

PARTIE 2 : CSS

[HTTP://JIGSAW.W3.ORG/CSS-VALIDATOR/](http://jigsaw.w3.org/css-validator/)

CHAPITRE 1 : CSS

PRÉAMBULE

**HISTORIQUE, MOTIVATION,
ENVIRONNEMENT**

HISTORIQUE

- Introduit très tôt 1990 mais instable
- CSS2.1 2007 est accepté
- CSS3 de 1990 à aujourd'hui → Gros changements
- CSS4 est en cours d'écriture depuis 2010 → Extension du 3

ENVIRONNEMENT

- Editeur de texte
- HTML
- Navigateur web
- Inspecteur

VOCABULAIRES

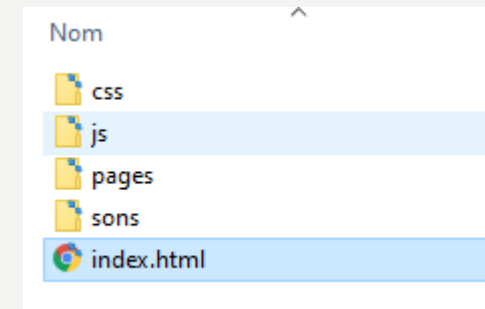
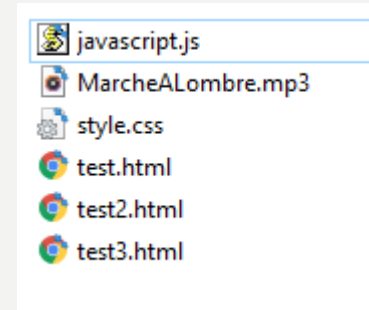
- **Sélecteurs**
 - Cible l/les élément(s) qui appliqueront le style définit
 - Contient un ensemble de propriétés (l à n)
- **Propriétés**
 - Cible la caractéristique à modifier
 - Contient un ensemble de valeur (l à n)
- **Valeurs**
 - Indique la valeur que prend une propriété

OU PLACE-T-ON LE CSS ?

- Externe
 - `<link />`
- Head
 - `<style></style>`
- Balises
 - Attribut *style*
- Chaque solution est valable selon le besoin
 - On ne met pas de style spécifique à une page dans un fichier qui sera importé dans toutes les pages

ORGANISONS NOTRE DOSSIER

- Séparer les différentes ressources
- Index → Page racine, chercher automatiquement par le serveur web
- Pages
- CSS
- JS
- Vidéos
- Images
- Sons



CHAPITRE 1 : PREMIER PAS EN CSS

BLOCK, INLINE, COMMENTAIRE, DIV ET SPAN

UN PREMIER STYLE

- Mettre la couleur des paragraphe en rouge

```
1  p{  
2      color:red;  
3  }
```

- Sélecteur, propriété, valeur

```
<head>  
  <meta charset="utf-8" />  
  <title>Test</title>  
  <link rel="stylesheet" href="css/style.css" />  
  <style>  
    p{  
      color:red;  
    }  
  </style>  
</head>
```

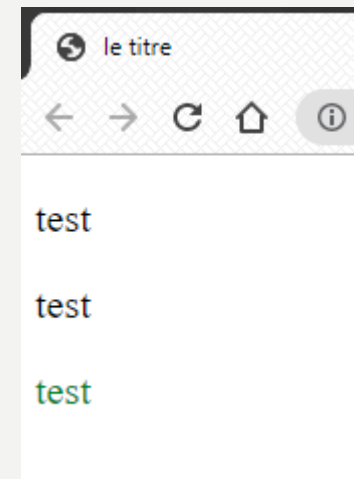
```
<p style="color:red">paragraphe</p>
```

CLASS ET ID

- Les sélecteurs les plus répandus
- Class
 - Un point
 - Peut sélectionner plusieurs éléments
- ID
 - Un dièse
 - Ne doit sélectionner qu'un seul élément

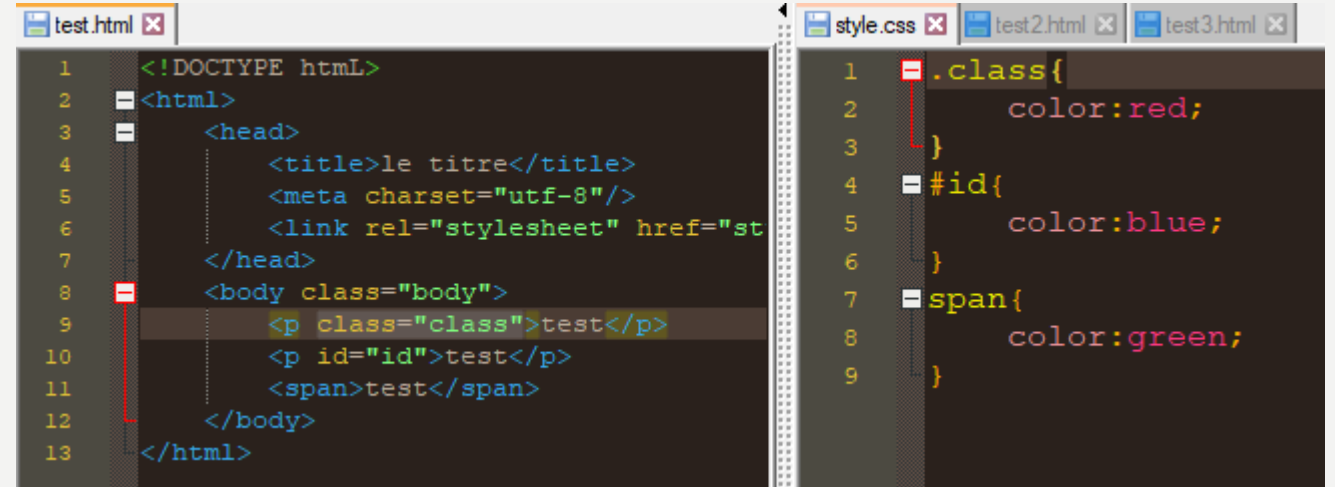
```
test.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>le titre</title>
5     <meta charset="utf-8"/>
6     <link rel="stylesheet" href="st
7   </head>
8   <body class="body">
9     <p class="class">test</p>
10    <p id="id">test</p>
11    <span>test</span>
12  </body>
13 </html>

style.css
1 .class{
2   color:red;
3 }
4 #id{
5   color:blue;
6 }
7 span{
8   color:green;
9 }
```



CLASS ET ID

- Les sélecteurs les plus répandus
- Class
 - Un point
 - Peut sélectionner plusieurs éléments
- ID
 - Un dièse
 - Ne doit sélectionner qu'un seul élément



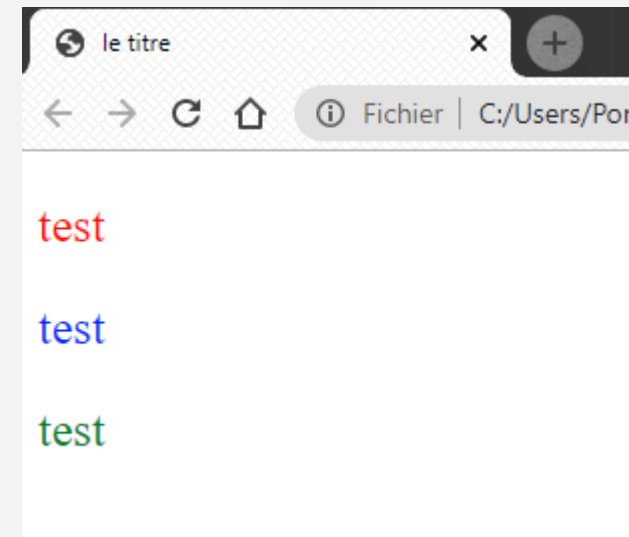
The image shows two code editors side-by-side. The left editor, titled 'test.html', contains the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>le titre</title>
5   <meta charset="utf-8"/>
6   <link rel="stylesheet" href="st
7 </head>
8 <body class="body">
9   <p class="class">test</p>
10  <p id="id">test</p>
11  <span>test</span>
12 </body>
13 </html>
```

The right editor, titled 'style.css', contains the following CSS code:

```
1 .class{
2   color:red;
3 }
4 #id{
5   color:blue;
6 }
7 span{
8   color:green;
9 }
```

A large yellow arrow points from the code editors down to the browser window below.



SELECTEURS

- Une multitude de sélecteurs à connaître et on va tous les voir !!!
- La liste complète dans la documentation : https://www.w3schools.com/cssref/css_selectors.asp
- La combinaison de class
 - .class1.class2
 - .class1 .class2

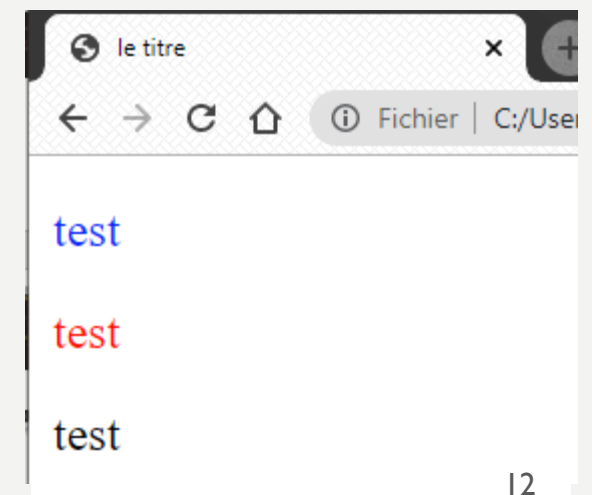


The screenshot shows a code editor with three tabs: test.html, style.css, and test2.html. The test.html file contains the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>le titre</title>
5     <meta charset="utf-8"/>
6     <link rel="stylesheet" href="style
7   </head>
8   <body class="body">
9     <p class="class1 class2">
10      test
11      <p class="class2">test</p>
12    </p>
13    <span>test</span>
14  </body>
15 </html>
```

The style.css file contains the following CSS code:

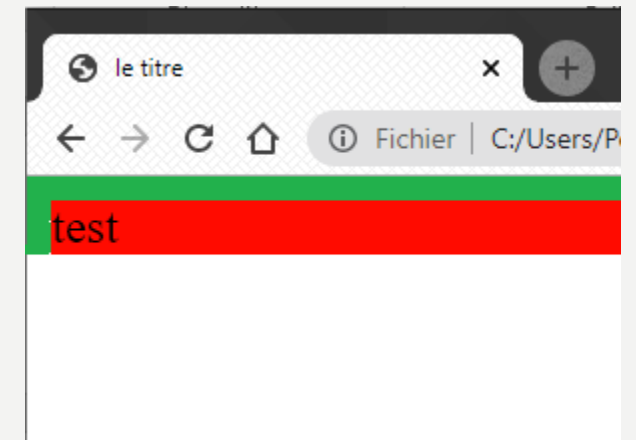
```
1 .class1.class2{
2   color:red;
3 }
4 .body .class2{
5   color:red;
6 }
7 .body .class1{
8   color:blue;
9 }
10
```



SELECTEURS : *

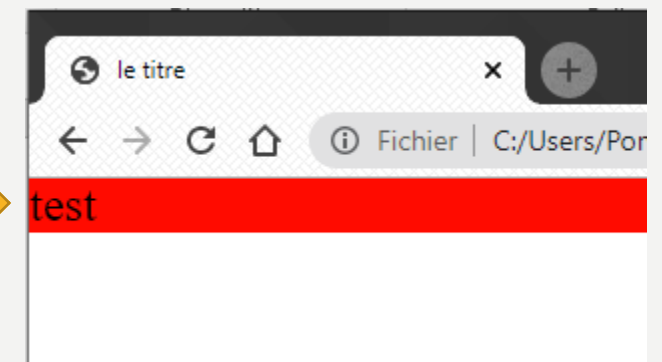
```
test.html x
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>le titre</title>
5     <meta charset="utf-8"/>
6     <link rel="stylesheet" href="style.css">
7   </head>
8   <body class="body">
9     <div class="test">
10      test
11    </div>
12  </body>
13 </html>
```

```
style.css x test2.html x test3.html x
1 .test{
2   background-color:red;
3 }
```



```
test.html x
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>le titre</title>
5     <meta charset="utf-8"/>
6     <link rel="stylesheet" href="style.css">
7   </head>
8   <body class="body">
9     <div class="test">
10      test
11    </div>
12  </body>
13 </html>
```

```
style.css x test2.html x test3.html x
1 .test{
2   background-color:red;
3 }
4 *{
5   margin:0px;
6   padding:0px;
7 }
8
```

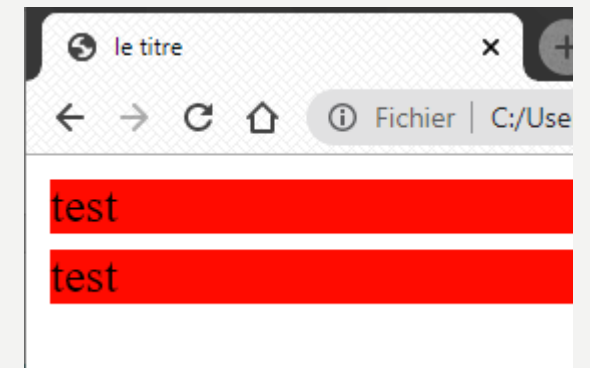


SELECTEURS : ,

- Virgule signifie "ET"

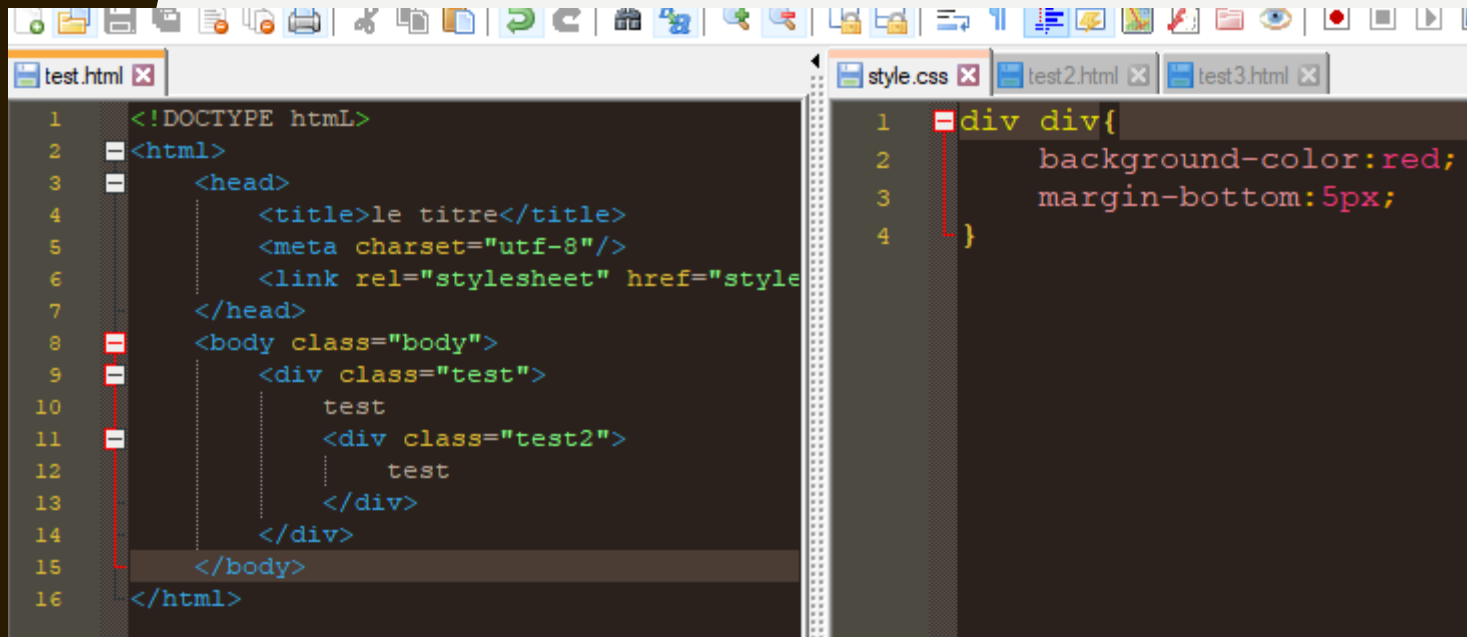
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>le titre</title>
5     <meta charset="utf-8"/>
6     <link rel="stylesheet" href="style
7   </head>
8   <body class="body">
9     <div class="test">
10       test
11     </div>
12     <div class="test2">
13       test
14     </div>
15   </body>
```

```
1 .test, .test2{
2   background-color:red;
3   margin-bottom:5px;
4 }
```



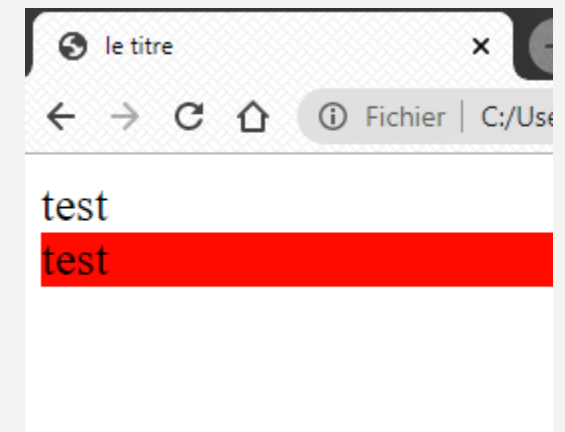
SELECTEURS : ESPACE

- Espace signifie INCLUS DANS



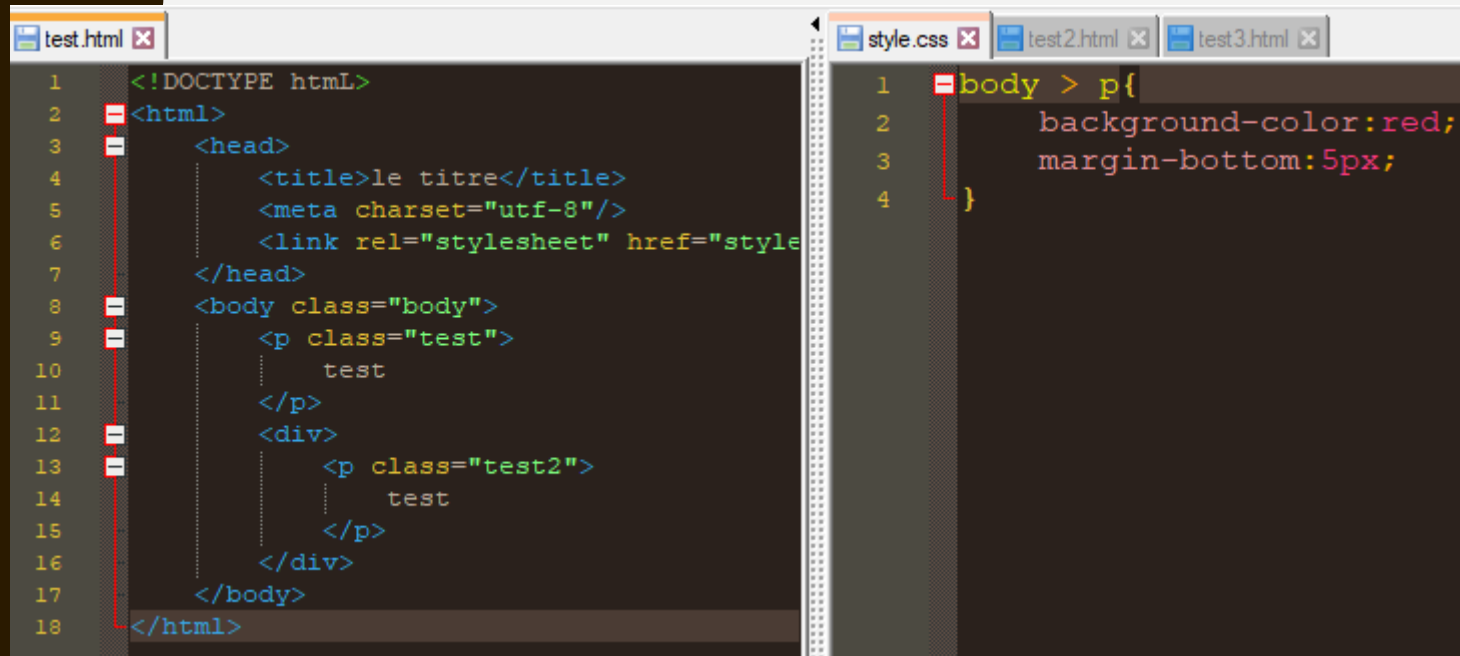
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>le titre</title>
5     <meta charset="utf-8"/>
6     <link rel="stylesheet" href="style.css">
7   </head>
8   <body class="body">
9     <div class="test">
10       test
11       <div class="test2">
12         test
13       </div>
14     </div>
15   </body>
16 </html>
```

```
1 div div{
2   background-color:red;
3   margin-bottom:5px;
4 }
```



