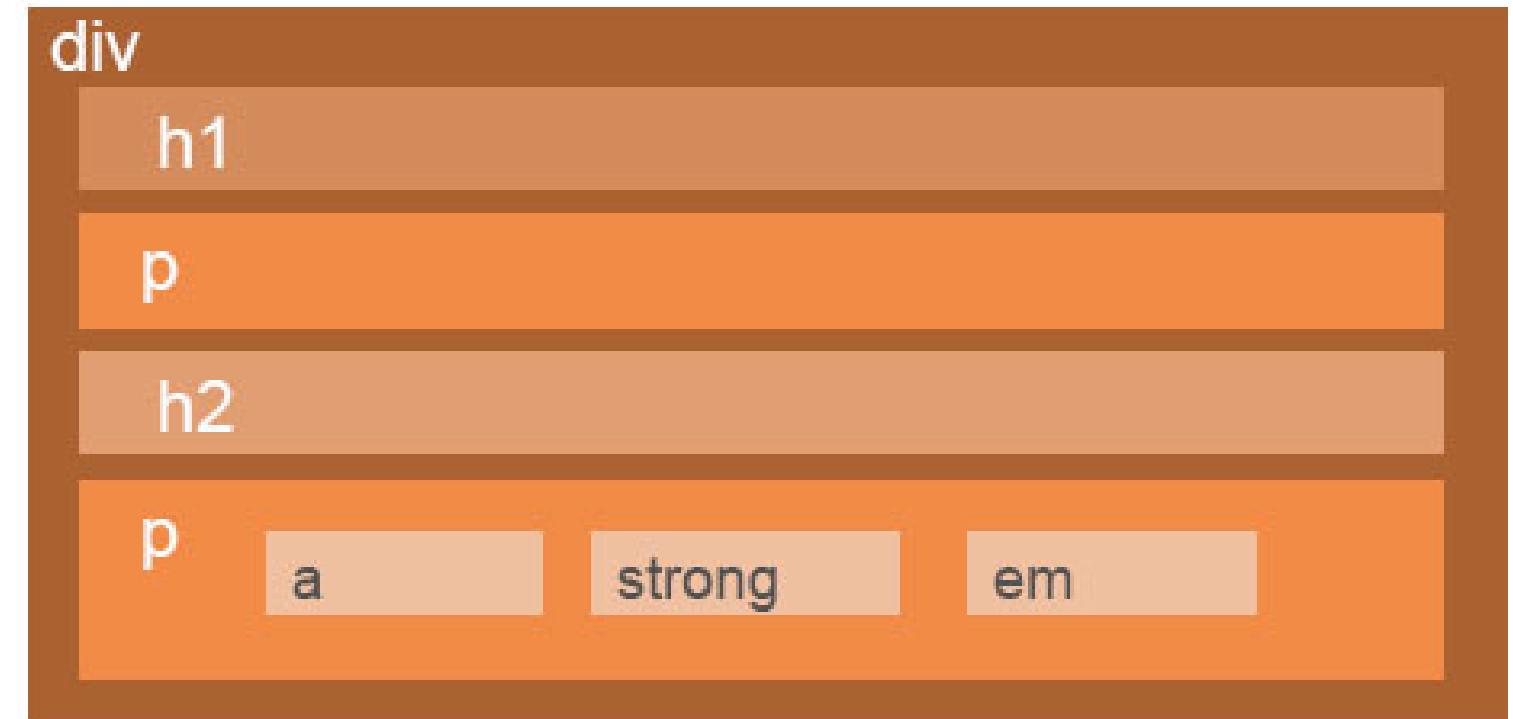


Notion de parent et d'ancêtre

div est **parent** de h1 p, h2, p Le 2ème

p est **parent** de a, strong et em.

div est **ancêtre** de a, strong et em (et
de h1, p h2 et p)



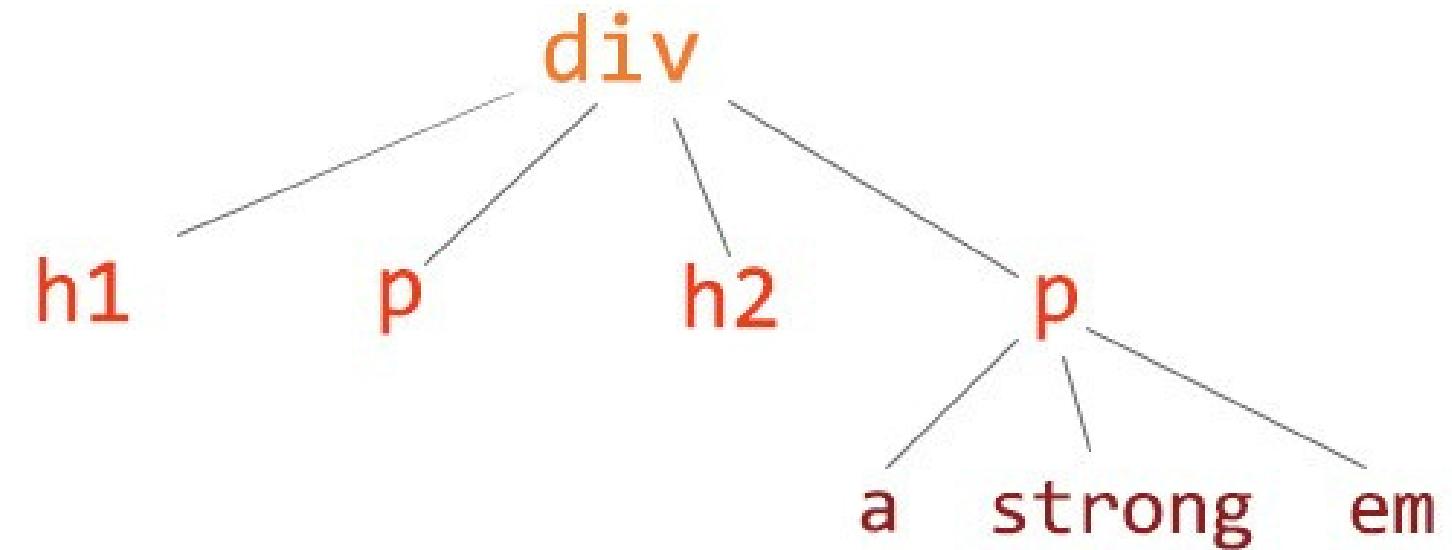
Sélecteur de hiérarchie

Pour sélectionner le `a` descendant de `p`,

nous allons pouvoir écrire :

`p a { ... }`

(notez l'espace entre le `p` et le `a`)

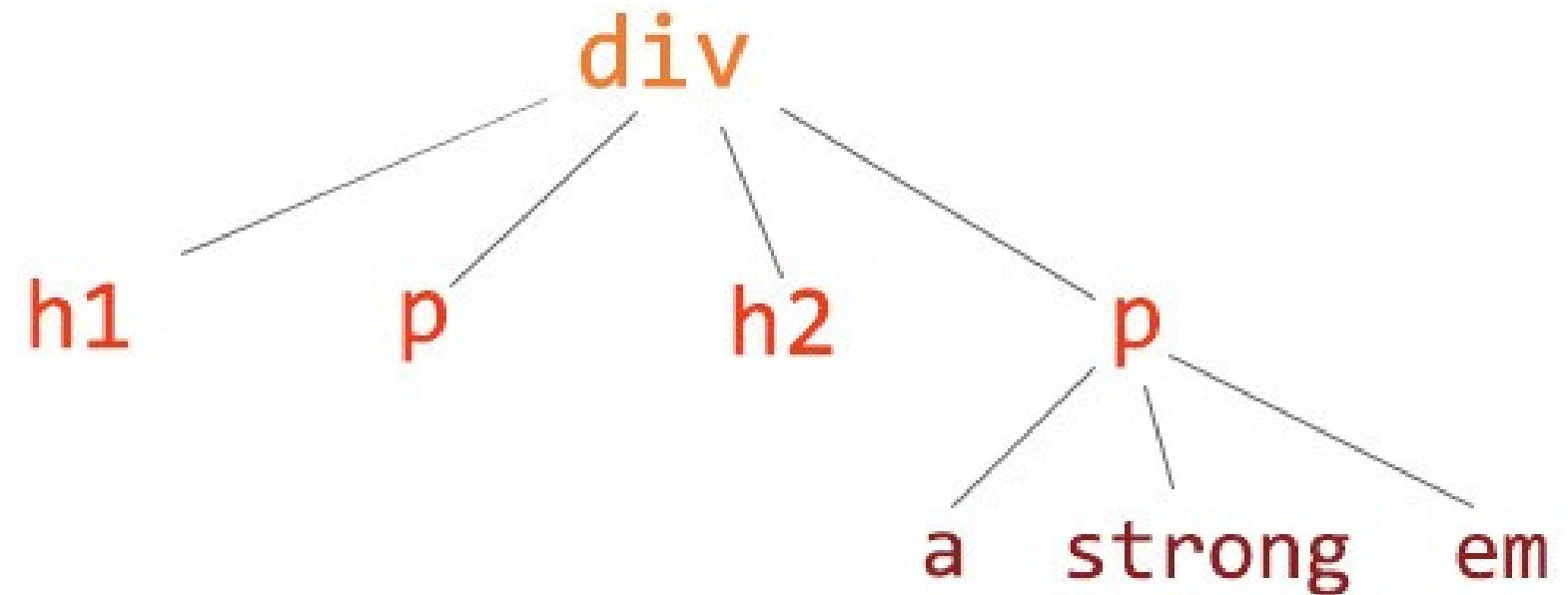
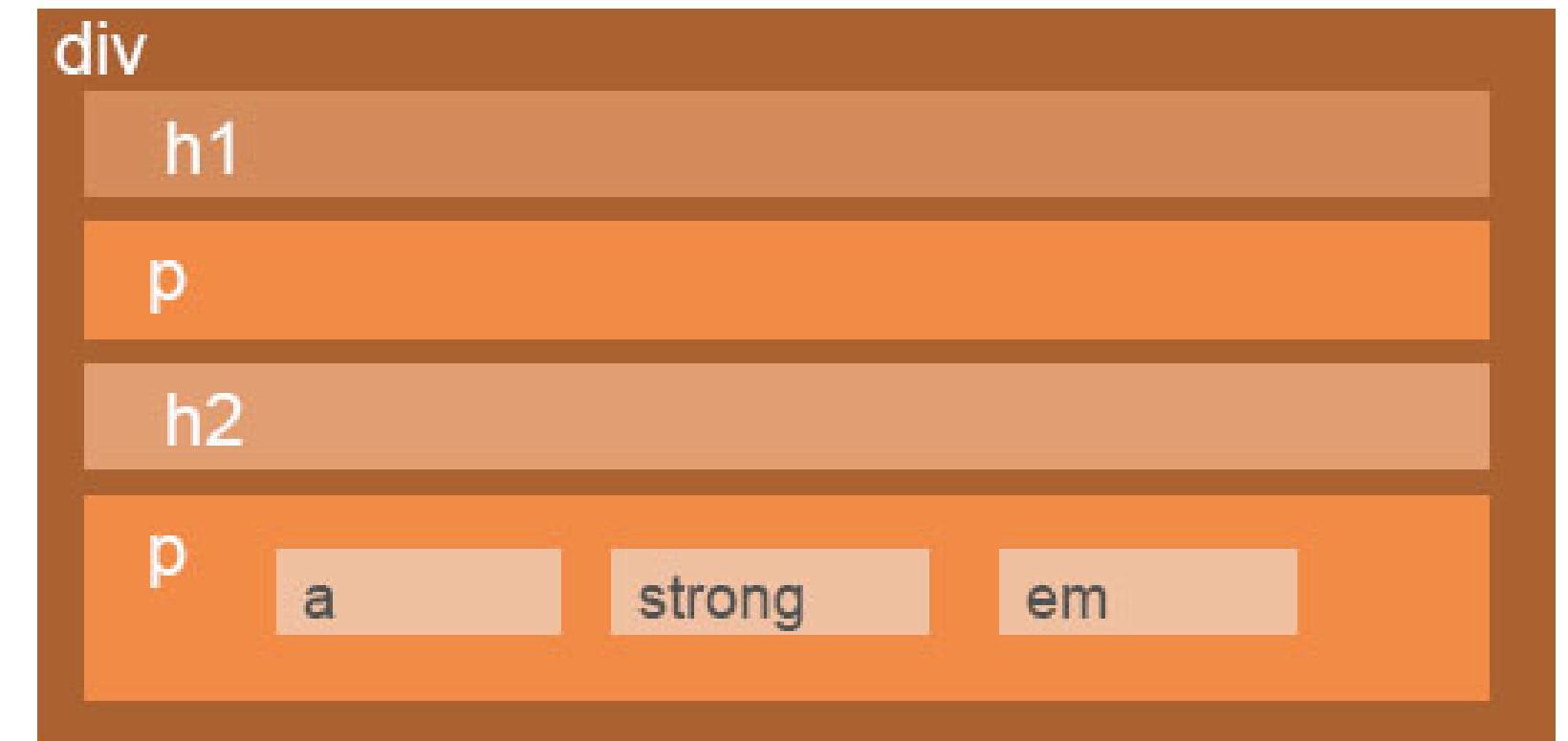


Sélecteur de groupe

Pour sélectionner plusieurs éléments et leur appliquer la même valeur, on les sépare par une virgule.

```
h1, h2 { color: red; }
```

=> Va donner la couleur rouge à tous les h1 ou h2



Les sélecteurs de classe

`class=""` est un attribut qui peut se mettre sur n'importe quelle balise
Il permet de cibler de manière plus « spécifique » certains éléments

`<p class="important">` Un paragraphe spécifique qui est mis en avant `</p>`

Un `<paragraphe>`

`Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor
 incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis
 nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
 Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu
 fugiat nulla pariatur. lorem ipsum quia dolor sit amet`



Classes multiples

Une même classe peut être utilisée sur plusieurs balises dans le même document

< p class="remarque" > une remarque paragraphe </ p >

< blockquote class="remarque" > une remarque citation </ blockquote >

Un <paragraphe>

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. lorem ipsum quia dolor sit amet

remarque

Un <blockquote>

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. lorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem.

remarque

Cibler une classe indépendamment de la balise

Si on veut cibler une classe, sans se préoccuper de la balise on utilise le « point ».

