Module 4

HTML5 / CSS3 / jQuery

Webdesign et Interactions

Table des matières

Nouvelles balises HTML5

Document HTML5

```
<!DOCTYPE html>
<html lang="fr">

  <head>
    <meta charset="utf-8" />
    <title>Titre du document</title>
    <link rel="stylesheet" href="" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <!--[if lt IE 9]>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/html5shiv/3.7.3/html5shiv.min.js">
    </script>
    <![endif]-->
  </head>

  <body>
            <!-- contenu -->
  </body>

</html>
```

Balisage sémantique

Par rapport à ce qui a été vu en HTML4, le principe d'imbrication des balises et de construction d'une page web restent les mêmes.

Le but du balisage sémantique introduit par HTML5 est de donner de façon explicite **un sens au contenu des différents éléments**, que ce soit pour les humains ou les navigateurs.

De nouvelles balises vont faire leur apparition dans la structuration d'un site, afin de remplacer l'élément qui en soit n'a aucune signification. Cet élément peut toujours être utilisé.

Par exemple, la section d'entête d'une page Web pouvait être présentée ainsi en HTML4 :

```
<div id="entete">...</div>
```

Désormais, une balise spécifique permet d'indiquer qu'il s'agit de l'entête

```
<header>...</header>
```

ou

```
<header id="entete">...</header>
```

Le code est ainsi plus logique et plus lisible.

Attention :
Les nouvelles balises vues dans ce module ne sont pas utilisables en XHTML et HTML4

Plus de souplesse dans la syntaxe

Ainsi vous pouvez écrire les balises en **minuscules ou majuscules**, ne pas mettre de **slash final** aux balises auto-fermantes, même les guillemets autour des attributs ne sont plus obligatoires. Cependant, il est fortement conseillé (surtout tant que HTML5 n'est pas généralisé) de conserver les « bonnes pratiques » vues dans les chapitres précédents.

Doctype simplifié

```
<!DOCTYPE html>
```

Meta et link simplifiées

```
    <meta charset="utf-8" />
    <link rel="stylesheet" href="style/blog_style.css" />
```

Nouvelles balises de structuration d'un site

<header>

Section d'introduction d'un article, d'une autre section ou du document entier (en-tête de page).

```
<article>
  <header>
    <h1>Titre de l'article</h1>
    <p>Auteur : bidule</p>
  </header>
  <p>Contenu de l'article</p>
</article>
```

<section>

Section générique regroupant un même sujet, une même fonctionnalité, de préférence avec un en-tête, ou bien section d'application web. Groupement de contenus liés.

```
<section>
     <header></header>
```

```
    <article></article>

    <article></article>
</section>
```

<main>

L'élément HTML **<main>** représente le contenu principal du `<body>` du document ou de l'application. Un document ne peut pas voir plus d'un seul élément `<main>. Il ne contribue pas au plan du document (purement informatif).`

```
<main>
  <h1>Titre principal</h1>
  <article>
    <h2>Titre de l'article</h2>
    <p>Contenu de l'article</p>
  </article>
  <article>
    <h2>Titre de l'article</h2>
    <p>Contenu de l'article</p>
  </article>
</main>
```

<article>

Section de contenu indépendante, pouvant être extraite individuellement du document ou syndiquée (flux RSS ou équivalent), sans pénaliser sa compréhension. Il vise à baliser des blocs de contenu utiles que l'on pourrait extraire du document tout en conservant leur sens et leurs informations.

```
<article>
  <h1>Titre de l'article</h1>
  <p>Contenu de l'article</p>
</article>
```

<nav>

Section possédant des liens de navigation principaux (au sein du document ou vers d'autres pages).

```
<nav>
  <ul>
    <li><a href="index.html">Page d'accueil</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</nav>
```

<aside>

Section dont le contenu est un complément par rapport à ce qui l'entoure, qui n'est pas forcément en lien direct avec le contenu mais qui peut apporter des informations supplémentaires.
L'élément `aside` est destiné au contenu *indirectement* lié à l'article lui-même : il représente ce qui l'entoure, de façon annexe.

```
<aside>
  <h4>Sources de l'article</h4>
  <ul>
    <li><a href="#">Lien 1</a></li>
    <li><a href="#">Lien 2</a></li>
    <li><a href="#">Lien 3</a></li>
  </ul>
</aside>
```

