

TD R1.02

Éléments de correction

C. Rozé

TD1 : HTML

Question 2 : Les sections.

Définir les sections principales du CV. Chaque section doit commencer par un titre de niveau 2.

```
<!doctype html>
<html lang="fr">
<head>
<title> mon cv </title>
<meta charset="utf-8">

</head>
<body>

<header>
    <nav>
        <ul>
            <li><a href="#info">Présentation</a></li>
            <li><a href="#formation">Formation</a></li>
            <li><a href="#langue">Langues</a></li>
            <li><a href="#contact">Contact</a></li>
        </ul>
    </nav>
</header>

<h1>CV</h1>
<figure>
    
    <figcaption>Portrait de l'artiste</figcaption>
</figure>

<section id="info">
<h2>A propos de moi</h2>
<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. <strong>Duis tortor enim</strong>, volutpat in lorem a, finibus vehicula ex.
Nulla ac dui sit amet nisi euismod pulvinar. Suspendisse id justo sem. Fusce et viverra libero. Suspendisse potenti. Phasellus sed
ullamcorper urna. Sed dictum efficitur dignissim. </p>
</section>

<section id="formation">
<h2>Mes études</h2>
<p> Lieu de formation <a href="http://www.uphf.fr" target="_blank"> UPHF </a></p>
<ul>
    <li> Algorithmique en Python </li>
    <li> Conception d'interface web </li>
    <li> Base de données </li>
</ul>
</section>

<section id="langue">
<h2>Langues parlées</h2>
<table>
    <tr><th>Langage</th><th>Expérience</th><th>Niveau</th></tr>
    <tr><td> Francais</td><td>30 ans</td><td>Débutant</td></tr>
    <tr><td> HTML</td><td>1h</td><td>Débutant</td></tr>
</table>
</section>

<section id="contact">
<h2>Information de contact</h2>
    Nom : John Doe<br>
    Téléphone : +33 6 12 34 56 78<br>
    Email : john.doe@example.com
</section>

</body>
</html>
```

cv.html

Question 2 : Les sections.

Définir les sections principales du CV. Chaque section doit commencer par un titre de niveau 2.

Faire une liste (non numérotée) indiquant les cours que vous suivez à l'IUT.

```
<!doctype html>
<html lang="fr">
<head>
<title> mon cv </title>
<meta charset="utf-8">

</head>
<body>

<header>
    <nav>
        <ul>
            <li><a href="#info">Présentation</a></li>
            <li><a href="#formation">Formation</a></li>
            <li><a href="#langue">Langues</a></li>
            <li><a href="#contact">Contact</a></li>
        </ul>
    </nav>
</header>

<h1>CV</h1>
<figure>
    
    <figcaption>Portrait de l'artiste</figcaption>
</figure>

<section id="info">
<h2>A propos de moi</h2>
<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. <strong>Duis tortor enim</strong>, volutpat in lorem a, finibus vehicula ex.
Nulla ac dui sit amet nisi euismod pulvinar. Suspendisse id justo sem. Fusce et viverra libero. Suspendisse potenti. Phasellus sed
ullamcorper urna. Sed dictum efficitur dignissim. </p>
</section>

<section id="formation">
<h2>Mes &eacute;tudes</h2>
<p> Lieu de formation <a href="http://www.uphf.fr" target="_blank"> UPHF </a></p>
<ul>
    <li> Algorithmique en Python </li>
    <li> Conception d'interface web </li>
    <li> Base de données </li>
</ul>
</section>

<section id="langue">
<h2>Langues parlées</h2>
<table>
    <tr><th>Langage</th><th>Expérience</th><th>Niveau</th></tr>
    <tr><td> Francais</td><td>30 ans</td><td>Débutant</td></tr>
    <tr><td> HTML</td><td>1h</td><td>Débutant</td></tr>
</table>
</section>

<section id="contact">
<h2>Information de contact</h2>
    Nom : John Doe<br>
    Téléphone : +33 6 12 34 56 78<br>
    Email : john.doe@example.com
</section>

</body>
</html>
```

cv.html

Question 2 : Les sections.

Définir les sections principales du CV. Chaque section doit commencer par un titre de niveau 2.

Faire une liste (non numérotée) indiquant les cours que vous suivez à l'IUT.

