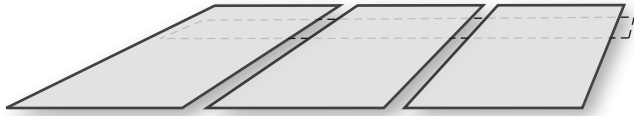


CSS: FLOAT

Float (flottement des blocs)

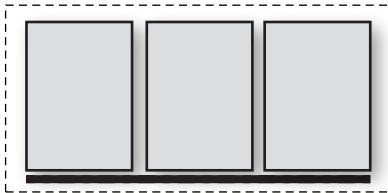


```
HTML
<div class="contenant">
  <div class="float"></div>
  <div class="float"></div>
  <div class="float"></div>
</div>
```

```
CSS
div.contenant{
  width:300px;
}
div.float{
  float:left;
  width:100px;
}
```

Supprimer (clear)l'effet de flottement

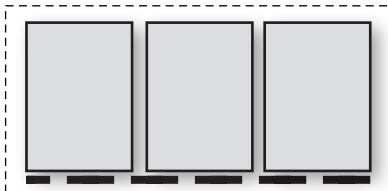
Option 1 : En créant un objet bloc en HTML disposant de la propriété CSS «clear»



```
HTML
<div class="contenant">
  <div class="float"></div>
  <div class="float"></div>
  <div class="float"></div>
  <div class="clear"></div>
</div>
```

```
CSS
div.clear{
  clear:both;
}
```

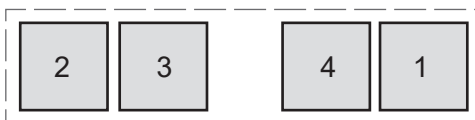
Option 2 : Utiliser le pseudo-élément ::after pour générer un espace à l'extrémité du contenant disposant la propriété «clear» depuis le CSS



```
HTML
<div class="contenant clearfix">
  <div class="float"></div>
  <div class="float"></div>
  <div class="float"></div>
</div>
```

```
CSS
.clearfix::after{
  content:" ";
  display:block;
  clear:both;
}
```

Priorité aux extrémités



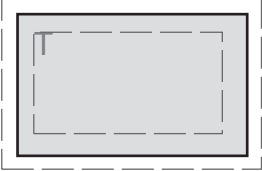
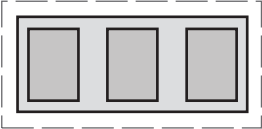
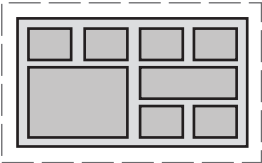
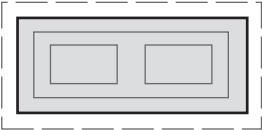
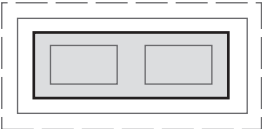
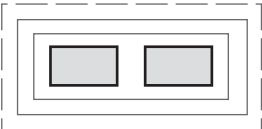
Combiner les blocs d'un même contenant pour qu'ils soient alignés à la même hauteur, les blocs en float:right doivent être indiqués en premier dans le HTML.

```
HTML
<div class="contenant">
  <div class="right"></div>
  <div class="left"></div>
  <div class="left"></div>
  <div class="right"></div>
</div>
```

```
CSS
div.right{
  float:right;
  width:65px;
}
div.left{
  float:left;
  width:65px;
}
```