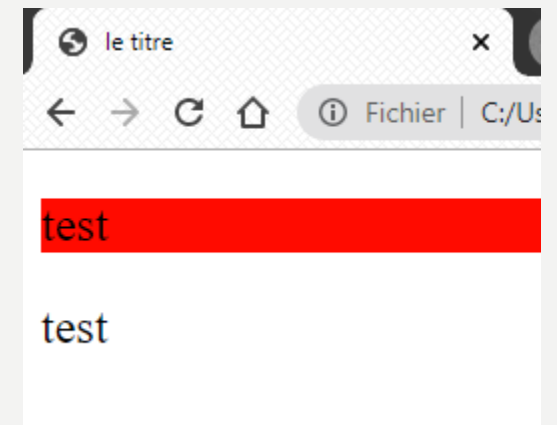
SELECTEURS : >

- > désigne un lien de filiation A>B : B est un fils de A



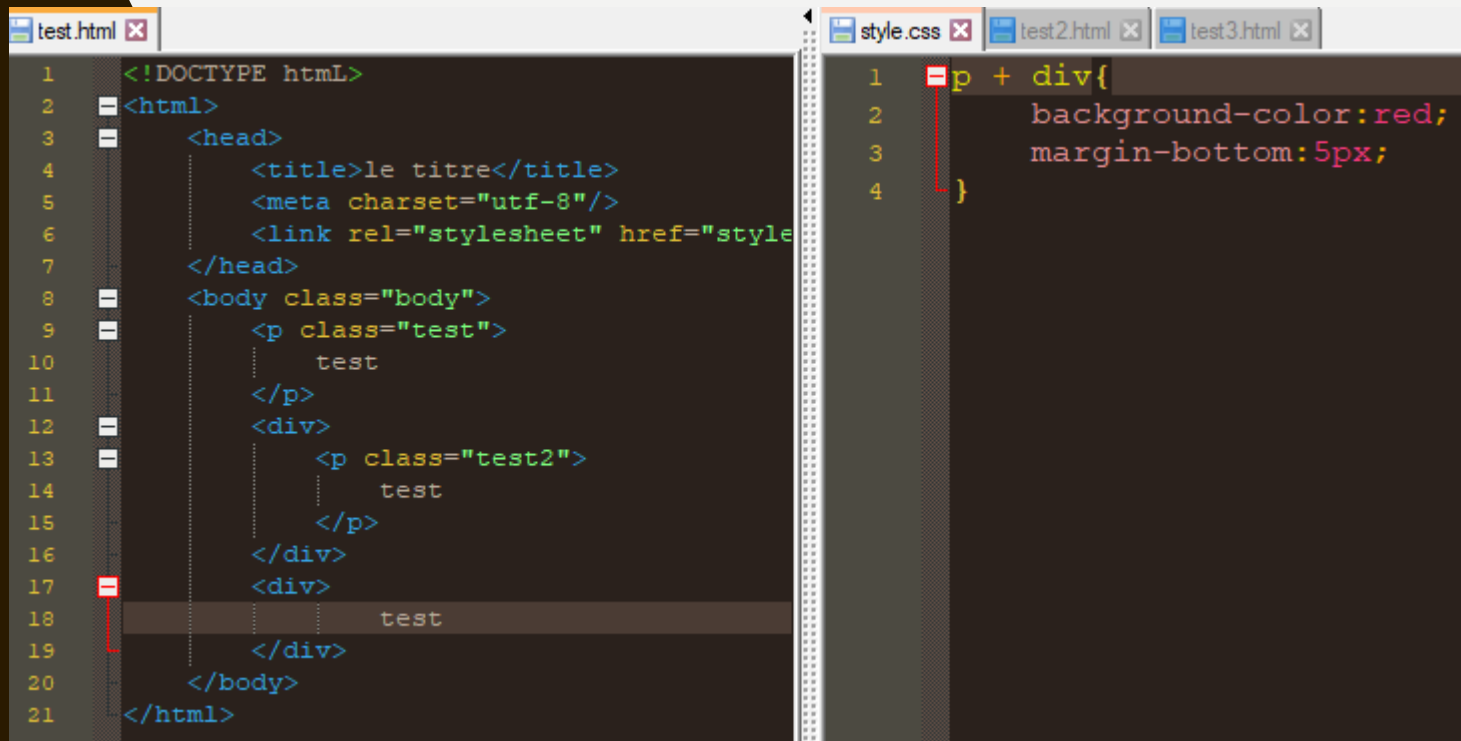
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>le titre</title>
5   <meta charset="utf-8"/>
6   <link rel="stylesheet" href="style.css">
7 </head>
8 <body class="body">
9   <p class="test">
10     test
11   </p>
12   <div>
13     <p class="test2">
14       test
15     </p>
16   </div>
17 </body>
18 </html>
```

```
1 body > p{
2   background-color:red;
3   margin-bottom:5px;
4 }
```



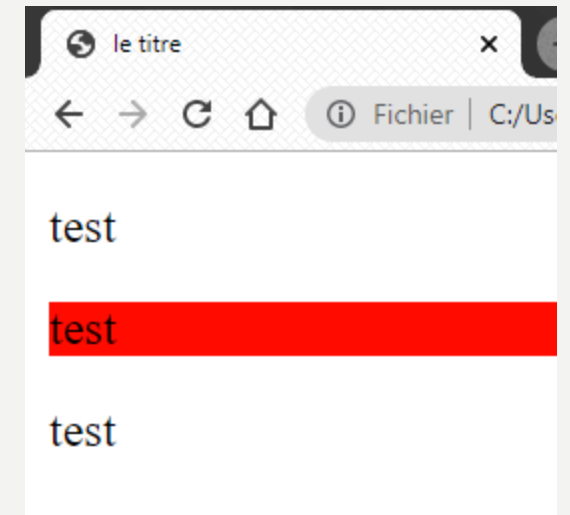
SELECTEURS : +

- Premier cadet



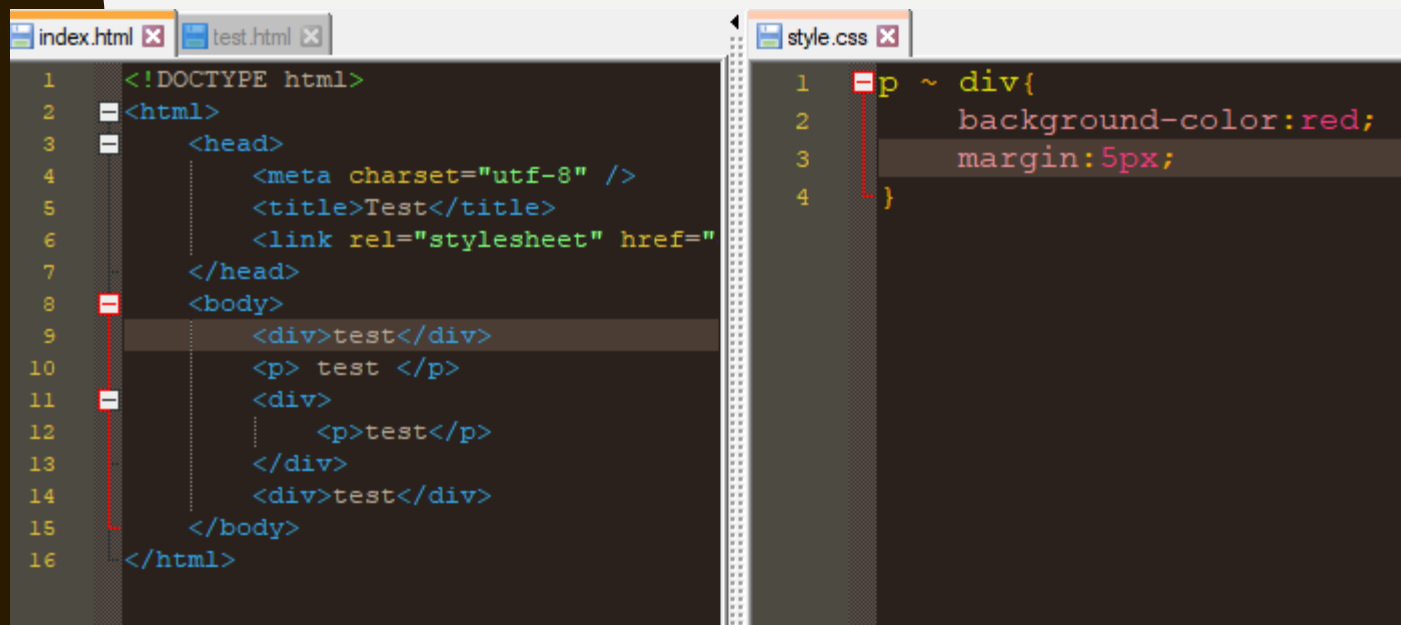
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>le titre</title>
5     <meta charset="utf-8"/>
6     <link rel="stylesheet" href="style
7   </head>
8   <body class="body">
9     <p class="test">
10      test
11    </p>
12    <div>
13      <p class="test2">
14        test
15      </p>
16    </div>
17    <div>
18      test
19    </div>
20  </body>
21 </html>
```

```
1 p + div{
2   background-color:red;
3   margin-bottom:5px;
4 }
```



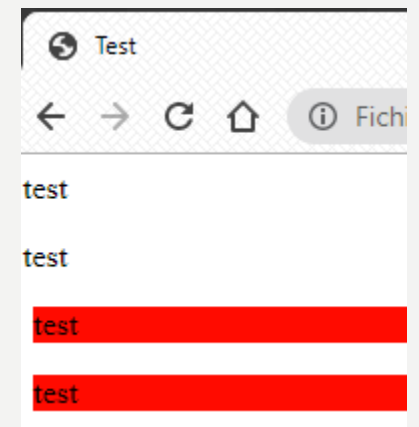
SELECTEURS : ~

- Ensemble qui se trouve quelque part dans la “cadèterie”



```
index.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Test</title>
6     <link rel="stylesheet" href="
7   </head>
8   <body>
9     <div>test</div>
10    <p> test </p>
11    <div>
12      <p>test</p>
13    </div>
14    <div>test</div>
15  </body>
16 </html>

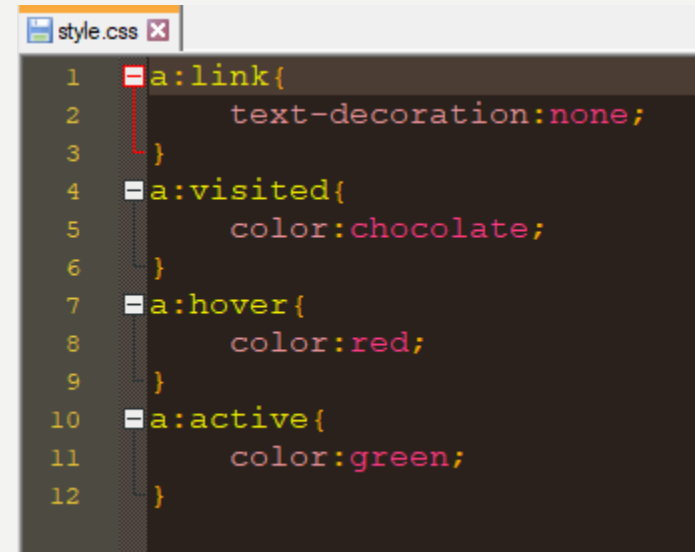
style.css
1 p ~ div{
2   background-color:red;
3   margin:5px;
4 }
```



SELECTEURS : LINK

- `:active` : l'élément en cours d'activation
- `:visited` : les liens visités
- `:link` : les liens pas encore visités
- `:hover` : le survol du lien avec la souris

- L'ordre est important et `:active` fonctionne aussi avec la tabulation et d'autre éléments que `a`

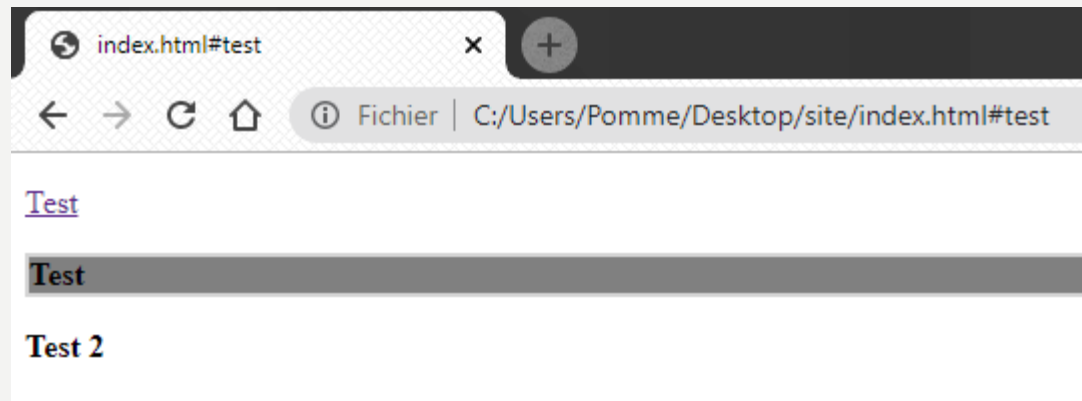


```
1 a:link{
2     text-decoration:none;
3 }
4 a:visited{
5     color:chocolate;
6 }
7 a:hover{
8     color:red;
9 }
10 a:active{
11     color:green;
12 }
```

SELECTEURS : :TARGET

- Id ciblé
- Se réfère à l'url

```
<head>
  <style>
    :target {
      border: 2px solid #d4d4d4;
      background-color: grey;
    }
  </style>
</head>
<body>
  <p><a href="#test">Test</a></p>
  <p id="test"><b>Test</b></p>
  <p><b>Test 2</b></p>
</body>
```



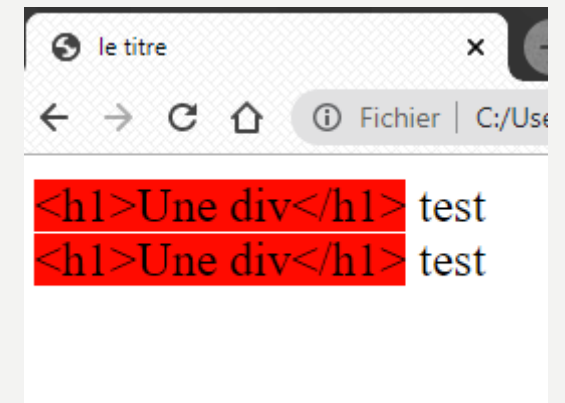
SELECTEURS : ::AFTER ET ::BEFORE

- Insère du contenu après ou avant un élément MAIS pas du html



```
test.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>le titre</title>
5     <meta charset="utf-8"/>
6     <link rel="stylesheet" href="style.css">
7   </head>
8   <body class="body">
9     <div>
10      test
11    </div>
12    <div>
13      test
14    </div>
15  </body>
16 </html>

style.css
1 div::before{
2   content:"<h1>Une div</h1>";
3   background-color: red;
4 }
```

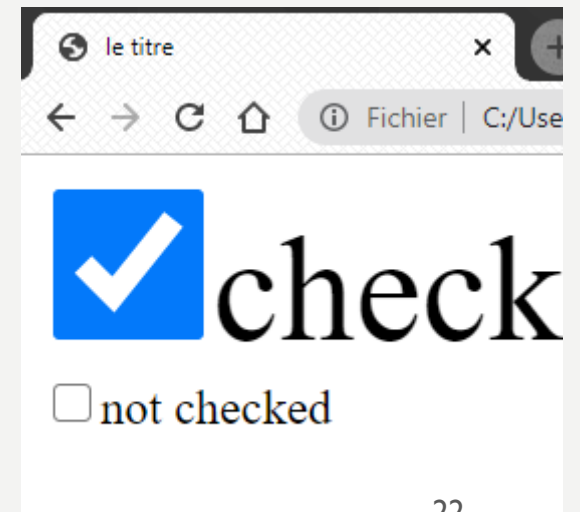


SELECTEURS : :CHECKED

- Valable pour radio, check et option (quand il y a une sélection possible...)

```
test.html x style.css x test2.html x test3.html x
1 PE htmL>
2
3 <ad>
4 <title>le titre</title>
5 <meta charset="utf-8"/>
6 <link rel="stylesheet" href="style.css">
7 </ad>
8 <body class="body">
9 <input type="checkbox" checked /><span>checked</span>
10 <input type="checkbox" /><span>not checked</span>
11 </body>
12
```

```
1 input:checked, input:checked + span {
2   height: 50px;
3   width: 50px;
4   font-size: 50px;
5 }
```

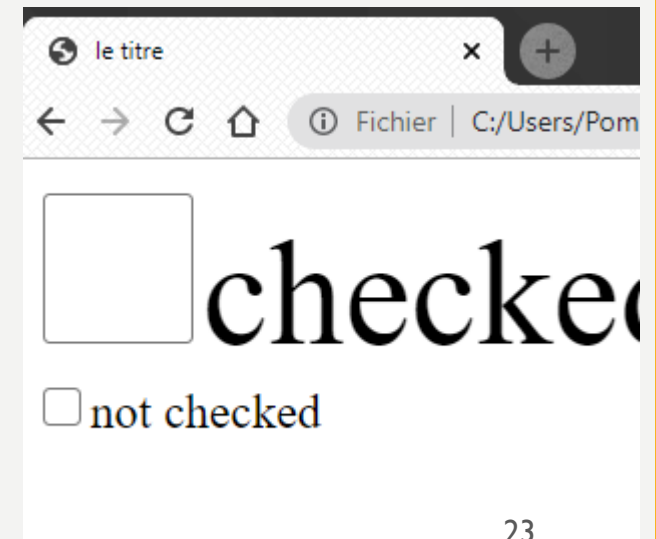


SELECTEURS : :DEFAULT

- Les éléments sélectionnés par défaut à l'ouverture de la page.
 - Bouton, radio, checkbox et option

```
test.html x
1 PE htmL>
2
3 <ad>
4 <title>le titre</title>
5 <meta charset="utf-8"/>
6 <link rel="stylesheet" href="style.css">
7 ead>
8 <body class="body">
9 <input type="checkbox" checked /><span>checked</span>
10 <input type="checkbox" /><span>not checked</span>
11 ody>
12
```

```
style.css x test2.html x test3.html x
1 input:default, input:default + span {
2   height: 50px;
3   width: 50px;
4   font-size:50px;
5 }
```

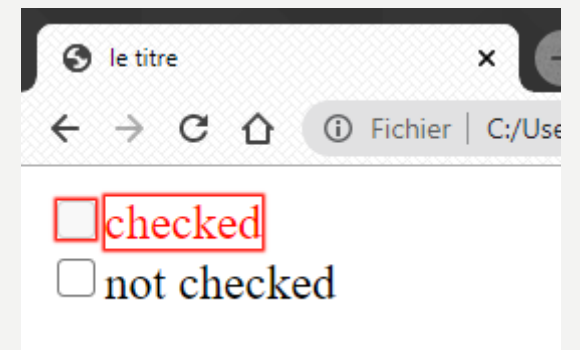


SELECTEURS : :DISABLED & ENABLED

- Input
- Input enabled
- Input disabled

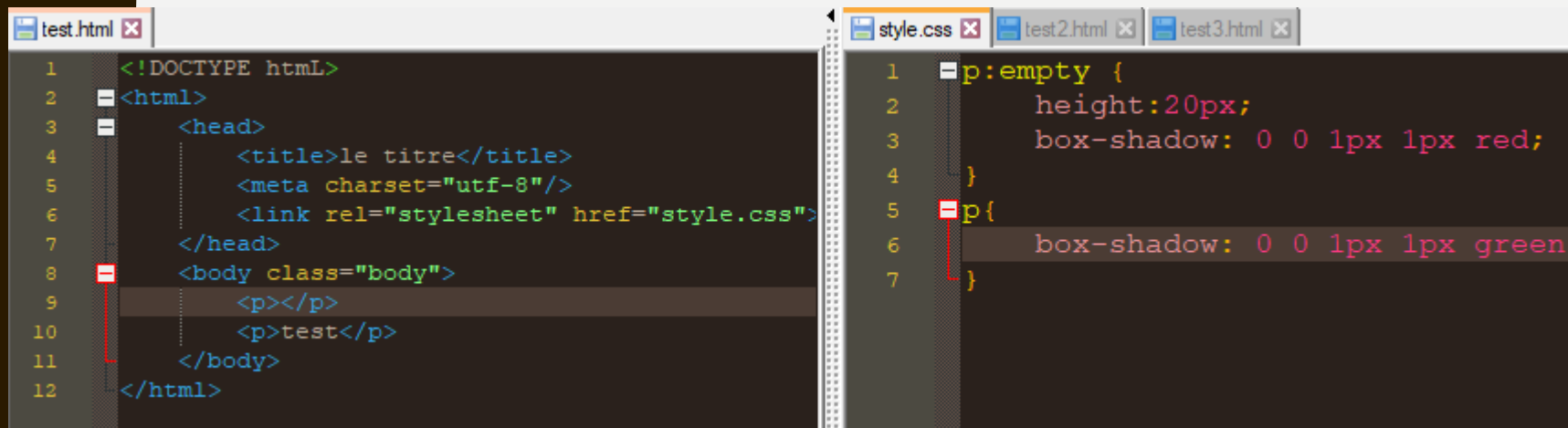
```
test.html x
1 !DOCTYPE html>
2 <html>
3   <head>
4     <title>le titre</title>
5     <meta charset="utf-8"/>
6     <link rel="stylesheet" href="style.css">
7   </head>
8   <body class="body">
9     <input type="checkbox" disabled /><span>checked
10    <input type="checkbox" /><span>not checked</spa
11  </body>
12 </html>

style.css x test2.html x test3.html x
1 input:disabled, input:disabled + span {
2   box-shadow: 0 0 1px 1px red;
3   color:red;
4 }
```



