

Chapitre 3

Propriétés CSS étendues (texte et graphique)

Objectifs du chapitre :

- Les préfixes propres aux navigateurs (moteur de rendu, préfixeurs CSS)
- Les propriétés de texte et de contenu étendues proposées par le CSS3 : *importer des polices de caractères, les césures, word-wrap, font-strech, font-size-adjust, font-feature-settings, resize, text-shadow, ...*
- Gérer les images d'arrière-plan et les propriétés graphiques fournies par le CSS3

1. PREFIXES PROPRES AUX NAVIGATEURS

1.1. Le moteur de rendu

Afin de permettre le développement progressif et l'implémentation des modules CSS, les éditeurs de navigateurs ont la possibilité d'introduire le support de propriétés spécifiques. Ces "propriétés propriétaires" doivent être préfixées d'un tiret et d'un code correspondant au moteur de rendu les exploitant.

Le cœur des navigateurs (desktop ou mobile) est le moteur de rendu, il est déterminant en ce qui concerne les CSS, il est responsable de l'affichage de la page. Ainsi, deux navigateurs implémentant la même version du même moteur afficheront la même chose.

Aujourd'hui, quatre moteurs de rendus permettent un rendu correct :

Navigateur	Moteur de rendu	Préfixe
Internet Explorer	Trident	-ms-
Firefox	Gecko	-moz-
Chrome	Webkit	-webkit-
Safari	Webkit	-webkit-
Opera ¹	Presto	-o-
Safari Mobile (Ipad, iPhone, Ipod)	Webkit	-webkit-
Android	Webkit	-webkit-

Statistiques de navigation web : <http://gs.statcounter.com/>, <http://www.w3schools.com/browsers/default.asp>

1.2. Les préfixes des moteurs de rendu

Chaque navigateur possède son propre jeu de fonctionnalités spéciales auquel vous pouvez accéder avec son préfixe (disponibles sur le site de développement des éditeurs). **Ajouter un préfixe dans votre sélecteur propre au navigateur ciblé permet de ne pas perturber d'autres agents utilisateur (navigateurs) ou invalider votre code.**

Grâce à ce mécanisme, vous pouvez tester de nouvelles fonctionnalités CSS avant qu'elles ne fassent partie de la recommandation officielle. Dans la pratique, cela nous contraint à multiplier les écritures de déclarations de propriétés pour parvenir à un accord sur tous les navigateurs mais cela évite les collisions de noms entre les fonctions standards et propriétaires :

```
-webkit-transform :rotateX(45deg) ; /* comprise par Chrome, Safari, safari mobile et Android */  
-moz-transform :rotateX(45deg) ; /*comprise par Firefox */  
-ms-transform :rotateX(45deg) ; /* comprise par I Explorer */  
-o-transform :rotateX(45deg) ; /* comprise par Opera */  
transform :rotateX(45deg) ; /* on termine par l'écriture dite nommée */
```

Dans certains cas, les propriétés n'ont pas besoin d'être toutes écrites (ex :box-shadow) car certain navigateurs n'ont jamais supporté certaines propriétés en version préfixée. **Il est conseillé de toujours terminer par la version non préfixée car les navigateurs compatibles avec les deux écritures utiliseront la dernière de la liste en suivant le principe de la cascade CSS.**

1.3. Préfixeurs CSS

Outils en ligne dans lequel nous collons notre code finalisé, préfixé ou non et celui-ci nous renvoie une version préfixée (agrémentée de tous les préfixes propriétaires nécessaires pour tous les navigateurs). Il suffit de remplacer notre code par celui délivré par l'outil.

Exemples : <http://prefixmycss.com/>, <http://prefixr.com>

¹ Opera 15+ et dérivé interprète certaines propriétés CSS3 avec le préfixe -webkit-

1.4. Utiliser `-prefix-free` (<http://leaverou.github.io/prefixfree/>)

`-prefix-free` est un petit fichier JavaScript (2 Ko) développé par Lea Verou, développeuse Web impliquée dans les standards Open Source, CSS et JavaScript.

Le rôle de ce script est d'ajouter automatiquement des préfixes propriétaires nécessaires à la reconnaissance de certaines propriétés de la spécification CSS3 et 4 par les différents navigateurs modernes.

Concrètement, `-prefix-free` adapte la propriété CSS au navigateur en ajoutant à la volée le bon préfixe propriétaire selon le navigateur affichant la page.

Ses avantages sont un gain de lignes de code, un gain de temps et un code plus lisible.

Comment faire fonctionner `-prefix-free` ?

Téléchargez le fichier javascript et placez-le sur votre serveur d'hébergement et insérez dans la partie `<head>` de vos pages HTML:

```
<script type="text/javascript" src="chemin/vers/prefixfree.min.js"></script>
```

Il est également possible d'appeler le fichier directement à partir de Github :

```
<script type="text/javascript" src="https://raw.githubusercontent.com/LeaVerou/prefixfree/master/prefixfree.min.js"></script>
```

Pour minimiser l'effet FOUC - c'est un phénomène décrit par bluerobot.com qui se traduit par l'affichage de la page brute avant celui de la page stylée - il est recommandé de placer la ligne d'appel du fichier JS juste après la déclaration de la CSS.

