

Inspiré d'un tutoriel proposé par Gaëtan Weltzer

La page d'accueil du site que nous allons créer est la suivante. Il s'agit d'un site vitrine pour une chaîne de magasin de thés :

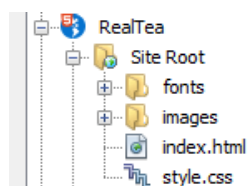


La création du site va se réaliser en 2 étapes :

1 Squelette HTML5 : écrire les balises, mettre en place la sémantique et le référencement naturel

2 Mise en page CSS3 : écrire le CSS et insérer les images du webdesign en même temps, en commençant du haut de la page vers le bas

La structure du site est la suivante :

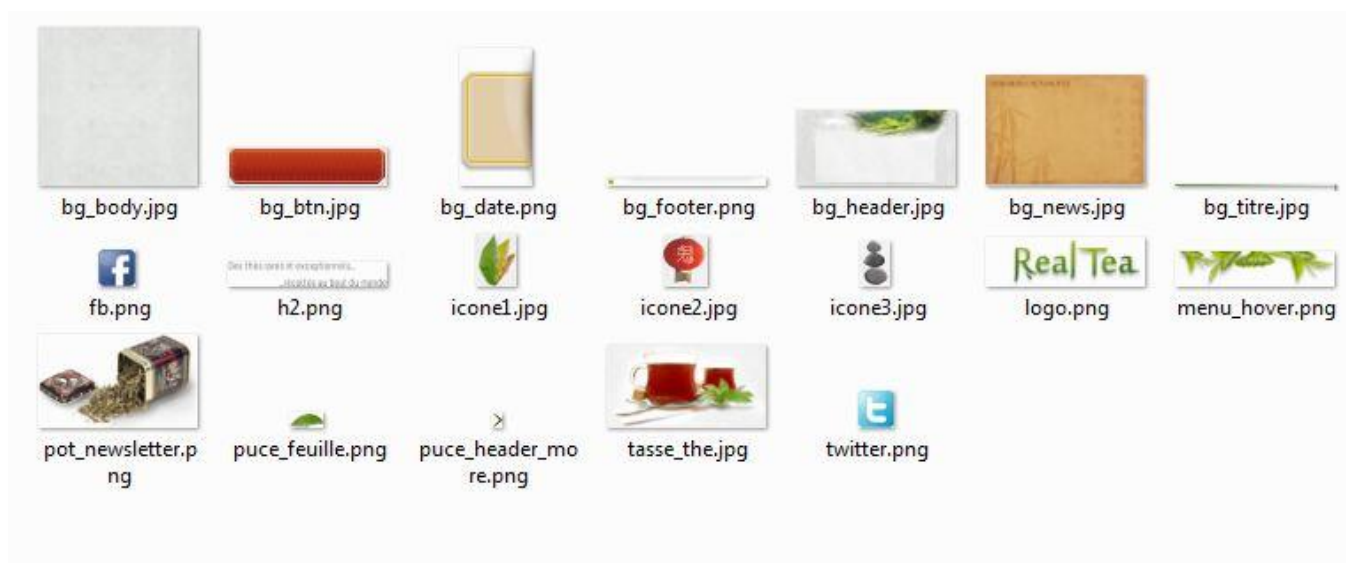


Le contenu des dossier fonts et images sont fournis

Etape 1 : Le découpage

En général je découpe la maquette en même temps que j'écris le CSS pour mettre en page le site mais pour être clair au début de ce tutoriel, voici comment la maquette a été découpée. Ce n'est pas compliqué, dans Photoshop à l'aide de l'outil Tranche, Sélection ou encore Recadrage, j'enregistre toutes mes images **dans un format optimisé**.

Qu'est-ce que j'entends par « optimisé » ? Tout simplement il s'agit de découper les images et images de fond de la meilleure taille possible afin de l'intégrer facilement en CSS. De même, souvent le choix entre le format Jpeg et PNG se pose. Il faut alors réfléchir à la contrainte d'intégration, de design et de poids de l'image. Les images suivantes sont à placer dans un dossier images à la racine du site :



Etape 2 : le squelette HTML5

On attaque ici un gros morceau ! Il faut d'une part comprendre la structure, le **découpage en balises** mais également ingérer **les nouveautés de l'HTML5** en plus de la « **logique sémantique** ».

Comme vous avez dû en entendre parler, l'HTML5 c'est plein de nouvelles choses :

