

La structuration logique en HTML5 et le modèle de boîte en CSS3

Table des matières

I. Contexte	3
II. Les balises sémantiques	3
III. Exercice : Appliquez la notion	7
IV. Les conteneurs génériques	8
V. Exercice : Appliquez la notion	10
VI. Le CSS des boîtes	11
VII. Exercice : Appliquez la notion	21
VIII. Auto-évaluation	22
A. Exercice final	22
B. Exercice : Défi	24
Solutions des exercices	26
Contenus annexes	36

I. Contexte

Durée : 1 h

Environnement de travail : *Visual Studio Code (cf. p.36)*

Pré-requis :

- Connaître la notion de balise
- Savoir rédiger une page HTML valide
- Connaître les bases de CSS3

Contexte

Nous allons construire un site personnel grâce aux éléments sémantiques qui permettront de donner du sens au contenu, et nous utiliserons les propriétés CSS sur les boîtes, qui nous permettront d'organiser le contenu, ainsi que de gérer l'espace.

II. Les balises sémantiques

Objectif

- Savoir utiliser les balises sémantiques afin de structurer une page

Mise en situation

Les balises sémantiques donnent d'importantes indications sur le rôle des données qu'elles contiennent. Leur utilisation permet de repérer facilement les éléments ou de les grouper en fonction de leur importance et de leur rôle sur la page.

Définition Balises sémantiques

Les balises sémantiques sont les suivantes :

- La balise `<header>` va grouper tous les éléments d'en-tête de la page, comme le logo.
- L'élément `<nav>` permet de naviguer à travers le site : il s'agit du menu de navigation.
- La balise `<main>` comporte toutes les informations importantes de la page : il s'agit du contenu principal.
- L'élément `<article>` contient une information qui est indépendante du reste de la page : par exemple, lorsque nous représentons une galerie d'articles, les articles sont indépendants les uns des autres (ils pourraient exister les uns sans les autres).
- La balise `<section>` est une balise d'organisation de contenu, mais cette fois-ci les données qu'elle contient sont dépendantes du contexte, une section isolée perd le sens de la donnée qu'elle contient.
- L'élément `<aside>` va regrouper des éléments en marge du contenu. Cela peut être intéressant pour contenir des publicités, par exemple. En effet, ces dernières n'ont pas toujours de lien avec le contenu de la page.
- La balise `<footer>` est le pied de page, il est possible de trouver les mentions légales et les liens vers les informations de contact.

Ces balises n'ont pas d'impact visuel sur la page, mais elles permettent d'améliorer le sens des données qu'elles contiennent. Elles doivent toutes être utilisées dans la balise `<body>`.

Exemple

Nous allons utiliser les balises sémantiques pour structurer notre site personnel.

Commençons par ajouter un `<header>`, contenant le nom du site :

```
1 <header>
2   <h1>Mon site</h1>
3 </header>
```

Nous ajouterons ensuite un menu de navigation comportant les liens "Ma présentation", "Mes projets" et "Me contacter". L'attribut `role="navigation"`¹ **peut être utilisé en complément** d'une balise `nav`. L'intérêt de l'ajouter est d'accroître la compatibilité avec les technologies d'assistance, notamment les moins récentes.

```
1 <nav role="navigation">
2   <ul>
3     <li><a href="#">Ma présentation</a></li>
4     <li><a href="#">Mes projets</a></li>
5     <li><a href="#">Me contacter</a></li>
6   </ul>
7 </nav>
```

Puis nous structurerons la partie principale du site, qui va contenir deux articles dans une section :

```
1 <main>
2   <section>
3     <article>
4       <h2>Mon site personnel</h2>
5       <p>
6         Lorem ipsum dolor sit amet, consectetur adipiscing elit.
7         Erabitur fringilla nisi neque, non commodo diam tincidunt quis.
8         Donec a nibh sed ipsum fermentum vehicula vel ac lectus.
9         Nulla congue feugiat tellus, ut imperdiet enim elementum a.
10        Sed sed odio eu nibh gravida ultrices.
11      </p>
12    </article>
13    <article>
14      <h2>Mon parcours</h2>
15      <p>
16        Lorem ipsum dolor sit amet, consectetur adipiscing elit.
17        Curabitur fringilla nisi neque, non commodo diam tincidunt quis.
18        Donec a nibh sed ipsum fermentum vehicula vel ac lectus.
19        Nulla congue feugiat tellus, ut imperdiet enim elementum a.
20        Sed sed odio eu nibh gravida ultrices.
21      </p>
22    </article>
23  </section>
24 </main>
```

Et nous ajouterons un `<aside>` qui nous permettra de nous présenter aux visiteurs :

```
1 <aside>
2   <article>
3     <h2>Qui suis-je ?</h2>
4     <p>
5       Lorem ipsum dolor sit amet, consectetur adipiscing elit.
6     <h3>Mes passions</h3>
7     <ul>
8       <li>Les chats</li>
9       <li>Le sport</li>
10      <li>La musique</li>
```

1 https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Roles/Navigation_Role

```

11     </ul>
12 </p>
13 </article>
14 </aside>

```

Avant d'ajouter un pied de page, dans lequel nous pourrions mettre les copyrights :

```

1 <footer>
2   <p>Tous droits réservés</p>
3 </footer>

```

Voici la page une fois structurée :

Mon site

- [Ma présentation](#)
- [Mes projets](#)
- [Me contacter](#)

Mon site personnel

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur fringilla nisi neque, non commodo diam tincidunt quis. Donec a nibh sed ipsum fermentum vehicula vel ac lectus. Nulla congue feugiat tellus, ut imperdiet enim elementum a. Sed sed odio eu nibh gravida ultrices.

Mon parcours

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur fringilla nisi neque, non commodo diam tincidunt quis. Donec a nibh sed ipsum fermentum vehicula vel ac lectus. Nulla congue feugiat tellus, ut imperdiet enim elementum a. Sed sed odio eu nibh gravida ultrices.

Qui suis-je ?

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Mes passions

- Les chats
- Le sport
- La musique

Tous droits réservés

Ajoutons un peu de style CSS dans une balise `<style>` dans le `<head>` et importons une police pour différencier plus simplement les éléments.

L'élément `<link href="https://fonts.googleapis.com/css?family=Lato:400,100,300" rel="stylesheet">` permet d'accéder au service d'une Google Fonts. Google Fonts est un service d'hébergement gratuit de polices d'écritures pour le Web, démarré en 2010. Dans notre cas, elle nous permet d'utiliser la police Lato dans notre application web.

```

1 <head>
2   <meta charset="utf-8">
3   <title>Mon site</title>
4   <link href="https://fonts.googleapis.com/css?family=Lato:400,100,300" rel="stylesheet">
5   <style>
6     body {
7       font-family: 'Lato', sans-serif;
8     }
9
10    header {
11      text-align: center;
12    }
13
14    nav {
15      background-color: lightgrey;
16    }
17
18    footer {
19      text-align: center;

```

```
20     background-color: #607D8B;
21     color: white;
22 }
23 </style>
24 </head>
```

Mon site

- [Ma présentation](#)
- [Mes projets](#)
- [Me contacter](#)

Mon site personnel

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur fringilla nisi neque, non commodo diam tincidunt quis. Donec a nibh sed ipsum fermentum vehicula vel ac lectus. Nulla congue feugiat tellus, ut imperdiet enim elementum a. Sed sed odio eu nibh gravida ultrices.

Mon parcours

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur fringilla nisi neque, non commodo diam tincidunt quis. Donec a nibh sed ipsum fermentum vehicula vel ac lectus. Nulla congue feugiat tellus, ut imperdiet enim elementum a. Sed sed odio eu nibh gravida ultrices.

Qui suis-je ?

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Mes passions

- Les chats
- Le sport
- La musique

Tous droits réservés

Remarque

Il est important de structurer correctement une page web, afin de permettre aux moteurs de recherche de mieux référencer le site, et aux personnes en situation de handicap d'utiliser leur lecteur d'écran.