Lecture Web :

Page de démonstration : <http://leaverou.github.io/prefixfree/#test-drive>

Plus d'infos sur cet outil : <http://www.alsacreations.com/article/lire/1306-prefix-free-prefixes-CSS3.html>

2. PROPRIETES DE TEXTE CSS3 (ET CSS4)

2.1. Importer des polices de caractères (CSS3)

Si l'ordinateur du visiteur ne dispose pas de la police souhaitée par le site web, on peut imaginer de la lui fournir. Le CSS 3 propose le **téléchargement automatique par le navigateur de la police utilisée** dans les pages. Toutefois, il faut posséder la police mise en oeuvre car elles tombent dans les **règles des droits d'auteur**.

Les différents formats de polices

La police de caractère, ou plutôt les différents formats de la police de caractère doivent être placés sur le serveur.

- **.ttf** : TrueType Font
- **.otf** : OpenType Font
- **.eot** : Embedded OpenType (propriétaire Microsoft, pour IE Explorer),
- **.svg, .svgz** : SVG Font (Iphone, smartphone)
- **.woff** : Web Open Font Format

Lecture Web :

Le site www.fontsquirrel.com crée pour vous des kits@font-face « prêt à l'emploi » contenant tous ces formats de police téléchargeables ainsi que deux fichiers permettant de copier-coller rapidement la syntaxe de @font-face dans nos pages.

Déclaration de base de@font-face

Comme pour une image de fond, elle sera chargée au moment de l'appel de la feuille de style dans laquelle est déclarée @font-face.

Cette déclaration est définie par une **@règle** et est composée de deux propriétés minimum : `font-family` permettant de nommer la police (nom ensuite employé pour l'appliquer aux textes) et `src` permettant de spécifier l'adresse de la font à télécharger.

```
@font-face {
  font-family: 'anudawitalic';
  src: url('ANUDI-webfont.eot');
  src: url('ANUDI-webfont.eot?#iefix') format('embedded-opentype'),
        url('ANUDI-webfont.woff') format('woff'),
        url('ANUDI-webfont.ttf') format('truetype'),
        url('ANUDI-webfont.svg#anudawitalic') format('svg');
  font-weight: normal;
  font-style: normal;
}
```

Utiliser la police déclarée

Il suffit ensuite de déclarer le nom de cette police dans la liste des valeurs de la propriété **font-family** d'un élément :

```
h1 {
  font-family: "anudawitalic", Arial, Helvetica, sans-serif;
}
```

Optimiser le téléchargement

Attention : il est essentiel de *préciser le format de la police* afin d'optimiser le chargement. En effet, un navigateur ne téléchargera pas un format de police qu'il ne supporte pas.

La succession des déclarations s'ordonne en fonction du poids des polices : *de la plus légère vers la plus lourde* (pour des raisons de performance).

Le format **WOFF**, spécialement pensé et conçu pour le web, est à présent supporté par tous les navigateurs récents.

Lecture Web :

FontSquirrel fournit les différents formats, mais il fournit aussi le code CSS qui va bien. Après avoir reclassé les formats par poids croissant.

<http://typographisme.net/post/Bonnes-pratiques-pour-les-déclarations-@font-face>

<http://typographisme.net/post/Les-formats-de-polices-typographiques-pour-le-Web>

Exploiter les variantes

Il est possible de forcer le navigateur à utiliser les polices de substitution adéquate pour les différentes variantes de police (gras, italique). Si rien n'est précisé, les navigateurs vont simuler ces variantes et provoquer parfois des rendus disgracieux.

Il est possible dans certains cas de télécharger plusieurs types d'une même police et de préciser pour celle-ci les règles `font-weight`, `font-style` ou `font-variant`.

```
@font-face {
  font-family: 'quadranta';
  src: url('quadranta.woff') format('woff'),
        url('quadranta.otf') format('truetype'),
        url('quadranta.eot?#iefix') format('embedded-opentype'),
        url('quadranta.svg#QuadrantaBold') format('svg');
  font-weight: normal;
  font-style: normal;
}
@font-face {
  font-family: 'quadranta';
  src: url('quadranta_bold.woff') format('woff'),
        url('quadranta_bold.otf') format('truetype'),
        url('quadranta_bold.eot?#iefix') format('embedded-opentype'),
```

```
url('quadranta_bold.svg#QuadrantaBold') format('svg');  
font-weight:bold;  
font-style:normal;  
}
```

L'appel à cette police se fera ensuite en précisant `font-weight:bold`; ce qui forcera le navigateur à récupérer la version gras de la police au lieu de la simuler.

Le « cas » Internet Explorer

Internet Explorer interprète exclusivement le format EOT (d'ailleurs reconnu par aucun autre navigateur) **jusqu'à sa version 8**.

