

# LES FONCTIONNALITÉS DU CCS3

# Table des matières



<b>I - Objectifs</b>	<b>4</b>
<b>II - Introduction</b>	<b>5</b>
<b>III - Les nouveaux sélecteurs</b>	<b>6</b>
1. Rappel .....	6
2. Les sélecteurs d'attributs .....	7
3. Le combinateur d'adjacence directe .....	9
4. Les pseudo-classes .....	9
5. Les pseudo-éléments .....	12
<b>IV - Évaluation 1</b>	<b>14</b>
<b>V - Les transformations et transitions</b>	<b>16</b>
1. Les transformations 2D .....	16
2. Les transformations 3D .....	16
3. Les transitions .....	17
<b>VI - Évaluation 2</b>	<b>18</b>
<b>VII - Les propriétés décoratives</b>	<b>20</b>
1. Les bords arrondis .....	20
2. Les ombres .....	20
3. Les types de dégradés .....	21
4. La transparence dans les couleurs .....	21
5. Les filtres .....	21
6. Les polices non-standard .....	22
<b>VIII - Évaluation 3</b>	<b>23</b>
<b>IX - Conclusion</b>	<b>25</b>





# *Objectifs*

- Utiliser les sélecteurs css3
- Utiliser les transformations, les transitions et les animations
- Utiliser les propriétés css3 décoratives

# Introduction



Les technologies CSS 3 facilitent notre quotidien de concepteur web plus que vous ne pourriez l'imaginer avec des techniques exploitables très séduisantes. Nous avons :

- *Les nouveaux sélecteurs CSS*: nth-child(odd), first-child ...
- *Les média queries* avec @media pour adapter l'affichage aux écrans
- *Nouveau support des polices* avec @font-face ...
- *Text wrapping* avec text-overflow;
- *Gestion de colonnes*: column-count, column-rule, column-gap
- *Bordures sur texte*: text-fill-color, text-stroke-color, text-stroke-width
- *Opacité* avec rgba
- *Saturation, luminosité* en CSS avec hsla
- *Bordures arrondies* avec border-radius
- *Fond dégradés* en CSS avec gradient
- *Ombres sur texte et sur box* avec text-shadow et box-shadow
- *Backgrounds améliorés*: Ajustement de la taille en fonction de la div, backgrounds multiples ...
- *Transitions* en CSS
- *Transformation et rotations* d'éléments HTML
- *Animations* en CSS
- Etc

Nous ne verrons pas toutes ces propriétés CSS dans le cadre de cette leçon. Certaines propriétés seront abordées dans d'autres leçons, et il reviendra à l'étudiant de s'informer et de découvrir le reste.

# Les nouveaux sélecteurs



Les sélecteurs permettent par le biais d'une "requête CSS" d'atteindre un ensemble de nœuds dans un document HTML et de lui donner un style.

L'univers de CSS 3 ne se limite pas à un ensemble de nouvelles propriétés : en effet, s'il est gratifiant d'appliquer des styles révolutionnaires à un élément, il est tout aussi utile de pouvoir le cibler au mieux. C'est dans cette optique que de nombreux nouveaux sélecteurs et pseudo éléments font leur apparition dans le CSS3.

## 1. Rappel

Voici un tableau sur la syntaxe des sélecteurs en CSS2

MOTIF	SIGNIFICATION
*	Correspond à tout élément.
E	Correspond à tout élément E (c.à.d., un élément de type E).
E F	Correspond à tout élément F qui est un descendant de l'élément E.
E > F	Correspond à tout élément F enfant direct de l'élément E.
E:first-child	Correspond à un élément E aussi le premier enfant de son élément parent.
E:link, E:visited	Correspond à un élément E qui est une ancre dans la source dont le lien n'a pas été visité (:link) ou bien l'a déjà été (:visited).
E:active, E:hover, E:focus	Correspond à l'élément E au cours de certaines actions de l'utilisateur.
E:lang(c)	Correspond à l'élément de type E qui emploie une langue c (la détermination de cette langue est spécifique au langage du document).
E + F	Correspond à tout élément F immédiatement précédé par un élément E.
E[foo]	Correspond à tout élément E avec l'attribut "foo" (quelles qu'en soient les valeurs).
E[foo="warning"]	Correspond à tout élément E dont l'attribut "foo" a exactement la valeur "warning".
E[foo~="warning"]	Correspond à tout élément E dont l'attribut "foo" a pour valeur une liste de valeurs séparées par des caractères blancs et dont une de celles-ci est "warning".
E[lang = "en"]	Correspond à tout élément E dont l'attribut "lang" a pour valeur une liste de valeurs séparées par des tirets, cette liste commençant (à gauche) par "en".
DIV.warning	Seulement en HTML. Identique à DIV[class~="warning"].
E#myid	Correspond à tout élément E dont l'ID est "myid".