Syntaxe

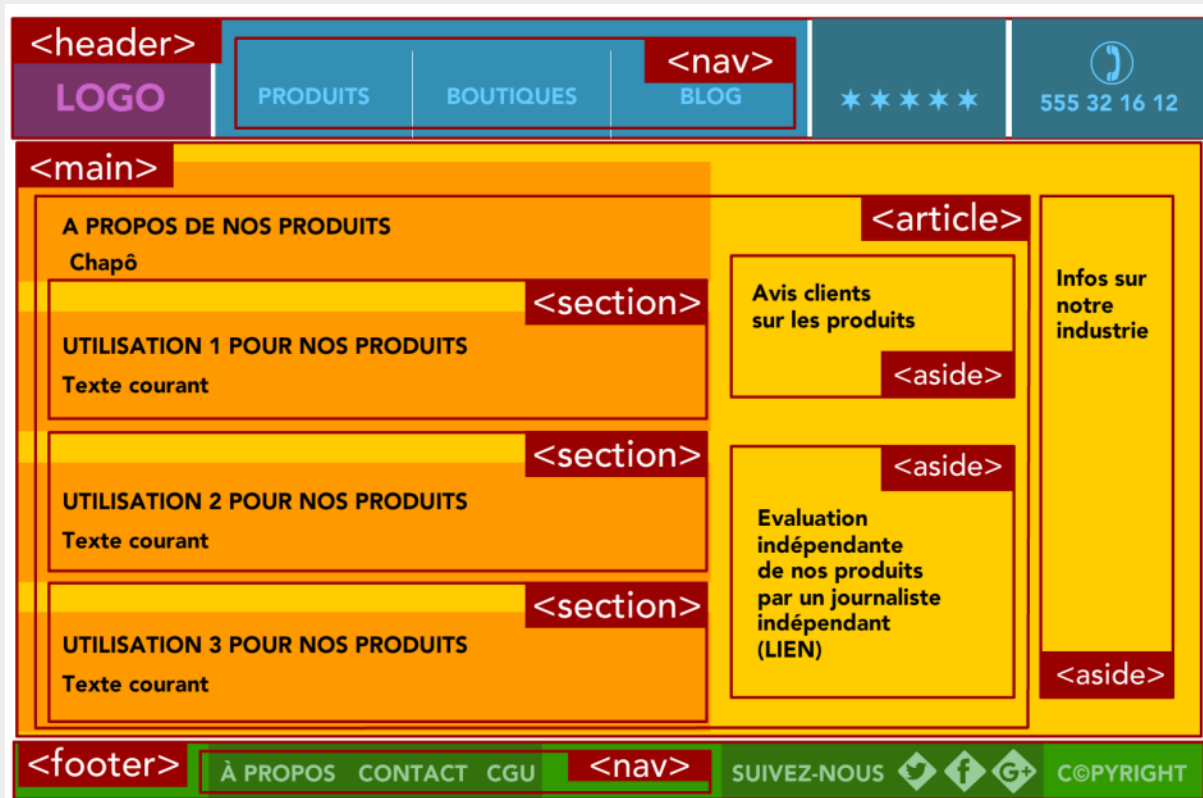
À retenir

Les balises sémantiques sont importantes et permettent de mieux structurer une page et son contenu, afin de leur donner plus de sens.

Nous avons vu qu'une page peut contenir :

- Un `header` : en-tête de la page
- Un `main` : partie principale de la page
- Des articles : contenant des informations indépendantes du contexte
- Des sections : contenant des informations dépendantes du contexte
- Un `aside` : partie secondaire d'une page
- Un `footer` : pied de la page

Schéma récapitulatif des balises sémantiques :



III. Exercice : Appliquez la notion

Question

[solution n°1 p.27]

Vous devez reprendre le code suivant d'un autre développeur pour la page d'un site de type blog. Le développeur a déjà construit la page avec des blocs, mais vous remarquez que certains éléments pourraient être échangés par d'autres plus sémantiques afin d'apporter une meilleure structure au contenu.

Remplacez les balises `<div>` par des balises sémantiques, là où cela vous semble nécessaire.

```

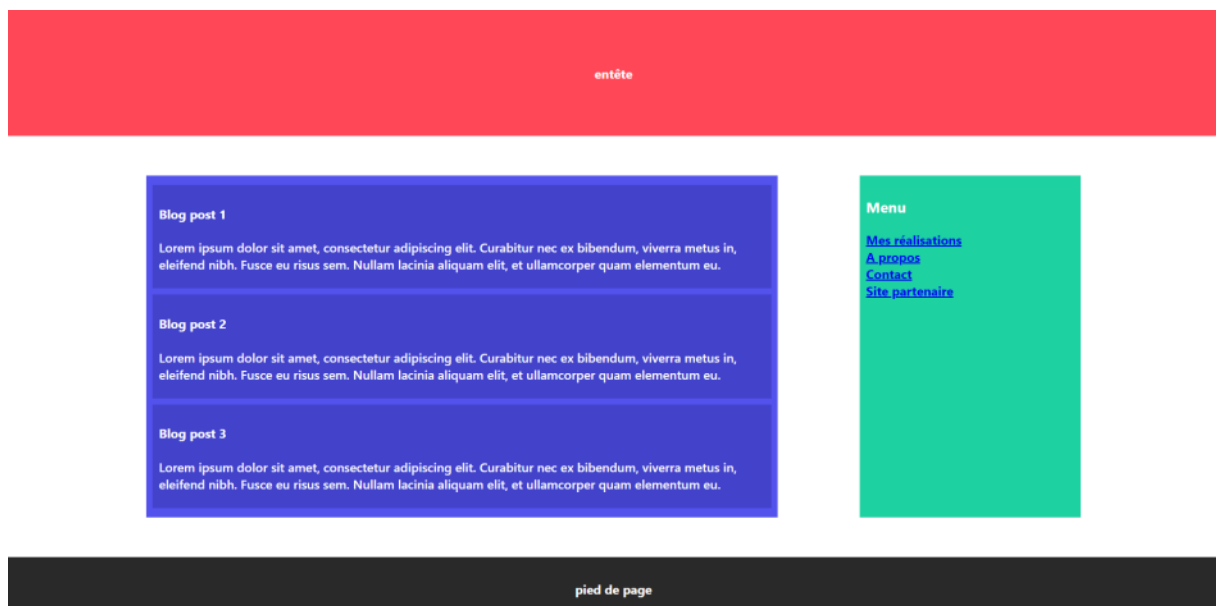
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <link rel="stylesheet" href="style.css">
7   <title>Document</title>
8 </head>
9 <body>
10  <div>entête</div>
11  <div class="content">
12    <div>
13      <div>
14        <h4>Blog post 1</h4>
15        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur nec ex bibendum,
viverra metus in, eleifend nibh. Fusce eu risus sem. Nullam lacinia aliquam elit, et
ullamcorper quam elementum eu.</p>
16      </div>
17      <div>
18        <h4>Blog post 2</h4>

```

```

19     <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur nec ex bibendum,
viverra metus in, eleifend nibh. Fusce eu risus sem. Nullam lacinia aliquam elit, et
ullamcorper quam elementum eu.</p>
20 </div>
21 <div>
22     <h4>Blog post 3</h4>
23     <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur nec ex bibendum,
viverra metus in, eleifend nibh. Fusce eu risus sem. Nullam lacinia aliquam elit, et
ullamcorper quam elementum eu.</p>
24 </div>
25 </div>
26 <div>
27     <div>
28         <h3>Menu</h3>
29         <ul>
30             <li><a href="#">Mes réalisations</a></li>
31             <li><a href="#">A propos</a></li>
32             <li><a href="#">Contact</a></li>
33             <li><a href="#">Site partenaire</a></li>
34         </ul>
35     </div>
36 </div>
37 </div>
38
39 <div>pied de page</div>
40 </body>
41 </html>

```



IV. Les conteneurs génériques

Objectifs

- Savoir différencier les conteneurs génériques des autres balises
- Savoir les utiliser pour grouper des éléments

Mise en situation

Il est possible de grouper des éléments HTML grâce à un type de balise neutre sémantiquement : les conteneurs génériques. Nous allons voir ce type de balises.

Définition Conteneurs génériques

Il existe deux conteneurs génériques : `<div>` et ``. Ces balises n'ont aucune valeur sémantique, c'est-à-dire qu'elles n'apportent aucune information sur les données qu'elles contiennent, contrairement aux balises sémantiques.

Le fait que les éléments `div` et `span` ne possèdent aucune valeur sémantique peut faire penser qu'ils vont à l'encontre même du rôle du HTML. C'est tout à fait vrai en soi, et c'est la raison pour laquelle on essaiera idéalement de n'utiliser ces éléments qu'en dernier recours et si nous n'avons aucun autre choix crédible.

Les éléments HTML `div` et `span` sont des conteneurs destinés à structurer le contenu, mais ils ont un rôle complémentaire et des règles de rendu différentes.

- `<div>` permet surtout de regrouper des éléments dans un même bloc (on dit alors que la `div` est un élément de type bloc), afin de les styliser avec du CSS plus simplement.
- ``, à la différence de `<div>`, place les éléments en ligne (balise de type `inline`), ce qui permet de l'inclure facilement dans des paragraphes ou des boutons, par exemple.

