

Travaux dirigés n°8

IMAC 1

HTML5 & CSS3

Dans ce TD, nous allons découvrir la nouvelle révision du langage HTML : HTML5. Pour illustrer ce dernier, nous développerons une galerie d'images en utilisant quelques-unes de ses dernières spécificités, aidés par la dernière version de CSS.

Avant-propos

HTML5, qu'est-ce que c'est ?

HTML5, on en parle depuis quelques années, déjà. Depuis 2003, un groupe de travail indépendant de développeurs de navigateurs (WHATWG) en rédige les spécificités, relayé en 2007 par le plus officiel W3C. La grosse difficulté de créer un successeur au HTML4, c'est la rétrocompatibilité. En effet, il faut que les nouveaux éléments soient compatibles avec les navigateurs, mais aussi avec les anciennes versions de HTML. Autant dire qu'il s'agit d'une tâche complexe... D'où la lenteur de son développement !

Quoi de neuf ?

Une des principales nouveautés de HTML5, c'est sa syntaxe. Le XHTML que nous avons vu dans tous les TDs précédents était, comme son nom l'indique, basé sur le XML. Il fallait donc qu'un document XHTML soit bien formé suivant les spécificités de XML : les balises bien fermées, en minuscules, avec des attributs prédéfinis, etc. En HTML5, c'est terminé. Par exemple, le code suivant, qui n'aurait pas du tout été valide en XHTML, l'est pour HTML5 :

```
<audio src="mysound.ogg" controls loop />
```

La syntaxe HTML5 n'est pas moins stricte, mais le développeur est plus libre. Vous noterez d'ailleurs que cette simple ligne permet d'afficher un lecteur audio sur votre page...

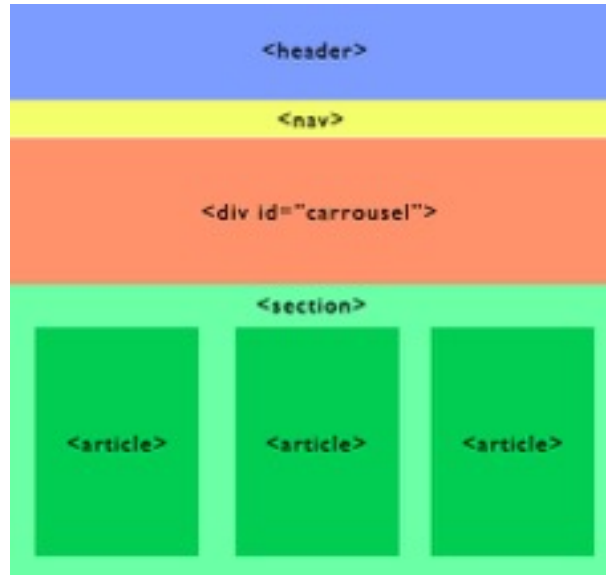
Autre nouveauté, les différents blocs ne sont plus gérés de la même façon. Plutôt d'être utilisés en fonction de leur rendu à l'affichage (<div>, <p>, , ...), on va plutôt privilégier le type de contenu. Cette alternative sémantique est très importante pour l'architecture de la page.

De nouvelles balises font donc leur apparition, parmi lesquelles :

- <section> pour les différentes sections d'un document : chapitres, en-têtes, etc.
- <article> comme son nom l'indique, pour l'article d'un blog ou d'un journal.
- <aside> pour le contenu annexe à un article (une barre latérale par exemple).
- <header>, <footer>.
- <nav> pour le contenu permettant la navigation.
- <canvas> pour avoir la possibilité de dessiner dans un élément.
- ...

Comment ça marche ?

HTML5 permet donc, entre autres, de mieux architecturer le contenu d'une page. Par exemple, le site que vous avez réalisé en XHTML dans le TD3 aurait une structure qui pourrait ressembler à celle-ci :



Vous pouvez bien sûr continuer d'utiliser des balises déjà connues, comme les `<div>`. Mais dès que vous en avez la possibilité, préférez-lui les nouvelles balises de types décrites plus haut. Après, évidemment, toutes les balises sont « stylables » en CSS pour en modifier l'apparence.

