

Design CSS : arrondis, ombres et dégradés

Nous entrons dans un domaine où il faudra faire attention à certaines règles. Ici encore, CanIUse (<https://caniuse.com/>) sera d'une grande utilité pour vérifier les compatibilités des navigateurs.

En général, on ajoute certains préfixes pour que les règles fonctionnent sur certaines familles de navigateurs. Ainsi, on aura : moz- pour Firefox, webki- pour Chrome et Safari, o- pour Opéra et ms- pour IE.

I. Effets sur le texte

CSS3 contient plusieurs fonctionnalités de texte.

Dans cette partie, vous apprendrez à connaître les propriétés de texte suivantes :

- text-shadow ;
- word-wrap.

A. Ombre sur le texte

En CSS3, la propriété text-shadow applique une ombre au texte.

Vous pouvez spécifier l'ombre horizontale, l'ombre verticale, la distance de flou, et la couleur de l'ombre.

Sur h2, on indique :

```
h2 {
  text-align: center;
  text-shadow: 2px 3px 2px #22ffef;
}
```

Fig. 1 Design1 © SkillandYou

La règle **text-shadow** se décompose comme suit :

Les deux premières valeurs se rapportent au décalage de l'ombre sur l'axe x (horizontal) et y (vertical).

Le dernier chiffre est lié au flou appliqué sur l'ombre (facultatif). Le rendu est le suivant.

Ici, le code hexadécimal qui vient en dernier indique la couleur de l'ombre.

Texte ombré

Fig. 2 Design2 © SkillandYou

On peut n'indiquer que deux chiffres et une couleur. Dans ce cas l'ordre de la couleur ou des chiffres est indifférent, mais le premier chiffre et le second chiffre vont toujours se rapporter d'abord aux x et ensuite aux y.

Par exemple :

```
h2 {
  text-align: center;
  text-shadow: rgba(20%, 30%, 40%, 0.5) 2px 3px;
}
```

Fig. 3 Design3 © SkillandYou

Donne :

Texte ombré

Fig. 4 Design4 © SkillandYou

On peut également placer plusieurs ombres, il suffit d'espacer chaque définition d'ombre par une virgule, comme ceci :

```
h2 {
  text-align: center;
  text-shadow: rgba(20%, 30%, 40%, 0.5) 12px 3px,
  0px 0px 15px #f00, blue -8px -3px;
}
```

Fig. 5 Design5 © SkillandYou

Qui nous donne :

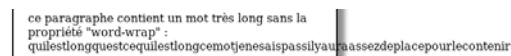
Texte ombré

Fig. 6 Design6 © SkillandYou

Évidemment, c'est difficile à lire, mais on peut distinguer l'ombre grise (en rgba), l'ombre bleue (les valeurs négatives des pixels) et l'ombre rouge (celle qui forme un halo dû au flou de 15px).

B. Forcer un retour à la ligne

Si un mot est trop long pour tenir dans une zone, il sort de cette dernière, comme sur l'image ci-dessous :



ce paragraphe contient un mot très long sans la propriété "word-wrap" : quillestlongquestcequillestlongcemotjenesaispassilyauraassezdeplacepourlecontenir

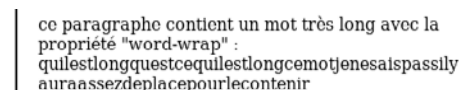
Fig. 7 Design7 © SkillandYou

On va demander aux mots longs d'avoir la capacité d'être coupés pour qu'ils s'étalent sur la ligne suivante :

```
p {  
  word-wrap: break-word;  
}
```

Fig. 8 Design8 © SkillandYou

Le problème est résolu.



ce paragraphe contient un mot très long avec la propriété "word-wrap" : quillestlongquestcequillestlongcemotjenesaispassilyauraassezdeplacepourlecontenir

Fig. 9 Design9 © SkillandYou

II. Bordures

Avec CSS3, vous pouvez créer des bordures arrondies, ajouter une ombre à des boîtes, et utiliser une image comme bordure sans l'aide d'un programme de conception, comme Photoshop.

Nous allons aborder des règles CSS3 ayant traits aux blocs, qui sont les suivantes :

- **border-radius;**
- **box-shadow;**
- **border-image**
- **overflow**

A. Coins arrondis

L'ajout des coins arrondis avec CSS2 était difficile car nous devons utiliser des images différentes pour chaque coin.

En CSS3, la création de coins arrondis est facile.

On peut ajouter des coins arrondis à un élément de bloc avec le HTML suivant :

```
<div class="coins-arrondis">  
  <p>exemple coins arrondis</p>  
</div>
```

Fig. 10 Design10 © SkillandYou

Dans le CSS, on utilise **border-radius** :

```
.coins_arrondis {  
  border-radius: 15px 20px 0 5px;  
  width: 350px;  
  padding: 20px;  
  background-color: #F90;  
  text-align: center;  
}
```

Fig. 11 Design11 © SkillandYou

Le rendu :

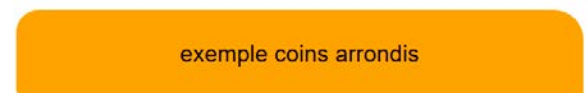


Fig. 12 Design12 © SkillandYou

Comme pour les raccourcis **margin** et **padding**, on indique la taille des arrondis pour chaque angle dans le sens des aiguilles d'une montre : haut, droite, bas, gauche.

B. Ombres

En CSS3, la propriété **box-shadow** est utilisée pour ajouter une ombre à des boîtes. On retrouve à peu près le même fonctionnement que pour **text-shadow**.

Exemple pour ajouter une ombre à un élément **div** :

HTML

```
<div class="ombre-boite">
  <p>exemple d'ombre</p>
</div>
```

Fig. 13 Design13 © SkillandYou

CSS

```
.ombre-boite {
  box-shadow: 15px 20px 5px #F00;
  width:350px;
  padding:20px;
  background-color:#F90;
  text-align:center;
}
```

Fig. 14 Design14 © SkillandYou

Le rendu :



Fig. 15 Design15 © SkillandYou

Syntaxe :

- la première valeur indique le décalage horizontal vers la droite, axe des x (ici 15px) ;
- le deuxième correspond au décalage vertical vers le bas, axe des y (ici 20px) ;
- le chiffre suivant indique la force du dégradé (ici 5px), c'est une indication facultative (on n'a pas forcément de flou à notre ombre portée) ;
- et enfin, la couleur (ici #F00).

