

# Création d'un site web

## Table des matières

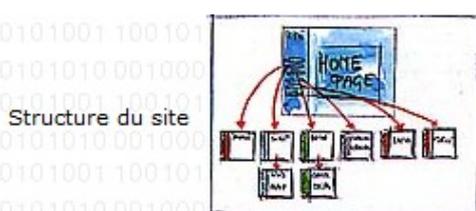
1. Plan du site.....	2
2. Maquettage du design.....	1
3. Organiser le contenu en HTML.....	2
4. Mettre en forme en CSS.....	5
4.1. Les polices personnalisées.....	5
4.2. Définition des styles principaux.....	6
4.3. En-tête et liens de navigation.....	6
4.4. Les liens de navigation.....	7
4.5. La bannière.....	8
4.6. Le corps.....	10
4.7. Le pied de page.....	12
5. Vérifier la validité.....	14



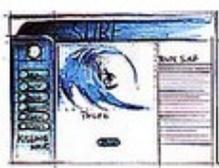
# 1. Plan du site

Réaliser le plan du site web, c'est d'abord organiser les rubriques et pages du site, afin de faciliter la navigation aux internautes. Il faut réaliser une arborescence et structurer toutes les informations du site web. La meilleure organisation consiste à regrouper les informations afin de faciliter l'accessibilité et de simplifier la lecture.

1. Pour commencer le plan et la structure, déterminer la nature du site : un site vitrine, un site marchand, un site de tourisme...
  - Quel est l'objectif visé ?
  - Que souhaitez-vous présenter ?
  - À qui s'adressera votre site ?
  - Quel nombre de pages prévoyez-vous de produire ?
  - Quelle doit être l'apparence des pages ?
  - Est-il prévu d'apporter par la suite des modifications ou de faire des mises à jour ?
2. Puis planifier le contenu en fonction de la globalité des produits et services puis définir la page d'accueil, le menu, la navigation, le pied de page...
3. Puis classer l'ensemble de votre contenu sous la rubrique : Catégories/Sous-catégories/Sujets, il est possible d'avoir une mise en page simplifiée. Celle-ci est représentée comme un arbre généalogique. Il est souhaitable de penser le site en largeur (dans l'idéal, ne pas dépasser **sept menus**) et en profondeur (dans l'idéal, ne pas dépasser **trois niveaux** pour respecter la règle des **trois clics**).



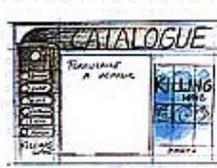
Page "Snow"



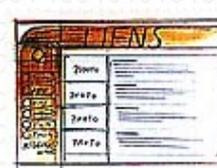
Pages "Surf"



Pages "Windsurf"



Page "Catalogue"



Page "Liens"