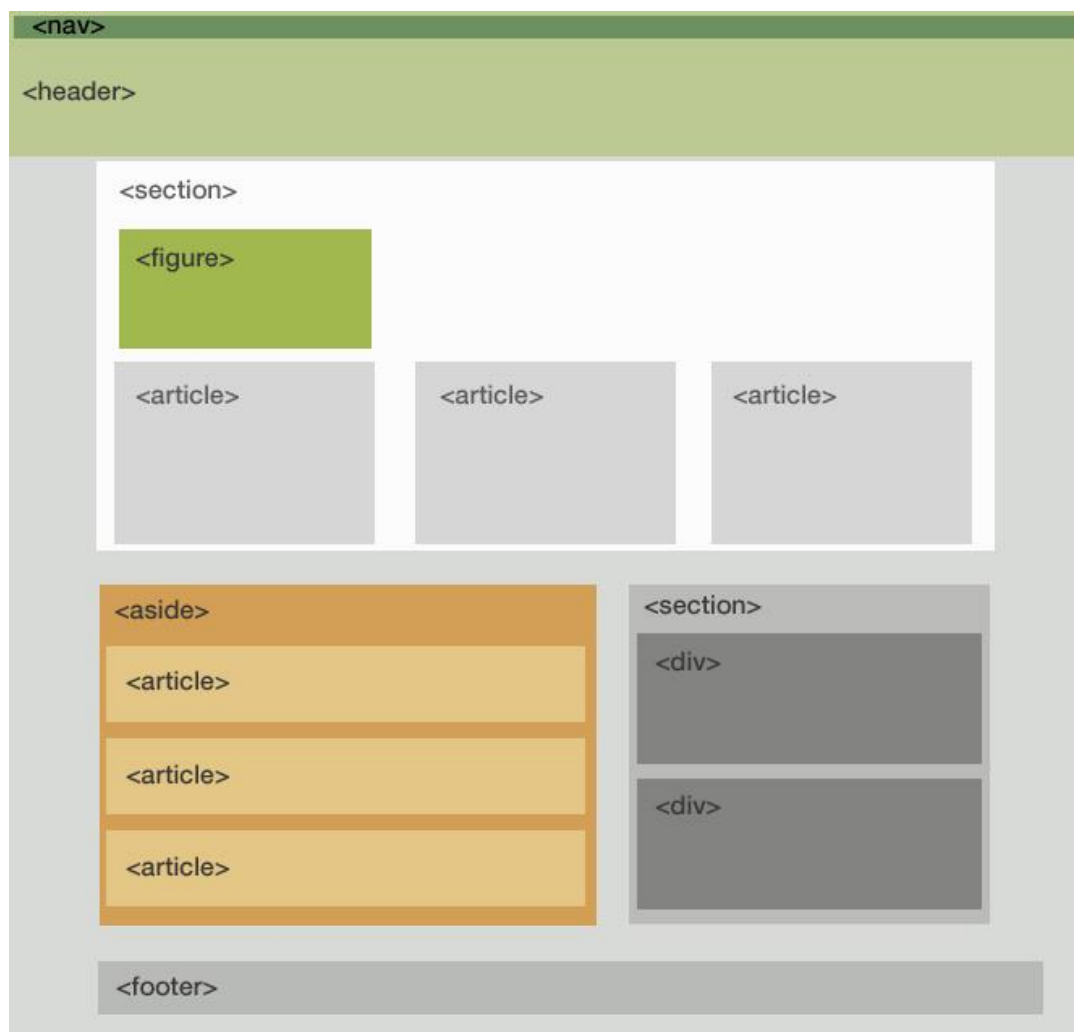
- des nouvelles balises comme <video>,
- des nouvelles propriétés,
- mais aussi une écriture qui respecte mieux la sémantique.

Avant d'attaquer le code, analysons la page

Etant donné que je code cette page en HTML5, j'ai préféré passer par une étape alternative afin de mieux comprendre la structure de notre page HTML. Cela m'a permis de réfléchir au découpage du site et notamment au choix des balises HTML5 les plus adaptées pour chaque bloc d'information.

Car oui, l'information est ce qui va orienter la structure de votre squelette HTML5

Si je reprends ma maquette sous les yeux, voici ce que je peux en tirer en montrant tout de suite les balises les plus importantes pour ma page. Cela permet en un clin d'œil de « prévoir » le code qui doit être écrit.



- La balise `<nav>` contiendra le menu
- La balise `<header>` contiendra l'entête, par exemple le logo, le slogan, la bannière du site.
- La balise `<section>` sert à découper la page en portion
- La balise `<figure>` sert à spécifier un contenu autonome comme des illustrations, des diagrammes, des photos...
- La balise `<article>` contiendra un article, une actualité, une note...
- La balise `<aside>` contiendra un contenu qui n'a qu'un rapport indirect avec le contenu principal du document.
- La balise `<div>` est une balise conteneur neutre
- La balise `<footer>` contiendra le pied de page.

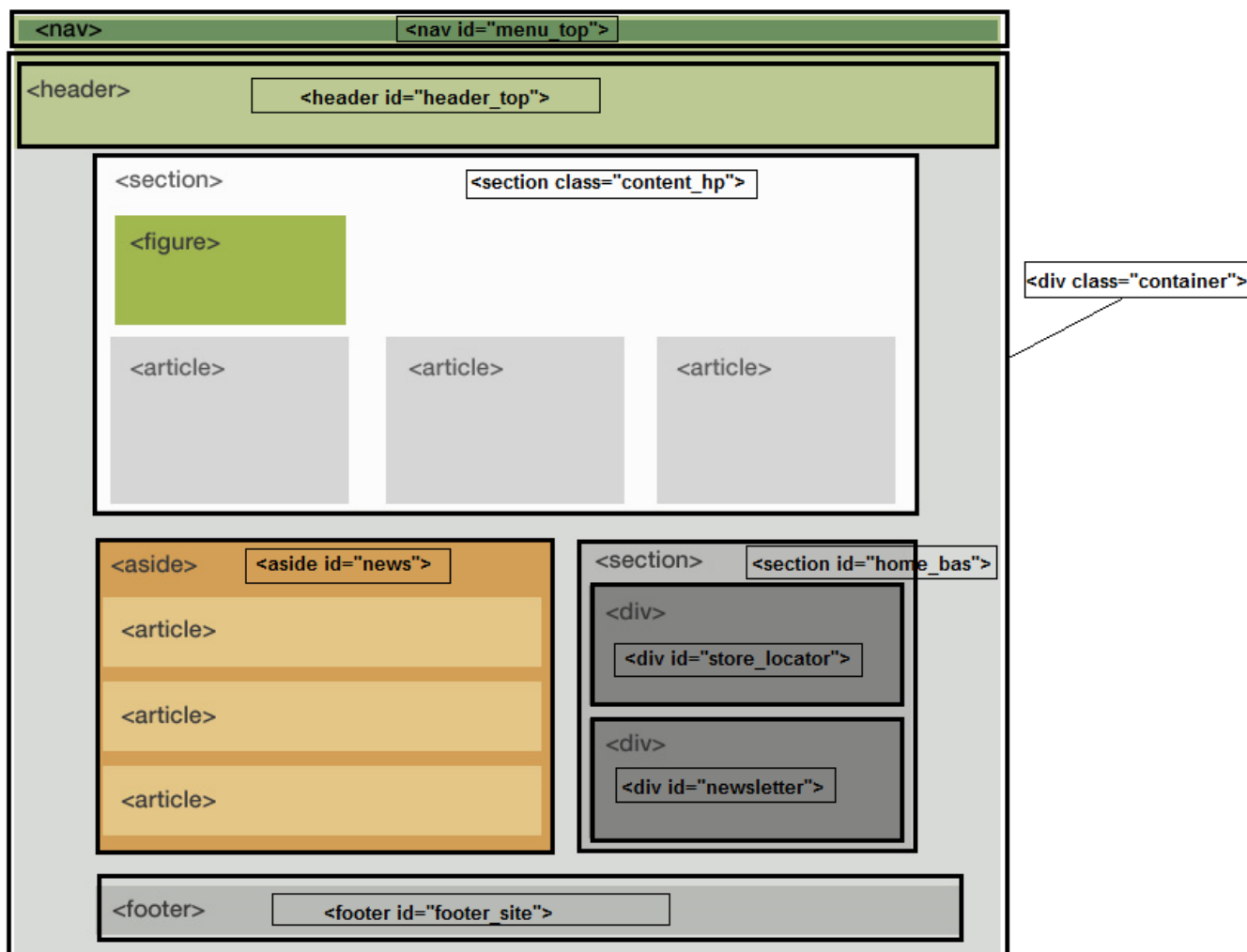
Le début est toujours le même : **le markup**