De plus, dans sa version de 6 à 8, il ne comprend pas la syntaxe CSS3 à base de déclarations multiples séparées par des virgules (parsage), utilisée pour la propriété `src` et renvoie un message d'erreur. La solution de contournement¹ pour corriger le problème d'Internet Explorer consiste à placer un point d'interrogation (?) juste après le nom du fichier EOT et ainsi lui faire croire que ce qui suit est une *query string*² et télécharger le fichier sans soucis.

Et pour Internet Explorer 9.0 et suite ...

Afin de profiter du support de la police au format WOFF, il est possible de « sauter » l'appel au format EOT. Pour ce faire, modifiez le nom du format de la police reconnue par les autres versions d'IE (EOT), de manière à ce qu'il ne le reconnaisse pas et teste alors le format suivant (WOFF) et le télécharge.

Un dernier problème subsiste lorsqu'Explorer 9.0 fonctionne en mode compatibilité – pour pallier à cela, une balise meta empêche le navigateur de basculer en ce mode là :

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

@font-face generator

<http://www.fontsquirrel.com/tools/webfont-generator>

Propose de créer nos propres kits à partir d'une police de notre création ou d'une police libre de droit.

Les sites de fonts

Il existe des services qui **fournissent des polices à intégrer directement dans son CSS depuis une URL distante**. Les polices proposées sont en général adaptées à l'affichage dans un navigateur, s'afficheront correctement sur la majorité des navigateurs, et enfin le service se chargera de fournir la police à l'utilisateur, ce qui peut s'avérer non négligeable étant donné que la plupart des polices font entre 50ko et 300ko !

Google Web Fonts (www.google.com/webfonts) propose d'héberger sur leurs serveurs les polices employées. Lors de l'appel du code généré par Google et intégré dans la CSS de notre site, celui-ci se charge de vérifier quel format de fonte est adapté au navigateur et ainsi ne faire télécharger que ce fichier par l'utilisateur.

Le choix d'une ou plusieurs polices pour notre site se fait ainsi en 3 étapes :

1. Choisir la police de caractère
2. Comparer et vérifier le contenu
3. Obtenir le code (>> *Quick use*)

La méthode d'importation peut être :

1. Via une balise `link`
2. Via la règle `@import`
3. Via Javascript

Ici, il n'y a donc pas de `@font-face` à écrire. Tout est géré dans la feuille de style hébergée chez Google, qui va effectuer tous les traitements de son côté afin d'assurer l'affichage correct de la police.

L'utilisation de l'API se résume donc à :

- Inclure la police qui nous intéresse
- L'appliquer à une balise ou classe CSS avec `font-family`

¹ Astuce mise au point par Ethan Dunham

² Une chaîne de caractère qui permet de passer des données de type clé/valeur dans l'URL.

Lecture Web :

<http://typekit.com>, <http://new.myfonts.com>

2.2. Word-wrap

Il est possible de forcer le retour à la ligne d'un mot long à l'aide de la propriété CSS3 `word-wrap`¹, appliquée au parent et qui aura pour effet de couper le mot à un endroit arbitraire afin de provoquer un retour à la ligne.

2.3. Les césures (hyphénation) avec `hyphens`

La propriété `hyphens` n'est supportée que par quelques navigateurs actuellement (elle nécessite d'ailleurs des préfixes vendeurs), elle va plus loin que `word-wrap` puisque la césure est "intelligente" (elle s'adapte aux règles typographiques de la langue employée) et affiche des traits d'union.

```
.bloc {  
  -webkit-hyphens: auto;  
  -moz-hyphens: auto;  
  -ms-hyphens: auto;  
  hyphens: auto;  
}
```

2.3.1. La propriété : `hyphenate-limit-lines` (CSS4)

Elle permet de définir le nombre maximum de tiret de césure dans une phrase ou une ligne

```
P {  
  max-width : 300px;  
  -webkit-hyphens : auto;  
  -ms-hyphens : auto;  
  hyphens : auto;  
  hyphenate-limit-lines : 2;  
  color : black;  
}
```

2.3.2. La propriété : `hyphenate-limit-chars` (CSS4)

Elle permet de définir le nombre de caractère minimum des mots qui vont subir la césure, le nombre de caractères minimum avant le tiret et le nombre de caractères minimum après le tiret. Cela correspond respectivement au premier, deuxième et troisième numéro.

```
P {  
  max-width : 300px;  
  -webkit-hyphens : auto;  
  -ms-hyphens : auto;  
  hyphens : auto;  
  hyphenate-limit-chars : 3 2 5;  
  color : black;  
}
```

2.3.3. La propriété : `hyphenate-limit-zone` (CSS4)

Elle permet de décider d'une zone de césure spécifiant l'espace restant pour la justification du texte. Si un mot est en dehors de la zone de césure alors il est renvoyé à la ligne

La zone de sécurité est représentée en rouge.

```
P {  
  max-width : 300px;  
  -webkit-hyphens : auto;
```

¹ Notons que depuis 2012, la propriété `word-wrap` a été remplacée par la propriété `overflow-wrap` au sein des spécifications CSS3 toutefois aucune implémentation n'a encore changé pour le nouveau nom à l'heure actuelle.

```
-ms-hyphens : auto;  
hyphens : auto;  
hyphenate-limit-zone : 70px;  
text-align : justify;  
color : black;  
}
```

3. Font-strech

La propriété font-stretch permet de condenser ou étirer les caractères d'une police. Le W3C définit 9 mots-clés allant du plus condensé au plus étiré :

```
font-stretch: normal ;  
font-stretch: [ultra-condensed | extra-condensed | condensed | semi-condensed] ;  
font-stretch: [semi-expanded | expanded | extra-expanded | ultra-expanded] ;
```

Ces valeurs sont applicables uniquement sur une famille de police dont les types condensés ou étirés existent.