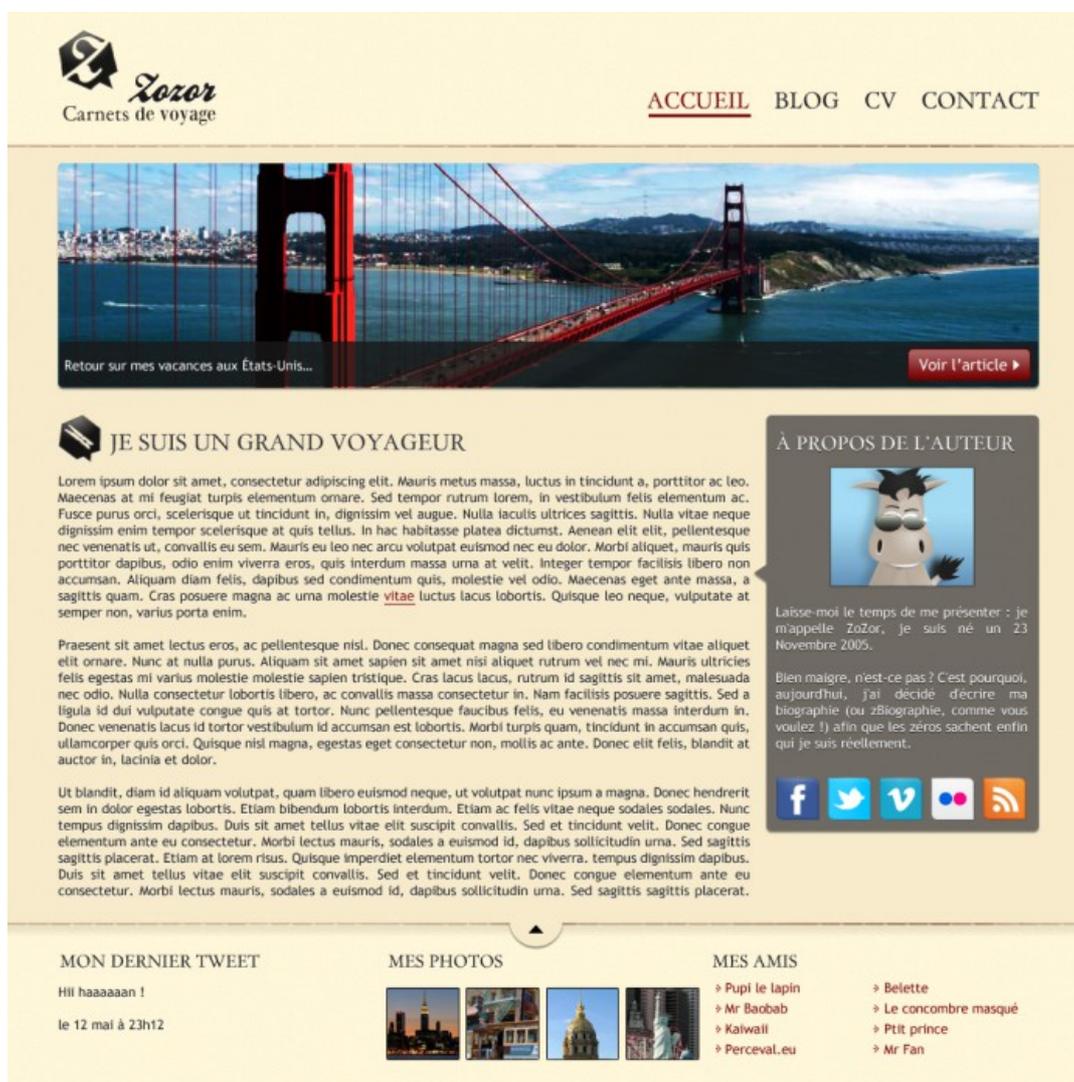
## 2. Maquettage du design

La deuxième étape consiste à maquetter le design, pour avoir un objectif du site web à réaliser. À partir de là, deux possibilités :

- soit vous êtes des graphistes ayant l'habitude d'imaginer des designs, avec des logiciels comme [GIMP](#)<sup>1</sup> ou Photoshop ;
- soit vous n'êtes pas très créatifs, vous manquez d'inspiration et, dans ce cas, vous allez chercher votre inspiration sur des sites web comme [freehtml5templates.com](#), qui vous proposent des idées de design et qui peuvent même vous donner le code HTML / CSS tout prêt !

Cette maquette est en fait une simple image du résultat que l'on veut obtenir. Le graphiste doit fournir les éléments qui vont construire le design, c'est-à-dire les codes couleurs utilisés, les images découpées ainsi que les polices utilisées.

Exemple de maquette



Exemple d'images « découpées » utilisées dans le design

1 GNU Image Manipulation Program



Dans la phase suivant, il faut procéder en deux temps :

1. construire le squelette HTML de la page.
2. mettre en forme et mettre en page avec CSS.

### 3. Organiser le contenu en HTML

La première chose à faire est de distinguer les principaux blocs sur la maquette. Ces blocs vont constituer le squelette de la page.

Pour créer ce squelette, il faut utiliser différentes balises HTML :

- les balises structurantes de HTML5 : <header>, <section>, <nav>, ...
- la balise universelle <div> quand aucune balise structurante ne convient

De préférence, utilisez une balise qui a du sens (comme les balises structurantes <header>, <section>, <nav>) mais, si aucune balise ne vous semble mieux convenir, optez pour la balise générique <div>.

Exemple de Maquette découpée en différentes sections



Toutes les balises que à utiliser n'apparaissent pas sur cette maquette mais cela permet d'avoir une

idée de l'imbrication.

