
Les bases du HTML

Les bases du HTML

Sommaire

1. Introduction	3
2. HTML 5	3
3. Un modèle client-serveur	4
4. Les outils nécessaires à l'édition de pages web	4
5. Votre première page web	5
6. La structure d'une page web	7
7. Les principales balises sémantiques de HTML5.	8
8. URL - Uniform Resource Locator	8
9. La balise lien	10
10. QCM	11
11. Résumé	11

1. Introduction

Même si les ordinateurs communiquent à grande échelle depuis les années 60, la naissance de l'internet moderne, tel qu'on le connaît aujourd'hui, date du début des années 90 et de la création du World Wide Web, aussi appelé web. Avec le web, sont arrivés le protocole réseau http (pour HyperText Transfer Protocol, le protocole géré par votre navigateur) permettant de transférer des données entre ordinateurs, la notion l'URL (pour Uniform Resource Locator) permettant d'identifier une ressource sur le web, et le langage HTML (pour HyperText Markup Language) permettant de décrire une page web, et incluant des images, des vidéos, des tableaux, des formulaires, des liens vers d'autres pages web, etc...

2. HTML 5

La définition du langage HTML (*HyperText Markup Language*), portée par le consortium W3C (pour World Wide Web Consortium, le groupement d'industriel régissant les normes du web), a aboutie en 1997 à une version 4 qui se voulait définitive mais n'a pas su évoluer.

Mais depuis, les usages du web ont explosé. Nous pouvons tweeter et visionner des vidéos sur nos smartphones, échanger avec nos amis sur Facebook, gérer nos relations professionnelles sur LinkedIn.

En 2004, la situation a évolué avec la sortie des navigateurs Firefox puis de Chrome et la création par Apple, Mozilla et Opera du groupe de travail WHATWG (pour Web Hypertext Application Technology Working Group) bientôt rejoint par le W3C pour aboutir sur la spécification de HTML5 qui s'est finalement imposée comme la nouvelle norme de l'HTML.

Autour de HTML, de nouvelles technologies sont présentes. Côté serveur, PHP, Python, ASP/.net ou NodeJS permettent de générer du contenu dynamique. JavaScript s'exécute sur le client et s'accompagne des bibliothèques de haut niveau, en particulier JQuery, ainsi que de framework (des environnements) comme Angular, React et VueJs.

Activité Web-conférence : pour aller plus loin, nous listerons quelques fonctionnalités ainsi que leur support par les navigateurs.



3. Un modèle client-serveur

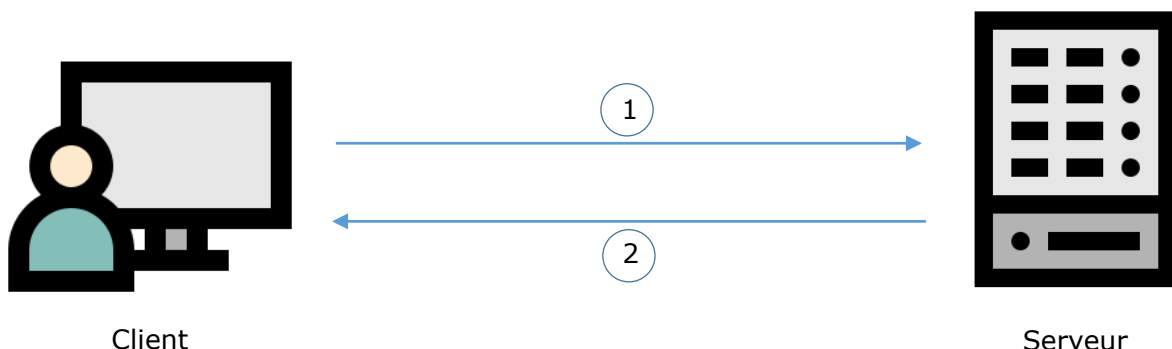
Sur internet, chaque machine est accessible via une adresse unique appelée adresse IP. Des protocoles réseau utilisent ces adresses IP pour communiquer : ce sont les règles qui régissent les échanges entre les différents services et logiciels.