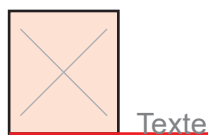
CSS: DISPLAY

Mode d'affichage (display) des objets

Mode d'affichage	Balises (mode par défaut)	Informations	Exemple
<code>display: block</code>	<code><div></code> , <code><p></code> , <code><h1></code> , <code><h2></code> , <code><h3></code> , <code><h4></code> , <code><h5></code> , <code><h6></code> , <code></code> , <code></code> , <code><dd></code> , <code><dl></code> , <code><dt></code> , <code><form></code> , <code><legend></code> , <code><fieldset></code> , <code><pre></code> , <code><code></code> , <code><address></code> , <code><blockquote></code> .	Définit un affichage bloc qui met à disposition des propriétés complémentaires pour la gestion de la forme (border) et la dimension de l'objet (width, height, padding & margin). La largeur par défaut est «auto» et occupe la pleine largeur du contenant.	
<code>display: inline</code>	<code></code> , <code><a></code> , <code></code> , <code></code> , <code><sup></code> , <code><sub></code> , <code><big></code> , <code><small></code> , <code></code> .	Organise le contenu selon une ligne de base établie selon la taille du caractère (font-size) et l'interlignage (line-height). Alignement sur la ligne de base établi avec vertical-align.	Texte courant d'une page Web
<code>display: list-item</code>	<code></code> .	Affichage s'apparentant à celui du bloc mais avec comme disposition la gestion des éléments des listes.	<ul style="list-style-type: none">• Element de la liste• Element de la liste• Element de la liste
<code>display: inline-block</code>	<code><input></code> .	Combine les propriétés prévues pour un affichage bloc et inline. Dimension de la ligne de base établie selon la hauteur de blocs. Alignement sur la ligne de base établi avec vertical-align.	Texte <input type="text"/>
<code>display: flex</code>		Gère l'alignement et la disposition de blocs contenus. Dispose de propriétés complémentaires d'alignement pour le contenant et de disposition pour les contenus.	
<code>display: grid</code>		Etablir une mise en forme de blocs contenus dans une grille. Dispose de propriétés complémentaires d'alignement pour le contenant et de disposition pour les contenus.	
<code>display: none</code>	<code><head></code> , <code><meta></code> , <code><title></code> , <code><link></code> , <code><style></code> , <code><script></code> , <code><input type="hidden"></code> .	invisible.	
<code>display: table</code>	<code><table></code> .		
<code>display: table-row</code>	<code><tr></code> .		
<code>display: table-cell</code>	<code><th></code> , <code><td></code> .	Alignement avec vertical-align.	

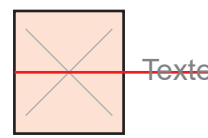
D'autres modes d'affichage spécifiques aux tableaux existent.

`display: inline-table`, `table-row-group`, `table-column`, `table-column-group`, `table-header-group`, `table-footer-group`, `table-cell` et `table-caption`



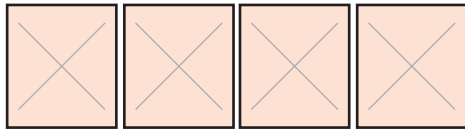
CSS

```
img{
  vertical-align:bottom;
}
```



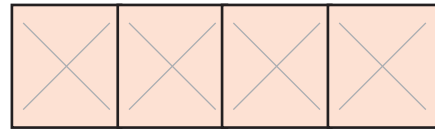
CSS

```
img{
  vertical-align:middle;
}
```



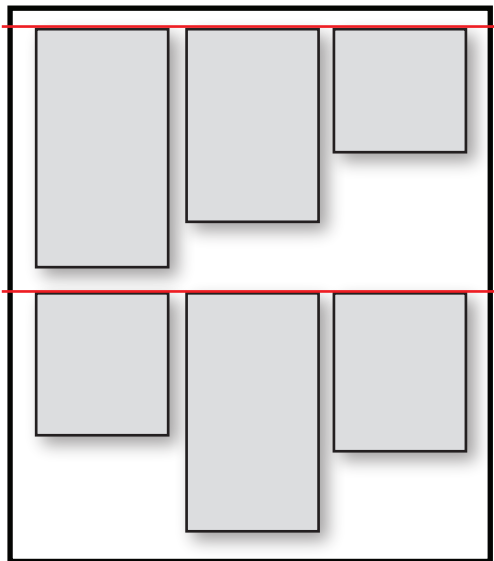
CSS

```
img{
  display:inline; /* par défaut */
}
```



