

Cours CSS3 - Bases

Préfixes, sélecteurs, nouveaux styles, Polices, positionnement

Gestion des nouveautés expérimentales

Utilisables (et utilisés) en production, il faut juste faire un travail de veille pour anticiper les évolutions et utiliser les préfixes :

- moz- Firefox
- webkit- Safari et Chrome
- ms- Internet Explorer

Exemple :

<http://codepen.io/smoothie-creative/pen/5f52c65b64ad2020d917bfa38cb86014>

Comment savoir si on a besoin ou pas de préfixe ?

1. Regarder sur CanIUse : <http://caniuse.com/>
2. Utiliser un préprocesseur comme Stylus qui le fait automatiquement
3. Utiliser Autoprefixer <https://github.com/postcss/autoprefixer>

Nous verrons Stylus dans le cours CSS Avancé

Sélecteurs CSS

Sélecteurs de base

Balises, identifiants, classes, descendance...

<http://codepen.io/smoothie-creative/pen/1f7917215fc0823e44b11ebd35a7d3c7/>

Cantonnez-vous aux sélecteurs de balise et de classes, et abandonnez autant que possible les autres, pour assurer une meilleure performance de votre navigateur

Sélecteurs par attributs

Exemples

<http://codepen.io/smoothie-creative/pen/1f7917215fc0823e44b11ebd35a7d3c7/>

Les attributs :

<code>[required]</code>	a cet attribut
<code>[type="text"]</code>	vaut exactement
<code>[class~="red"]</code>	a cette valeur dans la liste
<code>[lang="fr"]</code>	vaut ou commence par « fr- »
<code>[class^="red"]</code>	commence par
<code>[class\$="red"]</code>	fini par
<code>[class*="red"]</code>	contient

Attention : ces sélecteurs s'avèrent moins efficaces que les sélecteurs de base. Veillez à ne pas trop les utiliser.

Les pseudo classes

Les classiques pour les liens :

- `a:link`
- `a:hover`
- `a:visited`
- `element:target`

Actions de l'utilisateur :

- `element:hover`
- `element:active`
- `element:focus`

Etat des éléments interactifs (UI) :

- `element:enabled`
- `element:disabled`
- `element:checked`
- `element:indeterminate`

Langue du document :

- `:lang(fr)`

Etat des contrôles :

- `element:valid`
- `element:invalid`
- `element:required`
- `element:optional`

Pseudo Classes avancées

:only-child	Quand l'élément a qu'un seul enfant
:first-child	Premier enfant de l'élément
:last-child	Dernier
:empty	Pas d'enfant
:not(selecteur)	Qui n'est pas (par exemple un p)
:nth-child(3)	Troisième enfant (peut importe le type)
:nth-child(odd)	éléments paires (ex pour alterner lignes d'un tableau)
:nth-child(even)	éléments impaires
:nth-child(3n+1)	Premier élément tous les 3 enfants
:nth-child(4n+2)	Deuxième élément tous les 4 enfants
:nth-last-child(2)	Deuxième élément en partant de la fin
p:nth-of-type(n)	Enième de ce type (si on cible p, les autre balises seront ignorées)
:nth-last-of-type(n)	Enième en partant de la fin
:only-of-type	Seul enfant de ce type
:first-of-type	Premier enfant de ce type
:last-of-type	Dernier enfant de ce type

Pseudo classes de contenu

:first-line	change le style de la première ligne seulement
:first-letter	change le style de la première lettre d'un élément

Pseudo classes avant / après un élément

:before
:after

pseudo classe magique permettant bien plus que d'ajouter des guillemets avant et après le contenu

Permet de faire la methode du clearfix quand on a des éléments en float :

<http://codepen.io/smoothie-creative/pen/de4dd4b7227aa1d2217d1c3476c82e70>

(sera revu dans le chapitre positionnement)

ou de créer un élément supplémentaire qui n'aurait pas lieu d'être en HTML (niveau sémantique).