4. Font-size-adjust

La propriété font-size-adjust permet, lorsqu'une police n'est pas supportée et qu'elle va alors être remplacée par une police de substitution, de conserver la hauteur qu'occuperait la police initiale et éviter ainsi des mises en page qui « saute » :

```
<p style="font-family:verdana, Myriad Pro, sans-serif", font-size:12px;font-size-adjust:0.545;">
```

Lecture Web :

<http://www.brunildo.org/test/xheight.pl> pour tester les rendus de police et connaître la valeur d'aspect de chacune d'elles.

5. Font-feature-settings

Cette fonction est expérimentale, elle permet un contrôle des fonctions typographiques avancées des fontes OpenType. Elles peuvent comporter de multiples glyphes, lesquels sont différents des caractères.

Cette propriété propose un contrôle encore plus fin de la gestion des caractéristiques stylistiques par le biais de mots-clés définis au sein du registre OpenType disponible ici :

<http://www.microsoft.com/typography/otspec/featurelist.htm>

Lecture Web :

Plus d'infos : <http://typographisme.net/post/Les-ligatures-dans-les-navigateurs>

[http://msdn.microsoft.com/fr-fr/library/ie/bg124110\(v=vs.85\).aspx](http://msdn.microsoft.com/fr-fr/library/ie/bg124110(v=vs.85).aspx)

6. Resize

`resize` permet à l'utilisateur de redimensionner un bloc à l'aide d'un simple glisser-déposer de la souris, très pratique par exemple pour les éléments de textes libres d'un formulaire (`<textarea>`), que le visiteur peut à présent agrandir à loisir.

Les valeurs de cette propriété sont :

- `none` : l'élément ne peut pas être redimensionné par l'utilisateur.
- `vertical` : l'élément peut être étiré verticalement uniquement.
- `horizontal` : l'élément peut être étiré horizontalement uniquement.
- `both` : l'élément peut être contrôlé dans les deux sens.

```
textarea {  
resize: both; /* l'élément est étirable dans les deux sens */  
}
```

7. Text-shadow

Permet d'ajouter une ombre à un texte. Une ombre est créée à partir de quatre paramètres : trois longueurs et une couleur. Il est possible de spécifier les décalages de l'ombrage, la couleur et sa zone de flou. Ces effets s'appliquent dans l'ordre spécifié et peuvent ainsi se recouvrir, mais ceux-ci ne recouvriront jamais le texte lui-même.

```
h1 {  
  text-shadow: 0px 0px 9px #777;  
  color: #fff;  
}
```

Des valeurs négatives peuvent être définies pour donner un effet « creusé ».

Il s'agit d'une propriété finalisée, il n'y a donc pas lieu de lui affecter un préfixe (-moz-, -webkit-, ...), la propriété demeure text-shadow quel que soit le navigateur.

8. Les propriétés white-space, text-wrap et text-space-collapse

White-space permet de contrôler l'affichage des espaces blancs et le retour à la ligne automatique. On l'applique plutôt dans une boîte que dans une page entière.

Les codes de tabulation, retour à la ligne (sans balise
), espace entourant un code \A sont ignorés avec les valeurs: normal, nowrap, pre-line

Les espaces sont réduits à un seul, sauf avec les valeurs:

pre, pre-wrap

Les espaces en début et en fin de ligne sont ignorés avec les valeurs:

normal, nowrap, pre-line

Lecture Web :

Exemples ici : <http://www.xul.fr/css/white-space.php>

En CSS3, deux nouvelles propriétés permettent d'affiner ce réglage : **text-wrap** et **text-space-collapse**

La propriété text-wrap spécifie le mode d'adaptation du texte, la façon dont les sauts de ligne sont gérés.

La propriété text-space-collapse spécifie la façon dont les espaces blancs présents dans le code source sont gérés.

9. Les propriétés overflow et text-overflow

text-overflow permet de rogner et rendre invisibles les contenus qui débordent des blocs et ainsi ne pas dénaturer la mise en page d'un document, elle doit être employée avec la propriété overflow :

La propriété overflow admet trois valeurs : hidden, scroll ou auto,.

- visible : le bloc est automatiquement redimensionné selon le contenu, aucune barre de défilement n'apparaît. Il s'agit de la valeur par défaut.
- scroll : une barre de défilement apparaît constamment, même si le contenu ne dépasse pas du bloc.
- hidden : le contenu excédentaire est purement et simplement masqué, sans qu'un ascenseur n'apparaisse.
- auto : les dimensions du bloc sont fixes et la barre de défilement apparaît uniquement dans le cas d'un contenu long.