Définition : Un protocole de communication est un ensemble de règles et de procédures permettant de définir un type de communication particulier. Les protocoles sont hiérarchisés en couches, pour décomposer et ordonner les différentes tâches. Il existe plusieurs familles de protocoles ou modèles, chaque modèle étant une suite de protocoles entre diverses couches. Parmi ces modèles on trouve le OSI et le TCP/IP.

- http://www.zeitoun.net/articles/les_protocoles_reseaux/start
- https://fr.wikipedia.org/wiki/Protocole_de_communication

Parmi ces protocoles, on compte IMAP (pour Interactive Message Access Protocol) qui permet d'accéder aux courriers électroniques, ou HTTP (pour HyperText Transfer Protocol) qui régit les échanges pour accéder au Web et donc l'accès aux sites internet. C'est un modèle client-serveur où :

- 1) Le navigateur (le client) va faire une requête au serveur web (logiciel en permanence à l'écoute) pour demander les ressources (pages web, images, vidéos, ...) dont il a besoin.
- 2) Le serveur va fournir une réponse en transmettant soit le contenu demandé, soit un code d'erreur. Par exemple, l'erreur 404 indique que la ressource n'est pas disponible.



4. Les outils nécessaires à l'édition de pages web

Avant de créer votre première page web, il convient de disposer de deux logiciels :

- Un navigateur web moderne (Firefox, Chrome, ...)
- Un éditeur de texte permettant de modifier les pages web, CSS et JavaScript. Le logiciel Visual Studio Code sera choisi pour ce cours car il est largement utilisé par les professionnels et qu'il est gratuit. Vous pouvez aussi lui ajouter les extensions pour sauvegarder directement vos projets dans GitHub.

5. Votre première page web

Votre première page web est disponible à l'adresse suivante :

http://www.action-creation.fr/cnam/HTML_CSS/chapitre1/pageType/index.html

Ouvrez cette page dans votre navigateur et affichez-en le code source. Pour cela, une fois la page affichée, utilisez le bouton droit sur la page puis « [afficher le] code source de la page ».

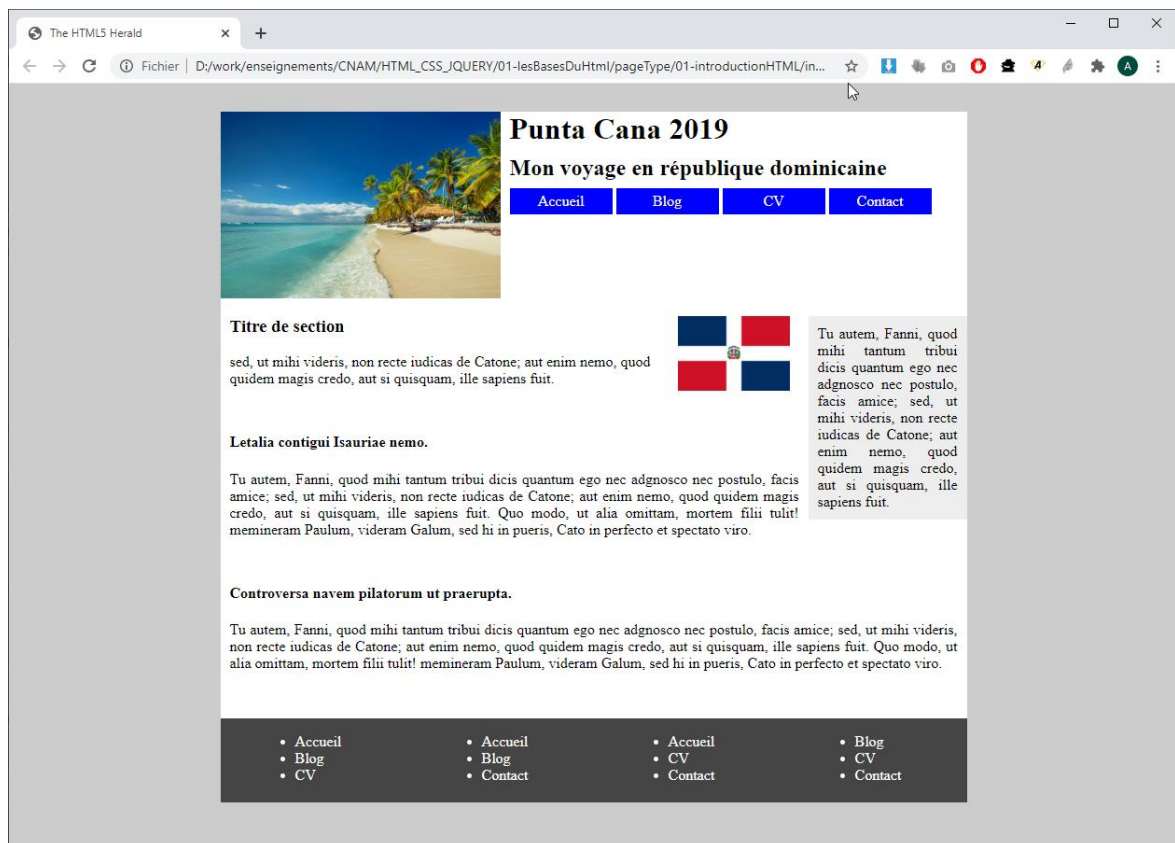
Vous pouvez aussi la télécharger, ainsi que toutes ses dépendances, à l'adresse suivante :