Par exemple `.remarque { }` va cibler

```
<p class="remarque"> ... </p>
```

```
<blockquote class="remarque"> .... </blockquote>
```

Une même classe peut être utilisée sur plusieurs balises HTML

Cibler une balise HTML avec une classe

Pour cibler un élément doté d'une classe en CSS on procède de la manière suivante (notez qu'il n'y a pas d'espace) **element.maclasse {}**

Par exemple : **p.toto {}** va cibler tous les éléments comme celui- ci :

```
<p class="toto"> ... </p>
```

Cibler une balise HTML avec une classe

Et **button.btn.btn-default{ }** va cibler

```
<button type="button" class="btn btn-default"> ... </button>
```

Si l'on met plusieurs classes et un élément sans espace dans la déclaration, il faut donc que **TOUTES** les conditions soient vraies



L'id (identifiant)

Il a le même rôle qu'une classe, mais doit être unique sur la page (donc on va moins l'utiliser)

Un seul attribut **id** par balise est possible On le note

<element id="monid">

On le cible en CSS avec **#monid { }**

Convention de nommage

- Un nom de classe ou d'ID ne doit pas avoir d'accent ou caractères spéciaux
- On a tendance à éviter le « tout majuscule » Un nom d'ID ne prend jamais d'espace
- Un nom de classe ne prend pas d'espace, sinon c'est une nouvelle classe

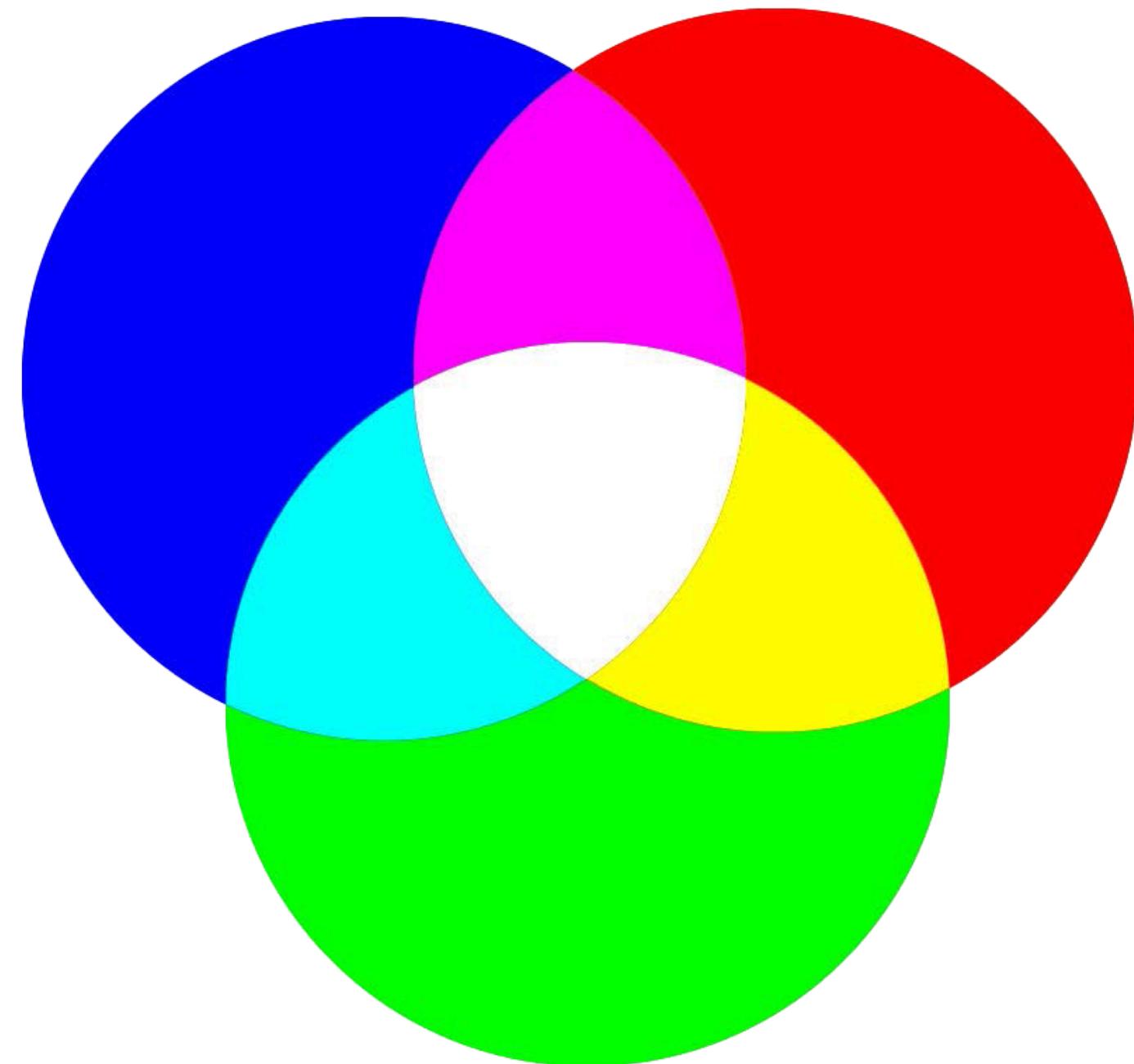
En résumé

CSS	HTML ciblé
p {}	<p> ... </p>
.maclasse {}	<element class="maclasse"> ... </element>
#monid {}	<element id="monid"> ... </element>
p.maclasse {}	<p class="maclasse"> ... </p>
p .maclasse {}	<p> <element class="maclasse"> ... </element></p>
p, div {}	<p> ... <p> <div> ... </div>

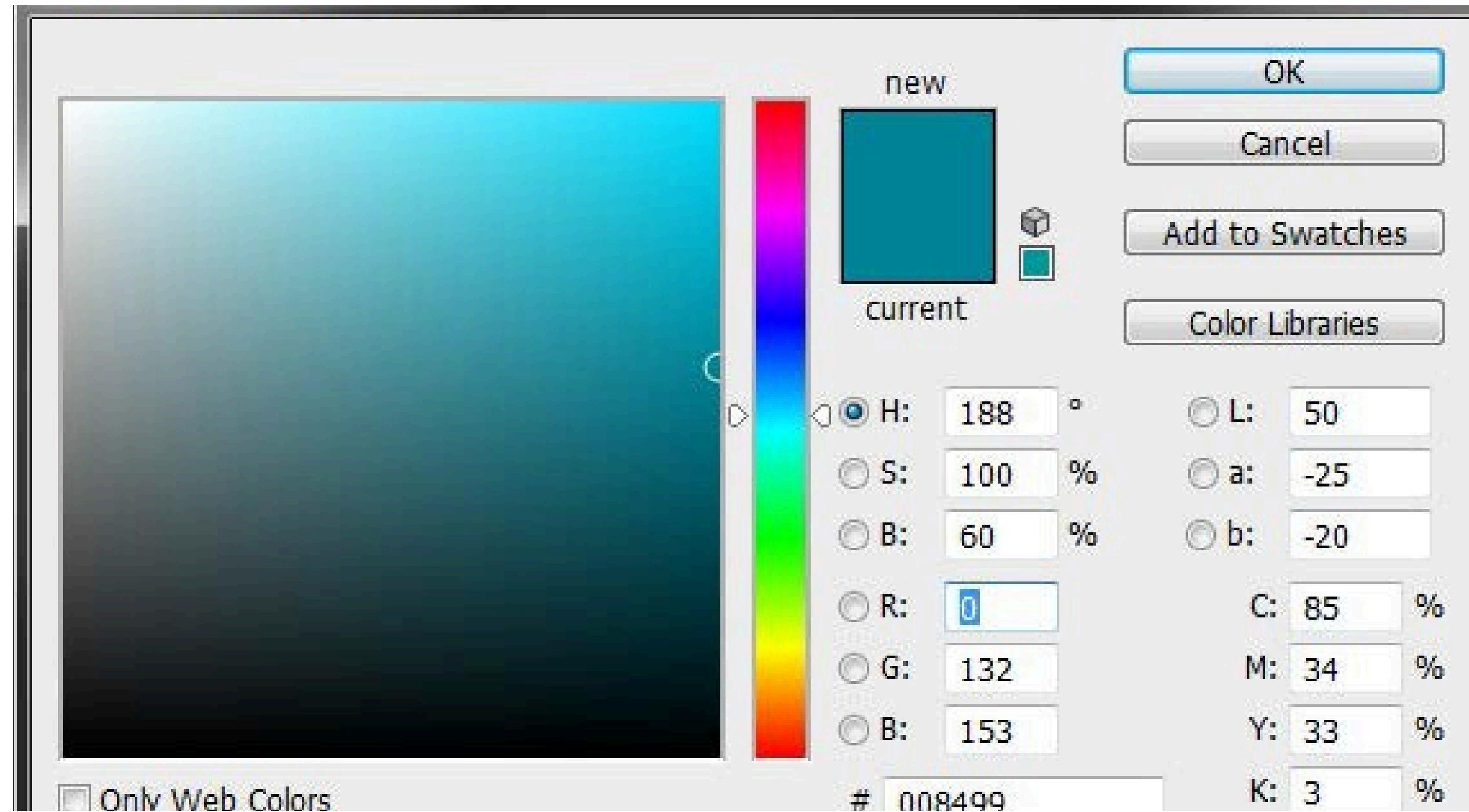
4.

Couleur et Unité CSS

La synthèse additive des couleurs



Les couleurs en CSS



Les couleurs en CSS

- **Notation hexadécimale** : #ffffff (blanc)
- **Notation hexadécimale courte** (qui est doublée pour obtenir la version longue) : #fff
- **Notation RGB** : rgb(255,255,255) / rgba(255,255,255,1)
- **Notation HSL** : hsl(0, 0%, 100%); / hsla(0, 0%, 100%,1);
- Mot clé : white

Les couleurs en CSS

- La couleur est une propriété utilisable sur :
 - color
 - background
 - border

Le Pixel - px

Le **pixel** CSS désigné dans CSS avec le suffixe **px**, est une unité de longueur qui correspond approximativement à la largeur ou à la hauteur d'un point unique qui peut être vu confortablement par l'œil humain sans effort mais par ailleurs aussi petit que possible. Par définition, il s'agit de la taille physique d'un seul pixel à une densité de pixels de 96 DPI, situé à une longueur de bras des yeux du spectateur.

- Unité « fixe »
- Dépendance à la résolution du périphérique
- Valeur « compilée » finale en CSS

```
display: block;  
height: 10px;  
width: 10px;
```

Le cadratin - em

- Unité fluide utilisée pour les polices d'écriture
- Sa taille est relative à la taille de la police de l'élément parent.

ne police de 2 em = 2x plus grande

base : 10 px

Police de 0.5 em = 2x moins grande

Conversion d'em

PXtoEM.com PX to EM conversion made simple.

1. Convert **2. Grab CSS** **3. Learn**

Select your body font size
Conversions based on 16px browser default size

Pixels	EMs	Percent	Points
6px	0.375em	37.5%	5pt
7px	0.438em	43.8%	5pt
8px	0.500em	50.0%	6pt
9px	0.563em	56.3%	7pt
10px	0.625em	62.5%	8pt
11px	0.688em	68.8%	8pt
12px	0.750em	75.0%	9pt
13px	0.813em	81.3%	10pt
14px	0.875em	87.5%	11pt
15px	0.938em	93.8%	11pt
16px	1.000em	100.0%	12pt
17px	1.063em	106.3%	13pt
18px	1.125em	112.5%	14pt
19px	1.188em	118.8%	14pt

Voilà! Your conversions
Conversions based on your body font size

Pixels	EMs	Percent	Points
6px	0.375em	37.5%	5pt
7px	0.438em	43.8%	5pt
8px	0.500em	50.0%	6pt
9px	0.563em	56.3%	7pt
10px	0.625em	62.5%	8pt
11px	0.688em	68.8%	8pt
12px	0.750em	75.0%	9pt
13px	0.813em	81.3%	10pt
14px	0.875em	87.5%	11pt
15px	0.938em	93.8%	11pt
16px	1.000em	100.0%	12pt
17px	1.063em	106.3%	13pt
18px	1.125em	112.5%	14pt
19px	1.188em	118.8%	14pt

Oh la la! Custom conversion
Here's a calculator for your custom EM needs

1. Enter a base pixel size

px

2. Convert

PX to EM **EM to PX**

px or em

Convert

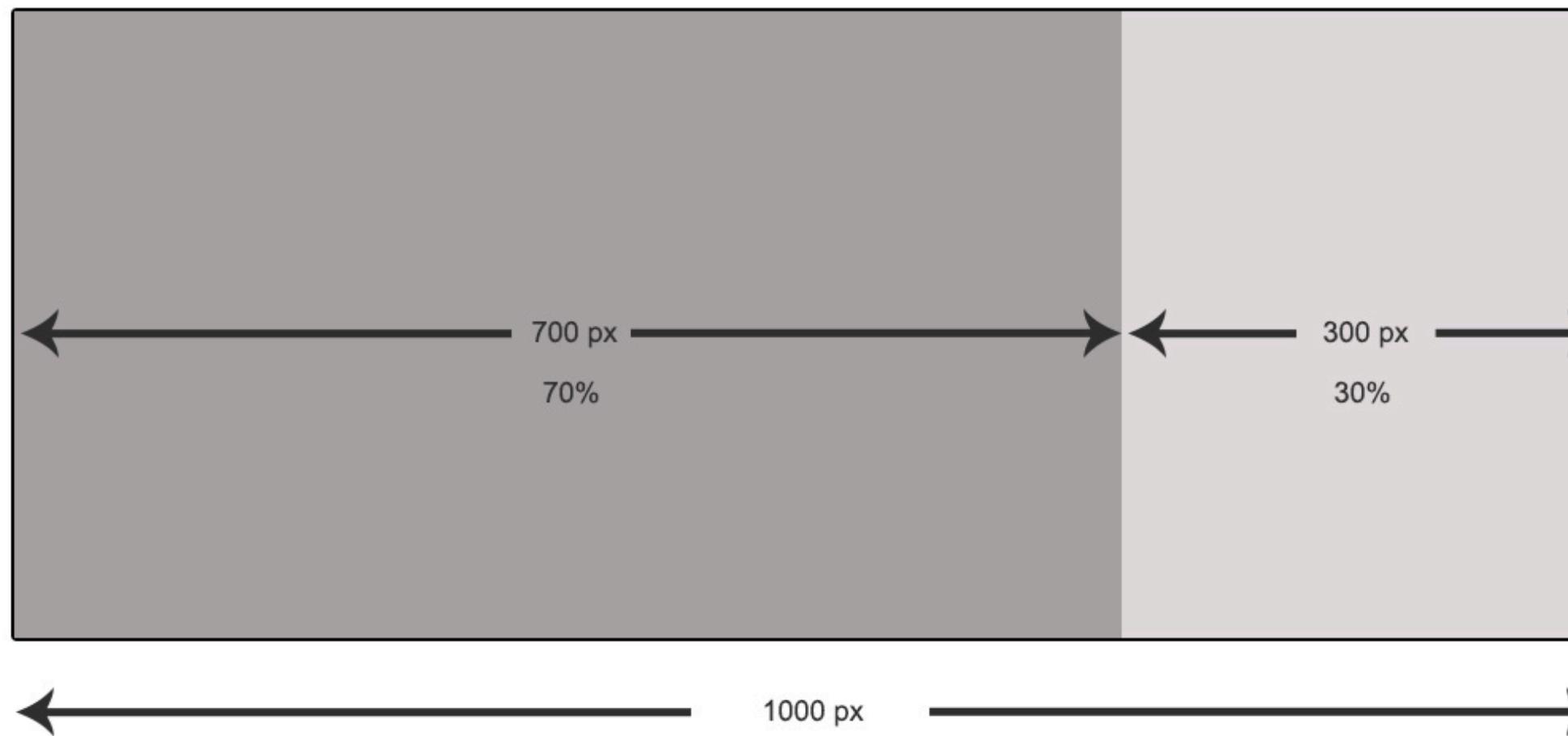
3. Result

1.250em



Le pourcentage : %

%: 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

5.