CSS

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



HTML

```
<div class="contenant">
  <div class="col"></div>
  <div class="col"></div>
  <div class="col"></div>
  <div class="col"></div>
  <div class="col"></div>
  <div class="col"></div>
</div>
```

CSS

```
div.contenant{
  width:300px;
}
div.col{
  width:100px;
  display:inline-block;
  vertical-align:top;
}
```

HTML

```
<div class="contenant">
  <div class="col"></div><div class="col"></div><div class="col"></div>
</div>
```

HTML

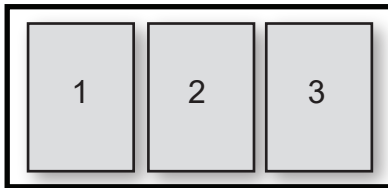
```
<div class="contenant">
  <div class="col"></div><!--
--><div class="col"></div><!--
--><div class="col"></div>
</div>
```

CSS : FLEX

Principes de flex (agir sur les contenus par le contenant)

Comp. : Google Chrome : 21, IE: 11, Firefox : 28, Safari : 6.1, Opera : 12.1

Pardéfaut



HTML

```
<div class="contenant">
  <div class="bloc1">1</div>
  <div class="bloc2">2</div>
  <div class="bloc3">3</div>
</div>
```

CSS

```
div.contenant{
  display:flex;
  flex-direction:row;/* défaut */
  flex-wrap:nowrap;/* défaut */
  justify-content:flex-start;/* défaut */
  align-items:auto;/* défaut */
}
```

