

Atelier N°6***Les grandes nouveautés du HTML5, API Validation et Hébergement du site***

- Récapituler les grandes nouveautés du HTML5
- Découvrir quelques outils et logiciel de création de Site Web
- Valider un site
- Héberger un site

6 Les grandes nouveautés du HTML5



Figure 1

Exemple de structure des éléments en HTML 5

6.1 Les nouveaux éléments de périphériques

HTML 5 définit trois nouveaux éléments destinés à gérer nativement les aspects visuels, sonores et graphiques de l'ère numérique moderne : <canva><audio>, <video>.

6.1.1 L'élément CANVA

Permet la création de dessins ou de graphiques. Représente une zone de dessin bitmap au sein de la page, pilotée par JavaScript et capable de restituer des formes, des diagrammes et graphiques, des animations ou autres applications visuelles

On commence par initialiser une balise canvas dans son .html.

```
<canvas id="canvas">
    Message qui s'affichera si la balise n'est pas supportée par le
    navigateur
</canvas>
```

Côté Javascript, on va récupérer la balise et la traiter un petit peu.

```
var canvas = document.getElementById("canvas");
var context = canvas.getContext("2d");
```

A partir de ce contexte vous allez pouvoir tout faire. Par exemple :

```
context.drawImage(img, x, y);           //Afficher une image
context.fillStyle = "red";             //Changer la couleur de remplissage
context.fillRect(x, y, width, height);
//Dessiner un carré plein (du coup rouge dans notre cas)
//etc...
```

L'utilisation de canva est très répandue dans les jeux et les pages animées. Pour ce tp on se limitera à cette description

6.1.2 L'élément AUDIO

Il est désormais inutile d'utiliser de tiers plugins pour lire un fichier audio. Pour le moment, seul les navigateurs les plus récents l'interprète.

```
1  <audio autoplay="autoplay" controls="controls">
2      <source src="file.ogg" />
3      <source src="file.mp3" />
4      <a href="file.mp3">Telecharger le morceau.</a>
5  </audio>
Plus simple encore :
```

Autoplay permet la lecture automatique et **controls** affichage des boutons de contrôle de lecture. Le principal problème provient du format audio. Firefox voudrait voir un fichier.Ogg, d'autre avec une extension différente... Cela signifie que, au moins pour l'instant, vous devez créer plusieurs versions du fichier audio.

Tableau 7-1 Reconnaissance des formats audio

	IE6 à IE8	IE9	Firefox	Chrome	Safari	Opera
MP3 (.mp3)		OK		OK	OK	
Ogg Vorbis (.ogg)			OK	OK		OK
Wave (.wav)		OK	OK		OK	OK

6.1.3 L'élément VIDEO

HTML 5 introduit l'élément `<video>` pour insérer des séquences vidéo dans une page web, ce qui offre une alternative aux plug-ins propriétaires d'Apple (QuickTime) ou d'Adobe (Flash) pour les plates-formes ne le reconnaissant pas (iPhone, iPod, iPad...).

Même chose que l'élément `<audio>` le principal problème provient du format vidéo. En tant que tel, lors de l'affichage vidéo HTML5, vous devez offrir plusieurs formats.

```

1 <video controls preload>
2   <source src="maVideo.ogv" type="video/ogg; codecs='vorbis, theora'" />
3   <source src="maVideo.ogv" type="video/mp4; 'codecs='avc1.42E01E, mp4a.40.2'" />
4   <p> Votre navigateur ne prend pas la video en charge <a href="maVideo.mp4">Telecharge la
5 </video>

```

Tableau 7-2 Reconnaissance des formats vidéo

	IE6 à IE8	IE9	Firefox	Chrome	Safari	Opera
H 264 (.mp4)		OK		OK*	OK	OK
Ogg Theora (.ogv)			OK	OK		OK
WebM (.webm)		OK	OK	OK		OK

* Google a annoncé en janvier 2011 qu'il cesserait de prendre en charge le codec H.264 au courant des prochaines versions de Chrome (à partir de Chrome 10 ?).

6.2 Une gestion étendue des formulaires :

Les éléments de saisie de la dernière génération se voient dotés de nouveaux types de champs, de nouvelles fonctions, voire de tests d'expressions régulières sur leur contenu textuel. Je vous en présente les plus prometteurs.

6.2.1 Nouveau type de la balise input

Le HTML5 supprime le recours au JavaScript. On rappelle les nouveaux types tel que :

`email, url, tel, number, range, date, datetime, month, week, time, color`

Chacun des types de champ obéit à une norme qui définit les valeurs acceptées ou rejetées.

