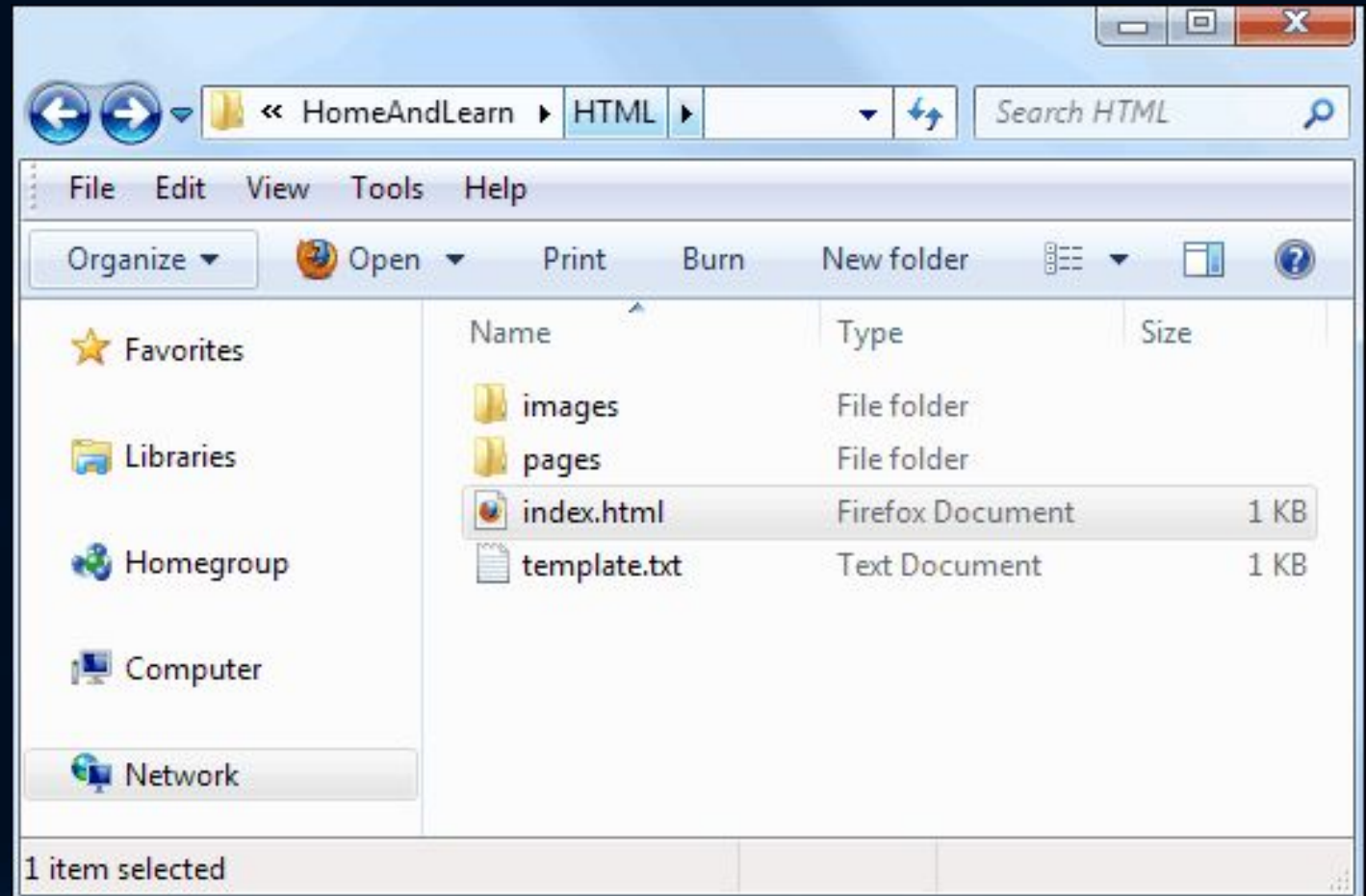


The background is a dark blue gradient. On the right side, there is a bright blue, glowing, curved shape that resembles a stylized 'S' or a series of concentric, overlapping curves. This shape is composed of many fine, parallel lines that create a grid-like texture, giving it a three-dimensional, tunnel-like appearance.

Savoir structurer sa
page

Page index.html

- La page index.html sera toujours celle qui sera appelé par le navigateur afin d'afficher votre site



Indiquer la langue d'un contenu

- **Il est important sur une page web d'indiquer clairement la langue du contenu.** Les informations données sur la ou les langues du contenu seront utiles pour les outils d'indexation (moteurs de recherche), les outils de traduction automatique ou encore ceux de synthèse vocale.
- Il existe aussi une méta-donnée qui porte sur l'ensemble du document: l'en-tête HTTP. Cette méta-donnée correspond à la langue primaire du document, c'est à dire la langue du public visé. Par exemple, un site gouvernemental français ou un site de commerce en ligne qui dessert uniquement la France.

```
<!DOCTYPE html>  
<html lang="fr">  
  <head>  
  </head>  
  <body>  
  </body>  
</html>
```

```
<p>  
  <cite lang="en">My name is </cite> est la  
  traduction française de « Mon nom est ».  
</p>
```

```
<meta lang="content-language"  
content="fr-FR" />
```


Les balises <meta>

- On place ces balises entre les balises <head></head>
- Les balises <meta> servent à donner des informations sur la page, comme :
 - Les mots-clés
 - La description
 - La langue
 - L'auteur
 - Ect...

```
<head>  
  <meta charset="UTF-8">  
  <title>Demo Portfolio Website</title>  
  <meta name="description" content="I'm a  
  photographer who's traveling around the world.">  
  <meta name="keywords" content="photographer  
  portfolio">  
</head>
```

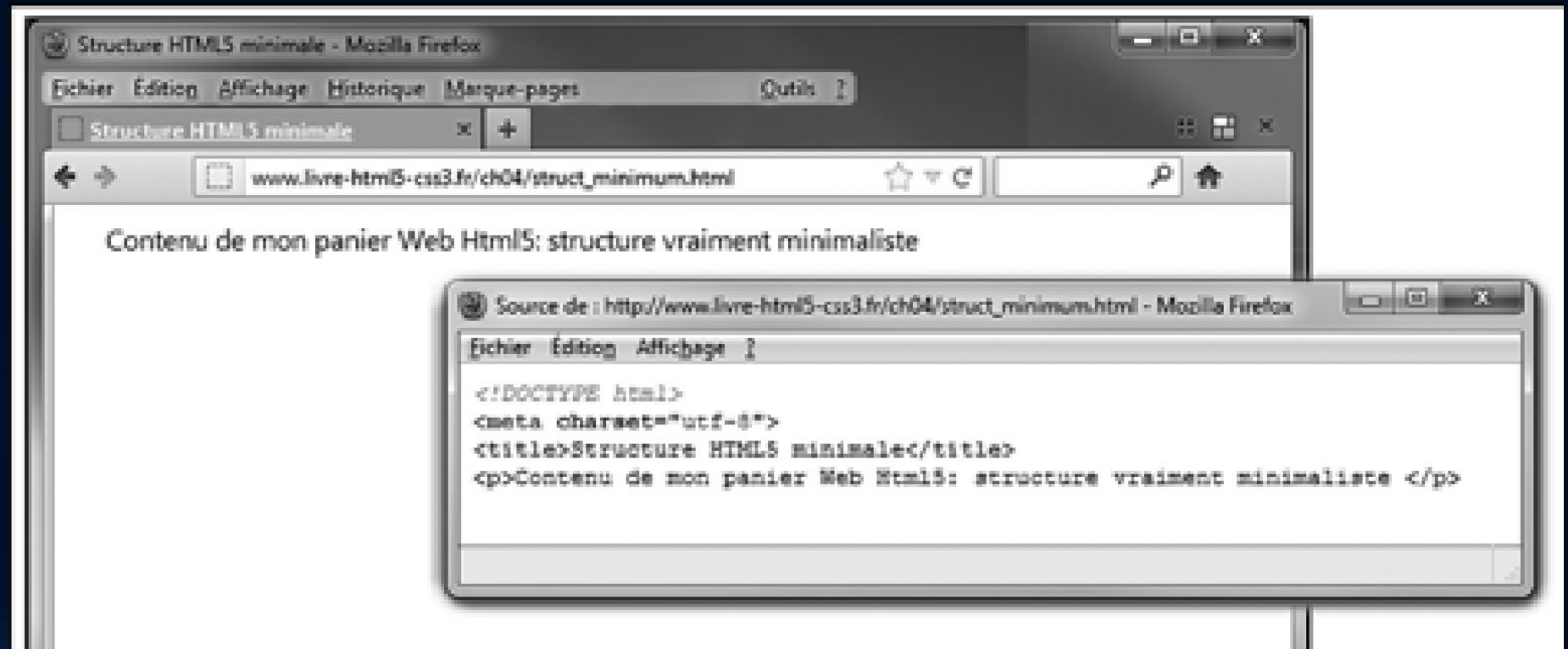
Balises structurantes et W₃C

Structures et balises


- L'une des principales nouveautés apportées par l'HTML5 est sans contexte l'arrivée de balises structurelles sémantiques :
 - Structurelles car elles portent sur l'organisation des informations disponibles sur un site.
 - Sémantiques car elles permettent de cataloguer le type de ces informations.
- Ces balises faisant partie du code HTML, le visiteur ne les remarque pas de prime abord, à l'inverse du navigateur et des divers robots des moteurs de recherche qui, eux, s'en rendent compte, et devraient même s'en féliciter s'ils le pouvaient, car c'est pour eux le gage d'une meilleure lecture des contenus leur permettant une meilleure interprétation.