L'élément de type bloc :

- Si aucune largeur n'est définie, il prend toute la largeur de son élément parent
- Il peut avoir des marges et des paddings
- Si aucune hauteur n'est définie, il prend la hauteur de ses éléments enfants
- Il ignore la propriété `vertical-align`

L'élément de type inline :

- Il s'inscrit dans le flux du texte
- Il ne 'saute' pas de ligne comme le ferait un élément bloc
- Il ignore les marges, `width` et `height`
- Il accepte les propriétés `white-space` et `vertical-align`

D'une manière générale, vous pouvez inclure n'importe quel élément block à l'intérieur d'un autre élément block. Vous pouvez également inclure n'importe quel élément inline dans un élément block et n'importe quel élément inline dans un autre élément inline. La seule chose qu'on ne puisse pas faire, c'est inclure un élément block à l'intérieur d'un élément inline, à une exception près : l'élément `<a>` peut envelopper n'importe quel type de contenu, block comme inline ou les deux.

Exemple

Utilisons le conteneur générique `<div>` pour grouper des éléments qui n'ont aucune balise dédiée. Nous allons ajouter un groupe d'éléments vers les réseaux sociaux, et un groupe contenant leurs logos. À l'intérieur du groupe d'éléments vers les réseaux sociaux on utilisera l'élément `` pour sélectionner une partie du texte à l'intérieur de l'élément `<p>`

```
1 <footer>
2 <div>
3 <p>Suivez moi sur les <span style="background-color:yellow"> réseaux sociaux </span>!</p>
4 </div>
5 
6 
7 
8 </div>
```

```

9 </div>
10 <p>Tous droits réservés</p>
11 </footer>

```



Syntaxe À retenir

Il existe deux conteneurs génériques : `<div>` et `` qui permettent de regrouper des éléments, quand aucun conteneur sémantique ne le permet :

- `<div>` permet de regrouper des éléments dans un bloc,
- `` place les éléments en ligne.

V. Exercice : Appliquez la notion

Question

[solution n°2 p.27]

La structure suivante correspond à une page de blog. On vous demande de créer un bloc regroupant le titre et le paragraphe 1 de chaque article. Chaque bloc permettra de centrer le titre et le paragraphe 1. Chaque titre aura son numéro affiché en rouge. Pour cela, vous utiliserez les conteneurs génériques afin de ne pas casser la sémantique de la page.

```

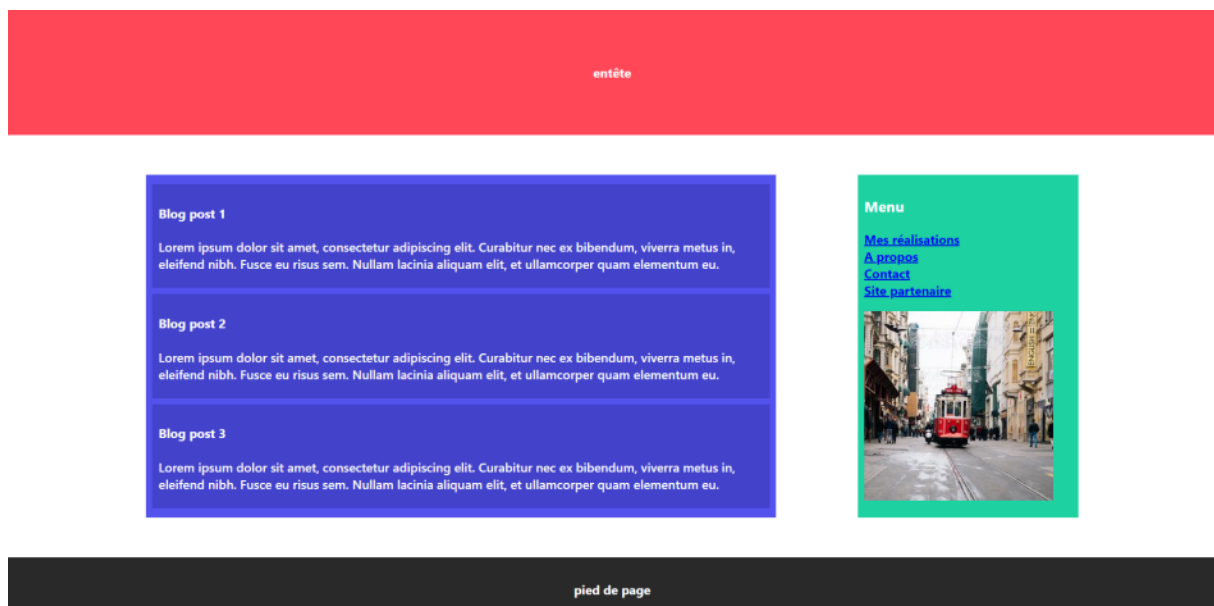
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <link rel="stylesheet" href="style.css">
7   <title>Blog</title>
8 </head>
9 <body>
10  <main>
11    <article>
12      <h4>Titre 1</h4>
13      <p>Paragraphe 1</p>
14      <p>Paragraphe 2</p>
15      <p>Paragraphe 3</p>
16    </article>
17    <article>
18      <h4>Titre 2</h4>
19      <p>Paragraphe 1</p>
20      <p>Paragraphe 2</p>
21      <p>Paragraphe 3</p>
22    </article>
23    <article>
24      <h4>Titre 3</h4>
25      <p>Paragraphe 1</p>
26      <p>Paragraphe 2</p>
27      <p>Paragraphe 3</p>
28    </article>
29  </main>
30 </body>

```

```

31 </html>
32
33 </code>
34
35 Fichier style.css
36
37 <code>
38   div {
39       text-align:center;
40   }
41   span {
42       color: red;
43   }

```



Indice :

Rendu de l'objectif à atteindre :

	Titre 1
Paragraphe 2	Paragraphe 1
Paragraphe 3	
	Titre 2
Paragraphe 2	Paragraphe 1
Paragraphe 3	
	Titre 3
Paragraphe 2	Paragraphe 1
Paragraphe 3	

VI. Le CSS des boîtes

Objectifs

- Modifier l'affichage des éléments les uns par rapport aux autres
- Gérer l'espace des éléments HTML
- Gérer la taille des éléments

Mise en situation

Nous allons maintenant voir comment gérer l'affichage des éléments HTML entre eux, et nous verrons qu'il est possible de modifier leur propre espace au moyen de propriétés CSS.

Fondamental La propriété display

La propriété CSS `display` permet de gérer l'affichage d'un élément par rapport aux autres dans la page.

Celle-ci dispose de trois valeurs, qu'il est possible d'attribuer à cette propriété pour modifier l'affichage des éléments.

Définition La valeur inline

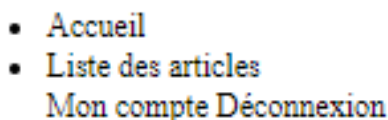
Avec la valeur `display: inline`, les éléments vont être placés côte à côte sur la même ligne. L'élément ne prendra que la largeur nécessaire à l'affichage de son contenu, jusqu'à ce qu'il n'y ait plus d'espace en longueur. Les éléments suivants seront alors mis à la ligne.

Les propriétés `width` et `height` n'ont aucun effet sur un élément `inline`.

```
1 .inline {
2     display: inline;
3 }
```

Prenons l'exemple d'une liste à puces dont on souhaiterait afficher certains éléments sur une même ligne. Voici ce que cela pourrait donner :

```
1 <ul>
2     <li>Accueil</li>
3     <li>Liste des articles</li>
4     <li class="inline">Mon compte</li>
5     <li class="inline">Déconnexion</li>
6 </ul>
```



Définition La valeur block

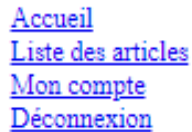
Avec la valeur `display: block`, les éléments vont s'empiler les uns sur les autres. Un élément occupera donc une ligne (il prendra toute la largeur, peu importe la taille de son contenu), il ne sera jamais côte à côte avec un autre élément. Prenons l'exemple de ces liens, par défaut ceux-ci seraient affichés côte à côte.

```
1 <nav>
2     <a href="#">Accueil</a>
3     <a href="#">Liste des articles</a>
4     <a href="#">Mon compte</a>
5     <a href="#">Déconnexion</a>
6 </nav>
```



En revanche, si l'on y applique le CSS suivant, voici comment ils seront affichés :

```
1 nav > a {  
2   display: block;  
3 }
```



[Accueil](#)
[Liste des articles](#)
[Mon compte](#)
[Déconnexion](#)

Définition La valeur inline-block

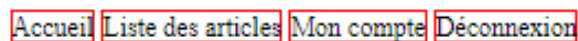
La valeur `display: inline-block`, quant à elle, partage certaines caractéristiques des propriétés précédentes. En effet, elle est affectée par les propriétés permettant la modification de ses dimensions, et les éléments seront placés côte à côte jusqu'à remplir tout l'espace sur la ligne.

À la différence de `inline`, les propriétés `width` et `height` sont actives : il est possible de modifier la taille de l'élément, tout en le gardant dans le flot de la ligne en cours.