Voici le code HTML correspondant à cette structure :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="style.css" />
    <title>Zozor - Carnets de voyage</title>
  </head>
  <body>
    <div id="bloc_page">
      <header>
        <div id="titre_principal">
          
          <h1>Zozor</h1>
          <h2>Carnets de voyage</h2>
        </div>

        <nav>
          <ul>
            <li><a href="#">Accueil</a></li>
            <li><a href="#">Blog</a></li>
            <li><a href="#">CV</a></li>
            <li><a href="#">Contact</a></li>
          </ul>
        </nav>
      </header>

      <div id="banniere_image">
        <div id="banniere_description">
          Retour sur mes vacances aux États-Unis...
          <a href="#" class="bouton_rouge">
            Voir l'article 
          </a>
        </div>
      </div>

      <section>
        <article>
          <h1>Je suis un grand voyageur</h1>
          <p>Lorem ipsum dolor sit amet...</p>
          <p>Vivamus sed libero nec mauris pulvinar facilisis ut non sem...</p>
          <p>Phasellus ligula massa, congue ac vulputate non, dignissim at augue...</p>
        </article>
        <aside>
          <h1>À propos de l'auteur</h1>
          
          <p id="photo_zozor"></p>
          <p>Laisse-moi le temps de me présenter : je m'appelle Zozor, je suis né un 23 novembre 2005.</p>
        </aside>
      </section>
    </div>
  </body>
</html>
```

<p>Bien maigre, n'est-ce pas ? C'est pourquoi, aujourd'hui, j'ai décidé d'écrire ma biographie (ou zBiographie, comme vous voulez !) afin que les zéros sachent qui je suis réellement.</p>

<p>









</p>

</aside>

</section>

<footer>

<div id="tweet">

<h1>Mon dernier tweet</h1>

<p>Hii haaaaaan !</p>

<p>le 12 mai à 23h12</p>

</div>

<div id="mes\_photos">

<h1>Mes photos</h1>

<p>









</p>

</div>

<div id="mes\_amis">

<h1>Mes amis</h1>

<ul>

<li><a href="#">Pupi le lapin</a></li>

<li><a href="#">Mr Baobab</a></li>

<li><a href="#">Kaiwaii</a></li>

<li><a href="#">Perceval.eu</a></li>

</ul>

<ul>

<li><a href="#">Belette</a></li>

<li><a href="#">Le concombre masqué</a></li>

<li><a href="#">Ptit prince</a></li>

<li><a href="#">Mr Fan</a></li>

</ul>

</div>

</footer>

</div>

</body>

</html>

Petite particularité : tout le contenu de la page est placé dans une grande balise <div> ayant pour id bloc\_page (on l'appelle aussi parfois main\_wrapper en anglais). Cette balise englobe tout le contenu, ce qui va nous permettre de fixer facilement les dimensions de la page et de centrer le site à l'écran.

## 4. Mettre en forme en CSS

Le but est d'obtenir le rendu le plus proche possible, sans chercher la perfection. Pour mettre en forme le design, il faut procéder en plusieurs étapes :

1. Polices personnalisées.
2. Définition des styles principaux de la page (largeur du site, fond, couleur par défaut du texte).
3. En-tête et liens de navigation.
4. Bannière (représentant le pont de San Francisco).
5. Section principale du corps de page, au centre.
6. Pied de page (footer).

### 4.1. Les polices personnalisées

Pour les besoins du design, le graphiste a utilisé trois polices sur sa maquette :

- Trebuchet MS (police courante) ;
- **BallparkWeiner** (police exotique) ;
- **Day Roman** (police exotique).

La plupart des ordinateurs sont équipés de Trebuchet MS. Par contre, les deux autres polices sont un peu originales et ne sont sûrement pas présentes sur les ordinateurs de vos visiteurs et devront être téléchargées.

Voici le code CSS associé :

```
/* Définition des polices personnalisées */
@font-face
{
  font-family: 'BallparkWeiner';
  src: url('polices/ballpark.eot');
  src: url('polices/ballpark.eot?#iefix') format('embedded-opentype'),
       url('polices/ballpark.woff') format('woff'),
       url('polices/ballpark.ttf') format('truetype'),
       url('polices/ballpark.svg#BallparkWeiner') format('svg');
  font-weight: normal;
  font-style: normal;
}
@font-face
{
  font-family: 'Dayrom';
  src: url('polices/dayrom.eot');
  src: url('polices/dayrom.eot?#iefix') format('embedded-opentype'),
       url('polices/dayrom.woff') format('woff'),
       url('polices/dayrom.ttf') format('truetype'),
       url('polices/dayrom.svg#Dayrom') format('svg');
  font-weight: normal;
```