La propriété text-overflow propose ellipsis afin d'ajouter des points de suspension à l'endroit où le terme est rogné et ainsi laisser un indice visuel pour indiquer la présence de ce contenu masqué

Attention : text-overflow à lui tout seul ne coupe rien du tout ; il ne sert qu'à donner une indication (des points de suspension, par exemple) sur un texte déjà coupé (via overflow:hidden généralement)

Exemple :

```
.box p {  
  padding: 0 20px;
```



```
white-space: nowrap;
overflow: hidden;
o-text-overflow: ellipsis; /* pour Opera 9 */
text-overflow: ellipsis; /* pour le reste des navigateurs */
}
```

Pour ajouter un effet de survol, nous pouvons ajouter la pseudo-classe `:hover` à l'élément et changer la largeur, attribuer la valeur visible à la propriété `overflow` et ré-activer le saut à la ligne (`white-space : normal`) :

```
.box p:hover {
  background-color: silver;
  overflow: visible;
  width: 480px;
  white-space: normal;
}
```

10. Les colonnes

La spécification CSS 3 permet de créer des colonnes de texte de hauteurs égales grâce aux propriétés `column-count` (nombre de colonnes), `column-width` (largeur des colonnes) et `column-gap` (interstice), exemple :

```
<h1>Socrate</h1>
<div class="col">
  <p>Après avoir philosophé un moment...</p>
</div>
```

columns Permet de présenter un texte sur plusieurs colonnes, les différentes propriétés permettent de gérer...

Le nombre et la taille des colonnes :

`columns` (raccourci de `column-count` et `column-width`) : nombre de colonnes et éventuellement largeur de chaque colonne ;

`column-min-width` : largeur minimale de chaque colonne ;

`column-width-policy` : le mode d'affichage des colonnes (valeurs "flexible", "strict" ou "inherit").

Ce qu'il y a entre les colonnes :

`column-gap` : distance entre chaque colonne ;

`column-rule` (raccourci de `column-rule-color`, `column-rule-style` et `column-rule-width`) : couleur, style et largeur de la séparation entre colonnes.

Permettre à un élément de s'étendre sur plusieurs colonnes : `column-span`

Exemples

```
p {
  -moz-column-count: 2;
  -moz-column-gap: 10px;
  -moz-column-rule: 1px solid #ccc;
  -webkit-column-count: 2;
  -webkit-column-gap: 10px;
  -webkit-column-rule: 1px solid #ccc;
  column-count: 2;
  column-gap: 10px;
  column-rule: 1px solid #ccc;
}
```

Exemple pour créer une mise en page à trois colonnes, chacune large de 14 cadratins¹ et écartée des autres d'un intervalle de 2 cadratins :

```
.col {  
  -moz-column-count: 3;  
  -moz-column-width: 14em;  
  -moz-column-gap: 2em;  
  -moz-column-rule: 1px solid #ccc;  
  -webkit-column-count: 3;  
  -webkit-column-width: 14em;  
  -webkit-column-gap: 2em;  
  -webkit-column-rule: 1px solid #ccc;  
  column-count: 3;  
  column-width: 14em;  
  column-gap: 2em;  
  column-rule: 1px solid #ccc;  
}
```

L'intérêt des colonnes CSS tient à leur **gestion lorsque l'espace disponible est moindre que la largeur des colonnes** définies, au lieu que les colonnes soient chassées à la ligne comme cela se produit avec les éléments flottants, leur nombre se réduit automatiquement.

Attention toutefois, **les colonnes posent un réel soucis d'accessibilité sur écran**, si les colonnes sont plus hautes que la hauteur visible par l'utilisateur, celui-ci, arrivé en bas d'une colonne, est obligé de remonter en haut pour continuer sa lecture ! Il est donc important de **ne pas utiliser les colonnes sur de trop grandes hauteurs !**

Les colonnes CSS ne sont pas encore largement prises en charge (OK sur FfX, Chrome et Safari NON sur IEx 9.0), vous devez donc précéder le code standard d'extensions spécifiques des navigateurs. Les autres navigateurs affichent le texte dans une seule colonne dont la largeur égale celle de toutes les colonnes combinées.

¹ Notons que la largeur du bloc de texte se mesure par son nombre de caractères (dont les espaces) par ligne. On fait varier le nombre optimal de caractères par ligne de 45 à 95, notamment en fonction du support.

3. GERER LES IMAGES D'ARRIERE-PLAN

Il y a deux méthodes pour insérer les graphiques d'une page web :

- × **Ajout du graphique dans l'arrière-plan d'un DIV à l'aide des CSS** (exemple : une texture pour créer un fond visuel) – *Il s'agira des images de décoration.*
- × **Ajout d'une balise `img` directement dans le DIV**, en HTML ou XHTML (exemple : insérer une photo dans une page pour illustrer un contenu) - *Il s'agira des images de contenu.*

Rem : Il est conseillé de stocker les graphiques de contenu et les graphiques de présentation dans des dossiers séparés, ainsi, si vous souhaitez modifier la conception du site en conservant le contenu, vous pouvez écartier facilement les éléments (CSS, graphismes) de l'ancienne présentation.

