**TP : Créez un site pas à pas**

Enfin, nous y voilà. C'est un chapitre un peu particulier, assez différent de ce que vous avez lu jusqu'à maintenant. En fait, c'est ce que j'appelle un « TP » (*travaux pratiques*). Maintenant, vous ne pouvez plus vous contenter de lire mes chapitres à moitié endormi, vous allez devoir mettre la main à la pâte en même temps que moi.

Vous avez lu beaucoup de théorie jusqu'ici, mais vous vous demandez sûrement comment font les webmasters pour créer d'aussi beaux sites. Vous vous dites que vous êtes encore loin d'avoir les connaissances nécessaires pour construire tout un site… Eh bien vous vous trompez !

**Maquettage du design**

Je vois d'ici le tableau. Vous vous dites « Chouette, on va créer un site complet », vous ouvrez votre éditeur de texte et vous me regardez en me demandant « Bon, par quelle ligne de code on commence ? ».

Et là, je dois justement vous arrêter. Prenez un crayon et un papier : il faut d'abord réfléchir à ce que vous voulez créer comme site. De quoi va-t-il parler ? Avez-vous un thème, un objectif ?

Je sais, par expérience, que la plupart d'entre vous « cherchent juste à apprendre » pour le moment. Vous n'avez donc peut-être pas encore d'idée précise en tête. Dans ce cas, je vous suggère de créer un site pour vous présenter, pour assurer votre présence sur le Web : ce site parlera de vous, il y aura votre CV, vos futures réalisations et pourquoi pas votre blog.

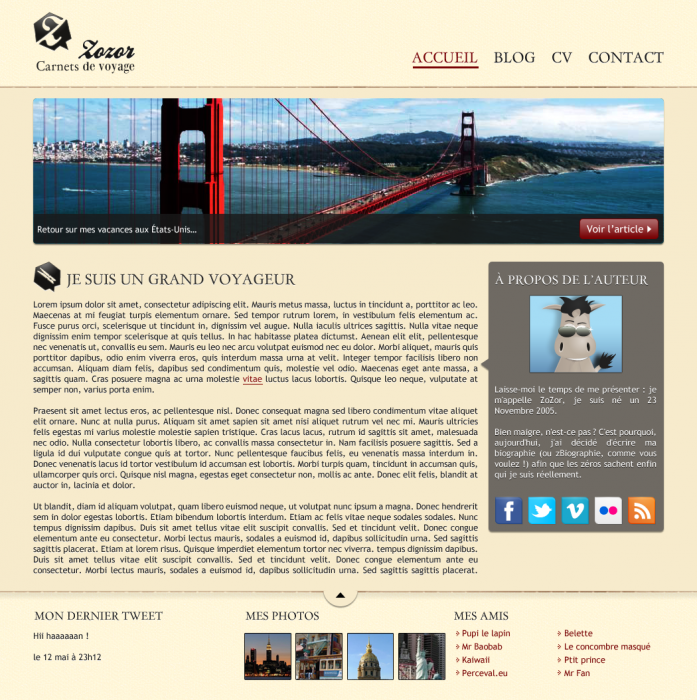
En ce qui me concerne, dans ce TP, je vais réaliser le site web de notre ami Zozor. Zozor a décidé de partir en voyage à travers le monde et sa première étape sera… San Francisco ! Il veut donc créer un site web pour qu'on le connaisse et pour qu'on suive son périple à travers le monde.

Zozor part en voyage et a besoin d'un site web

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

* soit vous êtes un graphiste (ou vous en connaissez un) ayant l'habitude d'imaginer des designs, avec des logiciels comme Photoshop ;
* soit vous n'êtes pas très créatif, vous manquez d'inspiration et, dans ce cas, vous allez chercher votre inspiration sur des sites web comme <https://html5up.net/>, qui vous proposent des idées de design et qui peuvent même vous donner le code HTML/CSS tout prêt !

Pour ma part, j'ai fait appel à un graphiste, Fan Jiyong, qui m'a proposé le design (qui me plaît beaucoup !) que vous pouvez voir à la figure suivante.



La maquette du site web que nous allons réaliser

*Conception de la maquette : Fan Jiyong*

Cette maquette est en fait une simple image du résultat qu'on veut obtenir. Je demande au graphiste de me fournir les éléments qui vont m'aider à construire le design, c'est-à-dire les codes couleurs utilisés, les images découpées (figure suivante), ainsi que les polices dont j'aurai besoin.