Les nouveaux sélecteurs `css3` incluent entre autres les sélecteurs d'attributs, le combinateur d'adjacence directe, les pseudo-classes et les pseudo-éléments.

## 2. Les sélecteurs d'attributs

Il y a 3 nouveaux sélecteurs

- $[attr \hat{=} "stringValue"]$

Ce sélecteur permet de sélectionner un élément DOM dont l'attribut "attr" commence exactement par la valeur "stringValue".

*Exemple :*

```
p.example[title^="ess"]{
  color:#fff;
  background:#333;
}
<p class="example"> je n'ai pas d'attribut title</p>
<p class="example" title="comment">
j'ai un attribut title mais il ne commence pas par "ess"
</p>
<p class="example" title="essai">
j'ai un attribut title commençant par "ess"
</p>
<p class="example" title="esson">
j'ai un attribut title commençant par "ess" également
</p>
```

- *[attr\$="stringValue"]*

Ce sélecteur permet de sélectionner un élément DOM dont l'attribut "attr" finit exactement par la valeur "stringValue".

*Exemple :*

```
p.example2[title$="sai"]{
  color:#fff;
  background:#045FB4;
}
<p class="example2"> je n'ai pas d'attribut title</p>
<p class="example2" title="comment"> j'ai un attribut title mais il ne commence pas par
"ess"</p>
<p class="example2" title="essai"> j'ai un attribut title commençant par "ess"</p>
<p class="example2" title="esson"> j'ai un attribut title commençant par "ess" également</p>
```

- *[attr\*="stringValue"]*

Ce sélecteur permet de sélectionner un élément DOM dont l'attribut "attr" comporte au moins une fois la valeur "stringValue".

*Exemple :*

```
p.example3[title*="val"]{
  color:#fff;
  background:#990000;
}
<p class="example3"> je n'ai pas d'attribut title</p>
<p class="example3" title="comment"> j'ai un attribut title mais il ne contient pas "val"</p>
<p class="example3" title="val"> j'ai un attribut title contenant au moins "val"</p>
<p class="example3" title="evaluer"> j'ai un attribut title contenant au moins "val"
également</p>
<p class="example3" title="eval"> j'ai un attribut title contenant au moins "val" également</p>
```



### 3. Le combinateur d'adjacence directe

Permet d'ajouter un style à tous les éléments qui suivent un élément particulier.

*Exemple :*

```
.example4 div~p{
color:#fff;
margin:20px;
width:200px;
padding:5px;
border:1px solid #333;
background:#006644;
}
<div class="example4">
<div>je suis l'élément particulier div</div>
<p> je suis un p qui suit le div (l'élément particulier)</p>
<p>je suis un p qui suit le div (l'élément particulier)</p>
<span>je suis un span</p>
<p>je suis un p qui ne suit pas le div (l'élément particulier)</p>
</div>
```

### 4. Les pseudo-classes

Plusieurs pseudo-classes ont été ajoutées :

- *:root*

Ce sélecteur représente un élément qui est la racine d'un document. Par exemple, en HTML 4, l'élément est html.

- *:empty*

:empty fait référence à un élément vide de tout contenu (balise ou texte).

Ainsi,

<p></p> sera concerné par le sélecteur :empty, contrairement à <p><span></span></p>, <p>kiwi</p> ou encore <p>&nbsp;</p>.

Cette pseudo-classe trouve habituellement son utilité au sein de tableaux de données dynamiques où des cellules vides de contenu peuvent ainsi se distinguer des autres :

```
td:empty {
background-color: gray;
}
```

- *:nth-child(expression)*

Ce sélecteur permet de cibler tous les éléments en se basant sur leur position dans la liste des enfants de leur parent.

expression peut être un nombre, une expression numérique ou un mot clé tel que "odd" ou "even".