```
font-style: normal;
}
```

Il faut bien entendu mettre à disposition les fichiers des polices (sous-dossier polices dans lequel on été mises les différentes versions des polices).

## 4.2. Définition des styles principaux

Il faut maintenant définir quelques styles globaux pour tout le design de la page :

- image de fond
- une police et une couleur de texte par défaut
- dimensionner la page et la centrer à l'écran

```
/* Eléments principaux de la page */
body
{
  background: url('images/fond_jaune.png');
  font-family: 'Trebuchet MS', Arial, sans-serif;
  color: #181818;
}
#bloc_page
{
  width: 900px;
  margin: auto;
}
section h1, footer h1, nav a
{
  font-family: Dayrom, serif;
  font-weight: normal;
  text-transform: uppercase;
}
```

Avec #bloc\_page, le bloc qui englobe toute la page, les limites ont été fixées à 900 pixels de large. Avec les marges automatiques, le design sera centré.

Pour s'adapter aux dimensions de l'écran du visiteur, définissez une largeur en pourcentage plutôt qu'en pixels.

La propriété CSS text-transform: uppercase; permet de faire en sorte que les titres soient toujours écrits en majuscules. Cette propriété transforme le texte en majuscules (elle peut aussi faire l'inverse avec lowercase).

## 4.3. En-tête et liens de navigation

L'en-tête contient aussi les liens de navigation. Commençons par définir l'en-tête et, en particulier, le logo en haut à gauche.

Code CSS de l'en-tête :

```
/* Header */
```

```
header
{
  background: url('images/separateur.png') repeat-x bottom;
}
#titre_principal
{
  display: inline-block;
}
header h1
{
  font-family: 'BallparkWeiner', serif;
  font-size: 2.5em;
  font-weight: normal;
}
#logo, header h1
{
  display: inline-block;
  margin-bottom: 0px;
}
header h2
{
  font-family: Dayrom, serif;
  font-size: 1.1em;
  margin-top: 0px;
  font-weight: normal;
}
```

La distinction entre l'en-tête et le corps de page est créée grâce à une image de fond. Les éléments sont positionnés en inline-block et nous personnalisons les polices et les dimensions.

## 4.4. Les liens de navigation

```
/* Navigation */
nav
{
  display: inline-block;
  width: 740px;
  text-align: right;
}
nav ul
{
  list-style-type: none;
}
nav li
{
  display: inline-block;
  margin-right: 15px;
}
nav a
```

```

{
  font-size: 1.3em;
  color: #181818;
  padding-bottom: 3px;
  text-decoration: none;
}
nav a:hover
{
  color: #760001;
  border-bottom: 3px solid #760001;
}

```

La définition CSS `list-style-type: none;` permet de retirer l'image ronde servant de puce. Chaque élément de la liste (`<li>`) est positionné en `inline-block`, ce qui permet de placer les liens côte à côte.

La figure suivante représente le résultat obtenu avec les derniers ajouts de CSS.



## 4.5. La bannière

La maquette comporte une jolie bannière représentant le pont de San Francisco. Cette bannière, sur le site, peut être amenée à évoluer. Ici, elle peut servir à illustrer, par exemple, le dernier billet de blog de notre ami Zozor, qui vient de visiter San Francisco.

La bannière est intéressante à plus d'un titre :

- elle comporte des angles arrondis ;
- la description est écrite sur un fond légèrement transparent ;
- le bouton « Voir l'article » est réalisé en CSS, avec des angles arrondis ;
- une ombre vient donner du volume à la bannière.

Voici le code utilisé pour réaliser toute la bannière :