[**Télécharger les images et les polices**](https://course.oc-static.com/ftp-tutos/cours/html-css/p3/ch4/tp_images_polices.zip)

Il ne nous reste plus qu'à réaliser ce site web ! Nous allons procéder en deux temps :

1. Nous allons construire le squelette **HTML** de la page.
2. Puis nous allons le mettre en forme et le mettre en page avec **CSS**.

Allez, au boulot !

**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 notre page.

Pour créer ce squelette, nous allons utiliser différentes balises HTML :

* les balises structurantes de HTML5, que nous connaissons : <header>  , <section>  , <nav>  ,   etc. ;
* la balise universelle <div>  quand aucune balise structurante ne convient.

Comment je sais quelle balise utiliser, moi ?

C'est à vous de décider. 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>  .

Regardez la figure suivante pour voir ce que je vous propose comme structure.



Maquette découpée en différentes sections

On peut imaginer d'autres façons de faire le découpage, retenez bien que ma proposition n'est pas forcément la seule et unique solution !

Toutes les balises que l'on va utiliser n'apparaissent pas sur cette maquette, mais cela vous permet d'avoir une idée de l'imbrication que je propose pour les éléments.

Le HTML n'est pas vraiment la partie complexe de la réalisation du site web. En fait, si vous avez bien compris comment imbriquer des balises, vous ne devriez pas avoir de mal à réaliser un code approchant du mien :

<!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">

<div id="logo">

<img src="images/zozor\_logo.png" alt="Logo de Zozor" />

<h1>Zozor</h1>

</div>

<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 <img src="images/flecheblanchedroite.png" alt="" /></a>

</div>

</div>

<section>

<article>

<h1><img src="images/ico\_epingle.png" alt="Catégorie voyage" class="ico\_categorie" />Je suis un grand voyageur</h1>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam nec sagittis massa. Nulla facilisi. Cras id arcu lorem, et semper purus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Duis vel enim mi, in lobortis sem. Vestibulum luctus elit eu libero ultrices id fermentum sem sagittis. Nulla imperdiet mauris sed sapien dignissim id aliquam est aliquam. Maecenas non odio ipsum, a elementum nisi. Mauris non erat eu erat placerat convallis. Mauris in pretium urna. Cras laoreet molestie odio, consequat consequat velit commodo eu. Integer vitae lectus ac nunc posuere pellentesque non at eros. Suspendisse non lectus lorem.</p>

<p>Vivamus sed libero nec mauris pulvinar facilisis ut non sem. Quisque mollis ullamcorper diam vel faucibus. Vestibulum sollicitudin facilisis feugiat. Nulla euismod sodales hendrerit. Donec quis orci arcu. Vivamus fermentum magna a erat ullamcorper dignissim pretium nunc aliquam. Aenean pulvinar condimentum enim a dignissim. Vivamus sit amet lectus at ante adipiscing adipiscing eget vitae felis. In at fringilla est. Cras id velit ut magna rutrum commodo. Etiam ut scelerisque purus. Duis risus elit, venenatis vel rutrum in, imperdiet in quam. Sed vestibulum, libero ut bibendum consectetur, eros ipsum ultrices nisl, in rutrum diam augue non tortor. Fusce nec massa et risus dapibus aliquam vitae nec diam.</p>

<p>Phasellus ligula massa, congue ac vulputate non, dignissim at augue. Sed auctor fringilla quam quis porttitor. Praesent vitae dignissim magna. Pellentesque quis sem purus, vel elementum mi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Maecenas consectetur euismod urna. In hac habitasse platea dictumst. Quisque tincidunt porttitor vestibulum. Ut iaculis, lacus at molestie lacinia, ipsum mi adipiscing ligula, vel mollis sem risus eu lectus. Nunc elit quam, rutrum ut dignissim sit amet, egestas at sem.</p>

</article>

<aside>

<h1>À propos de l'auteur</h1>

<img src="images/bulle.png" alt="" id="fleche\_bulle" />

<p id="photo\_zozor"><img src="images/zozor\_classe.png" alt="Photo de Zozor" /></p>

<p>Laisse-moi le temps de me présenter : je m'appelle Zozor, je suis né un 23 novembre 2005.</p>

<p>Bien maigre, n'est-ce pas ? C'est pourquoi, aujourd'hui, j'ai décidé d'écrire ma biographie afin que vous sachiez qui je suis réellement.</p>