On peut n'indiquer que deux chiffres et une couleur. Dans ce cas, l'ordre de la couleur ou des chiffres est indifférent, mais le premier chiffre et le second chiffre vont toujours se rapporter d'abord aux x et ensuite aux y. Enfin, on peut avoir plusieurs ombres, il suffit de séparer les déclarations par une virgule.

Par exemple :

```
.ombre-boite {
  box-shadow: 15px 20px 5px #F00, green
             -15px -10px;
  width:350px;
  padding:20px;
  background-color:#F90;
  text-align:center;
}
```

Fig. 16 Design16 © SkillandYou

Le rendu :



Fig. 17 Design17 © SkillandYou

C. Bordures avec images

La propriété **border-image** permet de spécifier une image pour bordure.

C'est une règle assez difficile à manipuler pour avoir un résultat correct.

On peut toutefois facilement obtenir un effet intéressant à l'aide d'un dégradé, que nous analyserons dans une autre section :

CSS

```
.image-de-bordure {
  border: 5px solid;
  width:350px;
  padding:20px;
  text-align:center;
  padding: 20px;
  border: 5px solid;
  border-image: linear-gradient(#434343, #acacac) 10;
}
```

Fig. 18 Design18 © SkillandYou

Le rendu :



Fig. 19 Design19 © SkillandYou

Cette règle n'est pas très bien supportée, donc nous n'allons pas nous y attarder.

D. Barre de défilement

Il est utile dans certains cas (design oblige) de savoir créer des blocs ayant une dimension fixe mais dont le contenu peut dépasser les limites dudit bloc. Le contenu sera donc visible grâce à une barre de défilement.

Graphiquement, cette barre de défilement est identique à celle du navigateur, et malheureusement vous ne pourrez pas changer son apparence (sauf sous IE).

Tableau n°1 Règles de style

overflow:scroll;	La barre de défilement sera toujours visible quel que soit le contenu.
overflow:auto;	La barre de défilement apparaîtra lorsque le contenu dépassera les limites du bloc.
overflow:hidden;	Ici, il ne s'agit pas tant d'agir sur la barre de défilement que d'indiquer au bloc ce qu'il doit faire en cas de débordement de contenu. Ici on demande de le rendre invisible. Nous l'avons déjà abordé pour mettre fin aux flottements (alternative à div.clear et à .clearfix).

HTML

```
<div class="bloc_defilement">
  <p>
    Cras nec est luctus, pellentesque arcu id,
    egestas libero. Nulla venenatis sagittis nibh
    sed hendrerit. Suspendisse at lectus purus. Sed
    faucibus mattis dui ut efficitur. Cras finibus
    posuere egestas. Aenean ac urna sit amet lacus
    varius scelerisque. Donec venenatis, diam
    congue placerat consectetur, sem ligula
    efficitur massa, a egestas odio lacus pharetra
    lectus. Integer eleifend, nunc ac auctor
    mattis, velit est imperdiet felis, quis
    fermentum justo nulla sit amet tortor. Maecenas
    at dolor id ex tincidunt cursus.
  </p>
</div>
```

Fig. 20 Design20 © SkillandYou

CSS

```
.bloc_defilement {
  width:250px;
  height:200px;
  overflow:auto;
  text-align:justify;
  padding:10px;
  border:1px solid #222;
}
```

Fig. 21 Design21 © SkillandYou

Rendu :

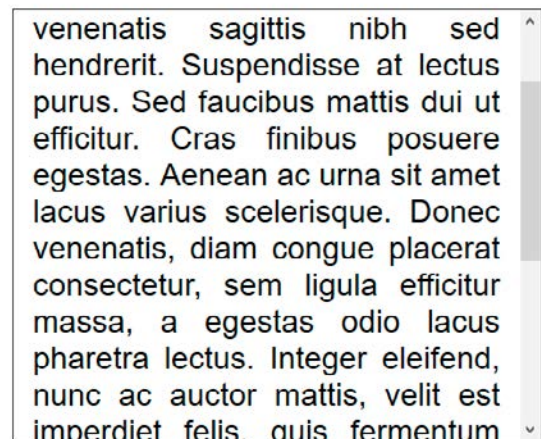


Fig. 22 Design22 © SkillandYou

III. Les images de fond

CSS3 contient des propriétés qui permettent un meilleur contrôle de l'élément de fond.

Nous allons nous intéresser aux propriétés suivantes :

- background-size;
- background-origin.

A. background-size

La propriété **background-size** spécifie la taille de l'image de fond.

Spécifier la taille de l'image de fond permet de réutiliser les images de fond dans des contextes différents.

Vous pouvez spécifier la taille en pixels ou en pourcentage. Si vous spécifiez la taille en pourcentage, la taille est relative à la largeur et à la hauteur de l'élément parent.

Exemple 1

Redimensionner une image de fond :

```
.image-fond {
  background-image: url(fond.jpg);
  background-size: 260px 260px;
  background-repeat: no-repeat;
  background-position: center right;
  width: 450px;
  height: 450px;
  border: 5px solid #ede1e1;
}
```

Fig.23 Design23 © SkillandYou

Image de fond redimensionnée.

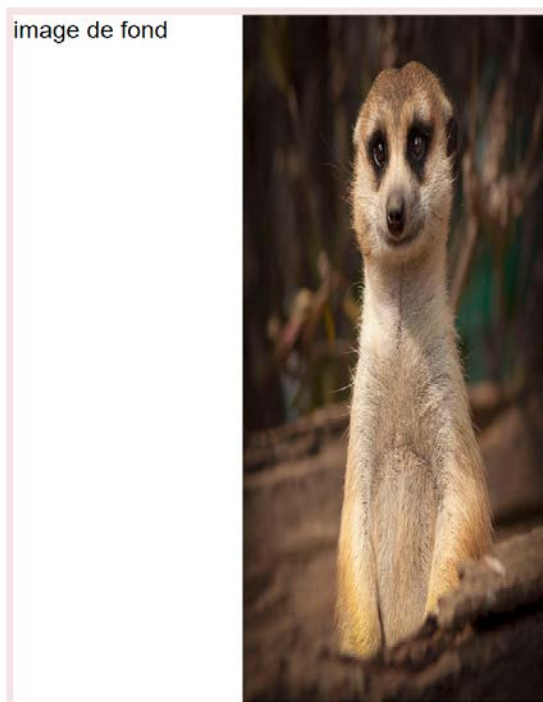


Fig.24 Design24 © SkillandYou

Exemple 2

Étirer l'image de fond pour remplir complètement la zone de contenu :

```
.image-fond {
  background-image: url(fond.jpg);
  background-size: 100% 100%;
  background-repeat: no-repeat;
  width: 450px;
  height: 450px;
  border: 5px solid #ede1e1;
  /*le background-position est inutile*/
}
```

Fig.25 Design25 © SkillandYou

B. background-origin

La propriété **background-origin** spécifie la zone de positionnement des images de fond.