Ainsi, il est possible de vérifier quels éléments sont invalides à l'aide des sélecteurs CSS 3

`:valid` et `:invalid`

```
| input[type="email"]:valid {background: green;}
```

6.2.2 Nouveaux attributs

Le HTML5 supprime le recours au JavaScript par de nouveaux attributs précisent le type d'entrée souhaité

- **required** : champ obligatoire (différent de vide)

```
1 | <input type="text" name="nomChamp" required>
```

ou de façon plus explicite :

```
1 | <input type="text" name="nomChamp" required="required">
```

- **autofocus** : confère la focalisation à un champ déterminé dès le chargement de la page.

```
1 | <input type="text" name="nomChamp" required autofocus>
```

- **pattern** : permet d'insérer une expression régulière directement dans notre balisage.

```
1 | <form action="" method="post">
2 |   <label for="nomChamp">Créer un pseudo : </label>
3 |   <input type="text"
4 |     name="username"
5 |     id="username"
6 |     placeholder="4 <> 10"
7 |     pattern="[A-Za-z]{4,10}"
8 |     autofocus
9 |     required>
10|   <button type="submit">Go </button>
</form>
```

Si vous êtes plus ou moins familiers avec les expressions régulières, vous serez conscient du fait que ce modèle: [A-Za-z] {} 4,10 accepte uniquement les lettres majuscules et minuscules. Cette chaîne doit également avoir un minimum de quatre caractères, et un maximum de dix.

Notez que nous commençons à combiner l'ensemble de ces nouveaux attributs, impressionnantes !

Il convient également de noter que tous les navigateurs actuels sont un peu instables quand il s'agit d'interpréter les éléments et attributs qu'ils ne prennent pas en charge ... mais vous pouvez toujours les utiliser avec parcimonie.

Je vous invite à étudier le **chapitre 8** du livre « HTML5, CSS3 & Jquery »

6.3 De nouvelles applications API

La majorité des sites (communautaires, en général) à fort trafic possèdent leur propre API (Application Programming Interface). Mais une API, qu'est-ce-que c'est ? C'est un moyen mis à la disposition de développeurs pour communiquer avec un service web (par exemple une base de données) d'un site et d'en récupérer des informations. Par exemple, l'API d'Allociné permet de récupérer toutes les informations de n'importe quel film, l'API de VieDeMerde permet de récupérer des VDM, etc.

Une large partie des spécifications tentaculaires de HTML 5 est allouée aux applications web (ou *Web Apps*). Ces extensions du langage comportent nativement de nouvelles passerelles de communication vers les protocoles habituels, mais également des fonctionnalités de stockage, de géolocalisation, de travail hors ligne, de messagerie et de traitements de fichiers.

Ces applications, généralement encore à l'état de brouillon (*working draft*), sont ou seront en grande majorité pilotées par JavaScript. En clair, JavaScript, que l'on a cru enterré il y a quelques années, et quelque peu remis au goût du jour grâce à Ajax et aux bibliothèques telles que jQuery, va dorénavant s'imposer comme un langage de tout premier ordre grâce à HTML 5.

6.3.1 Géolocalisation

L'API Geolocation (attention à l'orthographe différente du mot francophone) introduite par le W3C permet aux pages web d'interroger le navigateur sur la position de l'utilisateur, de manière bien plus précise et fiable que les tests de positionnement basés sur l'interprétation de l'adresse IP de la machine.

L'application récupère au travers de l'objet JavaScript `navigator.geolocation` les coordonnées complètes de la position du poste de l'utilisateur, exprimées en longitude, latitude et altitude.

L'API HTML Géolocalisation est utilisée pour obtenir la position géographique d'un utilisateur.

Avant toute chose, vous devez détecter si le navigateur de l'utilisateur permet la géolocalisation. En effet, comme toutes les nouvelles APIs, seuls les navigateurs récents (Firefox 3.5+, Safari 5+, Chrome 5+, IE9+)

```
function trouveLieu() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(montrerPosition);
    } else {
        alert("Ce navigateur ne supporte pas la géolocalisation");
    }
}

function montrerPosition(position) {
    var latitudeLongitude=position.coords.latitude+", "+position.coords.longitude;
    alert(latitudeLongitude)
}
```

ont accès à ces données.

La méthode `getCurrentPosition()` est utilisée pour obtenir la position de l'utilisateur.

Si la géolocalisation est possible, une fonction (dans l'exemple `montrePosition`) va être exécutée avec paramètre l'objet contenant la position (dans l'exemple `position`).

Vous notez que, lors de l'appel de cette méthode, le navigateur va vous demander s'il peut accéder à vos données de géolocalisation. En effet, pour des raisons de vie privée compréhensibles, l'utilisateur peut bloquer l'utilisation de l'API s'il le souhaite.

Les propriétés « `position.coords.latitude` », « `position.coords.longitude` » et « `position.coords.accuracy` » sont toujours renvoyées. Les autres propriétés, par exemple « `position.coords.altitude` », sont retournées seulement si elles sont disponibles.

Maintenant que nous avons récupéré ces données, nous allons pouvoir les utiliser pour les afficher au sein d'une carte. On va le faire avec « `Staticmaps` » de Google, qui est simplement une demande (url) qui nous montre une image statique de Google Maps sur des coordonnées spécifiées. La syntaxe est la suivant:

```
http://maps.googleapis.com/maps/api/staticmap?center=48.84,2.59&zoom=14&size=400x300&sensor=false
```

Application :

Testez la fonction simplement en copiant le lien dans votre navigateur. Notez que « `center` » contient les coordonnées selon la syntaxe `center=latitude,longitude`. Que se passe-t-il quand vous changez la valeur de « `zoom` » ou « `size` » ?