Troisième section « Langues parlées » : Créer un tableau (élément table)

```
<!doctype html>
<html lang="fr">
<head>
<title> mon cv </title>
<meta charset="utf-8">

</head>
<body>

<header>
    <nav>
        <ul>
            <li><a href="#info">Présentation</a></li>
            <li><a href="#formation">Formation</a></li>
            <li><a href="#langue">Langues</a></li>
            <li><a href="#contact">Contact</a></li>
        </ul>
    </nav>
</header>

<h1>CV</h1>
<figure>
    
    <figcaption>Portrait de l'artiste</figcaption>
</figure>

<section id="info">
<h2>A propos de moi</h2>
<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. <strong>Duis tortor enim</strong>, volutpat in lorem a, finibus vehicula ex.
Nulla ac dui sit amet nisi euismod pulvinar. Suspendisse id justo sem. Fusce et viverra libero. Suspendisse potenti. Phasellus sed
ullamcorper urna. Sed dictum efficitur dignissim. </p>
</section>

<section id="formation">
<h2>Mes études</h2>
<p> Lieu de formation <a href="http://www.uphf.fr" target="_blank"> UPHF </a></p>
<ul>
    <li> Algorithmique en Python </li>
    <li> Conception d'interface web </li>
    <li> Base de données </li>
</ul>
</section>

<section id="langue">
<h2>Langues parlées</h2>
<table>
    <tr><th>Langage</th><th>Expérience</th><th>Niveau</th></tr>
    <tr><td> Francais</td><td>30 ans</td><td>Débutant</td></tr>
    <tr><td> HTML</td><td>1h</td><td>Débutant</td></tr>
</table>
</section>

<section id="contact">
<h2>Information de contact</h2>
    Nom : John Doe<br>
    Téléphone : +33 6 12 34 56 78<br>
    Email : john.doe@example.com
</section>

</body>
</html>
```

cv.html

Faire une liste (non numérotée) indiquant les cours que vous suivez à l'IUT.

Troisième section « Langues parlées » : Créer un tableau (élément table)

Question 3 : Photo

Insérer une photo après le titre en utilisant l'élément `<img />`.

Indiquer sous la photo son origine. Pour cela, placer après la photo un élément `figcaption` (légende). L'image et sa légende vont dans un élément `figure`.

```
<!doctype html>
<html lang="fr">
<head>
<title> mon cv </title>
<meta charset="utf-8">

</head>
<body>

<header>
    <nav>
        <ul>
            <li><a href="#info">Présentation</a></li>
            <li><a href="#formation">Formation</a></li>
            <li><a href="#langue">Langues</a></li>
            <li><a href="#contact">Contact</a></li>
        </ul>
    </nav>
</header>

<h1>CV</h1>

<figure>
    
    <figcaption>Portrait de l'artiste</figcaption>
</figure>
```

```
<section id="langue">
<h2>Langues parlées</h2>
<table>
    <tr><th>Langage</th><th>Expérience</th><th>Niveau</th></tr>
    <tr><td> Francais</td><td>30 ans</td><td>Débutant</td></tr>
    <tr><td> HTML</td><td>1h</td><td>Débutant</td></tr>
</table>
</section>

<section id="contact">
<h2>Information de contact</h2>
    Nom : John Doe<br>
    Téléphone : +33 6 12 34 56 78<br>
    Email : john.doe@example.com
</section>

</body>
</html>
```

```
<section id="info">
<h2>A propos de moi</h2>
<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. <strong>Duis tortor enim</strong>, volutpat in lorem a, finibus vehicula ex.
Nulla ac dui sit amet nisi euismod pulvinar. Suspendisse id justo sem. Fusce et viverra libero. Suspendisse potenti. Phasellus sed
ullamcorper urna. Sed dictum efficitur dignissim. </p>
</section>

<section id="formation">
<h2>Mes &eacute;tudes</h2>
<p> Lieu de formation <a href="http://www.uphf.fr" target="_blank"> UPHF </a></p>
<ul>
    <li> Algorithmique en Python </li>
    <li> Conception d'interface web </li>
    <li> Base de données </li>
</ul>
</section>
```