<p><img src="images/facebook.png" alt="Facebook" /><img src="images/twitter.png" alt="Twitter" /><img src="images/vimeo.png" alt="Vimeo" /><img src="images/flickr.png" alt="Flickr" /><img src="images/rss.png" alt="RSS" /></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><img src="images/photo1.jpg" alt="Photographie" /><img src="images/photo2.jpg" alt="Photographie" /><img src="images/photo3.jpg" alt="Photographie" /><img src="images/photo4.jpg" alt="Photographie" /></p>

</div>

<div id="mes\_amis">

<h1>Mes amis</h1>

<div id="listes\_amis">

<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>

</div>

</footer>

</div>

</body>

</html>

Petite particularité : comme vous le voyez, 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 notre site à l'écran.

Pour le reste, aucune grosse difficulté à signaler. Notez que je n'ai pas forcément pensé à toutes les balises du premier coup : en réalisant le design en CSS, il m'est parfois apparu qu'il était nécessaire d'englober une partie des balises d'un bloc <div>  pour m'aider dans la réalisation du design.

Pour le moment, comme vous vous en doutez, le site web n'est pas bien beau (et encore, je suis gentil). Vous pouvez voir le résultat actuel à la figure suivante.

Apparence du site web constitué uniquement du HTML

C'est en CSS que la magie va maintenant opérer.

**Mettre en forme en CSS**

Les choses se compliquent un peu plus lorsqu'on arrive au CSS. En effet, il faut du travail (et parfois un peu d'astuce) pour obtenir un résultat se rapprochant de la maquette. Je dis bien « se rapprochant », car vous ne pourrez jamais obtenir un résultat identique au pixel près.

Mettez-vous bien cela en tête : le but est d'obtenir le rendu *le plus proche possibl*e, sans chercher la perfection. Même si vous obtenez selon vous « la perfection » sur un navigateur, vous pouvez être sûr qu'il y aura des différences sur un autre navigateur (plus ancien) ou sur une autre machine que la vôtre. Nous allons donc faire au mieux et ce sera déjà du travail, vous verrez.

Pour mettre en forme le design, je vais procéder en plusieurs étapes. Je vais m'occuper des éléments suivants, dans cet ordre :

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*).

**Les polices personnalisées**

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

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

Vous trouverez ces polices dans le fichier que je vous ai fait télécharger un peu plus haut. Si ce n'est pas encore fait, je vous encourage fortement à le télécharger.

La plupart des ordinateurs sont équipés de Trebuchet MS  (quoique pas nécessairement tous ; on pourrait la faire télécharger). Par contre, les deux autres polices sont un peu originales et ne sont sûrement pas présentes sur les ordinateurs de vos visiteurs. Nous allons les leur faire télécharger.

Comme vous le savez, il faut proposer plusieurs versions de ces polices pour les différents navigateurs. Dafont  ne propose que le .ttf  en téléchargement. Par contre, FontSquirrel  propose un [générateur de polices](http://www.fontsquirrel.com/fontface/generator) à utiliser en CSS3 avec @font-face  : vous lui envoyez un .ttf  , l'outil transforme le fichier dans tous les autres formats nécessaires, et vous fournit même le code CSS prêt à l'emploi !

Je m'en suis servi pour générer les différentes versions des deux polices exotiques que je vais utiliser. Ensuite, dans mon fichier CSS, je rajoute ce code qui m'a été fourni par FontSquirrel  pour déclarer les nouvelles polices :

/\* 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;

}

En plus de cela, il faut bien entendu mettre à disposition les fichiers des polices. Comme vous le voyez, j'ai créé un sous-dossier polices  dans lequel j'ai mis les différentes versions de mes polices.

**Définition des styles principaux**

On peut maintenant s'attaquer à définir quelques styles globaux pour tout le design de notre page. On va définir une image de fond, une police et une couleur de texte par défaut, et surtout on va dimensionner notre 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, j'ai fixé les limites à 900 pixels de large. Avec les marges automatiques, le design sera centré.

Si vous souhaitez créer un design qui s'adapte aux dimensions de l'écran du visiteur, définissez une largeur en pourcentage plutôt qu'en pixels.

J'ai utilisé la propriété CSS text-transform: uppercase;  (que nous n'avons pas vue auparavant) pour faire en sorte que mes titres soient toujours écrits en majuscules. Cette propriété transforme en effet le texte en majuscules (elle peut aussi faire l'inverse avec lowercase  ). Notez qu'on aurait aussi pu écrire les titres directement en majuscules dans le code HTML.

