

TP 4 CSS3

- 1- Visualisez le site « exp_site.html » donné par votre enseignante se trouvant sous le répertoire TP4CSS, tout en se servant du répertoire images, contenant toutes les images du site.

- 2- On veut rendre le site ergonomique et de bonne présentation, pour cela on va lui associer les styles nécessaires :
 1. Les styles principaux de la page (largeur du site, fond, police, couleur par défaut du texte).
 2. En-tête et liens de navigation.
 3. Bannière.
 4. Section principale du corps de page, au centre.
 5. Pied de page (*footer*).

Le code CSS3 qu'on va ajouter doit être écrit :

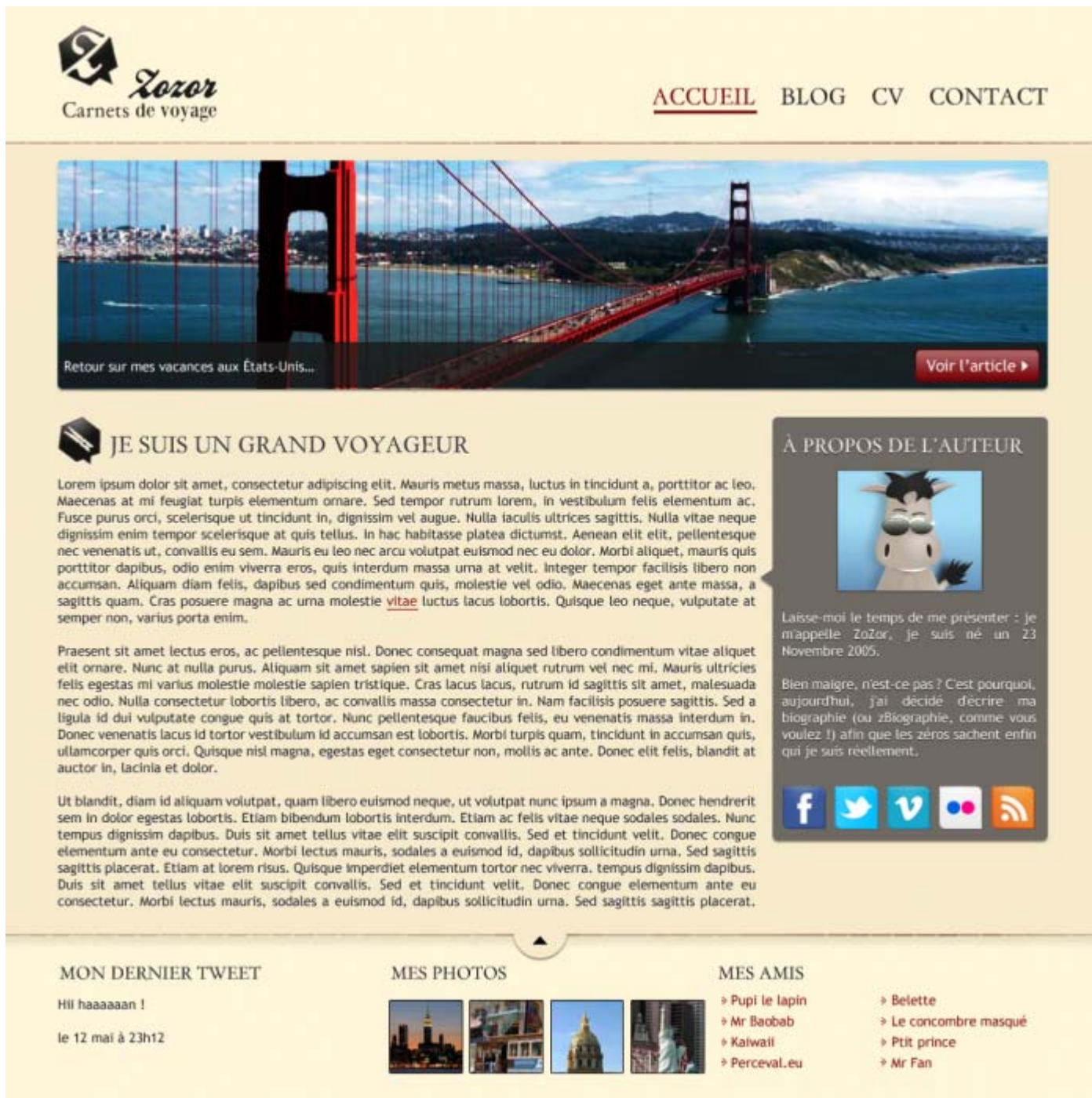
- Soit à l'intérieur du fichier exp_site.html
- Soit dans un fichier externe nommé style_exp_site.css.