Tendance vers la simplification du code

Le squelette d'une page HTML5 peut désormais être résumé à quelques lignes. Ainsi, une page en HTML5 pourrait s'écrire de cette manière :



Et serait considéré comme valide par le validateur W3C :

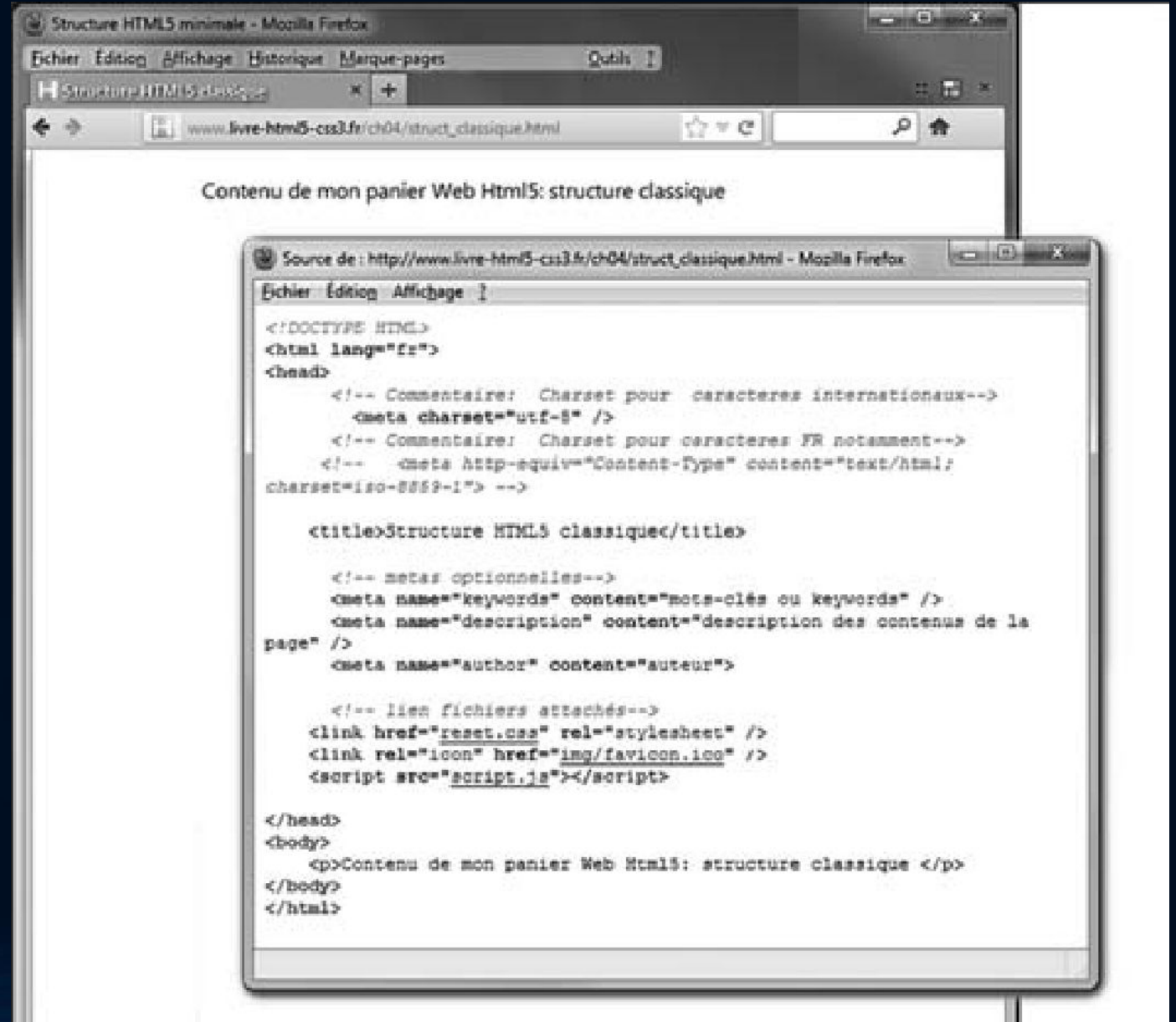
 **Markup Validation Service**
Check the markup (HTML, XHTML, ...) of Web documents

[Jump To:](#) [Notes and Potential Issues](#) [Congratulations · Icons](#)

This document was successfully checked as HTML5!

Result:	Passed	
Address :	<input type="text" value="http://www.livre-html5-css3.fr/ch04/struct_minimum.html"/>	
Modified:	(undefined)	
Server:	Apache/2.2.X (OVH)	
Size:	144	
Content-Type:	text/html	
Encoding :	utf-8	<input type="text" value="(detect automatically)"/>
Doctype :	HTML5	<input type="text" value="(detect automatically)"/>
Root Element:	html	

Cela dit, même si cette écriture minimaliste est validée, il est sans doute judicieux et plus prudent de garder les bonnes habitudes en continuant à inscrire les balises de bases dans la structure des pages, qu'elles fussent HTML 4.01 ou HTML5. Le code et vos collègues développeurs y gagneront en clarté et en compréhension car, bien sûr, elles restent toujours d'actualité et valides.



Accessoirement, on peut aussi se poser des questions concernant l'intérêt de la validité HTML par les outils W3C dont la suivante : un code validé est-il toujours meilleur ou préférable à un code non validé ? En fait, je suis d'avis de dire que la validation n'est qu'un moyen pour affirmer que votre code respecte la syntaxe et qu'il n'y a pas d'erreurs majeures, mais elle n'apporte rien au niveau de la qualité des contenus, de leur organisation, de leur hiérarchisation, de leur marquage sémantique, bref de ce qui rend un site intéressant pour le visiteur qui vient chercher une information.



Les balises structurantes HTML5

SUCCESEURS DES BALISES <DIV>

Structuration des contenus et balisage sémantique

- Architecturer les pages Web par un balisage sémantique présente de multiples intérêts. Cela permet de :
 - hiérarchiser l'information ;
 - uniformiser, simplifier et affiner la structure des pages ;
 - faciliter l'analyse et la lisibilité du code ;
 - permettre une meilleure analyse des contenus éditoriaux par les robots de référencement ;
 - ce qui, à terme, ne peut qu'améliorer le positionnement et l'accessibilité aux handicapés.

<header> : l'en-tête

- On retrouve les informations incontournables dans l'en-tête d'un site :
 - Logo
 - Bannière
 - Barre de recherche
 - Connexion
 - Etc...



Réduire la définition de cette balise au concept d'en-tête de page serait faux et très réducteur, tout comme croire que cela équivaldrait à ce que l'on faisait en HTML4 avec par exemple :