```
/* Bannière */
#banniere_image
{
    margin-top: 15px;
    height: 200px;
    border-radius: 5px;
    background: url('images/sanfrancisco.jpg') no-repeat;
    position: relative;
    box-shadow: 0px 4px 4px #1c1a19;
    margin-bottom: 25px;
}
#banniere_description
{
    position: absolute;
    bottom: 0;
    border-radius: 0px 0px 5px 5px;
    width: 99.5%;
    height: 33px;
    padding-top: 15px;
    padding-left: 4px;
    background-color: rgba(24,24,24,0.8);
    color: white;
    font-size: 0.8em;
}
.bouton_rouge
{
    display: inline-block;
    height: 25px;
    position: absolute;
    right: 5px;
    bottom: 5px;
    background: url('images/fond_degraderouge.png') repeat-x;
    border: 1px solid #760001;
    border-radius: 5px;
    font-size: 1.2em;
    text-align: center;
    padding: 3px 8px 0px 8px;
    color: white;
    text-decoration: none;
}
.bouton_rouge img
{
    border: 0;
}
```

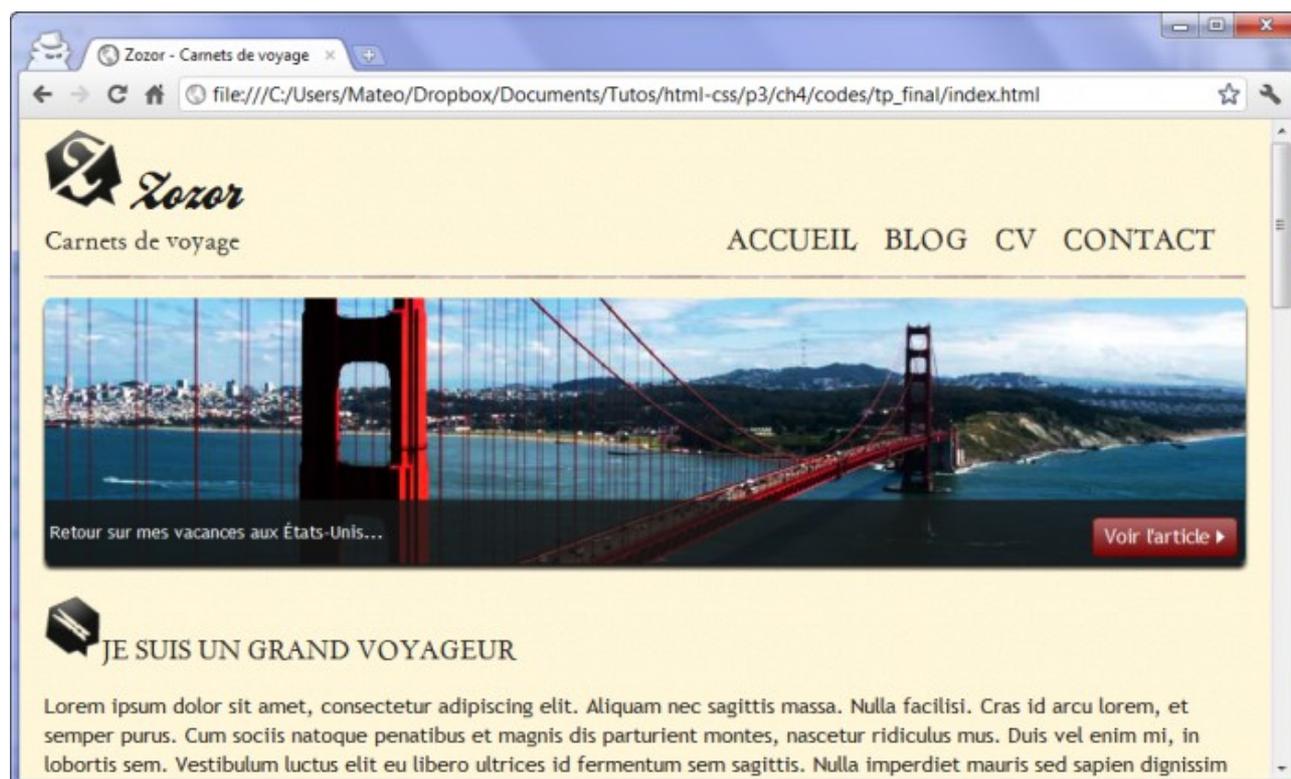
On a choisi d'afficher l'image du pont sous forme d'image de fond dans le bloc <div> de la bannière avec une position relative à la bannière, sans utiliser de propriétés pour en modifier le décalage...

Cela a été particulièrement utile pour placer le bouton « Voir l'article » en bas à droite de la bannière (en absolu à l'intérieur).

La bannière est positionnée en relatif (sans décalage). Comme le bouton est positionné en absolu à l'intérieur, il se place donc en bas à droite de la bannière !

Pour la légende de la bannière, la transparence a été gérée avec la notation RGBA plutôt que la propriété opacity. En effet, opacity aurait rendu tout le contenu du bloc transparent, y compris le bouton « Voir l'article » à l'intérieur.

Résultat intermédiaire



Pour réaliser le dégradé du bouton « Voir l'article », une image de fond représentant le dégradé a été utilisée et répétée horizontalement (la propriété CSS3 linear-gradient permet également de réaliser des dégradés sans avoir à recourir à une image de fond).

## 4.6. Le corps

Le corps, au centre de la page, est constitué d'une unique balise <section>.

Le positionnement du bloc « À propos de l'auteur » se fait en inline-block, on joue avec les angles arrondis et les ombres, on ajuste un peu les marges et les dimensions du texte.