L'image d'arrière-plan peut être placée à l'intérieur de la boîte, dans la marge intérieure ou dans la bordure.

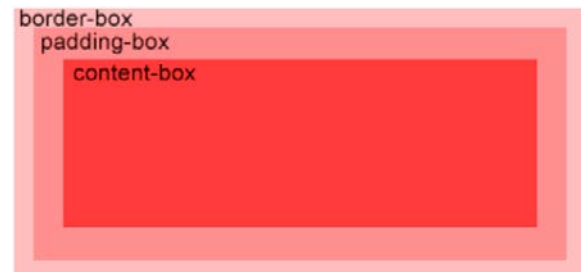


Fig.26 Design26 © SkillandYou

On peut placer l'image de fond à l'intérieur de la boîte sans mordre sur le padding :

```
.image-fond {
  background-image: url(fond.jpg);
  background-size: 100% 100%;
  background-repeat: no-repeat;
  background-position: center right;
  background-origin: content-box;
  width: 350px;
  height: 450px;
  border: 5px solid #ede1e1;
  padding: 15px;
}
```

Fig.27 Design27 © SkillandYou

Rendu :



Fig. 28 Design28 © SkillandYou

C. Un arrière-plan plein écran

Pour terminer, voici le moyen de réaliser une image de fond sur la totalité d'une page qui ne soit pas déformée malgré le redimensionnement de la fenêtre du navigateur. Il s'agit de combiner le fait que l'image soit immobile, positionnée au centre et couvre l'ensemble du body qui l'appelle grâce à la propriété **cover** sur **background-size**.

```
body {  
  background-image: url(fond.jpg);  
  background-size: cover;  
  background-attachment: fixed;  
  background-repeat: no-repeat;  
  background-position: center top;  
}
```

Fig. 29 Design29 © SkillandYou

Rendu :

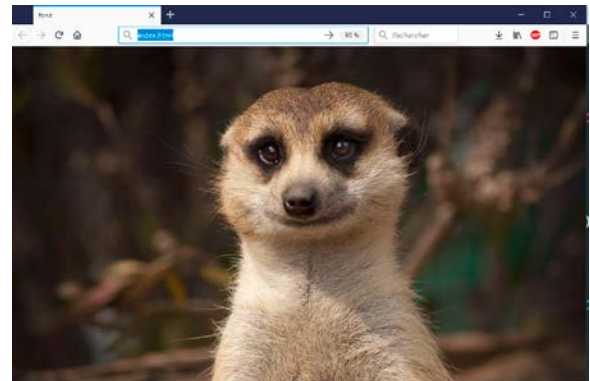


Fig. 30 Design30 © SkillandYou

On modifie la taille de la fenêtre :

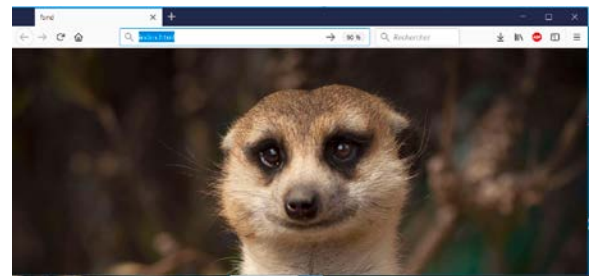


Fig. 31 Design31 © SkillandYou

IV. Les dégradés de couleur

Propriétés :

- **linear-gradient()**
- **radial-gradient()**

Voici l'écriture d'un dégradé linéaire :

```
.degrade {  
  padding: 50px;  
  background-image: linear-gradient(to bottom, #000, #ededed);  
  /*ou background: linear-gradient(to bottom, #000, #ededed);*/  
}
```

Fig. 32 Design32 © SkillandYou

Rendu :



Fig. 33 Design33 © SkillandYou

Il y a plusieurs possibilités au niveau de la syntaxe des paramètres entre parenthèses.

A. linear-gradient()

Un gradient linéaire est créé en spécifiant une ligne de gradient rectiligne, puis plusieurs couleurs placées le long de cette ligne.

linear-gradient (direction de la ligne de gradient, couleur, couleur...)

Le premier argument à la fonction précise la ligne du gradient, ce qui lui donne une direction et donc détermine de quelle façon les couleurs sont positionnées. Il peut être omis, auquel cas il est par défaut « **to bottom** » (vers le bas).

La ligne du gradient peut être spécifiée de deux façons :

En utilisant des angles

De **0deg** point haut, les angles positifs dans le sens horaire, de sorte que par exemple **90deg** est le point vers la droite. Les angles peuvent être négatifs et tournent donc dans le sens inverse des aiguilles d'une montre.

En utilisant des mots-clés

Si l'argument est « to top », « to right », « to bottom », ou « to left », l'angle de la ligne du gradient est respectivement de « **0deg** », « **90deg** », « **180deg** », ou « **270deg** ».

Pour vous y retrouver plus facilement, imaginez une ligne factice qui dessinerait votre dégradé de la couleur A vers la couleur B. Si « to bottom » est déclaré, le dégradé partira verticalement de haut en bas de la couleur A vers la couleur B. Les utilisateurs de Photoshop utilisent ce procédé pour créer des dégradés.