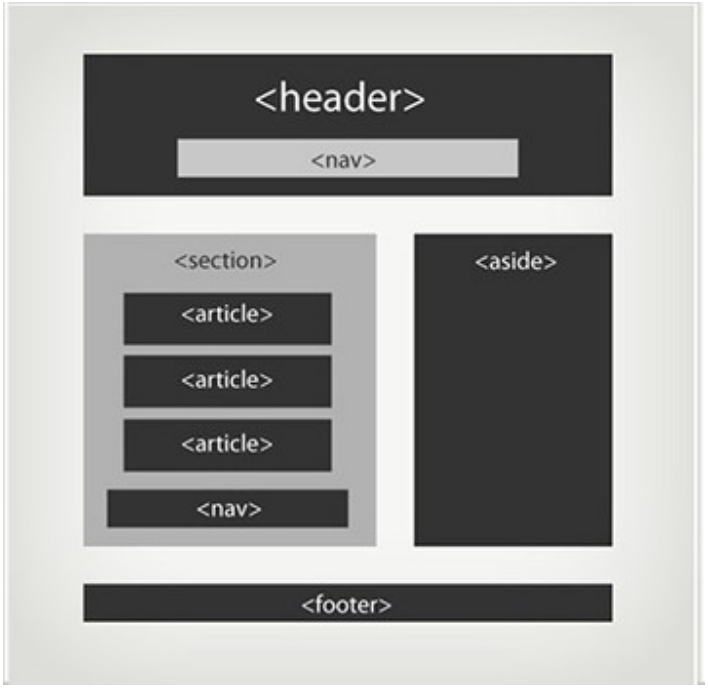
<footer>

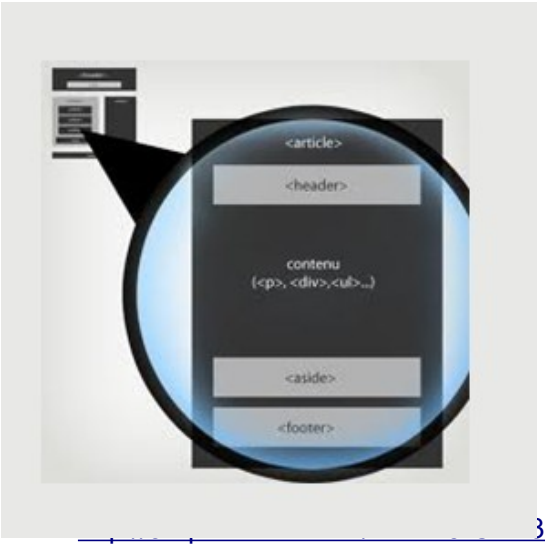
Section de conclusion d'une section ou d'un article, voire du document entier (pied de page).
On y place des informations concernant l'auteur, des mentions légales, une navigation ou une pagination (en combinaison avec `<nav>`), un logo de rappel, des coordonnées, des dates de publication.

```
<article>

  <footer>
    <p>Posté par Simon, le
      <time datetime="2012-02-02">2 février 2012</time>
    </p>
  </footer>
</article>
```

Source *: Alsacréation (retranscription des normes W3C : http://dev.w3.org/html5/spec/single-page.html)*





Source de l'image : alscreations

Compatibilité Internet Explorer 7 et 8

Internet Explorer version **7 et 8 ne reconnaissent pas ces nouvelles balises HTML5**. Pour assurer la compatibilité sur ces vieux navigateurs (éviter que le site « casse »), il est nécessaire de rajouter un bout de code alternatif entre les balises **<head> et </head>** de votre page Web.

```
    <!--[if lt IE 9]>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/html5shiv/3.7.3/html5shiv.min.js">
    </script>
    <![endif]-->
```

Ce bout de code permettant d'importer un script .js ne sera lu que par les versions d'Internet explorer inférieures à la version 9.

TP14_html5 / TP15_blog2

Nouvelles balises de contenu

<figure>

Unité de contenu, c'est-à-dire que cet élément sert de conteneur dans lequel s'insèrent divers éléments comme des images, des schémas, des vidéos, des tableaux ou encore des blocs de code. L'objectif est de lier ce contenu à une légende, définie par l'élément `<figcaption>` (facultatif).

```
<figure>
  <img src="image.jpg" alt="une image" />
  <figcaption>Légende associée</figcaption>
</figure>
```

<audio>

Insérer directement un fichier son dans une page web.
L'attribut control ajoute des boutons de contrôle play, stop, pause, volume. Privilégier les fichiers .ogg.
Si le premier format de fichier n'est pas reconnu (ogg), le navigateur tentera de lire le second (mp3) ...

```
<audio controls="controls">
 <source src="horse.ogg" type="audio/ogg" />
 <source src="horse.mp3" type="audio/mpeg" />
 Votre navigateur ne peut pas lire ce fichier audio
</audio>
```

<video>

Insérer directement une vidéo sans avoir recours à un lecteur flash par exemple.
L'attribut control ajoute des boutons de contrôle play, stop, pause, volume. Privilégier les fichiers .ogg.
Si le premier format de fichier n'est pas reconnu (mp4), le navigateur tentera de lire le second (ogg)

```
<video width="350" height="240" controls="controls">
  <source src="movie.mp4" type="video/mp4" />
  <source src="movie.ogg" type="video/ogg" />
  Votre navigateur ne peut pas lire ce fichier video
</video>
```

<canvas>

L'élément <canvas> a été introduit afin de pouvoir créer des éléments graphiques 2D en Javascript à la volée. Il permet de mettre en place une zone pour les dessins ou les applications graphiques.

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid
#d3d3d3;">
Your browser does not support the HTML5 canvas tag.
</canvas>

<script>

var c=document.getElementById("myCanvas");
var ctx=c.getContext("2d");
ctx.font="30px Arial";
ctx.strokeText("Hello World",10,50);