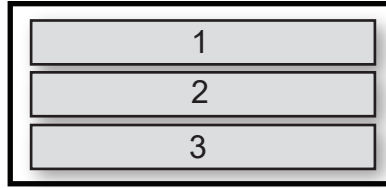
En rangée inversée



CSS

```
div.contenant{
  display:flex;
  flex-direction:row-reverse;
}
```

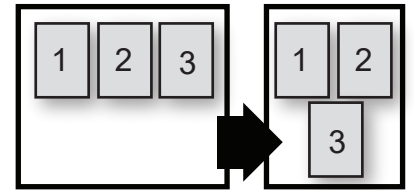
En colonne



CSS

```
div.contenant{
  display:flex;
  flex-direction:column;
}
```

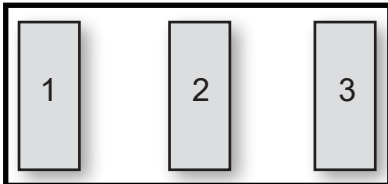
Gestion des objets



CSS

```
div.contenant{
  display:flex;
  flex-wrap:wrap;
}
```

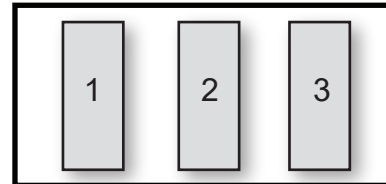
Gestion des espaces



CSS

```
div.contenant{
  display:flex;
  justify-content:space-between;
}
```

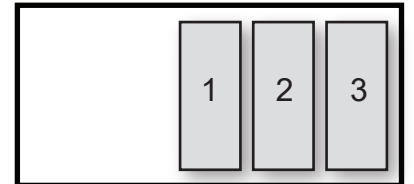
Gestion des espaces



CSS

```
div.contenant{
  display:flex;
  justify-content:space-around;
}
```

Gestion des espaces

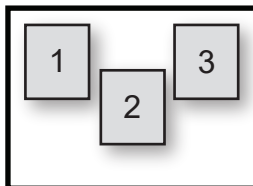


CSS

```
div.contenant{
  display:flex;
  justify-content:flex-end;
}
```

Agir directement sur les objets

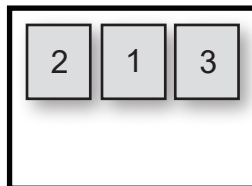
Alignement



CSS

```
div.bloc2{
  align-self:center;
}
```

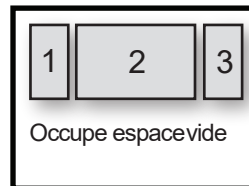
Ordre



CSS

```
div.bloc2{
  order:-1;
}
```

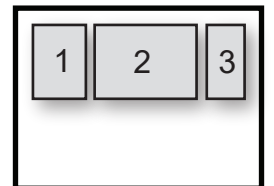
Flexibilité



CSS

```
div.bloc2{
  flex:1;
}
```

Flexibilité répartie



CSS

```
div.bloc1{
  flex:1;
}
div.bloc2{
  flex:2;
}
```

CSS: GRID

Principes de grid (agir sur les contenus par le contenant)

Comp. : Chrome : 57, IE: 10, Firefox : 52, Safari : 10.3, Opera : 44

Pardéfaut



HTML

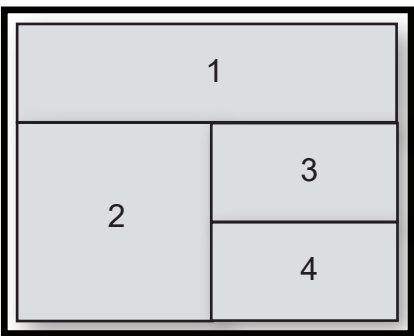
```
<div class="contenant">
  <div class="bloc1">1</div>
  <div class="bloc2">2</div>
  <div class="bloc3">3</div>
  <div class="bloc4">4</div>
</div>
```

CSS

```
div.contenant{
  display:grid;
  grid-template-columns:auto auto;
}
```

Gestion des zones

Définir des zones



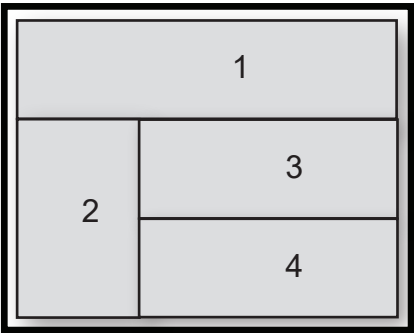
CSS

```
div.contenant{
  display:grid;
  grid-template-areas:
    "b1 b1"
    "b2 b3"
    "b2 b4";
}

div.bloc1{ grid-area:b1;}
div.bloc2{ grid-area:b2;}
div.bloc3{ grid-area:b3;}
div.bloc4{ grid-area:b4;}
```

Volume des zones

Unités de mesure en fraction (fr)



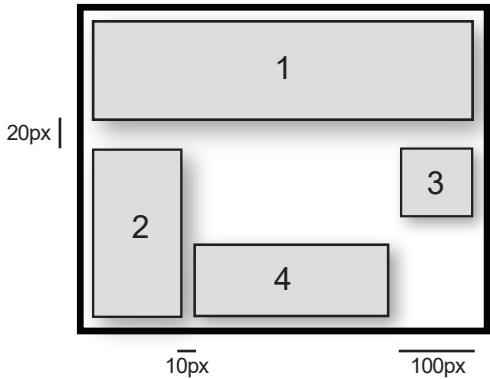
CSS

```
div.contenant{
  display:grid;
  grid-template-columns:1fr 2fr;
  grid-template-areas:
    "b1 b1"
    "b2 b3"
    "b2 b4";
}

div.bloc1{ grid-area:b1;}
div.bloc2{ grid-area:b2;}
div.bloc3{ grid-area:b3;}
div.bloc4{ grid-area:b4;}
```