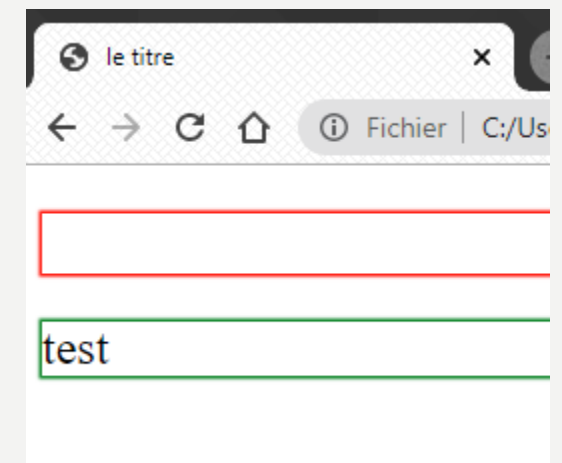
SELECTEURS : :EMPTY

- Cible les éléments qui n'ont pas de texte



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>le titre</title>
5     <meta charset="utf-8"/>
6     <link rel="stylesheet" href="style.css">
7   </head>
8   <body class="body">
9     <p></p>
10    <p>test</p>
11  </body>
12 </html>
```

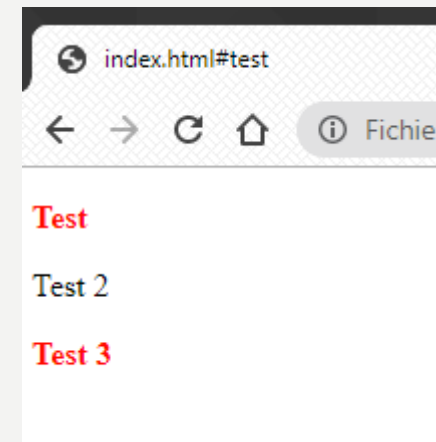
```
1 p:empty {
2   height:20px;
3   box-shadow: 0 0 1px 1px red;
4 }
5 p{
6   box-shadow: 0 0 1px 1px green;
7 }
```



SELECTEURS : :FIRST-CHILD

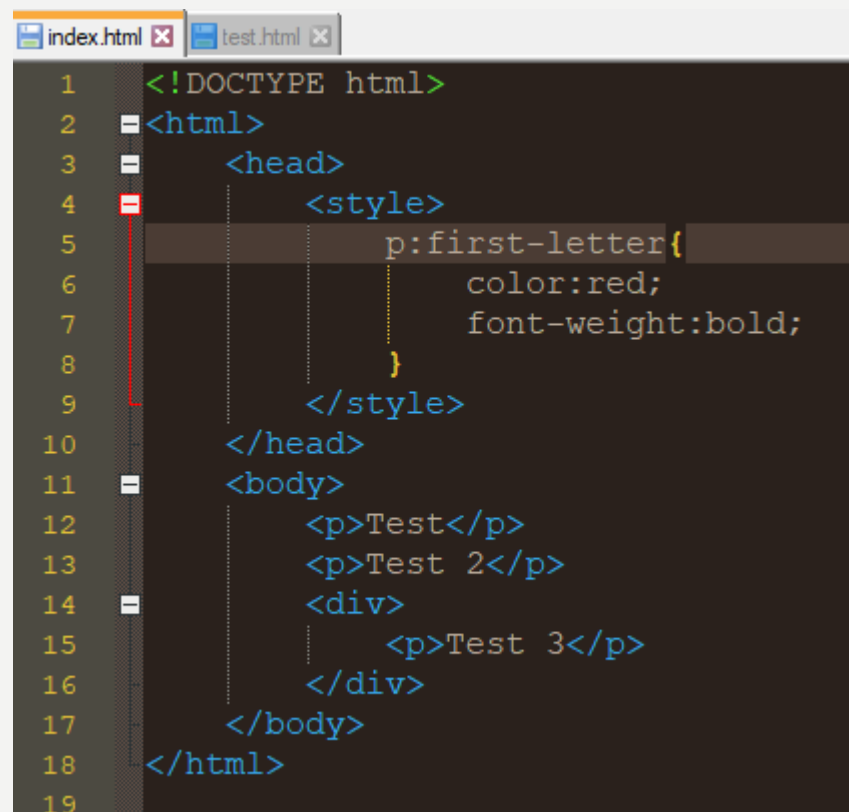
- Les éléments qui sont les “ainés”

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <style>
5      p:first-child{
6          color:red;
7          font-weight:bold;
8      }
9  </style>
10 </head>
11 <body>
12 <p>Test</p>
13 <p>Test 2</p>
14 <div>
15     <p>Test 3</p>
16 </div>
17 </body>
18 </html>
19
```



SELECTEURS : ::FIRST-LETTER

- La première lettre de la textnode d'un élément



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       p:first-letter{
6         color:red;
7         font-weight:bold;
8       }
9     </style>
10  </head>
11  <body>
12    <p>Test</p>
13    <p>Test 2</p>
14    <div>
15      <p>Test 3</p>
16    </div>
17  </body>
18 </html>
19
```



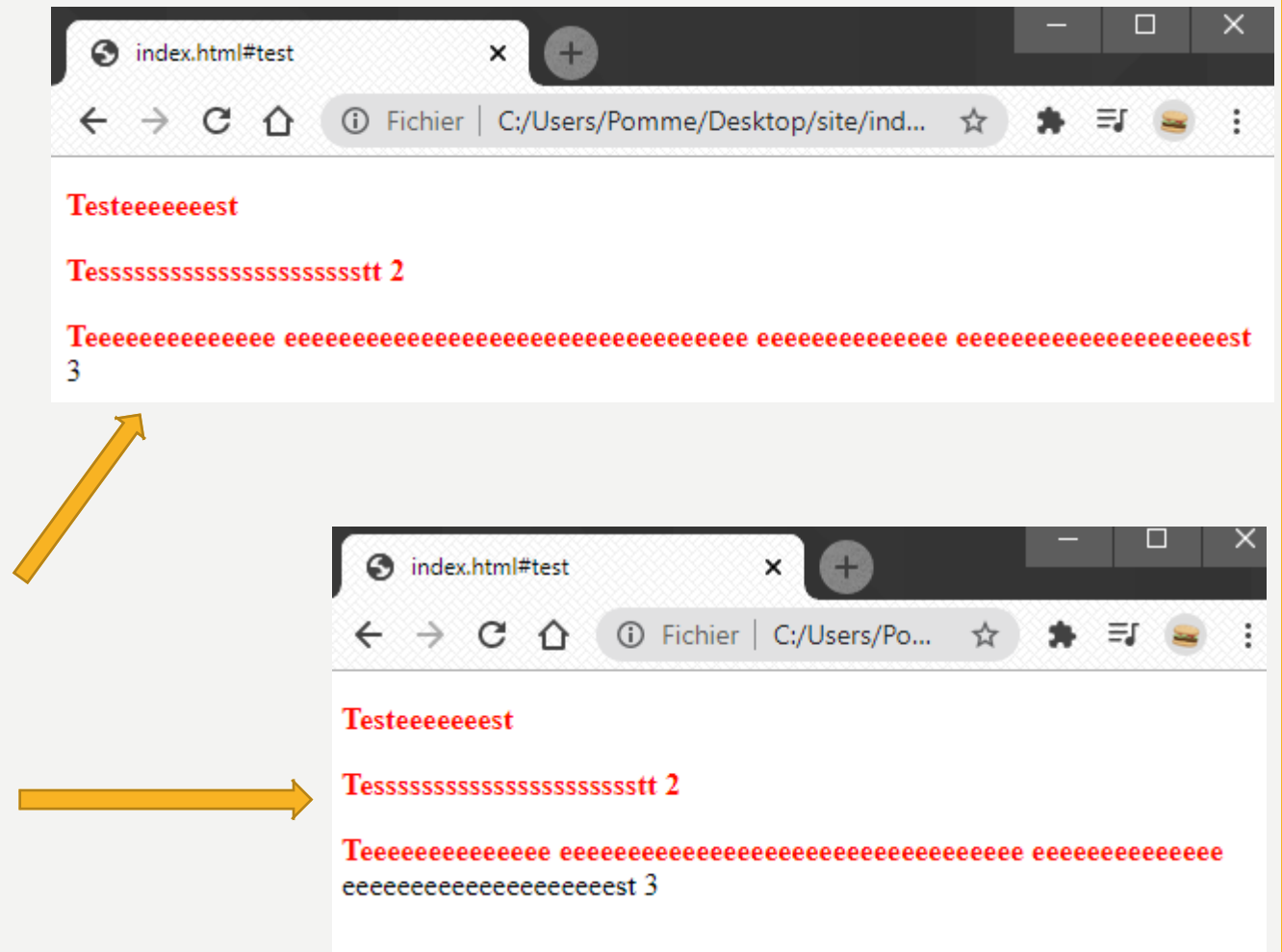
SELECTEURS : ::FIRST-LINE

- La première ligne
- Réactif au redimensionnement

```

2  = <html>
3  =     <head>
4  =         <style>
5             p:first-line{
6                 color:red;
7                 font-weight:bold;
8             }
9         </style>
10    </head>
11    <body>
12        <p>Testeeeeeeeest</p>
13        <p>Tesssssssssssssssssssssssstt 2</p>
14        <div>
15            <p>Teeeeeeeeeeeeeeeeee eeeeeeeeeeeeeee
16        </div>
17    </body>
18 </html>
19

```



SELECTEURS : :FIRST-OF-TYPE

- Le premier élément de ce type dans la fraternité

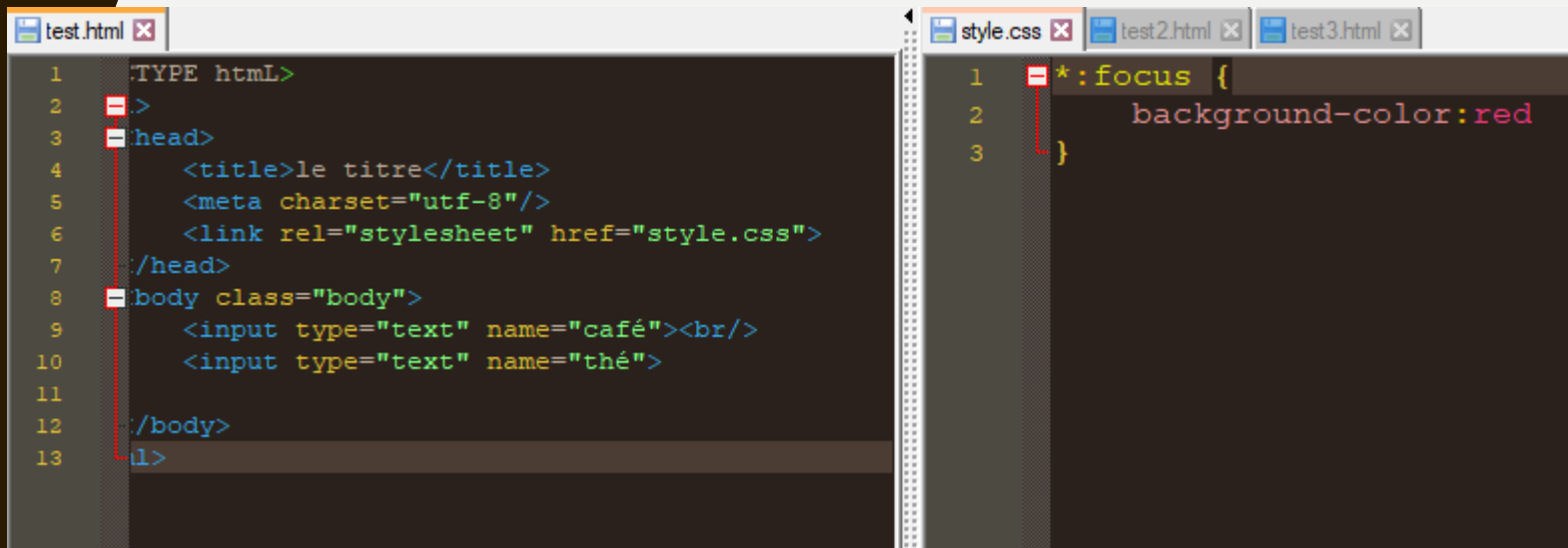
```
9      <p>test</p>
10     <p>Tesssssstt 2</p>
11     <div>
12         <p>Teeet 3</p>
13     </div>
14 </body>
```

```
p:first-of-type { color:red;}
```



SELECTEURS : :FOCUS

- L'élément qui possède le focus, qui est sélectionné



The screenshot shows a code editor with two files open. The left file, 'test.html', contains the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>le titre</title>
5   <meta charset="utf-8"/>
6   <link rel="stylesheet" href="style.css">
7 </head>
8 <body class="body">
9   <input type="text" name="café"><br/>
10  <input type="text" name="thé">
11
12 </body>
13 </html>
```

The right file, 'style.css', contains the following CSS code:

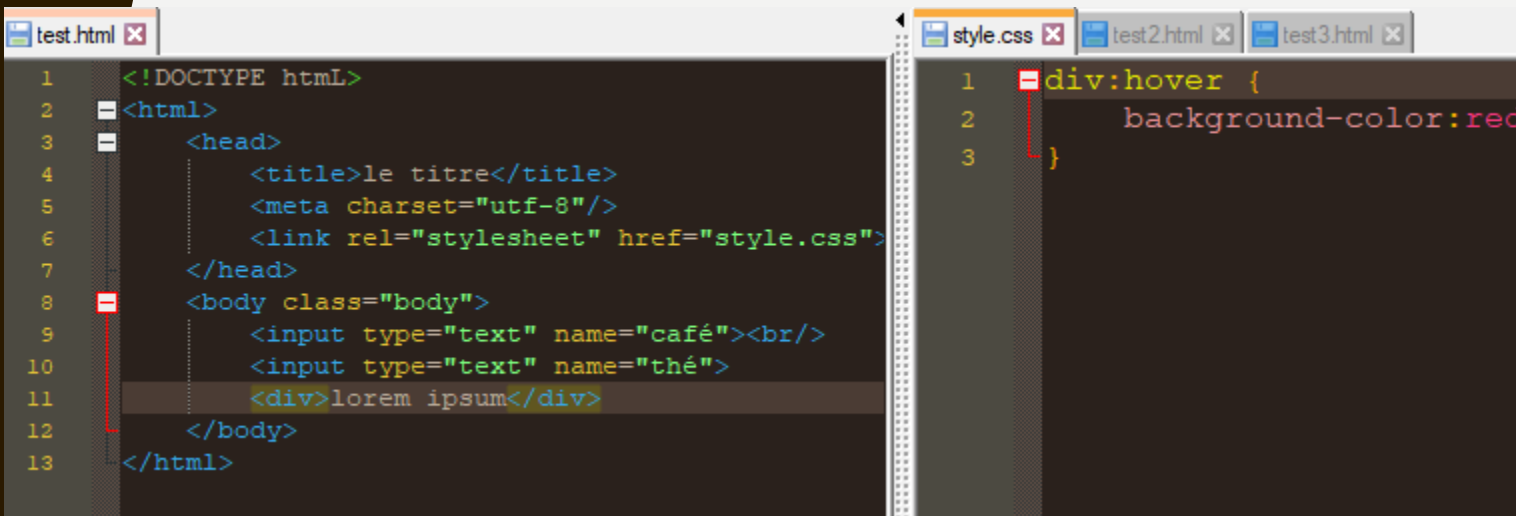
```
1 *:focus {
2   background-color:red
3 }
```

A red line in the HTML file connects the first text input to the CSS rule.



SELECTEURS : :HOVER

- L'élément survolé par la souris



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>le titre</title>
5   <meta charset="utf-8"/>
6   <link rel="stylesheet" href="style.css">
7 </head>
8 <body class="body">
9   <input type="text" name="café"><br/>
10  <input type="text" name="thé">
11  <div>lorem ipsum</div>
12 </body>
13 </html>
```

```
1 div:hover {
2   background-color:red
3 }
```

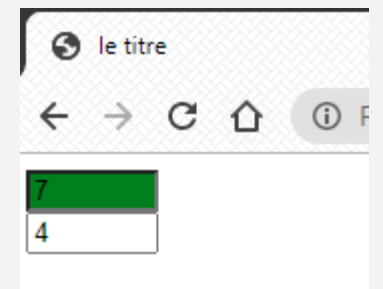


SELECTEURS : :IN-RANGE

- Ensemble qui se trouve quelque part dans la “cadèterie”

```
test.html x | style.css x | test2.html x | test3.html x
1 DOCTYPE html>
2 <html>
3 <head>
4   <title>le titre</title>
5   <meta charset="utf-8"/>
6   <link rel="stylesheet" href="style.css">
7 </head>
8 <body class="body">
9   <input min="5" max="10" type="number" /><b
10  <input min="5" max="10" type="number" /><b
11 </body>
12 </html>

1 input:in-range {
2   background-color:green;
3 }
```

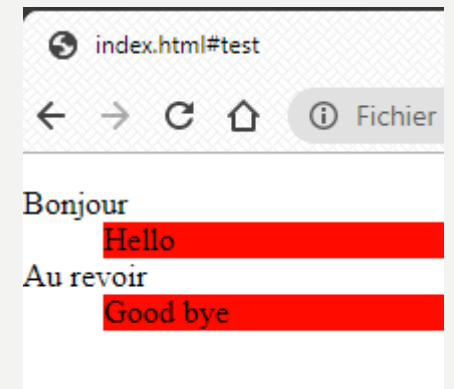


SELECTEURS : :LANG(..)

- Les éléments dont l'attribut lang vaut la valeur entre parenthèses

```
test.html | style.css
<!DOCTYPE html>
<html>
  <head>
    <style>
    </style>
    <link rel="stylesheet" href="css"
  </head>
  <body>
    <dl>
      <dt>Bonjour</dt>
      <dd lang="en">Hello</dd>
      <dt>Au revoir</dt>
      <dd lang="en">Good bye</dd>
    </dl>
  </body>
</html>

1 :lang(en) {
2   background-color:red;
3 }
4
5
```

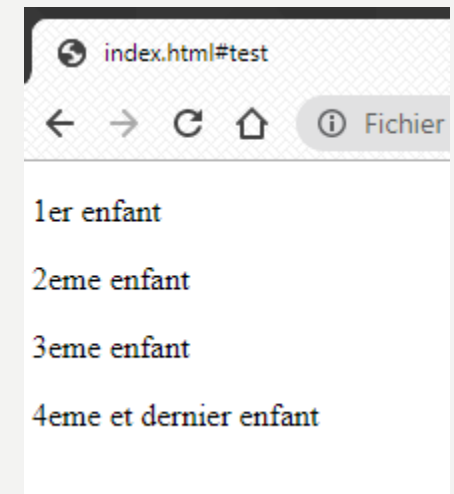


SELECTEURS : :LAST-CHILD

- Quand un élément est le dernier de la fraternité, ici c'est div, donc il n'y a pas de paragraphe en rouge

```
index.html x test.html x style.css x
1  DOCTYPE html>
2  <html>
3  <head>
4  <style>
5  </style>
6  <link rel="stylesheet" href="css/st
7  </head>
8  <body>
9  <div>
10 <p>1er enfant</p>
11 <p>2eme enfant</p>
12 <p>3eme enfant</p>
13 <p>4eme et dernier enfant</p>
14 <div></div>
15 </div>
16 </body>
17 </html>
18

1  p:last-child{
2      background-color:red;
3  }
```



SELECTEURS : :LAST-OF-TYPE

- Le dernier élément de ce type dans une fraternité

```
index.html x test.html x style.css x
1 DOCTYPE html>
2 <html>
3 <head>
4 <style>
5 </style>
6 <link rel="stylesheet" href="css/st
7 </head>
8 <body>
9 <div>
10 <p>1er enfant</p>
11 <p>2eme enfant</p>
12 <p>3eme enfant</p>
13 <p>4eme et dernier enfant</p>
14 <div></div>
15 </div>
16 </body>
17 </html>
18

1 p:last-child{
2     background-color:red;
3 }
4
5
```

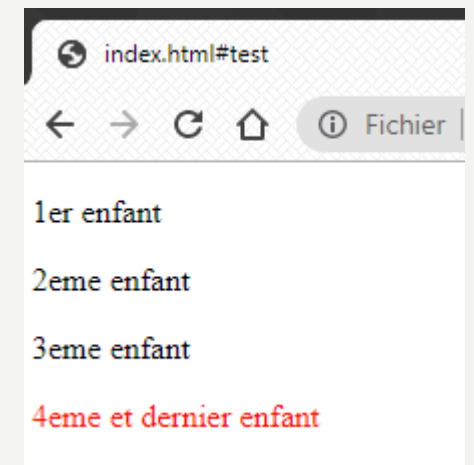


SELECTEURS : :NOT(..)

- Les éléments qui ne sont pas du type indiqué entre parenthèses

```
index.html x test.html x style.css x
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5     </style>
6     <link rel="stylesheet" href="css"
7   </head>
8   <body>
9     <p>1er enfant</p>
10    <p>2eme enfant</p>
11    <p>3eme enfant</p>
12    <div>4eme et dernier enfant</div>
13    <div></div>
14  </body>
15 </html>