6.3.2 Glisser-déposer : Drag and Drop

L'attribut `draggable`, abordé précédemment, rend un élément déplaçable. Tout son cheminement est traçable via JavaScript : le clic sur l'élément, son trajet, ainsi que l'endroit où il est déposé sur la page à l'aide de l'événement `ondrop`.

Des objets indépendants du document peuvent également être déposés sur la page, par exemple un fichier de votre poste de travail que vous pouvez télécharger sur l'application en l'y faisant tout simplement glisser.

Les fonctionnalités de ce genre, déjà reconnues par l'ensemble des navigateurs y compris Internet Explorer8 vont faciliter les échanges avec JavaScript au point de simplifier considérablement nos habitudes, voire l'ergonomie des pages web.

Le *drag and drop*, ou « glisser-déposer » en français, est l'un des principaux éléments d'une interface fonctionnelle. Il s'agit d'une manière de gérer une interface en permettant le déplacement de certains éléments vers d'autres conteneurs.

Ainsi, dans l'explorateur de fichiers d'un système d'exploitation quelconque, vous pouvez très bien faire glisser un fichier d'un dossier à un autre d'un simple déplacement de souris, ceci est possible grâce au concept du drag & drop.

Bien que cette fonctionnalité ait longtemps existé sur les sites Web grâce au JavaScript, avec HTML5 il est devenu possible de permettre un déplacement de texte, de fichier ou d'autres éléments depuis

n'importe quelle application jusqu'à votre navigateur.

La première chose à faire est de rendre un élément déplaçable. Pour cela, on donnera la valeur « true » à l'attribut draggable.

```
<h1 id="monTitre" draggable="true" >Essayez demedéplacer!</h1>
```

Par exemple :

Maintenant qu'on peut déplacer l'élément, on va utiliser les événements que l'API Drag & Drop nous fournit. Le premier événement est ondragstart qui, comme indique le nom, se déclenche lorsque l'élément ciblé commence à être déplacé.

Par exemple :

```
<h1 id="monTitre" draggable="true" ondragstart="glisse(event)">  
Essayez demedé placer!</h1>
```

La fonction « glisse » spécifié les données qui doivent être glissées. Dans cette fonction on utilisera la méthode dataTransfer.setData() qui définit le type (en général "text" suffit même s'il s'agit d'images) et la valeur des données.

Par exemple, la fonction

```
function glisse(monEvenement) {  
    monEvenement.dataTransfer.setData("text", monEvenement.target.id);  
}
```

permet de déplacer l'objet même (dans l'exemple ci-dessus, le h1 ayant pour id « monTitre »).

Maintenant qu'on a appris comment glisser un élément, on voudrait aussi le déposer quelque part. Par défaut, un élément ne peut pas être déposé dans un autre élément. Pour permettre une exception à cette règle, on utilisera la méthode preventDefault() pour l'événement ondragover.

Par exemple, dans le div suivant

```
<div id="maBoite" ondragover="permetDepose(event)">Déposez icisvp!</div>
```

il est permis de déposer des éléments, pourvu qu'on définisse dans notre fichier .js la fonction :

```
function permetDepose(monEvenement) {  
    monEvenement.preventDefault();  
}
```

Une fois l'autorisation de déposer une donnée, on peut finalement déposer l'élément en utilisant l'événement ondrop(), comme dans l'exemple suivant :

```
<div id="maBoite" ondragover="permetDepose(event)" ondrop="depose(event)">Déposez  
icisvp!</div>
```

où la fonction `depose()` nous permet d'ajouter nos données (stockées dans `dataTransfer`) grâce à la méthode `dataTransfer.getData()`.

Par exemple :

```
function depose(monEvenement) {  
    monEvenement.preventDefault();  
    var idDeLaSource=monEvenement.dataTransfer.getData("text");  
    monEvenement.target.appendChild(document.getElementById(idDeLaSource));  
}
```

6.3.3 Stockage des données : Web Storage

Le **Web Storage** permet au navigateur de stocker des informations de façon locale, sans avoir à communiquer avec un quelconque serveur. Il s'agit-là d'une grande nouveauté, introduite par HTML5. En effet, jusqu'alors, il était compliqué de stocker des informations complexes. L'utilisation des cookies, par exemple, n'était pas idéale : avec une taille limitée, ils n'étaient pas adaptés au stockage local. La solution la plus pratique était de stocker ces données en ligne, grâce à des appels AJAX : pratique lourde en ressources !

HTML 5 introduit la notion de stockage des données via ses API SessionStorage et LocalStorage :

- SessionStorage s'appuie sur JSON (JavaScript Object Notation), qui autorise le stockage d'informations relatives à la navigation de l'internaute dans toutes les pages d'un site (pourvu que la session s'effectue dans la même fenêtre) et ce, pour toute la durée de la session.
- LocalStorage permet, quant à lui, de conserver localement (dans le navigateur) des données d'une session à l'autre : votre navigateur conserve ainsi sur votre machine ces données qui vous sont personnelles. Vous pouvez les réutiliser à tout moment, en toute sécurité.

Reconnue partout depuis Internet Explorer 8 (mais avec une limite de poids des données), une application telle que LocalStorage permet de s'affranchir des techniques actuelles de cookies JavaScript ou sessions PHP pour le stockage des informations de l'utilisateur.