</script>
```

A voir (liens vers la doc) :

`<embed>, <mark>, <meter>, <progress>, <time>`

Et aussi (liens vers la doc) :

`<command>, <details>, <datalist>,<keygen>, <bb>, <output>, <ruby>, <rt>` et `<rp> <datatemplate>, <rule>, <nest>,<summary>`

Documentation :

- **Liste exhaustive des balises HTML par Mozilla :**

https://developer.mozilla.org/fr/docs/Web/HTML/Element

- **w3school – Site de référence (repérez dans la liste les balises les balises marquées « new »)**

http://www.w3schools.com/tags/default.asp

Nouveaux attributs

Documentation :

- **Liste des attributs HTML5 : http://41mag.fr/liste-des-balises-html5/liste-des-attributs-html5**
- **Liste de tous les attributs : https://developer.mozilla.org/fr/docs/Web/HTML/Attributes**

Zoom :

- **l'attribut data- permet d'ajouter librement des informations (chaîne de caractères) rattachées à une balise. http://www.alsacreations.com/article/lire/1397-html5-attribut-data-dataset.html**
- Nouveaux éléments et valeurs pour l'attribut input (formulaires) : https://developer.mozilla.org/fr/docs/Web/HTML/Element/input
- En développement, les microdonnées (attributs) qui permettent d'enrichir sémantiquement le contenu d'un site (un + pour le référencement). Le site de référence : schema.org

TP autonome :

- Lire la documentation et tester les balises :
  <embed>, <mark>, <meter>, <progress>, <time>

- Voir la documentation globale :
  https://developer.mozilla.org/fr/docs/Web/HTML/Element

- Lire l'article sur les microdonnées :
  http://www.alsacreations.com/article/lire/1509-microdata-microformats-schema-semantique.html
- Voir le générateur de microdonnées : http://schema-creator.org/

- Tester l'attribut "contenenteditable" :
  https://developer.mozilla.org/fr/docs/Web/HTML/Attributs_globaux/contenteditable

TP16_zone

Propriétés graphiques CSS3

L'évolution du langage CSS permet aujourd'hui de l'utiliser pour des applications graphiques poussées : effets d'ombrages, coins de conteneurs arrondis, effet de transition ...

Ce langage viendra ainsi parfois se **substituer à l'utilisation de scripts externes (.js)** optimisant ainsi le temps de chargement d'une page.
Également il viendra se substituer à des éléments graphiques (.jpg, .png) allégeant le poids d'une page Web.

Pour savoir au moment du développement du site quel navigateur interprète correctement quelles propriétés : http://caniuse.com/

Il faudra **préfixer certaines propriétés** pour assurer leur affichage sur certains navigateurs :

- Le préfixe -webkit- est destiné aux navigateurs safari, android browser, chrome
- Le préfixe -moz- est destiné au navigateur firefox (mozilla)
- Le préfixe -o- est destiné au navigateur opera
- Le préfixe -ms- est destiné au navigateur internet explorer

Pour tester, le rendu sur **Internet Explorer 7 et 8** (faut-il vraiment assurer la compatibilité aujourd'hui?), 9, 10, tapez **F12** et choisir dans l'onglet qui s'affiche « mode navigateur : ie8 »

Font-face

Il est maintenant possible de choisir simplement n'importe quelle police d'écriture (en plus des polices Web : arial, helvetica, trebuchet ms ...) pour rédiger son contenu.
Chose incroyable : cette propriété est supportée par ie7 et ie8 !

Avant d'utiliser une police :

- **Assurez-vous qu'elle est libre de droits**
- **Testez-la sur différents navigateurs (un fichier vous sera fourni)**
- **Ne prévoyez pas d'importer trop de polices pour un même site pour ne pas « l'alourdir »**

1 - Choisissez votre police, récupérez le fichier .ttf soit :

- sur votre ordinateur (Windows > Fonts)
- fichiers dans fournis dans [documents] > [fonts]
- sur un sites spécialisés : http://www.dafont.com/fr/

(Assurez vous que cette police est libre de droits)

2 - Allez à l'adresse : http://www.fontsquirrel.com/fontface/generator

3 - « add fonts » : chargez votre (vos) police(s). Cochez : **« optimal »** et « Agrement » puis « download your kit »

4 – Dé zippez le dossier téléchargé. Double clic sur le fichier .html pour vérifier le rendu (voir sous différents navigateurs)

5 – Créer un dossier [fonts] à la racine du site
Copiez/collez à l'intérieur seulement les fichiers : 2 fichiers police en : **.woff2** - **.woff**

6 – Ouvrez avec votre éditeur le fichier « stylesheet.css », copiez le bout de code généré au début de votre propre feuille de style :

7 – Attention à penser à modifier le chemin depuis votre feuille de style vers le dossier [fonts]
Exemple avec la police Lobster

```
@font-face {
    font-family: 'lobster_twoitalic';
    src: url('../fonts/lobstertwo-italic-webfont-webfont-webfont.woff2')
format('woff2'),
         url('../fonts/lobstertwo-italic-webfont-webfont-webfont.woff')
format('woff');
    font-weight: normal;
    font-style: normal;

}
```

8 – Vous pouvez maintenant utiliser cette police qui est hébergée sur votre serveur
HTML

```
<h1 class="titre_page">Un titre qui a du style</h1>
```

CSS

```
h1.titre_page {
font-family: 'lobster_twoitalic';
font-weight:normal;
}
```

Note : il est recommandé de préciser « **font-weight : normal** » pour les titres (h1 > h6)

TP autonome

Choisissez 1 ou 2 éléments textuels auxquels vous vous voulez appliquer une police spéciale (max 3 polices dans un site) : <h1>, <p> ... sur le blog du Webmaster et refaites la procédure ci dessus.

Google fonts

Un autre moyen d'importer une police est d'utiliser une de celle hébergée par Google sur le site : http://www.google.com/webfonts

La procédure (vue en cours) est simplifiée puisqu'il suffit de choisir sa police "add to collection" et de cliquer sur le bouton "use". Un code HTML est généré (à coller dans la partie `<head>` de votre site) ainsi que le CSS associé (à copier dans votre fichier CSS) .

Malheureusement, toutes les polices ne sont pas sur ce site, donc si vous ne trouvez pas celle que vous voulez, vous devrez faire la procédure vue avant afin d'héberger cette police sur votre propre site. De plus il semble qu'il y ait des problèmes de compatibilité sur certains navigateurs.

TP17_font_awesome

Border radius

Arrondir les coins de vos éléments : div, section, ul ...

La propriété border radius va vous permettre de définir la valeur de l'arrondi. Un site pratique pour faire des tests et générer le code : http://border-radius.com/
Il ne vous reste qu'à copier-coller le code généré dans votre propre feuille de style

Exemple entête dont tous les coins ont un arrondi de 5px

```
header {
border-radius: 5px;
border : 1px solid #000;
/* Pour voir l'arrondi ! Fonctionne aussi avec la propriété : background
(color et image ) */
}
```

Cette propriété n'est pas supportée par ie7 et ie8 mais est-ce vraiment grave ?

- **Non : l'affichage sera légèrement différent sous ces navigateurs**
- **Oui : il faudra alors renoncer à l'utiliser et remplacer par une image aux coins arrondis en png (transparence)**

Shadow

Text shadow

La propriété `text-shadow` permet de créer des ombres sur un texte.

Exemple

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

Les 4 valeurs représentent successivement :

- **Décalage de l'ombre vers la droite**
- **Décalage de l'ombre vers le bas**
- **Valeur du flou de l'ombre**
- **Couleur de l'ombre (hexadecimal possible)**

Il est possible d'ajouter plusieurs ombres portées à un texte en les séparant par des virgules.