1 :not (div) {color:red;}
2 p{color:black;}
3
4
```

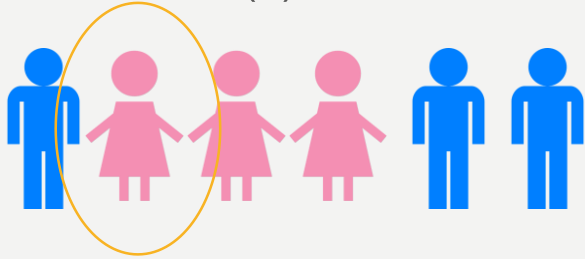


SELECTEURS : :NTH-

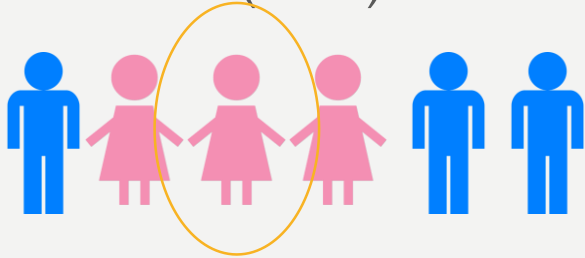
- `child(n)` : n-ème enfant
- `Last-child(n)` : n-ème enfant en partant de la fin
- `of-type(n)` : n-ème enfant de ce type
- `Last-of-type(n)` : n-ème enfant de ce type en partant de la fin
- `N`, peut-être...
 - Un nombre
 - Une formule $an + b \rightarrow 3n+2 \rightarrow 2, 5, 8, 11, \dots$
 - Un mot clef (odd : impaire) (even : paire)

SELECTEURS : :NTH-CHILD

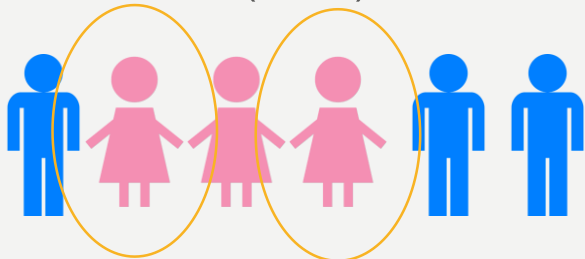
- fille:nth-child(2)



- fille:nth-child(2n+1)

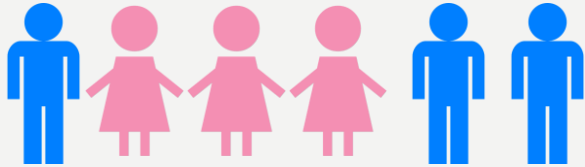


- fille:nth-child(even)

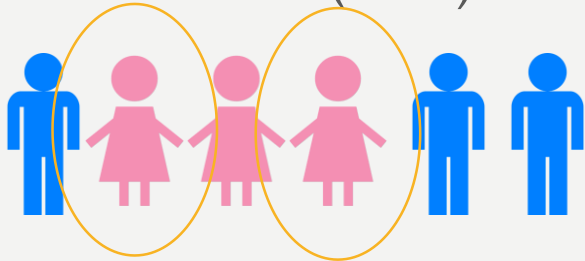


SELECTEURS : :NTH-LAST-CHILD

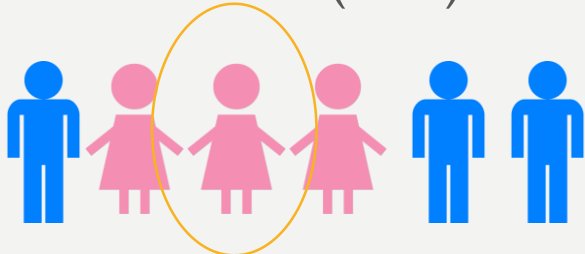
- fille:nth-last-child(2)



- fille:nth-last-child(2n+1)

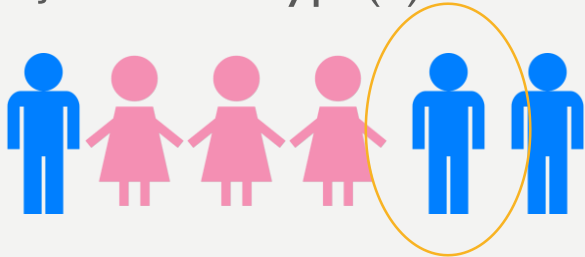


- fille:nth-last-child(even)

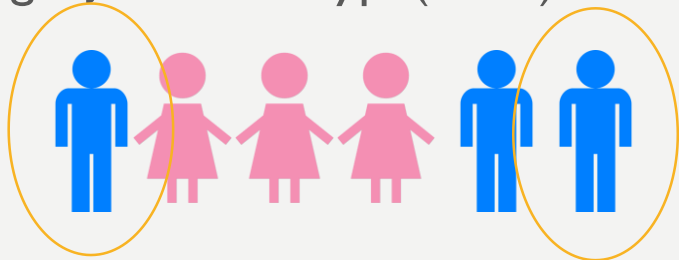


SELECTEURS : :NTH-OF-TYPE

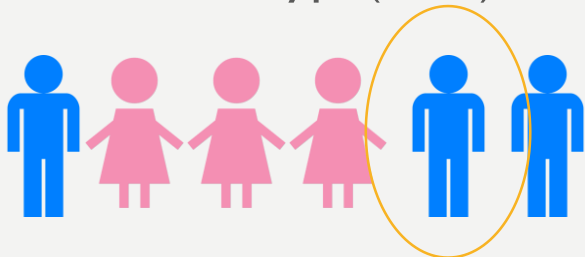
- garçon:nth-of-type(2)



- garçon:nth-of-type(2n+1)

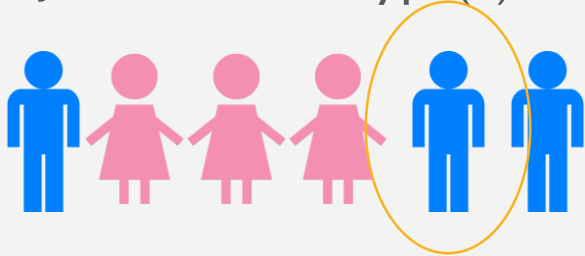


- garçon:nth-of-type(even)

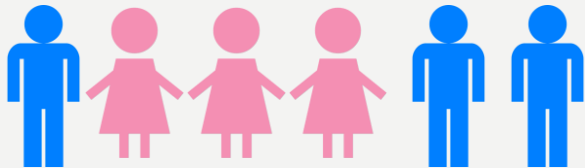


SELECTEURS : :NTH-LAST-OF-TYPE

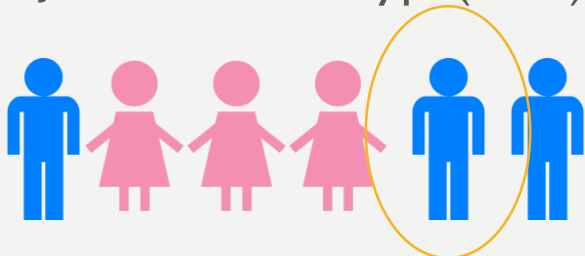
- garçon:nth-last-of-type(2)



- garçon:nth-last-of-type(2n+)

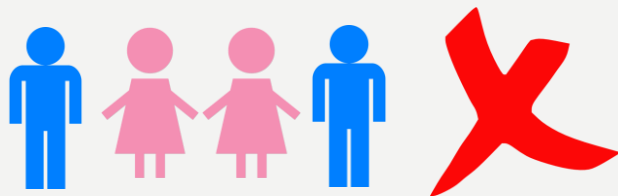


- garçon:nth-last-of-type(even)



SELECTEURS : :ONLY-OF-TYPE

- Le seul fils de ce type



garçon:only-of-type

SELECTEURS : :ONLY-CHILD

- Ensemble qui se trouve quelque part dans la “cadèterie”



SELECTEURS : LES INPUT

- `:required` : les inputs avec l'attribut *required*
- `:read-write` : les inputs qui ne sont pas en *readonly*
- `:read-only` : les inputs qui sont en *readonly*
- `::placeholder` : les inputs qui ont un *placeholder* spécifié
- `:out-of-range` : les inputs qui sont hors des limites, contraire de *in-range*
- `:optional` : les inputs qui ne sont pas *required*
- `:valid` : les inputs valides par rapport à leur type (email, numerique, min/max, etc.)
- `:invalid` : les inputs invalides

SELECTEURS : :ROOT

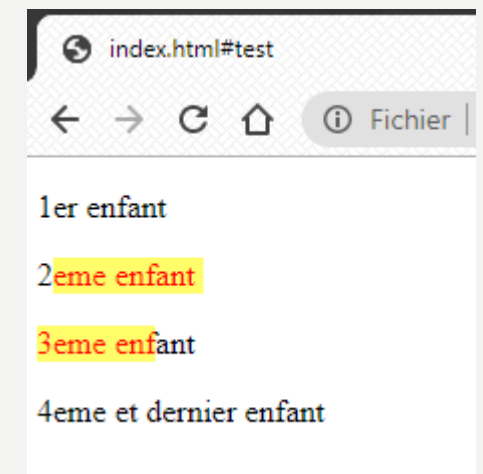
- La racine, très pratique pour créer des variables
- Appliquer des styles pour bénéficier de l'héritage

SELECTEURS : :SELECTION

- Change le style du texte sélectionné à la souris

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5     </style>
6     <link rel="stylesheet" href="css
7   </head>
8   <body>
9     <p>1er enfant</p>
10    <p>2eme enfant</p>
11    <p>3eme enfant</p>
12    <div>4eme et dernier enfant</div>
13    <div></div>
14  </body>
15 </html>
```

```
1 ::selection {
2   color: red;
3   background: yellow;
4 }
```



SELECTEURS ATTRIBUTS

- N'oublions pas qu'on peut créer nos propres attributs !

[attribute]	[target]	Sélectionne les éléments avec un attribut <i>target</i>
[attribute=value]	[target=_blank]	Dont l'attribut <i>target</i> vaut "_blank"
[attribute~=value]	[title~=flower]	Attribut <i>title</i> contenant le mot "flower"
[attribute =value]	[lang =en]	Attribut <i>lang</i> commençant par "en"
[attribute^=value]	a[href^="https"]	Idem =
[attribute\$=value]	a[href\$=".pdf"]	Tous les éléments <i>a</i> ayant un attribut <i>href</i> qui finit par ".pdf"
[attribute*=value]	a[href*="w3schools"]	Tous les éléments <i>a</i> ayant un attribut <i>href</i> qui contient "w3schools" quelque part

SCORE DE SÉLECTION

- <https://specificity.keegan.st/>
- !important --- 4
- Inline --- 3
- Id --- 2
- Class/Selecteur d'attributs/PseudoClass --- 1
- Element --- 0
- Important – inline – id – class – element
- `p ~ div.test:hover` : 0 – 0 – 0 – 2 – 2

```
2 p{
3   color:red !important;
4 }
```

with Plankton, Fish and Sharks

 universal selector 0-0-0	 1 element 0-0-1	 2 elements 0-0-2	 14 elements 0-0-14
 1 class 0-1-0	 1 universal selector 1 class 0-1-0	 1 universal selector 1 attribute selector 0-1-0	 1 pseudo-class 0-1-0
 1 element 1 class 0-1-1	 1 element 1 attribute 0-1-1	 2 elements 1 pseudo-class 0-1-2	 2 elements 1 attribute 0-1-2
 1 element 1 class 1 pseudo-class 0-2-1	 1 element 1 class 1 attribute 0-2-1	 10 class/attribute/pseudo-classes 0-10-0	 1 ID Selector 1-0-0
 2 types 2 class/attribute 1 ID Selector 1-2-2	 2 ID Selectors 1 type selector 2-0-1	 inline style 1-0-0-0	 !important 1-0-0-0-0

X-0-0: The number of ID selectors
 0-Y-0: The number of class selectors, attributes selectors, and pseudo-classes
 0-0-Z: The number of type selectors and pseudo-elements
 *, +, >, ~: The universal selector has no value and combinators do not increase specificity
 !not(x): The negation selector has no value, but the argument passed increases specificity

@estelleweyl * @estelleweyl * www.standardista.com * 2012

HERITAGE

- Toutes les propriétés ne sont pas héritables
 - inherit : héritage actif
 - Initial : style par défaut du navigateur
 - Unset : valeur par défaut de la propriété
 - Revert : retour vers son ancien style
- Ça peut être utile pour faciliter les changements : on définit la color dans le body puis le reste on fait du revert et si un jour on veut changer le style de tout le site, on n'aura plus qu'à modifier le color du body !

CHAPITRE 2 : FORMATER DU TEXTE

FONT-

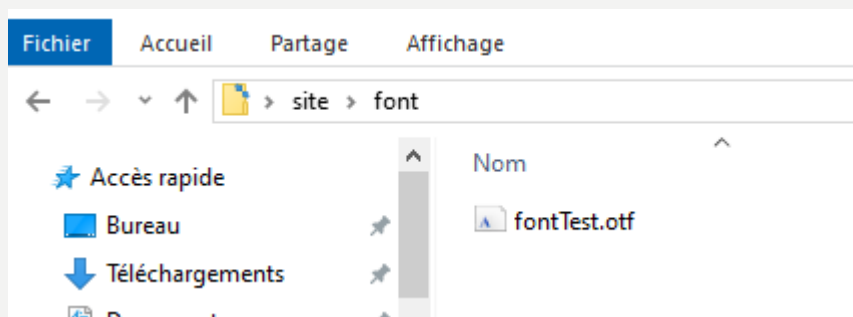
- font-style : orientation
- font-weight : épaisseur
- font-family : la police d'écriture
- font-size : taille
- ET d'autres... dans la doc

FONT FAMILY

- Web safe fonts, présents dans tous les navigateurs
 - https://www.w3schools.com/cssref/css_websafe_fonts.asp
- Serif, l'écrire historique avec empattements
 - Time New Roman : Exemple d'un texte avec empattements
- Sans serif, sans empattements
 - Arial : Exemple d'un texte sans serif
- Monospace (MS), chaque caractère occupe la même place
 - Lucida Console : Exemple d'un texte monospace

FONT FAMILY

- Importer sa propre police d'écriture



FONT FAMILY

- @font-face
 - *font-family* : On donne un nom à notre police
 - *src* : On charge le fichier de la police télécharger
- On réutilise notre police fraîchement créée avec la propriété *font-family*

```
style.css x
1  @font-face {
2      font-family: maFont;
3      src: url(../font/fontTest.otf);
4  }
5  div{
6      font-family:maFont;
7      font-size:20pt;
8  }
```



FONT FAMILY

- Les polices d'écriture Google
 - https://www.w3schools.com/howto/howto_google_fonts.asp

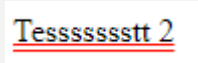
```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <style>
5  </style>
6  <link rel="stylesheet" href="css/style.css" />
7  <link href='https://fonts.googleapis.com/css?family=Aclonica' rel='stylesheet'>
8  </head>
9  <body style="font-family:Aclonica">
10 <div id="div1">Une div avec du texte</div>
11 </body>
12 </html>
13
```



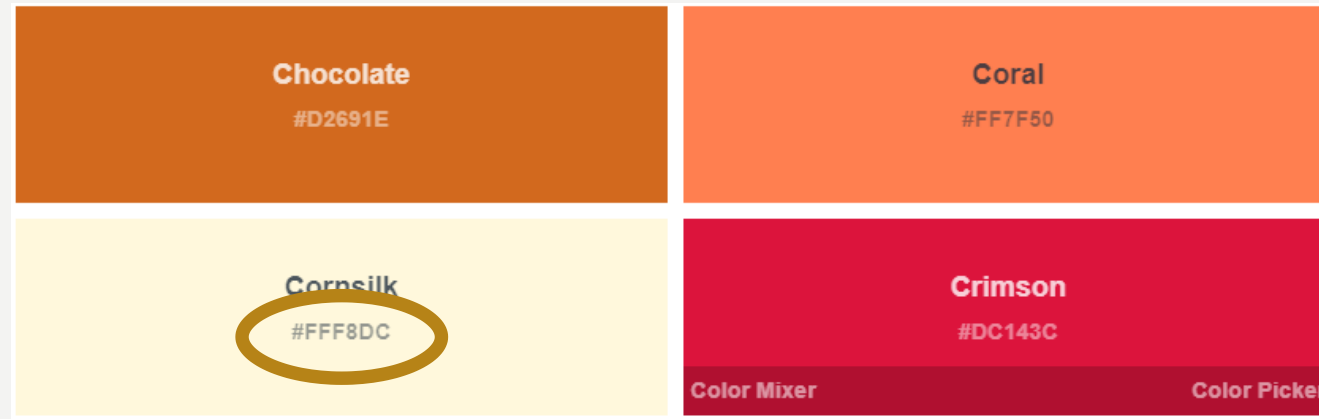
TEXT-

- *text-align* : alignement du paragraphe
- *text-align-last* : alignement de la dernière ligne d'un paragraphe
- *text-decoration*:

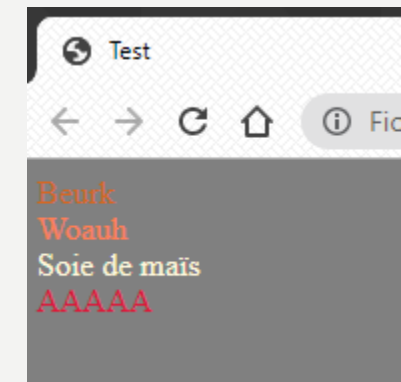
```
p {  
    text-decoration: underline red double;  
}
```


- *text-justify* : la méthode de justification
- *text-indent* : marge de la première ligne d'un paragraphe
- *text-overflow* : pour gérer le dépassement
- *text-shadow* : ajouter une ombre
- *text-transform* : mettre en majuscule, ou en minuscule ou en capitale

LES COULEURS MOT-CLEF



```
9  <body>
10    <div style="color:chocolate">Beurk</div>
11    <div style="color:coral">Woauh</div>
12    <div style="color:cornsilk">Soie de maïs</div>
13    <div style="color:crimson">AAAAA</div>
14  </body>
```



LES COULEURS

- Un mélange de trois couleurs : rouge vert bleu (RVB) red green blue (RGB)
- Chaque couleur est plus ou moins présente (de 0 à 255)
- Une couleur est donc la composition de différents niveaux de rouge/vert/bleu
- 0 rouge, 0 vert, 0 bleu donne du noir
- 255 rouge, 0 vert, 0 bleu donne du rouge
- 255 rouge, 255 vert, 255 bleu donne du blanc
- 125 rouge, 125 vert, 125 bleu donne du ...
- Pourquoi 0 à 255 ?