Comme tous les navigateurs ne supportent pas le nouvelles fonctionnalités d'HTML5, la première

```
if(typeof(Storage) !=="undefined") {  
    //CodepourleWebStorage.  
} else {  
    alert("Désolé,maisleWebStorage'estpasupporté");  
}
```

chose à faire est de vérifier que le navigateur supporte le *Web Storage* :

Pour stocker nos données, on utilisera l'objet `localStorage` pour le *Local Storage* et l'objet `sessionStorage` pour le *Session Storage*. La syntaxe est la même pour les deux.

Les méthodes « `setItem()` » and « `getItem()` » nous permettent respectivement de stocker et de récupérer le contenu de l'objet.

Dans la première on spécifie le nom et la valeur de la donnée (`.setItem("nom", "valeur")`), tandis que `getItem` nécessite seulement du nom (la syntaxe est `.getItem("nom")`).

```
localStorage.setItem("ville", "Rovaniemi");
$("#monDiv").html(localStorage.getItem("ville"));
```

Par exemple :

Une syntaxe alternative pour stocker et récupérer les données sans utiliser `setItem` et `getItem` est la suivante :

```
localStorage.nom="valeur";
```

Par exemple :

```
sessionStorage.ville="Rovaniemi";
sessionStorage.nom="PèreNoël";
$("#monDiv").html(sessionStorage.nom+" habite à "+sessionStorage.ville);
```

Remarquez que les noms et les valeurs sont toujours stockés comme des chaînes de caractères.

Même si en principe le **Web Storage** permet de stocker uniquement des chaînes de caractères, avec une petite astuce on peut mémoriser des objets entiers. Pour cela, nous allons utiliser la syntaxe JSON, en transformant des objets en chaîne de caractères, grâce à la méthode `JSON.stringify()`, pour après la ré-parser, grâce à `JSON.parse()`.

Par exemple :

```
//Construction d'un objet JSON
var album={}

//Remplissage de l'objet
album.artiste='Carola';
album.titre='Idennanattblirvarldenny';
album.annee='2007';
album.chansons=[ {"titre":"Stillanatt, heliganatt"}, {"titre":"Lyss tillanglasangensord"}, {"titre":"Glansoversjochstrand"}, {"titre":"Idennanattblirvarldenny"} ];

//On place l'objet en mémoire en transformant le JSON en chaîne de caractères
localStorage.album=JSON.stringify(album);

//Pour récupérer, on transforme la chaîne en JSON
var recupererAlbum=JSON.parse(localStorage.album);
```

6.3.4 Application web hors ligne

Complémentaire aux fonctions de stockage Web Storage et comptant parmi les applications les plus prometteuses de notre ère du Web nomade, l'API Offline Web Application se charge de la mise en cache constante de vos activités, tant et si bien qu'il devient possible de continuer à faire tourner les fonctionnalités tout en étant déconnecté d'Internet.

Que vous soyez hors ligne ne vous empêche plus de rédiger des courriers de messagerie, de planifier votre agenda, ni de rédiger des documents ou présentations destinés à être partagés avec d'autres personnes : dès lors que votre terminal trouvera un point de connexion Internet, vos travaux en cours se synchroniseront ou seront expédiés automatiquement si le développeur le prévoit.

6.3.5 Web Socket

L'API Web Sockets représente une connexion de poste bidirectionnelle permanente à un serveur via le protocole TCP (*Transmission Control Protocol*). En clair, les informations véhiculées par le serveur sont réceptionnées en temps réel par tous les postes connectés, et inversement. Parmi les applications les plus courantes de cette API, on trouve les discussions instantanées (tchat) ou encore les vidéoconférences avec échanges de données. La technologie employée côté serveur est une bibliothèque JavaScript utilisant **Node.js** ; côté client, on utilise directement l'API native du navigateur, ce qui fait que cette fonctionnalité ne s'exécute actuellement que sur les moteurs WebKit et Gecko très récents.

Exercice Pratique :

Exercice 1:

- Créer ce formulaire de contact pour lequel les nouveaux champs des formulaires HTML5 sont bien mis en évidence :

Contact

Be called back for more information

First name (*)	<input type="text"/>
Last name (*)	<input type="text"/>
You want to be contacted by	<input type="text"/> Phone <input checked="" type="checkbox"/>
Phone number	<input type="text"/>
When (*)	<input type="text"/> <input type="button" value="Calendar"/>
Between (*)	<input type="text"/> 07:00 PM <input type="button" value="Clock"/> and <input type="text"/> 07:00 PM <input type="button" value="Clock"/>
<input type="button" value="Cancel"/> <input type="button" value="Submit"/>	

- Dans ce formulaire de contact, on veut stocker le nom de l'utilisateur pour l'afficher dans la page suivante. Utilisant alors l'API SessionStorage :

```
<input type="text" placeholder="Name" required
oninput="sessionStorage.setItem('username', this.value);" name="Name">
Il est alors possible d'accéder à la valeur « username » depuis une autre page avec :
sessionStorage.getItem('username');
```