Reprenons l'exemple de notre liste. Au-delà d'aligner tous les éléments sur une même ligne, nous pourrions vouloir leur donner un peu d'espace avec le code ci-dessous :

```
1 <nav>  
2   <ul>  
3     <li>Accueil</li>  
4     <li>Liste des articles</li>  
5     <li>Mon compte</li>  
6     <li>Déconnexion</li>  
7   </ul>  
8 </nav>  
  
1   nav li {  
2     display: inline;  
3     width: 150px;  
4     height: 50px;  
5     line-height: 50px;  
6     text-align: center;  
7     border: 1px solid red;  
8   }
```

Pourtant, l'effet escompté n'est pas là.



Accueil Liste des articles Mon compte Déconnexion

Si l'on applique la valeur `inline-block` à la propriété `display`, l'affichage correspond désormais à nos attentes.



Accueil

Liste des articles

Mon compte

Déconnexion

Définition La hauteur et la largeur

Les éléments HTML possèdent une hauteur et une largeur, qu'il est possible de modifier grâce aux propriétés `height` et `width` en leur donnant n'importe quelle unité reconnue en CSS¹.

Il existe différentes familles d'unités, les unités absolues(px,in,cm,mm,pc,pt), les unités relatives au viewport (vh,vw,vmin,vmax), les unités en pourcentages (%) et les unités relatives au texte(Em,Rem,Ex,Ch).

Pour définir une <div> de 150 pixels de large et de 100 pixels de hauteur il suffira de déclarer le css suivant :

```
1 <code>
2 div {
3 width: 150px;
4 height: 100px;
5 }
6 </code>
```

La hauteur et la largeur des éléments ne peuvent pas être modifiées si l'élément est en `display: inline`.

Certains éléments, tels que les liens, les images et les balises permettant la mise en forme sémantique du texte (gras, emphase...), sont par défaut affichés en ligne, contrairement aux éléments de type bloc comme les titres, les paragraphes ou encore les listes.

Il est possible de définir la hauteur et la largeur minimales d'un élément grâce aux propriétés `min-width` et `min-height`. Si la taille du bloc diminue, elle ne pourra pas être inférieure aux valeurs des propriétés `min-width` et `min-height`.

De même qu'il est possible de définir la hauteur et la largeur maximales d'un élément avec les propriétés `max-width` et `max-height`.

Exemple Aligner des éléments

Si nous voulons aligner les éléments <main> et <aside>, il faut leur donner la propriété `display: inline-block` pour qu'ils puissent être placés côte à côte.

Comme les deux éléments vont prendre tout l'espace disponible, il est nécessaire de fixer une largeur qu'ils ne pourront pas dépasser.

Ici, le <aside> prendra un quart de la place disponible sur la ligne :

```
1 main {
2   display: inline-block;
3   width: 75%;
4 }
5
6 aside {
7   display: inline-block;
8   width: 25%;
9 }
```

1 https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Values_and_Units

Mon site

[Ma présentation](#) [Mes projets](#) [Me contacter](#)

Mon site personnel

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur fringilla nisi neque, non commodo diam tincidunt quis. Donec a nibh sed ipsum fermentum vehicula vel ac lectus. Nulla congue feugiat tellus, ut imperdiet enim elementum a. Sed sed odio eu nibh gravida ultrices.

Mon parcours

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur fringilla nisi neque, non commodo diam tincidunt quis. Donec a nibh sed ipsum fermentum vehicula vel ac lectus. Nulla congue feugiat tellus, ut imperdiet enim elementum a. Sed sed odio eu nibh gravida ultrices.

Qui suis-je ?

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Mes passions

- Les chats
- Le sport
- La musique

Suivez moi sur les réseaux sociaux !



Tous droits réservés

Remarque

Renvoi à la ligne

La propriété `word-wrap` permet de gérer la façon dont une ligne sera renvoyée à la ligne, par exemple si son contenant est trop petit.

Il est possible de modifier la césure pour que le mot ne soit pas coupé en plein milieu, ou bien de les couper avec un tiret.

Définition

Gérer l'espace

Lorsque vous créez du code HTML, vous utilisez des balises que vous allez imbriquer les unes dans les autres. Le code suivant se compose donc de trois commandes HTML. Une balise `<div>` contenant une balise `<h1>` et une balise `<p>`.

```
1 <code>
2 <div>
3   <h1>Titre</h1>
4   <p>Exemple de texte</p>
5 </div>
6 </code>
```

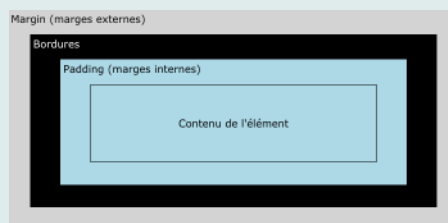
En CSS, chaque élément est une boîte, nous avons ainsi une boîte `<div>` contenant deux boîtes `<h1>` et `<p>` que nous pouvons représenter sous la forme suivante :



L'idée centrale du modèle des boîtes est que tout élément HTML (donc chaque boîte CSS) peut être représenté par un empilement de 4 boîtes :

- La première, centrale, va être composée du contenu de l'élément en soi,
- La deuxième, des marges internes de l'élément,
- La troisième, des bordures de l'élément,
- La quatrième, des marges externes de l'élément.

Voici la représentation d'un élément HTML selon le modèle des boîtes :



Les propriétés CSS liées aux différentes boîtes

Le CSS va déjà nous fournir différentes propriétés qui vont nous permettre de spécifier la taille des différents éléments composant les différentes boîtes :

- Les propriétés `width` et `height` vont nous permettre de définir la largeur et la hauteur de la boîte « contenu » ou « content »,
- La propriété `padding`, de définir la taille des marges internes,
- La propriété `border`, de définir des bordures pour notre élément,
- La propriété `margin`, de définir la taille des marges externes.

Remarque Utiliser la propriété `padding` (marges internes)



La propriété `padding` est une propriété raccourcie qui permet de définir les différents écarts de remplissage sur les quatre côtés d'un élément. Elle synthétise `padding-top`¹, `padding-right`², `padding-bottom`³, `padding-left`⁴.

On peut représenter cette définition par :

```
1 { padding : padding-top padding-right padding-bottom padding-left }
```

`padding` peut être suivi de différentes valeurs(1 à 4) de différents types :

- longueur - spécifie un remplissage en px, pt, cm, etc.
- % - spécifie un remplissage en % de la largeur de l'élément contenant
- inherit - spécifie que le remplissage doit être hérité de l'élément parent

Si le `padding` a une seule valeur, `div { padding:25px; }` cela correspond à la valeur du `padding` des quatre côtés.

```
1 <code>
2 div {
3 padding-top:25px;
4 padding-right:25px;
5 padding-bottom:25px;
6 padding-left:25px;
7 }
8 <code>
```

Si le `padding` a deux valeurs, `div { padding:25px 10%; }`, la première valeur correspond à `padding-top` et `padding-bottom`, la seconde à `padding-right` et `padding-left`.

1 <https://developer.mozilla.org/fr/docs/Web/CSS/padding-top>

2 <https://developer.mozilla.org/fr/docs/Web/CSS/padding-right>

3 <https://developer.mozilla.org/fr/docs/Web/CSS/padding-bottom>

4 <https://developer.mozilla.org/fr/docs/Web/CSS/padding-left>


```

1 <code>
2 div {
3 padding-top:25px;
4 padding-right:10%;
5 padding-bottom:25px;
6 padding-left:10%;
7 }
8 <code>

```

Si le padding à trois valeurs `div { padding:25px 15px 10px; }` cela correspond à :

```

1 <code>
2 div {
3 padding-top:25px;
4 padding-left:15px;
5 padding-right:15px;
6 padding-bottom:10px;
7 }
8 <code>

```

Si le padding à quatre valeurs `div { padding:25px 15px 10px 5px; }` cela correspond à :

```

1 <code>
2 div {
3 padding-top:25px;
4 padding-right:15px;
5 padding-bottom:10px;
6 padding-left:5px;
7 }
8 <code>

```

Chacune des propriétés `padding-top`, `padding-right`, `padding-bottom`, `padding-left` peuvent être utilisées indépendamment.

Remarque Utiliser la propriété border (bordure)



Cette élément div à une bordure verte de 25px.

La propriété CSS `border` est une propriété raccourcie qui permet de définir les propriétés liées à la bordure. `border` peut être utilisée pour définir les valeurs de `border-width`¹ (la taille), `border-style`² (le style de bordure) et `border-color`³ (la couleur).

On peut représenter cette définition par:

```

1 { border : border-width border-style border-color }
2
3 <code>
4 div {
5 border : 2px solid blue;
6 }
7 </code>

```

Le code suivant représente une balise `<div>` entourée d'une bordure : de 2px de large , de style `solid`⁴ (trait continu) et de couleur bleu.