```
text-shadow:0px 0px 8px #333333, -3px -2px 0px #333333;
```

Outil en ligne : http://westciv.com/tools/shadows/
Cette propriété s'utilise de manière non-préfixée et n'est pas supportée par ie7 et ie8.

Box Shadow

Même principe que pour text-shadow mais appliqué à des « boites » : div, article, ul … et prend des propriétés supplémentaires.

```
box-shadow:1px 1px 4px 4px rgb(0,0,0);
```

Les 5 valeurs représentent successivement :

- **Décalage de l'ombre vers la droite**
- **Décalage de l'ombre vers le bas**
- **Valeur du flou de l'ombre**
- **Largeur de l'ombre**
- **Couleur de l'ombre (hexadecimal possible)**

Un outil pratique pour tester les ombrages et générer le code.
http://www.cssmatic.com/box-shadow

Comme la propriété text-shadow, il est possible d'ajouter plusieurs ombres portées à un même élément en les séparent par des virgules.

TP Autonome

Border radius

Arrondissez l'angle d'un élément de votre site. Pour être visible, l'élément ainsi modifié doit avoir une couleur de fond ou une bordure (un bouton, l'entête, le conteneur, pied de page, une image ...)

Shadow

Appliquez une ombre à un élément (conteneur) de votre blog et/ou à du texte.

Transitions

Le principe de base d'une transition CSS3 est de permettre une transition douce entre l'ancienne valeur et la nouvelle valeur d'une propriété CSS lorsqu'un événement est déclenché : la plupart du temps au survol de cet élément (hover).

Cette propriété doit être préfixée. Voir http://caniuse.com/#search=transitions.
Elle n'est pas reconnue par ie7-ie8-**ie9**.

Voici quelques exemples d'utilisations courantes de cette propriété (alsacréations) :

- **Couleur de texte :**

http://www.alsacreations.com/xmedia/tuto/exemples/transitions/transition1.htm

- **Couleur de fond :**

http://www.alsacreations.com/xmedia/tuto/exemples/transitions/transition2.htm

- **Déplacement de texte :**

http://www.alsacreations.com/xmedia/tuto/exemples/transitions/transition3.htm

- **Dimensionnement d'éléments :**

http://www.alsacreations.com/xmedia/tuto/exemples/transitions/images1.htm

- **Menu animé :**

http://www.alsacreations.com/xmedia/tuto/exemples/transitions/menu.html

- **Menu graphique animé :**

http://www.alsacreations.com/xmedia/tuto/exemples/transitions/menu2.html

- **Infobulles :**

http://www.alsacreations.com/xmedia/tuto/exemples/transitions/infobulles.html

Sur chaque page, clic droit sur l'élément et « inspecter avec firebug » pour voir les propriétés CSS.

Propriétés :

`transition-property`	Précise les propriétés CSS à transformer. Possibilité d'indiquer la valeur "all"
`transition-duration`	Précise la durée de la transition Valeurs : seconde (s) ou la milliseconde (ms)
`transition-timing-function`	Précise la fonction de transition à utiliser, le modèle d'interpolation (accélération, décélération…) : • **ease** : Rapide sur le début et ralenti sur la fin. • **linear** : La vitesse est constante sur toute la durée de l'animation. • **ease-in** : Lent sur le début et accélère de plus en plus vers la fin. • **ease-out** : Rapide sur le début et décélère sur la fin. • **ease-in-out** : Le départ et la fin sont lents. Avancé : • **cubic-bezier**(.72, .13, .79, .73) : gérer la progression ***ctrl + e*** *pour aide à l'édition avec l'éditeur brackets*
`transition-delay`	Précise le retard (ou l'avance) du départ de la transition. Valeurs : seconde (s) ou la milliseconde (ms). Fixée à 1s l'animation se déclenchera 1 seconde après le survol. Inversement, à -1s donnera l'impression que l'animation a déjà commencée 1 seconde avant le survol.

Exemple : passage de la couleur gris clair à noir des liens au survol en 1seconde

```
a {
transition-duration: 1s; /* Durée */
transition-property: color; /* Propriété affectée */
transition-timing-function: ease;
color : #ccc;
}
a:hover{
color: #000;
}
```

Il est possible de raccourcir la syntaxe :

```
selecteur {
transition:
        <transition-property>
        <transition-duration>
        <transition-timing-function>
        <transition-delay>;
}
```

Il est possible de préciser plusieurs transitions séparées par des virgules :

```
selecteur {
  transition: width 2s ease, height 3s linear;
}
```

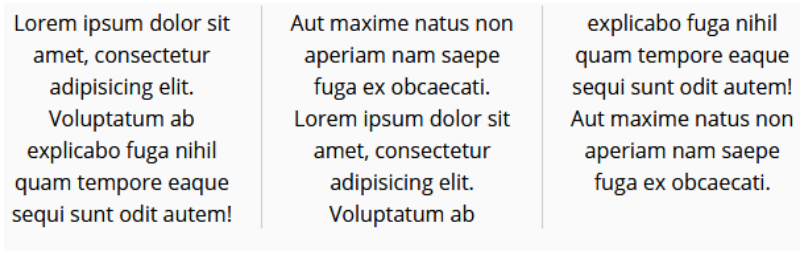
TP Guidé

Reproduire l'animation suivante en copiant-collant le code dans votre page « portfolio » (Utilisation de firebug).

http://www.alsacreations.com/xmedia/tuto/exemples/transitions/infobulles.html

Columns

Présentation de texte en colonnes.



Principales propriété :