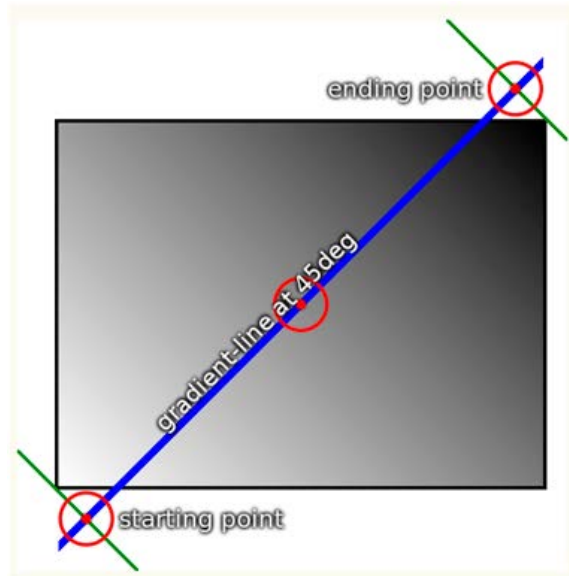


Fig. 34 Design34 © SkillandYou

Les couleurs peuvent être exprimées avec :

- leur nom : blue, black, etc. ;
- leur code hexadécimal : #cc0000 ;
- leur code rgba (a, b, c, d) ou a, b, c sont des nombres de 0 à 255 ou des pourcentages qui représentent respectivement la proportion de rouge, de vert et de bleu. D représente l'opacité ou la transparence exprimée de 0 (transparent) à 1 (opaque).

Gradient vertical

Voici différentes manières de spécifier un même gradient vertical de base.

```
background: linear-gradient(#eded, #fd2233);
background: linear-gradient(to bottom, #eded, #fd2233);
background: linear-gradient(180deg, #eded, #fd2233);
background: linear-gradient(to bottom, #eded 0%, #fd2233 100%);
```

Fig. 35 Design35 © SkillandYou

Rendu :



Fig. 36 Design36 © SkillandYou

Gradient oblique

```
background: linear-gradient(135deg, #ededed, #fd2233);  
background: linear-gradient(-45deg, #ededed, #fd2233);
```

Fig.37 Design37 © SkillandYou

Rendu :

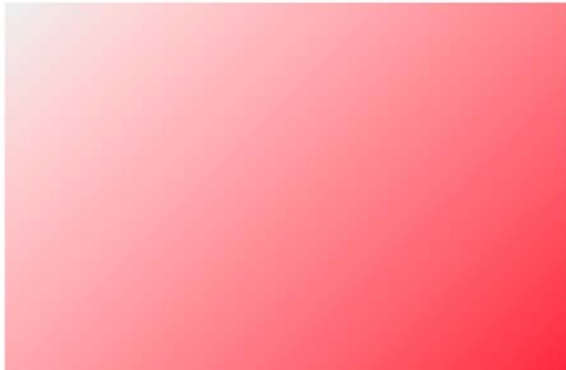


Fig.38 Design38 © SkillandYou

Gradient avec plusieurs couleurs avec arrêt

```
background: linear-gradient(yellow, blue 20%, #0f0);
```

Fig.39 Design39 © SkillandYou

Rendu :



Fig.40 Design40 © SkillandYou

Gradient oblique à plusieurs points d'arrêt

Exemple identique à l'exemple 2 mais en utilisant des mots-clés.

```
background: linear-gradient(to top right, #fd2233, #33aab2, #ededed, #23be1a);
```

Fig.41 Design41 © SkillandYou

Rendu :



Fig.42 Design42 © Skillandyou

To top right veut dire que la ligne de gradient part de bas en haut et vers la droite ce qui équivaut à un angle de $180 + 45 = 225$ degrés.

B. radial-gradient()

Le dégradé radial comme son nom l'indique est un dégradé dont la ligne de gradient est ovale. Nous ne parlerons dans ce cours que des positions principales.

Syntaxe : **radial-gradient(position, forme, taille, longueur)**

position est la position de départ du dégradé. Il prend les mêmes valeurs que la propriété **background-position**. Par défaut : **center**.

forme : **circle** ou **ellipse**.

taille détermine la taille du gradient de forme de fin. Par défaut :

- closest-side (le côté le plus proche)
- farthest-side (le côté le plus éloigné)
- closest-corner (le coin le plus proche)
- farthest-corner (le coin le plus éloigné)

longueur donne le rayon du cercle de manière explicite. Les valeurs négatives ne sont pas valides. Si deux longueurs sont indiquées, elles représentent le rayon X et le rayon Y. Notez que les pourcentages ne sont pas autorisés ici, ils ne peuvent être utilisés que pour spécifier la taille d'un dégradé elliptique.

Plusieurs syntaxes pour le même résultat :

```
background: radial-gradient(#fd2233, #33aab2);
background: radial-gradient(ellipse at center, #fd2233 0%, #33aab2 100%);
background: radial-gradient(farthest-corner at 50% 50%, #fd2233, #33aab2);
```

Fig.43 Design43 © Skillandyou

Rendu :

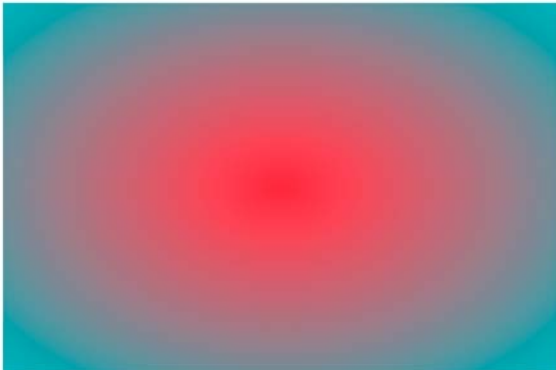


Fig.44 Design44 © Skillandyou

Autre syntaxe possible :

```
background: radial-gradient(circle, #23be1a, #fd2233);
```

Fig.45 Design45 © Skillandyou

Rendu :

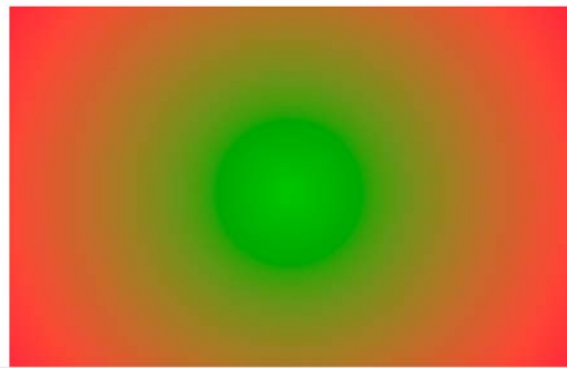


Fig.46 Design46 © Skillandyou

Autre syntaxe possible :

```
background: radial-gradient(#33aab2, #23be1a, #fd2233);
```