*Exemple :*

```
.exampleTable{
width:100%;
```

```

border:1px solid #444;
}
.exampleTable tr:nth-child(even){ //tous les enfants aux numéros pairs
background:#999999;
text-shadow: 2px 2px 5px #111;
color:#fff;
}
.exampleTable tr:nth-child(odd){ //tous les enfants aux numéros impairs
background:#990000;
color:#fff;
}
.exampleTable tr:nth-child(3n){ //tous les 3 enfants
background:#045FB4;
color:#fff;
}
.exampleTable tr:nth-child(7){ //l'enfant numéro 7
background:#006400;
text-shadow: 2px 2px 2px #fff;
color:#000;
}
<table class="exampleTable">
<tr>
<td>1ere ligne</td>
</tr>
<tr>
<td>2eme ligne</td>
</tr>
<tr>
<td>3eme ligne</td>
</tr>
<tr>
<td>4eme ligne</td>
</tr>
<tr>
<td>5ere ligne</td>
</tr>
</table>

```

- *:nth-last-child(expression)*

Ce sélecteur accepte les mêmes arguments que :nth-child() et correspond au dernier enfant d'un élément parent.

C'est le même principe que le :nth-child() sauf que l'on part de la fin.

*Exemple :*

```

.exampleTable2{
width:100%;
border:1px solid #444;
}
.exampleTable2 tr:nth-last-child(odd){ /*tous les enfants aux numéros impairs depuis la fin.*/
background:#990000;
color:#fff;
}
.exampleTable2 tr:nth-last-child(-n+2){ /*les 2 derniers enfants.*/
background:#045FB4;
color:#fff;
}
.exampleTable2 tr:nth-last-child(7){ /*l'enfant numéro 7 en partant de la fin donc la 2ème ligne du
tableau.*/
background:#006400;
text-shadow: 2px 2px 2px #fff;
color:#000;
}
<table class="exampleTable2">
<tr>
<td>1ere ligne</td>
</tr>
<tr>
<td>2eme ligne</td>
</tr>
<tr>
<td>3eme ligne</td>
</tr>
<tr>
<td>4eme ligne</td>
</tr>
<tr>
<td>5ere ligne</td>
</tr>
</table>

```

- *:last-child*

Correspond à :nth-last-child(1).

- *:first-child*

Correspond à :nth-child(1).

- *:nth-of-type(expression)*

Elle permet de cibler le/les n-ièmes enfants d'un élément d'un certain type.

*Exemple :*

Cible le/les h1 s'ils sont en deuxième position au sein de leur parent :

```
h1:nth-of-type(2){ ... }
```

- *:first-of-type*

Correspond à `:nth-of-type(1)`

`:first-of-type` représente un élément qui est le premier enfant de son type dans la liste des enfants de son élément parent.

- *:last-of-type*

Correspond à `:nth-last-of-type(1)`. Représente un élément qui est le dernier enfant de son type dans la liste des enfants de son élément parent.

- *:only-child*

Correspond à un élément qui n'a aucun frère. Cette pseudo-classe correspond à `:first-of-type:last-of-type` ou `:nth-of-type(1):nth-last-of-type(1)`.

- *:checked*

Correspond aux éléments cochés d'un formulaire.

- *:contains(value)*

Correspond aux éléments dont le contenu textuel contient la sous-chaine donnée en argument.

*Exemple :*

```
p:contains('essai'){
background:#900;
}
```

signifie que tous les éléments "p" contenant la sous chaine "essai" auront pour couleur d'arrière plan, la valeur '#900'.

L'usage de la pseudo-classe de contenu (`:contain`) est restreint aux médias statiques.

- *:not([expression])*

Représente un élément qui n'est pas représenté par l'expression donnée en argument.

*Exemple:*

```
button:not([DISABLED]){
...
}
a:not(:visited){
...
}
```

- *:target*

Représente un élément qui est la cible de l'URI.

*Exemple :*

```
p:target { background:#900}
```

Tout élément p qui sera la cible de l'URI (via l'ID # en tant que ancre) aura pour couleur d'arrière plan la valeur "#900".

## 5. Les pseudo-éléments

- *::first-line*

Applique la règle de style à la première ligne du texte de l'élément.

```
p::first-line { text-transform: uppercase }
```

La 1ère ligne de ou des éléments "p" est mise en majuscule.

- *::first-letter*

Applique la règle de style à la première lettre du texte de l'élément.

```
p::first-letter { font-size: 2em }
```

La 1ère lettre de ou des éléments "p" a une taille de police de 2em.

- *::selection*

Applique la règle de style à la sélection du texte de l'élément faite par l'utilisateur.

```
p::selection { background:#006644 }
```

A la sélection, le texte sélectionné aura une couleur d'arrière plan de valeur '#006644'.

- *::before* et *::after*

Génère un contenu avant ou après un contenu d'un élément.

Ces pseudo-éléments existent en CSS2.1 sous forme :before et :after.