La figure suivante vous montre ce qu'on obtient pour le moment avec le code CSS. On est encore loin du résultat final, mais on se sent déjà un petit peu plus « chez soi ».

Le fond et les limites de la page commencent à apparaître

**En-tête et liens de navigation**

D'après la structure que j'ai proposée, l'en-tête contient aussi les liens de navigation. Commençons par définir l'en-tête et, en particulier, le logo en haut à gauche. Nous verrons ensuite comment mettre en forme les liens de navigation.

**L'en-tête**

/\* Header \*/

header

{

*background*: url('images/separateur.png') repeat-x *bottom*;

*display*: *flex*;

*justify-content*: space-between;

*align-items*: flex-end;

}

#titre\_principal

{

*display*: *flex*;

*flex-direction*: column;

}

#logo

{

*display*: *flex*;

*flex-direction*: row;

*align-items*: baseline;

}

#logo img

{

*width*: 59px;

*height*: 60px;

}

header h1

{

*font-family*: 'BallparkWeiner', serif;

*font-size*: 2.5em;

*font-weight*: normal;

*margin*: 0 0 0 10px;

}

header h2

{

*font-family*: Dayrom, serif;

*font-size*: 1.1em;

*margin-top*: 0px;

*font-weight*: normal;

}

Nous créons une distinction entre l'en-tête et le corps de page grâce à une image de fond. Les éléments sont positionnés avec Flexbox : il y a plusieurs niveaux d'imbrication, certains sont positionnés verticalement, d'autres horizontalement. Nous personnalisons aussi les polices et les dimensions.

**Les liens de navigation**

La mise en forme des liens de navigation est intéressante. Vous l'avez vu, j'ai créé une liste à puces pour les liens… mais une telle liste s'affiche habituellement en hauteur, et non en largeur. Heureusement, cela se change facilement, vous allez voir :

/\* Navigation \*/

nav ul

{

*list-style-type*: none;

*display*: *flex*;

}

nav li

{

*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 principale nouveauté est la définition CSS list-style-type: none;  , qui permet de retirer l'image ronde servant de puce. Chaque élément de la liste ( <li>  ) est positionné dans une Flexbox, ce qui nous permet de placer les liens côte à côte comme nous le souhaitions.

Le reste des définitions ne contient rien d'extraordinaire : des dimensions, des couleurs, des bordures… Autant de choses que vous connaissez déjà. Notez que je ne trouve pas forcément les bonnes valeurs du premier coup, il me faut parfois tâtonner un peu pour trouver une apparence proche de la maquette d'origine.

La figure suivante représente le résultat que nous obtenons avec les derniers ajouts de CSS.

L'en-tête est mis en page

**La bannière**

Bien, passons maintenant à un exercice un peu plus difficile mais très intéressant : la bannière ! Notre maquette comporte une jolie bannière représentant le pont de San Francisco. Cette bannière, sur votre 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 que j'ai 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

{

*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;

}

Ce code est assez technique et riche en fonctionnalités CSS. C'est peut-être la partie la plus délicate à réaliser dans cette page.

Vous pouvez constater que j'ai choisi d'afficher l'image du pont sous forme d'image de fond dans le bloc <div>  de la bannière.

J'ai aussi donné une position relative à la bannière, sans utiliser de propriétés pour en modifier le décalage… Pourquoi ? *A priori*, une position relative sans décalage ne sert à rien… Et pourtant, cela m'a été particulièrement utile pour placer le bouton « Voir l'article » en bas à droite de la bannière. En effet, j'ai placé le bouton en absolu à l'intérieur.

Le bouton ne devrait-il pas se placer en bas à droite de la page ?

Non, souvenez-vous ce que je vous avais dit : si un bloc est positionné en absolu dans un autre bloc lui-même positionné en absolu, fixe ou relatif, alors il se positionne *à l'intérieur* de ce bloc.  
Notre 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 !

C'est une technique particulièrement utile et puissante dans la réalisation d'un design, souvenez-vous-en !

Dernier détail : pour la légende de la bannière, j'ai choisi d'utiliser la transparence 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. J'ai trouvé préférable de rendre transparente seulement la couleur de fond plutôt que tout le bloc.

Le résultat est plutôt sympathique (figure suivante).