- 3- Reprendre votre site « Real Tea » et lui écrire le code CSS3 nécessaire afin d'avoir une bonne présentation.

Annexe TP4CSS3

1-Maquettage du design

La première étape consiste à *maquetter le design*, pour avoir un objectif du site web à réaliser.



La Conception de la maquette

On doit préparer les codes couleurs utilisés, les images découpées ainsi que les polices dont nous aurons besoin.

2-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.

La figure suivante donne la structure de la maquette.



Maquette découpée en différentes sections

Consultez le code HTML5 fournis par votre enseignante.

NB : comme vous le voyez, tout le contenu de la page est placé dans une grande balise `<div>` ayant pour id `bl oc_page`. 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.



Apparence du site web constitué uniquement du HTML5

3-Mettre en forme en CSS

Pour mettre en forme le design, on va procéder en plusieurs étapes. On va s'occuper des éléments suivants, dans cet ordre :

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

5. Pied de page (*footer*).

Définition des styles principaux

Code : CSS

```
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, on a 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** pour faire en sorte que mes titres soient toujours écrits en majuscules. Cette propriété transforme en effet le texte en majuscules.

La figure suivante vous montre ce qu'on obtient pour le moment avec le code CSS.



Le fond et les limites de la page commencent à apparaître

En-tête et liens de navigation

D'après la structure 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

Code : CSS

```
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;
}

```

Les liens de navigation

La mise en forme des liens de navigation est un petit peu plus 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 :

Code : CSS