```
column-count : 3;  (3 colonnes)
Column-width : 200px; (largeur de chaque colonne, valeur possible : auto pour ajustement automatique de la largeur)
column-gap : 10px; (espace entre chaque colonne)
column-rule : 1px solid #ccc; (ligne grise entre chaque colonne)
```

En savoir plus : http://www.w3schools.com/css/css3_multiple_columns.asp

Gradient

Réaliser des dégradés linéaires ou circulaires comme valeur de la propriété background-image sur ce modèle : linear-gradiant(direction, couleur1,couleur2...)

Exemple (arc-en-ciel)

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

Note

spécifier une background-color pour les navigateurs ne supportant pas les gradients

Générateur

http://www.cssmatic.com/gradient-generator

RGBA

Toute couleur peut être exprimée en rgba ou le «a » (alpha) défini le degrés de transparence.

```
Div { background-color: rgba(0, 0, 0, 0.5);
/* 0.5 défini votre degrés de transparence. Valeur de O à 1 */ }
```

Note :

La propriété *opacity : 0.5* va appliquer une transparence à **l'ensemble** du bloc (texte y compris) et non seulement à une couleur comme avec la notation *rgba*

Transform

Propriété permettant la rotation, l'inclinaison, le changement d'échelles ... d'éléments HTML.
Peut être couplé avec la pseudo-classe **:hover** pour une inclinaison, un agrandissement d'échelle au survol. Et avec la propriété **transition** pour donner un effet de mouvement.

Exemple simple :

```
div {
        transform : rotate(10deg);
}
```

(la div sera penchée de 10 degrés)

Article complet :
http://www.alsacreations.com/article/lire/1418-css3-transformations-2d.html

Générateur :
http://www.css3maker.com/css3-transform.html

Animation

Permet de faire plusieurs transitions et transformations simultanées sur un/plusieurs éléments.

Exemple simple

```
.bloc {
   width: 200px;
   min-height: 200px;
   border: 1px solid #000;
   margin: auto;
   animation-duration: 5s;
   animation-iteration-count: infinite;
   animation-name: demo-animation;
   animation-timing-function: ease-out;
}
```

```
@keyframes demo-animation {
0% {
   transform: translateY(0%);
}
50% {
   transform: translateY(25%);
}
100% {
   transform: translateY(0%);
}
}
```

Illustration (bloc avec image de fond) : http://foundation.zurb.com/emails.html

Exemples et exercices

http://www.w3schools.com/css/css3_animations.asp

Générateur

http://www.css3maker.com/css3-animation.html

Voir des réalisations CSS3

- Site présentant des réalisations avancées en CSS3 : https://tympanus.net/codrops/
- Également très pointu : https://css-tricks.com/
- Terrain de jeux pour intégrateurs : https://codepen.io/
- Demos en CSS : https://codemyui.com/tag/pure-css/

TP Autonome

Dans la page zone.html créez un titre : <h2>Nouvelles propriétés CSS3</h2> puis <h3>Transition</h3> <h3>Gradients</h3> <h3>Animation</h3>... pour tester ces propriétés.

Responsive Web Design

Présentation

L'utilisation de plus en plus courante de smartphones et tablettes oblige les créateurs de sites à proposer un affichage différencié pour ces types de média afin de permettre une lecture et une navigation plus agréables.

On parle de responsive (sensible) design lorsque le **design d'un site va pouvoir s'adapter dynamiquement au format d'écran grâce au CSS**.

Il est notamment possible avec CSS3 de détecter la largeur en pixel de l'écran de l'utilisateur (tenu horizontalement ou verticalement) et d'adapter l'affichage en fonction.

Exemple simple : http://www.alsacreations.com/xmedia/tuto/exemples/mediaqueries/layout.html

Cette technique – Media Queries (requête des medias) – permet un affichage conditionnel de certaines propriétés de votre feuille de style. Ce qui se traduirait par exemple en CSS par « Si la largeur de l'écran est inférieure à 480px : fixer la largeur du conteneur à 400px ».

Présentation officielle : www.w3.org/TR/css3-mediaqueries/

Des conditions dans le CSS

Nous avons déjà vu qu'il était possible de déclarer des feuilles de style alternatives dans la partie <head></head> d'un document HTML.

Par exemple, pour proposer un autre type d'affichage à l'impression :

Code HTML

```
<head>
      <link rel="stylesheet" media="print" href="print.css" />
</head>
```

Il est possible de faire la même chose à l'intérieur de votre feuille de style (celle qui gère l'affichage du site à l'écran).

```
@media print { /* Début de la condition : si impression */
```

```
   nav, footer{
   display:none;
   }

} /* Fin de la condition : si impression */
```

Vous avez fait une condition : **Si** (@) le media utilisé est une imprimante (print) **Alors** ({}) appliquer le style compris entre les accolades. Notez l'imbrication des accolades.

Ce principe sera toujours le même, seule la condition va changer. Le plus souvent : « **Si** l'internaute utilise un smartphone **Alors** appliquer le style compris entre les accolades ».

Evolution

Nouveaux critères

Grâce aux media Queries, les critères de sélections vont être plus fins et ainsi permettre un ciblage plus précis du support utilisé.
Il devient alors possible d'utiliser des critères dans la condition :

- **and (et)**
- **only (uniquement)**
- **not (non)**
- **, (ou)**

Ce qui nous permettra par exemple de « dire » :
Si le media est un écran (screen) ET que la largeur maximum de la fenêtre (max-width) est de 650px
Alors appliquer le style entre accolades.

Cela se traduit par :