- Ajouter à ce formulaire la carte des magasins les plus proches de l'utilisateur. Pour cela, on utilise le service Google Maps et on passe en paramètre la latitude et la longitude :

```
if (navigator.geolocation) {
  navigator.geolocation.getCurrentPosition(successFunction, errorFunction);
} else {
  alert('No Geolocation.');
}
function successFunction(position) {
var map = new google.maps.Map(document.getElementById('google_canvas'),
{ zoom: 15, mapTypeId: google.maps.MapTypeId.ROADMAP });

var geolocate = new google.maps.LatLng(position.coords.latitude,
position.coords.longitude);

var infowindow = new google.maps.InfoWindow({ map: map, position: geolocate,
content: '' });
map.setCenter(geolocate);
```

Exercice 2 :

Modifiez le fichier `recherche.html` ci-joint à ce TD afin de permettre le glissement de l'image de la madeleine. Quand on commence à glisser le gâteau, l'arrière plan deviendra bleu, tandis que quand on relâche la souris l'arrière plan deviendra rouge (pensez à utiliser l'événement `ondragend`).

Exercice 3 :

Créez un élément `div` ayant comme arrière plan une image d'une tasse de thé (cherchez sur Internet l'image que vous préférez).

En utilisant les exemples ci-dessus, modifiez le fichier `recherche.html` pour permettre de tremper votre madeleine dans le thé.

Exercice 4 :

Créez un compteur de visites d'une page web. Vous devrez afficher le nombre de venues (c'est-à-dire le nombre d'actualisations de la page) de l'utilisateur. Pensez à utiliser `Number()` pour convertir l'objet en nombre.

Le résultat est-il le même si on utilise `localStorage` ou `sessionStorage` ?

Exercice 5 :

À l'aide d'un formulaire, demandez à l'utilisateur la première fois qu'il entre dans votre site (et que la première fois), son titre de civilité, son prénom et son nom. Modifiez le message de l'exercice précédente pour afficher aussi ces informations (en affichant, par exemple, " Bonjour M. Matti Meikäläinen, vous avez visité cette page 42 fois.").

Exercice 6 :

En vous appuyant sur cette technique, vous allez créer une application web de « to-do list ».

Les tâches à réaliser seront mémorisées dans le navigateur (en *Local Storage*) et listées sur la page, accompagnées d'un formulaire permettant à l'utilisateur d'en ajouter.

– Commencez par créer un formulaire pour l'ajout de tâche. Il sera composé du titre de la tâche, de la date à respecter pour sa réalisation (utilisez un `datePicker`), d'une description plus complète (utilisez un `<textarea>`) et d'un état d'avancement (0-100%) (utilisez un `<input>` de type `range`).

– Traitez ce formulaire grâce à jQuery. À la soumission de ce dernier, récupérez-en les valeurs pour les stocker dans le *Local Storage*, sous forme d'objets JSON.

– Listez les tâches déjà ajoutées. Vous pourrez utiliser du CSS3 pour un rendu en forme de post-its, par exemple. Et pourquoi pas des animations au survol?

– Faites en sorte de pouvoir modifier l'état d'avancement d'une tâche en cliquant sur un bouton.

Une fois effectuée, son affichage devra être modifié (contenu barré ? Changement de couleur ?)

– Ajoutez un bouton pour chaque tâche à fin de la supprimer.

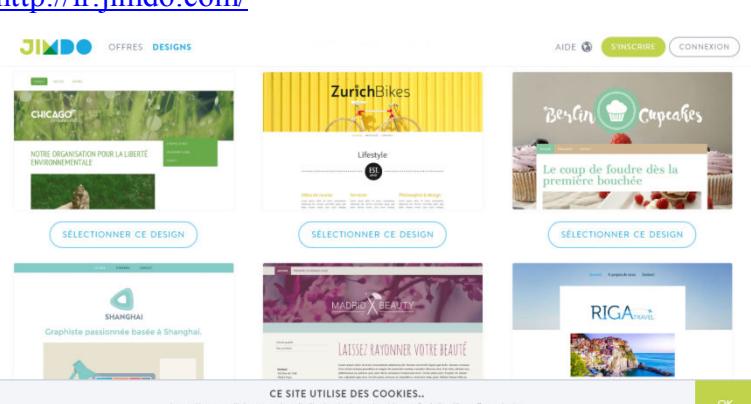
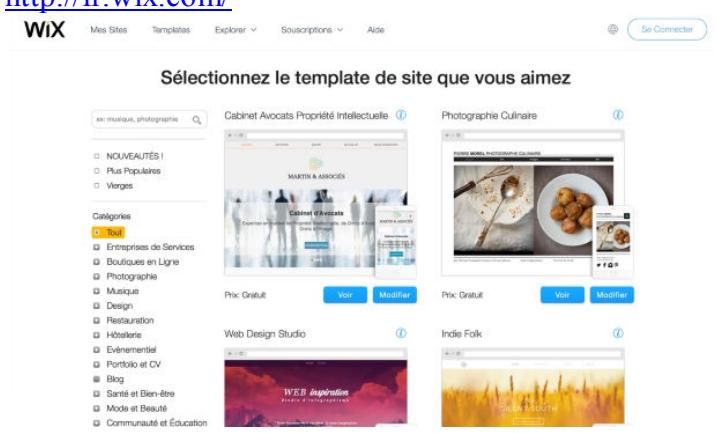
6.4 Les outils et logiciel de création de site Web

6.4.1 Plateforme en ligne

Il existe des outils en ligne de création de site Web, qui vous permet de créer des pages Web en déplaçant différents éléments de page (images, textes, etc.) sans l'écriture ni l'accès au code. Il fonctionne en mode glisser-déposer (drag-and-drop), vous pouvez ainsi laisser libre cours à votre créativité. Il est désormais possible sans aucune compétence « technique » de créer un site esthétiquement et fonctionnellement d'un très bon niveau.