Fig.47 Design47 © Skillandyou

Rendu :

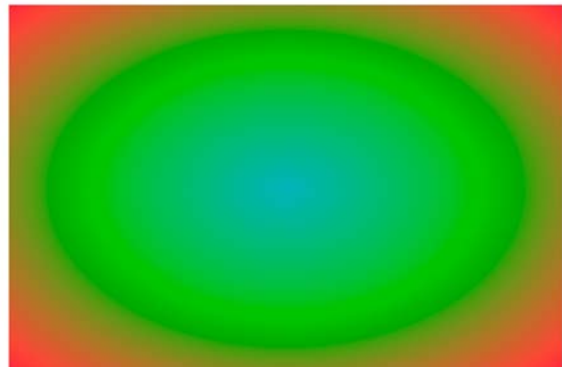


Fig.48 Design48 © Skillandyou

Autre syntaxe possible :

```
background: radial-gradient(farthest-side at left bottom, #33aab2, #fd2233 350px);
```

Fig.49 Design49 © Skillandyou

Rendu :



Fig.50 Design50 © Skillandyou

Autre syntaxe possible :

```
background: radial-gradient(closest-side at 50px 80px, #33aab2, #23be1a, #fd2233);
background: radial-gradient(50px 80px at 50px 80px, #33aab2, #23be1a, #fd2233);
```

Fig.51 Design51 © Skillandyou

Rendu :



Fig.52 Design52 © Skillandyou

Autre syntaxe possible :

```
background: radial-gradient(closest-side circle at 50px 80px, #33aab2, #23be1a, #fd2233);  
background: radial-gradient(50px 50px at 50px 80px, #33aab2, #23be1a, #fd2233);
```

Fig.53 Design53 © Skillandyou

Notez que pour obtenir un cercle, il suffit d'indiquer une taille identique de largeur (les deux premiers chiffres 50px et 50px).

Rendu :



Fig.54 Design54 © Skillandyou

Enfin sachez qu'il existe des générateurs de CSS en ligne pour les dégradés, par exemple : <https://www.colorzilla.com/gradient-editor/>

V. La boîte

A. Les multi-colonnes

Avec CSS 3 est apparue une règle qui permet de diviser un élément (souvent un paragraphe) en colonnes dans une mise en page de type « Journal ». Vous n'avez donc pas le choix dans le remplissage des colonnes, qui se fait automatiquement.

Il ne faut pas confondre ceci avec le colonnage des éléments en bloc et autres méthodes de mise en page (float, inline-block, inline-table ou flexbox).

Les multi-colonnes se décomposent en nombreuses règles mais les navigateurs demandent des préfixes ou emploient des instructions alternatives pour certaines d'entre elles quand elles n'en supportent tout simplement pas d'autres.

1. **column-count**

2. **column-width**

3. **column-gap**

4. **column-rule**

(raccourci : column-rule-width + column-rule-style + column-rule-color)

5. **columns**

(Raccourci : column-count + column-width)

On va demander trois colonnes :

```
.trois-colonnes {  
  -webkit-column-count: 3;  
  -moz-column-count: 3;  
  column-count: 3;  
}
```

Fig.55 Design55 © Skillandyou

Rendu :

Cras nec est luctus, pellentesque arcu id, egestas libero. Nulla venenatis sagittis nibh sed hendrerit. Suspendisse at lectus purus. Sed faucibus mattis dui ut efficitur. Cras	finibus posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis, diam congue placerat consectetur, sem ligula efficitur massa, a egestas odio	lacus pharetra lectus. Integer eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis fermentum justo nulla sit amet tortor. Maecenas at dolor id ex tincidunt cursus.
---	--	--

Fig. 56 Design56 © Skillandyou

On peut choisir de travailler non pas par le nombre de colonnes mais en indiquant une largeur de colonne :

```
.trois-colonnes {
  -webkit-column-width: 180px;
  -moz-column-width: 180px;
  column-width: 180px;
}
```

Fig. 57 Design57 © Skillandyou

Le rendu :

Cras nec est luctus, pellentesque arcu id, egestas libero. Nulla venenatis sagittis nibh sed hendrerit. Suspendisse at lectus purus. Sed faucibus	mattis dui ut efficitur. Cras finibus posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis, diam congue placerat	consectetur, sem ligula efficitur massa, a egestas odio lacus pharetra lectus. Integer eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis	fermentum justo nulla sit amet tortor. Maecenas at dolor id ex tincidunt cursus.
---	--	---	--

Fig. 58 Design58 © Skillandyou

Vous remarquez qu'on a une colonne de plus. Le navigateur calcule le nombre de colonnes qu'il est possible de disposer sur la largeur du bloc. En réalité, **column-width** fonctionne comme une sorte de **min-width**. La largeur de chaque colonne peut être plus élevée, mais jamais plus basse. Si d'aventure le bloc ne lui laissait pas de largeur suffisante, les colonnes seront moins nombreuses, jusqu'à ne laisser plus qu'une seule ligne.

Il est donc très utile de donner un nombre de colonnes préféré mais également une taille de colonne minimale afin de contrôler notre rendu et que les colonnes ne soient ni trop étroites ni trop nombreuses. Pour cela, on utilise la règle raccourcie **columns**, avec comme première propriété la largeur et comme seconde propriété le nombre de colonnes :

```
.trois-colonnes {
  -webkit-columns: 180px 3;
  -moz-columns: 180px 3;
  columns: 180px 3;
}
```

Fig. 59 Design59 © Skillandyou

Le rendu :

Cras nec est luctus, pellentesque arcu id, egestas libero. Nulla venenatis sagittis nibh sed hendrerit. Suspendisse at lectus purus. Sed faucibus mattis dui ut efficitur. Cras finibus posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis, diam congue placerat consectetur, sem ligula efficitur massa, a egestas odio lacus pharetra lectus. Integer eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis fermentum justo nulla sit amet tortor. Maecenas at dolor id ex tincidunt cursus. Cras

nec est luctus, pellentesque arcu id, egestas libero. Nulla venenatis sagittis nibh sed hendrerit. Suspendisse at lectus purus. Sed faucibus mattis dui ut efficitur. Cras finibus posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis, diam congue placerat consectetur, sem ligula efficitur massa, a egestas odio lacus pharetra lectus. Integer eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis fermentum justo nulla sit amet tortor. Maecenas at dolor id ex tincidunt cursus.