Voici le code du squelette de notre site qui contient l'appel à la feuille de style CSS :

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8" />
    <title>Real Tea : Tutoriel pour Living Tuts - fr.livingtuts.com</title>
    <link rel="stylesheet" href="style.css" type="text/css" />
  </head>
  <body>

  </body>
</html>
```

Voici la maquette du site où les différentes parties sont nommées :



Le menu de navigation



Le menu de navigation est l'occasion pour nous de présenter **la balise <nav>** dont le nom parle presque de lui-même.

Terminé les <div> pour faire vos menus, maintenant il y a la balise <nav> spécialement conçue pour englober les menus de navigation.

Le code n'est pas plus compliqué que ce qui suit :

```
<body>
  <nav id="menu_top">
    <ul>
      <li><a href="#">Accueil</a></li>
      <li><a href="#">Real tea</a></li>
      <li><a href="#">Nos thés</a></li>
      <li><a href="#">Trouvez-nous</a></li>
      <li><a href="#">Nous contacter</a></li>
    </ul>
  </nav>
</div class="c1"></div>
```

Dans cette balise, on utilise une liste non ordonnée pour lister les liens, tout ce qu'il y a de plus classique.

La ligne **<div class="c1"></div>** est placée ici (et sera placée ailleurs dans le code) pour se faire appliquer le style CSS : **cl {clear:both}** qui signifie que l'élément sera placé "au-dessous" de l'élément précédent. Pour information voici les valeurs possibles de clear :

- none, l'élément sera affiché à côté de l'élément précédent.
- left, l'élément sera affiché au-dessous de l'élément précédent si ce dernier est à float:left.
- right, l'élément sera affiché au-dessous de l'élément précédent si ce dernier est à float:right.
- both, l'élément sera affiché au-dessous de l'élément précédent si ce dernier est à float:left ou float:right (css2).

Le header

On a tout d'abord la partie tout en-haut contenant le logo et le visuel qui est le header de notre site. Quoi de plus logique que d'**utiliser la balise <header>** à la place d'une div ?



*Je n'ai même pas besoin de préciser à quoi sert cette balise, si ? Une précision tout de même alors : **il peut y avoir plusieurs balises <header> dans la même page**. En effet son emploi n'est pas restreint à la page entière mais peut aussi s'appliquer à d'autres balises. Comme on le verra plus tard, d'autres blocs peuvent avoir une entête (exemple : la balise <article>).*

Le code du header :

```
<div class="container">
<header id="header_top">
  <a id="logo" href="#">
    <h1>
    </h1>
  </a>
</header>
```

Le logo est placé dans une balise <a> puisqu'il sert aussi de lien vers la page d'accueil dans un soucis ergonomique.

***Note :** La div « container » ne sert ici qu'à la mise en page comme on le verra plus tard. Sémantiquement elle est neutre.*

La zone forte

Sous le header nous avons la zone forte qui contient la photo de la tasse de thé, le gros titre ainsi que 3 blocs de texte.



Voici le code