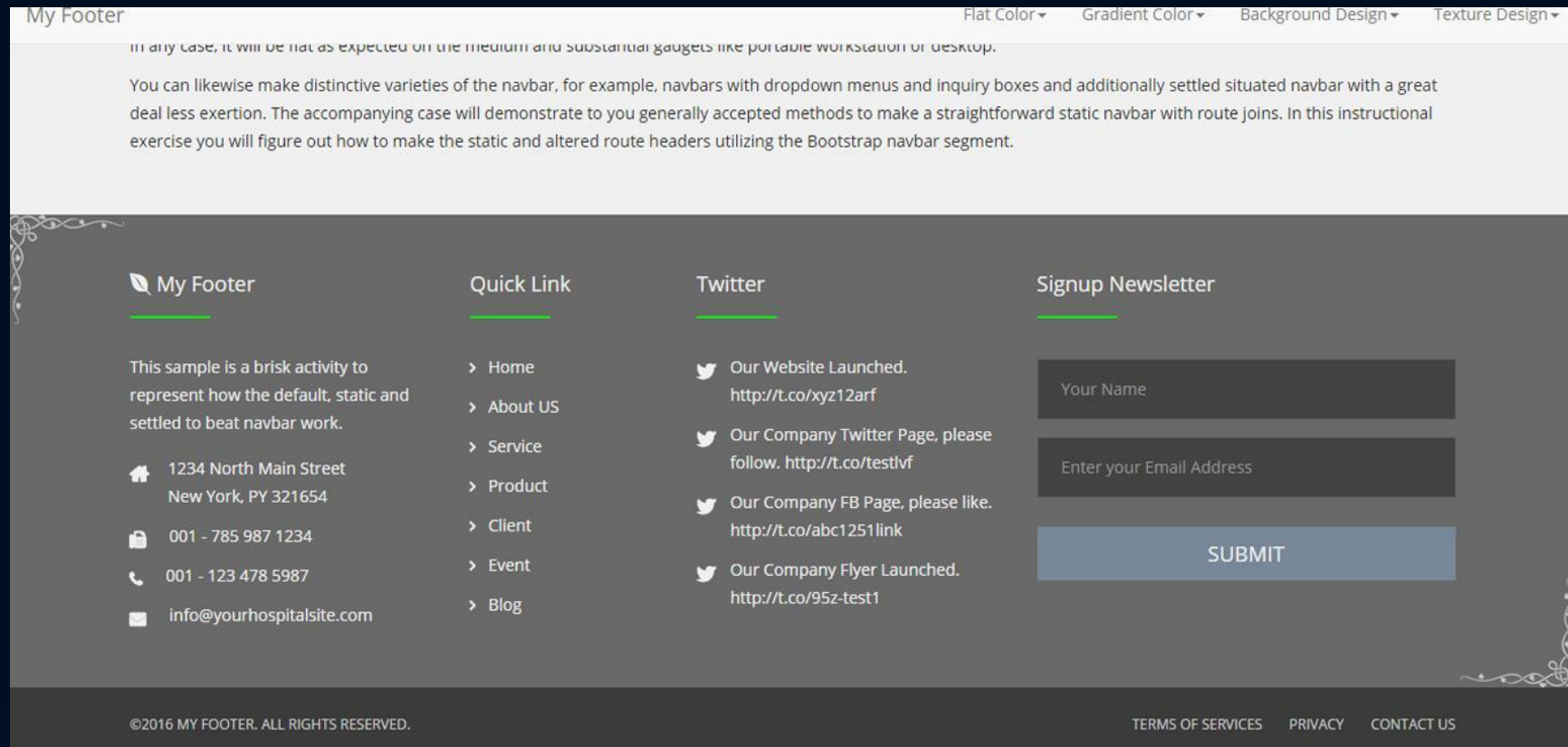
```
<div id="header">
  <div id="logo">.... </div>
  <ul id="menu">
    <li> menu1 </li>
    <li> .....</li>
  </ul>
</div>
```

En effet, à la différence de l'utilisation précédente, la balise <header> peut être présente à de multiples reprises dans la page et être la première rencontrée pour chaque élément différencié de la page (section, article, colonne...) afin d'apporter les éléments d'introduction, d'en-tête, de présentation, voire une table des matières pour chacun. Cela pourrait se traduire dans un schéma de page HTML5 comme ceci :

```
<header>
  <h1> Titre Page </h1>
</header>
<article>
  <header>.
    <h2> titre de l'Article</h2>
  </header>
  <p>.....</p>
</article>
```

<footer> : le pied de page

- Le pied de page se trouve tout en bas de la page.
- On y retrouve des informations telles que :
 - Contact
 - Mentions légales
 - Réseaux sociaux
 - Derniers articles
 - Map de navigation
 - Ect...



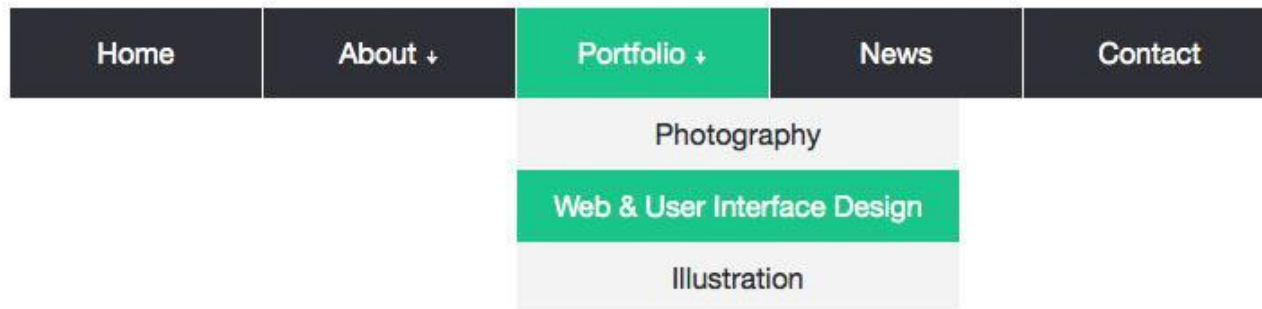
La balise *footer* est à la conclusion d'une section, d'une page ou d'un article ce qu'est la balise *header* à l'introduction ou la présentation de contenus des mêmes éléments. Comme cette dernière, elle est conçue pour être placée, de manière multiple, dans les différents nouveaux éléments structurants HTML5 de la page tels que <article>, <section>, etc., et représentera « le pied » de la balise parente ancêtre ou de la racine de la page, dans laquelle la balise <footer> est présente.

```
<footer>
  <ul>
    <li>Informations légales</li>
    <li>Contacts</li>
    <li>Où sommes nous ?</li>
  </ul>
</footer>
```


<nav> : les liens de navigation / menu

```
<nav>
  <ul>
    <li><a href="index.html">Accueil</a></li>
    <li><a href="blog.html">Blog</a></li>
    <li><a href="portfolio.html">Portfolio</a></li>
  </ul>
</nav>
```

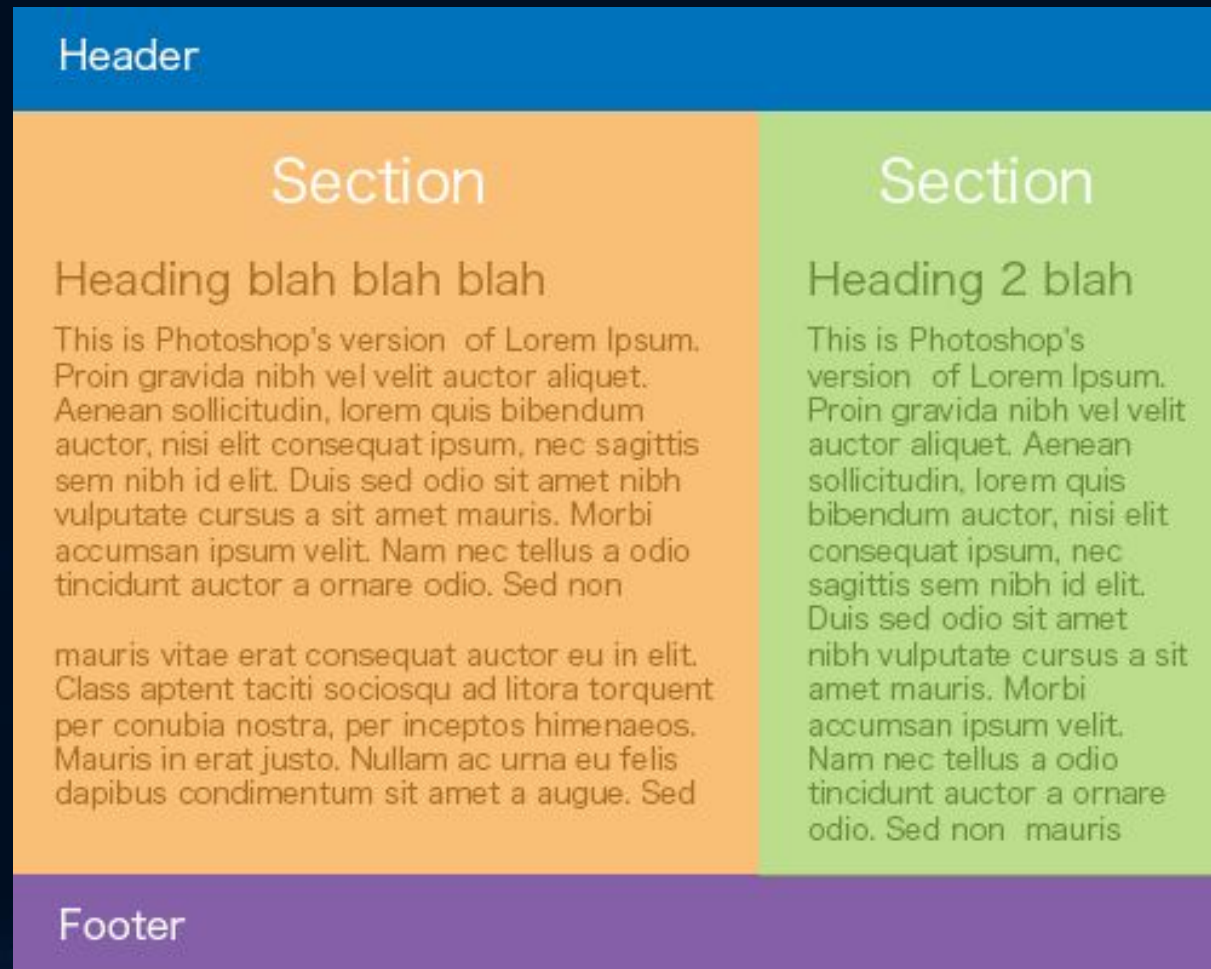
Cette balise inclut des éléments de navigation, plutôt de nature « navigation principale » de la page, et plutôt réservés à des liens de type interne.