# Évaluation 1



## Exercice 1

[Solution n°1 p 26]

Le sélecteur `[attr^="fruit"]` permet :

- ☐ de sélectionner les balises possédant un attribut "fruit"
- ☐ de sélectionner les balises dont l'attribut "attr" contient la valeur "fruit"
- ☐ de sélectionner les balises dont l'attribut "attr" commence exactement par la valeur "fruit"
- ☐ de sélectionner les balises dont l'attribut "attr" se termine par la valeur "fruit"
- ☐ de sélectionner les balises dont le nom est "attr" et l'identifiant est "fruit"

## Exercice 2

[Solution n°2 p 26]

le sélecteur `div[attr$="info"]` permet de sélectionner :

- ☐ toute balise dont l'attribut "attr" finit exactement par la valeur "info"
- ☐ toute balise div dont l'attribut "attr\$" égale à la valeur "info"
- ☐ toute balise div dont l'attribut "attr" contient exactement par la valeur "info"
- ☐ toute balise div dont l'attribut "attr" finit exactement par la valeur "info"
- ☐ toute balise div dont l'attribut "attr" commence exactement par la valeur "info"

## Exercice 3

[Solution n°3 p 26]

le sélecteur `div p` permet de :

- ☐ sélectionner toute balise p descendant d'une balise div
- ☐ sélectionner toute balise p enfant direct d'une balise div
- ☐ sélectionner toute balise div précédant une balise p
- ☐ sélectionner toute balise div parent ou ancêtre d'une balise p
- ☐ sélectionner toute balise div parent direct d'une balise p

## Exercice 4

---

*[Solution n°4 p 26]*

le sélecteur `li > a`, permet de sélectionner :

- ☐ toute les balises `li` et `a`
- ☐ toute balise `li` pères de la balise `a`
- ☐ toute balise `a` enfant direct d'une balise `li`
- ☐ toute balise `a` descendant d'une balise `li`
- ☐ tous les premiers lien `a`, enfant d'une balise `li`

## Exercice 5

---

*[Solution n°5 p 27]*

le sélecteur `h1 span:nth-child(2)` permet de sélectionner

- ☐ toute balise `h1` deuxième enfant d'une balise `span`
- ☐ toute balise `span` deuxième enfant d'une balise `h1`
- ☐ les 2 premières balises `span` enfants d'une balise `h1`
- ☐ les 2 dernières balises `span` enfant d'une balise `h1`
- ☐ les 2 premières balises `h1` et `span`

# Les transformations et transitions



## 1. Les transformations 2D

Elle permet de transformer un élément dans un espace en deux dimensions.

Il existe plusieurs types de transformation CSS 2D :

- `rotate(angle)` permet d'appliquer une rotation.
- `scale(x,y)`, `scaleX(x)` et `scaleY(y)` permettent de mettre à l'échelle.
- `skewX(angle)` et `skewY(angle)` permettent d'incliner un élément.
- `translate(x,y)`, `translateX(x)` et `translateY(y)` permettent d'appliquer une translation.
- `matrix(a, c, b, d, x, y)` spécifie une matrice de transformation.

*La syntaxe :*

```
transform: rotate(50deg);
```

```
transform: rotate(50deg) skewX(30deg) translate(100px);
```

*Axe de transformation*

Par défaut, une transformation s'applique par rapport au centre de l'élément. Pour modifier cette valeur, utilisez `transform-origin: x y`.

```
transform-origin: 30px top;
```

Dans cet exemple, le centre de transformation sera en haut et à 30px de la gauche.

*Support des navigateurs*

Actuellement, pour utiliser ces propriétés dans les navigateurs, vous devrez utiliser les préfixes propriétaires `-webkit-`, `-moz-`, `-ms-` et `-o-` :

```
-webkit-transform: rotate(50deg);
```

```
-moz-transform: rotate(50deg);
```

```
-ms-transform: rotate(50deg);
```

```
-o-transform: rotate(50deg);
```

```
transform: rotate(50deg);
```

## 2. Les transformations 3D

Elle permet de transformer un élément dans un espace en trois dimensions.

Les transformations 3D sont basées sur la même propriété que les transformations 2D, à savoir `transform`.