```
<section class="content_hp">
  <header>
    <figure></figure>
    <h2 id="titre_hp"></h2>
    <a href="#" id="header_more">Découvrir nos thés</a>
    <div class="cl"></div>
  </header>
  <article>
    <h2 class="titre1"><span>75 variétés de thé</span></h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor
in reprehenderit in voluptate velit.</p>
  </article>
  <article>
    <h2 class="titre2"><span>Des thés de Chine</span></h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
  </article>
  <article>
    <h2 class="titre3"><span>Des conseils détente</span></h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor
in reprehenderit in..</p>
  </article>
</section> <!-- end .content -->
<div class="cl"></div>
```

Il y a plusieurs choses à relever ici.

- La première est que toute cette partie est contenue dans **une balise <section>** qui comme son nom l'indique représente une section de la page distincte du reste. Elle sert simplement à découper les portions de la page (lorsqu'aucun autre type ne convient mieux comme <header>, <article>, etc...).
- **Dans cette <section>, nous avons un <header>** (et oui comme je l'expliquais au-dessus, il peut y avoir plusieurs headers dans une même page). Pourquoi cette balise <header> ? Tout simplement car si on isole la <section> du reste de la page (comme si elle était la page entière), nous avons au début une image plus un gros titre qui sert ici parfaitement de header (=entête) au bloc.
- L'emploi de la balise **<figure>** : elle sert à englober un contenu qui n'est pas indispensable à la page et est souvent associé à une légende (pas dans notre cas)
- Et juste en-dessous, plutôt que de faire 3 div pour les 3 colonnes de contenu, la **balise <article>** a toute son utilité.
- Le texte en latin est souvent utilisé pour mettre du texte "au kilomètre" pour ne pas laisser les rubriques vides.

*Quelques mots sur la balise <article> : En des termes simples, la balise <article> contient toute forme de texte pouvant être assimilée à un article et pouvant être lu hors du contexte de la page sans perdre son intégrité. Par exemple : article de blog, commentaire d'un blog, texte explicatif... Une **astuce** pour utiliser cette balise : si votre texte contient au moins un titre et un (ou plusieurs) paragraphe et qu'il pourrait très bien être lu dans un flux RSS ou encore partagé sur un réseau social sans perdre son sens et son intégrité, alors il s'agit d'un <article>.*

Autre information concernant la balise <article> : je le répète afin que cela soit clair, même une balise <article> peut contenir plusieurs types de balises pour structurer son contenu : <h1>, <h2>, <h3>, <p>, <etc>... mais aussi un <header>, <footer>, <figure>, <div>, etc...

***Note :** Si on voulait pousser le vice plus loin, il aurait été tout à fait juste pour chacun des trois <article> de rajouter une balise <header> pour chaque titre <h2>. Finalement, ce n'est pas si dur ! Allez, je sens que ça commence à venir, accrochez-vous !*

Le bloc d'actualités



La balise <aside>

Le bloc d'actualités aurait très bien pu se trouver dans une balise <section> tout comme le haut du site mais j'ai choisi d'utiliser à la place la balise <aside>. Comme vous l'avez compris maintenant, il est facile d'utiliser une balise à la place d'une autre en HTML5 et ça peut vite devenir un casse-tête.

Conseil : Dans ce cas, rappelons-nous notre premier conseil : le choix de la balise dépend du contenu du bloc et de sa fonction au sein de son contenant.

Pour le bloc d'actualité, le contenu se réfère aux dernières actualités de la société Real Tea. Le contenant de ce bloc n'est autre que la page entière. La page entière, qui est la page d'accueil, se veut avant tout présenter Real Tea et poursuivre sa stratégie de séduction des clients. Notre bloc d'actualité ne présente pas réellement Real Tea et ses informations **sont séparées du reste de la page**. C'est pourquoi la balise <aside> (= « à côté ») est parfaitement adaptée.

Originellement conçue pour les colonnes latérales, elle était même nommée <sidebar> à ses débuts. Tout comme pour une sidebar, notre bloc d'actualités a un lien avec la page (Informations concernant la société) mais son contenu n'est pas directement lié au reste de la page qui lui présente clairement, l'entreprise, ses produits, ses bénéfices, etc... On peut donc clairement ôter notre bloc <aside> sans empêcher le bon fonctionnement et la bonne compréhension de la page entière.

« The aside element represents a section of a page that consists of content that is tangentially related to the content around the aside element, and which could be considered separate from that content. Such sections are often represented as sidebars in printed typography. »

J'ai souhaité bien insister sur la différence entre <aside> et <section> car c'est un point assez délicat. N'oubliez pas que c'est le contenu et son contexte qui guident l'HTML5.

Les actualités

Maintenant qu'on a détaillé le contenant `<aside>`, passons au contenu ! Les actualités sont ensuite découpées comme il suit :

- le titre « DERNIERES ACTUALITES » dans une balise `<header>`
- une liste non ordonnée `` pour lister les 3 dernières news
- dans chaque ``, il y a un `<article>` pour chaque actualité (ici on peut facilement assimiler une actualité à un article de blog avec son titre, sa date et son texte, même si ce n'est peut-être pas l'article entier)
- au niveau encore plus profond, chaque `<article>` possède un `<header>` et un `<footer>`

Voici le code

```
<aside id="news">
  <header>
    <h2>ACTUALITES</h2>
  </header>
  <ul>
    <li>
      <article>
        <header>
          <h3><a href="#">Ouverture du nouveau Real Tea à
Strasbourg</a></h3>
        </header>
        <footer class="post-info">
          <time datetime="2022-01-01">01 <span><br/>jan</span></time>
        </footer>
        <div class="contenu_article">
          <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Quisque venenatis nunc vitae libero iaculis elementum. Nullam et justo dapibus blandit nec
et leo.</p>
          <p class="lire_suite"><a href="#">Lire la suite</a></p>
        </div><!-- /.entry-content -->
      </article>
    </li>
    <li>
      <article>
        <header>
          <h3><a href="#">Ouverture du nouveau Real Tea à
Strasbourg</a></h3>
        </header>
        <footer class="post-info">
          <time datetime="2022-01-01">01 <span><br/>jan</span></time>
        </footer>
        <div class="contenu_article">
          <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Quisque venenatis nunc vitae libero iaculis elementum. Nullam et justo dapibus blandit nec
et leo.</p>
          <p class="lire_suite"><a href="#">Lire la suite</a></p>
        </div><!-- /.entry-content -->
      </article>
    </li>
  </ul>
</aside>
```