La bannière est mise en forme

Cela en jette, vous ne trouvez pas ?

Pour réaliser le dégradé du bouton « Voir l'article », j'ai utilisé une image de fond représentant le dégradé, et j'ai répété cette image horizontalement. Sachez qu'il existe une propriété CSS3 , linear-gradient  , qui permet de réaliser des dégradés sans avoir à recourir à une image de fond. Son usage étant un peu complexe, j'ai choisi de ne pas l'utiliser dans cet exemple, mais vous pouvez vous documenter à son sujet si vous le souhaitez !

**Le corps**

Le corps, au centre de la page, est dans notre cas constitué d'une unique balise <section>  (mais il pourrait y en avoir plusieurs, bien sûr).

Le positionnement du bloc « À propos de l'auteur » se fait grâce à Flexbox. J'ai choisi de répartir la taille des éléments avec la propriété flex  , ce qui permet à l'article et au bloc sur le côté d'équilibrer leurs largeurs.

On joue avec les angles arrondis et les ombres, on ajuste un peu les marges et les dimensions du texte, et nous y voilà !

/\* Corps \*/

section

{

*display*: *flex*;

*margin-bottom*: 20px;

}

article, aside

{

*text-align*: justify;

}

article

{

*margin-right*: 20px;

*flex*: 3;

}

.ico\_categorie

{

*vertical-align*: middle;

*margin-right*: 8px;

}

article p

{

*font-size*: 0.8em;

}

aside

{

*flex*: 1.2;

*position*: relative;

*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 petite difficulté ici était 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 : je 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, je peux la placer avec précision où je veux, au pixel près (figure suivante) !

Le corps de la page est mis en forme

**Le pied de page**

Il ne nous reste plus que le pied de page à mettre en forme. Celui-ci est constitué de trois sous-blocs que j'ai matérialisés par des <div>  auxquels j'ai donné des id  pour mieux les repérer. Ces blocs sont positionnés, grâce à une Flexbox, les uns à côté des autres.

/\* Footer \*/

footer

{

*display*: *flex*;

*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

{

*width*: 28%;

}

#mes\_photos

{

*width*: 35%;

}

#mes\_amis

{

*width*: 31%;

}

#mes\_photos img

{

*border*: 1px solid #181818;

*margin-right*: 2px;

}

#listes\_amis

{

*display*: *flex*;

*justify-content*: space-between;

*margin-top*: 0;

}

#mes\_amis ul

{

*list-style-image*: url('images/ico\_liensexterne.png');

*padding-left*: 2px;

}

#mes\_amis a

{

*text-decoration*: none;

*color*: #760001;

}

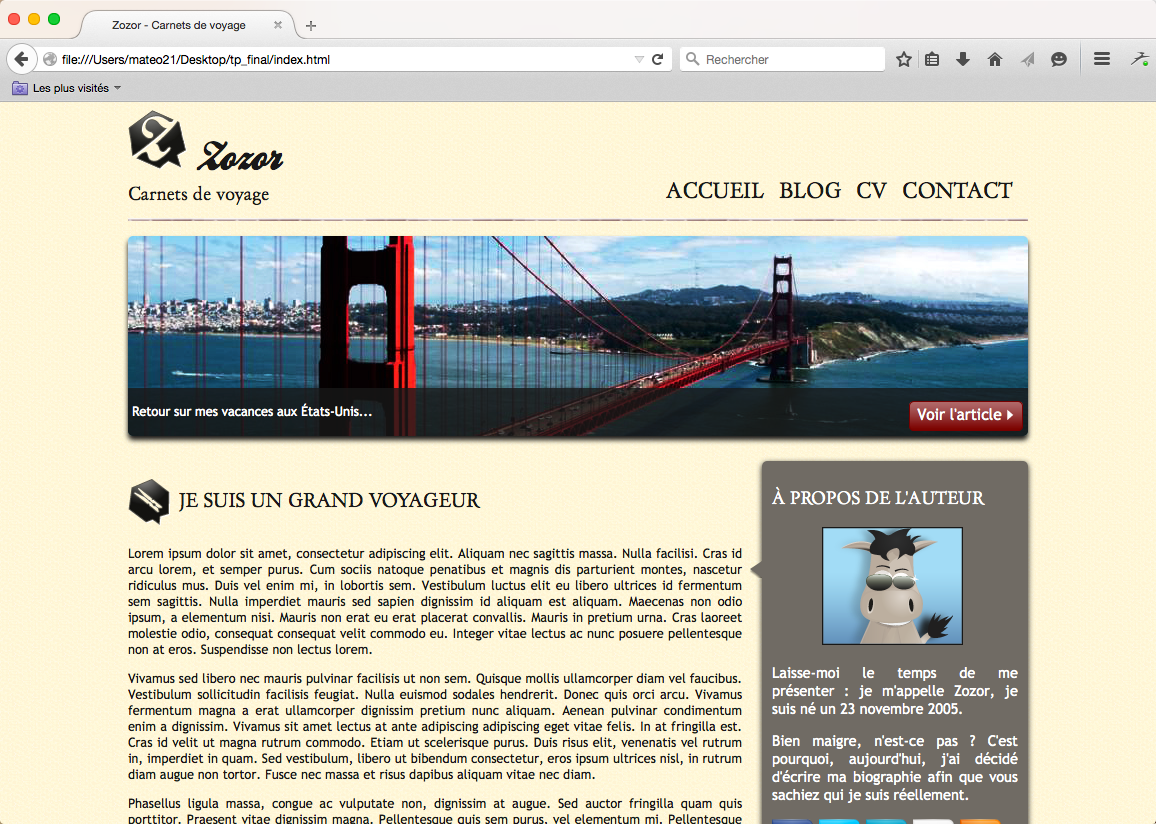
Deux petites particularités à signaler sur le pied de page :