En revanche, de nouvelles fonctions de transformations sont disponibles :

- `rotateX(angle)`, `rotateY(angle)` et `rotateZ(angle)` pour des rotations autour de chaque axe.
- `rotate3d(x,y,z,angle)` pour une rotation autour du vecteur 3D `x,y,z`.
- `translateZ(z)` pour des translations sur l'axe Z (profondeur).
- `translate3d(x,y,z)` pour des translations en 3D.
- `scaleZ(z)` pour des changements d'échelle sur l'axe Z



- `scale3d(x,y,z)` pour des changements d'échelle en 3D.
- `matrix3d()` spécifie une matrice de transformation 4x4.

Néanmoins, de nouvelles propriétés existent pour gérer cet affichage en 3D :

- `perspective` représente la distance entre la caméra et la scène (l'internaute et l'écran). Sans `perspective`, l'impression de 3D n'existe pas.
- `perspective-origin` pour l'origine de la perspective (Là où les points de fuite des éléments se rejoignent).
- `transform-style: flat|preserve-3d` défini si les éléments inclus subissent les transformations en 3D.
- `backface-visibility: visible|hidden` spécifie si les faces arrières des éléments sont visibles ou non.

### 3. Les transitions

C'est quoi une transition en CSS ?

Les Transitions permettent de modifier les valeurs de propriétés CSS dans le temps, d'un état A à un état B. Par exemple, il est possible de modifier la taille d'un élément progressivement.

*Utilisation des transitions*

La propriété se décompose en plusieurs autres propriétés, décrite ici :

- `transition-property` : la(les) propriété(s) CSS à modifier.
- `transition-duration` : le temps total de la transition ( en secondes, millisecondes).
- `transition-timing-function` : la méthode d'interpolation (accélération, décélération)
- `transition-delay` : le temps avant que la transition ne démarre.

Il existe aussi la propriété raccourcie `transition`.

Voici un exemple de transition qui modifie la taille de l'élément. Celle-ci dure 2 secondes et utilise la méthode `ease`, c'est à dire avec une accélération naturelle.

```
transition: width 2s ease;
```

*Le support dans les navigateurs*

Pour pouvoir les utiliser les transitions sous tous les navigateurs, il faut préfixer toutes les propriétés avec `-webkit`, `-moz-`, `-ms-` et `-o-`, ce qui nous donne :

```
-webkit-transition: width 2s ease ;
```

```
-moz-transition: width 2s ease ;
```

```
-ms-transition: width 2s ease ;
```

```
-o-transition: width 2s ease ;
```

```
transition: width 2s ease ;
```

*Remarque :*

Il existe aussi les animations, mais nous ne les verrons pas dans le cadre de cette leçon.

# Évaluation 2

IV

## Exercice 1

*[Solution n°6 p 27]*

Quatre fonctions de transformation en css sont :

- ☐ scale
- ☐ matrix\_power
- ☐ rotate
- ☐ animation
- ☐ skew
- ☐ translate
- ☐ moving
- ☐ 3dmax
- ☐ runaway
- ☐ transX

## Exercice 2

*[Solution n°7 p 27]*

Le code suivant :

```
transform: rotate(50deg) skewX(30deg) translateY(100px);
```

permet de :

- ☐ faire une rotation de 50deg et une inclinaison de 30deg selon l'axe Y, puis une translation de 100px selon l'axe Y
- ☐ faire une rotation de 25deg et une inclinaison de 30deg selon l'axe X, puis une translation de 100px selon l'axe X
- ☐ faire une rotation de 50deg et une inclinaison de 30deg selon l'axe X, puis une translation de 100px selon l'axe Y
- ☐ faire une rotation de 25deg et une inclinaison de 30deg selon l'axe X, puis une translation de 50px selon l'axe Y et X

## Exercice 3

[Solution n°8 p 27]

la propriété :

*transform-origin: 50px top;*

permet :

- ☐ d'appliquer une transformation sur un élément vers le bas et à 50px de la gauche
- ☐ d'appliquer une transformation sur un élément vers le haut et à 50px de la gauche
- ☐ d'appliquer une transformation sur un élément vers le bas et à 50px de la droite
- ☐ d'appliquer une transformation sur un élément vers le haut et à 50px de la droite

## Exercice 4

[Solution n°9 p 28]

Les Transitions permettent de :

- ☐ faire des animation et effets spéciaux en css
- ☐ modifier progressivement dans le temps les valeurs de propriétés CSS du balise lorsque l'utilisateur clique sur cette dernière
- ☐ marquer une pause lorsque les propriétés css d'une balise sont modifiées
- ☐ modifier progressivement dans le temps les valeurs de propriétés CSS du balise , d'un état A à un état B
- ☐ passer d'une balise à une autre pendant la navigation

## Exercice 5

[Solution n°10 p 28]

la propriété transition se décompose comme suit :