Créer un menu de navigation placé dans l'en-tête du body (en trouvant les éléments correspondants). Dans ce menu, créer des liens vers les différentes sections de votre CV : utiliser « # » dans les attributs href (par exemple href=« #Formation »).

```
<!doctype html>
<html lang="fr">
<head>
<title> mon cv </title>
<meta charset="utf-8">

</head>
<body>

<header>
  <nav>
    <ul>
      <li><a href="#info">Présentation</a></li>
      <li><a href="#formation">Formation</a></li>
      <li><a href="#langue">Langues</a></li>
      <li><a href="#contact">Contact</a></li>
    </ul>
  </nav>
</header>

<h1>CV</h1>
<figure>
  
  <figcaption>Portrait de l'artiste</figcaption>
</figure>

<section id="info">
<h2>A propos de moi</h2>
<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. <strong>Duis tortor enim</strong>, volutpat in lorem a, finibus vehicula ex.
Nulla ac dui sit amet nisi euismod pulvinar. Suspendisse id justo sem. Fusce et viverra libero. Suspendisse potenti. Phasellus sed
ullamcorper urna. Sed dictum efficitur dignissim. </p>
</section>

<section id="formation">
<h2>Mes études</h2>
<p> Lieu de formation <a href="http://www.uphf.fr" target="_blank"> UPHF </a></p>
<ul>
  <li> Algorithmique en Python </li>
  <li> Conception d'interface web </li>
  <li> Base de données </li>
</ul>
</section>

<section id="langue">
<h2>Langues parlées</h2>
<table>
  <tr><th>Langage</th><th>Expérience</th><th>Niveau</th></tr>
  <tr><td> Français</td><td>30 ans</td><td>Débutant</td></tr>
  <tr><td> HTML</td><td>1h</td><td>Débutant</td></tr>
</table>
</section>

<section id="contact">
<h2>Information de contact</h2>
  Nom : John Doe<br>
  Téléphone : +33 6 12 34 56 78<br>
  Email : john.doe@example.com
</section>

</body>
</html>
```

TD2 : CSS

Question 1 : CSS

Créer une feuille de style vide au format CSS (texte brut avec extension .css). Relier la feuille de style à votre page HTML en utilisant dans l'élément head la balise `<link rel="stylesheet" href="chemin/vers/votre/feuille/CSS"/>`.

Dans la feuille CSS, ajouter une ligne `h1 {color:valeur;}`, où valeur est un nom de couleur. Constaté le résultat.

Modifier la police utilisée dans certains titres (h1 ou h2, à votre choix). Fixer aussi la taille que vous désirez avec font-size.

style.css

```
<!doctype html>
<html lang="fr">
<head>
<title> mon cv </title>
<meta charset="utf-8">
<link rel="stylesheet" href="style.css" >
</head>
<body>

<header>
    <nav>
        <ul>
            <li><a href="#info">Présentation</a></li>
            <li><a href="#formation">Formation</a></li>
            <li><a href="#langue">Langues</a></li>
            <li><a href="#contact">Contact</a></li>
        </ul>
    </nav>
</header>

<h1>CV</h1>
<figure>
    
    <figcaption>Portrait de l'artiste</figcaption>
</figure>
```

```
body{
    background-color: #f4f4f4;
}

section{
    background-color: white;
    border-radius: 8px;
    padding-left:10px;
}

h1 {
    color:lightblue;
}

h2 {
    font-family: Arial, sans-serif;
}

img {
    width:100px;
    height:auto;
}
```

Question 2 : Positionnement de l'image

Nous allons redimensionner l'image et modifier l'apparence de la légende.

Appliquer à l'image les propriétés width:x; height:auto; en choisissant une bonne valeur pour x (en cm, ou px...). La valeur pour height garantit que l'image n'est pas déformée.

Modifier la légende pour que la police soit plus petite que la normale et en italique.

Utiliser la propriété CSS float avec une valeur left ou right pour placer l'image et sa légende sur le côté du texte.

```
img {  
    width:100px;  
    height:auto;  
}  
  
figure{  
    float:right;  
}  
  
figcaption{  
    font-size: 0.8em;  
    font-style: italic;  
    color: #555;  
}
```

Question 3 : Modification liste et tableau

Dans la liste des cours suivis, remplacer les puces de base par votre option préférée en appliquant le CSS à l'élément `<ul>`. Utiliser la propriété CSS `list-style-type` (<https://developer.mozilla.org/fr/docs/Web/CSS/list-style-type>)

Choisir un ou plusieurs cours que vous préférez, et leur donner une puce particulière et un nom mis en valeur. Pour cela, on donnera à l'élément `li` correspondant un attribut `class` pour pouvoir le sélectionner dans le CSS.

```
<section id="formation">
<h2>Mes études</h2>
<p> Lieu de formation <a href="http://www.uphf.fr" target="_blank"> UPHF </a></p>
<ul>
  <li> Algorithmique en Python </li>
  <li class="super"> Conception d'interface web </li>
  <li class="super"> Base de données </li>
</ul>
</section>
```

```
li {
  list-style-type:square;
}

li.super {
  list-style-type:circle;
  font-weight:bold;
}
```

Rendu visuel avec css

Mes études

Lieu de formation [UPHF](http://www.uphf.fr)

- Algorithmique en Python
- **Conception d'interface web**
- **Base de données**

Question 3 : Modification liste et tableau

Terminer la mise en forme du tableau en rajoutant des bordures à certains endroits (pas partout).