* j'ai utilisé la fonctionnalité des images de fond multiples de CSS3, ce qui m'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 ;
* j'ai modifié la puce de la liste « Mes amis », en bas à droite, avec la propriété list-style-image  qui m'a permis d'utiliser une image personnalisée plutôt que les puces standard. Il existe de nombreuses propriétés CSS spécifiques comme celle-ci et nous ne pouvons pas toutes les voir une par une dans ce cours mais, maintenant que vous êtes un habitué du CSS, vous n'aurez aucun mal à apprendre à les utiliser simplement en lisant l'[annexe listant les principales propriétés CSS](https://openclassrooms.com/courses/apprenez-a-creer-votre-site-web-avec-html5-et-css3/memento-des-proprietes-css).

Et voilà, notre design est terminé (figure suivante) !

Le pied de page est mis en forme

Ah, vous pensez en avoir fini ? Il reste hélas encore un peu de travail : il faut tester notre site sur différents navigateurs. Idéalement, il vaut mieux le faire au fur et à mesure de la mise en place du design. Ouvrez-le donc sur d'autres navigateurs pour vérifier que tout est en ordre !

Ouf ! Le site s'affiche bien aussi sur Firefox !

**Vérifier la validité**

Le W3C propose sur son site web un outil appelé le « validateur » (« Validator » en anglais).  
Le validateur est une sorte de programme qui va analyser votre code source et vous dire s'il est correctement écrit, ou s'il comporte des erreurs que vous devez corriger.

Souvenez-vous : le W3C a établi des normes. Il est nécessaire de les respecter, pour qu'on soit sûr que tous les sites web parlent la même « langue ».

Il existe un validateur pour HTML et un validateur pour CSS (à mettre dans vos favoris !). Celui pour CSS comportant quelques bugs (il signale comme invalides des feuilles CSS qui sont tout à fait valides), nous ne nous y attarderons pas. Par contre, le validateur HTML va être très intéressant pour nous ; voici son adresse : [http://validator.w3.org](https://validator.w3.org/).

Vous pouvez valider votre page web de trois façons différentes, c'est pour cela qu'il y a trois onglets :

* adresse (URL) ;
* envoi du fichier .html  ;
* copier-coller du code HTML.

Pour le moment, notre site web n'est pas encore disponible sur le Web, ce qui fait qu'il n'a pas d'adresse URL. Le mieux est donc d'envoyer le fichier .html  que l'on a fait, ou encore de copier-coller directement le code HTML.

Si vous envoyez votre code HTML et que tout se passe bien, le validateur va vous répondre avec le message représenté à la figure suivante.

Le validateur du W3C nous informe que notre page ne comporte pas d'erreurLe validateur du W3C nous informe que notre page ne comporte pas d'erreur

Dans ce cas, cela signifie que tout va bien et que vous avez bien construit votre page !