LA REPRÉSENTATION HEXADÉCIMAL

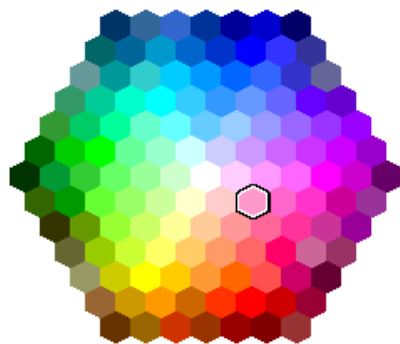
- On compte en décimal : base 10 (10 symboles)
 - 0,1,2,3,4,5,6,7,8,9
- L'ordinateur compte en binaire : base 2 (2 symboles)
 - 0,1
- Les couleurs peuvent être codées en hexadécimal : base 16 (16 symboles)
 - 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f
- En base 10 : $9+1 = 10$ ET $19+1 = 20$
- En binaire : $1+1 = 10$ ET $11+1 = 100$
- En base 16 : $9+1 = a$ ET $f+1 = 10$ ET $ff+1 = ???$
- Quand on atteint le dernier symbole on repasse à 0 et on fait passer la retenue de 1 à gauche
 - F = ..00000F
 - 1 = ..0001

LA REPRSENTATION HEXADECIMAL

- Les opérations pour repasser vers le décimal
 - $5AF = 15^2 \times 5 + 15^1 \times 10 + 15^0 \times 15 = 225 \times 5 + 150 + 15 = 1290$
 - En binaire, $100 = 2^2 \times 1 + 2^1 \times 0 + 2^0 \times 0 = 4$
- Décimal vers hexadécimal (Valable pour les valeurs entre 255)
 - $250 \% 16 = 10$
 - $250 - 10 / 16 = 15$
 - 15 c'est F en hexa et 10 c'est A
 - $250 = FA$

LES COULEURS

Pick a Color:

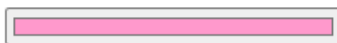


Or Enter a Color:

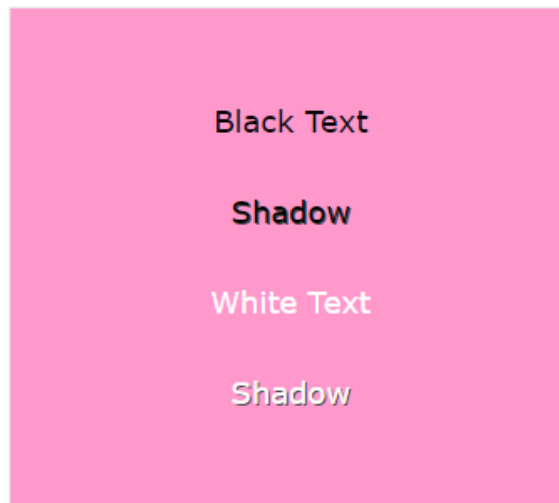
Color value

OK

Or Use HTML5:



Selected Color:



#ff99cc

rgb(255, 153, 204)

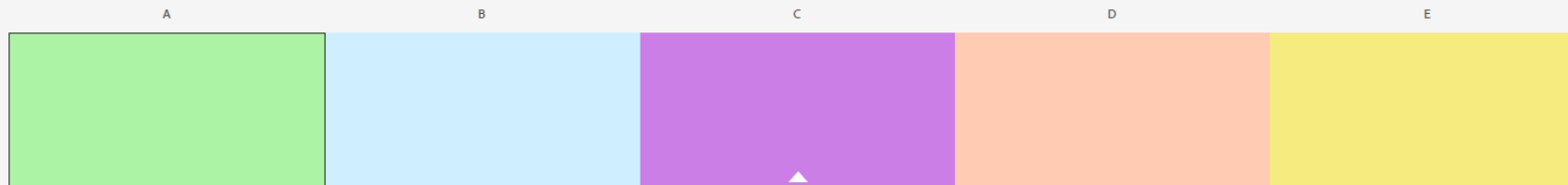
hsl(330, 100%, 80%)

Lighter / Darker:

100%	#ffffff
95%	#ffe6f2
90%	#ffcce6
85%	#ffb3d9
80%	#ff99cc
75%	#ff80bf
70%	#ff66b3
65%	#ff4da6
60%	#ff3399
55%	#ff1a8c
50%	#ff0080
45%	#e60073
40%	#cc0066
35%	#b30059
30%	#99004d
25%	#800040
20%	#660033
15%	#4d0026
10%	#33001a
5%	#1a000d
0%	#000000

CHARTE GRAPHIQUE

- Police d'écriture
- Couleurs
 - Couleurs complémentaires
 - <http://infohound.net/colour/>
 - <https://color.adobe.com/create/color-wheel>

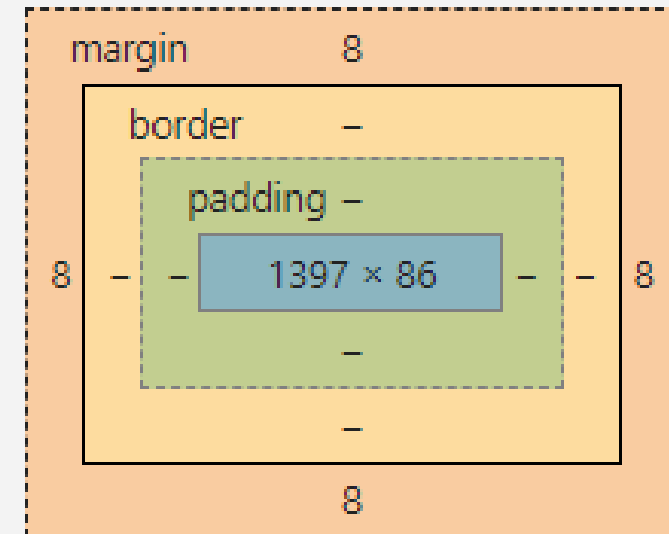


- Images

CHAPITRE 3 : LES BOITES

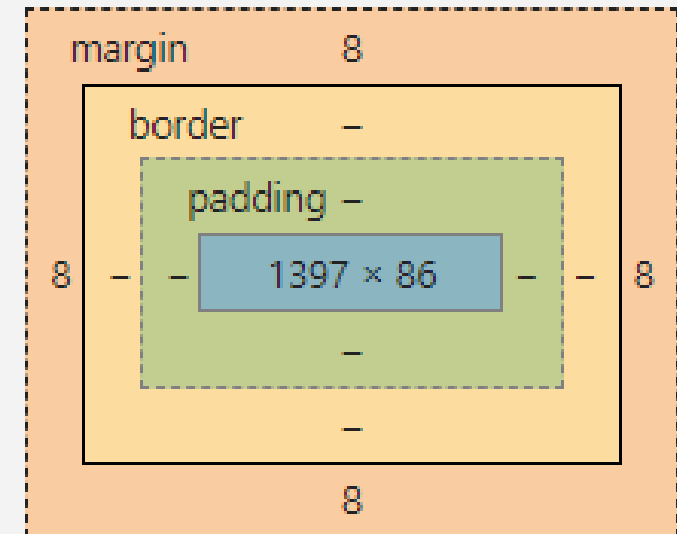
MARGE

- Margin : marge extérieur
- Bordure : avec une épaisseur
- Padding : marge intérieur
- La boîte
- La taille s'applique à la boîte (la partie bleue)
- *margin : auto*



BORDURE

- Un style
- Une épaisseur
- Une couleur
- On peut l'arrondir
- On peut mettre une image
- Box-sizing : border-box → La taille s'applique à partir de la bordure plutôt que le contenu



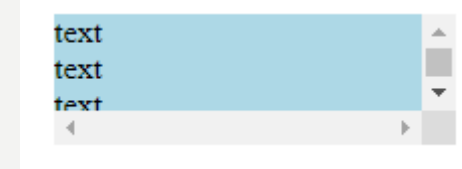
DEPASSEMENT

- *overflow*

- *visible*
- *hidden*
- *scroll*
- *auto*



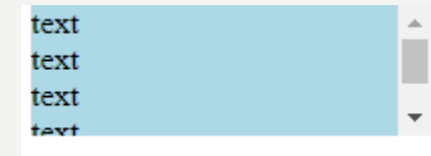
visible



scroll

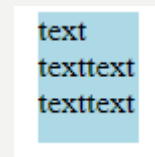
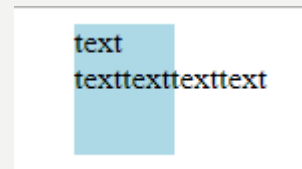


hidden



auto

- *word-wrap: break-word;*



- *text-overflow* : style pour indiquer comment est affiché le text masqué (... par exemple)

DIV ET SPAN

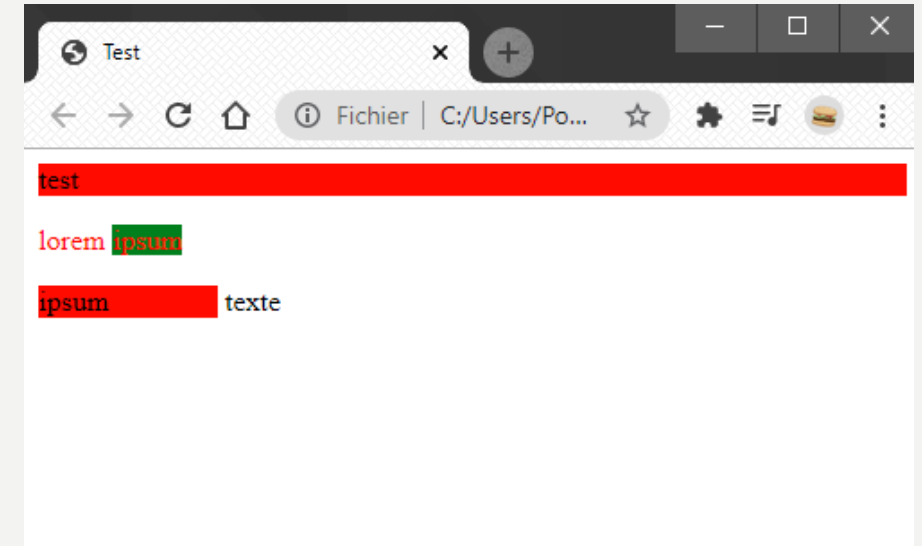
- Limiter l'usage car ils ne bénéficient d'aucune sémantique
- Mais très pratiques pour appliquer des styles
- La div pour des blocs
- Le span pour des morceaux de lignes

BLOCK, INLINE, INLINE-BLOCK

- Display
- Block
 - Passage à ligne obligatoire
 - Remplit automatiquement toute la largeur du parent
- Inline
 - S'insère dans la ligne
 - S'adapte au contenu
 - Ne peut-être redimensionner
- Inline-block
 - Un inscandalline redimensionnable

BLOCK, INLINE, INLINE-BLOCK

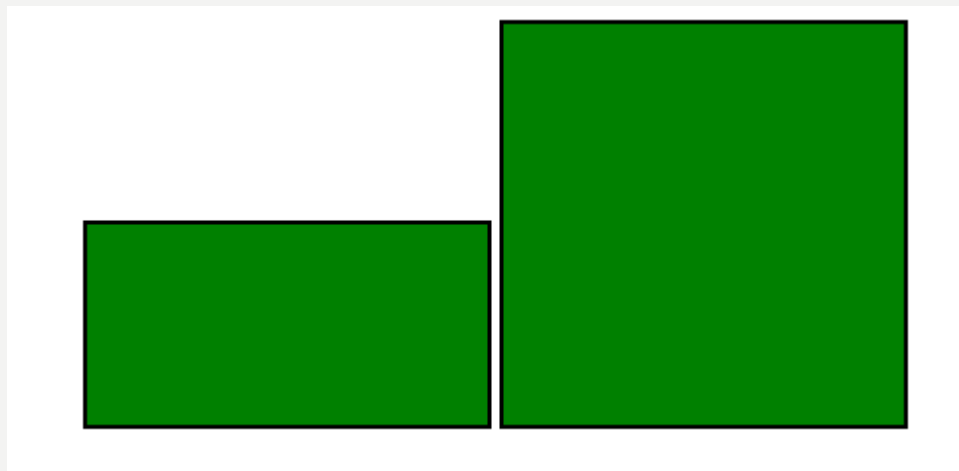
```
index.html x test.html x
1 <body>
2   <div>test</div>
3   <p> lorem <span> ipsum </span></p>
4   <div class="ib"> ipsum </div>
5   texte
6 </body>
7
style.css x
1 div{
2   background-color:red;
3 }
4 span{
5   background-color:green;
6   width:100px;
7 }
8 .ib{
9   display:inline-block;
10  width:100px;
11 }
```



BLOCK, INLINE, INLINE-BLOCK

- Inline-block
 - Baseline
 - Vertical-align

```
1  div {  
2    width: 200px;  
3    height: 100px;  
4    background-color: green;  
5    border: 2px solid black;  
6    display: inline-block;  
7    vertical-align: baseline;  
8  }  
9  
10 #myDiv {  
11   height: 200px;  
12 }
```



DIMENSION

- Height : la haute
- Width : la largeur
- % : taille en pourcentage par rapport à l'élément parent
- Min et max : taille minimal et maximal, notamment pour le responsive
- `*{box-sizing : border-box};`

CHAPITRE 4 : LE POSITIONNEMENT

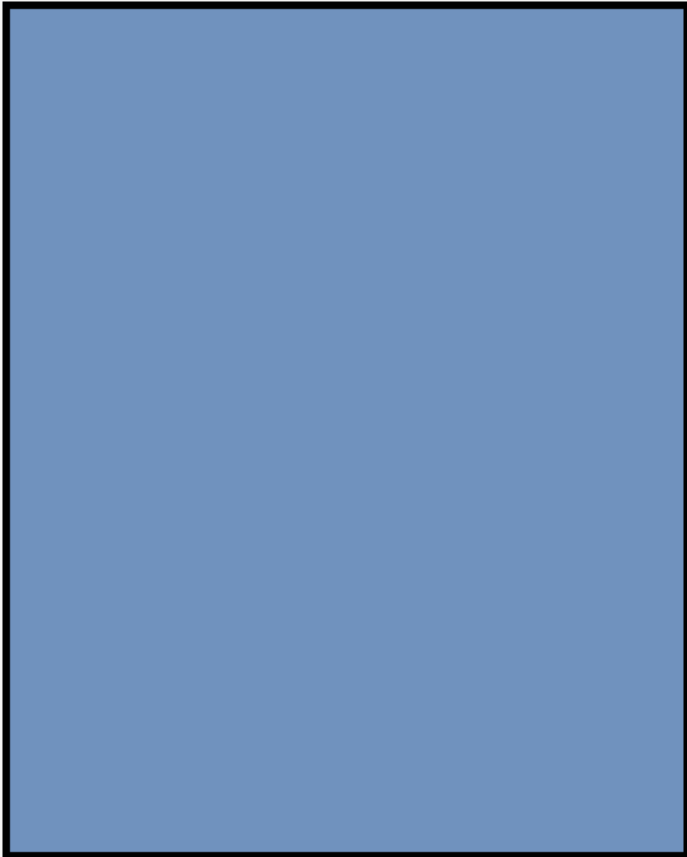
UN MOT SUR LE POSITIONNEMENT

- C'est l'une des fonctionnalités principales
- C'est à se tirer les cheveux
- Il n'y a pas qu'une seule solution
- On se retrouve parfois à "bidouiller"

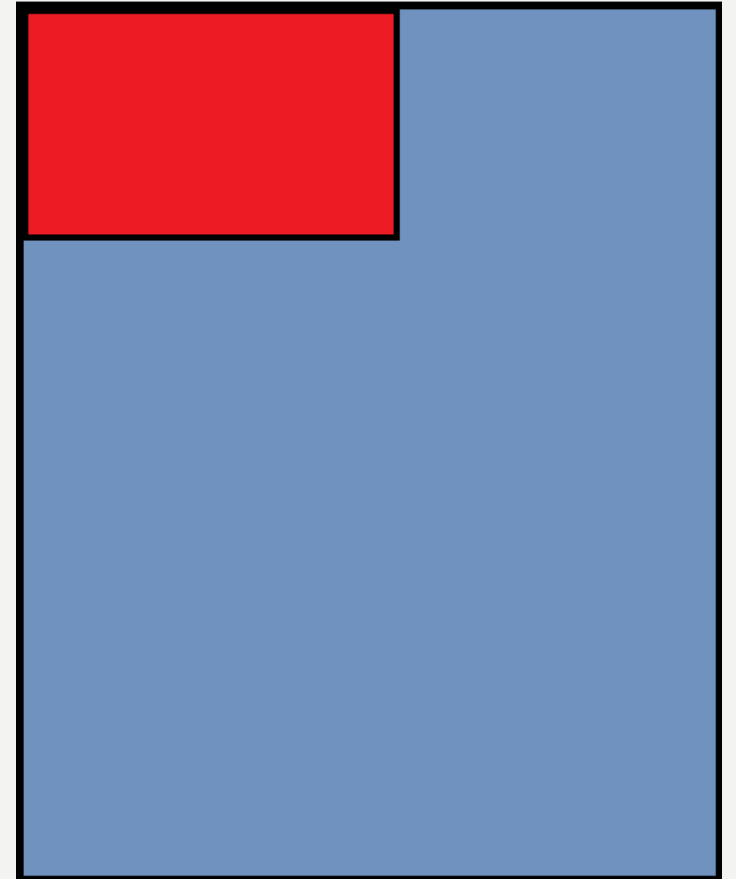
POSITIONNEMENT FLOTTANT

- float : left | right
- clear : left | right | both
- On place l'objet où c'est possible à gauche à droite

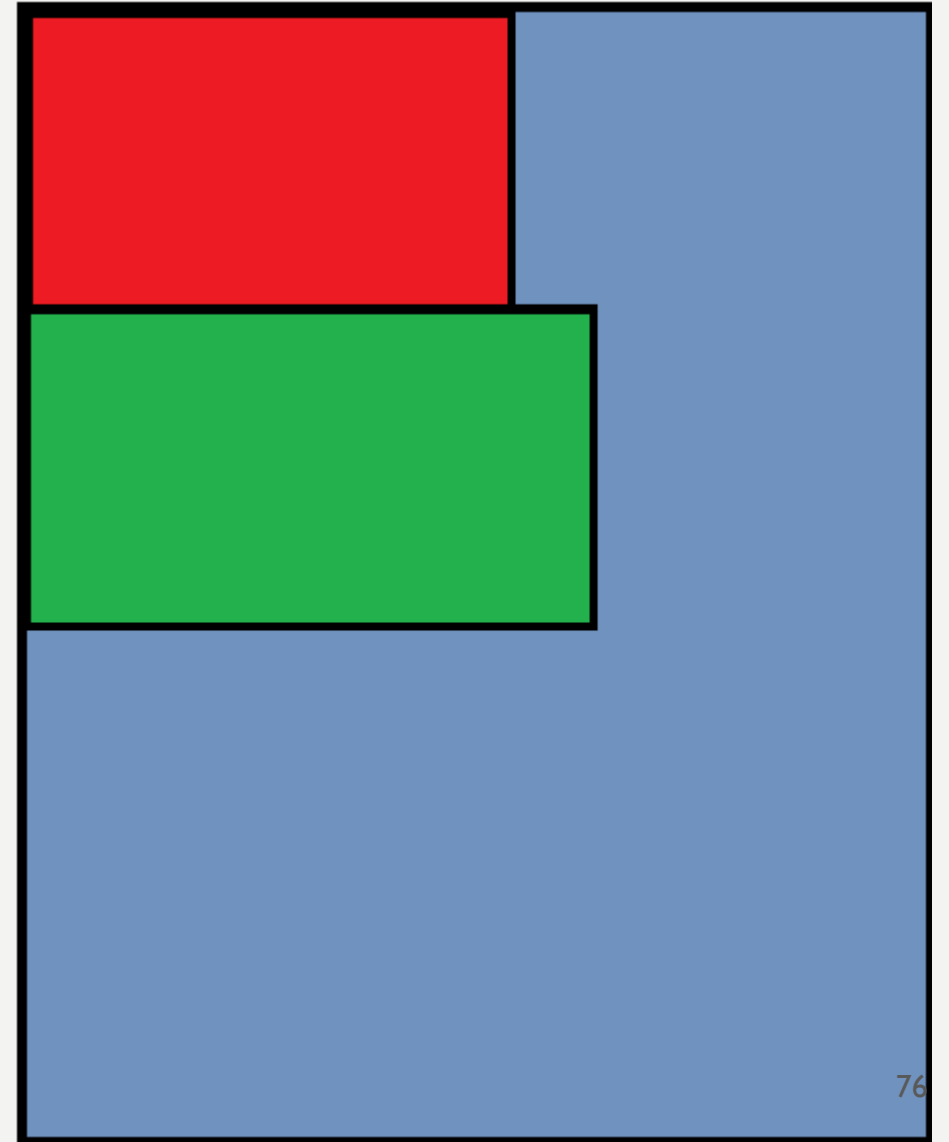
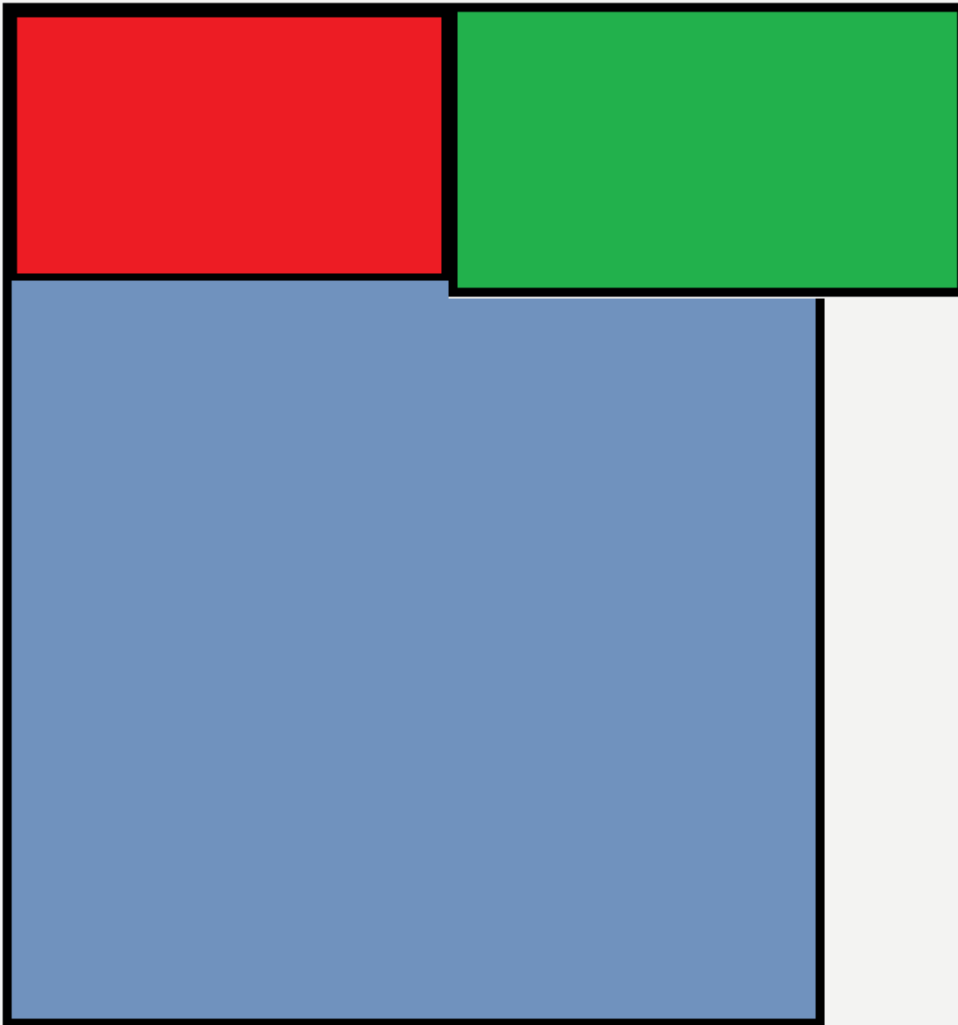
POSITIONNEMENT FLOTTANT