Pour créer des bordures, utiliser les propriétés border-bottom, border-top, border-right...

Pour ne pas avoir de bordures en double, appliquez la propriété border-collapse:collapse; à la table.

Améliorer le rendu visuel de la table grâce aux propriétés text-align, padding-right et padding-left.

```
<section id="langue">
<h2>Langues parlées</h2>
<table>
  <tr><th>Langage</th><th>Expérience</th><th>Niveau</th></tr>
  <tr><td>Français</td><td>30 ans</td><td>Débutant</td></tr>
  <tr><td>HTML</td><td>1h</td><td>Débutant</td></tr>
</table>
</section>
```

Rendu visuel sans css

Langues parlées

Langage	Expérience	Niveau
Français	30 ans	Débutant
HTML	1h	Débutant

Question 3 : Modification liste et tableau

Terminer la mise en forme du tableau en rajoutant des bordures à certains endroits (pas partout).

Pour créer des bordures, utiliser les propriétés border-bottom, border-top, border-right...

Pour ne pas avoir de bordures en double, appliquez la propriété border-collapse:collapse; à la table.

Améliorer le rendu visuel de la table grâce aux propriétés text-align, padding-right et padding-left.

```
<section id="langue">
<h2>Langues parlées</h2>
<table>
  <tr><th>Langage</th><th>Expérience</th><th>Niveau</th></tr>
  <tr><td>Francais</td><td>30 ans</td><td>Débutant</td></tr>
  <tr><td>HTML</td><td>1h</td><td>Débutant</td></tr>
</table>
</section>
```

```
table, th, tr, td {
  border:1px solid blue;
  border-collapse : collapse;
  text-align:center;
}
```

Rendu visuel sans css

Langues parlées

Langage	Expérience	Niveau
Francais	30 ans	Débutant
HTML	1h	Débutant

Rendu visuel avec css

Langues parlées

Langage	Expérience	Niveau
Francais	30 ans	Débutant
HTML	1h	Débutant

Rendu visuel avec css 2

Langues parlées

Langage	Expérience	Niveau
Francais	30 ans	Débutant
HTML	1h	Débutant

Question 4 : display

Les liens du menu de navigation ressemblent à des liens normaux. On voudrait maintenant le faire ressembler à un vrai menu : chaque lien est entouré de sa boîte, et que le lien soit suivi quand on clique dans la boîte.

Modifier le style des liens pour qu'ils ne soient pas soulignés ni en bleu ;

Pour créer une boîte visible, leur donner une border et une background-color de votre choix.

NB : Choisir le bon sélecteur pour vous assurer que seuls les liens du menu soient affectés.

Modifier les propriétés width et height pour faire des boîtes d'une taille agréable. Que se passe-t-il ? Et si vous leur appliquez les propriétés display:block ou display:inline-block ?

```
<body>
<header>
  <nav>
    <ul>
      <li><a href="#info">Présentation</a></li>
      <li><a href="#formation">Formation</a></li>
      <li><a href="#langue">Langues</a></li>
      <li><a href="#contact">Contact</a></li>
    </ul>
  </nav>
</header>
<h1>CV</h1>
```

Rendu visuel sans css

- [Présentation](#)
- [Formation](#)
- [Langues](#)
- [Contact](#)

CV

```
nav a {
  margin:auto;
  text-decoration:none;
  text-align:center;
  color:white;
  border: 1px solid black;
  background-color:black;
  width: 100px;
  height:20px;
  display:inline-block;
}
```

Rendu visuel avec css

- Présentation
- Formation
- Langues
- Contact

CV

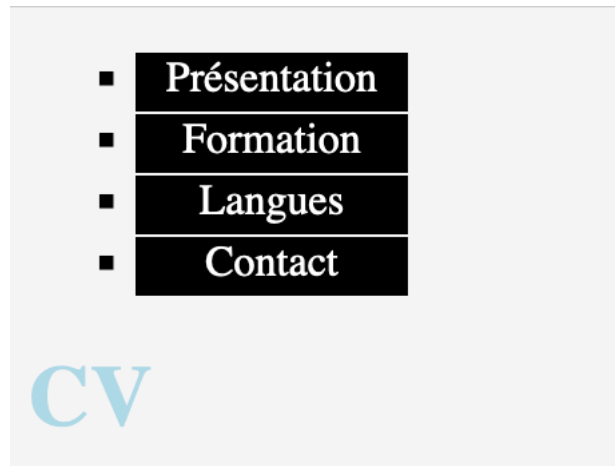
Question 4 : display

Améliorer l'apparence de votre menu en donnant une bordure et une couleur de fond au menu de navigation, et en utilisant les propriétés suivantes :