http://www.action-creation.fr/cnam/HTML_CSS/chapitre1/pageType.zip

Une fois l'archive extraite sur votre machine, ouvrez la page `index.html` (c'est le nom de la page web par défaut).

Exercice 1 :

Lancez votre navigateur web préféré et ouvrez la page *index.html*



Ouvrons maintenant le site dans l'éditeur *Visual Studio Code* :

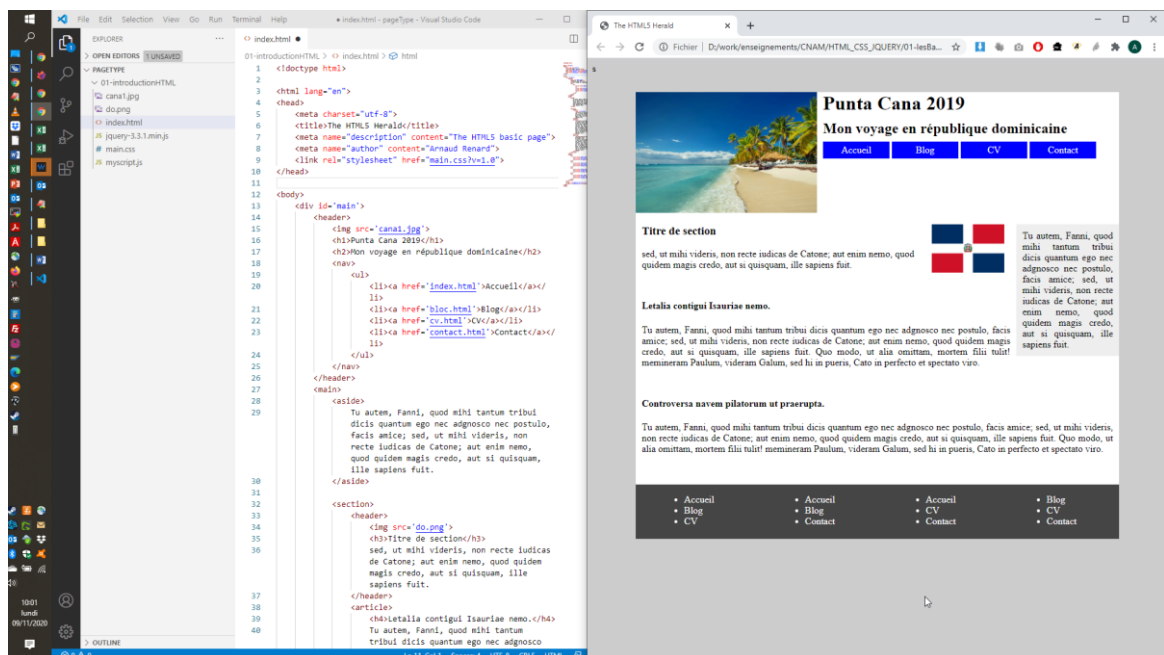
- Quelques personnalisations sont possibles :
 - Changer le thème : File>Preference>Color Theme> Light
 - Alt+Z permet d'activer ou de désactiver les sauts de ligne dans l'éditeur
- Ouvrir le répertoire dans lequel le site est enregistré : *file > Open Folder*
 - La liste des fichiers apparait ensuite sur la gauche.
- Double-cliquez sur le fichier index.html
 - Le document s'ouvre dans l'éditeur :

```

1 <!doctype html>
2
3 <html lang="en">
4 <head>
5   <meta charset="utf-8">
6   <title>The HTML5 Herald</title>
7   <meta name="description" content="The HTML5 basic page">
8   <meta name="author" content="Arnaud Renard">
9   <link rel="stylesheet" href="main.css?v=1.0">
10 </head>
11
12 <body>
13   <div id="main">
14     <header>
15       
16       <h1>Punta Cana 2019</h1>
17       <h2>Mon voyage en république dominicaine</h2>
18       <nav>
19         <ul>
20           <li><a href="index.html">Accueil</a></li>
21           <li><a href="blog.html">Blog</a></li>
22           <li><a href="cv.html">CV</a></li>
23           <li><a href="contact.html">Contact</a></li>
24         </ul>
25       </nav>
26     </header>
27     <main>
28       <aside>
29         Tu autem, Fanni, quod mihi tantum tribui dicis quantum ego nec agnosco nec postulo, facis amice;
30         sed, ut mihi videris, non recte iudicas de Catone; aut enim nemo, quod quidem magis credo, aut si
31         quisquam, ille sapiens fuit.
32       </aside>
33     </main>
34   </div>
35 </body>
36 </html>