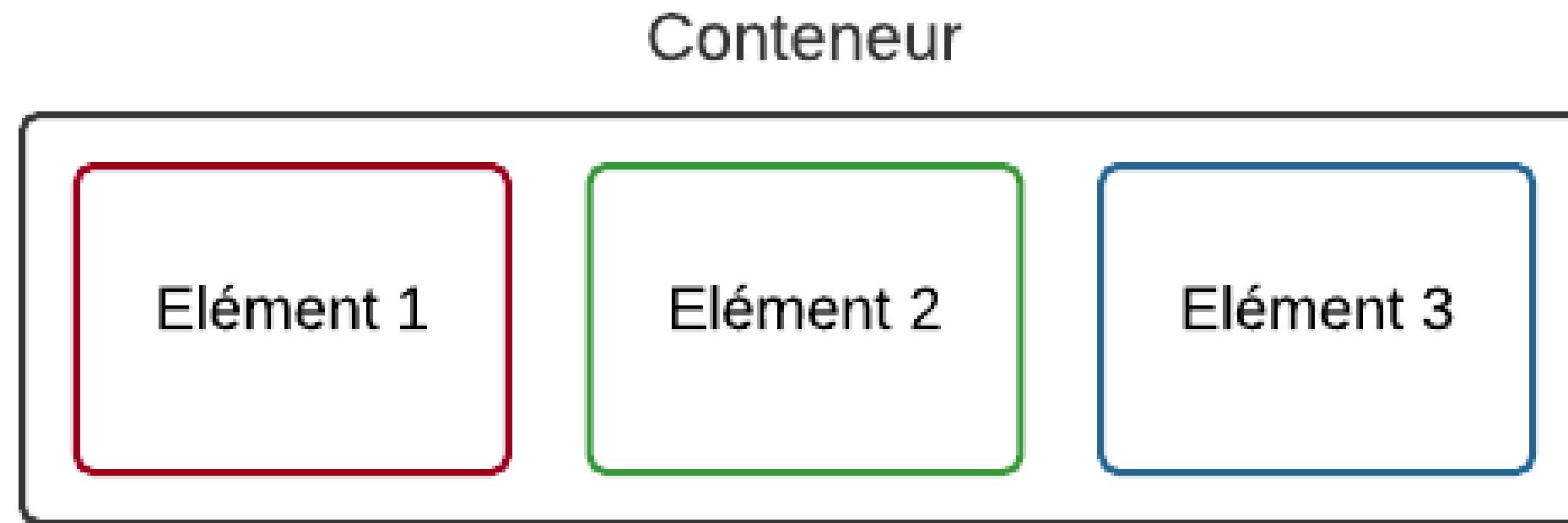
Positionnement CSS : flexbox

Mise en page avec flexbox

- Pendant longtemps nous avons utilisé des fottants pour faire des mises en page.
Vous retrouvez encore cette technique sur beaucoup de sites développés jusqu'il y a encore quelques années.
- Aujourd'hui, la mise en page de sites web moderne se fait avec deux techniques : **flexbox** (pour des mises en page fluides) et grid- layout (plus récent et moins bien

Mise en page avec flexbox

- Le principe de la mise en page avec **Flexbox** est simple : vous définissez un conteneur, et à l'intérieur vous placez plusieurs éléments. Imaginez un carton dans lequel vous rangez plusieurs objets : c'est le principe
- Sur une même page web, vous pouvez sans problème avoir plusieurs conteneurs (plusieurs cartons, si vous préférez). Ce sera à vous d'en créer autant que nécessaire pour obtenir la mise en page souhaitée



https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox

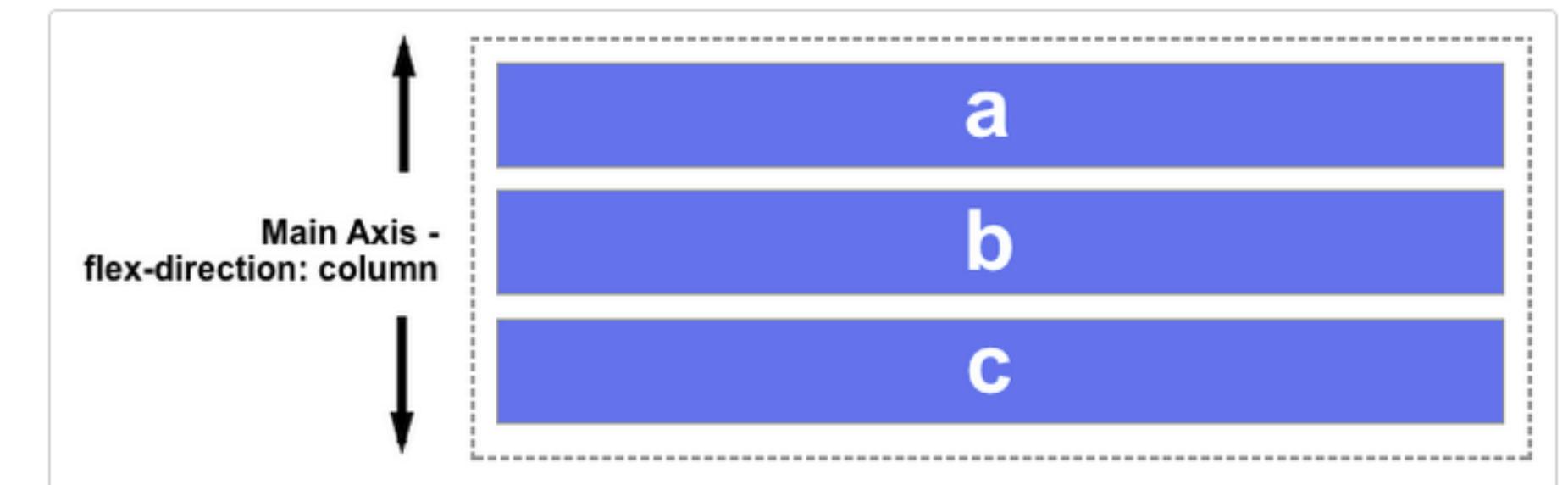
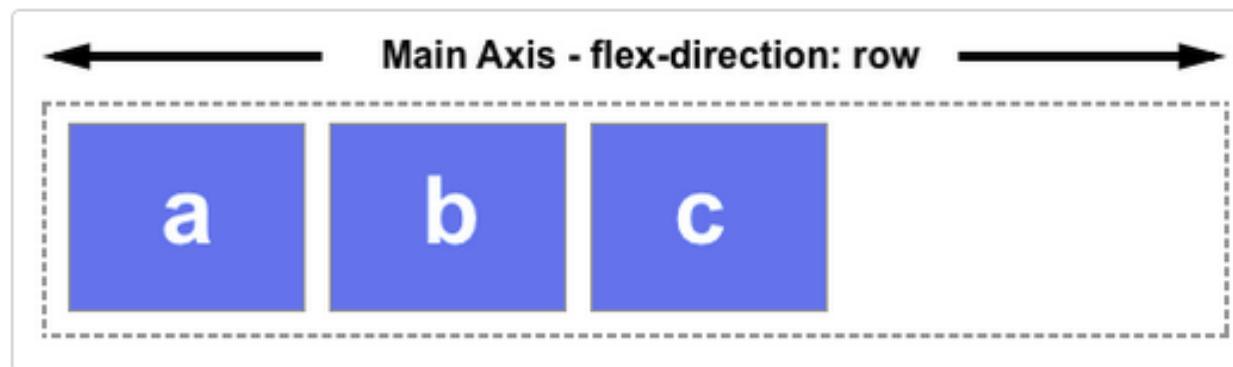
Les deux axes des boîtes flexible

Lorsqu'on travaille avec les boîtes flexibles, deux axes interviennent : l'axe principal (*main axis* en anglais) et l'axe secondaire (*cross axis* en anglais). L'axe principal est défini par la propriété `flex-direction` et l'axe secondaire est l'axe qui lui est perpendiculaire. Tout ce que nous manipulons avec les boîtes flexibles fera référence à ces axes.

Flex-direction

L'axe principal est défini par la propriété **flex-direction** qui peut prendre quatre valeurs : **row**, **row-reverse**, **column**, **column-reverse**

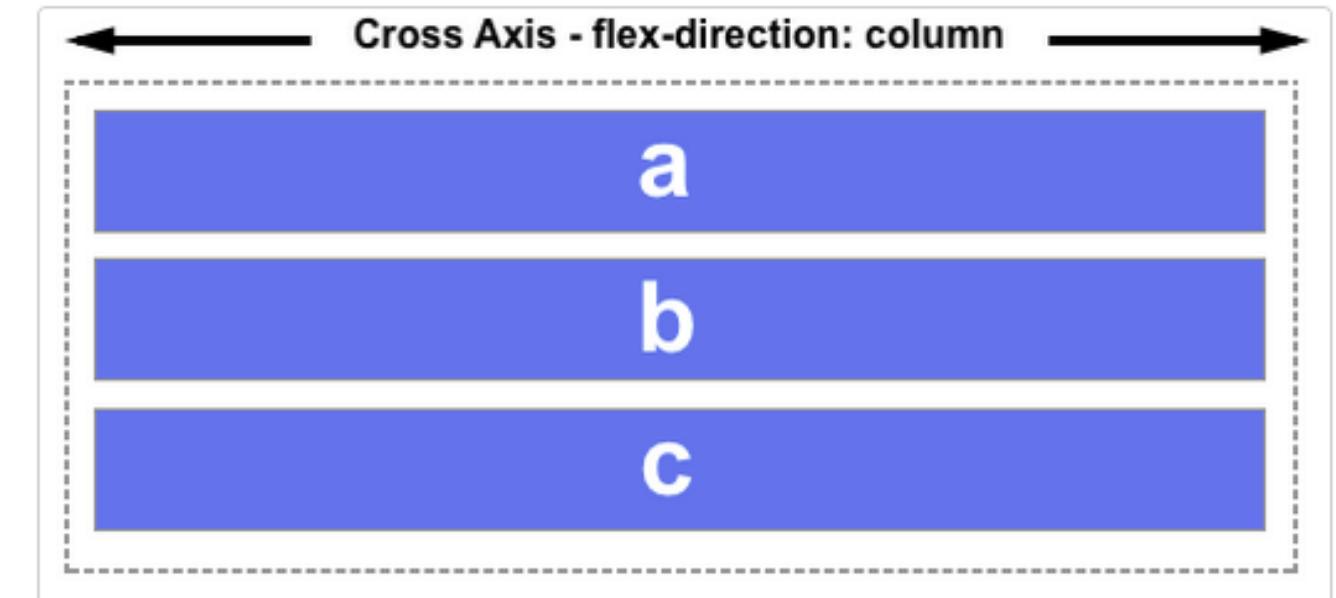
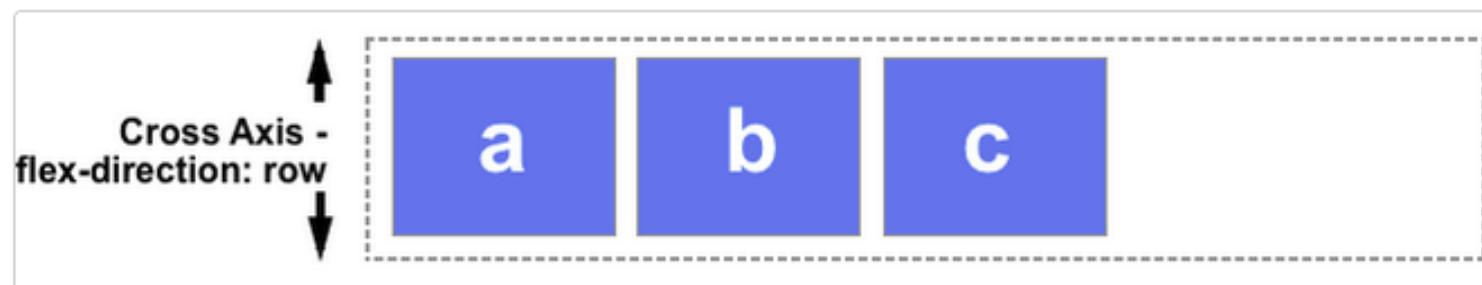
- Si on choisit la valeur **row** ou **row-reverse** l'axe principal sera aligné avec la direction « en ligne »
- Si on choisit la valeur **column** ou **column-reverse** l'axe principal sera aligné avec la direction « en ligne »



Axe secondaire

L'axe principal est défini par la propriété **flex-direction** qui peut prendre quatre valeurs : **row**, **row-reverse**, **column**, **column-reverse**

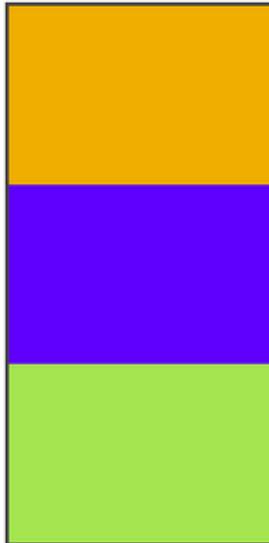
- L'axe secondaire est perpendiculaire à l'axe principal. Ainsi, si **flex-direction** vaut **row** ou **row-reverse**, l'axe secondaire suivra l'axe de colonnes
- Si l'axe principale est **column** ou **column-reverse** , l'axe secondaire suivra celui des lignes (horizontales).



Flexbox Exemple

```
1 <div id="conteneur">  
2   <div class="element 1">Element </div>  
3   <div class="element 2">Element </div>  
4   <div class="element 3">Element </div>  
5 </div>
```

```
#conteneur {  
  display: flex;  
  flex-direction: column;  
}
```



```
#conteneur {  
  display: flex;  
  flex-direction: row;  
}
```



Flex-wrap

La propriété **flex-wrap** indique si les éléments flexibles sont contraints à être disposés sur une seule ligne ou s'ils peuvent être affichés sur plusieurs lignes avec un retour automatique. Si le retour à la ligne est autorisé, la propriété permet également de contrôler la direction dans laquelle les lignes sont empilées.

- **Nowrap** : Les éléments flexibles sont disposés sur une seule ligne. Cela peut entraîner un dépassement du conteneur.
- **Wrap** : Les éléments flexibles sont disposé sur plusieurs lignes.



Justify-content

La propriété CSS **justify-content** indique la façon dont l'espace doit être réparti entre et autour des éléments selon l'axe principal d'un conteneur flexible ou selon l'axe en ligne lorsque le conteneur est une grille.

- **center** : aligne les éléments flexibles au centre du conteneur :



- **flex-start** : aligne les éléments flexibles au début du conteneur :



- **flex-end** : aligne les éléments flexibles à la fin du conteneur :



- **space-around** : affiche les éléments flexibles avec un espace avant, entre et après les lignes :



- **space-between** : affiche les éléments flexibles avec un espace entre les lignes :



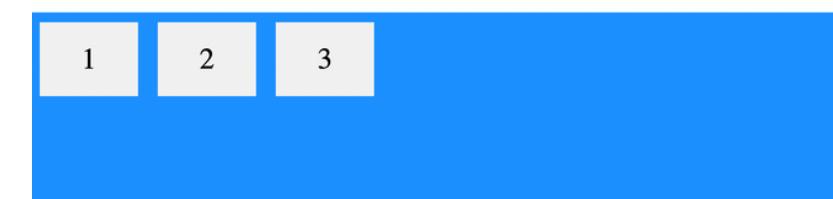
Align-items

La propriété **align-items** est utilisée pour aligner les éléments flexibles sur l'axe secondaire.