```

nav
{
    display: inline-block;
    width: 740px;
    text-align: right;
}

nav ul
{
    list-style-type: none;
}

nav li
{
    display: inline-block;
    margin-right: 15px;
}

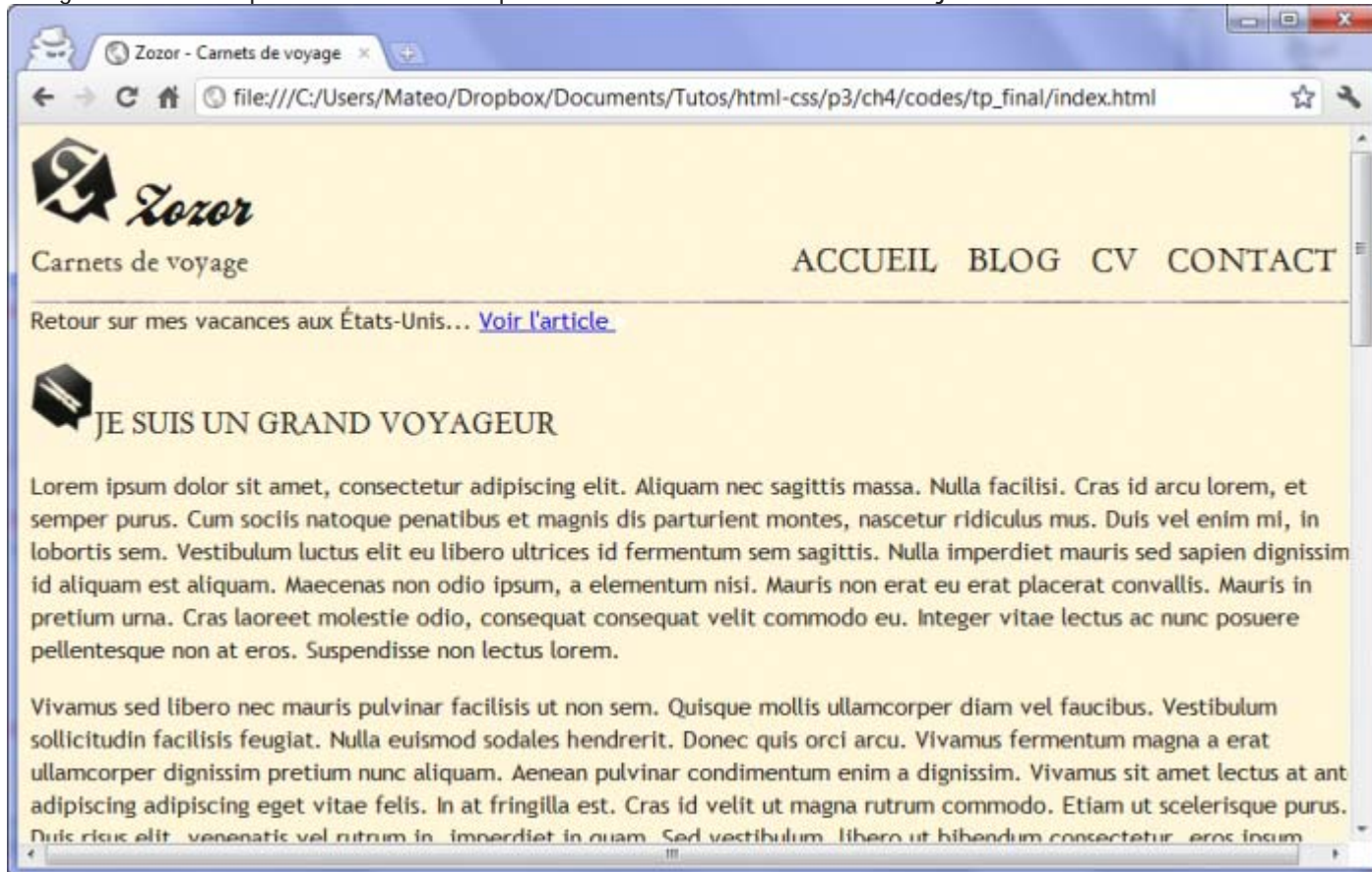
nav a
{
    font-size: 1.3em;
    color: #181818;
    padding-bottom: 3px;
    text-decoration: none;
}

nav a:hover
{
    color: #760001;
    border-bottom: 3px solid #760001;
}

```

La 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 (``) est positionné en inline-block, ce qui nous permet de placer les liens côte à côte comme nous le souhaitions.

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



L'en-tête est mis en page

La bannière

Notre maquette comporte une jolie bannière représentant le pont de San Francisco. Cette 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 pour réaliser toute la bannière :

Code : CSS

```
#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: rgb(24,24,24);