Lors de la création de votre site vous aurez le choix parmi des centaines de thèmes de tous styles, gratuits et bien sûr payants. Privilégiez une esthétique épurée qui saura mettre en valeur votre contenu. Pour que votre site soit considéré comme compatible avec les appareils mobiles, vous devez bâtir une version pour les écrans de différentes tailles et résolutions (**Responsive Web design**).

Voici trois sites internet (parmi les plus populaires, mais il en existe bien d'autres) qui vous proposent un site gratuit.

<p>http://fr.jimdo.com/</p> 	<p>http://fr.wix.com/</p> 
<p>https://www.e-monsite.com</p> 	

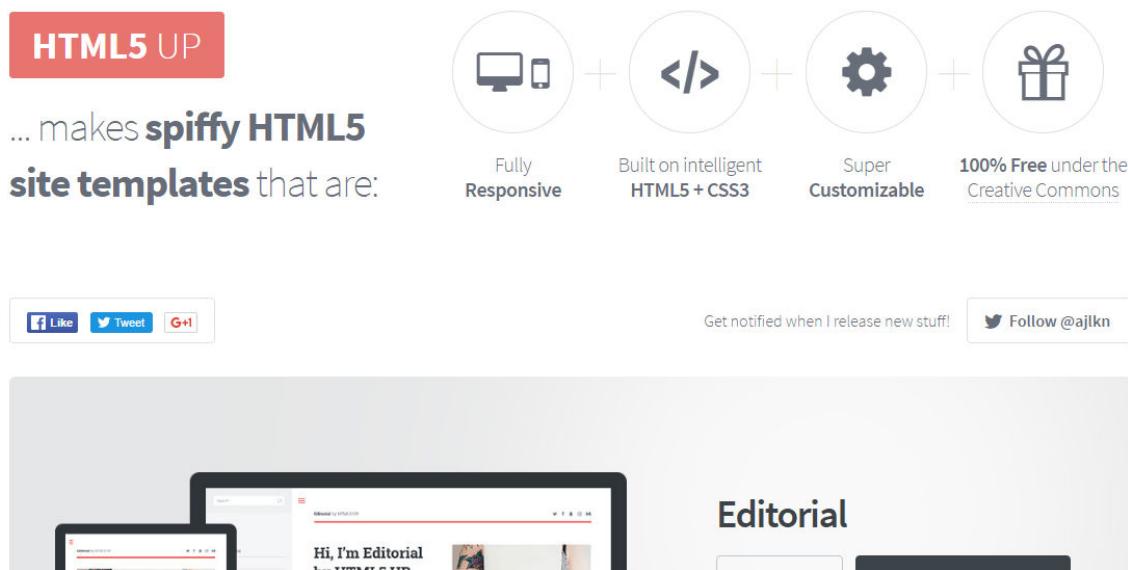
Les outils présentés ci-dessus vous proposent d'héberger votre site Web créé sur leurs serveurs. Le service est gratuit à condition de garder des liens de publicité, avec un nom de domaine non personnalisé. Comme il n'est pas possible de récupérer le code de votre site.

Toutefois, si vous voulez demeurer maître de vos données et de leur emplacement, vous pourriez acheter les services d'hébergement ou bien privilégier un outil qui vous donne l'option de créer et récupérer le code de votre site et d'utiliser le service d'hébergement de votre choix.

6.4.2 Les modèles prêt à l'emploi (thèmes)

Un Template ou thème ou encore modèle graphique est un moyen de séparer le contenu rédactionnel (contenu textuel) de sa forme . Un Template fait donc office de gabarit (modèle) où seuls certains éléments sont modifiables (le contenu texte, les images, les couleurs et le fond).

Vous pouvez télécharger des modèles (template) prêt à l'emploi payant ou gratuit disponible sur le net. Telque : <http://www.templatemonster.com> ou <https://html5up.net>