Qui l'utilise ?

Les deux gros problèmes de cette nouvelle norme, c'est qu'elle met du temps à être prise en charge et, encore une fois, la compatibilité avec les anciens navigateurs. Tant que des internautes navigueront avec des logiciels obsolètes, HTML5 ne pourra pas décoller. Vous l'aurez compris, Internet Explorer 6 (pour ne citer que lui !) ne gérant même pas toutes les normes CSS2, il est utopique de vouloir y faire tourner une application HTML5.

La plupart des navigateurs modernes (Firefox, Opéra, Chrome, Safari et même pour certaines Internet Explorer 9) implémentent au fur et à mesure les nouvelles règles. Pour ce TD, vous allez devoir travailler sur un d'eux !

Ressources

Voici quelques documentations officielles sur HTML5 :

- [Site officiel](#) (anglais)
- [Tableau de comparaison entre HTML 4 et HTML 5](#) (anglais)
- [Définition des balises](#) (anglais)
- [Migration de HTML4 à HTML5](#) (anglais)
- [Premier brouillon de HTML5](#) (français)

Les exercices de ce TP sont à rendre par e-mail en fin du TP.
L'objet du mail doit être "[IMAC1][html]TD8". Vous publierez également les pages sur votre compte étudiant.

Exercice 1

Re-réaliser une version simple du site web du TD 3 en utilisant HTML5 et CSS3 à la place de HTML4 et CSS2. L'interface doit rassembler à la page suivante (les images sont disponibles sur la page web du cours) :