P.S : La balise <nav> peut tout à fait se trouver dans la balise <header>

<section> : une partie de la page

- Idéale pour regrouper les différentes parties de votre page par thématique identiques ou proches.
- Les sections sont particulières et peuvent contenir un titre de niveau <h1> chacune sans que cela ne soit banni par les moteurs de recherche.



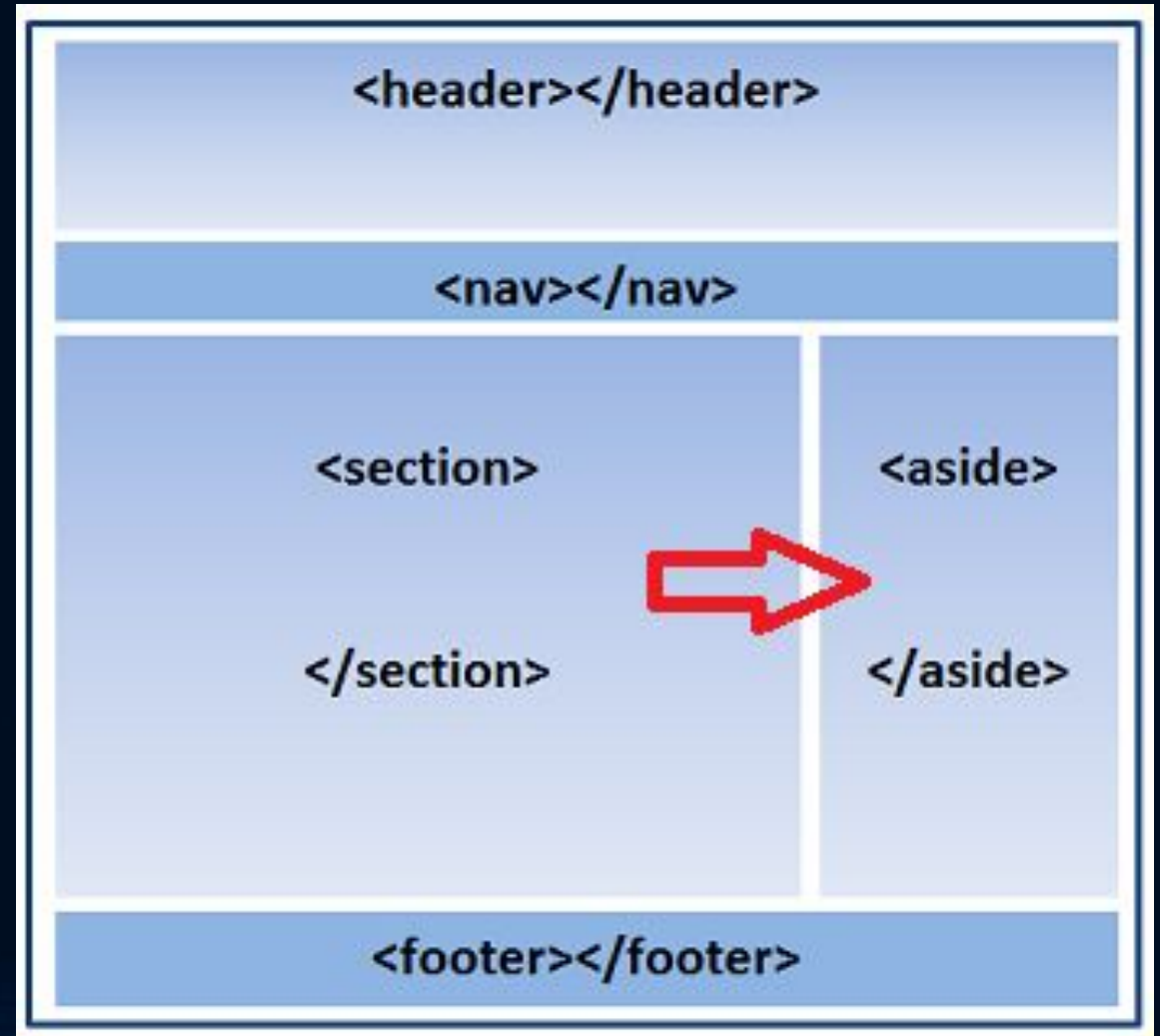
Encore un faux-ami, car son intitulé laisse penser qu'elle est adaptée pour élaborer la structure d'une page, en définissant ses grandes parties, comme on le fait avec la balise <div>. Eh bien non, car le W3C indique que la balise ne doit pas être utilisée à des fins génériques, pour une mise en page par exemple, ou pour faciliter un script. Dans ce cas, c'est bien la balise <div> qui reste recommandée.

La balise <section> est plutôt réservée au regroupement d'éléments traitant de la même thématique et délimitant de grandes zones génériques, comme la mise en place de chapitres, de dossiers ou d'articles conséquents. La première balise enfant rencontrée sera, dans la plupart des cas, une balise de type <h1 ...>. Son emploi reste donc encore assez flou et mal défini, surtout qu'il n'est pas rare de voir du code avec la balise <article> englobant la balise <section>, et inversement. L'analyse du contenu est donc particulièrement importante dans ce cas.

```
<article class="book">
  <header>
    <h1>My Book</h1>
    <h2>A sample with not much content</h2>
    <p>Published by Dummy Publicorp Ltd.</p>
  </header>
  <section class="chapter">
    <h1>Mon 1er Chapitre</h1>
    <p>Je ne dis pas tout dans mon premier
    chapitre</p>
    <p>Même avec deux paragraphes</p>
  </section>
  <section class="chapter">
    <h1>Continuons avec le deuxième
    chapitre</h1>
    <p>Bla bla bla. Vlan ! Damned, que d'actions
    !</p>
  </section>
  <section class="appendix">
    <h1>Annexe A</h1>
    <p>Le coin de mes bonnes adresses</p>
  </section>
</article>
```

<aside> : informations complémentaires

- La balise <aside> est conçue pour contenir des informations complémentaires au document que l'on visualise. Ces informations sont généralement placées sur le côté mais ce n'est pas une obligation.



La balise <aside> définit un contenu qui peut être considéré comme une annexe au contenu principal (contenu tangentiel suivant les termes du W3C, au contenu principal), et qui peut être autonome, comme pour <article>. Si jusque-là c'est encore assez clair, cela se complique suivant que cette balise sera une balise enfant d'une balise <article> ou non.

Dans le 1er cas, le contenu de la balise <aside> doit être en rapport avec celui de sa balise parente <article>. C'est le cas, par exemple, pour des contenus d'éléments annexes tels que des liens, des commentaires, des références bibliographiques pour un article par exemple.

```
<article>
  <section >
    <h1>Titre article</h1>
    <p>Jellentesque habitant morbi tristique
      senectus ....</p>
  </section>
  <section class="appendix">
    <h1>Annexe A</h1>
    <p>TLe coin de mes bonnes adresses</p>
  </section>
  <aside>
    <h1>Bibliographie</h1>
    <ul>
      <li>Maecenas pulvinar ...</li>
      <li>elit eget auctor posuere..</li>
    </ul>
  </aside>
</article>
```

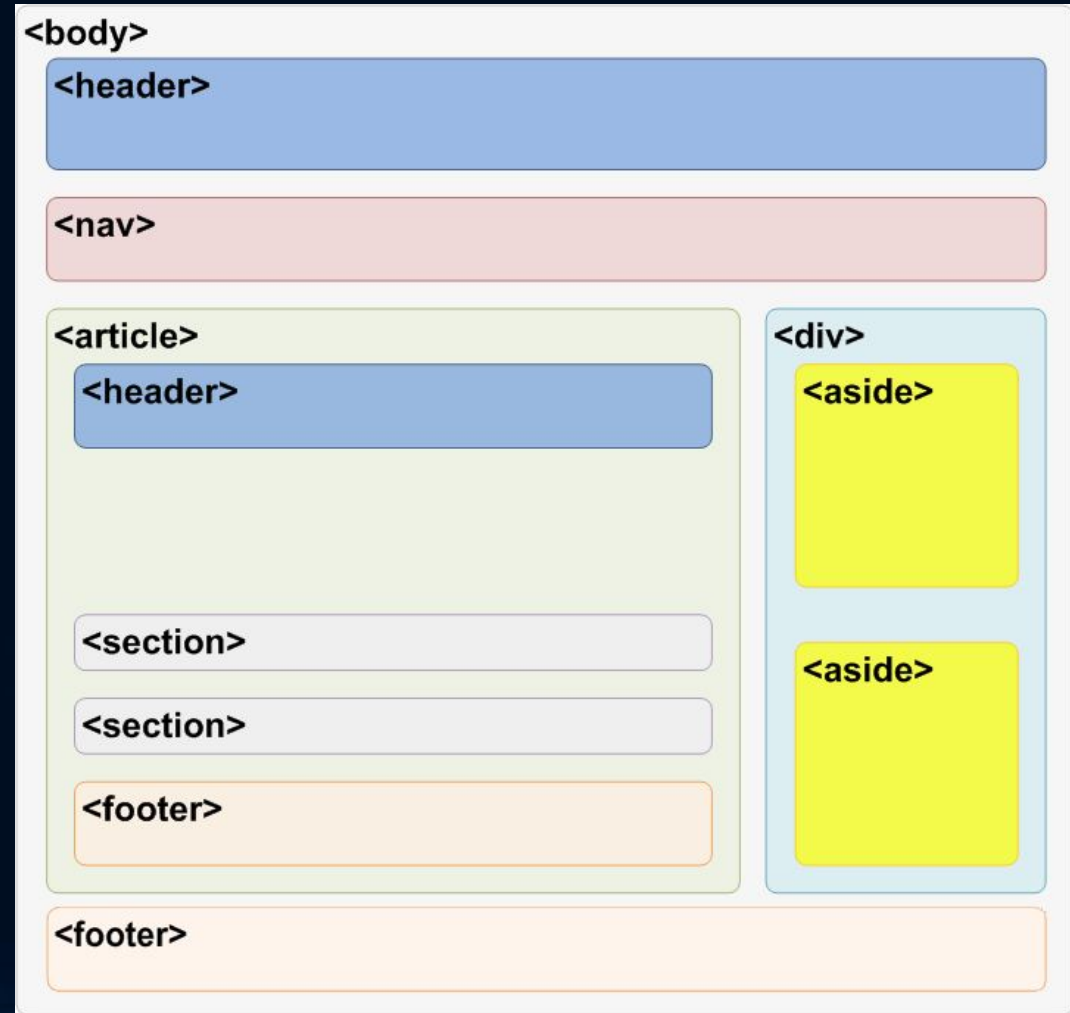
Par contre, si la balise `<aside>` est externe à toute balise `<article>`, elle doit alors être considérée comme une balise permettant d'inclure des éléments annexes à la page en général mais toutefois en relation avec ceux-ci. Dans ce cas, son utilisation est similaire à ce qu'on le fait avec les encadrés (encarts textes) du type : « Astuces », « À Savoir ».