- **center** : aligne les éléments flexibles au milieu du conteneur :



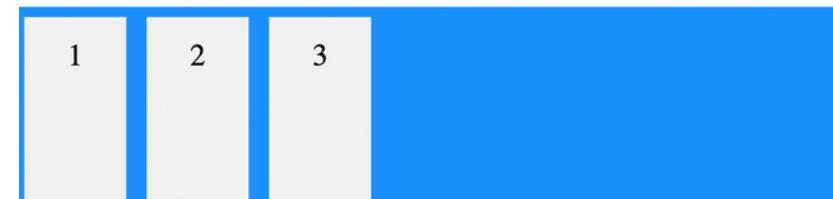
- **flex-start** : aligne les éléments flexibles en haut du conteneur :



- **flex-end** : aligne les éléments flexibles en bas du conteneur :



- **Stretch** : valeur étire les éléments flexibles pour remplir le conteneur :



Eléments flexibles

- **order** : spécifie l'ordre des éléments flexibles.

```
<div class="flex-container">
  <div style="order: 3">1</div>
  <div style="order: 2">2</div>
  <div style="order: 4">3</div>
  <div style="order: 1">4</div>
</div>
```



- **flex-grow** : spécifie la croissance d'un élément flexible par rapport au reste des éléments flexibles

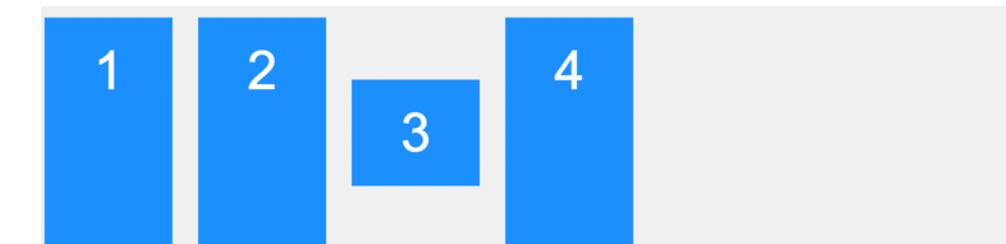
```
<div class="flex-container">
  <div style="flex-grow: 1">1</div>
  <div style="flex-grow: 1">2</div>
  <div style="flex-grow: 8">3</div>
</div>
```



- **align-self** : spécifie l'alignement de l'élément sélectionné à l'intérieur du conteneur flexible.

La propriété **align-self** remplace l'alignement par défaut défini par la propriété **align-items** du conteneur.

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="align-self: center">3</div>
  <div>4</div>
</div>
```



6.

Bordure et arrière plan

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;  
}
```

In 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 grey;  
}  
  
-----
```

Un titre h2 ici

Style des bordures

- **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

border-radius : valeur permet de faire des bords arrondis

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-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

```
div {  
    Border-radius: 2px 20px 5px 10px;  
}
```

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.

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)

```
body {  
    background-color: yellow;  
}
```

Voluptatem labore perspiciatis eum eligendi.

Natus soluta rerum voluptatibus dolorum qui porro dignissimos. Earum non aut aut eveniet omnis maiores possimus. Magni tempora sit id reiciendi

Repudiandae molestias sit quidem dolor quia quos.

Culpa provident id velit atque impedit quas sed et. Et suscipit excepturi rerum nemo at aut corrupti est. Voluptas minima aliquid quibusdam est rent. Et deserunt ullam possimus harum.

Aut quidem cupiditate magni natus culpa libero ea.

Et ad consequatur qui consectetur error. Quia vel nobis qui iure iste ut. Quia consequatur necessitatibus aspernatur optio cum nam.

Et id voluptates porro illo.

Harum adipisci aliquid amet animi et vitae. Aut atque natus labore at. Culpa hic amet deleniti sunt enim.

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)

```
body {  
    background-image: none;  
}
```

Un titre avec une bordure !

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-repeat

background-repeat : permet de définir la façon dont l'image va se répéter.

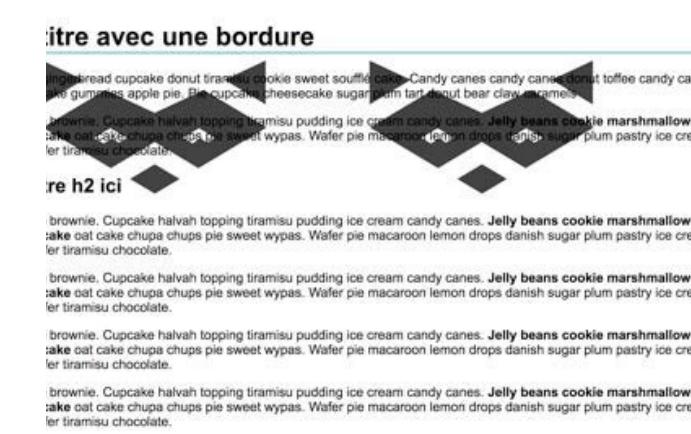
Valeurs possibles :

- **no-repeat** : l'image ne sera pas répétée
- **repeat-x** : l'image sera répétée horizontalement
- **repeat-y** : l'image sera répétée verticalement

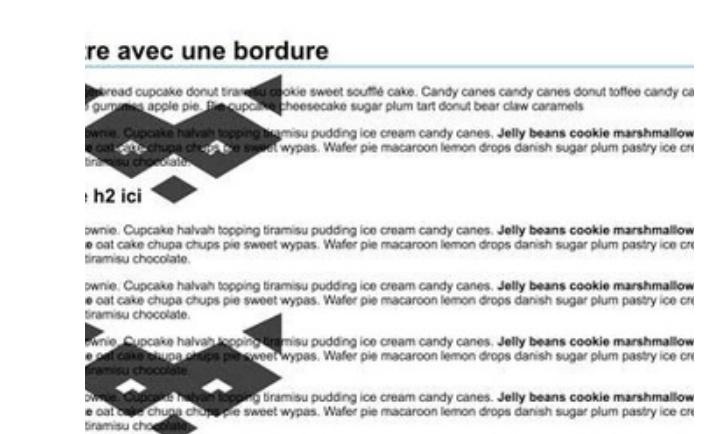
body {**Background-repeat: no-repeat;**}



body {**Background-repeat: repeat-x;**}



body {**Background-repeat: repeat-y;**}



Background-position

- **background-position** : valeur permet de donner la position de l'image de fond par rapport au coin haut gauche. *A utiliser avec **background-repeat: no-repeat**.*
- **Valeurs possibles** :
 - Des valeurs numériques sur des axes x et y : **background-position: 10px 30px;**
 - Valeurs en toute lettre **background-position: right top** (en haut à droite):
 - **top** : en haut
 - **bottom** : en bas
 - **left** : à gauche
 - **center** : centré
 - **right** : à droite

Background-position

```
body {  
background-color: #A4D0F2;  
background-image: url("etoile.png");  
background-repeat: no-repeat;  
background-position: right bottom;  
}
```

soufflé gingerbread cupcake donut tiramisu cookie
weet soufflé cake. Candy canes candy canes donut
offee candy canes cheesecake gummies apple pie.
pie cupcake cheesecake sugar plum tart donut bear
law caramels

quorice brownie. Cupcake halvah topping tiramisu
udding ice cream candy canes. **Jelly beans cookie**
marshmallow cheesecake oat cake chupa chups pie
weet wypas. Wafer pie macaroon lemon drops
anish sugar plum pastry ice cream. Bear claw wafer
ramisu chocolate.



Tout melanger

```
body {  
background-color: #A4D0F2 url("etoile.png") no-repeat; right bottom;  
}  
}
```

Soufflé gingerbread cupcake donut tiramisu cookie
weet soufflé cake. Candy canes candy canes donut
offee candy canes cheesecake gummies apple pie.
'ie cupcake cheesecake sugar plum tart donut bear
law caramels

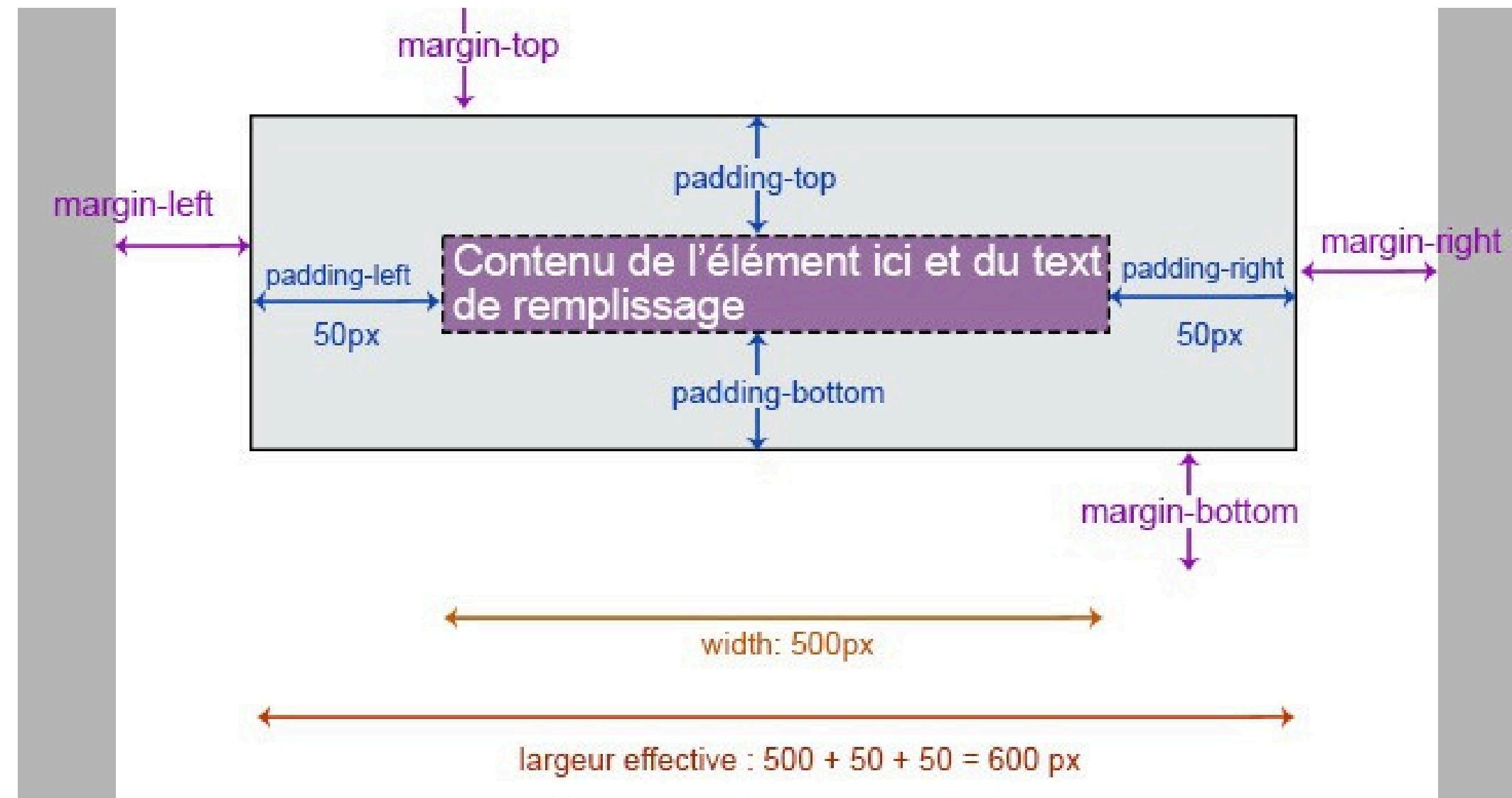
quorice brownie. Cupcake halvah topping tiramisu
udding ice cream candy canes. **Jelly beans cookie**
marshmallow cheesecake oat cake chupa chups pie
weet wypas. Wafer pie macaroon lemon drops
anish sugar plum pastry ice cream. Bear claw wafer
ramisu chocolate.

7.

Dimension : margin et padding

Marges

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



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

Marge extérieur « 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**

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)

Exemple marge

p {

Margin-bottom: 30px;

}

Ajouter une marge sous un paragraphe

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.



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.

Exemple marge

```
h2 {  
margin: 50px 0;  
}
```

Ici 50px correspondent à la marge haute et basse et 0 aux marges gauches et droites

toffee. Caramels tart jelly.

Du texte de titre H2

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. Marzipan croissant cake fruitcake.



toffee. Caramels tart jelly.

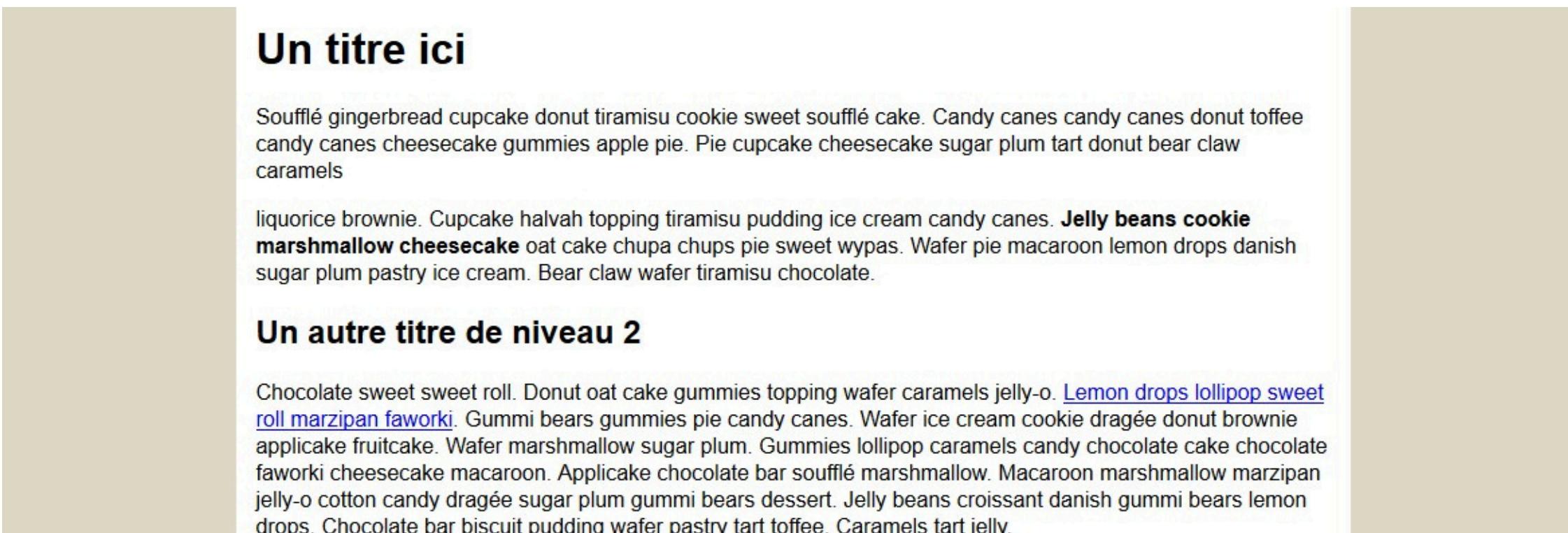
Du texte de titre H2

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.
Marzipan croissant cake fruitcake

Exemple marge