    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;
}

```

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 qu'on a choisi d'afficher l'image du pont sous forme d'image de fond dans le bloc `<div>` de la bannière.

on a 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 nous a été particulièrement utile pour placer le bouton « Voir l'article » en bas à droite de la bannière. En effet, on a placé le bouton en absolu à l'intérieur.

[Le bouton ne devrait-il pas se placer en bas à droite de la page ?](#)

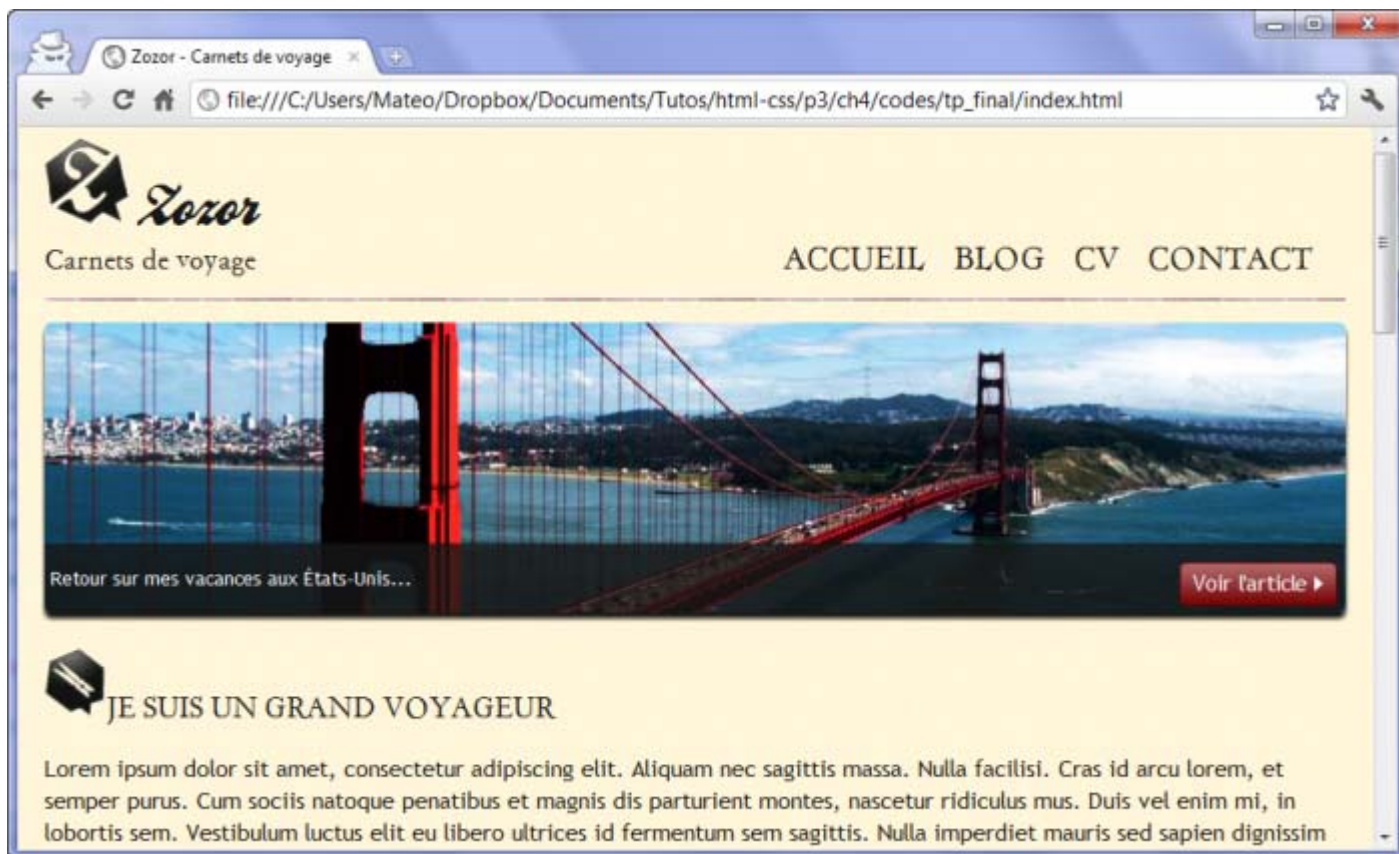
Non, car si un bloc est positionné en absolu dans un autre bloc lui-même positionné en absolu, fixe ou relatif, alors il se positionne *à l'intérieur* de ce bloc.

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.

Dernier détail : pour la légende de la bannière, on a 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 sympathique (figure suivante).



La bannière est mise en forme

Le corps

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

Pas beaucoup de difficultés sur le corps, 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.

Code : CSS

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 petite difficulté ici était de réussir à placer la flèche à gauche du bloc **<asi de>** « 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 **<asi de>** en relatif (sans effectuer de décalage), ce qui me permet ensuite de positionner l'image de la flèche en absolu par rapport au bloc **<asi de>** (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 en inline-block les uns à côté des autres.

Code : CSS

```

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 nous 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.
- On a modifié la puce de la liste « Mes amis », en bas à droite, avec la propriété **list-style-image** qui nous a permis d'utiliser une image personnalisée plutôt que les puces standard.
- Et voilà, notre design est terminé (figure suivante) !



Le pied de page est mis en forme