```
@media screen and (max-width: 900px) {

  #conteneur {
    width : 100%;
  }

} /* Fin de la condition */
```

Les conditions peuvent se faire à la fin de la feuille de style principale du site (ci-dessus) OU dans une feuille de style distincte, au moment de l'inclusion (ci-dessous smartphone.css)

```
<link rel="stylesheet" media="screen and (max-width: 900px)"
```

```
href="css/smartphone.css">
```

Propriétés utilisables pour les conditions

L'utilisation la plus répandue des media Queries porte sur la propriété width (max- et min-) afin de d'adapter le graphisme à la largeur de la fenêtre.
Il y a d'autres possibilités (en gras les plus utilisées) :

- **Color**

support de la couleur (bits/pixel)

- **color-index**

périphérique utilisant une table de couleurs indexées

- **aspect-ratio**

ratio du périphérique de sortie (par exemple 16/9)

- **device-aspect-ratio**

ratio de la zone d'affichage

- **device-height**

dimension en hauteur du périphérique
[valeur en px: préfixe min- et max- acceptées]

- **device-width**

dimension en largeur du périphérique
[valeur en px : préfixe min- et max- acceptées]

- **grid**

périphérique bitmap ou grille (ex : lcd)

- **height**

dimension en hauteur de la zone d'affichage
[valeur en px, em : préfixe min- et max- acceptées]

- **monochrome**

périphérique monochrome ou niveaux de gris (bits/pixel)

- **orientation**

orientation du périphérique
[valeurs : landscape (paysage) ou portrait]

- **resolution**

résolution du périphérique
[valeurs : dpi]

- **scan**

type de balayage des téléviseurs (`progressive` ou `interlace`)

- **width**

dimension en largeur de la zone d'affichage [valeur en px, em : préfixe min- et max- acceptées]

Différents supports mobiles (tablettes, smartphone)

Tableau de résolutions mis à jour : http://mydevice.io/devices/

Pour résumer (susceptible d'évolution), indications concernant les largeurs :

- **Smartphone : entre 320px et 480px**
- **Tablettes en portrait : entre 480px et 760px**
- **Tablette en paysage et écran ordinateur : entre 760px et 1024px**
- **Grand écran : supérieur à 1024px**

Remarque : la taille « standard » adoptée pour un affichage écran d'ordinateur est d'environ 1000px.

Permettre le Zoom

Afin d'avoir une meilleure compatibilité (iphone) et initialiser le zoom, vous devrez rajouter le bout de code suivant dans la partie <head> de votre fichier HTML, juste avant la balise <title>

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

TP18_affichage_mobile

Framework CSS

Définition

En programmation informatique, un framework ou structure logicielle est un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel (architecture).
https://fr.wikipedia.org/wiki/Framework

Cette méthode de travail déjà utilisée pour d'autres langages (PHP, javascript...) fait son apparition depuis peu en CSS. Il s'agira ici essentiellement de débuter un nouveau projet avec une bibliothèque CSS « prête à l'emploi ».

Si l'utilisation d'un framework n'est pas avantageuse pour certains projets (intégration d'une maquette libre non basée sur des grilles par exemple) il faudra **au minimum** pour tout site charger une feuille de style dite « **reset** » permettant d'assurer un même affichage sur tous les navigateurs. Il existe plusieurs reset.css en ligne (joindre cette feuille de style en plus de votre style propre), l'un des plus utilisé est :

- normalise.css : http://necolas.github.io/normalize.css/

Principaux frameworks

Les 2 plus utilisés sont :

- Bootstrap (développé par Twitter) : http://getbootstrap.com/
- Foundation : http://foundation.zurb.com/

Nouveau Framework (Google) :

- Materialize : http://materializecss.com/

Pus léger, plus facile à mettre en place :

- Knacks (Alsacréations) : http://knacss.com/

TP19_framework_css

Sass (Syntactically Awesome StyleSheets)

Sass (Scss) est une nouvelle syntaxe CSS permettant de coder plus vite (moins de lignes), d'utiliser des variables, d'automatiser certaines taches répétitives… Avant d'utiliser Sass il est important de bien connaître le CSS « classique ».

Il y aura 2 feuilles de styles : un feuille ayant l'extension .scss dans laquelle le code sera tapé, les modifications de cette feuille de style seront reportées au format css dans une 2ieme feuille de style, c'est cette dernière qui sera liée au site.

TP Guidé

- Installer l'extension «**Brackets SASS** » sur votre éditeur.
- Créer un nouveau projet [demo-sass] dans votre répertoire [www]
- Indiquez à Brackets **l'environnement** de travail : fichier > *ouvrir un dossier* > *demo-sass*
- Créez un fichier « index.html » et une feuille de style : « style**.scss »,** si tout est correctement configuré un fichier style**.css** a du être généré automatiquement. Liez ce fichier **.css** à votre index.html.
- Nous allons faire des modifications dans le fichier style.scss en nous aidant de la documentation : http://sass-lang.com/guide

Note : des commentaires sont ajoutés automatiquement dans votre feuille de style, c'est cette feuille .ccs qui sera mise en ligne. Dans tous les cas je vous conseille de **compresser la feuille de style avant la mise en ligne,** cela supprimera les commentaires et tous les caractères inutiles. Elle sera plus légère à charger. Vous pouvez pour cela utiliser l'outil en ligne : http://cssminifier.com/

TP jeu role

JavaScript avec jQuery

JavaScript

Le langage JavaScript (différent de Java) est un langage de programmation utilisé pour réaliser des pages Web dynamiques.

Il s'exécute coté client (navigateur) et s'insère dans le code HTML soit **directement**, soit par **inclusion** d'un script externe.

Ce langage a été inventé en 1995 par Brendan Eich (membre du conseil d'administration de Mozilla) Concrètement il va permettre de rendre les sites plus « vivants » et **faciliter les interactions** avec l'internaute. Ex : Slides d'images, affichage de messages informatifs, infobulles, effets graphiques, onglets dépliants…

Lier un fichier JavaScript

