

R102

DÉVELOPPEMENT D'INTERFACES WEB

Jean Carle jean.carle@univ-lille.fr

CSS VALEURS LOGIQUES VS PHYSIQUES

Définition

Les **propriétés logiques** sont des propriétés dont l'orientation est relative au mode d'écriture du document (défini par les propriétés `writing-mode`, `direction` et `text-orientation`)

À partir de là, on définit les deux directions suivantes :

- La direction **bloc** (`block`) : direction orthogonale au sens de lecture
- La direction **en ligne** (`inline`) : direction du sens de lecture

VALEURS LOGIQUES : EXEMPLES EN MODE PAR DÉFAUT (HORIZONTAL-TB)

Dans le nom des propriétés logiques, on pourra aussi trouver :

- start pour signifier le début de la ligne
- end pour signifier la fin de la ligne
- block-size / inline-size = height / width
- margin-block-start / padding-block-end : margin-top / padding-bottom
- margin-inline : Combinaison de margin-left et de margin-right
- padding-block : Combinaison de padding-top et de padding-bottom
- border-inline-start-color : border-left-color
- border-start-start-radius : border-top-left-radius

CSS : VALEURS UNITÉS

Pour détailler un peu plus :

- Spécification : <https://www.w3.org/TR/css-values-4>
- https://developer.mozilla.org/fr/docs/Learn/CSS/Building_blocks/Values_and_units#quelle_valeur_css
- Résumé des types de valeurs : https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Values_and_Units/CSS_data_types

VALEURS

- mots-clés : *red, bigger, hidden...*
- fonctions : `url()`, `rgb()`, `calc()`, `var()`, `counter()`...
- Valeurs numériques : entiers, décimaux avec ou sans unité
 - longueur,
 - fréquences audio,
 - angle



Possibilité de pourcentage relativement à la valeur de la propriété correspondante dans l'élément parent.

LONGUEURS ABSOLUES

Unité	Nom	Équivalence
cm	Centimètre	$1\text{cm} = 96\text{px}/2.54$
mm	Millimètre	$1\text{mm} = 1/10\text{e de } 1\text{cm}$
Q	Quart de millimètre	$1\text{Q} = 1/40\text{e de } 1\text{cm}$
in	Pouces (inches)	$1\text{in} = 2.54\text{cm} = 96\text{px}$
pc	Picas	$1\text{pc} = 1/16\text{e de } 1\text{in}$
pt	Points	$1\text{pt} = 1/72\text{e de } 1\text{in}$
px	Pixels	$1\text{px} = 1/96\text{e de } 1\text{in}$

LONGUEURS RELATIVES

À la police de caractères

Unité	Relative à
em	La taille (corps) de police de l'élément (ou son parent si non défini)
rem	La taille (corps) de police de l'élément racine
ex	La hauteur d'un x avec la police utilisée par l'élément
cap	La hauteur d'une majuscule nominale avec la police utilisée par l'élément
lh	La hauteur de la ligne de l'élément
rlh	La hauteur de la ligne de l'élément racine

LONGUEURS RELATIVES 2

À la zone d'affichage (viewport)

Unité	Relative à
vw	1% de la largeur de la zone d'affichage
vh	1% de la hauteur de la zone d'affichage
vi	1% de la taille de la zone d'affichage sur l'axe horizontal (inline axis)
vb	1% de la taille de la zone d'affichage sur l'axe vertical (block axis)
vmin	1% de la zone d'affichage selon sa plus petite dimension
vmax	1% de la zone d'affichage selon sa plus grande dimension

CSS : FUSION DES MARGES

La fusion des marges ne concernent **QUE** :

- les marges verticales
- les éléments de type **bloc** (pas les éléments de type **inline**)

De plus :

- Les marges de l'axe *horizontale* ne fusionnent jamais
- Les marges de l'axe *verticales* fusionnent si rien ne les en empêche :
- Pour les boites adjacentes : la marge sera la plus grande des deux
- Pour le parent : fusion avec la première et dernière boite enfant
- Pour une boite vide : fusion entre ses marges.

Pour plus de détail :

https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Box_Model/Mastering_margin_collapsing

BOITES VOISINES

Les marges fusionnent

CSS

```
p.un {  
    border: 1px solid red;  
    margin-bottom: 10px;  
}  
p.deux {  
    margin-top: 30px;  
    border: 1px dashed red;  
}
```

html

```
<p class="un">La marge basse de ce paragraphe est fusionnée  
<p class="deux">... avec la marge haute de celui-ci.</p>
```

Résultat

La marge basse de ce paragraphe est fusionnée...