Malheureusement, il arrivera souvent que vous ayez des erreurs. Dans ce cas, évitez de paniquer comme cela :

AU SEEECOUUUUUUUURS !!! Ma page web n'est pas valide, je ne vais pas m'en sortir, je suis cerné par les erreurs, faites quelque chose aidez-mmmoiiiiii !

Vous aviez une belle page web, elle s'affichait bien, elle était jolie, et pourtant le validateur vous répond avec un message rouge inquiétant, en vous disant que votre page web n'est pas bien construite.

Tout d'abord, mettez-vous bien ceci en tête : ce n'est pas parce que votre page web s'affiche correctement qu'elle ne comporte pas d'erreur. Votre page web peut être toute belle et comporter beaucoup d'erreurs.

Quel intérêt de les corriger, alors ?

Il faut savoir que les navigateurs « essaient » de ne pas afficher les erreurs, lorsqu'ils en rencontrent, pour ne pas perturber l'internaute. Mais rien ne vous dit que d'autres navigateurs ne vont pas se comporter bizarrement !

Avoir une page web valide, c'est donc avoir la possibilité de dormir tranquille en sachant que l'on a bien fait les choses comme il faut. Cela simplifie le travail des programmes qui lisent les pages web.  
De plus, et c'est vérifié, une page web correctement construite aura plus de chances d'être mieux positionnée dans les résultats de recherche de Google, ce qui vous amènera… plus de visiteurs !

Voici une liste de conseils qui peuvent vous aider à résoudre les erreurs qui risquent de vous être signalées tôt ou tard :

* tous vos textes doivent en général être dans des balises de paragraphes. Il est interdit de mettre du texte directement entre les balises <body></body>  sans l'avoir entouré des fameux <p></p>  . Ceci est aussi valable pour les retours à la ligne <br />  , qui doivent être à l'intérieur de paragraphes. C'est une erreur ultra-courante chez les débutants ;

<p>Ceci est un texte correctement placé dans un paragraphe.

<br />

Les balises <br /> doivent se trouver à l'intérieur d'un paragraphe, ne l'oubliez pas</p>

Ceci est un texte en-dehors d'un paragraphe. C'est interdit.

<br />

* *toutes vos images doivent comporter un attribut*alt  qui indique ce que contient l'image. Si, par hasard, votre image est purement décorative (vous ne pouvez pas lui trouver une description), vous êtes autorisé à ne rien mettre comme valeur pour l'attribut alt  .

<!-- L'image comporte une description -->

<img src="photo.jpg" alt="Une photo de moi" />

<!-- L'image ne comporte pas de description mais a quand même un attribut alt -->

<img src="deco.png" alt="" />

* vos balises doivent être *fermées dans l'ordre*.

<!-- Les balises ne sont pas fermées dans leur ordre d'ouverture -->

<p>Texte <em>important</p></em>

<!-- Les balises sont fermées dans leur ordre d'ouverture -->

<p>Texte <em>important</em></p>

Gardez bien ce schéma en tête, beaucoup de débutants font cette erreur ;

* si vos liens comportent des &  , vous devez les remplacer par le code &amp;  pour éviter toute confusion au navigateur ;

<!-- Exemple d'un mauvais lien en HTML -->

<a href="http://www.site.com/?jour=15&mois=10&an=2000">

<!-- Exemple d'un bon lien en HTML -->

<a href="http://www.site.com/?jour=15&amp;mois=10&amp;an=2000">

* vérifiez enfin que vous n'avez pas utilisé des balises anciennes et désormais obsolètes en HTML5 (comme le vieux <frame>  , la balise <marquee>  …).

Le validateur vous dira « Element *XXX* undefined » (balise inconnue) ou encore « There is no attribute *XXX* » (attribut inconnu).

Tout le monde fait des erreurs, alors ne paniquez pas. Corrigez pas à pas votre page web jusqu'à ce que le validateur vous affiche un beau résultat en vert.

**Le code final**

Je mets à disposition le code final de la page web que nous avons réalisée. Voici à quoi ça ressemble :



Apparence finale du site web

Vous pouvez télécharger un fichier ZIP contenant tous les fichiers du site pour pouvoir le tester chez vous :

[Télécharger les fichiers du site (500 Ko)](https://course.oc-static.com/ftp-tutos/cours/html-css/p3/ch4/tp_final.zip)

Essayez de refaire le site sans la solution sous les yeux, maintenant ! Ça va vous prendre du temps, vous n'y arriverez sûrement pas du premier coup, mais pas de panique : persévérez, ça finira par marcher !