Autres utilisations possibles : navigation secondaire (barre latérale), encart pub, nuage de tags, ect...

```
<aside>
  <h1>Dates à retenir</h1>
  <ul>
    <li><a href="/calandar/2012/jan.htm">Janv
      2012</a></li>
    <li><a href="calandar/2012/fev.htm">Fev
      2012</a></li>
    <li><a href="calandar/2012/mar.htm">Mar
      2012</a></li>
  </ul>
</aside>
<article>
....
<article>
```

<article> : un article indépendant

- La balise <article> sert à englober une partie indépendante de la page
- Cette partie pourra reprendre toutes les balises structurantes vues précédemment



Avec cette balise, les choses se compliquent, car c'est un faux-ami et il ne faudrait surtout pas croire que c'est prévu pour recevoir un article, au sens article de journaux. En fait, sa véritable vocation est de recevoir un contenu texte « remarquable » qui, bien qu'en rapport avec le reste des contenus éditoriaux de la page, peut être autonome et se suffire à lui-même. Ce peut être un article de blog, un commentaire, un message, une annonce. Mais ce n'est pas non plus un résumé, ni un chapeau d'article ; l'élément, même remarquable, reste discret.

Bref, la balise `<article>` est idéale pour encadrer et structurer tout ce qui peut se lire seul, sans l'environnement éditorial de la page. Elle permet à la fois de structurer les contenus mais également d'optimiser les fonctionnalités de recherche tout en facilitant la navigation.

```
<section id="main">
  <article>
    <H1>
      <a href="http://www.monsite.com"> La
      nouvelle version
      Firefox aime l'HTML5</a>
    </H1>
    <p>
      Lorem ipsum dolor sit amet, consectetur
      adipiscing elit.
      Vivamus elit ante, ullamcorper nec posuere.
    </p>
  </article>
</section>
```


Exemples de Structures

En HTML5, on remarque que l'emploi de telle ou telle balise est plus que jamais fonction de son environnement, de l'interprétation sémantique des contenus déployés et de l'expérience du développeur.

Pour vous aider à y voir plus clair, voici quelques exemples de structures relevées sur des sites actuels en HTML5 qui peuvent toutes être considérées comme valides.

Premier exemple avec la comparaison de la même structure écrite pour l'HTML4 et l'HTML5

Html 4

```
<body>
```

```
<div id="header">
```

```
<div id  
="colg">
```

```
<div class="txt">
```

```
<div class="txt">
```

```
<div id="footer">
```

Html5

```
<body>
```

```
<header>
```

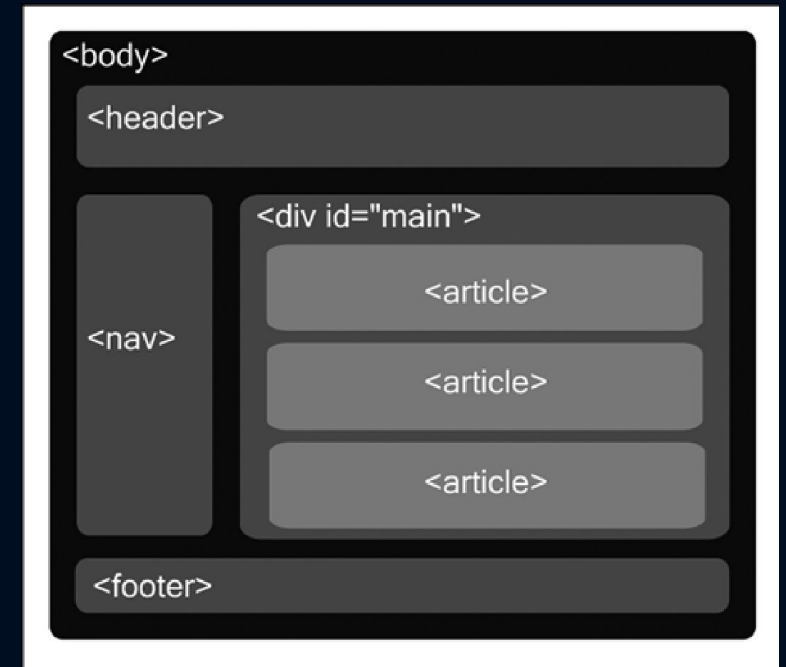
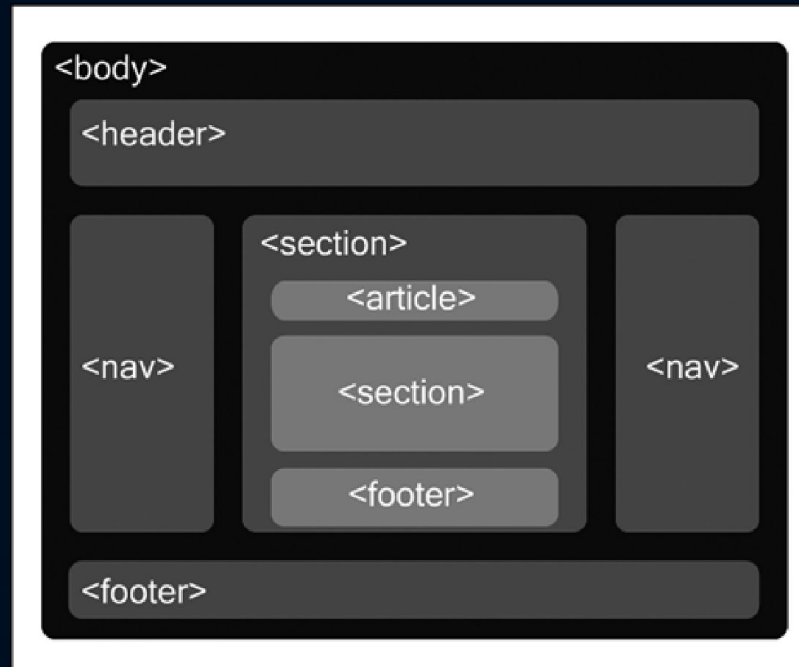
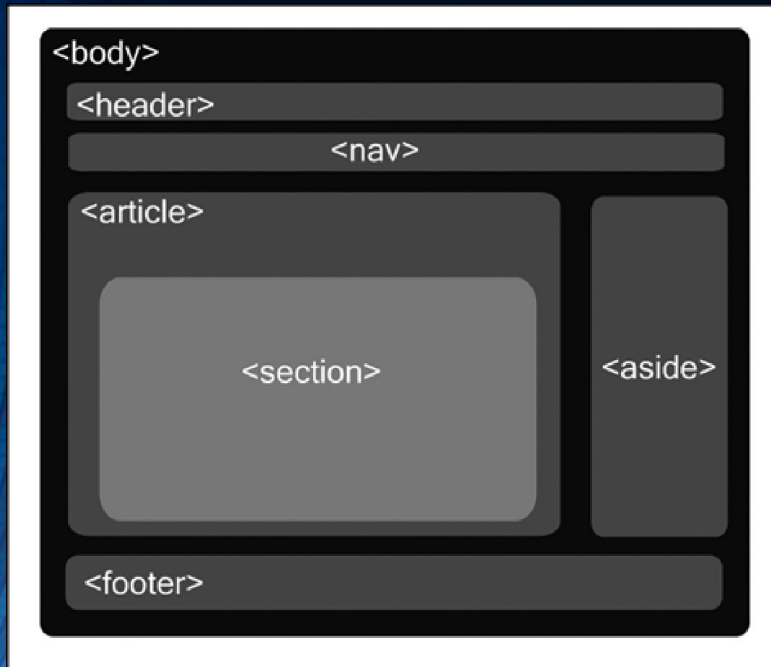
```
<nav>
```

```
<article>
```

```
<article>
```

```
<footer>
```