```

        <article>
            <header>
                <h3><a href="#">Ouverture du nouveau Real Tea à
Strasbourg</a></h3>
            </header>
            <footer class="post-info">
                <time datetime="2022-01-01">01 <span><br/>jan</span></time>
            </footer>
            <div class="contenu_article">
                <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Quisque venenatis nunc vitae libero iaculis elementum. Nullam et justo dapibus blandit nec
et leo.</p>
                <p class="lire_suite"><a href="#">Lire la suite</a></p>
            </div><!-- /.entry-content -->
        </article>
    </li>
</ul>
</aside>

```

Il y a dans ce code encore une nouvelle balise : **<time>**

La balise <time> définit aussi bien une heure qu'une date. Dès que vous avez une date ou une heure à écrire, rappelez-vous de cette balise comme un automatisme.

L'attribut datetime va de pair avec <time>. Lorsque vous l'utilisez, vous pouvez écrire n'importe quoi entre les balises <time> afin d'avoir d'autres repères temporels. Exemple :

```
<time datetime="21:00">à 9h pour ma soirée</time>
```

La partie de droite : « Trouvez nos espaces thés » + Newsletter



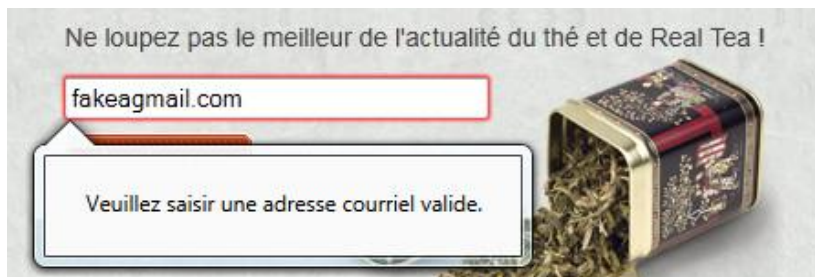
Tout ce bloc de droite est placé dans une balise `<section>`. Chacun des deux blocs à l'intérieur est ensuite placé dans une `<div>` (et oui, cette balise existe encore !).

Pour le bloc « Trouvez nos espaces thés », le code est tout ce qu'il y a de plus simple et classique. L'espace « Newsletter » est un peu plus intéressant car il nous permet d'introduire les nouveaux formulaires HTML5 (même si ce n'est pas énorme !). Voici le code de cette section du site :

```
<section id="home_bas">
  <div id="store_locator">
    <h2>Trouvez nos espaces thés</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua.</p>
    <ul>
      <li><a href="#">Strasbourg</a></li>
      <li><a href="#">Paris</a></li>
      <li><a href="#">Lyon</a></li>
      <li><a href="#">Rennes</a></li>
      <li><a href="#">Marseille</a></li>
      <li><a href="#">Bordeaux</a></li>
    </ul>
  <div class="cl"></div>
</div>
<div id="newsletter">
  <h2>Newsletter</h2>
  <p>Ne loupez pas le meilleur de l'actualité du thé et de Real Tea !</p>
  <form>
    <input name="email" type="email" placeholder="Mon adresse email"
required> <!-- Remplir ce champ est requis -->
    <br/>
    <input type="submit" value="Je m'inscris">
  </form>
</div>
</section>
```

Je vais expliquer quelques nouveautés des formulaires `<form>` en HTML5.

- En effet, toute une flopée de **nouveaux types d'input** ont fait leur arrivée, au total de 13. Il y a parmi eux `type="email"` (pour les adresses email), `type="url"` (pour les adresses URL, oui on s'en doutait...), `type="number"`, `type="range"`, etc...
- Ce qui est génial, c'est qu'ils n'ont pas fait de nouveaux types d'input juste pour en avoir de nouveaux. **Il y a une utilité et des spécificités pour ces types**, par exemple avec le champ email, si vous tapez un texte qui n'a pas la structure d'un email (sans @ par exemple), le navigateur va vous le signaler. C'est d'ailleurs le cas de notre Real Tea (que vous pouvez tester vous-même directement sur la page de démo) :

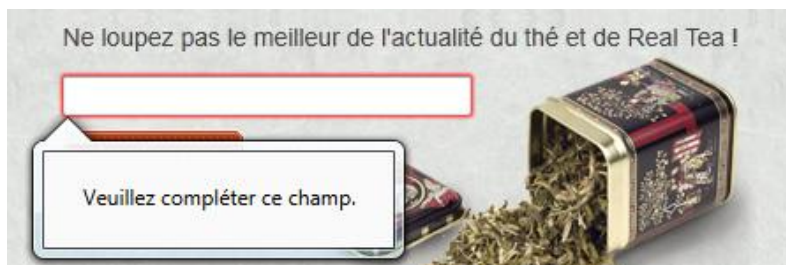


A noter que le navigateur stylise lui-même le champ email : l'ombre portée rouge tout comme la petite tooltip n'ont pas été ajoutée en CSS. C'est le navigateur qui les transforme. C'est super je sais... 😊

- Si vous avez bien analysé notre code HTML5 vous avez remarqué l'attribut « **placeholder** » qui permet d'afficher un texte dans le champ et de le faire disparaître lorsque l'utilisateur clique dans celui-ci. Plus besoin de Javascript !



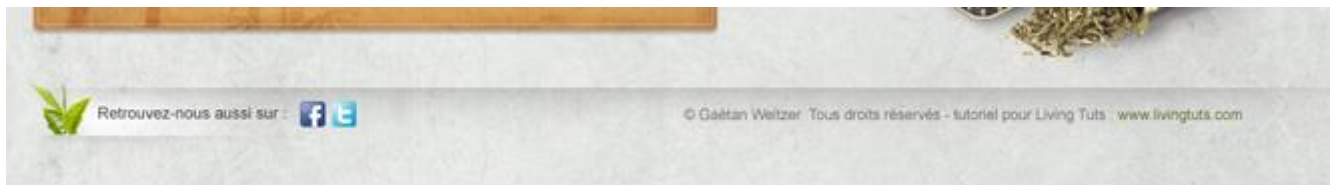
- Il y a aussi l'attribut « **required** » que j'ai ajouté. Fort utile, il permet au navigateur de comprendre que remplir ce champ est nécessaire avant de pouvoir envoyer le formulaire. Si je ne remplis pas le champ et que je clique sur le bouton submit, mon navigateur (moderne) va automatiquement m'avertir que je n'ai pas rempli un champ requis :



NB : *Sous les anciens navigateurs, les types **email** et **url** entre autres sont interprétés comme des input de type **text**, il n'y a donc pas d'inquiétude à avoir ! Par contre certains autres input ne se dégradent pas si bien et sont actuellement inutilisables comme l'input **color** ou encore l'input **range**.*

Je ne sais pas pour vous mais ces petites nouveautés m'ont fait aimer encore plus l'HTML5 !

Le footer



Il n'y a pas grand-chose à dire sur le footer, il est bien entendu inclus dans une balise `<footer>` étant donné que c'est bien un pied-de-page. Les liens vers les réseaux sociaux sont quant à eux placés dans une balise `<aside>` et le reste n'est que du texte. Rien de bien compliqué en somme !