3.1. Insérer et répéter une image d'arrière-plan

Plusieurs propriétés sont associées aux arrière-plans, nous retiendrons :

`background-image`, `background-repeat` et `background-position`.

L'image de fond se déclare en général pour l'élément `body`, mais on peut aussi l'appliquer à un bloc `div`, ou un titre, un lien afin de créer un bouton, etc.

Par défaut, si l'image est plus petite que son conteneur, elle se répète pour le remplir en mosaïques – horizontalement et verticalement.

L'effet de répétition est contrôlé avec `background-repeat` et accepte 4 valeurs en CSS 2.1 :

`no-repeat` (image unique) ;

`repeat-x` pour une répétition de l'image uniquement horizontale ;

`repeat-y` pour une répétition uniquement verticale ;

`repeat` pour la mosaïque complète.

Deux nouvelles méthodes de répétition s'ajoutent avec le CSS31 :

`space` ajoute des espaces vides entre les répétitions.

L'image est répétée dans la direction renseignée autant que nécessaire pour remplir la plus grande partie de la surface dessin de l'image d'arrière-plan, sans avoir à couper la dernière image. L'espace restant non recouvert est distribué entre chaque image de la mosaïque. La première et la dernière image touchent les bords de l'élément.

`round` redimensionne la taille des images.

L'image est répétée dans la direction renseignée autant que nécessaire pour remplir la plus grande partie de la surface de dessin de l'image d'arrière-plan, sans avoir à couper la dernière image. Si cela ne recouvre pas exactement la surface, les images de la mosaïque sont redimensionnées dans la direction de la répétition afin de correspondre.

3.2. Gestion d'un dégradé (avec une image)

Pour appliquer un dégradé à une page, créez une image de dégradé étroite mais très longue, puis appliquez-la à l'élément `body` de la page en la répétant verticalement ou horizontalement. La longueur des pages étant rarement prévisible, ajoutez une couleur d'arrière-plan identique à celle du bas du dégradé. Les images d'arrière-plan viennent toujours se placer par-dessus la couleur d'arrière-plan et lorsque l'image s'interrompt, c'est la couleur d'arrière-plan qui prend le relais. La transition entre l'image et la couleur d'arrière-plan est parfaitement invisible.

```
body {  
  background-image: url(/img/gradient.gif);  
  background-repeat: repeat-x;  
  /* ici pour un dégradé vertical */  
  background-color: #ccc;  
}
```

¹ Supportées à ce jour par peu de navigateurs.

3.3. Gérer la position d'un arrière-plan

`background-position` est la propriété qui permet de définir avec précision l'emplacement d'une image dans une page ou un div :

`top` = image au centre/haut de la page ou du bloc.

`center` = image au centre/centre de la page ou du bloc.

`bottom` = image au centre bas de la page ou du bloc..

`right` = image au centre/droit de la page ou du bloc.

`left` = image au centre/gauche de la page ou du bloc.

`pourcentage pourcentage` (exemple 20% 50%) est la valeur exprimée en pourcentage se rapportant à la taille de la boîte elle-même, lorsqu'on ne donne qu'une seule valeur, celle-ci concerne la position horizontale, et la position verticale est automatiquement fixée à 50%, les positions négatives sont permises.

Pour une image **en haut à droite** :

```
background-position:right top; ou background-position:100% 0%;
```

Pour une image **au centre de la page** :

```
background-position:center center; ou background-position:50% 50%;
```

Pour une image **en bas à droite** :

```
background-position:right bottom; ou background-position:100% 100%;
```

La spécification CSS3 offre à présent la possibilité d'écrire trois ou quatre valeurs séparées par une virgule :

Ces valeurs doivent être précédées d'un mot-clé définissant le côté d'où s'applique le décalage, exemple :

```
background-position: bottom 10px right 20px;
```

Décale une image d'arrière-plan de 10px du bas et de 20px du de la droite de la boîte, par défaut, ce décalage s'effectue par rapport au coin supérieur gauche.

3.4. Utiliser les images pour créer un menu avec puces

La propriété `list-style-type` propose plusieurs styles de puces (carrés, disques, cercles, ...).

La propriété `list-style-image` permet d'y placer des icônes personnalisées en précisant le chemin d'accès à l'image de puce. Les formats d'image reconnus sont GIF, JPEG ou PNG. En toute logique, privilégiez des images de petite taille (environ 10 à 15 pixels pour un texte de taille « normale »).