- ☐ tansion-prop
- ☐ tansion-property
- ☐ transition-duration
- ☐ transition-time
- ☐ transition-timing-function
- ☐ transition-function
- ☐ transition-value
- ☐ transition-delay
- ☐ transition-delayed
- ☐ transition-run

# Les propriétés décoratives



## 1. Les bords arrondis

La propriété `border-radius` permet de créer des coins arrondis sur un bloc. Cette propriété s'utilise de la façon suivante :

`border-radius: 5px; border-radius: 5px 10px; border-radius: 10px 0px 0px 10px;`

Il existe plusieurs façons d'utiliser cette règle, à la manière des marges internes et externes (`padding` et `margin`) :

- Une valeur : les 4 coins sont arrondis avec cette valeur.
- Deux valeurs : la première valeur arrondit le coin haut-gauche et bas-droit, la seconde arrondit le coin haut-droit et bas-gauche.
- Quatre valeurs : chaque coin a son propre arrondi.

## 2. Les ombres

La propriété `box-shadow` permet de créer des ombres sur un bloc. Cette propriété s'utilise de la façon suivante :

`box-shadow: 1px 1px 5px 2px black;`

Où les 5 valeurs correspondent, dans l'ordre :

- Décalage de l'ombre vers la droite
- Décalage de l'ombre vers le bas
- Valeur du flou de l'ombre
- Valeur de l'étendue de l'ombre
- *Couleur de l'ombre*

Si une valeur est omise, la valeur par défaut est de 0px. Il existe également le mot-clé `inset`, qui permet de créer une ombre interne. Il est possible d'ajouter des ombres multiples en les séparant par une virgule. Les ombres s'appliquent de haut en bas.

`box-shadow: 0px 0px 2px white, 2px 2px 5px 4px black inset;`

La propriété `text-shadow` permet de créer des ombres sur un texte. Cette propriété s'utilise de la façon suivante :

`text-shadow: 2px 2px 2px black;`

Où les 4 valeurs correspondent, dans l'ordre :

- Décalage de l'ombre vers la droite
- Décalage de l'ombre vers le bas
- Valeur du flou de l'ombre
- Couleur de l'ombre

Il est possible d'ajouter des ombres multiples en les séparant par une virgule. Les ombres s'appliquent alors de haut en bas.

`text-shadow: 2px 2px 2px black, -2px -2px 2px white;`



### 3. Les types de dégradés

Il existe 4 types de dégradés, qui sont les suivants :

- linear-gradient permet de créer un dégradé linéaire.
- radial-gradient permet de créer un dégradé radial.
- repeating-linear-gradient permet de créer un dégradé linéaire avec répétition.
- repeating-radial-gradient permet de créer un dégradé radial avec répétition.

*La syntaxe du W3C*

```
background: linear-gradient( to top, red, green);
background: linear-gradient( to top right, #000, #FFF);
background: radial-gradient( ellipse at top left, red, green);
background: radial-gradient( closest-corner at center, #000, #FFF);
```

*L'implémentation dans les navigateurs*

La syntaxe des dégradés doit être préfixée pour être reconnue sous tous les navigateurs.

```
background: -webkit-linear-gradient( ... );
background: -moz-linear-gradient( ... );
background: -ms-linear-gradient( ... );
background: -o-linear-gradient( ... );
background: linear-gradient( ... ); }
```

### 4. La transparence dans les couleurs

rgba() n'est pas une propriété CSS, mais une fonction. Elle s'utilise sur toutes les propriétés qui spécifient une couleur, comme par exemple :

- background
- color
- border-color
- ...

*La syntaxe*

Comme pour la fonction rgb(), rgba() utilise les valeurs rouge, vert et bleu d'une couleur, et ajoute l'opacité.

L'exemple suivant affecte donc la couleur Rouge:156, Vert:25, Bleu:26 et une valeur de transparence de 80%.

```
background: rgba( 156, 25, 26, 0.8);
```

*Note* : Contrairement à opacity qui rend transparent un élément et tous ses enfants, rgba() applique cette transparence uniquement à la couleur appliquée.

### 5. Les filtres

Nous avons 3 types de filtres :

- grayscale(valeur) : niveau de gris
- sepia(valeur) : effet sépia
- saturate(valeur) : saturation
- hue-rotate(angle) : teinte
- invert(valeur) : effet négatif
- opacity(valeur) : opacité
- brightness(valeur) : luminosité
- contrast(valeur) : contraste
- blur(valeur) : effet de flou
- drop-shadow(ombre) : ombre portée

pour appliquer les filtres on utilise la propriété `filter`, qui peut être préfixée pour être reconnue sur tous les navigateurs.