```
<div class="cl margin20"></div>
<footer id="footer_site">
  <aside>
    <p>Retrouvez-nous aussi sur : </p><p style="float:left;"><a href="#"></a> <a href="#"></a></p>
  </aside>
  <p id="copyright">© Gaétan Weltzer Tous droits réservés - tutoriel pour Living Tuts :
<a href="http://www.livingtuts.com">www.livingtuts.com</a></p>
  <div class="cl"></div>
</footer>
```


Etape 3 : La mise en page avec CSS3

Voici l'étape la plus longue mais aussi la plus plaisante pour tout webdesigner car c'est là que votre design prend littéralement vie en habillant votre squelette HTML.

Je vais découper le CSS pour chaque partie sans expliquer tout le code en détail. Je pars du principe que vous avez déjà des connaissances basiques en CSS. Je n'expliquerai que les nouveautés CSS3 ainsi que 2-3 points délicats. Pour le reste, j'estime que le code ainsi que les commentaires sont suffisants. Bien entendu, toutes les propriétés et possibilités offertes par le CSS3 ne seront pas abordées dans le cas de ce tutoriel, mais j'ai cherché à les utiliser au maximum.

L'ajout des typographies avec @font-face

Si vous avez réalisé la première étape du webdesign ou simplement si vous avez l'œil affuté, vous aurez remarqué qu'il y a 3 polices différentes dans cette page : Arial pour les paragraphes, Melbourne pour les titres et le menu de navigation ainsi que Ginko pour le bloc d'actualités.

Melbourne et Ginko n'étant pas natives sur les systèmes d'exploitation d'aujourd'hui, je les ai incorporées dans ma page via la propriété CSS @font-face comme il suit :

```
@font-face { font-family: Melbourne;
             src: url('fonts/Melbourne_regular_basic.otf');
}
@font-face { font-family: Ginko;
             src: url('fonts/GINKO.TTF');
}
```

Qu'est-ce que ça signifie ? bêtement que nous allons créer une nouvelle « font-family » qui pourra être appelée par la suite dans notre feuille de style. La deuxième propriété est « src » qui spécifie où se trouve le fichier de votre typographie. Cela nécessite donc d'uploader la typographie sur un serveur afin de pouvoir l'utiliser sur votre site. Grâce à ces quelques lignes, vous avez ajouté les deux polices prêtes à l'emploi.

Un léger reset CSS

```
body {
    font-family: Arial, Helvetica, Verdana, sans-serif;
    font-size:14px;
    margin: 0;
    padding: 0;
    color: #000;
}

ul, ol, dl, h1, h2, h3, h4, h5 {
    padding: 0;
    margin: 0;
    font-weight:normal;
}

ul, ol {list-style-type:none;}

a img {    border: none;}

/*Prise en charge de HTML 5 - Définit les nouvelles balises HTML5 sur display:block afin que
les navigateurs sachent comment effectuer un rendu correct des balises. */
header, section, footer, aside, nav, article, figure {
    display: block;
    margin:0;
}
```

J'ai simplement repris et épuré le reset CSS que Dreamweaver m'avait généré. Il permet d'éviter certaines différences d'affichages entre les navigateurs et également de définir quelques styles de base, notamment pour body.

La ligne intéressante dans notre cas est la **prise en charge des nouvelles balises HTML5**. Sachant qu'elles sont récentes, certaines version de navigateurs ne connaissent pas les propriétés associées, notamment qu'il s'agit bien d'éléments de type « block » et non pas « inline ».

Classes réutilisables et style des textes

```
.cl {clear:both}
.margin20 { margin-bottom: 20px;}

/* La définition du style des liens de votre site doit respecter cet ordre, y compris le
groupe de sélecteurs qui créent l'effet de survol. */
a:link {color: #42413C;}
a:visited {color: #2b2a25;}
a:hover, a:active, a:focus { /* ce groupe de sélecteurs offrira à un navigateur au clavier
la même expérience de survol que celle d'une personne employant la souris. */
    text-decoration: none;
}

h2 {font-size:28px;
    font-family:Melbourne; /*j'utilise ma custom font Melbourne*/
    padding-bottom:6px;
    background:url(images/bg_titre.jpg) no-repeat bottom left;
}
```

J'ai pour habitude de créer plusieurs classes au début de ma feuille de style que je réutilise tout le long du site comme par exemple la classe « cl » (clear:both) ou encore des classes pour rendre des textes gras, ou des float, etc... Cela peut être très utile et peut alléger considérablement votre code dans le cas de gros sites. Ici nous n'avons qu'une seule page, c'est donc relativement léger.

Ensuite je stylise mes liens <a>, bien que souvent j'applique ensuite à chacun un style particulier.

Enfin, dernier point, je définis un style pour tous mes titres <h2>. Il est intéressant ici de voir **comment je change la police par Melbourne**, celle que nous avons ajouté au début de ce document. Il suffit d'assigner « font-family:Melbourne » qui reprend le nom donné plus haut.

Structure générale de la page