Espaces entre les cellules

Marges



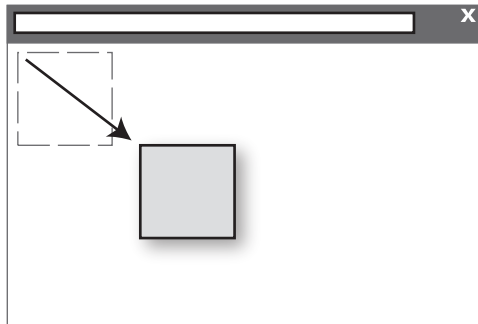
CSS

```
div.contenant{
  display:grid;
  grid-template-columns:1fr 2fr 100px;
  grid-gap:20px 10px;
  grid-template-areas:
    "b1 b1 b1"
    "b2 . b3"
    "b2 b4 .";
}

div.bloc1{ grid-area:b1;}
div.bloc2{ grid-area:b2;}
div.bloc3{ grid-area:b3;}
div.bloc4{ grid-area:b4;}
```

CSS: POSITION

Positions



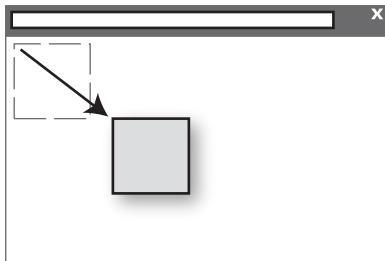
HTML

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

CSS

```
div.contenant{
  width:100px;
  height:100px;
  position:relative;
  left:110px;
  top:100px;
}
```

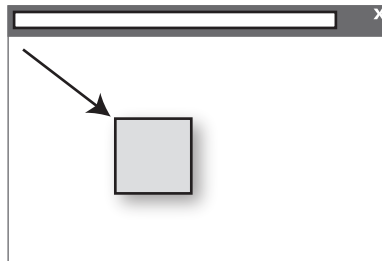
La propriété position se combine avec les propriétés (left, right, top ou bottom).



position:relative

L'objet se déplace mais laisse occupé sa position d'origine.

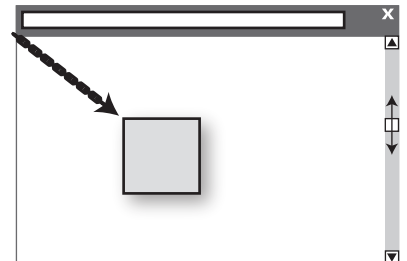
1. Il reste active dans le flux.
2. L'objet se déplace depuis sa position d'origine (avec «top»/«bottom» et «left»/«right»).



position:absolute

L'objet se déplace et libère sa position d'origine.

1. Il n'est plus en interaction avec le flux.
2. L'objet se déplace à partir d'un point de référence (avec «top»/«bottom» et «left»/«right»).

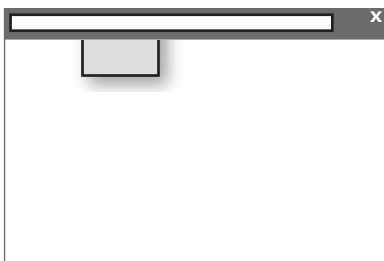


position:fixed

L'objet est ancré à la fenêtre du navigateur. Il ne se déplace pas avec le reste du contenu lors du déroulement de la page.

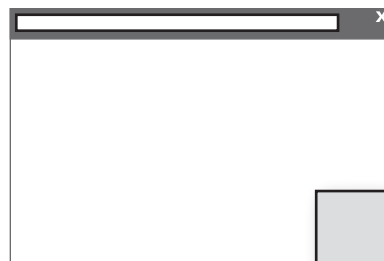
1. Il n'est plus en interaction avec le flux.
2. L'objet se déplace à partir d'un point de référence (avec «top»/«bottom» et «left»/«right»).