... avec la marge haute de celui-ci.

FUSION ENFANTS-PARENTS

CSS

```
.un {margin: 10px 0; background-color: blue;}  
.deux {margin: 10px 0; background-color: lightgreen}  
.trois {margin: 50px 0; background-color: violet;}  
/*.un, .deux, .trois {border: 2px solid;}*/
```

html

```
<div class="un">  
  <div class="deux">  
    <p class="trois">Trois : Un paragraphe</p>  
    Deux : Parent du paragraphe  
  </div>  
  Un : Grand-parent du paragraphe  
</div>
```

Résultat

Trois : Un paragraphe

Deux : Parent du paragraphe

Un : Grand-parent du paragraphe

Résultat avec border activée

Trois : Un paragraphe

Deux : Parent du paragraphe

Un : Grand-parent du paragraphe

DES COMPTEURS EN CSS

html

```
<h4>Ceci n'est pas une liste</h4>
<p>premier paragraphe</p>
<p>second paragraphe</p>
<p>3e paragraphe</p>
<p>4e paragraphe</p>
```

Résultat

CECI N'EST PAS UNE LISTE

premier paragraphe
second paragraphe
3e paragraphe
4e paragraphe

DES COMPTEURS EN CSS

html

```
<h4>Ceci va devenir une liste</h4>
<p>Un premier item</p>
<p>Un second item</p>
<p>Un 3e item</p>
<p>Un 4e item</p>
```

Résultat

CECI VA DEVENIR UNE LISTE

02) Un premier item - II

04) Un second item - IV

06) Un 3e item - VI

08) Un 4e item - VIII

Ajoutons quelques règles CSS

```
body {
    counter-reset: cpt1;
}
p:before {
    counter-increment: cpt1 2;
    content: counter(cpt1, decimal-leading-zero) " ";
}

p:after {
    content: " - " counter(cpt1, upper-roman) ;
}
```

COMPTEURS : STYLES PERSONNALISÉS

On peut changer les valeurs utilisées avec une règle At :

@counter-style

html

```
<h4>Ceci n'est pas une liste (en html)</h4>
<p>Un premier item</p>
<p>Un second item</p>
<p>Un 3e item</p>
<p>Un 4e item</p>
<p>Un 5e item</p>
```

CECI N'EST PAS UNE LISTE (EN HTML)

1.  Un premier item
2.  Un second item
3.  Un 3e item
4.  Un 4e item
5.  Un 5e item

CSS

```
@counter-style ma-liste {
    system: cyclic;
    symbols: 🙌 😬 💋 ;
    suffix: " ";
}

body {
    counter-reset: cpt1;
}
p:before {
    counter-increment: cpt1;
    content: counter(cpt1) ". " counter(cpt1, ma-liste)
}
```

DES VARIABLES EN CSS

Propriétés personnalisées

Commencent par `--` (double tiret)

```
:root { 1  
  --main-bg-color: brown; 2  
}  
.un {  
  background-color: var(--main-bg-color); 3  
  width: 50px;  
  height: 50px;  
}  
.deux {  
  background-color: var(--main-bg-color); 3  
  margin: 10px;  
  display: inline-block;  
}
```

- 1 Variable globale dans root (locale sinon)
- 2 Définition des variables
- 3 Utilisation (avec la fonction `var()`)

EXEMPLE D'UTILISATION

Variable locale avec héritage pour définition sur éléments enfants

html

```
<article class="cat-sport">
  <h2>Sport</h2>
  <blockquote>
    Lorem ipsum dolor sit amet consectetur ...
  </blockquote>
  <button class="btn">Bouton</button>
</article>

<article class="cat-programming">
  <h2>Programmation</h2>
  <blockquote>
    Sunt cillum consectetur ut quis est fugiat
  </blockquote>
  <button class="btn">Bouton</button>
</article>
```

CSS

```
.cat-sport {
  --theme-color: #4A772F;
}
.cat-programming {
  --theme-color: #005E7C;
}
.cat-culture {
  --theme-color: #F5841A;
}
h2 {
  color: var(--theme-color);
  font-variant-caps: small-caps;
}
blockquote {
  border-left: 3px solid var(--theme-color);
```

EXEMPLE D'UTILISATION

Variable locale avec héritage pour définition sur éléments enfants

Résultat

SPORT

Lorem ipsum dolor sit amet consectetur ...

SPORT

PROGRAMMATION

Sunt cillum consectetur ut quis est fugiat ...

PROGRAMMATION

CULTURE

Labore laborum elit eu excepteur sunt eiusmod ...

CULTURE

C'EST TOUT POUR CETTE FOIS