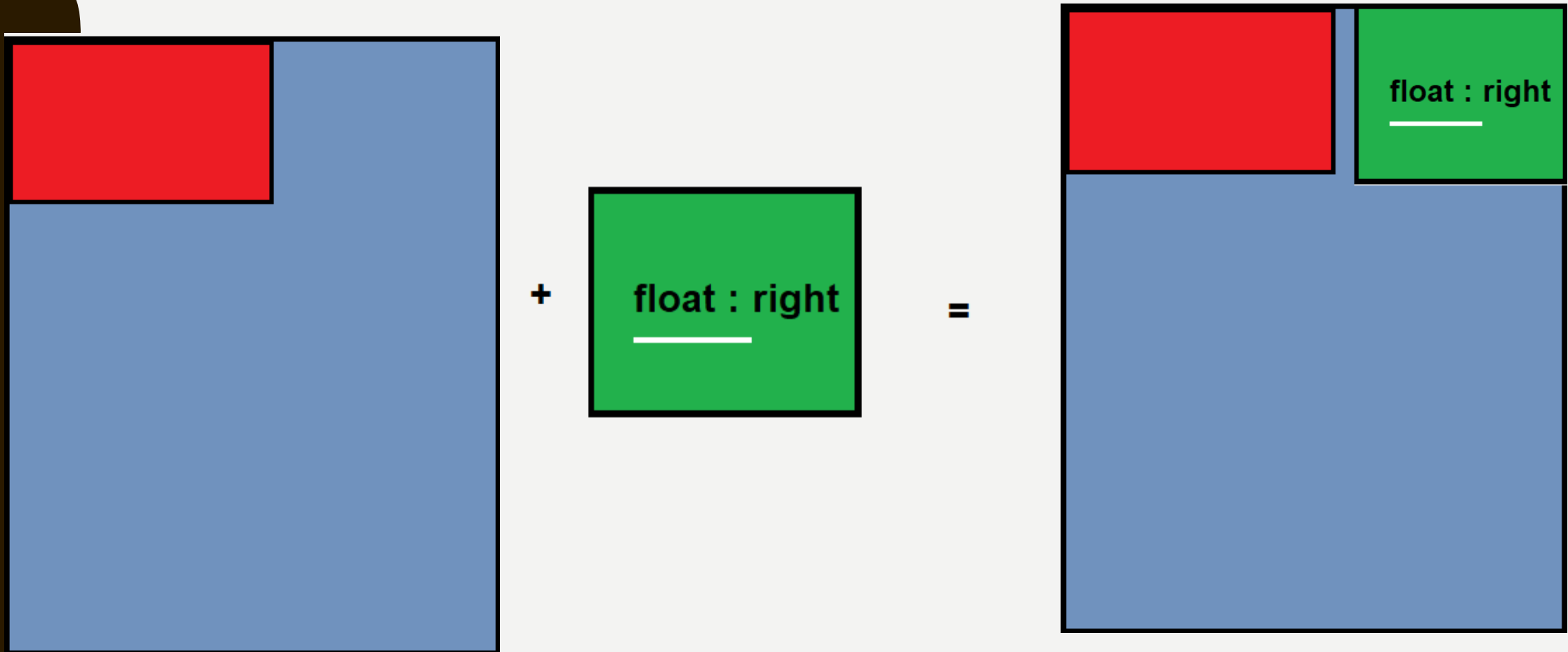
+



POSITIONNEMENT FLOTTANT

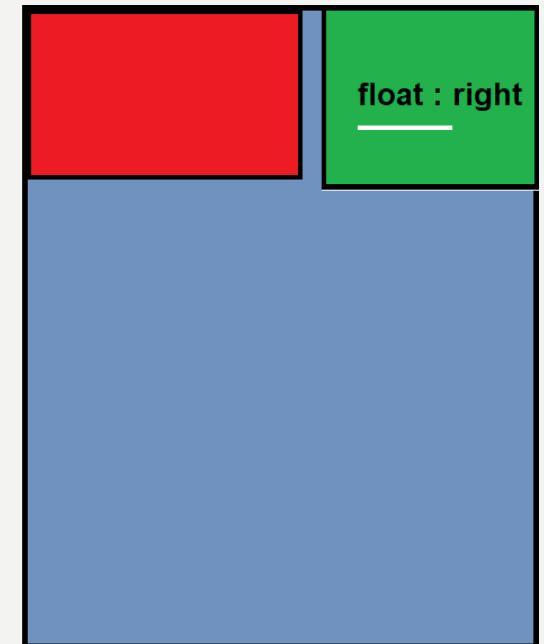


POSITIONNEMENT FLOTTANT



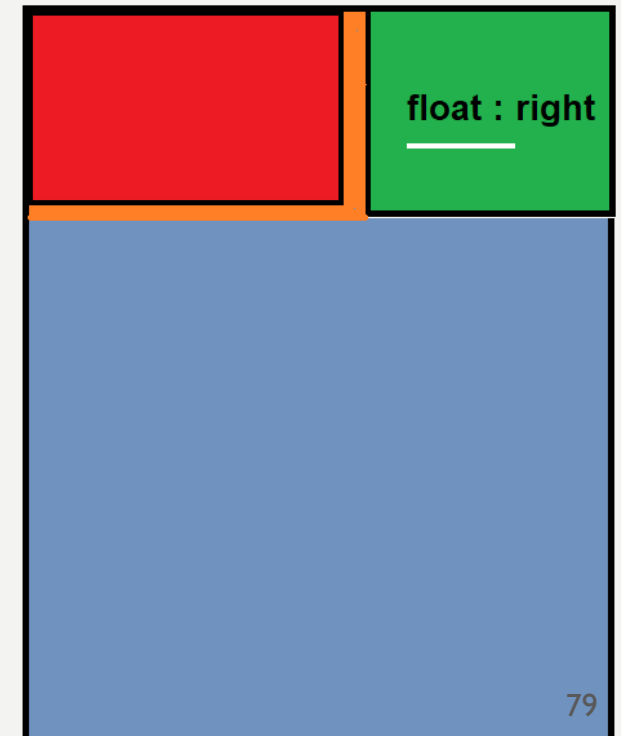
POSITIONNEMENT FLOTTANT

- L'espace entre les deux boites est de 25px
 - Si on rajoute une boite de 10px avec float il viendra se placer entre les deux
 - Si on rajoute une boite de plus de 25 px, il n'aura pas la place il passera donc en dessous du rouge
 - Si on ne met pas de float, il passera en dessous, sauf s'il est en inline ou inline-block
- clear : left | right | both
- On place l'objet où c'est possible à gauche à droite



POSITIONNEMENT FLOTTANT : CLEAR

- *right* et *left* sont deux flux différent
- *clear* consiste à remplir automatiquement les espaces vides dans le flux
- Les prochains éléments iront donc en dessous du bloc rouge par rapport au plus bas des éléments, ici le vert

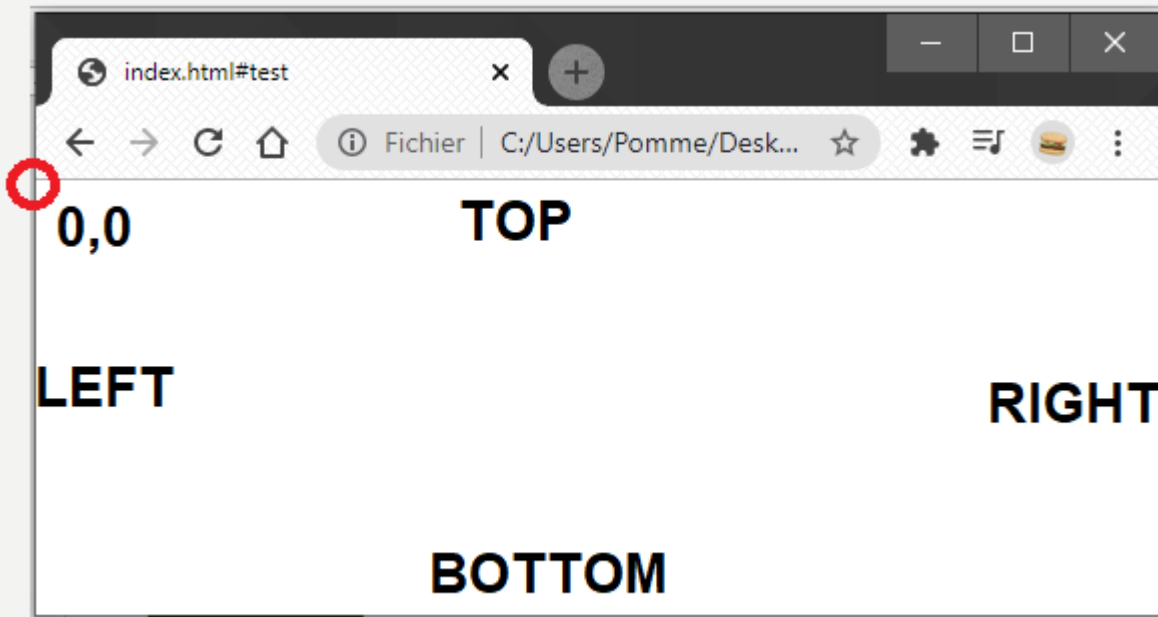


POSITION

- Absolute : positionnement dans l'absolu, des coordonnées précises
- Relative : positionnement par rapport à sa position normale
- Fixed : position par rapport à l'écran

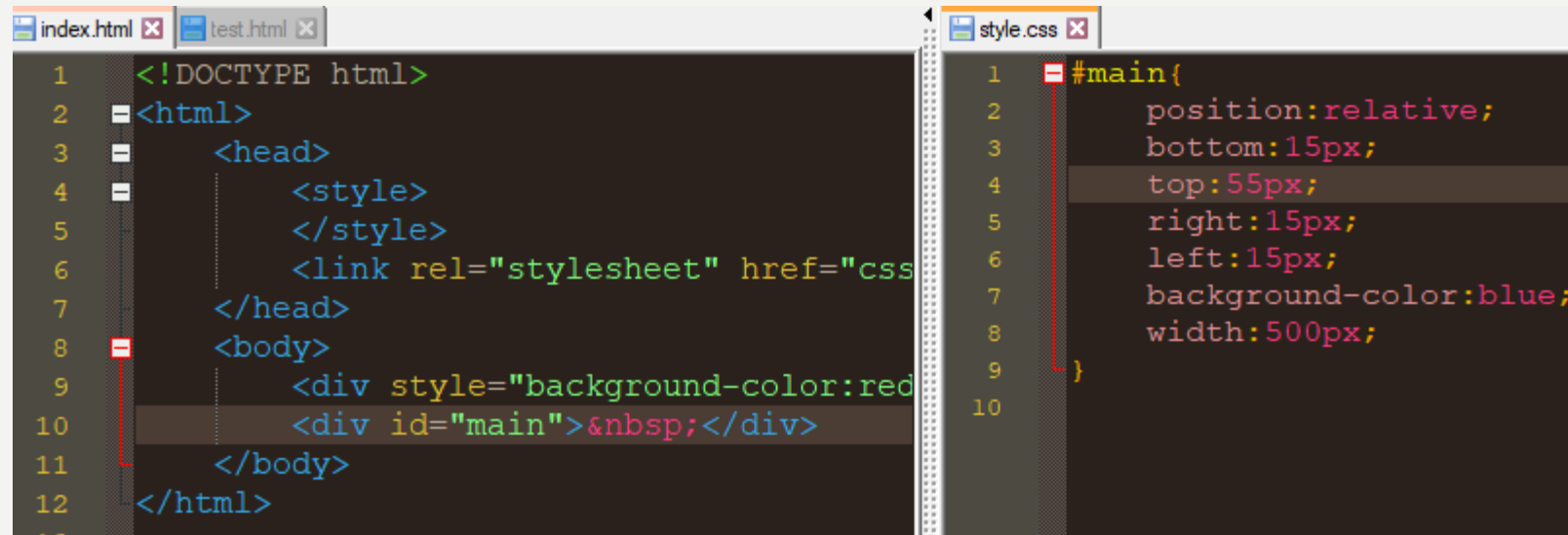
POSITION : ABSOLUTE

- Left, right, top, bottom
- Si on définit left et right, left gagne
- Si on définit top et bottom, la boîte va s'agrandir en haute pour remplir les deux propriétés



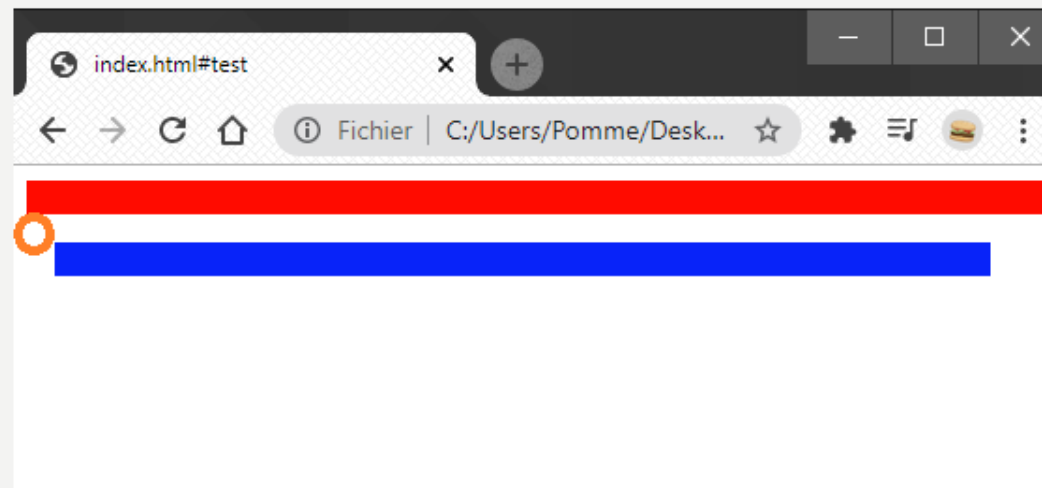
POSITION : RELATIVE

- On prend pour point de référence l'endroit où la boîte se serait placée par défaut



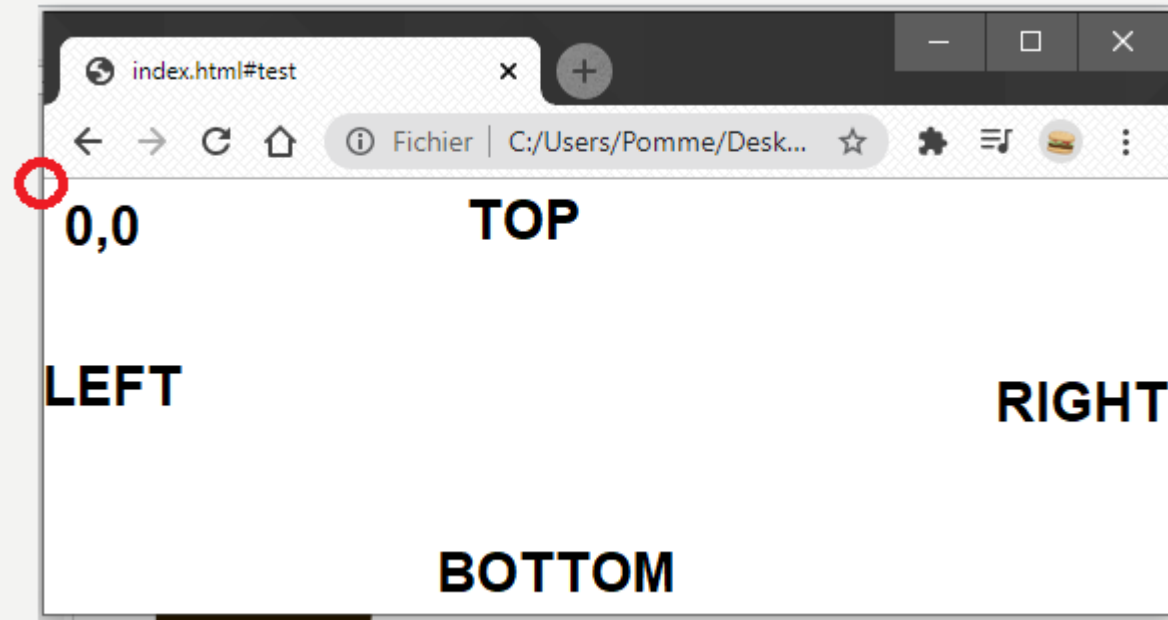
```
index.html x test.html x
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5     </style>
6     <link rel="stylesheet" href="css"
7   </head>
8   <body>
9     <div style="background-color:red"
10    <div id="main">&nbsp;</div>
11  </body>
12 </html>
13

style.css x
1 #main{
2   position:relative;
3   bottom:15px;
4   top:55px;
5   right:15px;
6   left:15px;
7   background-color:blue;
8   width:500px;
9 }
10
```



POSITION : FIXED

- Le point de référence est le même qu'absolute
- Si on scroll l'élément ne se déplace pas



GRID

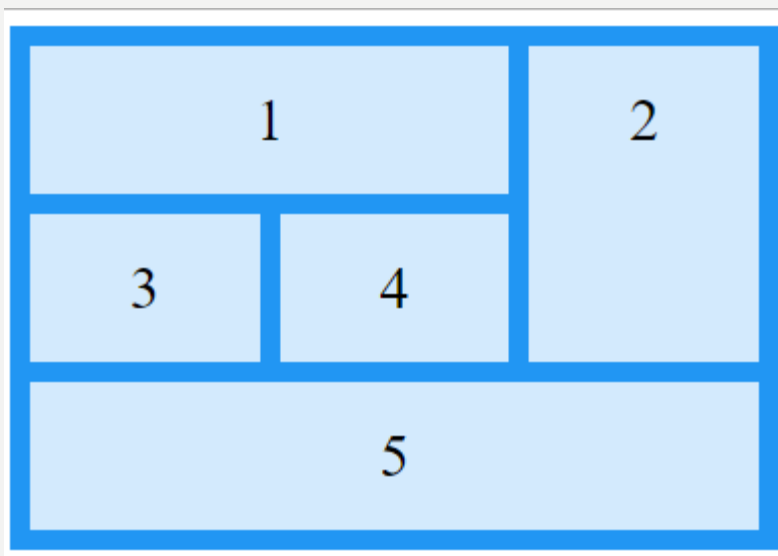
- La solution de facilité, d'une redoutable efficacité
- Out les float et les positions
- Un conteneur et des éléments (les enfants directs du conteneur)
- Pour le conteneur
 - display : grid;
 - grid-template-columns : auto auto auto; **OU** 50px 100px 50px auto;
 - Idem pour rows
 - grid-gap : 10px;
 - justify-content
 - align-content

GRID

- Les éléments
 - grid-column
 - grid-row
 - départ / fin
 - départ / étendu
 - grid-area : ligne départ / colonne départ / fin ou étendu de ligne / fin ou étendu de colonne

GRID : EXEMPLE

```
5 <body style="padding-left:100px;">
6   <div class="grid-container">
7     <div class="grid-item item1">1</div>
8     <div class="grid-item item2">2</div>
9     <div class="grid-item item3">3</div>
10    <div class="grid-item item4">4</div>
11    <div class="grid-item item5">5</div>
12  </div>
13 </body>
```



```
style.css
1 .grid-container {
2   display: grid;
3   grid-gap: 10px;
4   background-color: #2196F3;
5   padding: 10px;
6 }
7
8 .grid-item {
9   background-color: rgba(255, 255, 255, 0.8);
10  text-align: center;
11  padding: 20px;
12  font-size: 30px;
13 }
14
15 .item1 {
16   grid-column: 1 / span 2;
17   grid-row: 1;
18 }
19
20 .item2 {
21   grid-column: 3;
22   grid-row: 1 / span 2;
23 }
24
25 .item5 {
26   grid-column: 1 / span 3;
27   grid-row: 3;
28 }
```

CHAPITRE 5: LES AUTRES CONCEPTS

FONCTION ATTR

- attr(nomDeLAttribut)

```
8 <body>
9 <a href="page1.html" data-txt="vers la page1"><a>
10 <a href="page2.html" data-txt="vers la page2"></a>
11 </body>
```

```
style.css x
1 a:after{
2   content: attr(data-txt) ;
3 }
4
```

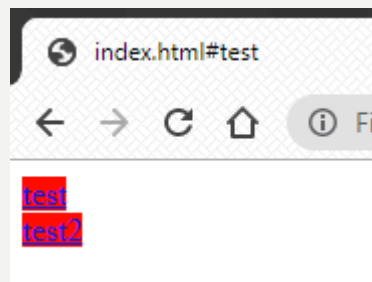


LES VARIABLES

- `var()`
- On définit une variable dans `root`
- On met deux tirets au début du nom de la variable

```
1  =:root {  
2      --main-color:red;  
3  }  
4  =a{  
5      background-color:var(--main-color);  
6  }  
7
```