```

Pour une bonne visibilité, il est conseillé, au moins dans un premier temps, de positionner deux fenêtres l'une à côté de l'autre, comme la capture ci-dessous :



Dans notre activité de développement de pages web, nous allons très régulièrement passer d'une page à l'autre :

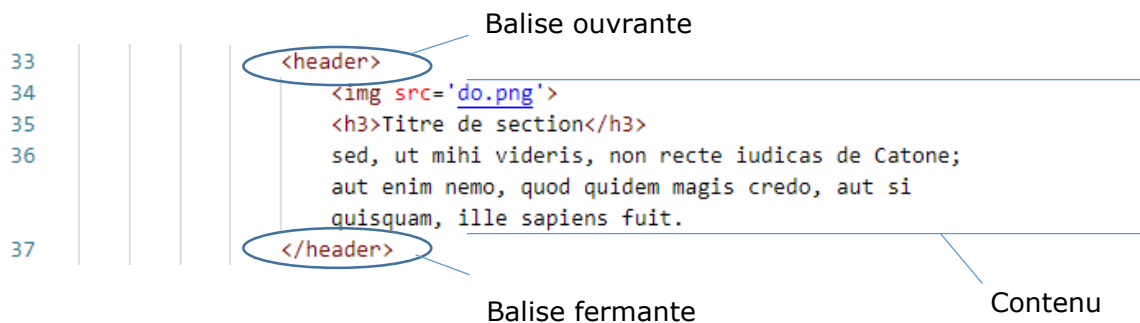
- Dans Visual Studio Code
 - Modification du code html
 - Je vous invite à changer le contenu de la ligne 16 : remplacez 2019 par 2020 !
 - Enregistrement des modification (Ctrl+s)
- Changer d'application (Alt+tab)
- Dans votre navigateur
 - Recharger la page (Ctrl+r)
 - Vérifier les modifications
- Changer d'application (Alt+tab)

Pour gagner du temps au cours de vos travaux, il est intéressant de vous familiariser avec ces 3 raccourcis.

6. La structure d'une page web

A la lecture du code source de la page web, vous pouvez remarquer que la page est structurée de manière arborescente par des éléments qui contiennent d'autres éléments.

Chaque élément est délimité par une balise ouvrante et une balise fermante. Par exemple, pour la balise *header* présente dans la balise *section* :



Entre les deux balises, on trouve le contenu de l'élément : du texte, ainsi que d'autres balises, ...

Exception : quelques balises sont auto-fermantes et il ne sera pas nécessaires de créer une balise fermante, c'est le cas de la balise *img* qui permet d'intégrer une image dans la page.

Indentation : Afin d'optimiser la lecture du code source et dans le but d'éviter les erreurs, on prendra soin d'écrire la balise ouvrante et la balise fermante au même niveau par rapport à la marge de gauche, et à décaler le contenu vers la droite (d'une tabulation ou de 4 espaces)

Exercice 2 : listez les balises présentes dans la page.