Exemple du triangle :

Nouvelles syntaxes

Rappels sur les couleurs



Il existe plusieurs notations pour les couleurs :

- red couleurs simples
- #00FF00 hexadécimal complet
- #0F0 hexadécimal raccourci
- rgb(0,255,0) red, green, blue

Une couleur est un mélange de rouge, vert et bleu, à un degré allant de 0 à 255. L'hexadécimal correspond au même concept en base 16 (FF = 255).

- Noir absence rgb(0,0,0) #000
- Blanc mélange rgb(255,255,255) #FFF

Nouveaux formats de couleur

- rgba(0, 255, 0, 0.5) alpha channel (opacité)
- hsl(0, 50%, 50%) hue, saturation, lightness
- hsla(0, 50%, 50%, 0.5)

Hue = teinte sur la roue des couleurs, de 0 à 360

Rouge 0 ou 360

Vert 120

Bleu 240

Tests opacité :

<http://codepen.io/smoothie-creative/pen/fqLso/>

Unités



Fixes

- px (pixels)

- pt (points)
- pc (pica = 12pt)
- cm,mm

Relatives :

- % par rapport à la taille parente
 - em multiplicateur par rapport à la taille parente
 - rem multiplicateur par rapport à la taille globale
 - vw, vh
-

Nouveaux styles CSS3

Le CSS3 apporte son lot de nouveaux styles, qui va faire gagner beaucoup de temps et de flexibilité.

Pensez à vérifier la compatibilité sur CanIUse et d'appliquer en cas les préfixes nécessaires.

Les bordures arrondies

Grâce à la propriété Border-radius, vous pourrez définir des bordures arrondies à vos blocs

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

Faire un cercle

Vous pouvez du coup facilement faire des cercles, ou des boutons arrondis en utilisant

```
border-radius : 50%;
```

Exemples :

<http://codepen.io/smoothie-creative/pen/ebb82dd2f019ee36ba18a10ea984a5f7>

Bordures en images

```
border-image-source: url(image.png);  
border-image-width: 10;  
border-image-repeat: stretch | repeat | round;
```

raccourci : border-image

Les ombres

Vous pouvez faire des ombres sur les textes et les blocs, à l'extérieur comme à l'intérieur

Sur les blocs

```
box-shadow: 1px 1px 3px #292929;
```

Propriétés :

- Intérieur / extérieur
- décalage horizontal
- décalage vertical
- atténuation
- couleur de l'ombre

Exemples :

<http://codepen.io/smoothie-creative/pen/6b10c98534552ed2f418a82f62d3a7fc>

Exemple fun avec des papiers empilés : <http://jsfiddle.net/chriscoyier/8jmLY/2/>

Sur les textes

C'est à peu près la même méthode, avec la propriété text-shadow

```
text-shadow: 0 1px 0 #333;
```

Il n'est actuellement pas possible de faire une ombre interne sur un texte.

Exemples :

<http://codepen.io/smoothie-creative/pen/f32827e857d76ed26343d2cd3f6cd0af>

Les dégradés

Dégradés de couleurs linéaires ou radiaux

```
linear-gradient(45deg, blue, white 20%, red);  
radial-gradient(10px circle at top left, blue, red);
```

Pour générer facilement un dégradé, des outils en lignes existent :

<http://www.colorzilla.com/gradient-editor/>

Les nouveaux fonds

background-repeat:	round / space
background-clip:	border-box / padding-box / content-box
background-origin:	border-box / padding-box / content-box
background-size:	150px 100px / contain / cover
background-attachment:	scroll local;

Multiples

background: url(top.png) top no-repeat, url(bottom.png) bottom no-repeat;

box-decoration-break: slice | clone;

Filtres

Nouvelle propriété filter:

- grayscale(100%);
- sepia(100%);
- saturate(20%);
- hue-rotate(90deg);
- invert(100%);
- brightness(150%);
- contrast(150%);
- blur(10px);
- opacity(50%);
- drop-shadow(2px 2px 2px grey);

Et bien d'autres

Il y a pas mal d'autres propriétés pour définir par exemple des bordures personnalisées, ajouter "..." quand le texte va déborder d'un bloc à taille définie, etc.

Source :

<http://code.tutsplus.com/tutorials/10-css3-properties-you-need-to-be-familiar-with--net-16417>

Positionnement CSS

Selon qu'on construise une page web ou une application, on ne va pas forcément utiliser les mêmes méthodes de mise en page.