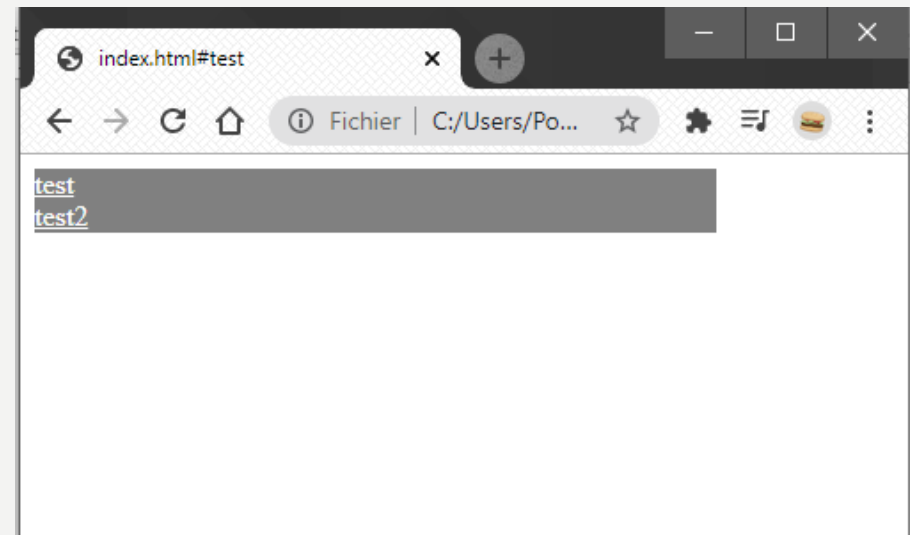
```
8  =      <body>  
9          <a href="page1.html">test</a><br/>  
10         <a href="page2.html">test2</a>  
11      </body>
```



FONCTION CALC

- Calc()
- Espace entre les opérateurs !!!!!
- On peut mélanger les unités
- Peut-être associés à var et attr

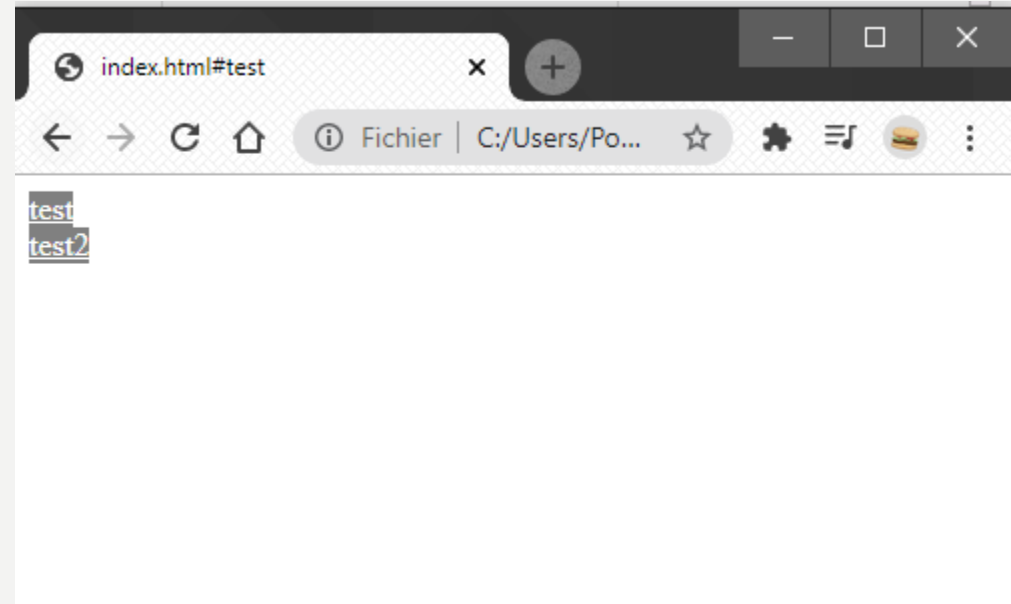
```
1  :root {  
2    --marge-droite:100px;  
3  }  
4  a{  
5    display:inline-block;  
6    width: calc(100% - var(--marge-droite));  
7    background-color:grey;  
8    color:white;  
9  }  
10  
11
```



FONCTION CALC

- Calc()
- Espace entre les opérateurs !!!!!
- On peut mélanger les unités
- Peut-être associés à var et attr

```
5 display: inline-block;  
6 width: calc(100%-var(--marge-droite));  
7 background-color: grey;
```



TRANSPARANCE

- La valeur/le canal alpha : la transparence appliquée à la couleur
 - La fonction RGBa
 - Une valeur entre 0 et 1
 - `rgba(85,85,85,0.5)` → à moitié transparent
 - En hexademical : on rajoute une valeur
 - `#555555FF` → Opaque
 - `#55555500` → Invisible

TRANSPARENCE

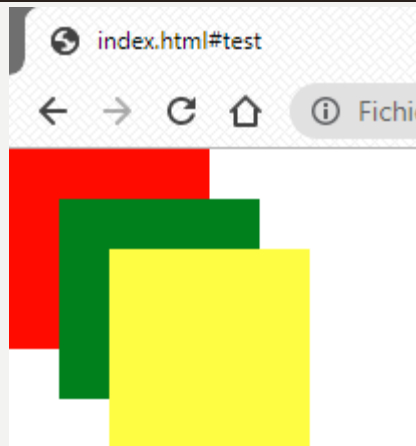
- L'attribut `opacity` : la transparence appliquée à un ensemble d'éléments
- Cette propriété est héritable et n'est pas réversible sur les enfants



Z-INDEX

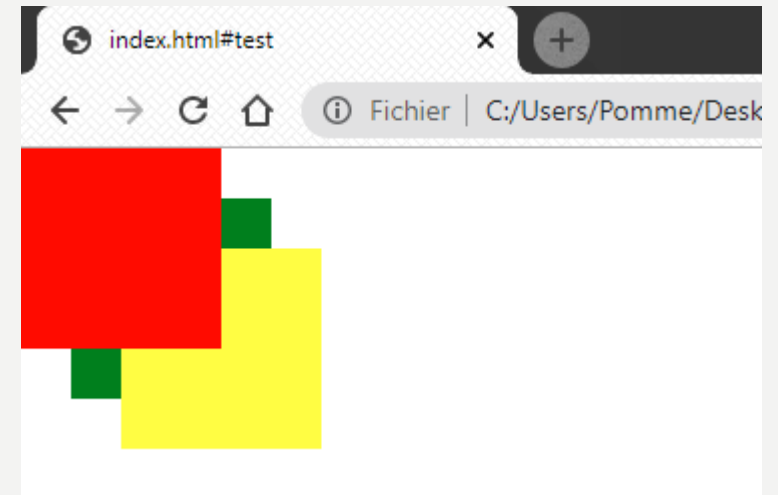
```
index.html x test.html x
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5     </style>
6     <link rel="stylesheet" href="css/style.css" />
7   </head>
8   <body>
9     <div id="div1"></div>
10    <div id="div2"></div>
11    <div id="div3"></div>
12  </body>
13 </html>
14

style.css x
1 div{
2   width:100px;
3   height:100px;
4   position:absolute;
5 }
6 #div1{
7   top:0px;
8   left:0px;
9   background-color:red;
10 }
11
12 #div2{
13   top:25px;
14   left:25px;
15   background-color:green;
16 }
17 #div3{
18   top:50px;
19   left:50px;
20   background-color:yellow;
21 }
```



Z-INDEX

```
style.css
1  =div{
2      width:100px;
3      height:100px;
4      position:absolute;
5  }
6  =#div1{
7      top:0px;
8      left:0px;
9      background-color:red;
10     z-index:999;
11 }
12 =#div2{
13     top:25px;
14     left:25px;
15     background-color:green;
16 }
17 =#div3{
18     top:50px;
19     left:50px;
20     background-color:yellow;
21 }
22
```



BORDURE

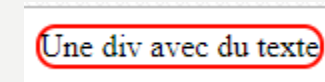
- Une taille : border-width
- Une couleur : border-color
- Un style : border-style

```
1 div{  
2   display:inline-block;  
3   border-style: dotted;  
4   border-width: 2px;  
5   border-color: red;  
6 }
```



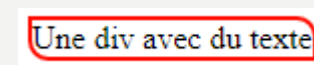
- Arrondir les bords : border-radius

```
6 border-radius : 10px;
```



- Arrondir certains bords (sens des aiguilles d'une montre en partant d'en haut à gauche)

```
6 border-radius : 0px 10px 0px 10px;
```



LONG HAND, SHORT HAND

- Notation raccourcis, on rassemble plusieurs propriétés sous la forme d'une seule
- Exemple :

```
1  #p1{  
2    border-width: 2px;  
3    border-style: solid;  
4    border-color: red;  
5  }  
6
```

```
7  #p1{  
8    border: 2px solid red;  
9  }
```

- Limite : on ne peut pas utiliser les valeurs inherit, initial et unset

LONG HAND, SHORT HAND

Short Hand	Equivalent Long Hand
font	font-style, font-variant, font-weight, font-size, line-height, font-family
border	border-width, border-style, border-color
margin	margin-top, margin-right, margin-bottom, margin-left
padding	padding-top, padding-right, padding-bottom, padding-left
background	background-image, background-position, background-size, background-repeat, background-origin, background-clip, background-attachment, background-color
transition	transition-property, transition-duration, transition-timing-function, transition-delay
animation	animation-name, animation-duration, animation-timing-function, animation-delay, animation-iteration-count, animation-direction, animation-fill-mode, animation-play-state
flex	flex-shrink, flex-grow, flex-basis

BACKGROUND

- *background-image* : spécifie l'emplacement de l'image à mettre en fond
- *background-attachment* : spécifie si le background va bouger quand on scroll
- *background-blend-mode* : très étrange
- *background-clip* : l'étendu du background dans la boite (border, padding, content)
- *background-color* : trivial !
- *background-origin* : point d'origine du background dans la boite
- *background-size* : la taille de l'image
- *background-position* : la position de départ du background
- *background-repeat* : la manière dont le background va se répéter ou non

BACKGROUND : POSITION

- Mots clefs
 - horizontal
 - *left*
 - *right*
 - *center*
 - vertical
 - *top*
 - *bottom*
 - *center*
 - Associé une valeur de chaque
- Pourcentage
- Pixel

BACKGROUND : SIZE

- Mots clefs
 - *Auto* : taille originale de l'image
 - *cover* : remplir le contenant
 - *contain* : l'image est agrandie au maximum mais doit rester entièrement visible
- Largeur/Hauteur
 - *Pourcentage*
 - *Pixel*
- Attention à la dimension native de l'image

DEGRADÉS

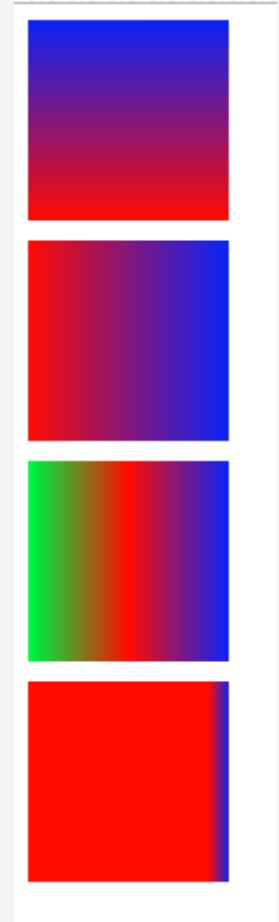
- Linéaires
- Répétitifs
- Radiaux
- Multi-couleurs
- Transparence
- Positionnement

DEGRADÉS : LINEAIRE

- background-image
- linear-gradient(blue, red);
- linear-gradient(to left, blue, red);
- linear-gradient(to left, blue, #F00, rgb(0, 255, 0));
- linear-gradient(to left, blue 0%, red 10%);
- Direction, couleur I position I, ..., couleurN, positionN)
- Direction Optionnel
- Position Optionnel

```
<body class="body">
  <div></div>
  <div></div>
  <div></div>
  <div></div>
</body>
```

```
1  =div{
2    display:block;
3    width:100px;
4    height:100px;
5    margin-bottom:10px;
6  }
7  div:nth-of-type(1)
8  = {
9    background-image:linear-gradient(blue, red);
10 }
11 div:nth-of-type(2)
12 = {
13   background-image:linear-gradient(to left, blue, red);
14 }
15 div:nth-of-type(3)
16 = {
17   background-image:linear-gradient(to left, blue, #F00, rgb(0, 255, 0));
18 }
19 div:nth-of-type(4)
20 = {
21   background-image:linear-gradient(to left, blue 0%, red 10%);
22 }
23 }
```

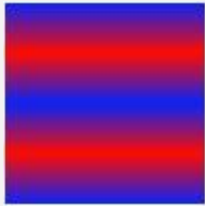


DEGRADÉS : RADIAL

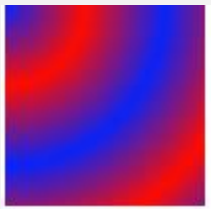
- closest-corner : vers le coin le plus proche
- closest-side : vers le côté le plus proche
- farthest-corner : vers le coin le plus éloigné
- farthest-side : vers le côté le plus éloigné
- Circle/Ellipse

DEGRADÉS : REPETITIF

- Rajouter repeating et donner la dimension
- `repeating-linear-gradient(blue, red, blue 100px);`



- `repeating-radial-gradient(circle at top left, blue, red, blue 80px);`



TRANSFORMATION

- 2D : 2 axes
- 3D (Pour les stars) : 3 axes
- *transform* : le type de transformation
- *transform-origin* : le point d'origine de la transformation

TRANSFORMATION : 2D

- Rotation : *rotate()*
 - L'origine
- Changement d'échelle : *scale(xy)*, *scaleX()*, *scaleY()*, *scale(x,y)*
 - En proportion
- Translation : *translate()*, *translateX()*, *translateY()*
 - En px
- Oblique : *skew()*, *skewX()*, *skewY()*
 - En degrés
- Le MUST, la matrice !! : *matrix(...)*

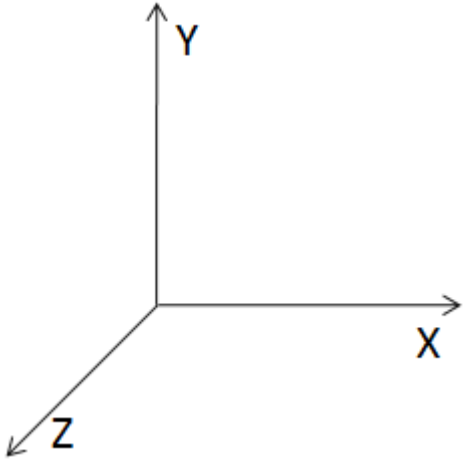
TRANSFORMATION : 3D

- La perspective : *perspective*
- Il n'y a plus *skew()*
- On rajoute "3d" : *translate3d()*, *rotate3D()*, *scale3d()*, *matrix3d()*
- Apparition de la coordonnée Z
- *backface-visibility*

```
<html>
  <head>
    <link rel="stylesheet" href="style.css"/>
  </head>
  <body>
    <div>
      Basic
    </div>

    <div id="myDiv">
      rotated
    </div>
  </body>
</html>
```

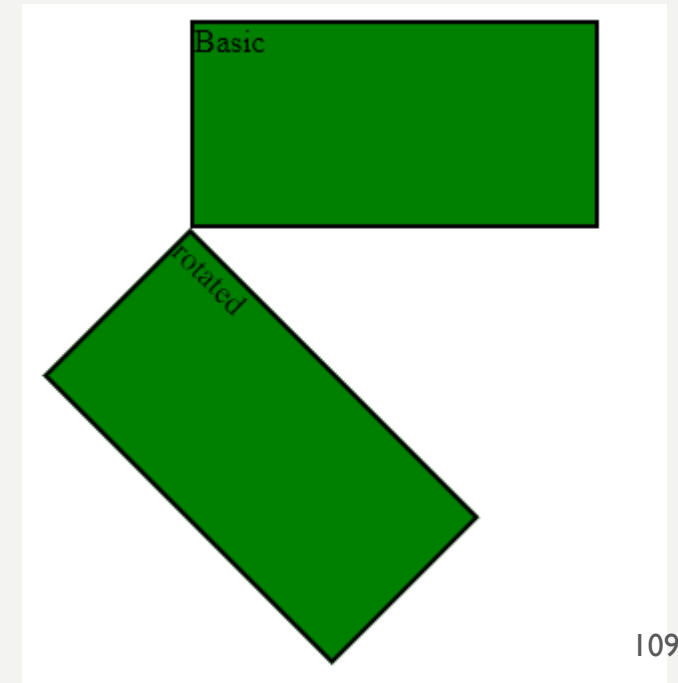
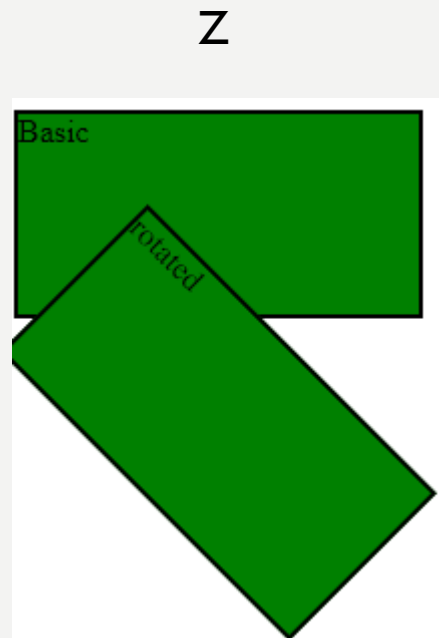
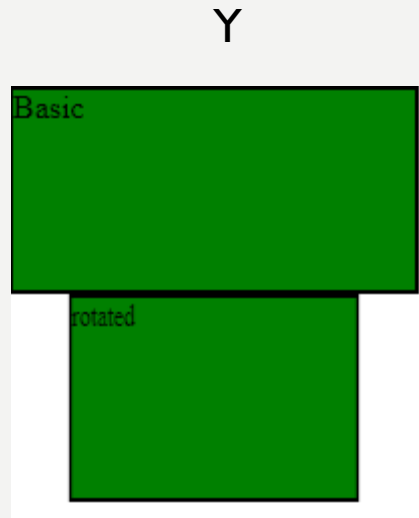
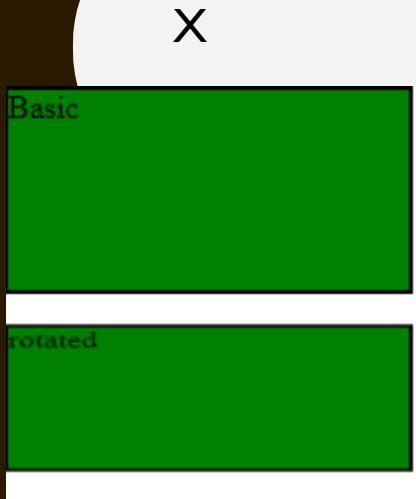
TRANSFORMATION : 3D EXEMPLE ROTATE



```
<html>
  <head>
    <link rel="stylesheet" href="style.css"/>
  </head>
  <body>
    <div>
      Basic
    </div>

    <div id="myDiv">
      rotated
    </div>
  </body>
</html>
```

Z avec
transform-origin : 0 0;

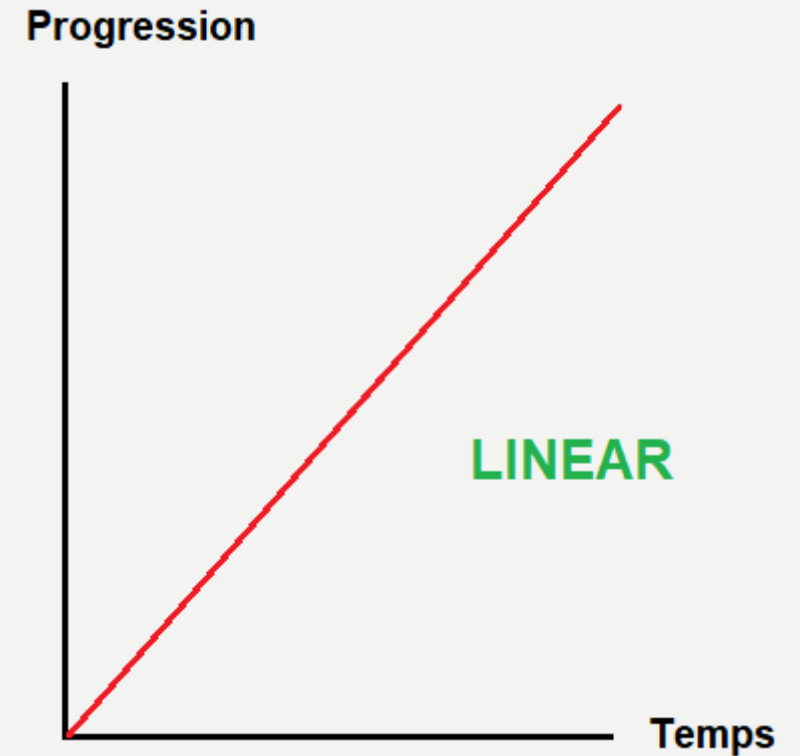


TRANSITION

- `transition-property`
 - La/les propriétés à faire évoluer
- *transition-duration*
 - Pensez aux fps (~60 pour un navigateur et ENCORE !)
- *transition-delay*
 - Retarder le départ