```
body { background: url(images/bg_header.jpg) center top no-repeat, url(images/bg_body.jpg)
repeat;} /*multiple backgrounds : 2 fonds à body*/

#header_top {height:176px; padding-top:29px;}

.container {
    width: 1000px;
    margin: 0 auto; /* la valeur automatique sur les côtés, associée à la largeur, permet
de centrer la mise en page */
}

.content ul, .content ol {
    padding: 0 15px 15px 40px;
}
```

On peut noter ici l'emploi d'une propriété **CSS3** extrêmement pratique et même jouissive ! Il s'agit du **multiple backgrounds**. Si vous regardez bien j'ai ajouté à ma balise body deux images de fond à partir d'une seule propriété background. Il suffit de les séparer par une virgule et le tour est joué !

Vous comprendrez aussi en lisant cela à quoi sert notre div .container qui avait été créée précédemment pour un seul soucis de mise en page. Elle sert à **centrer tout le site en lui assignant la largeur de 1000px**.

Le menu de navigation

```
#menu_top {
    height:26px;
    background-color:rgb(107,158,46); /*alternative pour les vieux navigateurs (sans
transparence ici)*/
    background-color:rgba(107,158,46,0.6); } /*le fond uni avec transparence comme on
aime*/

#menu_top ul {
    width:565px;
    padding-left:435px;
    margin: 0 auto;
    list-style: none;
}

#menu_top ul li {float:left;}

#menu_top ul a, #menu_top ul a:visited {
    padding: 2px 17px 0px;
    display: block;
    font-family:Melbourne;
    font-size:18px;
    color:#fff;
    text-shadow:0 1px 2px #555;
    text-decoration: none;
}

#menu_top ul a.aktif, #menu_top ul a:hover, #menu_top ul a:active, #menu_top ul a:focus {
    background:url(images/menu_hover.png) center top no-repeat;
}
```

J'utilise pour le menu du haut 2 propriétés introduites par le CSS3 :

- **background-color en rgba** qui permet de gérer la couche alpha, donc la transparence du fond
- **text-shadow** : il s'agit de l'ombre portée qu'on peut ajouter à du texte. Les valeurs sont dans l'ordre : *text-shadow : coordonnée x coordonnée y flou couleur;*

Pour le reste le code est classique pour un menu horizontal. Il est toujours bon de le garder sous la main !

La partie haute du site : la zone forte

```
.content_hp {margin-bottom:20px; color:#595959;}
.content_hp h2 { padding:0 0 6px 30px; background:none;}
.content_hp h2.titre1 { background:url(images/icone1.jpg) left top no-repeat,
url(images/bg_titre.jpg) no-repeat bottom left;}
.content_hp h2.titre2 { background:url(images/icone2.jpg) left top no-repeat,
url(images/bg_titre.jpg) no-repeat bottom left;}
.content_hp h2.titre3 { background:url(images/icone3.jpg) left top no-repeat,
url(images/bg_titre.jpg) no-repeat bottom left;}
.content_hp header {padding:10px 0; height:160px;}
.content_hp figure { float:left; margin-left: 20px; margin-top:-8px; }

.content_hp article {
    width:290px;
    margin:0 21px;
    float:left;
}

.content_hp a#header_more {
    float:right;
    margin-right:94px;
    padding-left:12px;
    font-size:13px;
    color:#517317;
    text-decoration:none;
    background:url(images/puce_header_more.png) left center no-repeat}
.content_hp a#header_more:hover { text-decoration:underline;}
.content_hp p { color:#595959; font-size:14px; }
```

Il n'y a ici rien de compliqué, prenez le temps de bien lire chaque propriété.

Le bloc d'actualités

```

aside#news {
    /*On définit bêtement les propriétés de placement et de dimensions au bloc...*/
    width:530px;
    height:325px;
    padding:35px 15px 40px 15px;
    margin-top:40px;
    /*Puis on le fait flotter à gauche pour qu'après les 2 blocs suivants se placent à
côté (et non en-dessous)*/
    float:left;
    background:url(images/bg_news.jpg) top left no-repeat;}

#news article { position:relative; padding-left:50px;} /*Chaque article reçoit une
position:relative afin de placer les éléments qu'il contient à partir de son origine. C'est
utile pour le footer de l'article (voir plus bas)*/

#news h2 { visibility:hidden;} /*On cache le <h2> mais on laisse l'espace qu'il occupait
pour pousser le contenu vers le bas. Le titre est intégré dans le background en image.*/
#news h3 a{
    font-size:15px;
    font-weight:bold;
    color:#270f0a;
    text-decoration:none;}
#news p{ font-size:13px; color:#270f0a}
#news p.lire_suite a{
    font-size:13px;
    font-style:italic;
    color:#830f00;
    text-decoration:none;}
#news p.lire_suite a:hover{ text-decoration:underline;}