La page web aura plutôt une dimension verticale, paginée, avec une largeur fixe ou non, tandis que l'application web aura une dimension plein écran sans défilement.

Nous allons directement nous servir de cette excellente ressource pour revoir le positionnement <http://fr.learnlayout.com/>

Une très bonne ressource pour résumer ce que l'on va voir : <http://adamschwartz.co/magic-of-css/>

Types d'affichage

Display

- block p, H1, div
- inline-block
- inline span, strong, em

Block

un bloc prend à l'origine toute la largeur de la page. On peut le contraindre en largeur. Si un deuxième block a la place de se placer à côté, il restera toutefois en dessous. Il faudra donc utiliser le float

Inline

Les éléments de textes sont affichés à la suite les uns des autres.

Inline-Block

Parfois, on a besoin d'un mix des deux propriétés. Un block passé en inline-block pendra le moins de largeur possible, et les blocs pourront s'afficher les uns à côté des autres, c'est pratique pour faire un menu horizontal par exemple.

Par contre, comme c'est considéré comme un texte, les espaces dans le code source seront pris en compte, c'est pour cela qu'il y a quelques pixels d'écart entre des blocs

Exemple: <http://fr.learnlayout.com/inline-block.html>

Box-sizing

On verra que cette propriété s'avère très efficace pour le responsive Web design.
Le padding et le border n'augmentent plus la largeur d'un bloc, c'est donc plus facile pour calculer des colonnes

<http://fr.learnlayout.com/box-sizing.html>

Positionnement statique, absolu et fixe

Il existe plusieurs types de positionnement qui sortent du flux du document et peuvent s'avérer très utiles dans certains cas.

<http://fr.learnlayout.com/position.html>

- Statique : positionnement par défaut dans la page
- Absolute : à partir d'un parent mis en relative, possibilité de positionner où on le souhaite via des coordonnées
- Fixed : position fixe par rapport à l'écran, pratique pour faire un menu qui est toujours là même si on scroll

Flotement des blocs

Technique la plus utilisée : elle permet de positionner des blocs les uns à côté des autres.
Pratique pour faire des colonnes.

<http://fr.learnlayout.com/float.html>

Nécessite d'utiliser un clearfix à la fin du flotement pour éviter tout problème de mise en page

Exemple :

<http://codepen.io/smoothie-creative/pen/d70d375b994471e4f3038f65d2381838>

Mise en page type magazine avec les colonnes

Avec CSS3, il est désormais possible de faire une mise en page magazine, avec un contenu commençant sur une colonne et terminant automatiquement sur une autre.

Exemple:

<http://fr.learnlayout.com/column.html>

toutes les propriétés :

http://www.w3schools.com/cssref/css3_pr_columns.asp

Finalement, cette technique n'est pas tellement utilisée.

Il est possible de terminer le contenu d'une colonne

Technique la plus prometteuse : le flexbox

Pour faire une mise en page par colonnes, on utilise une grille fluide faite de float, mais qui apporte son lot de problèmes. Dans un avenir proche (et même dès maintenant), vous pouvez utiliser le modèle flexbox, qui s'avère très pratique

<http://fr.learnlayout.com/flexbox.html>

Le flexbox va d'autant plus loin car on peut définir l'ordre d'affichage des blocs, et le changer par la suite, notamment quand on fait du responsive avec les media queries.

Le Flexbox propose :

- Plusieurs lignes dans le conteneur avec flex-wrap
- D'équilibrer les colonnes en modifiant la taille avec flex-grow
- Inversement de réduire une colonne moins importante avec flex-shrink
- De choisir la direction (gauche vers droite par défaut) horizontale, verticale ...
- ...

Guide complet :

<http://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Exemple :

<http://codepen.io/smoothie-creative/pen/eefdfd19c1435377224932a15bf6d98f>

Le principe est de donner un "poids" aux colonnes pour qu'elles soient plus ou moins grandes.

Affichage à la Pinterest : Masonry

Cet affichage n'est pas actuellement possible en natif, il faut donc utiliser un script qui va faire les calculs nécessaires et positionner les blocs correctement.