Web-conférence : au cours de la séance, nous proposerons une arborescence représentant la structure de cette page.

7. Les principales balises sémantiques de HTML5.

Parmi ces balises, certaines sont dites sémantiques car elles ont une signification. Voici les principales balises sémantiques de HTML5 :

Nom	Détail
<code><section></code>	Section générique regroupant un même sujet (que ces voisins), une même fonctionnalité, de préférence avec un en-tête.
<code><article></code>	Section de contenu indépendante (de ses voisins), pouvant être extraite individuellement du document ou syndiquée (flux RSS ou équivalent), sans pénaliser sa compréhension
<code><nav></code>	Section possédant des liens de navigation principaux (au sein du document ou vers d'autres pages)
<code><aside></code>	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.
<code><header></code>	Section d'introduction d'un article, d'une autre section ou du document entier (en-tête de page).
<code><footer></code>	Section de conclusion d'une section ou d'un article, voire du document entier (pied de page).

L'ensemble des balises sont décrites dans l'annexe 1 :

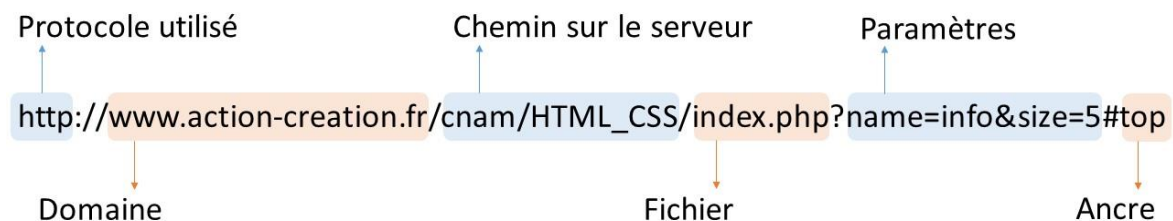
<https://jaetheme.com/balises-html5>

8. URL - Uniform Resource Locator

Afin de localiser une ressource sur internet, on utilise une URL (Localisation Uniforme de Ressource, ou Uniform Resource Locator). Une ressource pouvant être une page web, une image, une vidéo, un fichier son, un fichier de feuille de style ou un fichier *JavaScript*.

URL Absolue

Une URL absolue fournit une identification complète de la ressource :



Voici le détail :

- **http** : le protocole réseau utilisé pour accéder à la ressource. Https est la version chiffrée du protocole http
- **:** : le séparateur après la déclaration de protocole.
- **//** : début de définition d'un chemin d'accès
- **www.action-creation.fr** : nom de domaine correspondant à un serveur. Peut aussi être remplacé par l'adresse IP du serveur.
- **/cnam/HTML_CSS/** : le chemin de répertoires où se trouve la ressource (/ permet de séparer deux noms de répertoires)
- **index.php** : nom du fichier

- **name=info&size=5** : Les paramètres envoyés au serveur, en mode GET. Les paramètres débutent par un point d'interrogation (?) sont notés sous la forme *clé=valeur* séparés par une esperluette (&).
- **top** : L'ancre permet de créer un lien vers un endroit précis d'une page web

URL Relative

Une **URL relative** permet de localiser une ressource par rapport au contexte (généralement la page qui utilise ces URL pour désigner une autre page via un lien ou inclure par exemple une image, une feuille de style ou un fichier JavaScript). Dans ce contexte, .. (point-point) désigne le répertoire supérieur et . (point) désigne le répertoire courant.

Dans le contexte de la page désignée ci-dessus, voici des URL relatives et les URL absolues correspondants

- **info.html** ou **./info.html** :
http://www.action-creation.fr/cnam/HTML_CSS/info.html
- **../all.php** :
<http://www.action-creation.fr/cnam/all.php>

Conseil : Pour donner l'adresse d'un site, on donnera l'adresse absolue, mais une fois sur le site, on privilégiera les URL relatives, ce qui permettra au développeur de préparer le site sur son ordinateur puis de le déplacer sur le serveur sans mettre à jour les URL.

Attention : `C:\User\toto\Desktop\info.html` n'est pas une URL, c'est le chemin du fichier sur votre ordinateur. Les internautes n'ont pas cette image sur leur disque dur. Dans votre page web, il faut donner une URL (qui correspond à un chemin sur un serveur) et pas un chemin sur votre disque dur.