Exemple :

```
ul#menu {  
margin: 0;  
padding: 0;  
font: bold 1em Georgia, Times, serif;  
list-style-image: url(dossier/puce.png);  
}
```

On constate que les puces s'affichent hors du conteneur, pour y remédier, on peut intervenir sur la marge de gauche de la liste (jusqu'ici de valeur nulle) ou placer les puces dans la liste avec la déclaration `list-style-position: inside` :

```
ul#menu {  
margin: 0;  
padding: 0;  
font: bold 1em Georgia, Times, serif;  
list-style-image: url(dossier/puce.png);  
list-style-position: inside;  
}
```



Comment modifier celle-ci lors d'un survol de son élément de liste par le pointeur de la souris ?

Il s'agit d'associer des images de fond différentes aux éléments et aux éléments survolés¹ (*li:hover*), qui remplaceront les puces des listes :

```
/* gestion générale du menu */
ul {
list-style-type: none;
margin: 0;
padding: 0;
font: bold 1em Georgia, Times, serif;
}
/* gestion des éléments du menu */
#menu ul li{
padding-bottom:20px;
list-style-image: url(puceorange.gif);
padding-left: 20px;
}
/* gestion des éléments du menu au survol */
#menu li:hover, #menu li:focus{
list-style-image: url(puceverte.gif);
}
```

3.5. Images d'arrière-plan multiples (CSS3)

Pour certains designs de site, plusieurs images d'arrière-plan se combinent. Le CSS3 permettant la gestion de plusieurs images d'arrière-plan², il suffit de définir toutes les images à utiliser avec la propriété `background-image`, préciser si elles doivent se répéter et fixer leurs positions respectives. Préférez la version raccourcie, plus simple et moins génératrice d'erreur :

```
.box {
    background:
        url(bordure-vigne-haut-d.png) no-repeat right top,
        url(bordure-vigne-haut-g.png) no-repeat left top,
        url(bordure-vigne-bas-g.png) no-repeat bottom left,
        url(bordure-vigne-bas-d.png) no-repeat bottom right ;
}

<div class="box">
  <h2>Titre</h2>
  <p>Contenu<p>
</div>
```

L'ordre de déclaration est important : l'image la plus proche de la propriété (la première énumérée) sera l'image de premier plan. Si une couleur de fond est déclarée, elle sera toujours reléguée au dernier plan.

3.6. Background-origin (CSS3)

Positionne la zone de l'arrière plan (ou plutôt le point d'origine de l'image d'arrière plan), les valeurs possibles sont :

- `border-box` : La position de l'arrière plan est relative au bord : l'image peut donc se positionner derrière les bords.
- `padding-box` : La position de l'arrière plan est relative au padding (marge interne)
- `content-box` : La position de l'arrière plan est relative au contenu.

Exemple 1 : le point d'origine est celui du contenu

```
div {
```

¹ Le navigateur Internet Explorer jusqu'à sa version 7 ne reconnaît cette pseudo-classe qu'appliquée à la balise <a>.

² Il fallait, avant le CSS3, créer des containers (div) uniquement pour leur assigner ces images.

```
background: url(image.jpg) no-repeat;  
background-origin: content-box;  
}
```

Exemple 2 : le point d'origine commence sous la bordure

```
div {  
  border: 20px dotted #393838;  
  background-origin: border-box;  
}
```

Les préfixes sont requis pour les navigateurs utilisant des pré-versions de cette propriété.

Démonstrations de `background-origin` : <http://www.alsacreations.com/article/lire/1385-css3-backgrounds.html>

3.7. background-size (CSS3)

La propriété CSS `background-size` permet de spécifier la taille de l'image appliquée à l'arrière plan, les valeurs possibles sont :

Dimensions : par exemple "px" ou "%" : précise la hauteur ou la largeur de l'image.

"cover" : forcera à couvrir toute la surface sans déformer l'image (quitte à la rogner).

"contain" : forcera l'image à ne pas dépasser de l'élément sans la déformer.

Exemple 1 : fond qui occupe 30% de la largeur du bloc avec une répétition

```
div {  
  background-image: url(soif.jpg);  
  background-size: 30% 30%;  
}
```

Exemple 2 : "Full-Background" sans déformation

```
html {  
  margin: 0;  
  padding: 0;  
  background: url(image/bg.jpg) no-repeat center center fixed;  
  background-size: cover;  
}
```

Les préfixes sont requis pour les navigateurs utilisant des pré-versions de cette propriété.

Lecture Web :

<http://www.alsacreations.com/astuce/lire/1216-arriere-plan-background-extensible.html>

3.8. background-clip (CSS3)

La propriété `background-clip` permet de définir les limites auxquelles une image d'arrière-plan est rognée (clip en anglais) et donc n'est plus visible. Trois valeurs possibles :

`border-box` : le fond est visible jusqu'à l'extrême limite de la bordure (mais ne la superpose pas). Il s'agit de la valeur par défaut.

`padding-box` : l'extrême limite sera celle du padding, aucun arrière-plan ne sera présent en-dessous de la bordure.

`content-box` : L'arrière plan se limite au contenu, cela correspond à la taille spécifiée dans width et height dans un modèle de boîte classique.