- margin et padding dans chaque boîte a, pour ajouter de l'espace dans et entre les boîtes ;
- text-align pour centrer le menu ;
- width et height pour fixer la hauteur des boîtes.

```
nav a {  
  margin:auto;  
  text-decoration:none;  
  text-align:center;  
  color:white;  
  border: 1px solid black;  
  background-color:black;  
  width: 100px;  
  height:20px;  
  display:inline-block;  
}
```

Rendu visuel avec css



```
nav {  
  margin:auto;  
  padding:5px;  
  width:425px;  
  height:25px;  
}  
  
nav li { display:inline;}  
  
nav ul {  
  margin:auto;  
  padding:0px;  
}  
  
nav ul li a:hover {  
  background-color: lightblue;}
```

Rendu visuel avec les 2 css



Question 5 : Positionnement fixed

Dans le pied de page (footer), tout en bas de la page, on va rajouter un petit bloc centré, d'une couleur différente. Vous y mettrez un message personnel.

Définir un bloc et y ajouter un contenu. Choisir une couleur de fond. Lui donner une largeur et une hauteur égales, soit absolue (px, em, cm), soit en pourcentage.
Pour améliorer la mise en page, vous pouvez choisir un alignement (text-align).

Pour centrer le bloc, text-align ne marche pas car il ne s'agit pas d'un texte (élément inline). À la place, il faut appliquer au bloc margin:auto; qui place des marges égales des deux cotés.

```
<footer>
  &#xa9; 2024 Mon site
</footer>

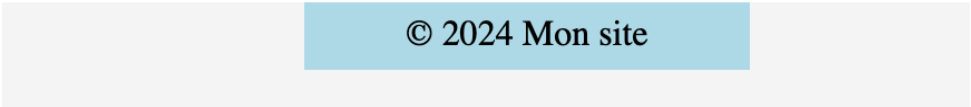
</body>
</html>
```

Rendu visuel css



```
footer {
  background-color:lightblue;
  width : 200px;
  height: 30px;
  line-height:30px;
  text-align: center;
  margin:auto;
  text-align:center;
}
```

Rendu visuel avec css



Question 5 : Positionnement fixed

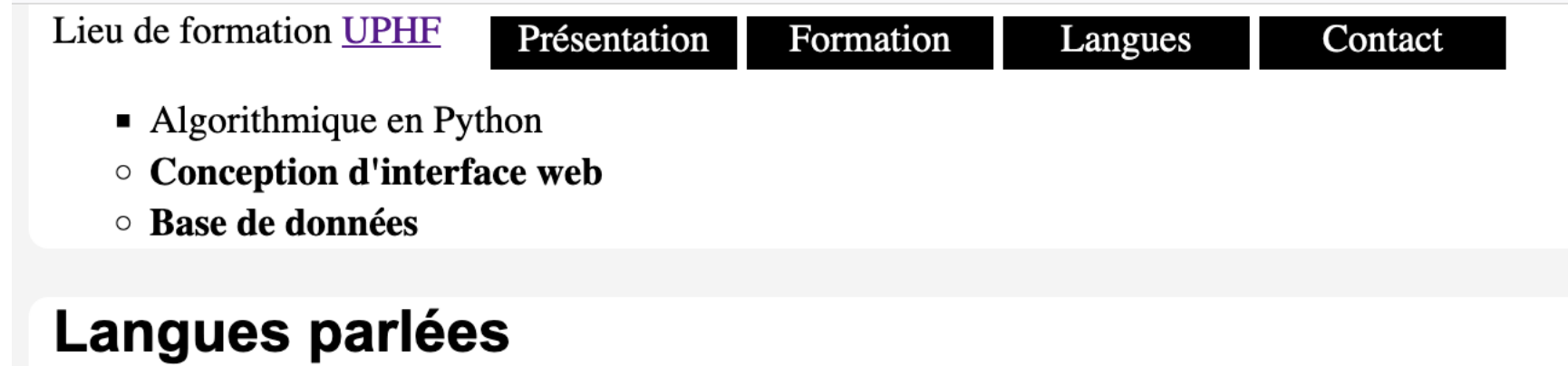
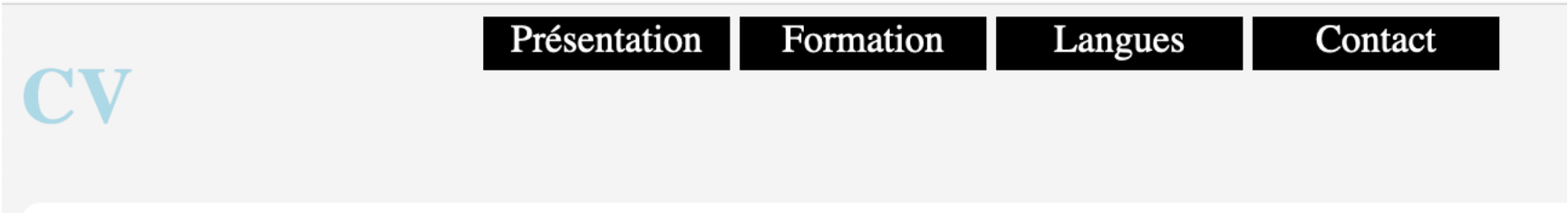
Nous allons maintenant faire en sorte que l'en-tête de navigation soit toujours visible. C'est une méthode courante pour les sites avec beaucoup de contenu.
Appliquer la propriété position:fixed au header. Pour voir le résultat, zoomer sur la page et la faire défiler.