```
/* Corps */
article, aside
{
  display: inline-block;
  vertical-align: top;
  text-align: justify;
}
```

```
article
{
  width: 625px;
  margin-right: 15px;
}
.ico_categorie
{
  vertical-align: middle;
  margin-right: 8px;
}
article p
{
  font-size: 0.8em;
}
aside
{
  position: relative;
  width: 235px;
  background-color: #706b64;
  box-shadow: 0px 2px 5px #1c1a19;
  border-radius: 5px;
  padding: 10px;
  color: white;
  font-size: 0.9em;
}
#fleche_bulle
{
  position: absolute;
  top: 100px;
  left: -12px;
}
#photo_zozor
{
  text-align: center;
}
#photo_zozor img
{
  border: 1px solid #181818;
}
aside img
{
  margin-right: 5px;
}
```

La difficulté ici est de réussir à placer la flèche à gauche du bloc <aside> « À propos de l'auteur » pour donner l'effet d'une bulle. Là encore, notre meilleur ami est le positionnement absolu. La technique est la même : on positionne le bloc <aside> en relatif (sans effectuer de décalage), ce qui me permet ensuite de positionner l'image de la flèche en absolu par rapport au bloc <aside> (et non

par rapport à la page entière). En jouant sur le décalage de l'image, on peut la placer avec précision au pixel près (figure suivante) !

## 4.7. Le pied de page

Celui-ci est constitué de trois sous-blocs, matérialisés par des <div> auxquels on a donné des id pour mieux les repérer. Ces blocs sont positionnés en inline-block les uns à côté des autres.

```
/* Footer */
footer
{
  background: url('images/ico_top.png') no-repeat top center, url('images/separateur.png') repeat-x
top, url('images/ombre.png') repeat-x top;
  padding-top: 25px;
}
footer p, footer ul
{
  font-size: 0.8em;
}
footer h1
{
  font-size: 1.1em;
}
#tweet, #mes_photos, #mes_amis
{
  display: inline-block;
  vertical-align: top;
}
#tweet
{
  width: 28%;
}
#mes_photos
{
  width: 35%;
}
#mes_amis
{
  width: 31%;
}
#mes_photos img
{
  border: 1px solid #181818;
  margin-right: 2px;
}
#mes_amis ul
{
  display: inline-block;
  vertical-align: top;
```

```
margin-top: 0;
width: 48%;
list-style-image: url('images/ico_liensexterne.png');
padding-left: 2px;
}
#mes_amis a
{
text-decoration: none;
color: #760001;
}
```

Deux petites particularités à signaler sur le pied de page :

- on a utilisé la fonctionnalité des images de fond multiples de CSS3, ce qui a permis de réaliser le séparateur entre le corps et le pied de page. Il est constitué de trois images : le séparateur, la petite flèche vers le haut et un léger dégradé.
- la puce de la liste « Mes amis », en bas à droite, a été modifiée avec la propriété list-style-image qui a permis d'utiliser une image personnalisée plutôt que les puces standard.

Le design est terminé !



## 5. Vérifier la validité

Le W3C propose sur son site web un outil appelé le « Validateur » (« Validator » en anglais).

Le validateur est un programme qui va analyser le code source et dire s'il est correctement écrit ou s'il comporte des erreurs qu'il faudra corriger.

Il existe un validateur pour [HTML](#) et un validateur pour [CSS](#).

Si le code HTML est correct, le validateur doit répondre avec le message suivant :

**This document was successfully checked as HTML5!**

Dans ce cas, le site est terminé !