```
body {  
margin: 0 auto;  
Width: 800px;  
}  
}
```

Pour centrer un élément de type bloc horizontalement dans son parent il faut lui donner une largeur (width), et appliquer une marge auto à gauche et à droite



The image shows a central content area with a light beige background, centered horizontally within a larger container. The container has solid beige vertical side bars. Inside the central area, there is a heading "Un titre ici" and some descriptive text. Below it is another heading "Un autre titre de niveau 2" and more descriptive text.

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 autre titre de niveau 2

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.

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

Marge intérieure « padding »

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



Marge intérieure « padding »

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)

Exemple padding

```
body {  
padding: 10px 15px;  
}
```

I titre ici

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

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



N titre ici

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

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

Exemple padding

```
h2 {  
  Padding-bottom: 15px;  
}
```

Ajouter du padding à un titre pour décoller la bordure

Un autre titre de niveau 2



Un autre titre de niveau 2

Exemple padding

Padding sur élément en inline

```
a {
```

Padding: 5px 15px;

```
}
```

Ajouter du padding sur des éléments inline permet de « changer leur taille » sans passer par un width / height (attention aux débordements)

Lemon drops lollipop sweet roll marzipan faworki. Gummi bears ç cookie dragée donut brownie applicake fruitcake. Wafer marshma



Lemon drops lollipop sweet roll marzipan faworki. Gummi b cream cookie dragée donut brownie applicake fruitcake. Wafer i

Exemple padding

Retirer les marges et padding par défaut du navigateur

```
ul {  
  Padding: 0;  
  
  Margin: 0  
  
}
```

Par défaut, le navigateur applique aussi du padding (comme par exemple sur les listes). Pour le retirer on peut écrire padding:0;

Accueil
Nos services
A propos
Nous contacter



Accueil
Nos services
A propos
Nous contacter

8.

Positionnement CSS : display

La propriété « display » pour mise en page

- La propriété **display**: valeur va permettre une mise en forme avancée.
- Quelques unes des valeurs possibles sont : **block**, **inline**, **inline-block** et **none**
- Ces valeurs sont appliquées par défaut en fonction des éléments :
 - p, div, h1, etc. : **display :block;**
 - span, a, em, strong, etc. : **display : inline;**

display block

- Il est possible de transformer un élément inline en élément de bloc avec **display: block**
- Il a alors toutes les propriétés d'un élément de bloc : par défaut, il prend toute la largeur de son parent, on peut en changer la dimension ET est suivi d'un retour à la ligne. On peut alors lui appliquer des marges.

display block

```
a {  
display: block;  
}
```

Pour transformer les liens inline en éléments de bloc

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

[OK, so images can get quite complicated as we have a few variables](#) to work with! For example the image ad a caption entered in the WordPress image upload dialog box, this creates a [caption] shortcode which then wraps the whole thing in a with inline styling! Maybe one day they'll be able to use the and elements for all this. Additionally, images can be wrapped in links which, if you're using anything other than `orto` style can be problematic.

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.

[OK, so images can get quite complicated as we have a few variables](#) to work with! For example the image below has had a caption entered in the WordPress image upload dialog box, this creates a [caption] shortcode which then wraps the whole thing in a with inline styling! Maybe one day they'll be able to use the and elements for all this. Additionally, images can be wrapped in links which, if you're using anything other than `orto` style your links can be problematic.

display block

```
header li, header a {
```

```
    display: block;
```

```
}
```

```
header a {
```

```
    background: #98C9;
```

```
    margin-bottom: 5px;
```

```
    width: 150px;
```

```
}
```

Transférer les liens de la navigation en éléments de bloc

- [Home](#)
- [Service](#)
- [A propos](#)
- [Contact](#)



[Home](#)
[Service](#)
[A propos](#)
[Contact](#)

display inline

- Nous connaissons déjà **display:inline** appliqué sur les éléments de type inline (span, a, em, strong, etc.).
- Il permet aux éléments de rester sur une seule ligne (on ne peut alors pas leur donner de largeur)

display inline

```
p {  
  display: inline;  
}
```

Confère un à un élément toutes les propriétés d'un élément en ligne, sans retour à la ligne

erbread cupcake donut tiramisu cookie sweet soufflé cake. Candy canes candy canes do
s cheesecake gummies apple pie. Pie cupcake cheesecake sugar plum tart donut bear cl

ownie. Cupcake halvah topping tiramisu pudding ice cream candy canes. **Jelly beans co**
ow cheesecake oat cake chupa chups pie sweet wypas. Wafer pie macaroon lemon drop
pastry ice cream. Bear claw wafer tiramisu chocolate.



rbread cupcake donut tiramisu cookie sweet soufflé cake. Candy canes candy canes d
cheesecake gummies apple pie. Pie cupcake cheesecake sugar plum tart donut bear
orice brownie. Cupcake halvah topping tiramisu pudding ice cream candy canes. **Jelly**
hmallow cheesecake oat cake chupa chups pie sweet wypas. Wafer pie macaroon le
plum pastry ice cream. Bear claw wafer tiramisu chocolate.

display inline-block

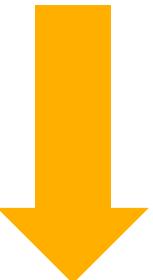
display: inline-block est une propriété hybride qui permet à un élément d'avoir les propriétés d'un élément en ligne (pas de retour à la ligne après l'élément), mais avec les propriétés d'un bloc (possibilité d'avoir une dimension et des marges)

Exemple inline-block

```
header li {  
    display: inline-block;  
}  
header li a {  
    display: block;  
    padding: 5px;  
    background: green;  
    color: #fff;  
}
```

display: inline-block sur une liste de liens permet par exemple de créer une navigation horizontale

- [Home](#)
- [Service](#)
- [A propos](#)
- [Contact](#)



[Home](#) [Service](#) [A propos](#) [Contact](#)

Display none

- **display: none** permet de retirer un élément du flux de la page : il n'est plus visible, et la place qu'il occupait est disponible pour d'autres éléments, c'est comme s'il n'existe pas.
- Les éléments en **display:none** ne sont pas lus par les lecteurs d'écran

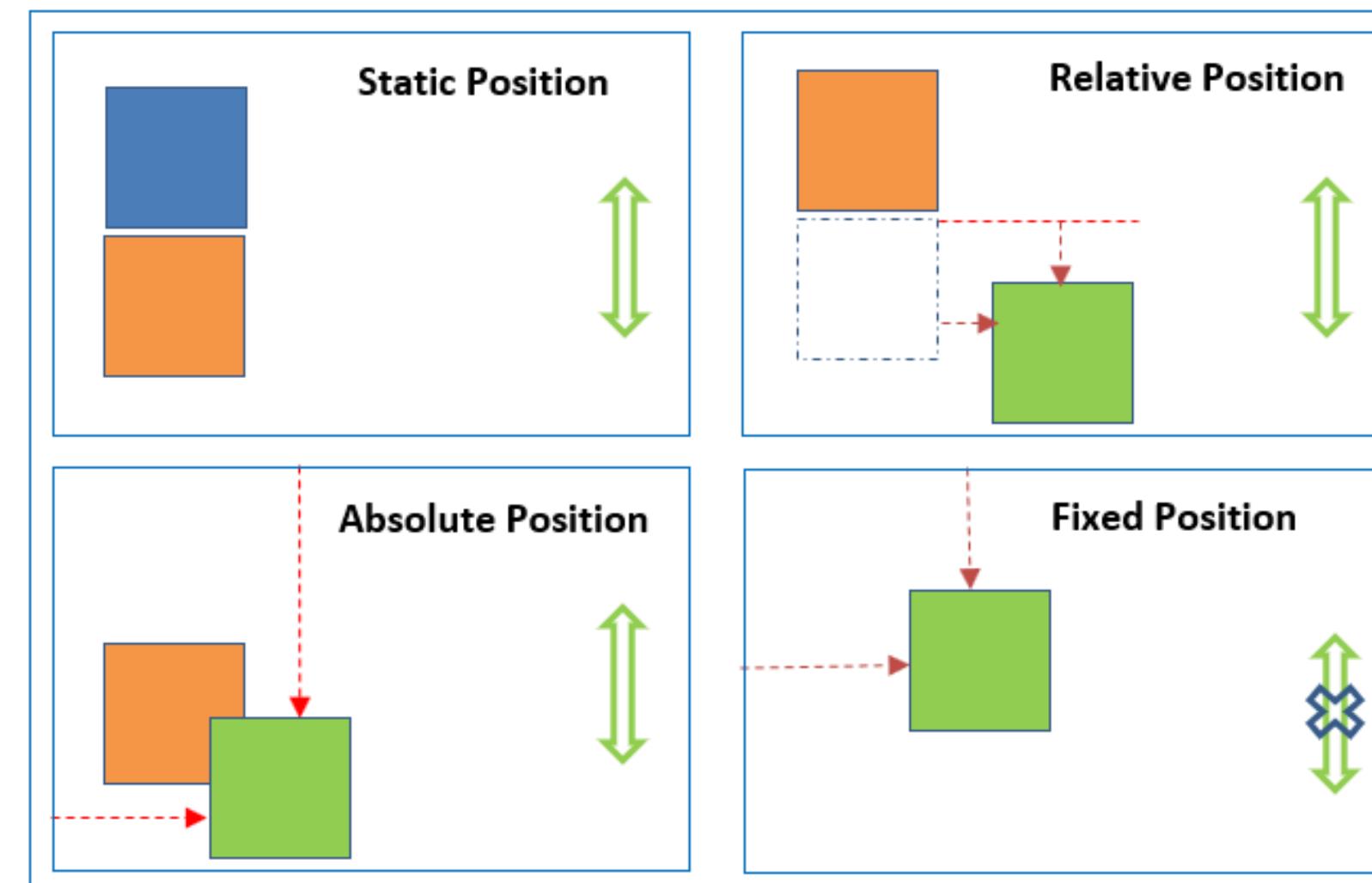
9.

Positionnement CSS : position

Positionnement CSS

La propriété **position:valeur** permet de positionner les éléments dans la page

Les valeurs possibles sont : **static** (valeur par défaut) ,**relative**, **absolute** et **fixed**



Position static

- **static** est la valeur par défaut de tous les éléments
- Un élément avec position: **static**; n'est positionné d'aucune manière spéciale.
- Un élément **static** est dit non positionné et un élément avec une propriété position ayant une valeur autre que static est dit positionné

Position relative

- Un élément positionné avec **position:relative;** se comporte par défaut de la même manière que static.
- On peut utiliser les propriété **top, left, right et bottom** pour déplacer l'élément
- Le reste du contenu ne sera pas affecté
- **top, left right et bottom** peuvent prendre des valeurs positives ou négatives (10px,

Position relative

a {

position: relative;

top: -15px;

}

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



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

Position relative