1 <https://developer.mozilla.org/fr/docs/Web/CSS/border-width>

2 <https://developer.mozilla.org/fr/docs/Web/CSS/border-style>

3 <https://developer.mozilla.org/fr/docs/Web/CSS/border-color>

4 https://www.w3schools.com/css/css_border.asp

Remarque Utiliser la propriété margin (marges externes)

Cette élément div à la bordure verte à un margin de 70px.

Cette élément div à la bordure verte à un margin de 70px.

Les marges sont utilisées pour créer un espace autour des éléments, en dehors de toute bordure définie. Elles s'utilisent comme la propriété padding. C'est une propriété raccourcie qui permet de manipuler les autres propriétés de marges : `margin-top`¹, `margin-right`², `margin-bottom`³ et `margin-left`⁴.

On peut représenter cette définition par:

```
1 margin : margin-top margin-right margin-bottom et margin-left
```

Si le margin a une seule valeur, `div { margin:25px; }` cela correspond à la valeur du margin des quatres côtés.

```
1 <code>
2 div {
3 margin-top:25px;
4 margin-right:25px;
5 margin-bottom:25px;
6 margin-left:25px;
7 }
8 <code>
```

Chacune des propriétés `margin-top`, `margin-right`, `margin-bottom`, `margin-left` peuvent être utilisées indépendamment.

Remarque Dépasser d'une boîte

La propriété `overflow` permet de gérer la façon dont le contenu dépasse d'un élément :

- `overflow: hidden` permet de le cacher, le contenu sera alors tronqué
- `overflow: scroll` fait apparaître les barres de défilement, qui permettront de scroller dans l'élément pour afficher le contenu

Par défaut, la propriété est `overflow: visible`, ce qui implique que le contenu (s'il dépasse les limites de son conteneur) ne sera pas rogné.

Remarque Centrer horizontalement un élément de type bloc

Utiliser la propriété de largeur `width`, associée à la propriété et la valeur `margin: auto`, permet de centrer horizontalement le contenu d'un élément.

Remarque Centrer horizontalement un élément de type inline

Les éléments de type `inline` n'occupent que l'espace nécessaire à l'affichage de leur contenu dans leur élément parent.

On va pouvoir centrer ce type d'éléments en appliquant un `text-align: center` à l'élément conteneur (élément parent) de type `block` contenant l'élément `inline` en question. Cela aura pour effet de centrer tous les éléments `inline` dans cet élément `block`.

1 <https://developer.mozilla.org/fr/docs/Web/CSS/margin-top>

2 <https://developer.mozilla.org/fr/docs/Web/CSS/margin-right>

3 <https://developer.mozilla.org/fr/docs/Web/CSS/margin-bottom>

4 <https://developer.mozilla.org/fr/docs/Web/CSS/margin-left>

Exemple Aérer du contenu

Utilisons les propriétés CSS de gestion de l'espace pour aérer le contenu de la page. Pour cela, ajoutons des marges autour des éléments du menu de navigation, puis entre le menu et le body, et enfin entre le footer et le body :

```

1 nav {
2   background-color: lightgrey;
3   margin-bottom: 10px;
4 }
5
6 nav li {
7   display: inline;
8   margin: 10px;
9 }
10
11 footer {
12   text-align: center;
13   background-color: #607D8B;
14   color: white;
15   margin-top: 20px;
16 }
```



Il est possible d'ajouter une bordure à droite de la partie principale de la page afin de créer une séparation verticale avec la partie `<aside>` :

```

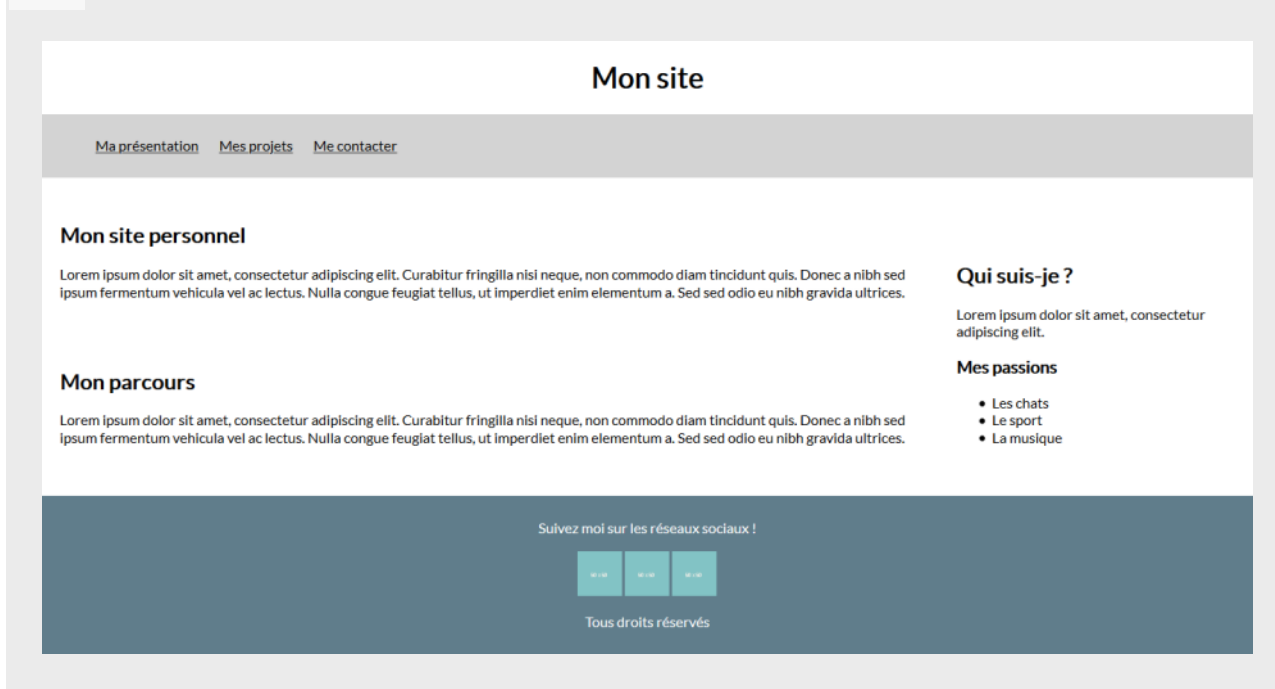
1 main {
2   display: inline-block;
3   width: 74%;
4   border-right: solid 1px lightgrey;
5 }
```

Attention, il faut penser à diminuer la place que peut prendre l'élément `<main>`, car nous venons d'ajouter une bordure de 1 pixel de large.



Maintenant, nous pouvons ajouter des marges internes pour que le contenu soit plus espacé du bord de son élément. Par exemple, dans le menu de navigation, dans les articles et le footer :

```
1 nav {
2     padding: 10px;
3     background-color: lightgrey;
4     margin-bottom: 10px;
5 }
6
7 article {
8     padding: 20px;
9 }
10
11 footer {
12     text-align: center;
13     background-color: #607D8B;
14     color: white;
15     margin-top: 20px;
16     padding: 10px;
17 }
```

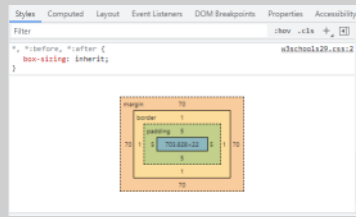


Syntaxe À retenir

- Il existe des propriétés CSS dédiées à la gestion de l'espace autour des éléments, comme `margin` et `padding` spécialisées dans la gestion des marges, ou encore `height` et `width` pour gérer les dimensions d'un élément.
- Nous avons également vu la propriété `display` qui gère le placement des éléments les uns par rapport aux autres.

Conseil Observer le modèle de boîte

Pour observer facilement les différentes boîtes composant vos éléments HTML, il vous suffit de faire un clic droit "inspecter" pour ouvrir l'inspecteur de code natif du navigateur, dans l'onglet style se trouve une représentation de l'élément que vous venez d'inspecter et des différentes couches qui le composent.



VII. Exercice : Appliquez la notion

On dispose du code suivant :

```

1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="utf-8"/>
5 </head>
6 <body>
7 <nav>
8   <ul>
9     <li>Premier élément de navigation</li>
10    <li>Second élément de navigation</li>
11    <li>Troisième élément de navigation</li>
12  </ul>
13 </nav>
14
15 </body>
16 <style>
17   nav li {
18     border: 1px solid red;
19   }
20 </style>
21 </html>

```

Question 1

[solution n°3 p.29]

Faites en sorte que les éléments de cette liste soient affichés en ligne.

Ceux-ci se verront appliquer une marge interne de 10 px et une marge externe de 25 px.