Cras nec est luctus, pellentesque arcu id, egestas libero. Nulla venenatis sagittis nibh sed hendrerit. Suspendisse at lectus purus. Sed faucibus mattis dui ut efficitur. Cras finibus posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis, diam congue placerat consectetur, sem ligula efficitur massa, a egestas odio lacus pharetra lectus. Integer eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis fermentum justo nulla sit amet tortor. Maecenas at dolor id ex tincidunt cursus.

Fig.60

Maintenant nous allons ajouter un peu de texte et une balise <h2> ainsi que demander des marges pour ces deux éléments, afin de démontrer que si le parent (ici div.trois-colonnes) est en mode multi-colonnes, nous n'observons pas de changement significatif dans la disposition :

HTML

```
<div class="trois-colonnes">
  <p>
    Cras nec est luctus, pellentesque arcu id, egestas libero. Nulla venenatis sagittis nibh sed
    hendrerit. Suspendisse at lectus purus. Sed faucibus mattis dui ut efficitur. Cras finibus
    posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis, diam congue
    placerat consectetur, sem ligula efficitur massa, a egestas odio lacus pharetra lectus. Integer
    eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis fermentum justo nulla sit
    amet tortor. Maecenas at dolor id ex tincidunt cursus.
    Cras nec est luctus, pellentesque arcu id, egestas libero. Nulla venenatis sagittis nibh sed
    hendrerit. Suspendisse at lectus purus. Sed faucibus mattis dui ut efficitur. Cras finibus
    posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis, diam congue
    placerat consectetur, sem ligula efficitur massa, a egestas odio lacus pharetra lectus. Integer
    eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis fermentum justo nulla sit
    amet tortor. Maecenas at dolor id ex tincidunt cursus.
  </p>
  <h2>j'intertitre sans bevue</h2>
  <p>
    Cras finibus posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis,
    diam congue placerat consectetur, sem ligula efficitur massa, a egestas odio lacus pharetra
    lectus. Integer eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis fermentum
    justo nulla sit amet tortor. Maecenas at dolor id ex tincidunt cursus.
    Cras nec est luctus, pellentesque arcu id, egestas libero. Nulla venenatis sagittis nibh sed
    hendrerit. Suspendisse at lectus purus. Sed faucibus mattis dui ut efficitur. Cras finibus
    posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis, diam congue
    placerat consectetur, sem ligula efficitur massa, a egestas odio lacus pharetra lectus. Integer
    eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis fermentum justo nulla sit
    amet tortor. Maecenas at dolor id ex tincidunt cursus.
  </p>
</div>
```

Fig.61

CSS

```

.trois-colonnes {
  -webkit-columns: 180px 3;
  -moz-columns: 180px 3;
  columns: 180px 3;
}
h2 {
  margin: 0.8em 0;
}
p {
  margin: 0 0 0.2em;
}

```

Fig. 62

Le rendu :

Cras nec est luctus, pellentesque arcu id, egestas libero. Nulla venenatis sagittis nibh sed hendrerit. Suspendisse at lectus purus. Sed faucibus mattis dui ut efficitur. Cras finibus posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis, diam congue placerat consectetur, sem ligula efficitur massa, a egestas odio lacus pharetra lectus. Integer eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis fermentum justo nulla sit amet tortor. Maecenas at dolor id ex tincidunt cursus. Cras nec est luctus, pellentesque arcu id, egestas libero. Nulla venenatis sagittis nibh sed hendrerit. Suspendisse at lectus purus. Sed faucibus mattis dui ut efficitur. Cras

finibus posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis, diam congue placerat consectetur, sem ligula efficitur massa, a egestas odio lacus pharetra lectus. Integer eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis fermentum justo nulla sit amet tortor. Maecenas at dolor id ex tincidunt cursus.

j'intertitre sans bévue

Cras finibus posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis, diam congue placerat consectetur, sem ligula efficitur massa, a egestas odio lacus pharetra lectus. Integer

eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis fermentum justo nulla sit amet tortor. Maecenas at dolor id ex tincidunt cursus. Cras nec est luctus, pellentesque arcu id, egestas libero. Nulla venenatis sagittis nibh sed hendrerit. Suspendisse at lectus purus. Sed faucibus mattis dui ut efficitur. Cras finibus posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis, diam congue placerat consectetur, sem ligula efficitur massa, a egestas odio lacus pharetra lectus. Integer eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis fermentum justo nulla sit amet tortor. Maecenas at dolor id ex tincidunt cursus.

Fig. 63

Le premier <p>, <h2>, et le second <p> s'intègrent parfaitement dans le bloc.

On peut également indiquer l'écart entre les colonnes grâce à la propriété **column-gap**.

```

.trois-colonnes {
  -webkit-columns: 180px 3;
  -moz-columns: 180px 3;
  columns: 180px 3;
  -webkit-column-gap: 50px;
  -moz-column-gap: 50px;
  column-gap: 50px;
}

```

Fig. 64

Rendu :

Cras nec est luctus, pellentesque arcu id, egestas libero. Nulla venenatis sagittis nibh sed hendrerit. Suspendisse at lectus purus. Sed faucibus mattis dui ut efficitur. Cras finibus posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis, diam congue placerat consectetur, sem ligula efficitur massa, a egestas odio lacus pharetra lectus. Integer eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis fermentum justo nulla sit amet tortor. Maecenas at dolor id ex tincidunt cursus. Cras nec est luctus, pellentesque arcu id, egestas libero. Nulla venenatis sagittis nibh sed hendrerit. Suspendisse at lectus purus. Sed faucibus mattis dui ut efficitur. Cras

finibus posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis, diam congue placerat consectetur, sem ligula efficitur massa, a egestas odio lacus pharetra lectus. Integer eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis fermentum justo nulla sit amet tortor. Maecenas at dolor id ex tincidunt cursus.

j'intertitre sans bévue

Cras finibus posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis, diam congue placerat consectetur, sem ligula efficitur massa, a egestas odio lacus pharetra lectus. Integer

eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis fermentum justo nulla sit amet tortor. Maecenas at dolor id ex tincidunt cursus. Cras nec est luctus, pellentesque arcu id, egestas libero. Nulla venenatis sagittis nibh sed hendrerit. Suspendisse at lectus purus. Sed faucibus mattis dui ut efficitur. Cras finibus posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis, diam congue placerat consectetur, sem ligula efficitur massa, a egestas odio lacus pharetra lectus. Integer eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis fermentum justo nulla sit amet tortor. Maecenas at dolor id ex tincidunt cursus.