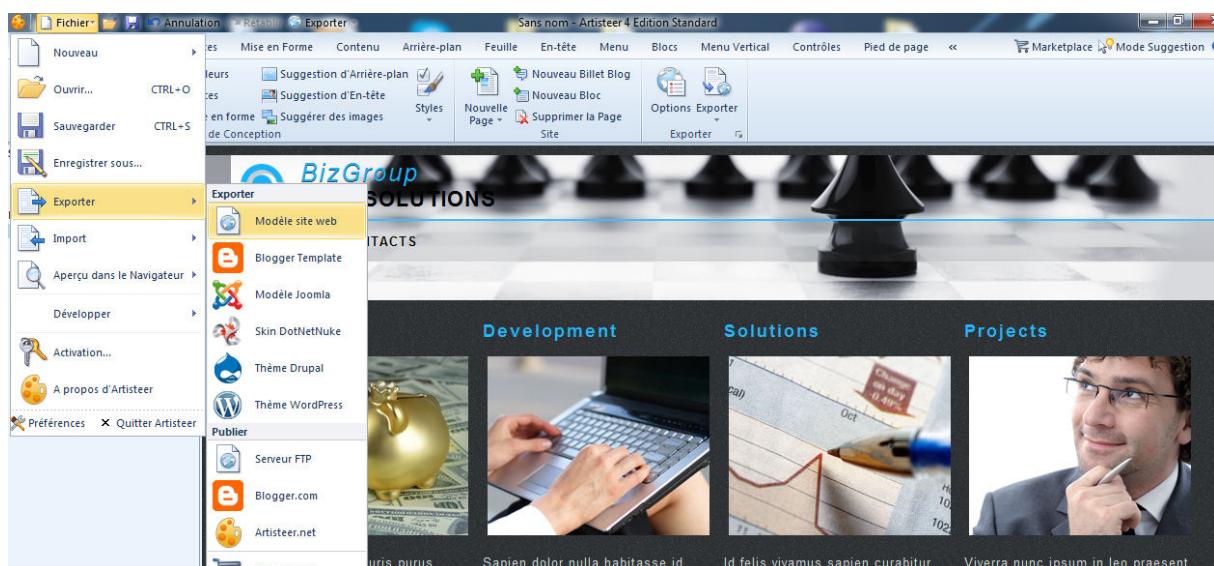
L'utilisation d'un Template facilite la conception d'un site Internet et sa mise à jour, aussi bien en terme de contenu que sur sa présentation, tout en restant dans le cadre imposé par le Template avec ses limites. Le risque d'avoir le même site que votre voisin n'est pas exclu.

Si votre projet impose un design particulier, des évolutions fréquentes, une arborescence spécifique... le Template ne pourra pas satisfaire toutes vos exigences, le sur-mesure sera plus adapté.

Le logiciel « Artisteer » est suggéré pour créer vos propres templates. Jetez donc un oeil sur la démo animée pour vous faire une idée. <http://www.artisteer.com/?p=demo>

Tout ce que vous pouvez accomplir avec une feuille de style et du code HTML est réalisable avec Artisteer, à la seule différence près qu'au lieu de travailler dans le code vous créez votre design avec des boutons, et que le logiciel se charge de vous produire les pages, les images, la CSS et le fichier JavaScript qui vont bien, le tout parfaitement rangé dans un dossier avec les bons répertoires.

Pour récupérer le code du site vous devez exporter le modèle site web.



Application 3: Créer le modèle de votre site soit en se basant sur un template téléchargé du net tel que html5up, soit créée par vous-même avec Template Toaster, ou avec les modèle de Dreamweaver.

6.5 Valider un site

Pour réaliser un site conforme aux standards du Web, il faut :

1. Préciser la version de HTML ou de XHTML utilisée. On indiquera aux navigateurs la version du HTML retenue pour un document donné en préférant celui-ci d'un doctype. Ce doctype est un code spécifique placé au tout début d'un document HTML. Il précise au navigateur le langage retenu pour l'écriture de la page : HTML ou XHTML, en version stricte ou transitionnelle, etc.

C'est non seulement un moyen de s'assurer que le document sera bien interprété par les navigateurs, mais aussi un élément indispensable à sa validation par l'organisme de référence, le W3C.

Le doctype est précisé dans la balise <html>, la spécification de la langue est précisée dans la balise <html>. L'encodage est spécifié dans l'espace délimité par les balises <head> et </head>, à l'aide de l'attribut charset. Par exemple :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
    <head>
        <title>titre évocateur</title>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-15" />
    </head>
```

