

Nous avons déjà appris qu'un site était constitué de **pages Web** définissant un **contenu**, et d'un **fichier CSS** indiquant **comment présenter ce contenu** (polices, couleurs etc...). Il faut à présent travailler la **mise en page du site** : le **design**.

Une page web est très souvent constituée d'un **en-tête** (bannière en haut), de **menus de navigation** (en haut ou sur les côtés), de **différentes sections** au centre dans le "corps" de la page... et d'un **pied de page** (en bas).

De nouvelles balises ont été introduites par HTML5 pour structurer les sites.

Travail préalable :

Créez un nouveau dossier sous votre dossier Web, que vous nommez "**DesignSite**" (Rappel : pas d'espace, ni accents, ni caractères spéciaux dans les noms de dossiers Web, ou nom de pages html ou CSS etc...), et récupérez le dossier images et le document word mis à votre disposition sous Ressources.

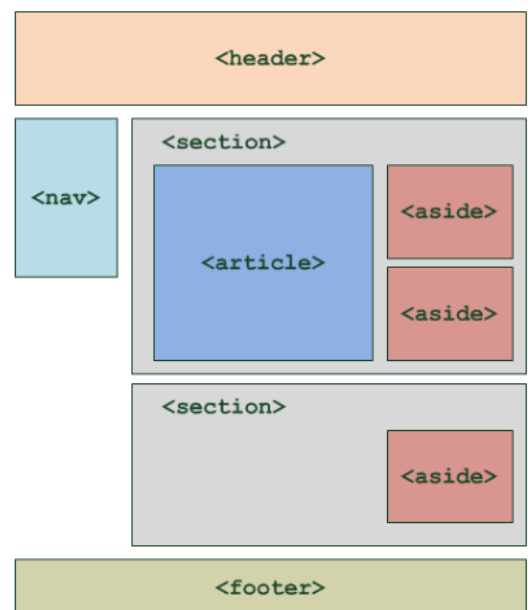
1 – Structurer le site avec les balises HTML5

Avant de créer un site Web, il faut tout d'abord réaliser le **maquettage** (dessin avec la structure des pages) et créer une **charte graphique** (couleurs, polices etc...) que l'on retrouvera sur toutes les pages du site. Dans chaque page, on doit pouvoir distinguer du 1^{er} coup d'œil les grandes zones qui la composent.

Ensuite seulement, on passe au HTML.

Les **nouvelles balises HTML5** permettent de mettre en forme les pages, comme dans l'exemple ci-contre :

Remarque : HTML5 a introduit un ensemble de nouveaux éléments ayant chacun une fonction sémantique précise. Ceci a pour but principal de faciliter l'indexation du contenu des pages par les moteurs de recherche.



Code HTML :

- Les informations de l'en-tête, ou la bannière sont placées à l'intérieur de la balise **<header>**. Dans le header, on peut placer des images, des liens, des textes...

```
<header>
  <!-- contenu de l'en-tête de la page -->
</header>
```

- Les informations du pied de page sont placées à l'intérieur de la balise **<footer>**. On y trouve des informations comme des liens de contact, le nom de l'auteur, les mentions légales, etc.

```
<footer>
  <!-- contenu du pied de page -->
</footer>
```

- ☞ Les principaux liens de navigation sont regroupés dans la balise **<nav>**.
Généralement, le menu est réalisé sous forme de liste à puces à l'intérieur de la balise **<nav>** :

```
<nav>
  <ul>
    <li><a href="index.html">Accueil</a></li>
    <li><a href="contact.html">Contact</a></li>
    <li><a href="pagebts.html">BTS SIO</a></li>
    <li><a href="equipe.html">L'équipe</a></li>
  </ul>
</nav>
```

- ☞ La balise **<section>** sert à regrouper des contenus par thème. C'est en général une partie dans le corps de la page.

```
<section>
  <h1>Voici une section de page</h1>
  <p>et mon contenu de section ...</p>
</section>
```

A l'intérieur de la balise section :

- ☞ La balise **<aside>** contient des informations complémentaires à la page, placées par exemple sur le côté droit (une explication complémentaire, une photo, etc ...).
On peut avoir plusieurs blocs **<aside>** dans la page.

```
<aside>
  <!-- informations complémentaires -->
</aside>
```

- ☞ La balise **<article>** contient une partie autonome de la page : par exemple des actualités.

```
<article>
  <h1>Mon article</h1>
  <p>rédaction ...</p>
</article>
```

Remarque1 : on peut aussi trouver la balise **<div>** **</div>** pour structurer le corps de la page.