On dispose du code suivant :

```

1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="utf-8"/>
5 </head>
6 <body>
7
8 <article>
9   Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod fugiat nulla
10  pariat.
11 </article>
12 <nav>
13   <h2>Menu de navigation vertical</h2>
14   <ul>
15     <li>Premier élément de navigation</li>
16     <li>Second élément de navigation</li>

```

```

16         <li>Troisième élément de navigation</li>
17     </ul>
18 </nav>
19 </body>
20 <style>
21     nav {
22         background-color: gray;
23         vertical-align: top;
24     }
25
26     article {
27         vertical-align: top;
28         background-color: #9fcdff;
29     }
30 </style>
31 </html>
32

```

Question 2

[solution n°4 p.29]

- Les éléments `nav` et `article` doivent être alignés sur une même ligne et avoir une hauteur de 500 px.
- L'élément `nav` doit avoir une largeur de 15 %.
- L'élément `article` doit avoir une largeur de 60 %.

VIII. Auto-évaluation

A. Exercice final

Exercice 1

[solution n°5 p.30]

Exercice

On dit de la balise `<div>` qu'il s'agit d'une balise de type...

- ☐ bloc
- ☐ inline

Exercice

Afin de respecter la sémantique, dans quelle balise vaut-il mieux regrouper des éléments d'en-tête, tels qu'un logo ou le nom de la société ?

- ☐ `<div>`
- ☐ `<nav>`
- ☐ `<header>`

Exercice

Contrairement aux balises `<div>` ou ``, le fait d'utiliser des balises telles que `<nav>` ou `<footer>` impacte le résultat visuel de la page.

- ☐ Vrai
- ☐ Faux

Exercice

Si l'on souhaite présenter des informations indépendantes les unes des autres, telles qu'un listing de différents produits sur un récapitulatif d'achat, on préférera l'utilisation de la balise...

- ☐ <section>
- ☐ <aside>
- ☐ <article>

Exercice

Pour forcer l'affichage de plusieurs éléments sur la même ligne, dans la limite de leur largeur, on utilisera...

- ☐ display: inline;
- ☐ display: block;

Exercice

Dans quel cas les propriétés `width` et `height` d'un élément sont-elles prises en compte ?

- ☐ display: inline
- ☐ display: inline-block

Exercice

Si l'on souhaite définir la largeur maximale d'un élément, on utilisera la propriété...

- ☐ max-width
- ☐ max-height
- ☐ width
- ☐ height

Exercice

Par défaut, un paragraphe ou une liste sont affichés...

- ☐ En ligne
- ☐ En bloc

Exercice

Laquelle de ces propriétés représente la marge interne d'un élément ?

- ☐ margin
- ☐ padding

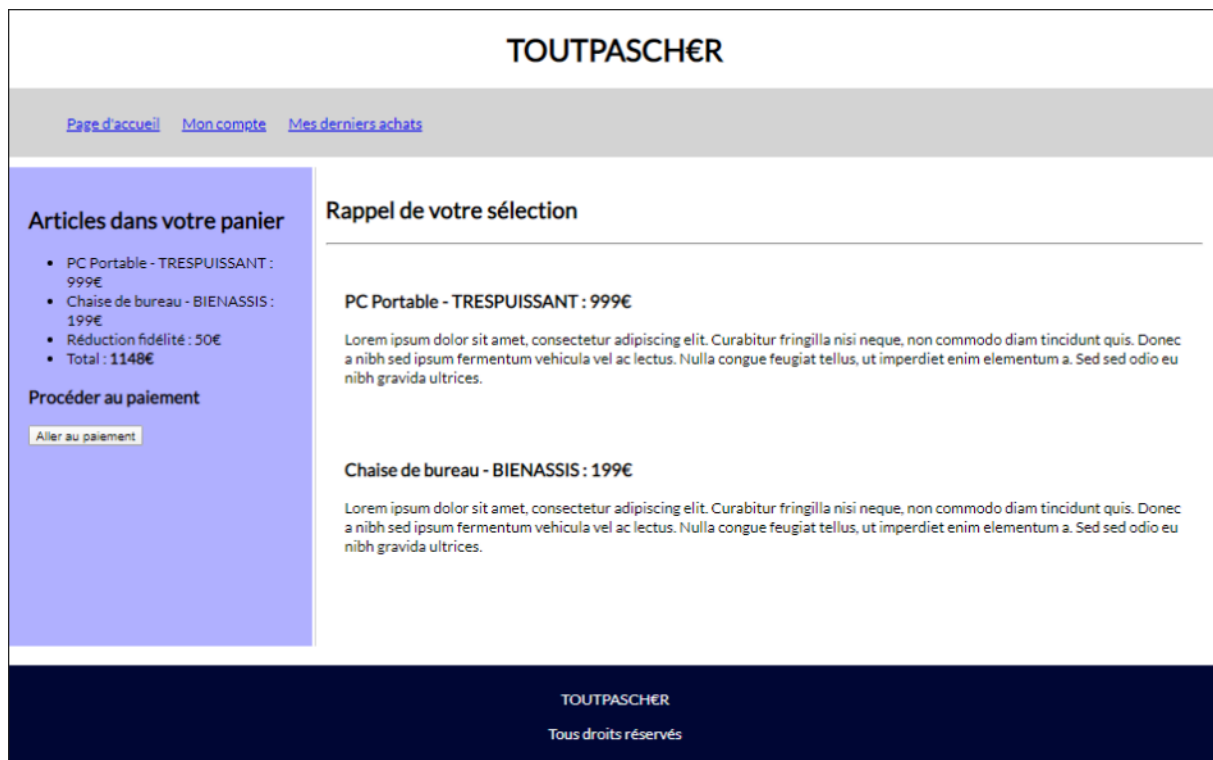
Exercice

La propriété `overflow` permet de gérer...

- ☐ L'espacement sur la page
- ☐ Les renvois à la ligne
- ☐ La gestion du dépassement d'une boîte

B. Exercice : Défi

Pour le compte d'un site de vente en ligne, vous êtes chargé de mettre en page le récapitulatif des achats. Voici un aperçu de ce qui est attendu :



Vous disposez du code suivant :

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4     <meta charset="utf-8">
5     <title>Finaliser mon achat</title>
6     <link href="https://fonts.googleapis.com/css?family=Lato:400,100,300" rel="stylesheet">
7     <style>
8         body {
9             font-family: 'Lato', sans-serif;
10            margin: 0;
11        }
12
13        .header {
14            text-align: center;
15        }
16
17        .nav {
18            background-color: lightgrey;
19        }
20
21        .nav li {
22        }
23
24        .main {
25            border-left: solid 1px lightgrey;
26        }
27
```



```

28     .article {
29     }
30
31     .aside {
32         background-color: #b0b0ff;
33         vertical-align: top;
34     }
35
36     .section {
37     }
38
39     .footer {
40         text-align: center;
41         background-color: #000634;
42         color: white;
43         margin-top: 20px;
44     }
45 </style>
46 </head>
47 <body>
48     <div class="header">
49         <h1>TOUTPASCHER</h1>
50     </div>
51     <div class="nav" role="navigation">
52         <ul>
53             <li><a href="#">Page d'accueil</a></li>
54             <li><a href="#">Mon compte</a></li>
55             <li><a href="#">Mes derniers achats</a></li>
56         </ul>
57     </div>
58     <div class="aside">
59         <div class="article">
60             <h2>Articles dans votre panier</h2>
61             <ul>
62                 <li>PC Portable - TRESPUISSANT : 999€</li>
63                 <li>Chaise de bureau - BIENASSIS : 199€</li>
64                 <li>Réduction fidélité : 50€</li>
65                 <li>Total : <strong>1148€</strong> </li>
66             </ul>
67             <h3>Procéder au paiement</h3>
68             <button>Aller au paiement</button>
69         </div>
70     </div>
71     <div class="main">
72
73         <div class="section">
74             <h2>Rappel de votre sélection</h2>
75             <hr>
76             <div class="article">
77                 <h3>PC Portable - TRESPUISSANT : 999€</h3>
78                 <p>
79                     Lorem ipsum dolor sit amet, consectetur adipiscing elit.
80                     Curabitur fringilla nisi neque, non commodo diam tincidunt quis.
81                     Donec a nibh sed ipsum fermentum vehicula vel ac lectus.
82                     Nulla congue feugiat tellus, ut imperdiet enim elementum a.
83                     Sed sed odio eu nibh gravida ultrices.
84                 </p>
85             </div>

```