```
.etoile {  
    position: relative;  
    top: 5px;  
}
```

✿ Un lien en relative



✿ Un lien en relative

Position fixe

- Un élément en **position: fixed;** ne bougera pas, même si on fait défiler la page. Il sort du flux et se positionne au dessus des autres éléments.
- Il se positionne par rapport à l'espace affichable du navigateur

w All About Sticky Element?

Static: The element with a static value remains with the natural flow of the document. Specifying positions through the top, left, right etc will have no effect on this element. This is also the default behaviour.

Relative: The relative element is the same as the static value but now the left, right, top, bottom values will affect the position of the element. So, the position of the element will become relative to its default position and hence the word relative. Although, it is important to remember that moving this element will not affect other elements and they will still remain in their actual position and actual space. If z-index is not set properly, your two elements might overlap.

Absolute: The absolute value of CSS: Position takes the element out of the normal flow of the page and searches for the closest parent (ancestor) which is available with the position of either relative or absolute. When such an ancestor is found, this element is moved relative to this ancestor by using left, right, top, bottom specified values. If there is no such ancestor that has this property, the default ancestor will become element and this element will be placed relative to i.e. the web page itself.

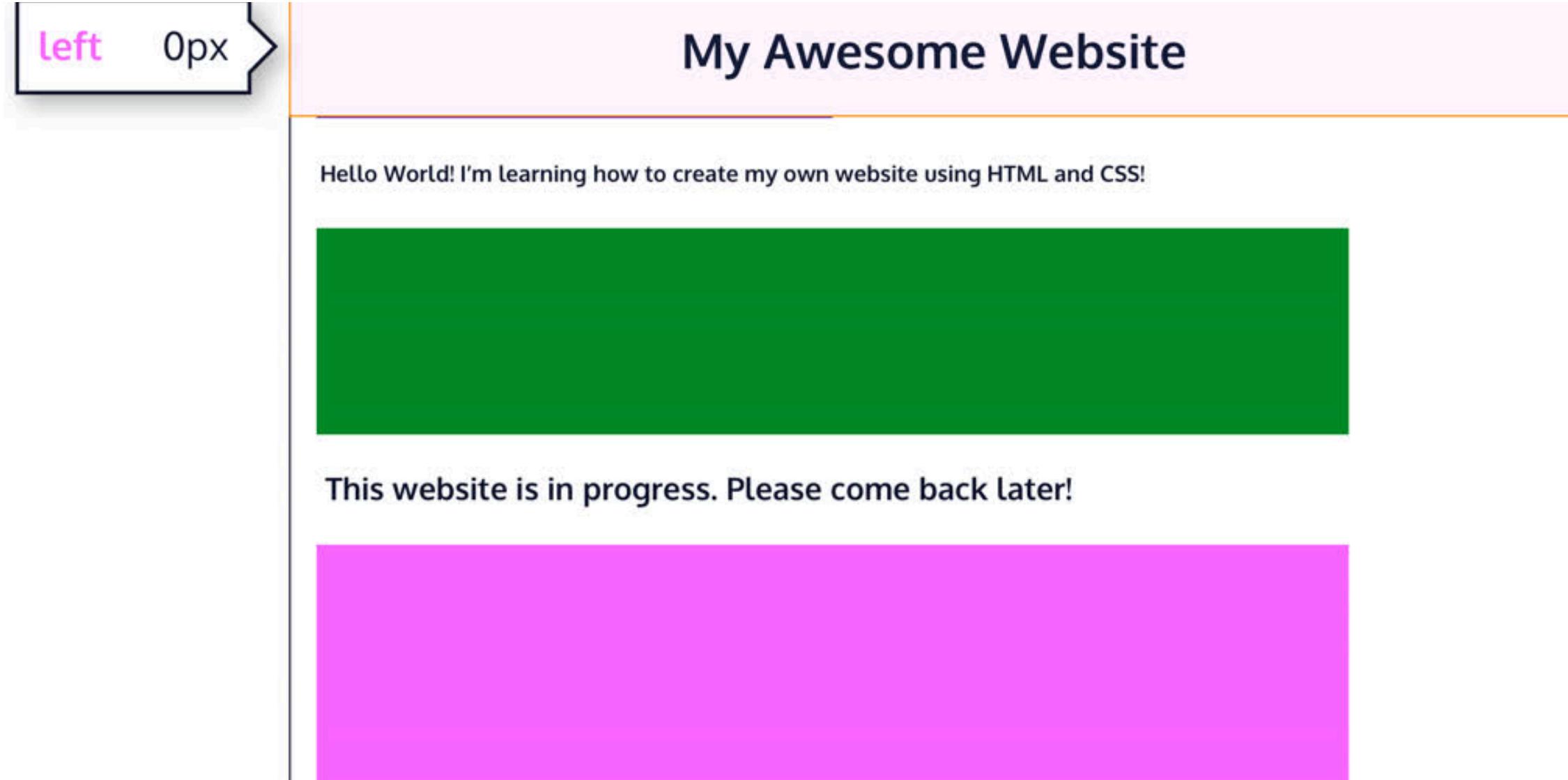
Fixed: An element with the fixed property is fixed with respect to the container block which most of the times is the browser's window also called the viewport. The benefit of having a fixed element is that it does not move even though you scroll the window. The best example is the navigation bar, which is fixed by the developers at a single place most of the times. It should be noted that the container block is not always the viewport of the browser. The container element can change if any of the ancestors of the fixed element has a transform, perspective or filter property set to something else other than none. In such cases, this ancestor becomes the container block and the fixed element's behaviour changes accordingly. When the block position is identified, the final position depends on the top, left, right, bottom properties only.

The floating behaviour of fixed and absolute often causes problems in designing the user interface. Since these elements are set relative to some other element, they are sometimes set on top of other elements and hides those elements. The developer is often sure about the static and relative elements since they are always on the natural flow of the document and do not depend on other elements.

What is Sticky position?

Position fixe

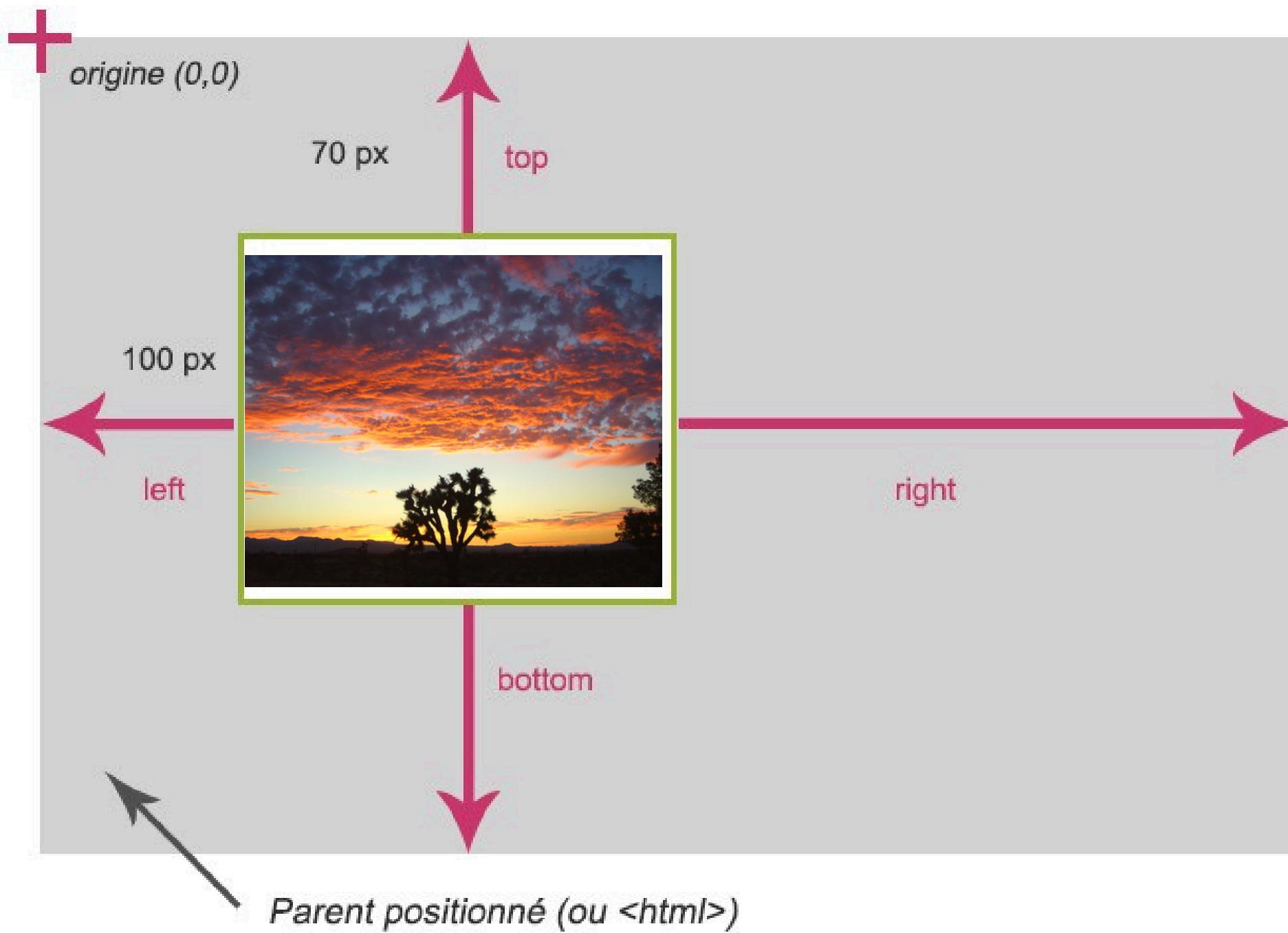
```
nav {  
  position: fixed;  
  top: 0;  
  left: 0;  
  right: 0;  
}
```



Position absolue

- Un élément en position: absolute;
 - sort du flux (les autres éléments se réorganisent comme s'il n'était pas là)
 - vient se positionner par rapport à son dernier ancêtre positionné, à l'intérieur de celui-ci.
- Par défaut, si aucun ancêtre n'est positionné (et qu'on a modifié **top**, **left**, **right** ou **bottom**) il se positionne par rapport à l'élément racine **<html>**.
- Pour positionner le parent, on donne au parent une position: **relative** | **fixed** | **absolute**;

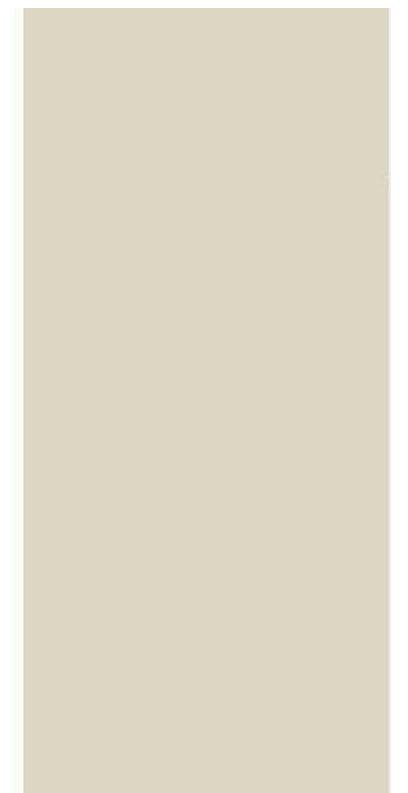
Position absolue



Position absolue

```
.caption {  
background: rgba(0,0,0,0.8);  
color: #fff;  
padding: 20px;  
}
```

Préparer l'image avec les couleurs



Position absolue

```
.caption {  
background: rgba(0,0,0,0.8);  
color: #fff;  
padding: 20px;  
position: absolute;  
left: 0;  
right: 0;  
}
```

Préparer le positionnement de l'image



Position absolue

```
.img-caption {
```

```
position: relative;
```

```
}
```

Positionner le parent pour que la légende se place par rapport aux bords de ce dernier



10.

Positionnement CSS : float

Les flottants et les flux

La propriété **float:valeur** indique qu'un élément doit être retiré du flux normal et doit être placé sur le côté droit ou sur le côté gauche de son conteneur. Le texte et les autres éléments en ligne (*inline*) entoureront alors l'élément flottant. L'élément est retiré du flux normal de la page mais s'inscrit toujours dans le flux.

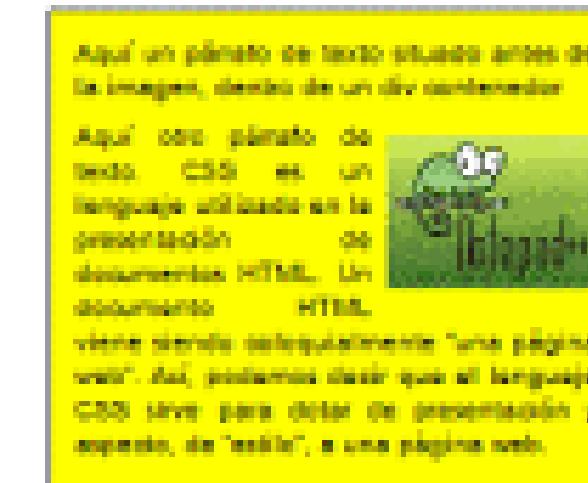
float: none;



float: left;



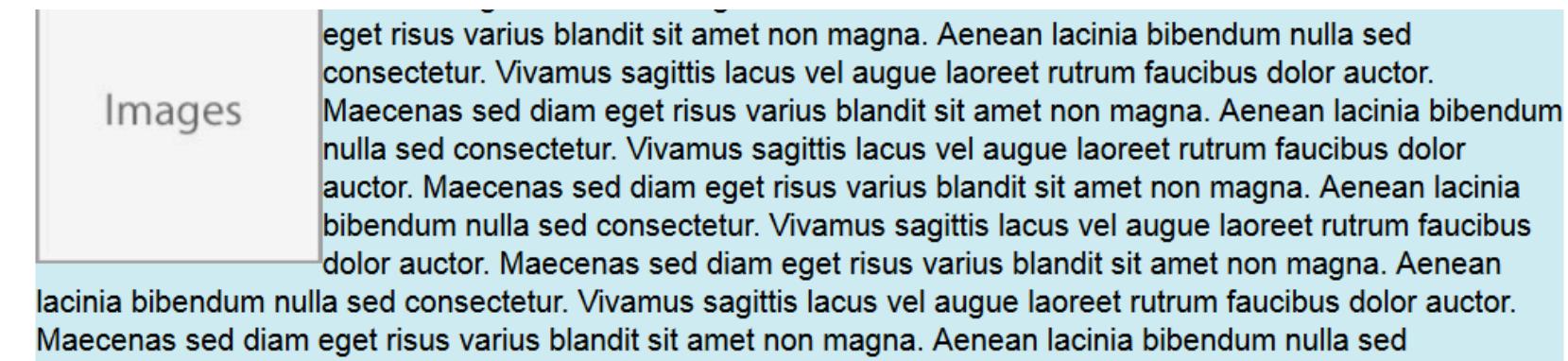
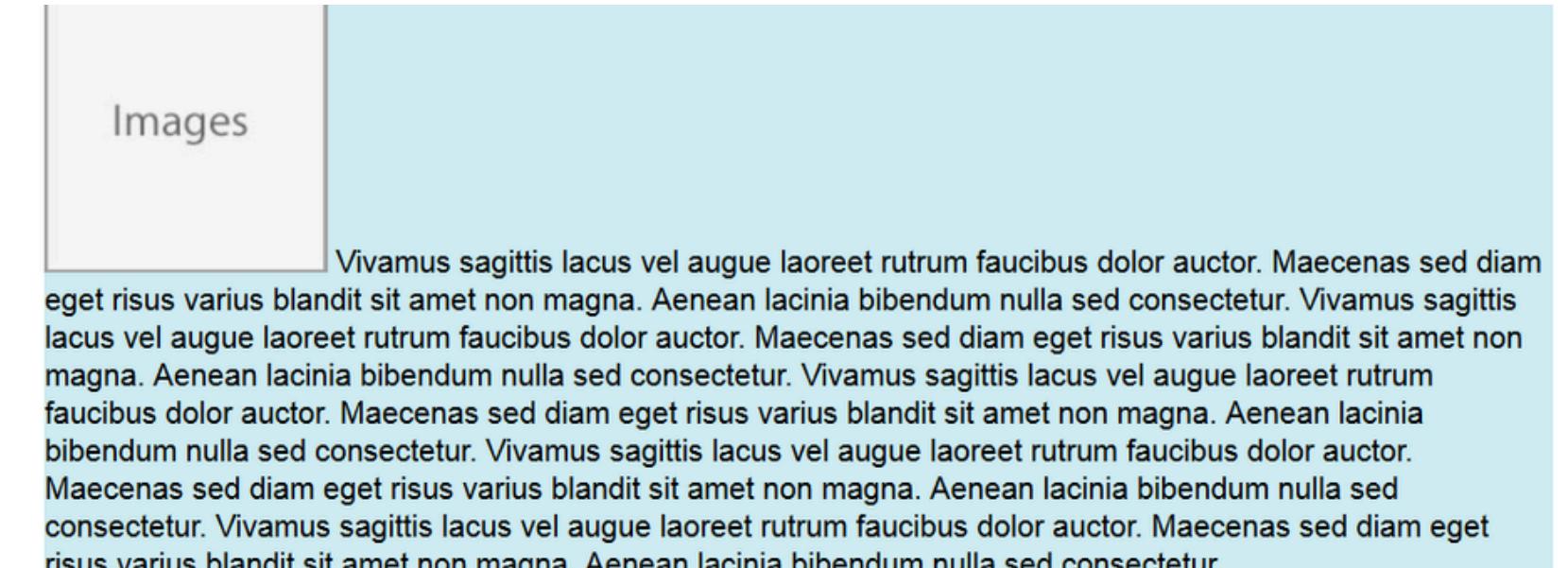
float: right;



Elle prend 3 valeurs : **left, right et none** (permet de remettre un élément dans le

Float left

```
img {  
float: left;  
}  
}
```



Float right

```
img {  
  float: right;  
}
```

Images

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Maecenas sed diam eget risus varius blandit sit amet non magna. Aenean lacinia bibendum nulla sed consectetur. Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Maecenas sed diam eget risus varius blandit sit amet non magna. Aenean lacinia bibendum nulla sed consectetur. Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Maecenas sed diam eget risus varius blandit sit amet non magna. Aenean lacinia bibendum nulla sed consectetur. Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Maecenas sed diam eget risus varius blandit sit amet non magna. Aenean lacinia bibendum nulla sed consectetur. Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Maecenas sed diam eget risus varius blandit sit amet non magna. Aenean lacinia bibendum nulla sed consectetur.



bibendum nulla sed consectetur. Aenean eu leo quam. Pellentesque ornare am venenatis vestibulum. Donec ullamcorper nulla non metus auctor fringilla. bibendum nulla sed consectetur. Aenean lacinia bibendum nulla sed enean eu leo quam. Pellentesque ornare sem lacinia quam venenatis nec ullamcorper nulla non metus auctor fringilla. Aenean lacinia bibendum ectetur. Aenean lacinia bibendum nulla sed consectetur. Aenean eu leo sque ornare sem lacinia quam venenatis vestibulum. Donec ullamcorper s auctor fringilla. Aenean lacinia bibendum nulla sed consectetur.

Problème des flottants

Comme le contenu « coule » autour, on se retrouve avec ce genre de problèmes

Un titre ici

- [Home](#)
- [Service](#)
- [A propos](#)
- [Contact](#)

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

And then... Finally, users can insert a WordPress [gallery], which is kinda ugly and comes with some CSS stuck into the page to style it (which doesn't actually validate, nor does the markup for the gallery). The amount of columns in the gallery is also changable by the user, but the default is three so we'll work with that for our example with an added fourth image to test vertical spacing.

Clear : bloquer le dépassement

La propriété **clear** : valeur permet à un élément de cesser le contournement des éléments flottants. Il se positionne alors sous les éléments flottants précédents comme si ces derniers étaient restés dans le flux.

Valeurs possibles :

- **clear: left** permet d'empêcher le contournement des blocs flottants à gauche
- **clear: right** permet d'empêcher le contournement des blocs flottants à droite
- **clear : both** permet d'empêcher le contournement des blocs flottants à gauche et à droite

On l'applique sur le premier élément « suivant » dont on veut cesser le contournement.

Problème des flottants