Autres exemples



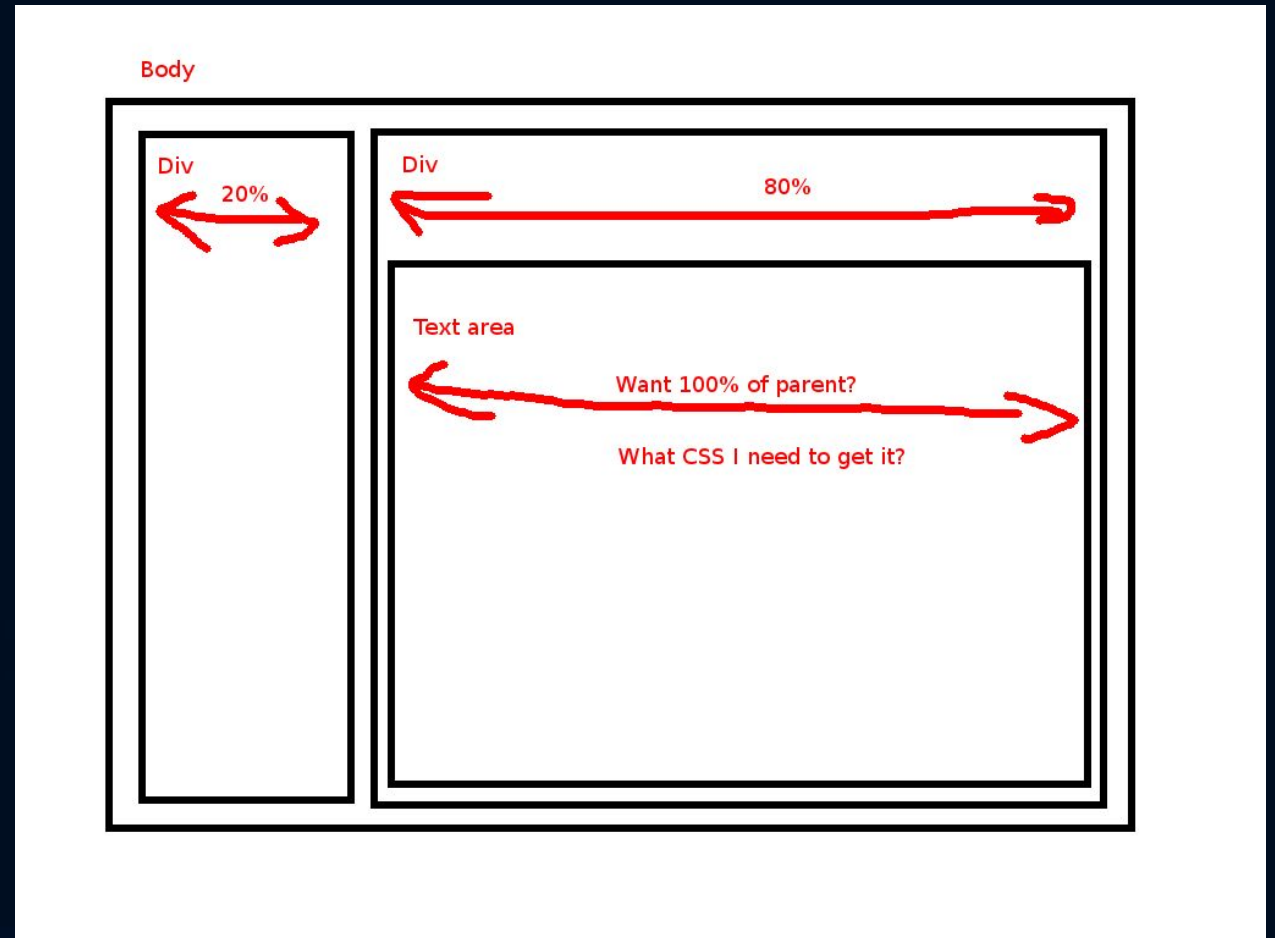
The background is a deep blue gradient. On the right side, there is a complex, three-dimensional grid pattern that curves and spirals inward, creating a tunnel-like effect. The grid lines are lighter blue and become more pronounced as they approach the center of the curve. The overall composition is modern and tech-oriented.

Jouer avec les tailles

Les tailles

- **width:** c'est la largeur du bloc. À exprimer en pixels (px) ou en pourcentage (%).
- **height:** c'est la hauteur du bloc. À exprimer en pixels (px) ou en pourcentage (%).

```
img
{
    width: 80%
}
```



Minimum et maximum

min-width: largeur minimale ;
min-height: hauteur minimale ;
max-width: largeur maximale ;
max-height: hauteur maximale;