```

86         <div class="article">
87             <h3>Chaise de bureau - BIENASSIS : 199€</h3>
88             <p>
89                 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
90                 Curabitur fringilla nisi neque, non commodo diam tincidunt quis.
91                 Donec a nibh sed ipsum fermentum vehicula vel ac lectus.
92                 Nulla congue feugiat tellus, ut imperdiet enim elementum a.
93                 Sed sed odio eu nibh gravida ultrices.
94             </p>
95         </div>
96     </div>
97 </div>
98
99     <div class="footer">
100         <div>
101             <p>TOUTPASCHER</p>
102         </div>
103         <p>Tous droits réservés</p>
104     </div>
105 </body>
106 </html>
107

```

Question 1

[solution n°6 p.31]

Modifiez le code dont vous disposez pour adapter la sémantique. Aidez-vous des classes CSS ayant été définies.

N'oubliez pas de modifier la définition des éléments CSS au fil de vos adaptations.

Indice :

Le code ci-dessus utilise trop largement la balise `<div>`. L'objectif de cette première question sera donc d'utiliser les balises HTML5 afin d'ajouter de la sémantique à notre code HTML (`header`, `footer`, `nav`, `main`, `article`, `aside`, `section`).

Question 2

[solution n°7 p.33]

En vous appuyant sur le résultat de la question précédente, modifiez le code dont vous disposez dans ce sens :

- La navigation doit disposer d'un `padding` de 10 px et d'un `margin-bottom` de 10 px
- Les éléments de liste de la navigation doivent être affichés en ligne et disposer de marges de 10 px
- Le récapitulatif et le rappel doivent être situés sur une même ligne
- Le rappel sera affiché sur 70 % de la largeur, tandis que le récapitulatif sera affiché sur 25 %, et ils auront tous deux une hauteur de 500 px
- Les articles doivent disposer d'un `padding` de 20 px
- Les sections et le pied de page disposeront d'un `padding` de 10 px

Solutions des exercices

p. 7 Solution n°1

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <link rel="stylesheet" href="style.css">
7   <title>Document</title>
8 </head>
9 <body>
10  <header>entête</header>
11  <div class="content">
12    <main>
13      <article>
14        <h4>Blog post 1</h4>
15        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur nec ex
bibendum, viverra metus in, eleifend nibh. Fusce eu risus sem. Nullam lacinia aliquam elit,
et ullamcorper quam elementum eu.</p>
16      </article>
17      <article>
18        <h4>Blog post 2</h4>
19        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur nec ex
bibendum, viverra metus in, eleifend nibh. Fusce eu risus sem. Nullam lacinia aliquam elit,
et ullamcorper quam elementum eu.</p>
20      </article>
21      <article>
22        <h4>Blog post 3</h4>
23        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur nec ex
bibendum, viverra metus in, eleifend nibh. Fusce eu risus sem. Nullam lacinia aliquam elit,
et ullamcorper quam elementum eu.</p>
24      </article>
25    </main>
26    <aside>
27      <nav>
28        <h3>Menu</h3>
29        <ul>
30          <li><a href="#">Mes réalisations</a></li>
31          <li><a href="#">A propos</a></li>
32          <li><a href="#">Contact</a></li>
33          <li><a href="#">Site partenaire</a></li>
34        </ul>
35      </nav>
36    </aside>
37  </div>
38
39  <footer>pied de page</footer>
40 </body>
41 </html>

```

p. 10 Solution n°2

```

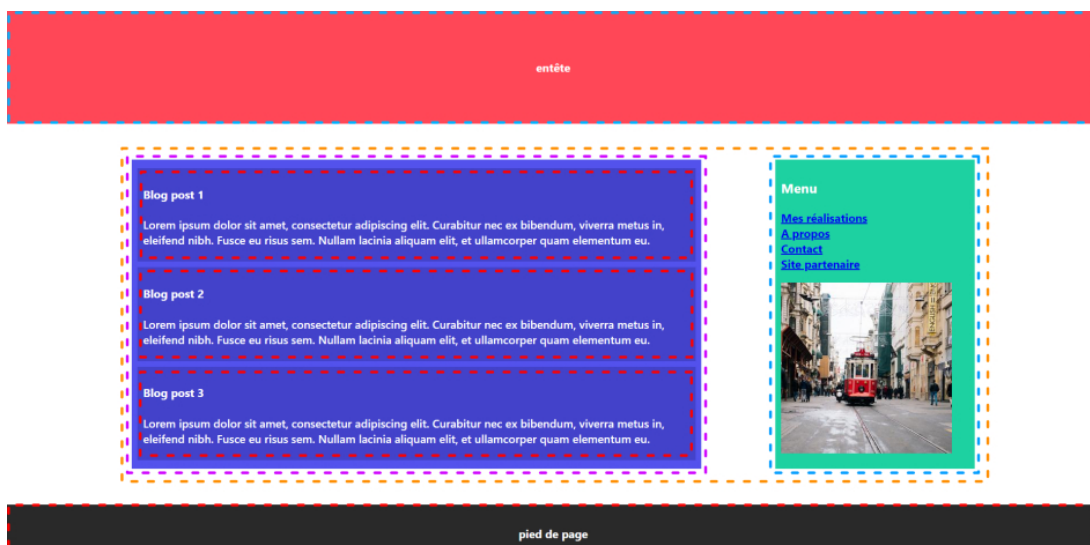
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">

```

```

6      <link rel="stylesheet" href="style.css">
7      <title>Blog</title>
8  </head>
9  <body>
10     <main>
11         <article>
12             <div>
13                 <h4>Titre <span>1</span></h4>
14                 <p>Paragraphe 1</p>
15             </div>
16             <p>Paragraphe 2</p>
17             <p>Paragraphe 3</p>
18
19
20         </article>
21         <article>
22             <div>
23                 <h4>Titre <span>2</span></h4>
24                 <p>Paragraphe 1</p>
25             </div>
26             <p>Paragraphe 2</p>
27             <p>Paragraphe 3</p>
28
29         </article>
30         <article>
31             <div>
32                 <h4>Titre <span>3</span></h4>
33                 <p>Paragraphe 1</p>
34             </div>
35             <p>Paragraphe 2</p>
36             <p>Paragraphe 3</p>
37
38         </article>
39
40     </main>
41 </body>
42 </html>
43
44

```



p. 21 Solution n°3

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="utf-8"/>
5 </head>
6 <body>
7 <nav>
8   <ul>
9     <li>Premier élément de navigation</li>
10    <li>Second élément de navigation</li>
11    <li>Troisième élément de navigation</li>
12  </ul>
13 </nav>
14
15 </body>
16 <style>
17   nav li {
18     border: 1px solid red;
19     display: inline;
20     padding: 10px;
21     margin: 25px;
22   }
23 </style>
24 </html>
```

p. 22 Solution n°4

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="utf-8"/>
5 </head>
6 <body>
7
8 <article>
9   Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod fugiat nulla
10  pariatur.
11 </article>
12 <nav>
13   <h2>Menu de navigation vertical</h2>
14   <ul>
15     <li>Premier élément de navigation</li>
16     <li>Second élément de navigation</li>
17     <li>Troisième élément de navigation</li>
18   </ul>
19 </nav>
20 </body>
21 <style>
22   nav {
23     display: inline-block;
24     background-color: gray;
25     width: 15%;
26     height: 500px;
27     vertical-align: top;
28   }
```

```

28
29     article {
30         display: inline-block;
31         width: 60%;
32         vertical-align: top;
33         height: 500px;
34         background-color: #9fcdff;
35     }
36 </style>
37 </html>
38

```

Exercice p. 22 Solution n°5

Exercice

On dit de la balise <div> qu'il s'agit d'une balise de type...

- ☒ bloc
- ☐ inline

Exercice

Afin de respecter la sémantique, dans quelle balise vaut-il mieux regrouper des éléments d'en-tête, tels qu'un logo ou le nom de la société ?

- ☐ <div>
- ☐ <nav>
- ☒ <header>

Exercice

Contrairement aux balises <div> ou , le fait d'utiliser des balises telles que <nav> ou <footer> impacte le résultat visuel de la page.

- ☐ Vrai
- ☒ Faux
- ☐ C'est faux, il ne s'agit que de sémantique.

Exercice

Si l'on souhaite présenter des informations indépendantes les unes des autres, telles qu'un listing de différents produits sur un récapitulatif d'achat, on préférera l'utilisation de la balise...

- ☐ <section>
- ☐ <aside>
- ☒ <article>

- ☐ L'élément <article> contient une information qui est indépendante du reste de la page. Par exemple, lorsque nous représentons une galerie d'articles, ceux-ci sont indépendants les uns des autres (ils pourraient exister les uns sans les autres).

Exercice

Pour forcer l'affichage de plusieurs éléments sur la même ligne, dans la limite de leur largeur, on utilisera...

- ☒ `display: inline;`
- ☐ `display: block;`

Exercice

Dans quel cas les propriétés `width` et `height` d'un élément sont-elles prises en compte ?

- ☐ `display: inline`
- ☒ `display: inline-block`

Exercice

Si l'on souhaite définir la largeur maximale d'un élément, on utilisera la propriété...