Exemple : **<div id="section1">** ... avec dans le code CSS une balise : **#section1** pour les propriétés CSS de la section n°1.

(revoir class et id dans le chapitre précédent)

2 – Premier exemple d'application : structurer un site

 Sous votre éditeur Notepad++, réalisez la structure de l'exercice "site du lycée" comme ci-dessous:
Code HTML de la page **index.html**. Vous trouverez dans le fichier **Brassens.docx** quelques contenus de textes à insérer (vous mettrez seulement les 1ères parties entre les tirets pour commencer), puis testez.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="design_brassens.css" />
    <title>mon site brassens - Page index.html</title>
  </head>
```

```

<body>
  <header>
    <!-- en-tête -->
    <h1> Mon site du Lycée Brassens </h1>
  </header>

  <nav>
    <!-- menu -->
    <ul>
      <li><a href="index.html">Accueil</a></li>
      <li><a href="contact.html">Contact</a></li>
      <li><a href="pagebts.html">Le BTS SIO</a></li>
      <li><a href="equipe.html">L'équipe</a></li>
    </ul>
  </nav>

  <section>
    <article>
      <!-- article présentation du lycée -->
      <h1>Présentation du Lycée</h1>
      <p>. . . (1ère partie du texte page 1 du doc)</p>
    </article>
    <aside>
      <!-- info complémentaire -->
      <h1>À propos de Georges Brassens</h1>
      <p>. . . (1ère partie du texte page 2 du doc)</p>
    </aside>
  </section>

  <footer>
    <!-- pied de page -->
    <p>Copyright - S T - Tous droits réservés<br /></p>
  </footer>

</body>
</html>

```

→ Vous vous apercevez que le design n'est pas encore fait à ce stade, c'est le rôle du CSS : on indiquera en background du header la photo bannière lycée, la position de l'article et de l'information complémentaire etc...

3 – Les blocs

Une page web est constituée de plusieurs "**blocs**". La plupart des éléments vus jusqu'à présent sont des blocs : <header>, <article>, <nav>... ainsi que les paragraphes <p>, les titres <h1>...

Nous allons apprendre à utiliser ces blocs pour les placer correctement dans les pages d'un site : il faut leur donner des dimensions, les agencer en jouant sur leurs marges, gérer leur contenu...

Ce sont des **notions fondamentales** pour mettre en page un site web.

3.1 Les différents types de balises

- **block** : une balise de type **block** sur votre page web crée automatiquement un retour à la ligne avant et après. La page web sera en fait constituée d'une série de blocs les uns à la suite des autres. Mais on peut aussi mettre un bloc à l'intérieur d'un autre, ce qui va augmenter considérablement les possibilités pour créer le design des sites.

- **inline** : une balise de type **inline** se trouve obligatoirement à l'intérieur d'une balise block. Une balise inline ne crée pas de retour à la ligne, le texte qui se trouve à l'intérieur s'écrit donc à la suite du texte précédent, sur la même ligne (c'est pour cela que l'on parle de balise « en ligne »).

Exemples :

Balises block	Balises inline
<p>	
<footer>	
<h1>	<mark>
<h2>	<a>
<article>	

Les balises universelles :

Ce sont les balises qui n'ont aucun sens particulier (contrairement à <p> qui veut dire « paragraphe », « important », etc.).

L'intérêt de ces balises est que l'on peut leur appliquer une class (ou un id) pour le CSS quand aucune autre balise ne convient.

Il existe deux balises génériques: l'une inline et l'autre block :

- (**inline**)
- <div></div> (**block**)

Attention, ne les utilisez que lorsque vous ne pouvez pas utiliser des balises déjà existantes !

<div class="titre"> est à proscrire puisque les balises <h1>, <h2> existent déjà...

3.2 Propriétés CSS dans les balises "block"

☞ Pour dimensionner les blocs, on dispose de 2 propriétés CSS :

- **width** : largeur du bloc. À exprimer en pixels (px) ou en pourcentage (%).
- **height** : hauteur du bloc. On l'exprime soit en pixels (px), soit en pourcentage (%).

Par défaut, un bloc prend 100% de la largeur disponible

Exemple CSS