Déplacement depuis quel côté



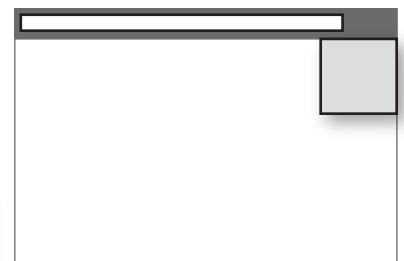
✓ top:-50px
left:100px

L'objet se déplace hors de la fenêtre sans qu'il n'apparaisse la barre de défilement.



✓ bottom:0px
right:0px

La propriété utilisée pour définir l'emplacement de l'objet détermine le côté à partir duquel l'objet sera positionné.



✗ left:0px
right:0px

Deux propriétés indiquant des côtés opposés pour le positionnement d'un objet ne peuvent pas être combinés. Le dernier des deux sera appliqué.

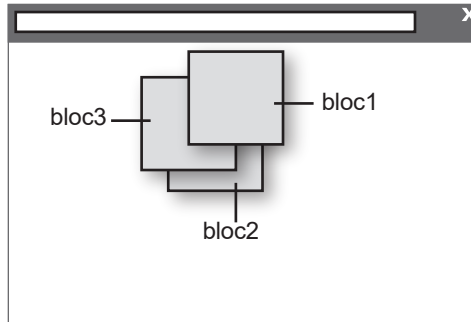
CSS: POSITION

z-index

HTML

```
<div class="bloc1"></div>
<div class="bloc2"></div>
<div class="bloc3"></div>
```

La propriété z-index permet de redéfinir l'ordre de superposition des objets en position relative, absolue ou fixed. Par défaut, c'est le dernier élément HTML qui est au-dessus des autres.



CSS

```
div.bloc1{
  z-index:3;
}
div.bloc2{
  z-index:1;
}
div.bloc3{
  z-index:2;
}
```

Combine : Positionner un objet par rapport à un autre

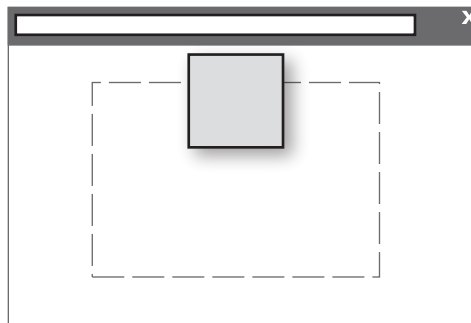
A savoir :

Par défaut un élément en position absolue se positionne par rapport à la fenêtre du navigateur.

Les conditions :

1. L'objet à positionner doit être contenu dans l'objet de référence.
2. Le contenant de référence doit détenir une position relative, absolue ou fixed.

Cas1 : Bloc contenu



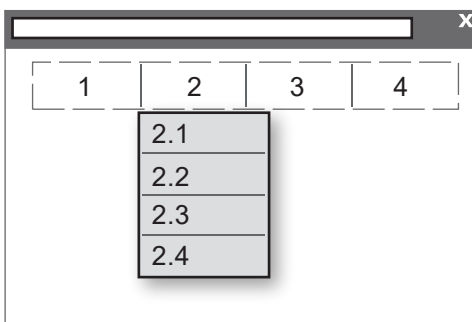
HTML

```
<div class="contenant">
  <div class="bloc"></div>
</div>
```

CSS

```
div.contenant{
  position:relative;
  /* absolute ou fixed */
}
div.bloc{
  position:absolute;
  top:-30px;
  left:200px;
}
```

Cas2 : Menu-déroulant



HTML

```
<ul class="menu">
  <li>1</li>
  <li>2
    <ul>
      <li>2.1</li>
      <li>2.2</li>
      <li>2.3</li>
      <li>2.4</li>
    </ul>
  </li>
  <li>3</li>
  <li>4</li>
</ul>
```

CSS

```
div.menu > li{
  height:70px;
}
div.menu > li{
  position:relative;
}
div.menu > li > ul{
  position:absolute;
  top:70px;
}
```