#### *Exemple*

Un effet de flou

```
div{ -webkit-filter: blur(5px);
-moz-filter: blur(5px);
-ms-filter: blur(5px);
-o-filter: blur(5px);
filter: blur(5px); }
```

## 6. Les polices non-standard

Il est possible avec le css 3 de télécharger ou de référencer une police et de l'intégrer dans une page afin de l'utiliser.

La règle `@font-face` permet au développeur web d'utiliser une police "exotique", et ainsi de s'affranchir des polices uniquement disponibles sur l'environnement du client.

#### *La syntaxe*

```
@font-face{ <Description de la police> }
```

Pour décrire la police utilisée, on utilise différents "descripteurs", tel que :

- `font-family` : pour le nom de la police.
- `src` : pour le chemin de la police (URI).
- `font-style`, `font-weight` et `font-stretch` : pour les propriétés de la police.

Ce qui nous donne :

```
@font-face{
font-family: "nomPolice";
src: url("polices/maPolice.ttf");
font-weight: bold;
}
```

#### *Et dans les navigateurs ?*

Tous les navigateurs connaissent `@font-face`. Cependant, tous n'utilisent pas le même format de police. Voici un récapitulatif (non exhaustif) :

- Firefox : .ttf, .woff
- Chrome : .ttf, .woff, .svg
- Safari : .ttf, .svg
- Opéra : .ttf, .svg
- Internet Explorer : .eot
- iPhone,iPad : .svg

Il faut donc spécifier plusieurs sources, sachant qu'à partir du moment où une police est trouvée, le navigateur ne cherche pas plus loin :

```
@font-face{
font-family: "nomPolice";
src: url("polices/maPolice.eot?"), /* pour IE */ url("polices/maPolice.woff"), /* pour Firefox,
Chrome*/ url("polices/maPolice.ttf"), /* pour Safari et Opéra*/ url("polices/maPolice.svg"); /*
pour iPhone, iPad */
font-weight: normal;
}
```

# Évaluation 3

VI

## Exercice 1

---

*[Solution n°11 p 28]*

*border-radius: 5px;*

- ☐ permet de créer des bordures arrondies de 5px aux quatre coins d'une balise
- ☐ permet de créer une bordure arrondie de 5px au coin haut et gauche d'une balise
- ☐ permet de créer des bords de 5px autour d'une balise
- ☐ permet de créer des bordures douces de 5px aux quatre coins d'une balise
- ☐ permet de créer une bordure arrondie de 5px en haut d'une balise

## Exercice 2

---

*[Solution n°12 p 28]*

*pour créer des ombres autour d'une balise, on utilise la propriété :*

- ☐ box-shadowing
- ☐ box-shading
- ☐ box-shade
- ☐ shadow
- ☐ box-shadow

## Exercice 3

---

*[Solution n°13 p 29]*

*Deux propriétés css permettant de créer des dégradés sont :*

- ☐ linear-gradient
- ☐ line-grade
- ☐ liner-gradient
- ☐ radial-grade
- ☐ radial-gradient
- ☐ radial-gradient

## Exercice 4

---

*[Solution n°14 p 29]*

Quatre fonctions pour appliquer un filtre sur une image ou bloc en css3 sont :

- ☐ bluring
- ☐ gray
- ☐ darkness
- ☐ flou\_gaussien
- ☐ grayscale
- ☐ sepia
- ☐ shadow
- ☐ brightness
- ☐ light
- ☐ blur

## Exercice 5

---

*[Solution n°15 p 29]*

pour importer une police particulière en css et l'appliquer à un texte, on utilise :

- ☐ @face-font
- ☐ @font
- ☐ @font-face
- ☐ font-family
- ☐ font-name



# Conclusion

## *RÉFÉRENCE :*

<http://www.css3create.com/>

<https://debray-jerome.developpez.com/articles/les-transformations-en-css3/>

<https://debray-jerome.developpez.com/articles/les-selecteurs-en-css3/>

<https://debray-jerome.developpez.com/articles/les-transformations-3d-en-css3/>

<https://openclassrooms.com/courses/animez-votre-site-avec-css3>

CSS avancées vers html5 et css 3 de Raphael Goetter

# Solutions des exercices

## > Solution n°1

*Exercice p. 14*

- ☐ de sélectionner les balises possédant un attribut "fruit"
- ☐ de sélectionner les balises dont l'attribut "attr" contient la valeur "fruit"
- ☒ de sélectionner les balises dont l'attribut "attr" commence exactement par la valeur "fruit"
- ☐ de sélectionner les balises dont l'attribut "attr" se termine par la valeur "fruit"
- ☐ de sélectionner les balises dont le nom est "attr" et l'identifiant est "fruit"