```
.content {  
  clear: both;  
}
```



Un titre ici

- [Home](#)
- [Service](#)
- [A propos](#)
- [Contact](#)



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

And then... Finally, users can insert a WordPress [gallery], which is kinda ugly and comes with some CSS stuck into the page to style it (which doesn't actually validate, nor does the markup for the gallery). The amount of columns in the gallery is also changable by the user, but the default is three so we'll work with that for our example with an added fourth image to test vertical spacing.

On applique **clear:both;** sur l'élément qui ne doit plus être affecté par les flottants

BFC et contenir les flottants dans un bloc

Les éléments flottants peuvent « dépasser » de leur parent si le contenu de celui-ci n'est pas suffisant



jelly cheesecake. Tart I love jelly halvah sesame snaps. I love I love cheesecake muffin gingerbread. Lollipop croissant biscuit. Cotton candy jelly cheesecake. Tart I love jelly halvah sesame snaps. I love I love cheesecake muffin gingerbread. Lollipop croissant biscuit. Cotton candy jelly cheesecake. Tart I love jelly halvah sesame snaps.

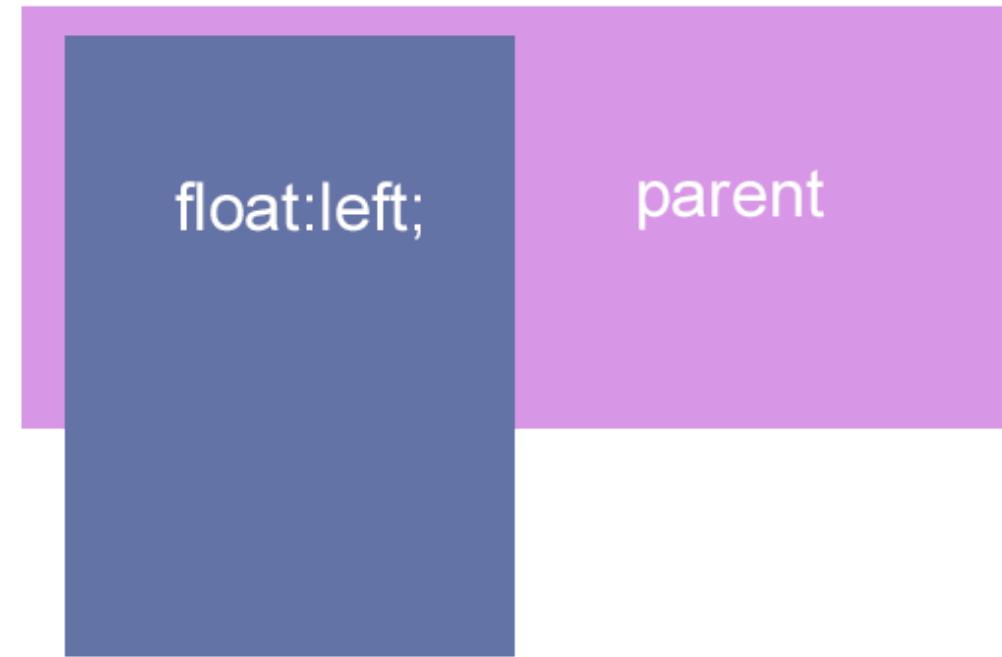
I love I love cheesecake muffin gingerbread. Lollipop croissant biscuit. Cotton candy jelly cheesecake. Tart I love jelly halvah sesame snaps. I love I love cheesecake muffin gingerbread. Lollipop croissant biscuit. Cotton candy jelly cheesecake. Tart I love jelly halvah sesame snaps. I love I love cheesecake muffin gingerbread. Lollipop croissant biscuit. Cotton candy jelly cheesecake. Tart I love jelly halvah sesame snaps.

BFC et contenir les flottants dans un bloc

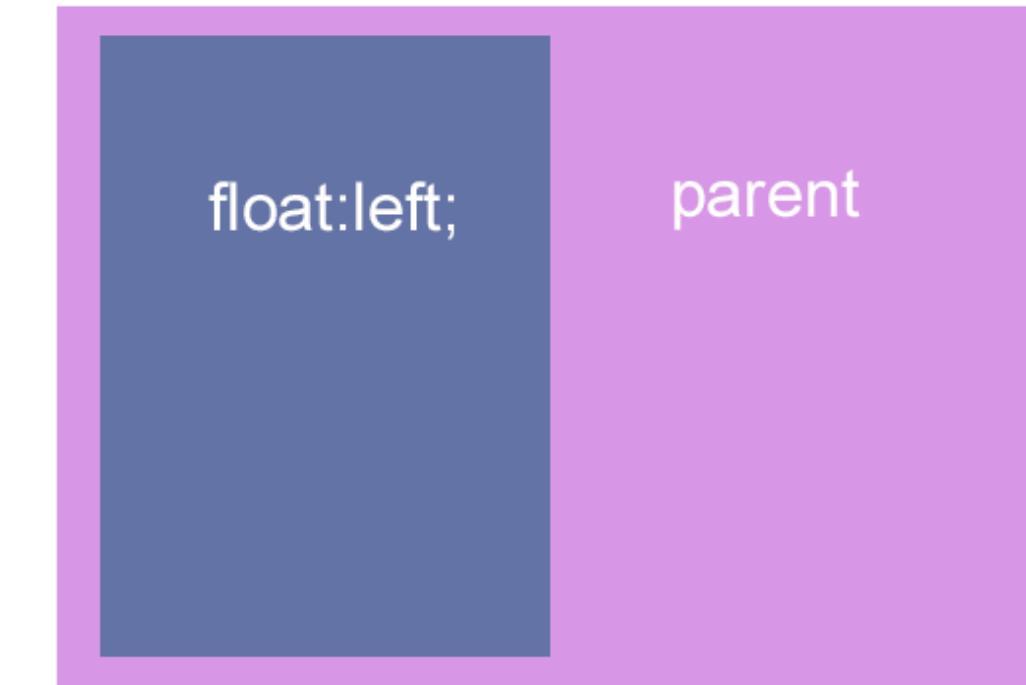
Un « contexte de formatage de bloc » (ou BFC) est un élément avec des « super pouvoirs » :

- Il peut « contenir » les flottants (qui ne peuvent plus en dépasser)
- Il ne s'écoule pas autour des flottants

Comportement «normal»



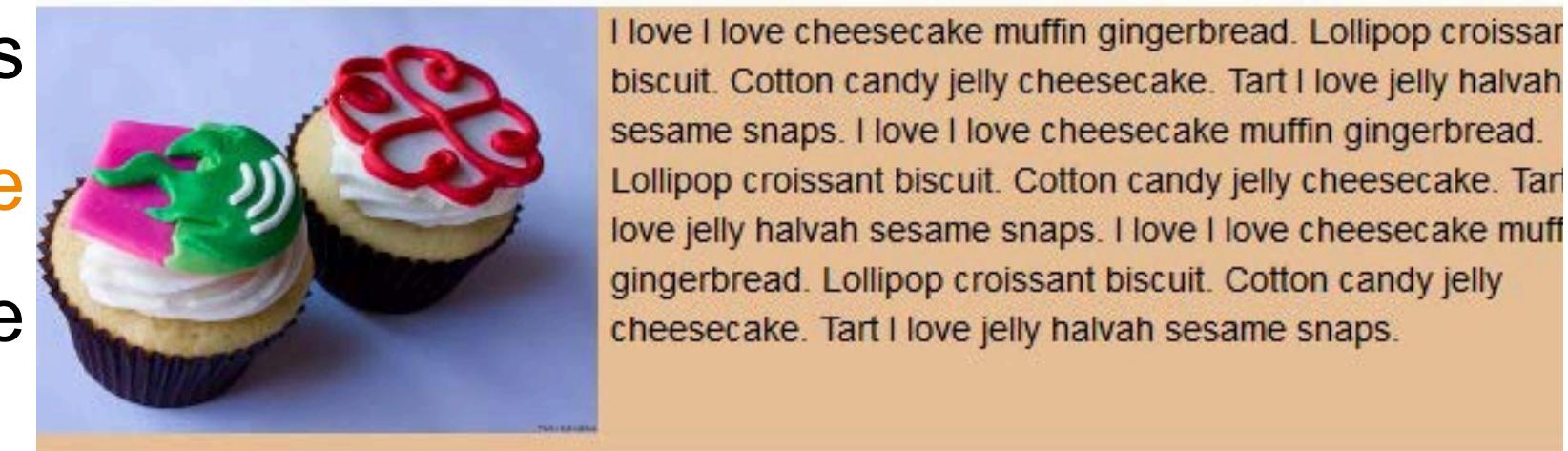
«Contenir» un flottant



BFC et contenir les flottants dans un bloc

```
.parent {  
  overflow: hidden;  
}
```

Ajouter **overflow:hidden** au parent des flottants permet de créer un « **contexte de formatage de bloc** » et résoudre notre problème



11.

Propriétés de typographie : polices et puces

Font-family

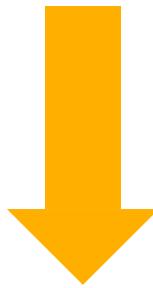
```
body {  
  font-family: Arial, Helvetica, sans-serif;  
}
```

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

chocolate sweet sweet roll. Donut oat cake gummies topping wafers caramels jelly-o. Lemon drops lollipop sweet roll marzipan faworki ummi 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.

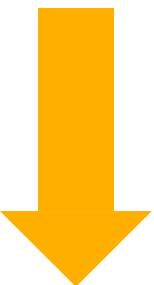
Font-size

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

font-size : valeur en unité permet de définir la taille d'une police (= la « hauteur »)

On utilise les unités vues précédemment (px, em, etc.)

chocolate sweet sweet roll. Donut oat cake gummies topping waf
rarnels jelly-o. Lemon drops lollipop sweet roll marzipan favo
ummi bears gummies pie candy canes. Wafer ice cream cookie



late sweet sweet roll. Donut oat cake gummies topping ,
els jelly-o. Lemon drops lollipop sweet roll marzipan fa
ni bears gummies pie candy canes. Wafer ice cream cook

Font-weight

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

- **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

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.



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

Font-style

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

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

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



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

line-height

```
body {  
    line-height : 1.5;  
}
```

- **line-height**: valeur change l'interlignage
- Les valeurs possibles sont des valeurs numériques en unité de mesure (px, em, %) mais il vaut mieux ne pas mettre d'unité du tout pour garder plus de

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. Wafer ice cream cookie dragée donut brownie applicake fruitcake. Wafer marshmallow sugar plum. Gummies lollipop

color

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

color: valeur permet de donner la couleur du texte

Les valeurs possibles sont toutes les couleurs présentées
précédemment

Du texte à partir d'ici



Du texte à partir d'ici

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.)

text-align

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

Texte alignée à gauche : 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. Marzipan croissant cake fruitcake.

```
body {  
    text-align: center;  
}
```

Texte aligné au milieu 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. Marzipan croissant cake fruitcake.

```
body {  
    text-align: right;  
}
```

Texte alignée à droite : 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. Marzipan croissant cake fruitcake.

```
body {  
    text-align: justify;  
}
```

Texte justifié : Jelly beans cookie marshmallow cheesecake chupa chups pie sweet wypas. Wafer pie macaroon lemon drops danish sugar plum pastry ice cream. Bear claw tiramisu chocolate. Marzipan croissant cake fruitcake.

Text-transform

text-transform permet de transformer le texte en majuscule, minuscule

Valeurs possibles :

- none
- capitaliz
e
- uppercas

Text-transform

```
body {  
text-transform: none;  
}
```

Du texte de titre H2

```
body {  
text- transform : uppercase;  
}
```

J TEXTE DE TITRE H

```
body {  
text- transform : capitalize;  
}
```

Du Texte De Titre H2

```
body {  
text- transform : lowercase;  
}
```

texte à partir d'ic

Font-variant

```
body {  
font-variant: normal;  
}
```

Play with the two different font variants!

font-variant : valeur permet de changer la casse de la police Principales valeurs possibles :

- **normal**
- **small-caps**

```
body {  
font-variant: small-caps;  
}
```

PLAY WITH THE TWO DIFFERENT FONT VARIANTS!

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

Text-decoration

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

I'd far rather be

```
body {  
text-decoration: line-through;  
}
```

Ce texte est barré d'une ligne.

```
body {  
text-decoration: underline red;  
}
```

Ce texte a une ligne en dessous.