<header>

<nav>

<div id="carrousel">

<section>

<footer>

<article>

<article>

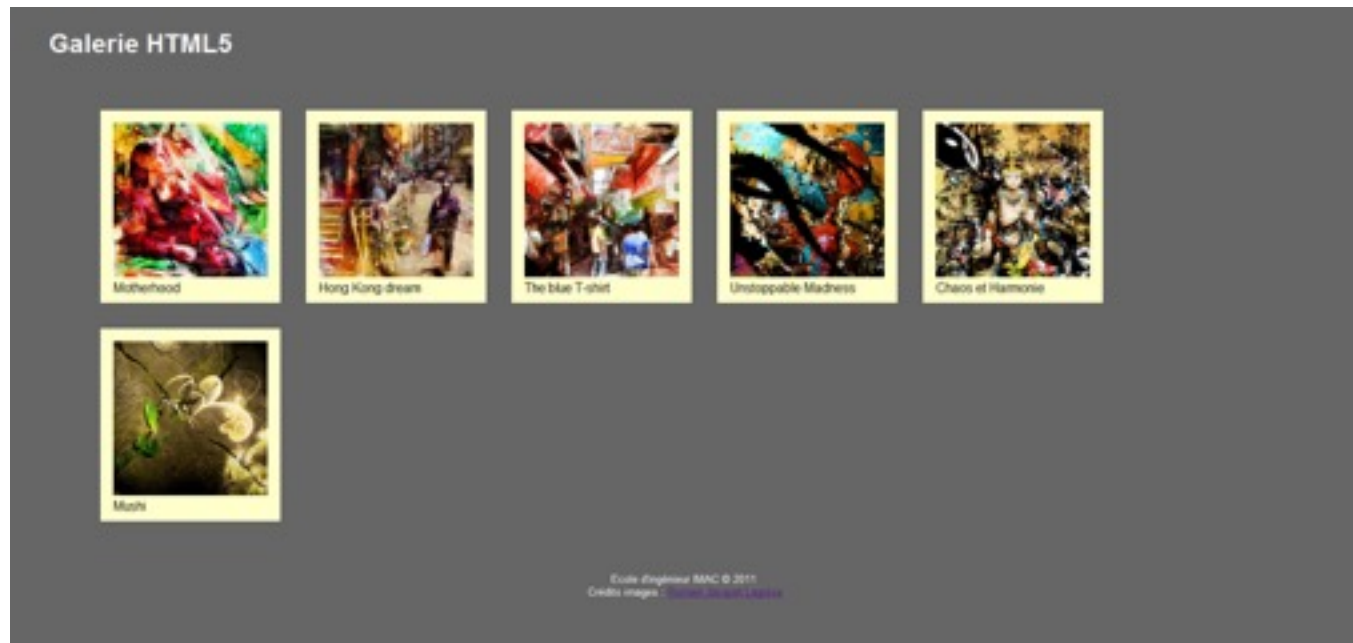
<article>

Exercice 2

Dans ce TD, vous devriez créer une galerie d'images en HTML5 et CSS3. Dans l'archive fournie, vous avez six images que vous allez devoir mettre en place sur la même page.

Structure HTML

Grâce aux nouvelles balises HTML5 (en particulier header, footer et section) et à une feuille de style CSS basique, créer la structure de votre page de façon à avoir le résultat suivant (cette capture d'écran est également dans l'archive en meilleure résolution) :



Quelques pistes :

- Utilisez toute la largeur de votre fenêtre.
- Chaque élément peut être un `<figure>` et `<figcaption>`.
- Pensez à utiliser comme unité « em » plutôt que « px », pour chacun des éléments (même la taille d'une vignette et d'une image). Pour plus d'informations sur cette unité :

<http://www.alsacreations.com/article/lire/563-gerer-la-taille-du-texte-avec-les-em.html>

Typographie

HTML5 permet l'utilisation de polices externes. Ici, nous allons utiliser **l'API Google Font** afin de modifier la typographie des légendes. Grâce au **Google Font Previewer**, trouvez une police calligraphiée qui vous convient. Suivez les explications du site pour l'appliquer sur le texte de vos vignettes.

Ombre CSS

Vous allez ensuite appliquer une ombre à vos vignettes pour obtenir un effet Polaroid. CSS3 le rend possible grâce à la propriété `box-shadow`. Seulement, comme toutes les règles CSS3, cette dernière n'est pas implémentée de la même façon suivant les navigateurs. Vous allez donc devoir écrire une règle différente pour chaque moteur de rendu de navigateur. Seuls trois supportent la propriété : Firefox, Safari / Chrome et Opéra.

- Firefox (moteur Gecko) : `-moz-box-shadow`.

- Safari / Chrome (moteur Webkit) : `-webkit-box-shadow`.
- Opera (Presto) : `box-shadow`.

En vous basant sur la **documentation** de `box-shadow`, créez une ombre noire d'opacité 70% décalée de 5 pixels et avec un dégradé.

Rotation CSS

Autre nouveauté en CSS3 : la rotation d'éléments. Comme `box-shadow`, `rotate` n'est pris en charge que par les navigateurs récents.

- Firefox : `-moz-transform:rotate(angle)`
- Safari / Chrome : `-webkit-transform:rotate(angle)`.
- Opera : `o-transform:rotate(angle)`.

Vous allez appliquer une **rotation** à vos vignettes. Seulement, on aimerait avoir un angle de rotation différent pour chacun des éléments, pour entretenir l'effet Polaroid. Ce n'est pas possible de calculer un angle aléatoirement et ce serait rébarbatif de créer une classe différente pour chaque élément, donc vous allez devoir ruser.

Il existe une pseudo-classe en CSS3 qu'on pourrait qualifier de « modulo » permettant de sélectionner tous les N éléments : `nth-child`. Par exemple :

`figure:nth-child(even)`

sélectionne un élément sur deux. De la même façon :