```
<!DOCTYPE html>
<html lang="fr" />
    <head>
        <title>titre évocateur</title>
        <meta charset=utf-8 />
    </head>
```

2. Se conformer à la grammaire retenue. Pour l'aspect syntaxique, les validateurs du W3C conviendront, mais aucun programme ne peut garantir ou contrôler la bonne sémantique des balises d'un document.

Comme le validateur (X)HTML (<http://validator.w3.org/>) ou le validateur CSS (<http://jigsaw.w3.org/css-validator/>), ou le validateur HTML5 <http://html5.validator.nu>, des outils en ligne vous aideront à vous conformer aux règles du langage correspondant au doctype choisi pour le document.

Google aussi propose un validateur et vous propose un ensemble de directives à suivre pour corriger votre site.

6.6 Hébergement du Site Statique

Vous êtes maintenant arrivé au terme de votre apprentissage. Quel chemin parcouru ! Vous êtes désormais capable de définir vos propres pages Web en y incluant du code HTML5, CSS3 et jQuery. Pour l'instant, ce code fonctionne en local sur votre ordinateur.

Il est temps de le rendre accessible sur le Web. Tout ce que vous devez savoir à ce sujet se trouve dans le chapitre 15 « mise en ligne d'un site », du livre « HTML5, CSS3 & jQuery » de Michel Martin fourni en classe. Il est temps de passer à la pratique.

1. Crédation d'un nouveau compte

www.hostinger.fr est un hébergeur gratuit. On l'utilisera pour héberger nos Sites.

Créer avant tout un compte utilisateur sur www.hostinger.fr puis se connecter et choisir dans le menu Hébergement « nouveau compte » puis commander le plan gratuit

<http://cpanel.hostinger.fr/>

The screenshot shows the 'New Hosting Account' page on the Hostinger website. The 'Gratuit' plan is selected, showing a price of 0,00 € and a 'Commander' button highlighted with a red box. Other plans shown are 'Premium' at 2,41 € and 'Business' at 5,63 €. The page includes a sidebar with options like 'Installateur' and 'Gagner de l'argent'.

Plan	Prix	Action
Gratuit	0,00 €	Commander
Premium	2,41 €	Commander
Business	5,63 €	Commander



Entrer le domaine et le mot de passe

Choisir le type de domaine:	Sous domaine
Sous domaine	VotreNom .esy.es
Mot de passe *	***** Générer
Confirmer le mot de passe *	***** Entrer le mot de passe à nouveau
Continuer	

Conservez votre mot de passe dans un support papier

≡ Order New Hosting "Gratuit" - Step 3 of 3

1 Choisir le plan

2 Hosting Setup

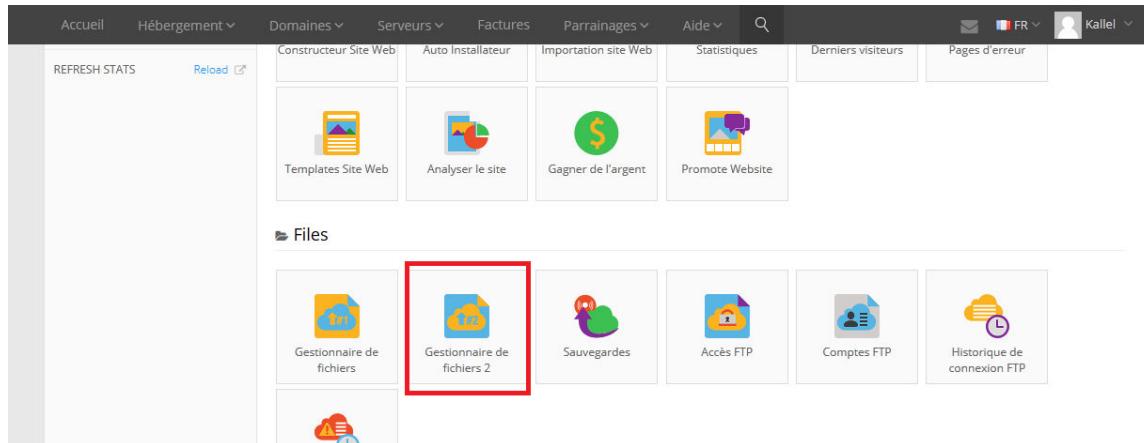
3 Résumé de la commande

Confirmier votre commande

Plan:	Gratuit
Domaine:	VotreNon.esy.es
Prix:	0,00 €
<input checked="" type="checkbox"/> Je suis d'accord avec termes et conditions	
Retour Commander	

2. Transfert des fichiers

Une fois que votre commande est réalisée, choisir « Gestionnaire de fichier 2 » pour télécharger votre dossier du site « SitePerso » des premières séances. Mettre le dossier du site en archive .ZIP



Dans le dossier « public_html » créer un nouveau dossier « Applications »

All	Name	Type	Size	Owner	Group	Perms	Mod Time	Actions
<input type="checkbox"/>	Up..							View Edit Open
<input type="checkbox"/>	public_html	Directory	4096	372325706	u372325706	rwxr-xr-x	Feb 24 21:29	View Edit Open
<input type="checkbox"/>	DO NOT UPLOAD HERE DO NOT UPLOAD HERE	File	0	372325706	u372325706	rwr--r--	Dec 4 11:35	View Edit Open

Directories: 1
Files: 1 / 0 B
Symlinks: 0
Unrecognized FTP output: 0

All	Name	Type	Size	Owner	Group	Perms	Mod Time	Actions
<input type="checkbox"/>	Up..							View Edit Open
<input type="checkbox"/>	.quarantine	Directory	4096	372325706	u372325706	rwxr--r--	Feb 24 21:23	View Edit Open
<input type="checkbox"/>	.tmb	Directory	4096	372325706	u372325706	rwxrwxrwx	Feb 24 21:23	View Edit Open
<input checked="" type="checkbox"/>	SiteStatique	Directory	4096	372325706	u372325706	rwxr--r--	Feb 24 21:54	View Edit Open
<input type="checkbox"/>	file-manager	Directory	4096	372325706	u372325706	rwxr--r--	Feb 24 21:22	View Edit Open
<input type="checkbox"/>	Go to the subdirectory SiteStatique	File	115	372325706	u372325706	rwr--r--	May 20 2011	View Edit Open
<input type="checkbox"/>	default.php	PHP script	6996	372325706	u372325706	rwr--r--	Sep 3 19:42	View Edit Open

Dans le dossier « Application » faites un upload et choisir votre fichier .zip puis validez le transfert

Le site alors est opérationnel à cette URL :
<http://votrenom.esy.es/Application/siteperso/index.htm>