Fig. 65

À cela s'ajoute une propriété qui permet de placer une sorte de bordure pour distinguer les colonnes et qui ressemble justement à la propriété raccourcie **border**. Il s'agit de **column-rule**. Cette dernière définit la largeur, le style et la couleur de la ligne verticale de séparation entre les colonnes.

Exemple :

```
.trois-colonnes {
  -webkit-columns: 180px 3;
    -moz-columns: 180px 3;
     columns: 180px 3;
  -webkit-column-gap: 50px;
    -moz-column-gap: 50px;
     column-gap: 50px;
  -webkit-column-rule: 1px solid #f00;
    -moz-column-rule: 1px solid #f00;
     column-rule: 1px solid #f00;
}
```

Fig. 66

column-rule est donc un raccourci de l'épaisseur (**column-rule-width**), du style (**column-rule-style**) et de la couleur (**column-rule-color**).

Rendu :

Cras nec est luctus, pellentesque arcu id, egestas libero. Nulla venenatis sagittis nibh sed hendrerit. Suspendisse at lectus purus. Sed faucibus mattis dui ut efficitur. Cras finibus posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis, diam congue placerat consectetur, sem ligula efficitur massa, a egestas odio lacus pharetra lectus. Integer eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis fermentum justo nulla sit amet tortor. Maecenas at dolor id ex tincidunt cursus. Cras nec est luctus, pellentesque arcu id, egestas libero. Nulla venenatis sagittis nibh sed hendrerit. Suspendisse at lectus purus. Sed faucibus mattis dui ut efficitur. Cras

finibus posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis, diam congue placerat consectetur, sem ligula efficitur massa, a egestas odio lacus pharetra lectus. Integer eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis fermentum justo nulla sit amet tortor. Maecenas at dolor id ex tincidunt cursus.

j'intertitre sans bévée

Cras finibus posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis, diam congue placerat consectetur, sem ligula efficitur massa, a egestas odio lacus pharetra lectus. Integer

eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis fermentum justo nulla sit amet tortor. Maecenas at dolor id ex tincidunt cursus. Cras nec est luctus, pellentesque arcu id, egestas libero. Nulla venenatis sagittis nibh sed hendrerit. Suspendisse at lectus purus. Sed faucibus mattis dui ut efficitur. Cras finibus posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis, diam congue placerat consectetur, sem ligula efficitur massa, a egestas odio lacus pharetra lectus. Integer eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis fermentum justo nulla sit amet tortor. Maecenas at dolor id ex tincidunt cursus.

Fig. 67

Nous ne nous attarderons pas sur les autres règles qui sont très mal supportées, mais restez en éveil sur ce sujet.

B. La propriété box-sizing

Nous avons déjà abordé cette propriété qui vous permet de modifier le mode de calcul par défaut de la boîte CSS.

En effet cette dernière, comme vous le savez, se calcule en prenant en compte la largeur ou hauteur, les marges internes et externes ainsi que les bordures. Cela rend parfois les choses compliquées, surtout lorsque l'on va se confronter au **Design Responsive** dont le mode de calcul est en pourcentage. Cette propriété permet donc de modifier le calcul des dimensions de la boîte CSS par le navigateur en jouant sur la prise en compte ou non de la bordure (**border**), ou de la marge intérieure (**padding**). Seules les marges externes (**margin**) qui gèrent l'espace entre les boîtes ne sont pas modifiées par les propriétés de **box-sizing**.

Exemple :

Sans modifier le mode de calcul de la boîte, c'est-à-dire la valeur par défaut qui est équivalente à **box-sizing:content-box** :

```
.default-box-sizing {  
  width: 250px;  
  border: 10px solid #da2;  
  margin: 25px;  
  padding: 25px;  
  background-color: #a23;  
}
```

Fig. 68

Le rendu :

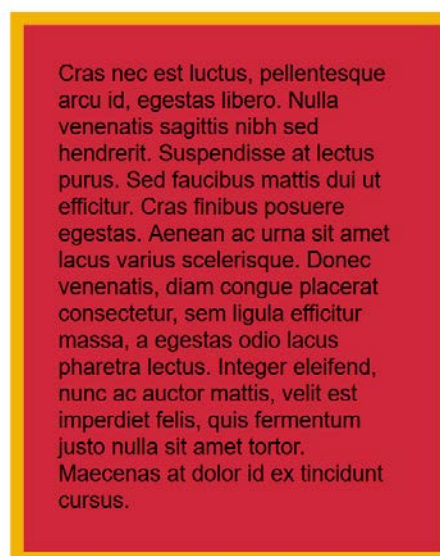


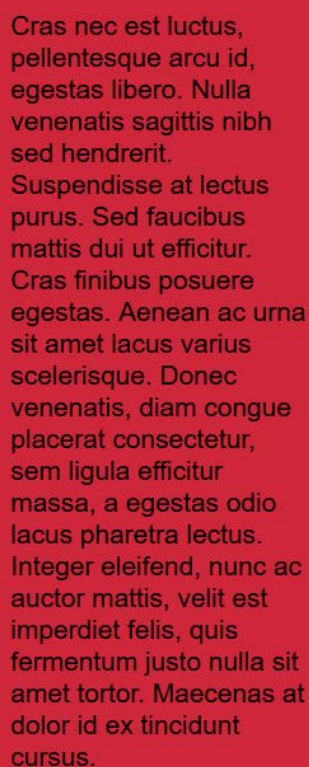
Fig. 69

Avec **box-sizing: border-box**, la largeur de la boîte CSS ne prend pas en compte les valeurs de padding et de border :

```
.border-box {  
  width: 250px;  
  border: 10px solid #da2;  
  margin: 25px;  
  padding: 25px;  
  background-color: #a23;  
  box-sizing: border-box;  
}
```

Fig.70

Ce qui nous donne :



Cras nec est luctus,
pellentesque arcu id,
egestas libero. Nulla
venenatis sagittis nibh
sed hendrerit.
Suspendisse at lectus
purus. Sed faucibus
mattis dui ut efficitur.
Cras finibus posuere
egestas. Aenean ac urna
sit amet lacus varius
scelerisque. Donec
venenatis, diam congue
placemat consectetur,
sem ligula efficitur
massa, a egestas odio
lacus pharetra lectus.
Integer eleifend, nunc ac
auctor mattis, velit est
imperdiet felis, quis
fermentum justo nulla sit
amet tortor. Maecenas at
dolor id ex tincidunt
cursus.