SIDEBAR A

min-width: 300px

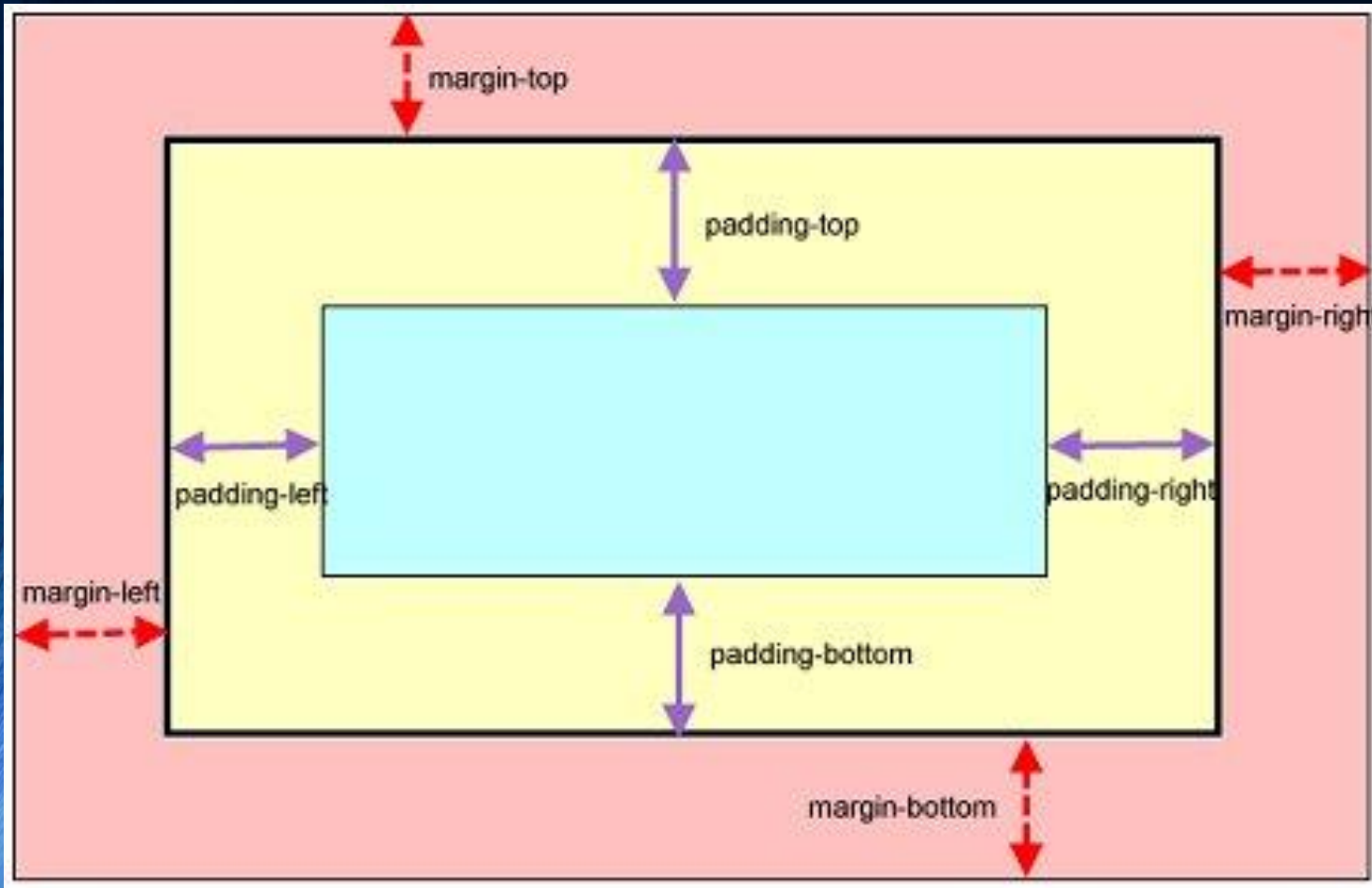
MIDDLE COL

width: fill

SIDEBAR B

min-width: 300px

Les marges



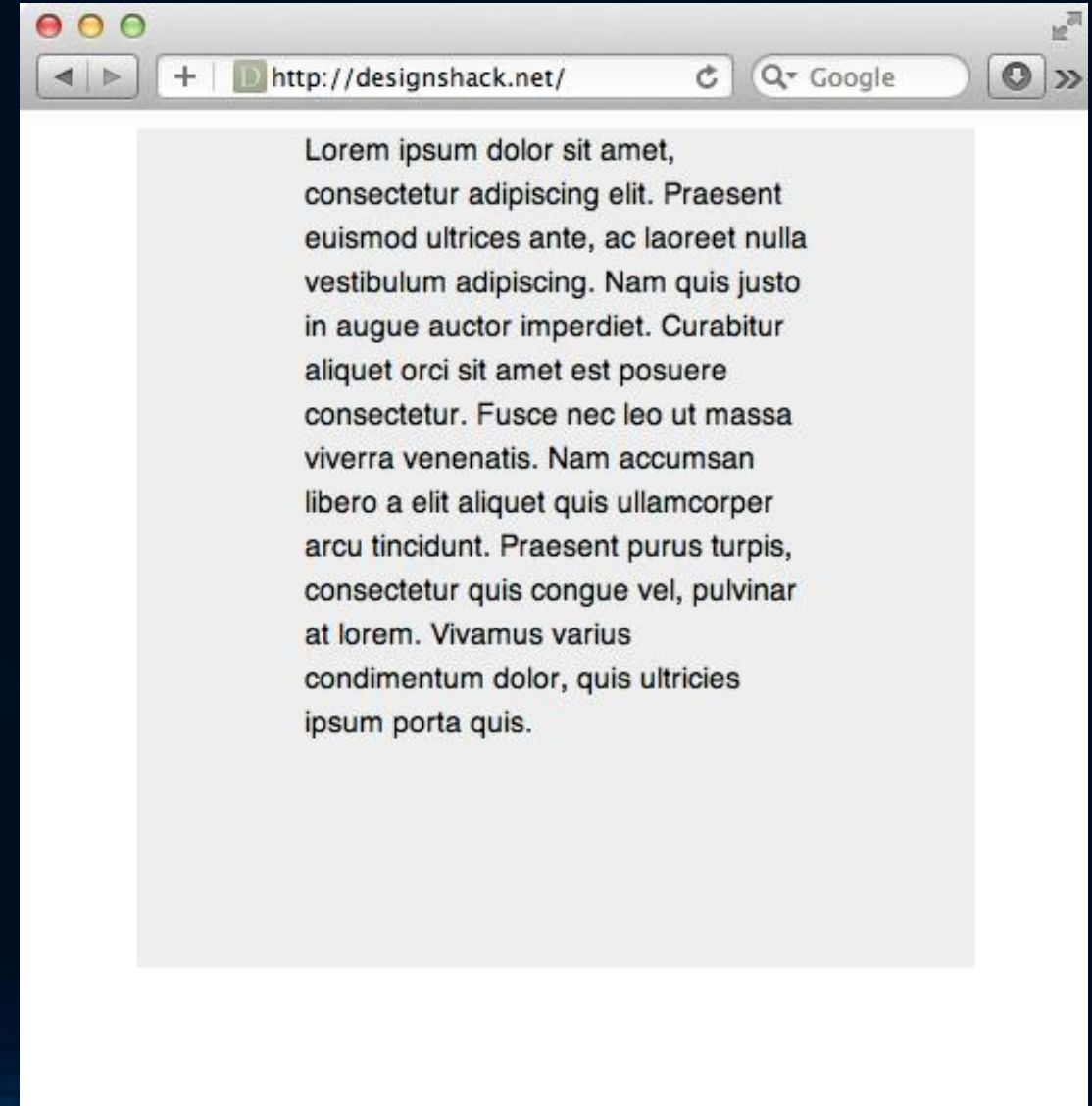
En CSS, on peut modifier la taille des marges avec les deux propriétés suivantes :

- **padding:** indique la taille de la marge (en px)
- **margin:** indique la taille de la marge extérieure (en px)

Vous pouvez également ajouter les options : top / right / bottom / left

Centrer

- Afin de pouvoir centrer un bloc, il faudra :
 - Que le bloc ait une largeur (width)
 - Mettre des marges automatiques (margin:auto;)



Overflow : rogner du contenu

- La propriété overflow définit comment rogner le contenu, comment afficher les barres de défilement ou encore comment afficher le contenu sortant du cadre dans un élément de type bloc.

```
/* Le contenu n'est pas rogné */  
overflow: visible; (par défaut)
```

```
/* Le contenu est rogné */  
overflow: hidden;
```

```
/* Le contenu est rogné et affiche des ascenseurs */  
overflow: scroll;
```

```
/* On laisse l'agent utilisateur décider */  
overflow: auto;
```

```
/* Valeurs globales */  
overflow: inherit;  
overflow: initial;  
overflow: unset;
```

Balises et fonctionnalités obsolètes

Grand nettoyage de printemps chez HTML

Cette arrivée de l'HTML5 a été l'occasion pour le W3C, instance décisionnelle internationale sur les langages HTML et CSS, de déclarer certaines balises de l'HTML 4.01 obsolètes :

- Soit parce qu'elles n'offraient qu'un service limité, et souvent du genre rendu visuel, qui peut désormais être assuré, en mieux par les CSS.
- Soit parce que les navigateurs ont de leur côté évolué (et de façon conséquente pour certains d'entre eux tels que IE qui, il faut le reconnaître, avait accumulé défauts et retards depuis sa création).
- Parce que les mentalités des concepteurs ont évolué également, tout comme l'enseignement, au niveau de l'éthique et des méthodologies à employer.
- Parce que le référencement naturel et l'importance d'obtenir un bon positionnement dans les pages de résultats des moteurs de recherche ont pris de plus en plus d'importance dans la construction d'un site ; de ce côté-là, l'HTML5 et ses nouvelles balises ont incontestablement une portée.
- Autre élément qui a depuis pris beaucoup de valeur : l'accessibilité des sites aux handicapés, laquelle n'est pas favorisée par ces anciennes balises qui auraient plutôt tendance à compliquer les situations ou à être un obstacle aux périphériques de lecture.

Balises HTML 4.01 dépréciées et obsolètes

Que signifie ce terme de dépréciation ? En fait, pour des raisons de compatibilité avec les anciens sites, le W3C ne peut se permettre de supprimer purement et simplement du « catalogue » des balises, même obsolètes. Par contre, il a décidé de :

- Ne plus les lister, ni les faire apparaître dans leurs documentations.
- Ne plus assurer de support à leur sujet.
- Ne plus les valider lors des tests par les validateurs en ligne, et même les déclarer invalides.

Liste des balises dépréciées et obsolètes

<applet>
<basefont>
<big>
<center>
<dir>

<strike>
<tt>
<u>
<acronym>
<noscript>



Fonctionnalités obsolètes

- Ces pratiques ne généreront pas d'erreur au validateur, mais des avertissements :
 - attribut `http-equiv` de l'élément `meta` : l'attribut `lang` doit être utilisé en remplacement ;
 - attribut `border` sur l'élément `img` : CSS doit être utilisé en remplacement ;
 - attribut `language` sur l'élément `script` : il doit être omis ou remplacé par l'attribut `type` avec la valeur `text/javascript` ;
 - attribut `name` sur l'élément `a` : il doit être omis ou remplacé par l'attribut `id`. Une exception cependant : s'il est présent, il ne doit pas être vide, mais posséder la même valeur que l'`id`.

Bonnes pratiques pour un site accessible

Au préalable à toute mise en oeuvre technique au niveau du code HTML, des bonnes pratiques existent pour structurer et produire du contenu pour le Web. En voici quelques-unes, ainsi que les cas dans lesquels elles se révèlent salutaires.