Attention : les espaces et autres caractères spéciaux (accents, etc ...) ont une interprétation différente d'un ordinateur à l'autre, on ne les utilisera pas dans les noms des fichiers ou dossiers sur les serveurs.

Exercice 3 :

Pour cet exercice, nous allons modifier la page type présentée précédemment.

- Ajoutez une seconde section en dessous de la première. Elle contiendra 3 articles. Chaque article est composé d'un titre et d'une image spécifique. Vous trouverez des images à l'aide d'un moteur de recherche.
- Mettez toutes les images dans un répertoire « images ». Modifiez le code HTML pour prendre en compte les modifications.
 - Indice : il faut modifier les url qui se terminent par `.jpg` et `.png`
- Créez un répertoire « css » et placez-y le fichier .CSS actuel. Modifiez le code HTML pour prendre en compte les modifications.
- Question : quelle est la balise utilisée pour intégrer un fichier de style CSS. A quel endroit de la page l'utilise-t-on ?
- Faites la même chose que c) avec le fichier JavaScript (.js) et un dossier « js »
- Question : quelle est la balise utilisée pour intégrer un fichier JavaScript « .js ». A quel endroit de la page est-ce utilisé ?

9. La balise lien

Dans notre page, dans l'élément de navigation (nav), est présent une liste non numérotée (balise *ul*) qui contient quatre items (*li*) qui contiennent chacun un lien (balise *a*). La balise *a*, utilisée comme un lien, nécessite l'ajout d'un attribut *href* pour indiquer l'url de destination du lien.

```
18 <nav>
19   <ul>
20     <li><a href='index.html'>Accueil</a></li>
21     <li><a href='blog.html'>Blog</a></li>
22     <li><a href='cv.html'>CV</a></li>
23     <li><a href='contact.html'>Contact</a></li>
24   </ul>
25 </nav>
```

Dans notre cas, chaque lien renvoie vers une page différente : index.html (c'est notre page modèle), blog.html, cv.html et contact.html.

Exercice 4 :

Copiez 3 fois la page index.html, pour créer les trois autres pages.

Modifiez les textes contenus dans les balises (les titres, les textes, ...) ainsi que les images pour créer un site complet :

- la page Blog parle d'une de vos passions,
- la page CV contient votre curriculum vitae,
- et la page contact donne les éléments pour vous contacter. Elle sera complétée ultérieurement avec un formulaire de contact.

Exercice 5 :

Maintenant que nous connaissons les balises utilisées pour intégrer des fichiers Javascript et des fichiers CSS, et que nous savons à quel endroit les intégrer dans le site, nous allons pouvoir récupérer des fichiers sur internet pour les intégrer à notre site.

Nous allons le faire pour la bibliothèque Bootstrap (une bibliothèque est un ensemble d'outils)



- Trouver le site internet contenant la documentation de Bootstrap et permettant de le télécharger.
- Parcourez le site, les exemples, la doc et le « Getting Started »
- Télécharger la dernière version compilée (dans le sens « agrégé ») de l'outil
- Pour l'intégrer à notre arborescence de fichier, nous allons mettre Le répertoire « bootstrap-xxx » dans un répertoire « vendor » (dans lequel il est usage de mettre toutes les bibliothèques nécessaires à notre page web que l'on aurait pu télécharger.)
- Intégrer *bootstrap* à la page CV.

10.QCM

- A - Une URL permet de décrire :
1. Une page web en respectant l'indentation
 2. Une ressource sur le web
 3. L'adresse d'une image sur l'ordinateur de l'internaute
 4. Un cri strident en cas de douleur
- B - La balise sémantique *section* contient
1. Des informations en relation avec ses voisins
 2. Des informations indépendantes pouvant être syndiqué
 3. Obligatoirement un *header* et un *footer*
 4. Des informations sur le groupe Section d'Assaut
- C - La balise *img* permet d'intégrer dans une page :
1. Un tableau
 2. Une image
 3. Un fichier CSS
 4. Un fichier JavaScript

11.Résumé

Au cours de ce premier chapitre, nous avons appris à modifier une page web, ainsi que son contenu, et ses images. Les grands principes d'une page HTML sémantique ont été présentés et un premier site web a pu être réalisé.