```
p
{
    width: 50%;           /* tous les paragraphes ont une largeur de 50% */
    ou width: 200px;      /* tous les paragraphes font 200 pixels de large */
}
```

On peut aussi spécifier :

```
p
{
    width: 50%;           /* les paragraphes occupent 50% de la largeur */
    min-width: 400px;     /* et ils font au moins 400 pixels de large */
}
```

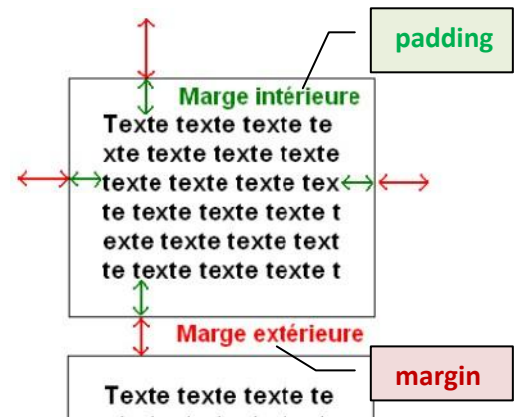
☞ Pour **spécifier des marges** dans les blocs :

Marges intérieures : espace entre le texte et la bordure du bloc

Marges extérieures : espace entre la bordure du bloc et le bloc suivant.

En CSS, on peut modifier la taille des marges avec les 2 propriétés suivantes :

- **padding** : indique la taille de la **marge intérieure** - à exprimer en général en pixels (px).
- **margin** : indique la taille de la **marge extérieure** - en pixels.



🔗 Testez avec le code ci-dessous dans un fichier CSS "design_martin.css" que vous créez pour ce chapitre, et votre page HTML précédente. Testez avec 2 paragraphes <p> dans votre partie <article> : présentation du lycée.

```
p
{
  width: 350px;
  border: 1px solid black; /* bordure pour bien voir les marges */
  text-align: justify;
  padding: 15px;           /* marge intérieure à tester */
  margin: 50px;           /* marge extérieure à tester */
}
```

Si l'on veut spécifier des marges différentes en haut, en bas, à gauche et à droite, on doit utiliser des propriétés plus précises :

- **margin-top** : marge extérieure en haut
- **margin-bottom** : marge extérieure en bas
- **margin-left** : marge extérieure à gauche
- **margin-right** : marge extérieure à droite

idem pour padding.

☞ Pour **centrer les blocs** :

- donnez une largeur au bloc (avec la propriété width)
- indiquez que vous voulez des marges extérieures automatiques avec : **margin: auto**

```
p
{
  width: 350px;           /* indication de la largeur (obligatoire) */
  margin: auto;           /* le bloc est centré avec auto */
  border: 1px solid black;
  text-align: justify;
  padding: 15px;
  margin-bottom: 50px;
}
```

🔗 Testez sur vos 2 blocs paragraphe → centrage horizontal

☞ Pour **couper un bloc** : Si le texte ne "tient" pas dans votre cadre, on peut mettre en place une barre de défilement

```
p
{
    width: 350px;
    height: 110px;
    text-align: justify;
    border: 1px solid black;
    overflow: auto;          /* barre de défilement pour le texte qui dépasse */
}
```

 Testez avec un texte qui dépasse du bloc

4 – Positionner les éléments sur une page avec le CSS

Il existe différentes techniques permettant d'effectuer la mise en page de son site.

4.1 Le positionnement flottant

Si on veut positionner par exemple le **menu** (avec les liens de navigation) à gauche et la **section** (corps) à droite

→ On va faire "**flotter**" le menu à gauche et laisser le reste du texte se placer à sa droite.

 Testez le CSS suivant

```
nav
{
    float: left;                /* le menu doit flotter à gauche */
    width: 150px;              /* le menu occupe 150 pixel de large */
    border: 1px solid black;    /* bordure noire autour du menu */
}

section
{
    margin-left: 160px;        /* marge extérieure à gauche de 160 pixel */
    height: 500px;             /* hauteur du bloc de 500 px */
    border: 1px solid blue;     /* bordure bleue autour du corps */
}
```

A BIEN COMPRENDRE !!! La marge extérieure à gauche de la <section> est de **160 px** car elle qui doit être supérieure à la largeur du menu <nav> de **150 px**, pour que ce menu puisse se caler à sa gauche.

4.2 La propriété display

La propriété CSS display est très puissante car elle **permet de transformer** n'importe quel élément de votre page **d'un type vers un autre**.

Exemples :