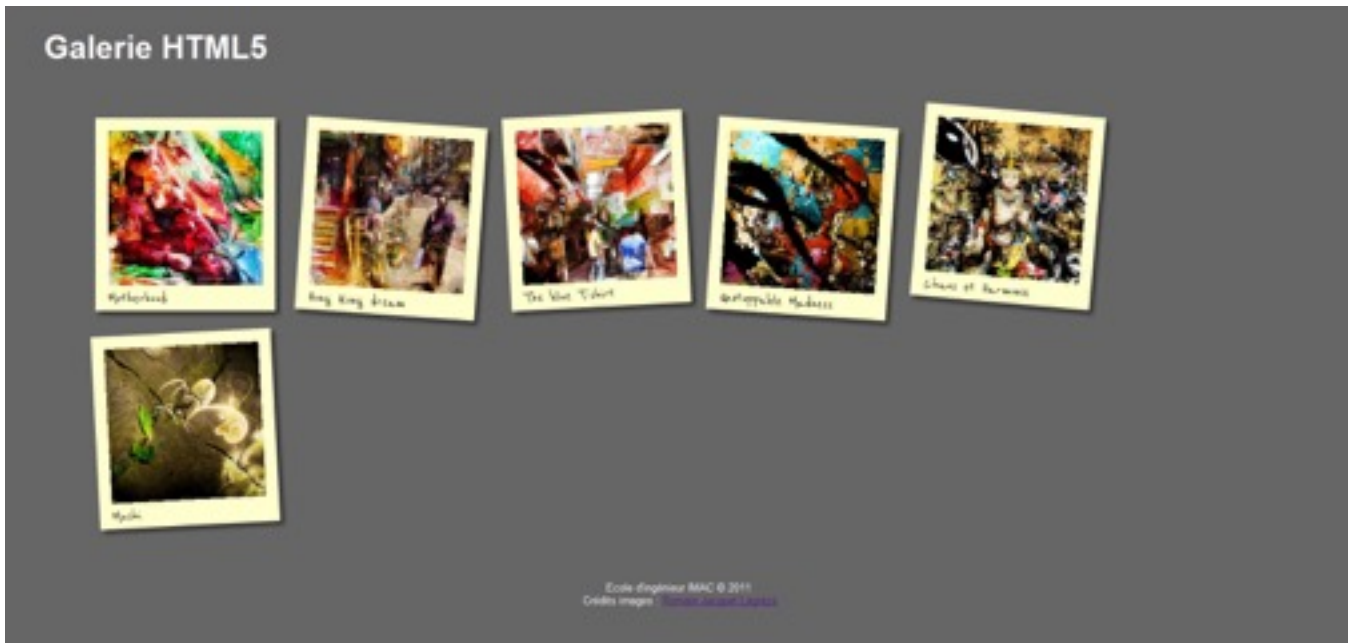
`figure:nth-child(3n)`

sélectionne un élément sur trois.

Grâce à cette pseudo-classe et en combinant les angles de rotation, donnez un aspect aléatoire au placement des vignettes.

Réactivité

Vous devez maintenant avoir le résultat suivant :



Maintenant, on voudrait qu'au survol, les images s'agrandissent. Pour cela, vous allez utiliser la pseudo-classe `:hover` que vous connaissez déjà, ainsi que la propriété CSS3 `scale`. Cette dernière ressemble beaucoup à `rotate` :

- Firefox : `-moz-transform:scale(ratio)`.
- Safari / Chrome : `-webkit-transform:scale(ratio)`.
- Opera : `o-transform:scale(ratio)`.

Lors du survol sur la vignette, agrandissez-la avec un ratio de 1,3. Vous veillerez également à ce que la rotation soit rétablie pour que la vignette soit droite.

Animation

Enfin, pour que l'agrandissement soit plus fluide, vous allez utiliser votre première propriété CSS3 d'animation : **transform**. Cette dernière demande littéralement au navigateur : « si quelque chose modifie l'apparence de cet élément, effectue ce changement en l'animant ». Avec le premier paramètre de cette transition, nous allons détecter le changement de taille (donc la transformation vue précédemment).

- Firefox : `-moz-transition:-moz-transform delai type`
- Safari / Chrome : `-webkit-transition:-webkit-transform delai type`
- Opera : `-o-transition:-o-transform delai type`

Placez une animation linéaire de 100 millisecondes dans la pseudo-classe `:hover` de votre vignette.

Au final, vous devez avoir le résultat suivant :



Aller plus loin

- Modifiez les couleurs des vignettes en simulant l'aspect aléatoire.
 - Créez un fond dégradé sans image, uniquement à l'aide de CSS.
- ★ Implémentez le drag & drop pour pouvoir modifier l'ordre des vignettes à votre guise.