<http://masonry.desandro.com/>

Une bibliothèque Js sympa qui utilise Mansonry :

Isotope

<http://isotope.metafizzy.co/>

Développer plus rapidement avec les frameworks CSS

Lorsque vous voulez prototyper un projet, ou lancer un site sans y passer trop de temps, vous pouvez utiliser des plateformes où le travail est amplement maché.

Voici une liste des frameworks les plus en vue :

<http://fr.learnlayout.com/frameworks.html>

Bootstrap est fourni par Twitter

<http://getbootstrap.com/>

Il existe aussi Foundation qui est pas mal

<http://foundation.zurb.com/>

Polices d'écritures Web

Pendant longtemps, ce qui a fait défaut au web est l'incapacité d'inclure une police d'écriture dans un site de la même manière que l'on inclue une image ou une feuille de style.

Depuis le CSS3 c'est possible. Bien souvent, le choix typographique va s'avérer crucial pour la lisibilité, l'ergonomie et le design.

Beaucoup de sites sont aujourd'hui minimalistes et centrés sur la typographie.

Voyons comment les intégrer dans notre page web.

Intégrer une police en CSS3

Le code pour insérer une police en CSS est :

```
@font-face {  
  font-family: 'MyWebFont';  
  src: url('fonts/webfont.eot'); /* IE9 Compat Modes */  
  src: url('fonts/webfont.eot?#iefix') format('embedded-opentype'), /* IE6-IE8 */  
        url('fonts/webfont.woff') format('woff'), /* Modern Browsers */  
        url('fonts/webfont.ttf') format('truetype'), /* Safari, Android, iOS */
```

```
url('fonts/webfont.svg#svgFontName') format('svg'); /* Legacy iOS */  
}
```

Où vous mettriez vos polices dans le dossier fonts/
et la police s'appellerait webfont.

Pourquoi tant de formats ?

Historiquement, chaque navigateur et système utilise un type de rendu de police différent. Aujourd'hui, il faudra quelques années avant de régulariser ça, mais pour le moment on est obligé de rendre plusieurs types de polices :

- .eot
- .woff
- .ttf (à la mode Windows)
- .svg (format vectoriel web)

On va voir ci-dessous comment générer ces formats à partir d'une simple police.

Plus d'informations à ce sujet sur <http://css-tricks.com/snippets/css/using-font-face/>

Intégrer depuis un service de typographie en ligne

Parmis les services les plus réputés :

- Google Web Fonts <https://www.google.com/fonts>
- Typekit d'Adobe <https://typekit.com/> (payant)

L'intégration est ultra simple puisqu'il suffit de sélectionner les typographies vous intéressant pour votre site, puis le service vous donne le code nécessaire : la feuille de style à appeler, puis le nom de la typo à utiliser.

Exemple :

<http://codepen.io/smoothie-creative/pen/ac77b32d9cda778fd78e6135791cafd7>

Créer ses propres polices compatibles Web

Vous avez une typographie mais pas sa version compatible web ?

Le site Font Squirrel pourra vous aider à générer les 4 formats typographiques pour le web ainsi que le code CSS nécessaire à son affichage.

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

Si vous achetez une typo aujourd'hui, il est fort possible que les formats web soient déjà inclus.

Typographies Web - quelques conseils

Parceque les typographies sont lourdes à charger. Veillez à ne pas en charger trop sur son site Web.

De plus, intégrer trop de polices différentes peuvent nuire à l'harmonie de la charte graphique.

Privilégiez 2 à 3 typographies différentes maximum par site.

Picto Fonts

Vous trouverez aussi des icones et pictogrammes intégrés sous forme de police d'écriture. Quel est son intérêt ?

Avec une police d'écriture vous pouvez :

- Choisir la taille
- Choisir la couleur
- Choisir la graisse
- Mettre des effets d'ombre

Quand les pictos sont intégrés dans une police d'écriture, vous aurez les mêmes avantages !

De plus, vous chargez une police seulement qui contient des dizaines de pictos, soit une requête HTTP plus légère.