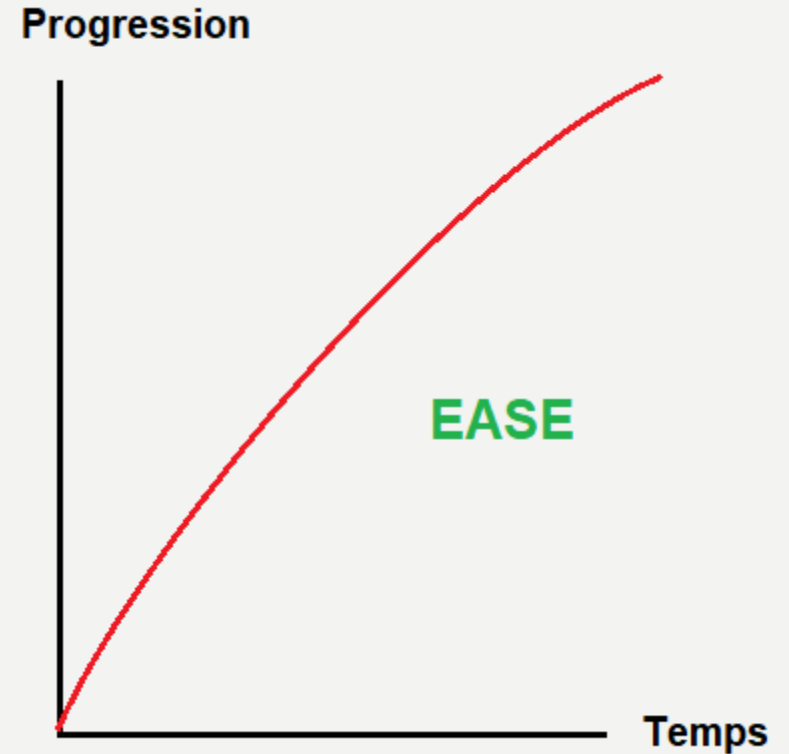
TRANSITION : TIMING FUNCTION

- *transition-timing-function*
 - ease
 - *linear*
 - ease-in
 - ease-out
 - ease-in-out



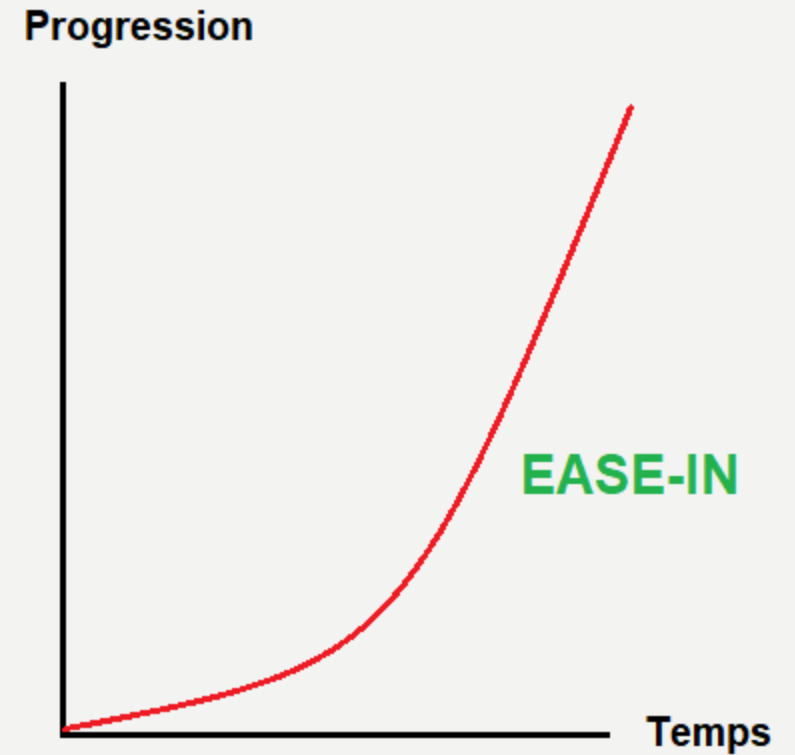
TRANSITION : TIMING FUNCTION

- *Transition-timing-function*
 - *ease*
 - *linear*
 - *ease-in*
 - *ease-out*
 - *ease-in-out*



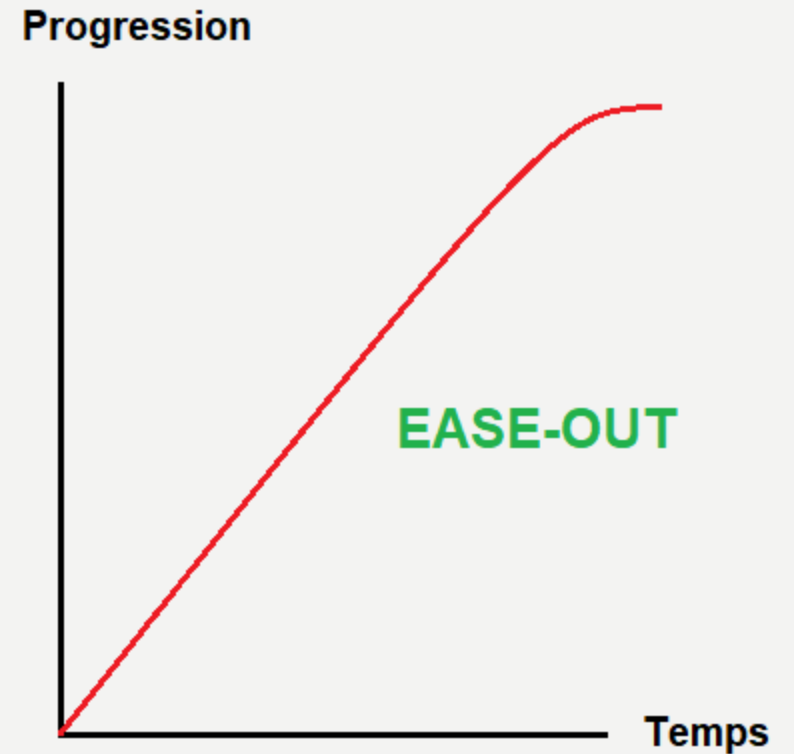
TRANSITION : TIMING FUNCTION

- *Transition-timing-function*
 - *ease*
 - *linear*
 - *ease-in*
 - *ease-out*
 - *ease-in-out*



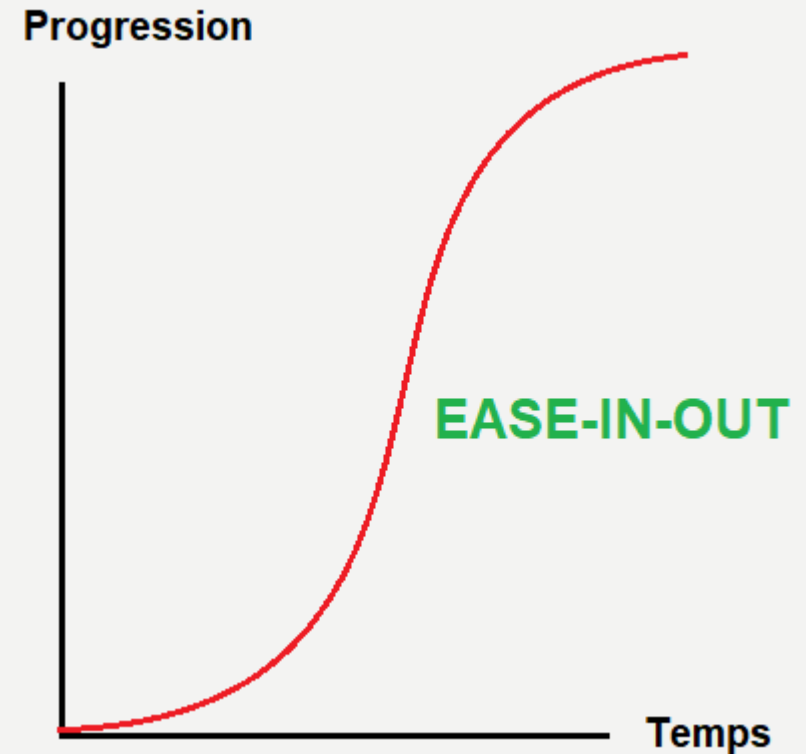
TRANSITION : TIMING FUNCTION

- Transition-timing-function
 - ease
 - linear
 - ease-in
 - **ease-out**
 - ease-in-out



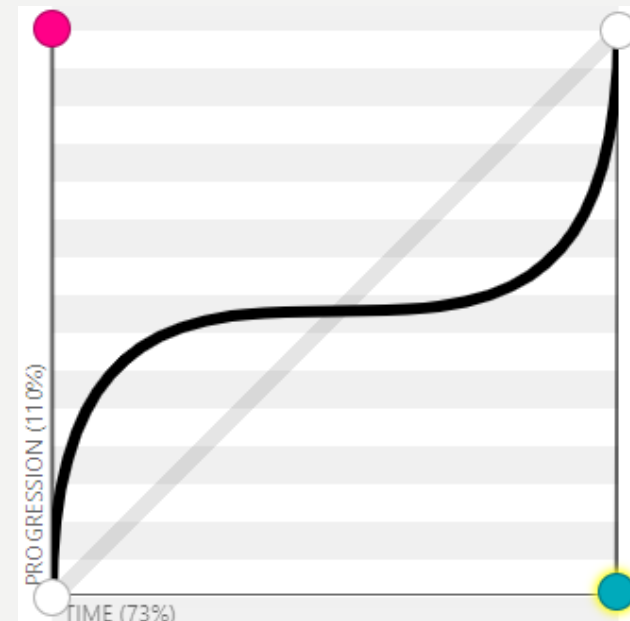
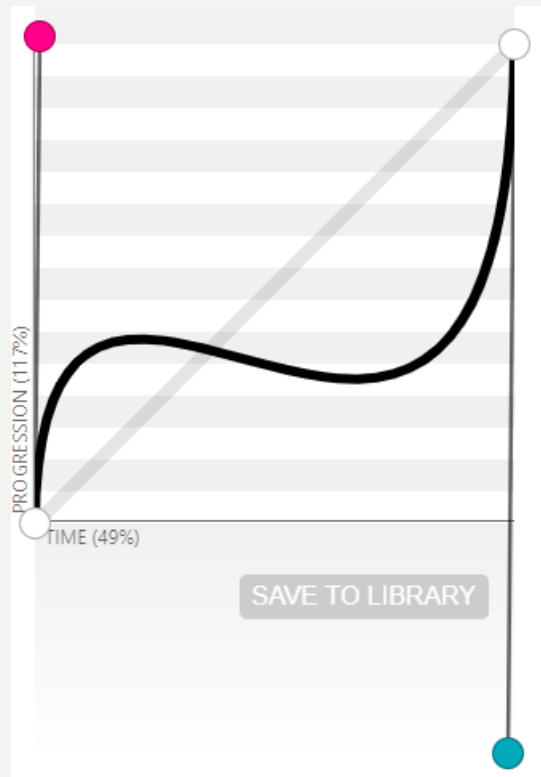
TRANSITION : TIMING FUNCTION

- *Transition-timing-function*
 - *ease*
 - *linear*
 - *ease-in*
 - *ease-out*
 - *ease-in-out*



TRANSITION : CUBIC BEZIER

- Cubic-bezier : <https://cubic-bezier.com/#.59,1.41,.33,-0.38>



ANIMATION

- `@keyframes`
 - Déclare une animation
 - Myanim est le nom de l'animation
- *Animation*
 - La propriété short hand permettant d'appeler une animation déclarée avec certains paramètres
- On définit des étapes

```
style.css x test2.html x test3.html x
1  @keyframes myanim {
2      from {
3          transform: translateX(0px);
4      }
5
6      30% {
7          transform: translateX(150px);
8      }
9
10     60% {
11         transform: translateX(0px);
12     }
13
14     80% {
15         transform: translateX(150px) ;
16     }
17
18     to {
19         transform: translate(75px) rotate(360deg);
20         background-color: red;
21     }
22 }
23
24
25 div
26 {
27     margin: 100px;
28     width: 200px;
29     height: 150px;
30     background-color: #008;
31 }
32 div:hover
33 {
34     animation: myanim 4s steps(100);
35 }
```

ANIMATION

- *animation-name*
 - Le nom de l'animation à déclencher
- *animation-duration*
 - La durée de l'animation
- *animation-delay*
 - Le délai de départ de l'animation
- *animation-timing-function*
 - Voir les transitions !
- *animation-iteration-count*
 - Nombre de répétition de la l'animation
- *animation-direction*
 - *alternate* pour faire revenir l'animation dans l'autre sens
- *animation-fill-mode*
 - *forwards* ou *backwards*
- *animation-play-state*

CYCLE DE VIE D'UNE PAGE

- Comprendre le cycle de vie d'un page
- Permet d'améliorer les performances des animations
- Qu'est-ce qu'un cycle de vie ?
- Quel est le cycle de vie graphique d'une page html ?
 - Style : association des règles aux structures
 - Layout : tailles et positions
 - Paint : transformation du résultat de l'étape 2 en pixels
 - Composition : Combinaison de toutes les étapes précédentes pour déterminer ce qu'il faut afficher à l'écran
- Quand on reprendre une étape, il faut re-dérouler celles qui suivents

CYCLE DE VIE D'UNE PAGE

- Chaque étape nécessite un temps de calcul
- La propriété width par exemple va entraîner une modification de l'étape2
- La propriété transform va entraîner une modification de l'étape composition
- csstriggers.com : pour connaître la correspondance propriété/étape

UNITES ABSOLUES

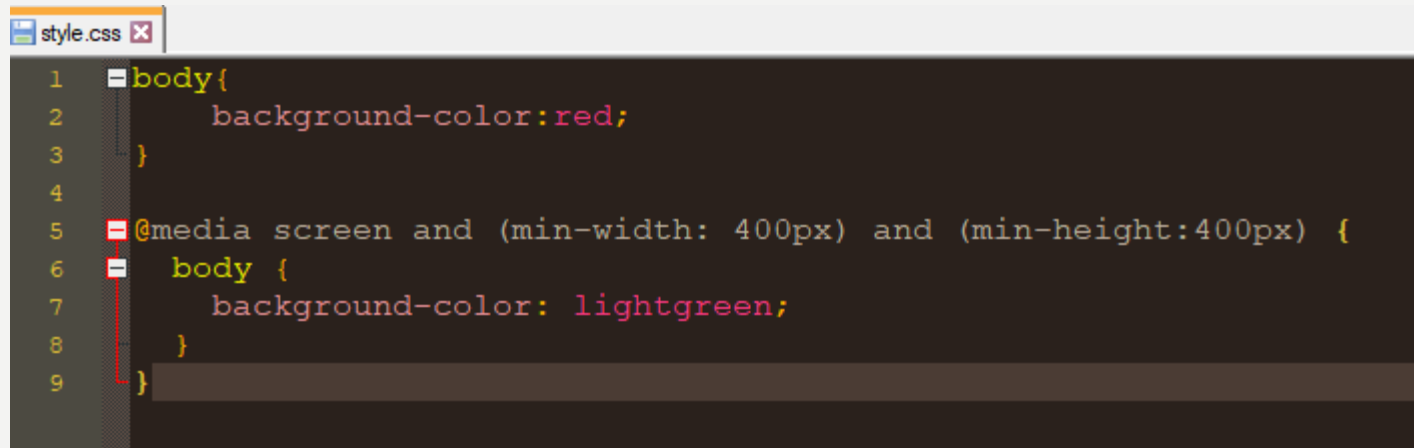
- Millimètres (mm)
- Centimètres (cm) = 10 mm
- Pouces (in) = 2,54 cm
- Pixels (px) = 1/96 in
- Points (pt) = 1/72 in
- Picas (pc) = 12pt

UNITES RELATIVES

- em : relatif à la font-size en cours
- vw : 1% de la largeur de la vue
- vh : 1% de la hauteur de la vue
- vmin : 1% de la plus petite dimension de la vue
- vmax : 1% de la plus grande dimension de la vue
- % : relatif à l'élément parent
- ex, ch, rem : Voir la documentation...

RESPONSIVE

- Permet de spécifier un style en fonction de la taille de l'écran
- Attribut media dans link
- Syntaxe : `@media not|only mediatype and (mediafeature and|or|not mediafeature)`
- Toutes les valeurs de *mediatype* et *mediafeature* sont dans la documentation

A screenshot of a code editor window titled 'style.css'. The editor has a dark background with light-colored text. It shows two CSS rules. The first rule is on line 1: 'body{' followed by 'background-color:red;' on line 2, and a closing brace on line 3. The second rule is a media query starting on line 5: '@media screen and (min-width: 400px) and (min-height:400px) {' followed by 'body {' on line 6, 'background-color: lightgreen;' on line 7, a closing brace for the inner rule on line 8, and a closing brace for the media query on line 9. Line numbers 1 through 9 are visible on the left side of the editor.

```
1 body{
2     background-color:red;
3 }
4
5 @media screen and (min-width: 400px) and (min-height:400px) {
6     body {
7         background-color: lightgreen;
8     }
9 }
```

RESPONSIVE

- Chrome : CTRL+SHIFT+J, puis CTRL+SHIFT+M, permet d'émuler un device
- Les téléphones mobiles ont une batterie et il est important d'optimiser votre site dans ce sens

FLEX

- L'avenir ?
- Le passé : *float* ou même encore des tableaux ?
- Ce qu'il nous faut pour faire du flex
 - Un conteneur
 - Des éléments
- Le conteneur
 - *display: flex;*

FLEX

- Grid flottant ?
- Le conteneur
 - *display : flex;*
 - *flex-direction : column / row (reverse)*
 - *flex-wrap: wrap* ⇔ *float:left*
 - *justify-content*
 - *align-items* : sur l'axe secondaire
 - *align-content*

FLEX

- Les éléments
 - *flex-basis* : pondération de base
 - *flex-grow* : facteur de grossissement
 - *flex-shrink* : facteur de rétrécissement
 - *flex* : short hand

```
1  #container
2  {
3      display: flex;
4      width: 500px;
5      height: 100px;
6  }
7  #flex1{
8      background-color: red;
9      flex: 2;
10 }
11 #flex2{
12     background-color: green;
13     flex: 1;
14 }
15 #flex3{
16     background-color: blue;
17     flex: 1;
18 }
```

```
<body>
  <div id="container">
    <div id="flex1"></div>
    <div id="flex2"></div>
    <div id="flex3"></div>
  </div>
</body>
```



FLEX

- Grid flottant ?
- Le conteneur
 - `display : flex;`
 - `flex-direction : column / row (reverse)`
 - `flex-wrap: wrap` ⇔ `float:left`
 - `justify-content`
 - `align-items` : sur l'axe secondaire
 - `align-content`



```
<body>
  <div id="container">
    <div id="flex1"></div>
    <div id="flex2"></div>
    <div id="flex3"></div>
  </div>
</body>
```

```
style.css
1  #container
2  {
3      display: flex;
4      width: 500px;
5      height: 100px;
6      flex-direction: column;
7  }
8  #flex1{
9      background-color: red;
10     flex: 2;
11 }
12 #flex2{
13     background-color: green;
14     flex: 1;
15 }
16 #flex3{
17     background-color: blue;
18     flex: 1;
19 }
```


FLEX

- Les éléments
 - *flex-basis* : taille de base
 - *flex-grow* : facteur de grossissement
 - *flex-shrink* : facteur de rétrécissement
 - *flex* : short hand



```
<body>
  <div id="container">
    <div id="flex1"></div>
    <div id="flex2"></div>
    <div id="flex3"></div>
  </div>
</body>
```

```
1  #container
2  {
3      display: flex;
4      width: 500px;
5      height: 100px;
6  }
7  #flex1{
8      background-color: red;
9      flex: 2;
10 }
11 #flex2{
12     background-color: green;
13     flex: 1;
14 }
15 #flex3{
16     background-color: blue;
17     flex: 1;
18 }
```

LISTE

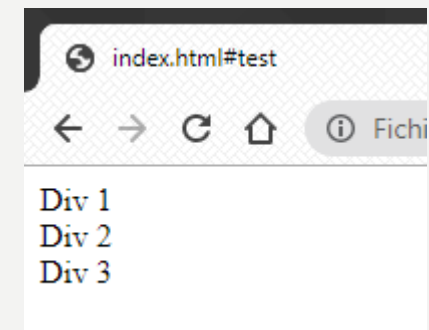
- List-style-position
 - La puce des items se trouvent en ou en dehors de la liste
- List-style-type : le type de la puce (Il y en a beaucoup)
 - https://www.w3schools.com/cssref/pr_list-style-type.asp
- List-style-image
 - Remplacer la puce par une image personnel
- List-style
 - Ecriture short-hand : type position image

COUNTER

```
index.html x test.html x
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5     </style>
6     <link rel="stylesheet" href="css/style.css" />
7   </head>
8   <body>
9     <div id="div1"></div>
10    <div id="div1"></div>
11    <div id="div1"></div>
12  </body>
13 </html>
14

style.css x
1 :root{
2   counter-reset:compteur;
3 }
4 div:after{
5   counter-increment: compteur;
6   content: "Div " counter(compteur);
7 }
```

- *counter-reset* : création d'un compteur (démarré à 0)
- *counter-increment* : +1 au compteur
- *counter(unCompteur)* : récupération de la valeur du compteur



VALIDATION ET TOUT FAIT

- <http://jigsaw.w3.org/css-validator/>
- <https://www.w3schools.com/howto/default.asp>

ALLER PLUS LOIN

- Le CSS semi compilé : SASS
 - Améliore l'efficacité
 - Variable
 - Fonctions mathématiques
 - Boucles
 - etc.
 - A la fin on ne peut pas faire mieux que css
- Bootstrap ou W3.CSS
 - Des collections css