```
<script type="text/javascript" src="js/script.js" async="async"></script>
```

Attributs :

- type - optionnel en HTML5
- src - indique l'emplacement du fichier source
- async – attention : ne s'utilise qu'avec HTML5 !!! - Optionnel - Permet d'exécuter le script après l'affichage des pages et ainsi gagner en performances.

Insérer du code JavaScript dans la page HTML5

Exemple de code Javascript (*changer le mot « Texte » en « Modification de texte »)*

```
<p id="demo1">Texte</p>

 <script type="text/javascript">
      document.getElementById("demo1").innerHTML="Modification de texte";
 </script>
```

Même chose avec jQuery

```
<p id="demo1">Texte</p>

 <script type="text/javascript">
      $('#demo1').html('Modification de texte');
 </script>
```

Javascript ou CSS3 ?

Nous avons vu que les propriétés CSS3 tendaient à remplacer l'utilisation de JavaScript (jQuery) notamment avec les animations rendues possibles en CSS3

Le débat est ouvert :

- L'avantage de JavaScript est d'être reconnu par les navigateurs anciens (ie7et8) , et jQuery (vu plus bas) permet d'assurer une compatibilité maximale entre navigateurs.
- L'avantage de CSS3 est qu'il est plus facile à modifier à et n'a pas besoin d'être chargé en plus (rapidité d'affichage). Cependant, les propriétés nouvelles (animation...) ne seront pas forcement bien implémentées dans tous les navigateurs.

Je pencherai donc pour dire : Tout ce qui peut-être fait en HTML5 et CSS3 doit l'être, tant que la compatibilité avec les navigateurs est assurée. S'il y a des problèmes de compatibilité ou si CSS3 trouve ses limites utiliser JavaScript ou, plus facile : faites du JavaScript avec jQuery :-)

jQuery : "écrire moins, faire plus"

JQuery est une bibliothèque JavaScript gratuite, utilisée par les « grands sites » (Daily Motion, Amazon, Facebook, Google Code...) aussi bien que par les petits. Le site de référence : http://jquery.com/

Chargement de la bibliothèque

Lorsqu'on utilise jQuery, on charge par inclusion de script (.js) une bibliothèque de fonctions « prêtes à l'emploi ». Ce script peut être hébergé sur votre site ou chargé depuis un autre site.

Inclusion de la bibliothèque, **au choix :**

+ Inclusion bibliothèque jQuery hébergée sur votre site :
Copier-coller le code récupéré à cette adresse : `http://code.jquery.com/jquery-1.12.4.min.js` dans un fichier `jquery.min.js` et liez-le à votre page juste **avant la fermeture de la balise </body>**

```
<script type="text/javascript" src="js/jquery.min.js"></script>
```

+ Inclusion bibliothèque jQuery (version 1.12.4) hébergée en externe (site jQuery.com) :

```
<script src="http://code.jquery.com/jquery-1.12.4.min.js" integrity="sha256-ZosEbRLbNQzLpnKIkEdrPv7lOy9C27hHQ+Xp8a4MxAQ=" crossorigin="anonymous"></script>
```

Note :

> *La version chargée est plus ancienne (1.12.4) car c'est celle qui est utilisée en environnement WordPress. Ceci afin que les développement que vous allez travailler puissent être intégrés à un futur thème.*

Inclusion de votre fichier .js personnel (**APRÈS** la bibliothèque jQuery)

Tous les scripts jQuery devront être insérés après l'inclusion de la bibliothèque avant la fermeture de </body>. Ceci afin que le javascript se charge en dernier et na ralentisse pas l'affichage HTML/CSS de la page.

```
<script type="text/javascript" src="js/script.js"></script>
```

script.js doit contenir ce code (une seule fois, le code est développé à l'intérieur)

```
$(function() {

  // Commencer à coder en jquery ici

});
```

Le D.O.M (arborescence du document) doit être chargé avant l'exécution des fonctions jquery. Cette fonction assure cela en 1er lieu, elle doit donc **encadrer systématiquement** tous les développements jquery.

Coder en jQuery

Le langage utilisé pour créer vos propres applications est **simplifié et plus fiable** (compatibilité entre navigateurs) qu'en JavaScript « traditionnel ».

Le principal avantage est donc le gain de temps lors de la création d'un site, également le fait que ce langage est bien plus facile à utiliser et à comprendre pour un « non développeur ».

L'inconvénient est qu'il requiers le chargement de la bibliothèque Jquery pour fonctionner (temps de chargement du fichier).

Fonctionnement
Une fois l'élément HTML sélectionné **$()** - il se «transforme» en objet jquery et vous pouvez lui appliquer toutes les **.methodes** (fonctions) incluant des **paramètres** d'exécutions.

Sur le principe :

```
$(selecteur).methode(parametres);
```

Exemple :

```
<p id='demo1'>Ceci est un texte ciblé par jQuery</p>
```

```
$(function() {

  $('#demo1').css('color', 'red');

});
```

Passe la couleur du texte en rouge (propriété CSS)

- Fonctions jQuery : http://api.jquery.com/

TP Guidé

- Lier la bibliotheque jquery sur l'index du Blog (hébergée en interne)
- Codez un script jquery directement dans la page HTML (entre les balises `<script type="text/javascript"></script>`).
- Ce script permettra de changer le CSS (modifier la couleur) de tous les <h2> de cette page.

Les sélecteurs

La sélection en jquery se fait avec la syntaxe :

```
$('')
```

La syntaxe à ajouter à l'intérieur des simples quotes reprend celle des **sélecteurs CSS3**

Ainsi pour « attraper via jquery » un titre HTML