Les pictos sous police sont évidemment

Parmi les services proposant des polices de pictogrammes :

- Icomoon <https://icomoon.io/app/#/select>
- Pictos <http://pictos.cc/>

Ainsi que des polices toutes prêtes à l'emploi comme

- Ligature symbols http://kudakurage.com/ligature_symbols/
 - Linecons <http://designmodo.com/linecons-free/>
-

Animations et transitions CSS3

Transformations 2D

Vous allez pouvoir faire des rotations, des déplacements, des agrandissements

`obj` transform:

- rotate(45deg);
- translate(10px, 5px);
- translateX(5px); translateY(5px);
- scale(2,1);
- scaleX(2); scaleY(1)
- skewX(5deg); skewY(5deg);

Exemple :

<http://codepen.io/smoothie-creative/pen/a39f1d63ceda56432eb85e7a400c11f4>

Préliminaires pour la 3D

Le conteneur doit créer un environnement 3D :

- position: relative;
- perspective: none | 500px;

Valeur élevée : objet éloigné / perspective légère

Valeur faible : objet proche / perspective forte

Les blocs 3D doivent permettre la superposition :

- position: absolute;
- transform-style: flat | preserve-3d;

Transformations 3D

Transform

- translateZ(100px);
- translate3d(0, 0, 100px);
- scaleZ(2);
- scale3d(1, 1, 2);

Rotations 3D

Transform:

- rotateX(45deg); / rotate3d(1, 0, 0, 45deg);
- rotateY(45deg); / rotate3d(0, 1, 0, 45deg);
- rotateZ(45deg); / rotate3d(0, 0, 1, 45deg);

Il s'agit du mouvement autour des axes :

- x : penché en avant / arrière
- y : penché sur les côtés
- z : tourné sur les côtés

Options pour les transformations

Changer l'origine d'une transformation 2D :
transform-origin: 50% 50% | top left;

Changer la position du point de fuite en 3D :
perspective-origin: 50% 50% | top left;

Pour masquer le côté opposé d'un objet 3D :
backface-visibility: hidden;

Exemple et ressources

<http://desandro.github.io/3dtransforms/docs/introduction.html>

Attention à la compatibilité : <http://caniuse.com/#feat=transforms3d>

Chrome Experiments : de très beaux exemples de ce qui peut être fait dans un navigateur
<http://www.chromeexperiments.com/>

Blogs et tutos CSS :

<http://tympanus.net/codrops/>

<http://css-tricks.com/>

Transitions

transition-property:	all none propriete1, etc;
transition-duration:	3s .2s
transition-timing-function:	ease linear;
transition-delay:	0 2s;

Raccourci : transition

Une transition n'a de sens que s'il y a un changement de style.

Exemples :

<http://codepen.io/smoothie-creative/pen/7a9c1fc3e9582ff46c2a4d7d6fc7f3e2>

Animations

animation-name:	none nomdelanimation;
-----------------	-------------------------

animation-duration:	1s;
animation-timing-function:	ease linear;
animation-iteration-count:	1 infinite;
animation-direction:	normal reverse alternate;
animation-delay:	0s 2s;
animation-fill-mode:	none forwards backwards both;

Raccourci : animation

animation-play-state: running | paused;

Keyframes

```
@keyframes nomdelanimation {  
  from {  
    width: 200px; }  
  50% {  
    width: 300px;  
  }  
  to {  
    width: 500px;  
  }  
}
```

Il est possible de définir des animations très complexes avec cette technique, un peu comme à l'époque de Flash.

Ressources

En cas de doute, le site W3School apporte pas mal de tutos et d'explications sur les styles CSS
http://www.w3schools.com/css/css3_2dtransforms.asp

Un très bon exemple des possibilités des transformations :

<http://rupl.github.io/unfold/>

Transitions expliquées par Alsacrations :

<http://www.alsacreations.com/tuto/lire/873-transitions-css3-animations.html>

Un bon récapitulatif des animations CSS3 :

<http://www.justinaguilar.com/animations/>

Animations, les effets :
<http://easings.net/fr>

Exercices et exemples

Création d'une page web en mode landing page sur le modèle de <http://teamtreehouse.com/>

Création d'un menu déroulant multi niveau CSS réutilisable.