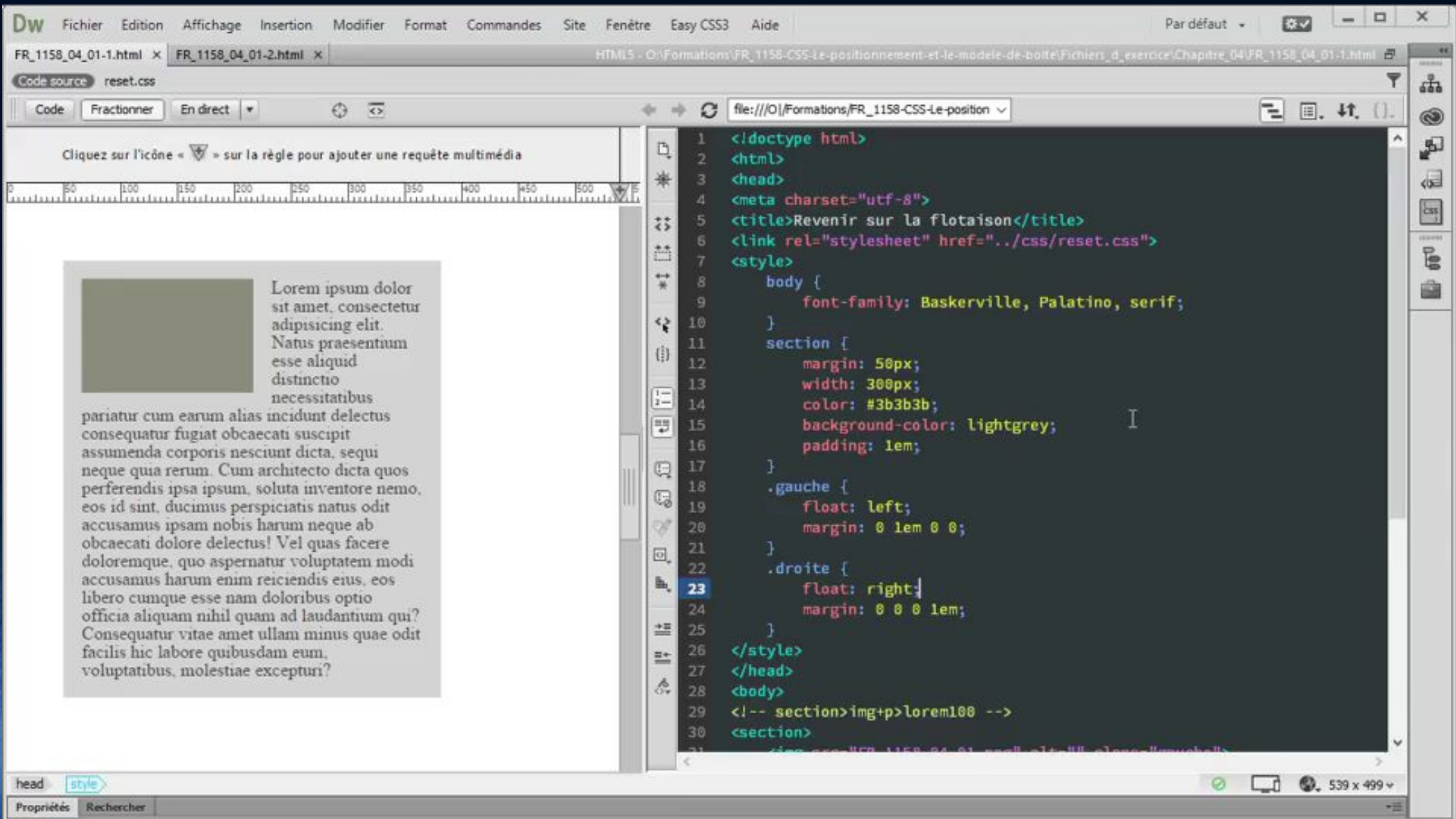
- Ne pas baser l'information uniquement sur les couleurs (daltonisme et achromatopsie), par exemple éviter « cliquez sur le bouton rouge ».
- Ne pas baser l'information uniquement sur le son (surdit ).
- Permettre l'agrandissement des polices dans le navigateur (myopie, fatigue visuelle).
- Ne pas exiger un court d lai de r action (troubles moteurs), par exemple pour la navigation et les menus d roulants.
- Favoriser le contraste du texte (d ficiences visuelles).
- Produire des alternatives telles que l'audiodescription ou la transcription pour des contenus vid o (surdit  partielle ou totale).
- Permettre la navigation au clavier (troubles moteurs) et pr voir des liens d' vitement.
- Ne pas utiliser d'images pour afficher du texte, sans leur adjoindre une alternative texte (attribut alt).
- Concevoir une navigation claire et simple (aptitudes mentales, vieillesse).
- Structurer le contenu de fa on logique avec une hi archie de titres <hX> ais ment compr hensible.
- Exploiter HTML et CSS pour s parer le fond de la forme,  viter les mises en pages en tableaux, suivre les recommandations du W3C.
- Utiliser des liens d' vitement en d but de page pour offrir l'acc s direct au contenu,   la navigation,   la recherche.

Comprendre le positionnement

CHAPITRE EN VIDEO

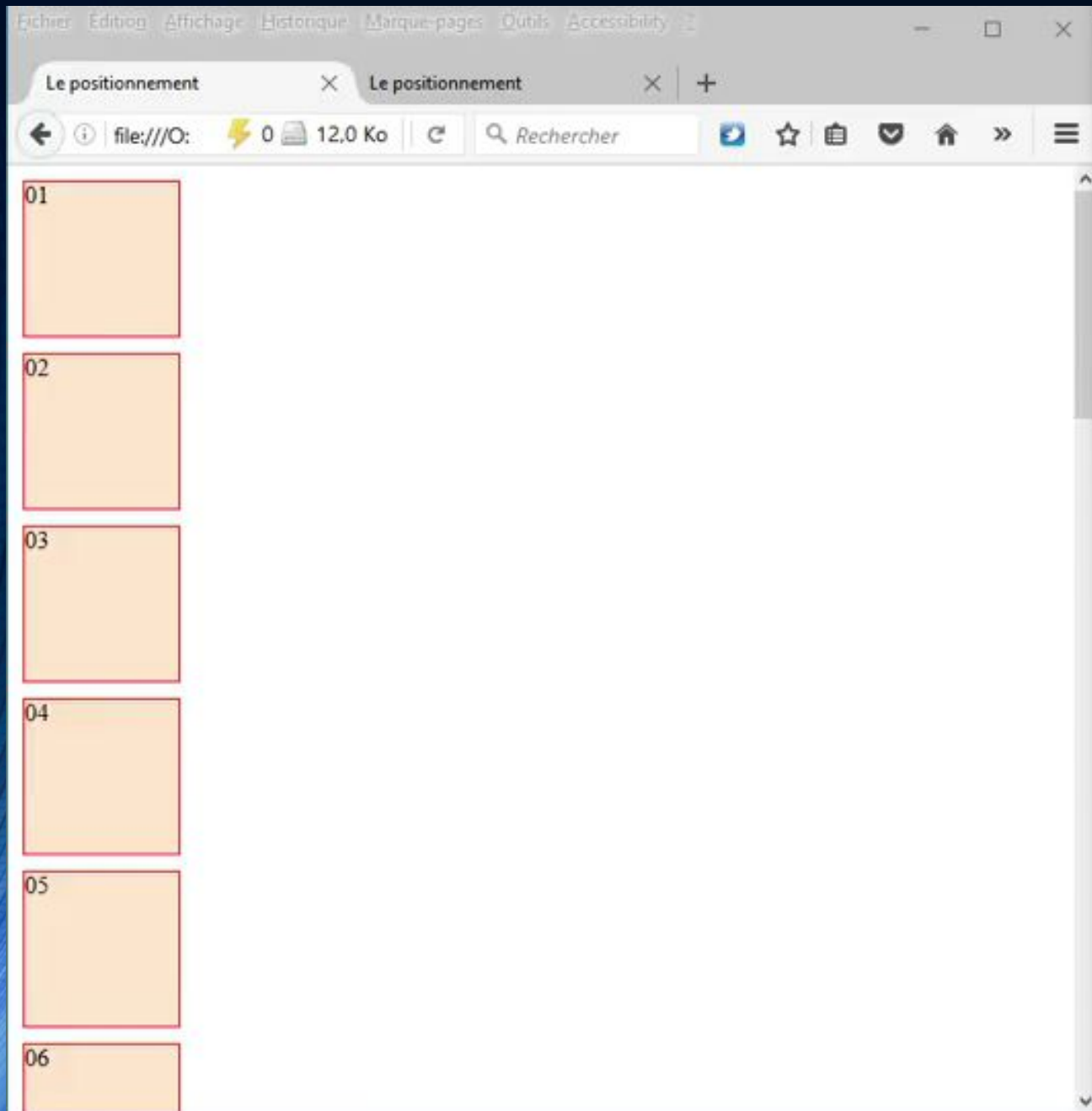
Les flottaisons

The background is a deep blue gradient. On the right side, there is a complex, three-dimensional grid pattern that curves and spirals inward, creating a tunnel-like effect. The grid lines are lighter blue and become more pronounced as they approach the center of the curve. The overall composition is modern and technological.



Les positions :

FIXED - ABSOLUTE - RELATIVE



The background is a dark blue gradient. On the right side, there is a complex, swirling pattern of thin, light blue lines that form a grid-like structure, resembling a stylized tunnel or a digital wave. The word "Flexbox" is written in a clean, white, sans-serif font on the left side of the image.

Flexbox

Début
Milieu
Fin

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="UTF-8">
5   <title>Le mode flex-box</title>
6   <style>
7     body {
8       font-family: "Avant Garde", Avantgarde, sans-serif;
9       color: grey;
10      font-size: 20px;
11      line-height: 20px;
12    }
13  </style>
14 </head>
15 <body>
16   <section class="flex">
17     <div class="debut">Début</div>
18     <div class="milieu">Milieu</div>
19     <div class="fin">Fin</div>
20   </section>
21 </body>
22 </html>
```