```
<h1 id='titre-site'>Le Blog du Webmaster</h1>
```

Le sélecteur jQuery sera

```
$('#titre-site')
```

Tous les paragraphes

```
$('p')
```

Les images ayant la class="logo"

```
$('img.logo')
```

Liste des sélecteurs, sur la doc officielle : http://api.jquery.com/category/selectors/

Intéressant :

- Élément qui contient un mot: http://api.jquery.com/contains-selector/
- Élément vide : http://api.jquery.com/contains-selector/

Les méthodes

Une fois la (les) balises) HTML « attrapée» nous allons la manipuler grâce aux méthodes jQuery.

Nous avons vu précédemment 2 méthodes :

- .html() qui permet de remplacer le contenu textuel d'un bloc HTML (paragraphe par exemple)
- .css() qui permet d'appliquer une propriété CSS

Syntaxe simple : la méthode *« s'accroche »* au sélecteur avec un **point** suit son **nom** (html, css, animate, attr...) puis des **parenthèses** qui pourront contenir des paramètres ou rester vides. Enfin un **point-virgule** pour indiquer que l'instruction est terminée.

Ce qui donne :

```
$('#titre-site').html();
```

Il est possible de chaîner les méthodes.

Ainsi, si on veut appliquer plusieurs modifications au titre, il est possible d'écrire :

```
$('#titre-site').html().css().after();
```

Quelques méthodes utiles :

- .attr(); Créer un attribut HTML ou change sa valeur (src, href, target...)
- .css(); Ajouter une propriété CSS
- .addClass(); Ajouter une classe
- .append(); Insère du contenu a la fin de la sélection
- .after() ; Insère du contenu après la sélection
- .hide(); Cacher un éléments
- .show(); Afficher un élément

DOC : Toutes les méthodes listées en page d'accueil de l'API : http://api.jquery.com/

Les paramètres (arguments)

Indiqué **entre parenthèses** de la méthode= selon quels paramètres va s'exécuter l'action ?

Ils sont séparés entre eux par une **virgule**. Une méthode peut être écrite sans paramètres, dans ce cas on ecrit quand même les parenthèses.

Change le titre en *titre modifié* (paramètre de .html) de couleur rouge (2 paramètres de .css) et insère un paragraphe après (.after).

```
$('#titre-site').html('Titre modifié').css('color','red').after('<p>Paragraphe introductif</p>');
```

DOC : Les paramètres dépendront de la méthode dans laquelle ils s'insèrent (voir la doc spécifique de la méthode).

TP20_jquery_introduction

Les événements

Méthode permettent de déclencher une action (=lancer d'autres methodes) : au click sur un bouton, au survol d'une image, lorsque la molette de la souris est tournée, au clic droit...

HTML

```
<div id="demo2">
      <p>Afficher une bordure dans le block conteneur</p>
</div>
<button class="voir">Voir le block</button>
```

JQuery

```
$('.voir').click(function () {
        $('#demo2').css('border', '1px solid #ccc');
 });
```

Au .click sur le bouton ayant la class=''voir'', je lance une fonction (événement) qui va ajouter (en css) à la div ayant l'id= ''demo2'' une bordure grise de 1px.

DOC : Liste de événements jQuery : http://api.jquery.com/category/events/

Les variables

Lors de développement Javascript (et autres), vous aurez besoin de stocker des unités d'informations pour les réutiliser plus tard (les tester, les modifier...). Cela peut être : du texte, des nombres ou un boléen (vrai/faux).

Ces unités d'informations s'appellent des variables. Voici comment on crée une variable en Jquery, on lui affecte la valeur placée à droite (exemples) :

```
var monTexte = 'Ceci est un texte écrit via jQuery';
var numeroLoto = 6;
var monTitre = $('#titre-site');
var monTitreModifie = $('#titre-site').text('Hello World');
```

TP21_code_jquery

TP Autonome

Revoyez la syntaxe jQuery pas à pas avec les exercices guidés : https://www.codecademy.com/fr/tracks/jquery

Adapter des fonctionnalités issues du libre

On va ainsi pouvoir, avec de bonnes bases en HTML et CSS adapter rapidement des scripts réalisés par d'autres et mis en ligne gratuitement sur le WEB.

Voir le site jQuery **U**ser **I**nterface, qui offre des fonctions jquery facilement adaptables. Il faut pour cela, en plus de la bibliothèque jQuery, lier à votre fichier HTML la **bibliothèque jQuery UI**

- **jQueryUI**
  http://jqueryui.com/demos/

TP22_jquery_accordion

Exemples de sites dédiés :

- **Pop-up**

http://fancyapps.com/fancybox/

TP23_jquery_fancybox

- **Zoom**

http://www.jacklmoore.com/zoom

- **Slides**

http://www.owlcarousel.owlgraphic.com/index.html

- **Divers (commentés)**

http://tympanus.net/codrops/tag/jquery/

TP24_jquery_slide

TP autonome

Choisissez une application jquery de votre choix trouvée sur Internet et intégrez-la au Blog (dans zone-jquery ou autre page). Vous pouvez par exemple en trouver à cette adresse :
https://tympanus.net/codrops/tag/jquery/

TP autonome – finalisation des modules 1 à 4

> Finalisez votre Blog : vérifier le code (commenter/organiser), testez sous différents navigateurs, « peaufinez » le CSS, sortez du dossier racine tout ce qui ne doit pas être mis en ligne (images lourdes, psd, fichiers tests…)

> Intégrez du Jquery à votre mini-projet réalisé en cours (slide, fancybox…)

> Mettez au propre le site one-page travaillé en cours (mini-cv)

> Une fois fait, réalisez une **page index.html** qui serait un « portail » renvoyant (dans un nouvel onglet) vers vos différentes réalisations web (un dossier par réalisation) :

- Blog
- Mini Projet
- Page CV
- Maquettes

> **Mettez en ligne** sur votre espace Alwaysdata