3.9. border-image (CSS3)

La propriété CSS `border-image` permet de déclarer une image sur les bords d'un élément, elle est utilisée en lieu et place de `border-style`.

```
element {
```

```
border: 1px solid black; /* obligatoire */  
border-image: url("image.png") 42 round stretch;  
}
```

En savoir plus sur `border-image` : <https://developer.mozilla.org/fr/docs/Web/CSS/border-image>
Générateur de `border-image` : <http://border-image.com/>

3.10. La gestion des dégradés en CSS(3)

`background-image` possède, depuis le css3, deux nouvelles valeurs : **linear-gradient** pour les dégradés linéaires et **radial-gradient** pour les dégradés radiaux.

3.10.1. Syntaxe pour `linear-gradient`

Les dégradés linéaires utilisent la valeur **linear-gradient()**, il convient de préciser au minimum le point d'arrivée du dégradé ainsi que deux couleurs (ou plus) :

background : linear-gradient(x y, couleur1, couleur2, couleurN)

x et/ou **y** définissent la direction du dégradé. Exemple : "to right" ou "to left bottom",

couleur : le nom, la valeur hexadécimale ou la valeur RGBa

Attention ! A ne pas laisser d'espace entre le mot-clé `linear-gradient` et l'ouverture de parenthèse "(" renfermant les propriétés.

Exemple 1 : du rouge au violet

```
background: linear-gradient(to right, red, orange, yellow, green, blue, indigo,  
violet);
```

Exemple 2 : du vert au transparent

```
background: linear-gradient(to right, green, rgba(0,255,0,0));
```

Anciennes syntaxes `linear-gradient`¹

```
background: linear-gradient(top, green, blue);  
/* faux : il faut indiquer la position finale, donc to bottom */  
background: linear-gradient(bottom right, green, blue);  
/* faux : il faut indiquer la position finale, donc to left top */
```

3.10.2. Syntaxe pour `radial-gradient`

Les dégradés radiaux utilisent la valeur **radial-gradient()**. Elle nécessite d'indiquer la forme du dégradé : circulaire (**circle**) ou elliptique (**ellipse**) ainsi que son point de départ.

background : radial-gradient(forme at x y, couleur1, ..., couleurN)

x et/ou **y** définissent le point d'origine du dégradé radial,

couleur : le nom, la valeur hexadécimale ou la valeur RGBa

Exemple :

```
background: radial-gradient(ellipse at center top, green, blue);
```

Attention ! A ne pas laisser d'espace entre le mot-clé `linear-gradient` et l'ouverture de parenthèse "(" renfermant les propriétés.

Générateur de dégradés: <http://www.cssmatic.com/gradient-generator> , <http://www.css3factory.com/linear-gradients/>

¹ L'ancienne syntaxe fonctionne encore avec les préfixes, tandis que la nouvelle la supporte sans préfixe pour les navigateurs actuels.

3.11. Le préchargement des images

En général, le pré-chargement des images se fait par programmation mais peut être réalisé à l'aide d'un code CSS. Cette technique de pré-chargement CSS comporte des avantages sur un langage de programmation : plus rapide à l'exécution, plus simple à écrire qu'un programme et fonctionne sur tous les navigateurs.

3.11.1. Masquer les images

Deux déclarations CSS permettent de masquer un élément d'une page : `visibility: hidden` et `display:none`. **visibility: hidden** masque l'objet mais réserve sa position et ses dimensions, l'élément occupe donc de l'espace sur la page.

display: none fait abstraction complète de l'objet dans le média concerné (Web,impression, etc.). Tout se passe comme s'il n'existait pas.

La deuxième solution ne crée pas de vides dans le document.

Exemple :

```

```

On assurera le chargement prioritaire de cette image en plaçant ce code au tout début du document, juste sous la balise `<body>`. L'attribut `alt` vide, important, évite aux navigateurs en mode texte ou pour non voyants de tenir compte de cette image.

Pour appliquer cette technique à un groupe d'images, nous créerons une classe (exemple `:cache`)

```
.cache {display: none;}
```

Que nous appliquerons aux images :

```



```

On allégera le code et rassemblera ces images dispersées en regroupant toutes les illustrations à masquer au sein d'une balise `<div>` :

```
<div class="cache">



</div>
```

3.11.2. L'affichage hors de la zone de visualisation

On peut faire appel à une astuce, placer les illustrations hors des limites de la zone vue. 5 000 pixels plus haut que le début de la page devraient suffire :

```
<div class="cache">



</div>
```

CSS

```
.cache {
position: absolute;
left: 0;
top: -5000px;
}
```

Ainsi, tous les navigateurs traiteront ces images... en les reproduisant à un endroit inaccessible au visiteur.

Conclusion sur le préchargement des images

Ces deux méthodes échouent chez les visiteurs pour qui les styles CSS sont inactifs ou qui les ignorent. Dans ces conditions, les images s'affichent alors là où le code HTML les mentionne (probablement en début de document), hors contexte. Une autre solution acceptable est d'écrire les préchargements en JavaScript. Là où ce langage sera désactivé, les images ne seront pas préchargées, ce qui ne nuira pas à la lisibilité du document et ne perturbera personne.