Fig.71

Avec **box-sizing:padding-box**, la largeur de la boîte CSS ne prend pas en compte la marge intérieure, le padding :

```
.padding-box {  
  width: 250px;  
  border: 10px solid #da2;  
  margin: 25px;  
  padding: 25px;  
  background-color: #a23;  
  box-sizing: padding-box;  
}
```

Fig.72

Le rendu :

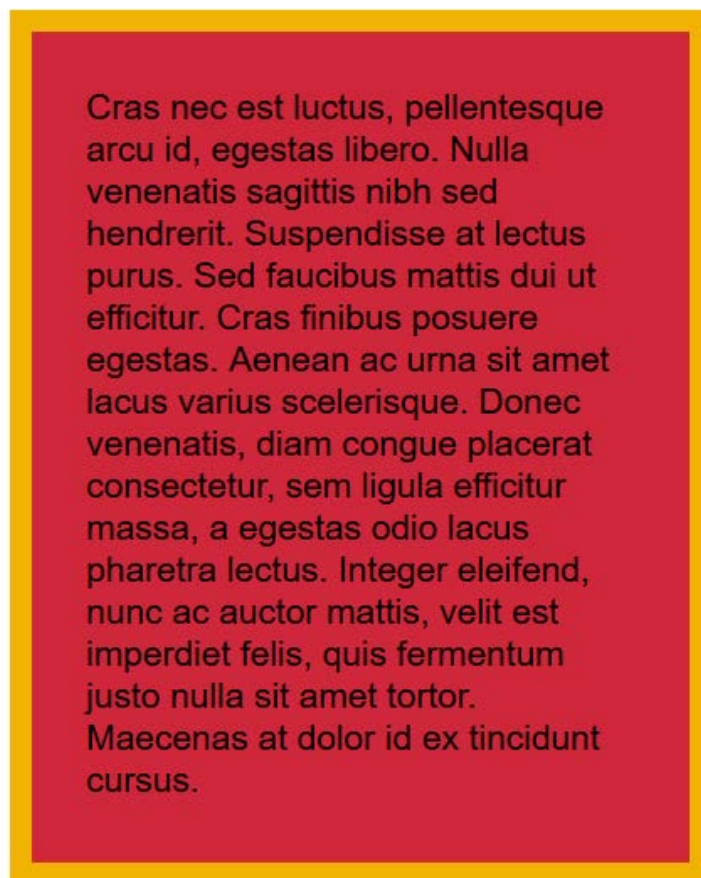


Fig.73

C. La propriété outline-offset

La modification du mode de calcul de la boîte CSS, s'il permet de simplifier certains calculs, ne résout pas forcément tous les problèmes. En particulier, l'usage des bordures peut se révéler surprenant. Avec **box-sizing: border-box** lors d'un calcul précis, si le container d'éléments dont la succession doit être égale à la largeur du parent possède un border, celle-ci prend sur la valeur width. Il arrive dès lors que l'un des enfants soit rejeté à la ligne.

Cet exemple nous montre que la bordure de boîte n'est pas aussi anodine qu'on y pense. Mais il existe la propriété **outline** qui forme une bordure non comprise par le modèle de boîte CSS.

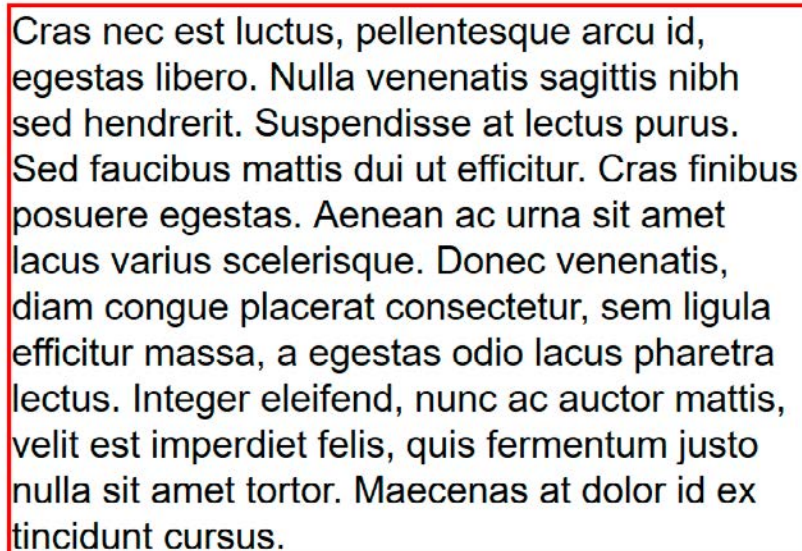
De plus, il est possible de spécifier avec **outline-offset** à quelle distance de la propriété border se situe cet « **outline** ».

On peut par exemple indiquer un contour extérieur à 10px de la bordure :

```
.outline
{
border:2px solid black;
outline:2px solid red; /* style de la bordure exterieure */
outline-offset:15px; /* marge entre les 2 bordures */
}
```

Fig.74

Rendu :



Cras nec est luctus, pellentesque arcu id, egestas libero. Nulla venenatis sagittis nibh sed hendrerit. Suspendisse at lectus purus. Sed faucibus mattis dui ut efficitur. Cras finibus posuere egestas. Aenean ac urna sit amet lacus varius scelerisque. Donec venenatis, diam congue placerat consectetur, sem ligula efficitur massa, a egestas odio lacus pharetra lectus. Integer eleifend, nunc ac auctor mattis, velit est imperdiet felis, quis fermentum justo nulla sit amet tortor. Maecenas at dolor id ex tincidunt cursus.

Fig.75