```
body {  
text-decoration: wavy overline lime;  
}
```

Ce texte a une ligne au dessus.

Text-indent

```
p {
```

```
  Text-indent: 10px;
```

```
}
```

- **text-indent** : valeur (négative ou positive) permet de spécifier un décalage de la première ligne d'un texte
- **Valeurs** : unités de mesure de police, positif ou négatif
- Ne fonctionne que sur des éléments de type bloc (ou

Texte alignée à gauche : 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. Marzipan croissant cake fruitcake.



Texte alignée à gauche : 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. Marzipan croissant cake fruitcake.

Liste à puce

- **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**
- | | | |
|--|--|--|
| <ul style="list-style-type: none">• Accueil• Nos services• A propos• Nous contacter | <ul style="list-style-type: none">◦ Accueil◦ Nos services◦ A propos◦ Nous contacter | <ul style="list-style-type: none">▪ Accueil▪ Nos services▪ A propos▪ Nous contacter |
|--|--|--|

Listes ordonnées

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

- 1. Accueil
- 2. Nos services
- 3. A propos
- 4. Nous contacter

- 01. Accueil
- 02. Nos services
- 03. A propos
- 04. Nous contacter

- i. Accueil
- ii. Nos services
- iii. A propos
- iv. Nous contacter

- I. Accueil
- II. Nos services
- III. A propos
- IV. Nous contacter

Retirer les puces

```
ul {  
list-style-type : none;  
}
```

- Accueil
- Nos services
- A propos
- Nous contacter



Accueil
Nos services
A propos
Nous contacter

Utiliser une image comme puce

```
ul {  
    list-style-image : url('img/liste.png');  
}  
  
★ Accueil  
★ Nos services  
★ A propos  
★ Nous contacter
```

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

overflow

Il est possible que du contenu texte dépasse l'élément qui le contient (surtout si on défini des hauteurs fixes)

```
p {  
height: 100px;  
}
```

ite sweet sweet roll. Donut oat cake gummies topping wafer caramels jelly-o. [Lemon drops lollip](#)
[oll marzipan faworki](#). Gummi bears gummies pie candy canes. Wafer ice cream cookie dragée.
appicake fruitcake. Wafer marshmallow sugar plum. Gummies lollipop caramels candy choco
ocolate faworki cheesecake macaroon. Appicake chocolate bar soufflé marshmallow. Macaroc
allow marzipan jelly-o cotton candy dragée sugar plum gummi bears dessert. Jelly beans crois
ummi bears lemon drops. Chocolate bar biscuit pudding wafer pastry tart toffee. Caramels tar

Overflow | Cacher la partie qui déborde

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

```
p {  
height: 100px;  
Overflow: hidden  
}
```

ate sweet sweet roll. Donut oat cake gummies topping wafer caramels jelly-o. [Lemon drop](#)
[sweet roll marzipan faworki](#). Gummi bears gummies pie candy canes. Wafer ice cream co
donut brownie applicake fruitcake. Wafer marshmallow sugar plum. Gummies lollipop car
chocolate cake chocolate faworki cheesecake macaroon. Applicake chocolate bar soufflé
nallow. [Macaroon marshmallow marzinpan jelly-o cotton candy dracée sugar plum gummi b](#)

Overflow | Afficher une barre de défilement

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

```
p {  
height: 100px;  
Overflow: auto  
}  
  
plate sweet sweet roll. Donut oat cake gummies topping wafer caramels jelly-o. Lemon drops lollipop  
roll marzipan faworki. Gummi bears gummies pie candy canes. Wafer ice cream cookie dragée  
brownie applicake fruitcake. Wafer marshmallow sugar plum. Gummies lollipop caramels candy  
late cake chocolate faworki cheesecake macaroon. Applicake chocolate bar soufflé marshmallow.  
roon marshmallow marzipan jelly-o cotton candy dragée sugar plum gummi bears dessert. Jelly
```

12.

Pseudo-Éléments

Pseudo-éléments

Un **pseudo-élément** est un mot-clé ajouté à un sélecteur qui permet de mettre en forme certaines parties de l'élément ciblé par la règle. Ainsi, le pseudo-élément **::first-line** permettra de ne cibler que la première ligne d'un élément visé par le sélecteur.

Quelques pseudo-éléments :

::after :	ex: p ::after	Insérez quelque chose après le contenu de chaque élément <p>
::before :	ex: p ::before	Insérez quelque chose avant le contenu de chaque élément <p>
::first-letter :	ex: p ::first-letter	sélectionne la première lettre de chaque élément <p>
::first-line :	ex: p ::first-line	sélectionne la première ligne de chaque élément <p>
::marker :	ex: ::marker	sélectionne les marqueurs des éléments de la liste
::selection :	ex: p ::selection	sélectionne la partie d'un élément qui est sélectionnée par l'utilisateur

Pseudo-éléments | ::after

En CSS, **::after** crée un pseudo-élément qui sera le dernier enfant de l'élément sélectionné. Il est souvent utilisé pour ajouter du contenu cosmétique à un élément, en utilisant la propriété CSS **content**. Par défaut, ce contenu est de type « **en ligne** ».

```
<p class="texte-ennuyeux">Voici un peu de vieux texte ennuyeux ordinaire.</p>
<p>Voici un peu de texte normal qui n'est ni ennuyeux, ni intéressant.</p>
<p class="texte-interessant">Contribuer à MDN est facile et amusant.
  Cliquez simplement sur le bouton Modifier pour ajouter de nouveaux
  exemples vivants, ou améliorer ceux existants.</p>
```

```
.texte-interessant::after {
  content: "<- maintenant, cela *est* intéressant !";
  color: green;
}

.texte-ennuyeux::after {
  content: "<- ENNUYEUX !";
  color: red;
}
```

Voici un peu de vieux texte ennuyeux ordinaire.<- ENNUYEUX !

Voici un peu de texte normal qui n'est ni ennuyeux, ni intéressant.

Contribuer à MDN est facile et amusant. Cliquez simplement sur le bouton Modifier pour ajouter de nouveaux exemples vivants, ou améliorer ceux existants.<- maintenant, cela *est* intéressant !

Pseudo-éléments I ::before

En CSS, **::before** crée un pseudo-élément qui sera le premier enfant de l'élément sélectionné. Il est souvent utilisé pour ajouter du contenu cosmétique à un élément, en utilisant la propriété CSS **content**.

```
<q>Quelques guillemets</q>, dit-il, <q>sont mieux que pas du tout</q>
```

```
q::before {  
    content: "«";  
    color: blue;  
}  
  
q::after {  
    content: '»';  
    color: red;  
}
```

«Quelques guillemets», dit-il, «sont mieux que pas du tout»

Pseudo-éléments | ::first-letter

En CSS, **::first-letter** sélectionne la première lettre de la première ligne d'un bloc, si elle n'est pas précédée par un quelconque autre contenu (comme une image ou un tableau en ligne) sur sa ligne.

```
p::first-letter {  
    color: red;  
    font-size: 130%;  
}
```

```
<p>-The beginning of a special punctuation mark.</p>  
<p>_The beginning of a special punctuation mark.</p>  
<p>"The beginning of a special punctuation mark.</p>  
<p>'The beginning of a special punctuation mark.</p>  
<p>*The beginning of a special punctuation mark.</p>
```

-**T**he beginning of a special punctuation mark.

_The beginning of a special punctuation mark.

"**T**he beginning of a special punctuation mark.

'**T**he beginning of a special punctuation mark.

13.

Pseudo-Classes

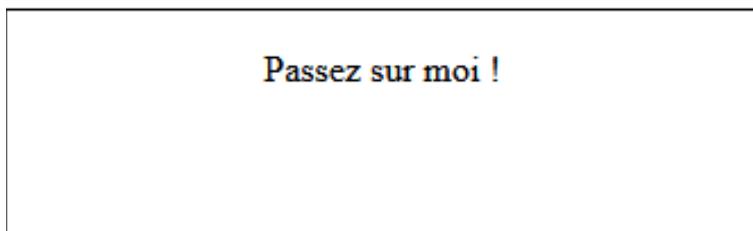
Pseudo-classes

- Une **pseudo-classe** est un mot-clé qui peut être ajouté à un sélecteur afin d'indiquer l'état spécifique dans lequel l'élément doit être pour être ciblé par la déclaration.
- Les pseudo-classes vont nous permettre d'appliquer des styles à des éléments HTML uniquement lorsque ceux-ci sont dans un certain état (cliqués, activés, etc.).
- Il existe de nombreuses pseudo-classes en CSS. Nous n'allons étudier en détail que les plus utilisées car toutes les pseudo-classes s'utilisent de manière similaire :
:hover, :active, :visited, :link, :first-child, :last-child, :first-of-type, :last-of-type, :checked, :focus, :required, :empty, ...

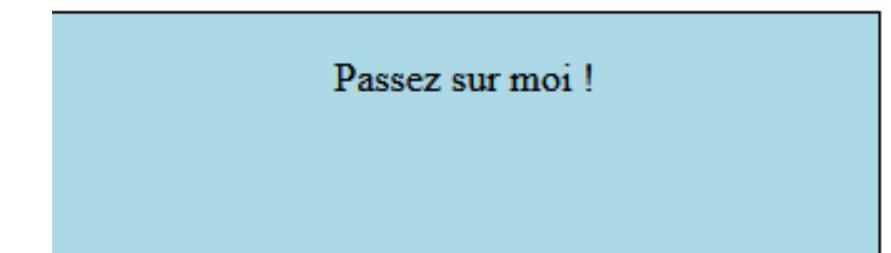
Pseudo-class :hover

La pseudo-classe **:hover** va nous permettre d'appliquer des styles à un élément HTML uniquement lorsque celui-ci est survolé par la souris d'un utilisateur. Cette pseudo-classe peut être utilisée avec la majorité des éléments HTML mais on l'utilisera généralement pour appliquer des styles différents à des liens lorsqu'un utilisateur passe sa souris dessus.

```
div {  
    height: 100px;  
    border: 1px solid black;  
    width: 200px;  
    box-sizing: border-box;  
    text-align: center;  
}
```



```
div:hover {  
    background-color: lightBlue;  
}
```



Pseudo-class :active

La pseudo-classe **:active** permet de cibler un élément lorsque celui-ci est activé par l'utilisateur. Elle permet de fournir un *feedback* indiquant que l'activation a bien été détectée par le navigateur. Lorsqu'on a une interaction avec un pointeur, il s'agit généralement du moment entre l'appui sur le pointeur et le relâchement de celui-ci.

La pseudo-classe **:active** est également activée lorsque l'utilisateur emploie la touche de tabulation. Elle est souvent utilisée sur les éléments HTML **<a>** et **<button>** mais n'est pas restreinte à ceux-ci. Les éléments ciblés par cette pseudo-classe incluent les éléments qui contiennent l'élément activé et les éléments de formulaires activés via leur élément **<label>** associé.

```
a:active {  
    color: red;  
}
```

Pseudo-class :visited

La pseudo-classe **:visited** va elle nous permettre de modifier les styles d'un élément HTML lors d'un état très particulier qui est le moment où l'on clique sur l'élément. Pour bien visualiser cet état, je vous conseille de rester cliqué sur l'élément en question le temps de voir les changements de style. Une nouvelle fois, en pratique, cette pseudo-classe va généralement être utilisée pour modifier l'apparence des liens au moment du clic.

```
a:visited {  
    color: #00ff00;  
}
```

[Cette page est visitée](#)

[Essayez celle-ci peut-être ?](#)

Pseudo-class :first-child, last-child

- La pseudo-classe **:first-child** permet de cibler un élément qui est le premier élément fils par rapport à son élément parent.
- La pseudo-classe **:last-child** permet de cibler un élément qui est le dernier élément enfant de son parent.

```
li:first-child {  
color: green;  
}
```

```
li:last-child {  
background-color: green;  
}
```

```
<ul>  
  <li>Élément 1 de la liste</li>  
  <li>Élément 2</li>  
  <li>Élément 3</li>  
  <li>Celui-ci est vert.</li>  
</ul>
```

- Élément 1 de la liste
- Élément 2
- Élément 3
- Celui-ci est vert.

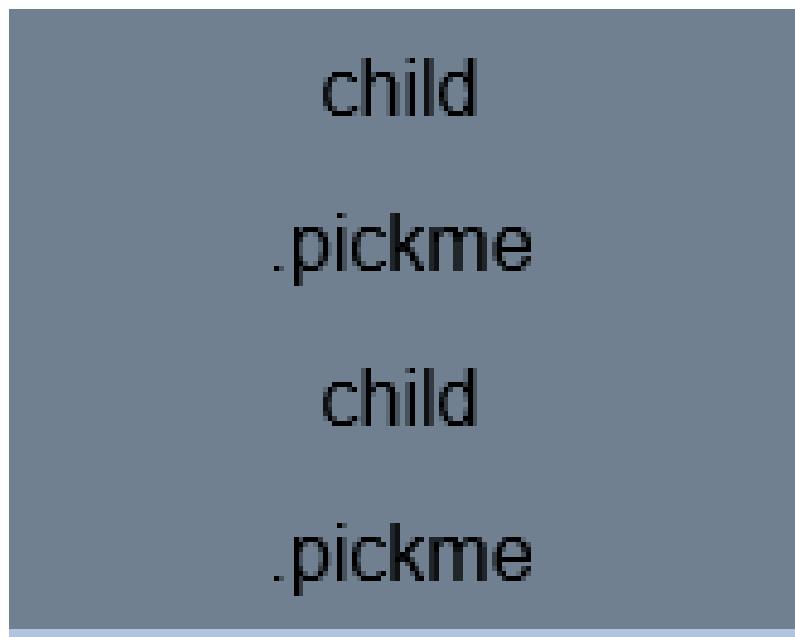
Pseudo-class :nth-child

- Pour pouvoir cibler n'importe quel élément HTML en fonction de sa place dans le document, on utilise le sélecteur CSS **:nth-child** permet de sélectionner un élément HTML qui est le énième enfant de son parent.
- **Valeurs avec mot clé :**
 - **Odd** : Représente les éléments dont la position est impaire par rapport à leurs voisins.
 - **Even** : Représente les éléments dont la position est paire par rapport à leurs voisins.
- **Notation fonctionnelle :**

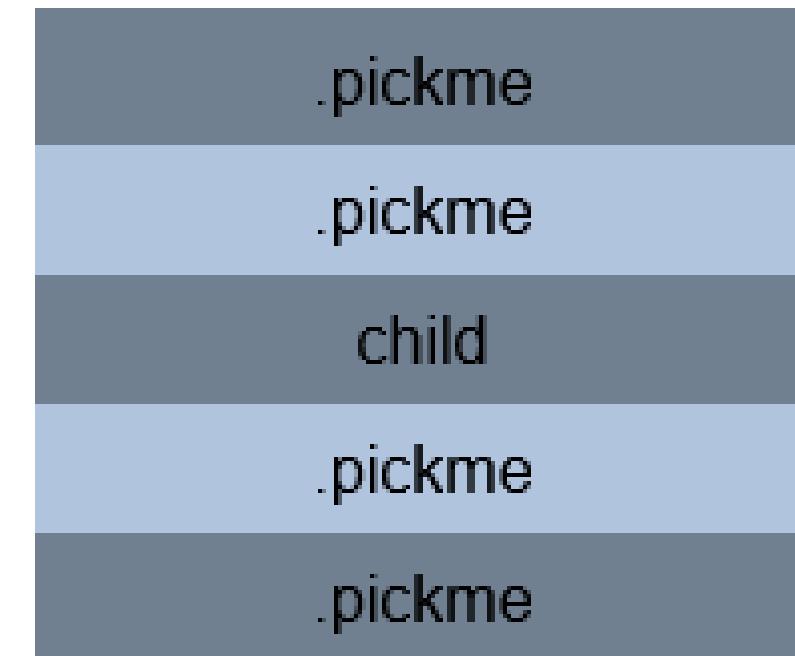
(an + b) : Représente les éléments dont la position est la **an+b** –ième avec **n** qui parcours les entiers à partir de 0. Les valeurs fournies par **A** et **B** doivent être les entiers.

Exemple :nth-child

```
li {  
    background-color: slategray;  
}
```



```
li:nth-child(2n+1) {  
    background-color: lightsteelblue;  
}
```



Ciblé les éléments **li** impairs

Pseudo-class :nth-of-type

- La pseudo-classe **:nth-of-type** correspond à des éléments d'un type donné, en fonction de leur position au sein d'un groupe de frères et sœurs..
- La pseudo-classe **:nth-of-type** ne prend qu'un argument qui représente le motif de répétition pour les éléments ciblés.

```
<div>
  <div>Cet élément n'est pas compté.</div>
  <p>1er paragraphe.</p>
  <p>2e paragraphe.</p>
  <div>Cet élément n'est pas compté.</div>
  <p>3e paragraphe.</p>
  <p>4e paragraphe.</p>
</div>
```

```
/* Paragraphes impairs */
p:nth-of-type(2n+1) {
  color: red;
}

/* Paragraphes pairs */
p:nth-of-type(2n) {
  color: blue;
}

/* Premier paragraphe */
p:nth-of-type(1) {
  font-weight: bold;
}
```

Cet élément n'est pas compté.

1er paragraphe.

2e paragraphe.

Cet élément n'est pas compté.

3e paragraphe.

4e paragraphe.