```
a
{
    display: block;          /* transforme des liens à l'origine "en ligne" */
                                /* sous forme de bloc, donc les uns en dessous des autres */
}

.invisible
{
    display: none;          /* rend invisible un élément possédant cette propriété */
}
```

4.3 Le positionnement inline-block

Nous avons vu le mode de positionnement "flottant". Il existe une autre technique qui consiste à transformer vos éléments en **inline-block** avec la propriété display.

Les éléments de type **inline-block** :

- Ils se positionnent les uns à côté des autres : comme le menu et le corps de la page par exemple, ou pour 2 blocs de texte dans le corps de la page (bloc section).
- On peut leur donner des dimensions précises.

Nous allons transformer en **inline-block** les deux éléments que nous voulons **placer côte à côte** : le bloc <article> et le bloc <aside> par exemple :

(On pourrait faire exactement la même chose avec le menu <nav> et le corps de la page <section>)

 Testez le CSS suivant

```
article
{
    display: inline-block;          /* pour les positionner côte à côte */
    width: 600px;
    vertical-align: top;           /* aligné en haut */
    border: 1px solid black;
}
aside
{
    display: inline-block;          /* pour les positionner côte à côte */
    width: 300px;
    vertical-align: top;           /* aligné en haut */
    border: 1px solid blue;
}
```

Remarque : la propriété **vertical-align** peut prendre les valeurs suivantes :

- **baseline** : aligne de la base de l'élément avec celle de l'élément parent (par défaut) ;
- **top** : aligne en haut ;
- **middle** : centre verticalement ;
- **bottom** : aligne en bas ;
- (valeur en px ou %) : aligne à une certaine distance de la ligne de base (baseline).

Le positionnement absolu et relatif

Ces positionnements fonctionnent sur des balises de type inline.

La propriété CSS **position** peut prendre les valeurs suivantes :

- **absolute** : positionnement absolu ;
- **fixed** : positionnement fixe ;
- **relative** : positionnement relatif.

Exemple : l'élément est positionné tout en haut à droite (0 px par rapport à la droite de la page, et 0 par rapport au haut de la page).

```
element
{
    position: absolute;
    right: 0px;
    top: 0px;
}
```

Avec `position: fixed;`

C'est à peu près la même chose qu'avec **absolute**, sauf que, le bloc reste fixe à sa position, même si on descend plus bas dans la page.

Le positionnement **relative** permet d'effectuer des « ajustements » : l'élément est décalé par rapport à sa position initiale. Ce positionnement est difficile à utiliser.

5 – Finalisation du site "Mon site de Georges Brassens"

Travail préalable :  Supprimez les balises tests de votre fichier CSS, sauf <nav> et <section> (Ceci pour repartir proprement)

Amélioration progressive du design :

 Testez en ajoutant les propriétés CSS petit à petit pour voir les transformations du site (actualisation F5). Tapez les commentaires pour comprendre ce qu'on souhaite faire ici (et les retrouver plus tard).

Dans votre code HTML :

- Videz votre **header** (ne gardez que les 2 balises <header> et </header>)
- Dans votre bloc <aside>, ajoutez la photo de Georges brassens :

```
<h1> À propos de Georges Brassens</h1>
<p></p>
```

Ajoutez dans votre code CSS :

```
/* fichier css */
body
{
    width: 850px;                /* Pour fixer la largeur de la page */
    margin: auto;                /* Pour centrer la page */
    margin-top: 20px;            /* Pour éviter de coller avec le haut du navigateur */
    margin-bottom: 20px;         /* Idem pour le bas du navigateur */
    background-image: url("images/fond1.png");    /* Image de fond */
}
h1
{
    text-align: center;          /* pour centrer les titres */
}
header /* en-tête */
{
    width: 850px;
    height: 95px;
    background-image: url("images/banniere.jpg"); /* bannière en fond */
    background-repeat: no-repeat;
    margin-bottom: 10px;
    border-radius: 5px;
}
```



```

footer      /* pied-de-page */
{
    text-align: right;      /* signature alignée à droite */
    font-style: italic;    /* en italique */
}
nav         /* menu */
{
    float: left;
    width: 150px;
    border: 1px solid black;    /* à enlever par la suite */
}
section
{
    height: 500px; margin-
    left: 160px;
    border: 1px solid blue;    /* à enlever par la suite */
}