## > Solution n°2

*Exercice p. 14*

- ☐ toute balise dont l'attribut "attr" finit exactement par la valeur "info"
- ☐ toute balise div dont l'attribut "attr\$" égale à la valeur "info"
- ☐ toute balise div dont l'attribut "attr" contient exactement par la valeur "info"
- ☒ toute balise div dont l'attribut "attr" finit exactement par la valeur "info"
- ☐ toute balise div dont l'attribut "attr" commence exactement par la valeur "info"

## > Solution n°3

*Exercice p. 14*

- ☒ sélectionner toute balise p descendant d'une balise div
- ☐ sélectionner toute balise p enfant direct d'une balise div
- ☐ sélectionner toute balise div précédant une balise p
- ☐ sélectionner toute balise div parent ou ancêtre d'une balise p
- ☐ sélectionner toute balise div parent direct d'une balise p

## > Solution n°4

*Exercice p. 15*

- ☐ toute les balises li et a
- ☐ toute balise li pères de la balise a
- ☒ toute balise a enfant direct d'une balise li
- ☐ toute balise a descendant d'une balise li

- ☐ tous les premiers lien a, enfant d'une balise li

### > Solution n°5

Exercice p. 15

- ☐ toute balise h1 deuxième enfant d'une balise span
- ☒ toute balise span deuxième enfant d'une balise h1
- ☐ les 2 premières balises span enfants d'une balise h1
- ☐ les 2 dernières balises span enfant d'une balise h1
- ☐ les 2 premières balises h1 et span

### > Solution n°6

Exercice p. 18

- ☒ scale
- ☐ matrix\_power
- ☒ rotate
- ☐ animation
- ☒ skew
- ☒ translate
- ☐ moving
- ☐ 3dmax
- ☐ runaway
- ☐ transX

### > Solution n°7

Exercice p. 18

- ☐ faire une rotation de 50deg et une inclinaison de 30deg selon l'axe Y, puis une translation de 100px selon l'axe Y
- ☐ faire une rotation de 25deg et une inclinaison de 30deg selon l'axe X, puis une translation de 100px selon l'axe X
- ☒ faire une rotation de 50deg et une inclinaison de 30deg selon l'axe X, puis une translation de 100px selon l'axe Y
- ☐ faire une rotation de 25deg et une inclinaison de 30deg selon l'axe X, puis une translation de 50px selon l'axe Y et X

### > Solution n°8

Exercice p. 19

- ☐ d'appliquer une transformation sur un élément vers le bas et à 50px de la gauche
- ☒ d'appliquer une transformation sur un élément vers le haut et à 50px de la gauche

- ☐ d'appliquer une transformation sur un élément vers le bas et à 50px de la droite
- ☐ d'appliquer une transformation sur un élément vers le haut et à 50px de la droite

### > Solution n°9

*Exercice p. 19*

- ☐ faire des animation et effets spéciaux en css
- ☐ modifier progressivement dans le temps les valeurs de propriétés CSS du balise lorsque l'utilisateur clique sur cette dernière
- ☐ marquer une pause lorsque les propriétés css d'une balise sont modifiées
- ☒ modifier progressivement dans le temps les valeurs de propriétés CSS du balise , d'un état A à un état B
- ☐ passer d'une balise à une autre pendant la navigation

### > Solution n°10

*Exercice p. 19*

- ☐ tansion-prop
- ☒ tansion-property
- ☒ transition-duration
- ☐ transition-time
- ☒ transition-timing-function
- ☐ transition-function
- ☐ transition-value
- ☒ transition-delay
- ☐ transition-delayed
- ☐ transition-run

### > Solution n°11

*Exercice p. 23*

- ☒ permet de créer des bordures arrondies de 5px aux quatre coins d'une balise
- ☐ permet de créer une bordure arrondie de 5px au coin haut et gauche d'une balise
- ☐ permet de créer des bords de 5px autour d'une balise
- ☐ permet de créer des bordures douces de 5px aux quatre coins d'une balise
- ☐ permet de créer une bordure arrondie de 5px en haut d'une balise

### > Solution n°12

*Exercice p. 23*

- ☐ box-shadowing