#news footer {
    /*On positionne ce bloc en absolu par rapport à l'article conteneur (grâce à notre
précédent position:relative sur <article>)*/
    position: absolute;
    /*Ainsi il se place tout en haut à gauche de l'article.*/
    left: 0px;
    top:0px;
    width: 38px;
    height:61px;
    padding:19px 0 0 13px;
    background:url(images/bg_date.png) no-repeat top left;
}
#news footer time { font-size:15px; color:#5c1100; font-family:Ginko;}
#news footer time span { font-family:Arial, Helvetica, Verdana, sans-serif;}
.contenu_article {margin-bottom:23px;}
.contenu_article p {margin:7px 0 0px;}

```

Toujours de haut en bas et de gauche à droite dans ma méthode de mise en page CSS, passons au bloc d'actus.

Les commentaires devraient suffire à éclaircir ce gros bout de code mais je préfère expliquer deux points moins habituels que les autres.

Le titre du bloc (<h2>) a bien été écrit en HTML mais dans un soucis de gain de temps, le titre a été intégré directement dans l'image de fond du bloc, il n'est donc pas nécessaire de laisser s'afficher celui en

HTML par-dessus. Cela explique la propriété **visibility:hidden** qui lui est appliquée. La balise <h2> est bien là, notamment pour le **référencement**, l'espace qu'elle occupe est conservé mais en revanche le texte est simplement caché pour laisser apparaître le texte de l'image de fond.

Le point plus délicat est mon **placement des <footer>** (le petit bloc de date à gauche des actualités). Ils sont placés en **position:absolute** par rapport à l'origine de la balise <article> qui le contient. Pour cela chaque <article> s'est vu assigné un position:relative. Ainsi lorsqu'un élément à l'intérieur est placé en absolute, le navigateur comprend qu'il est placé par rapport à la position de la balise <article>.

Ainsi avec **top:0px;** et **left:0px;** je place mes <footer> tout en haut à gauche dans le bloc de la balise <article>. Ensuite je « pousse » le titre et le texte de l'actualité en ajoutant un padding à <article>. Essayez par vous-même en écrivant les lignes une par une, cela sera plus clair si ça ne l'est pas !

Les blocs en bas à droite

Courage, on approche de la fin de la page !

```
#store_locator ul { list-style-image:url(images/puce_feuille.png); padding-left:28px;} /*On
ajoute notre image de feuille comme puce pour chaque élément de la liste*/
#store_locator ul li {float:left; width:160px;} /*On indique que tous les <li> flottent à
gauche*/
#store_locator ul li:nth-child(2n) {float:right;} /*Ici le sélecteur CSS3 permet de faire
flotter à droite un <li> sur deux (2n)*/
#store_locator ul a { text-decoration:none; color:#3c580c;}
#store_locator ul a:hover { text-decoration:underline;}

#home_bas {
    width:393px;
    float:left;
    color:#414141;
    margin-top:40px;
    margin-left:32px;}

#newsletter {
    margin-top:30px;
    height:225px; /*La hauteur est nécessaire pour pouvoir afficher l'image tout en bas*/
    background:url(images/pot_newsletter.png) no-repeat bottom right;}
#newsletter input[type=email] {width:225px;}
#newsletter input[type=submit] {
    width:103px;
    color:#ffe1c1;
    height:26px;
    margin-top:8px;
    text-shadow:0px -1px #802a15;
    border:none;
    background:url(images/bg_btn.jpg) no-repeat center top;
}
```

Il n'y a dans cette partie rien de très compliqué mais il y a **une nouveauté CSS3 qui repose dans les nouveaux sélecteurs**. Cet outil très puissant permet de choisir de façon périodique un élément comme bon vous semble. J'exécute la ligne :

```
#store_locator ul li:nth-child(2n) {float:right;}
```


Comme il est dit en commentaire, cette syntaxe « **:nth-child(2n)** » permet ici de sélectionner tous les n éléments spécifiés. Donc ici j'applique mon *float:right* à 1 sur 2. Si j'avais eu « **li:nth-child(4n)** », j'aurais appliqué ma propriété à 1 sur 4. A chaque fois 1 flotterait à droite, puis 3 à gauche, puis à nouveau 1 flotter à droite, puis 3 à gauche, etc... Le reste est relativement basique, je n'entrerais pas dans le détail !

Le footer... et c'est *presque* fini !

```
#footer_site {
    padding: 15px 0;
    height:65px;
    position: relative;/* donne hasLayout à IE6 de façon à permettre un effacement correct
*/
    background:url(images/bg_footer.png) no-repeat center top;
}

#footer_site aside {float:left; padding-left:56px;}
#footer_site aside p+p {margin:-1px 0 0 7px;}
#footer_site p { margin: 3px 0; float:left; color:#45444; font-size:13px;}
#footer_site p#copyright { float:right; color:#6e6e6e; font-size:12px;}
#footer_site p#copyright a { color:#556d2f;}
```

Ici le code est tout à fait basique : j'ajoute un fond au footer et je place les différents éléments à l'aide de *float left* et *right*.

Fin du tutoriel

Enfin, voici la fin de ce tutoriel long et compact. Voici ce que nous avons réalisé et appris :

- découpage de la maquette en plusieurs images optimisées
- écriture du squelette HTML en suivant les recommandations sur l'HTML5
- explication des nouvelles balises HTML5 et de nombreux attributs et spécificités
- mise en page en CSS
- emploi et explication des nouveautés apportées par le CSS3 (multiple backgrounds, nouveaux sélecteurs, text-shadow, etc...)
- techniques et astuces nécessaires pour l'emploi de l'HTML5 pour votre projet

NB : Il existe toujours plusieurs façons d'intégrer une page HTML/CSS il est donc possible que vous soyez tentés d'utiliser différentes méthodes ou différentes balises et propriétés selon les cas. Cela est normal !

Fin du tutoriel