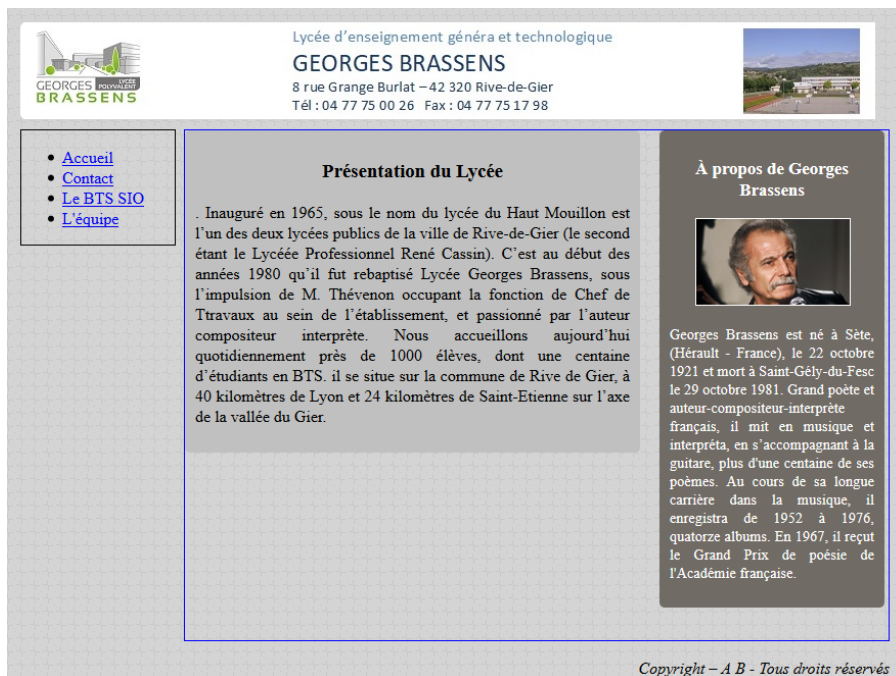
article, aside    /* propriétés pour les 2 attributs */
{
    display: inline-block;    /* pour les positionner côte à côte */
    vertical-align: top;      /* Tous deux alignés en haut */
    text-align: justify;      /* texte justifié (aligné à droite et à gauche) */
}
article         /* bloc : présentation du lycée */
{
    width: 425px;            /* largeur du bloc */
    background-color: #C0C0C0;
    border-radius: 5px;
    color: black;
    padding: 10px;          /* marge intérieure pour le texte */
}

aside           /* bloc : à propos de Georges Brassens */
{
    position: relative;      /* position relative % à l'article ci-dessus */
    width: 200px;
    background-color: #706b64;
    border-radius: 5px;
    margin-left: 15px;        /* marge ext. gauche pour ne pas coller à l'article */
    padding: 10px;          /* marge intérieure pour le texte */
    color: white; font-
    size: 0.9em;
}

.photoGeorges    /* propriétés de la photo de Georges */
{
    margin-left: 25px;
    width: 150px;
    border: 1px solid white;
}

```

Vous devriez avoir quelque chose qui ressemble à ceci pour le moment ...



Continuez à améliorer ce site :

- ☞ **Menu** à mettre en forme ☐ fond blanc, bords arrondis, liens "purple" en italique non soulignés, changement de couleur gris au survol...
- ☞ **Photo** du lycée à insérer dans l'article (image flottante par exemple), bord blanc, bulle ...
- ☞ Ajoutez **les sauts de ligne** dans les textes (pour espacer les paragraphes, utilisez les balises `<p>` plutôt que les `
`)
- ☞ Textes à **compléter** avec le document Word (suite articles), avec mise en place des **puces** pour les tirets du texte, et testez la mise en place d'une **barre de défilement** pour la section (vous devrez alors réduire la largeur de l'article à 400px environ)
- ☞ Page **contact** à faire (contact.html) : **recopiez le même code HTML** que la page index.html, et **modifier votre <section> seulement**, à l'aide du document Word fourni.
- ☞ Page **BTS SIO** (pagebts.html) : idem, recopiez la page index.html et ne modifiez que la partie `<section>`.