- ☒ `max-width`
- ☐ `max-height`
- ☐ `width`
- ☐ `height`

Exercice

Par défaut, un paragraphe ou une liste sont affichés...

- ☐ En ligne
- ☒ En bloc

Exercice

Laquelle de ces propriétés représente la marge interne d'un élément ?

- ☐ `margin`
- ☒ `padding`

Exercice

La propriété `overflow` permet de gérer...

- ☐ L'espacement sur la page
- ☐ Les renvois à la ligne
- ☒ La gestion du dépassement d'une boîte

p. 26 Solution n°6

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="utf-8">
5   <title>Finaliser mon achat</title>
6   <link href="https://fonts.googleapis.com/css?family=Lato:400,100,300" rel="stylesheet">
7   <style>
```

```

8      body {
9          font-family: 'Lato', sans-serif;
10         margin: 0;
11     }
12
13     header {
14         text-align: center;
15     }
16
17     nav {
18         background-color: lightgrey;
19     }
20
21     nav li {
22     }
23
24     main {
25         border-left: solid 1px lightgrey;
26     }
27
28     article {
29     }
30
31     aside {
32         background-color: #b0b0ff;
33         vertical-align: top;
34     }
35
36     section {
37     }
38
39     footer {
40         text-align: center;
41         background-color: #000634;
42         color: white;
43         margin-top: 20px;
44     }
45     </style>
46 </head>
47 <body>
48 <header>
49     <h1>TOUTPASCHER</h1>
50 </header>
51 <nav role="navigation">
52     <ul>
53         <li><a href="#">Page d'accueil</a></li>
54         <li><a href="#">Mon compte</a></li>
55         <li><a href="#">Mes derniers achats</a></li>
56     </ul>
57 </nav>
58 <aside>
59     <article>
60         <h2>Articles dans votre panier</h2>
61         <ul>
62             <li>PC Portable - TRESPUISSANT : 999€</li>
63             <li>Chaise de bureau - BIENASSIS : 199€</li>
64             <li>Réduction fidélité : 50€</li>
65             <li>Total : <strong>1148€</strong> </li>

```



```

66
67     </ul>
68     <h3>Procéder au paiement</h3>
69     <button>Aller au paiement</button>
70 </article>
71 </aside>
72 <main>
73
74     <section>
75         <h2>Rappel de votre sélection</h2>
76         <hr>
77         <article>
78             <h3>PC Portable - TRESPUISSANT : 999€</h3>
79             <p>
80                 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
81                 Curabitur fringilla nisi neque, non commodo diam tincidunt quis.
82                 Donec a nibh sed ipsum fermentum vehicula vel ac lectus.
83                 Nulla congue feugiat tellus, ut imperdiet enim elementum a.
84                 Sed sed odio eu nibh gravida ultrices.
85             </p>
86         </article>
87         <article>
88             <h3>Chaise de bureau - BIENASSIS : 199€</h3>
89             <p>
90                 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
91                 Curabitur fringilla nisi neque, non commodo diam tincidunt quis.
92                 Donec a nibh sed ipsum fermentum vehicula vel ac lectus.
93                 Nulla congue feugiat tellus, ut imperdiet enim elementum a.
94                 Sed sed odio eu nibh gravida ultrices.
95             </p>
96         </article>
97     </section>
98 </main>
99
100 <footer>
101     <div>
102         <p>TOUTPASCHER</p>
103     </div>
104     <p>Tous droits réservés</p>
105 </footer>
106 </body>
107 </html>
108

```

p. 26 Solution n°7

```

1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4     <meta charset="utf-8">
5     <title>Finaliser mon achat</title>
6     <link href="https://fonts.googleapis.com/css?family=Lato:400,100,300" rel="stylesheet">
7     <style>
8         body {
9             font-family: 'Lato', sans-serif;
10            margin: 0;
11        }

```

```

12
13     header {
14         text-align: center;
15     }
16
17     nav {
18         background-color: lightgrey;
19         margin-bottom: 10px;
20         padding: 10px;
21     }
22
23     nav li {
24         display: inline;
25         margin: 10px;
26     }
27
28     main {
29         display: inline-block;
30         width: 70%;
31         height: 500px;
32         border-left: solid 1px lightgrey;
33     }
34
35     article {
36         padding: 20px;
37     }
38
39     aside {
40         display: inline-block;
41         width: 25%;
42         background-color: #b0b0ff;
43         height: 500px;
44         vertical-align: top;
45     }
46
47     section {
48         padding: 10px;
49     }
50
51     footer {
52         text-align: center;
53         background-color: #000634;
54         color: white;
55         margin-top: 20px;
56         padding: 10px;
57     }
58 </style>
59 </head>
60 <body>
61 <header>
62     <h1>TOUTPASCHER</h1>
63 </header>
64 <nav role="navigation">
65     <ul>
66         <li><a href="#">Page d'accueil</a></li>
67         <li><a href="#">Mon compte</a></li>
68         <li><a href="#">Mes derniers achats</a></li>
69     </ul>

```

```

70 </nav>
71 <aside>
72   <article>
73     <h2>Articles dans votre panier</h2>
74     <ul>
75       <li>PC Portable - TRESPUISSANT : 999€</li>
76       <li>Chaise de bureau - BIENASSIS : 199€</li>
77       <li>Réduction fidélité : 50€</li>
78       <li>Total : <strong>1148€</strong> </li>
79
80     </ul>
81     <h3>Procéder au paiement</h3>
82     <button>Aller au paiement</button>
83   </article>
84 </aside>
85 <main>
86
87   <section>
88     <h2>Rappel de votre sélection</h2>
89     <hr>
90     <article>
91       <h3>PC Portable - TRESPUISSANT : 999€</h3>
92       <p>
93         Lorem ipsum dolor sit amet, consectetur adipiscing elit.
94         Curabitur fringilla nisi neque, non commodo diam tincidunt quis.
95         Donec a nibh sed ipsum fermentum vehicula vel ac lectus.
96         Nulla congue feugiat tellus, ut imperdiet enim elementum a.
97         Sed sed odio eu nibh gravida ultrices.
98       </p>
99     </article>
100    <article>
101      <h3>Chaise de bureau - BIENASSIS : 199€</h3>
102      <p>
103        Lorem ipsum dolor sit amet, consectetur adipiscing elit.
104        Curabitur fringilla nisi neque, non commodo diam tincidunt quis.
105        Donec a nibh sed ipsum fermentum vehicula vel ac lectus.
106        Nulla congue feugiat tellus, ut imperdiet enim elementum a.
107        Sed sed odio eu nibh gravida ultrices.
108      </p>
109    </article>
110  </section>
111 </main>
112
113 <footer>
114   <div>
115     <p>TOUTPASCHER</p>
116   </div>
117   <p>Tous droits réservés</p>
118 </footer>
119 </body>
120 </html>
121

```

Contenus annexes

1. Visual Studio Code

Objectifs

- Comprendre pourquoi utiliser un éditeur de texte
- Découvrir et installer notre premier éditeur de texte

Mise en situation

Afin de créer notre premier site web, il va nous falloir télécharger les bons outils et comprendre en quoi ils nous sont utiles.

Définition Présentation

Les pages web HTML sont écrites dans des fichiers textes avec l'extension *.html*.

Ce sont des fichiers contenant du texte, au même titre qu'une image peut se trouver sous l'extension *.jpg* ou encore les chansons que l'on trouve dans des fichiers *.mp3*.

Il faut un logiciel pour les créer et les éditer : pour cela, nous utiliserons un éditeur de texte.

Conseil

On pourrait utiliser un éditeur de texte standard (Bloc-Notes, vi, emacs...), mais, pour des raisons de confort, nous utiliserons un éditeur de texte plus avancé : Visual Studio Code. Il permettra notamment de colorer automatiquement les éléments du code, afficher des aides, etc.

Installation

Maintenant, installons notre premier logiciel qui nous permettra d'écrire du HTML : <https://code.visualstudio.com/>.

À la fin de l'installation, ouvrons le logiciel Visual Studio Code :

- Cliquez sur le menu "File" puis "New file", un nouveau fichier vierge apparaît au centre.
- Sauvegardez le nouveau fichier au format *.html*. Par exemple, appelons-le *premiere-page.html*.
- Écrivez une phrase (par exemple, "Voici un fichier HTML") et sauvegardez.
- Double-cliquez sur le fichier pour l'ouvrir dans le navigateur ou effectuez un glisser-déposer vers votre navigateur préféré, le texte va s'afficher !

Les pages web HTML sont des fichiers textes presque comme les autres !

Syntaxe À retenir

- Un éditeur de texte est un logiciel qui permet de créer et d'éditer des fichiers notamment des fichiers textes.
- Pour écrire du code HTML, des éditeurs plus adaptés que les éditeurs de texte standard existent, offrant de multiples fonctionnalités.