Comme le bloc a été sorti du flot normal, l'espace qui lui était réservé a été enlevé. Pour résoudre ce problème, une méthode est d'ajouter de la marge supérieure (margin-top) au bloc body contenant toute la page, puis de positionner le menu avec top dans l'espace libéré.

```
header{
  position:fixed;
  top:0px;
  text-align:center;
  width: 100%;
}

body {
  margin-top:50px;
}
```

Rendu visuel avec css



TD3 : Responsive design

Nous allons améliorer la manière dont l'image s'affiche à toutes les tailles. Dans le CSS, donnez à l'image une largeur en pourcentage et utilisez les propriétés min-width et max-width pour que l'image se redimensionne et reste à une taille appropriée pour toutes les tailles d'écrans.

style1.css

```
img {width:20%;  
    height:auto;  
    min-width:50px;  
    max-width:150px;}
```

Faites en sorte que, pour les très petits écrans (mobiles), l'image n'ait pas de texte à sa gauche ou à sa droite ; autrement dit, désactivez sa propriété `float`.

```
@media (max-width: 600px) {  
    figure {  
        float: none;  
    }  
}
```

En utilisant les requêtes media, on peut afficher une information différente pour chaque taille. Sous le titre, mettez un texte qui indique :

- "Version mobile" si l'écran fait moins de 640 pixels ;
- "Version tablette" si l'écran est entre 641 et 1000 pixels ;
- "Version bureau" si l'écran fait plus de 1001 pixels.

```
<h1>CV</h1>
```

```
<!-- Textes à afficher selon la taille de l'écran -->
```

```
<p id="mobile">Version mobile</p>
```

```
<p id="tablet">Version tablette</p>
```

```
<p id="desktop">Version bureau</p>
```

style1.css

```
#mobile, #tablet, #desktop {  
    display: none;  
}
```

```
/* Pour les petits écrans (moins de 640 pixels) */
```

```
@media (max-width: 640px) {
```

```
    #mobile {
```

```
        display: block;
```

```
    }
```

```
}
```

```
/* Pour les écrans intermédiaires (tablettes : 641px à 1000px) */
```

```
@media (min-width: 641px) and (max-width: 1000px) {
```

```
    #tablet {
```

```
        display: block;
```

```
    }
```

```
}
```

```
/* Pour les grands écrans (plus de 1001 pixels) */
```

```
@media (min-width: 1001px) {
```

```
    #desktop {
```

```
        display: block;
```

```
    }
```

```
}
```

Pour terminer, améliorons l’affichage du menu de navigation. Faites en sorte que les liens dans le menu soient l’un au-dessus de l’autre dans la version mobile et sur la même ligne pour la version tablette et bureau.

style1.css

```
nav ul li { display:inline-block;
}
```

```
@media (max-width: 640px) {
  #mobile {
    display: block;
  }
  nav ul li {
    display: block;
    margin: 10px 0;
  }
}
```

Utilisation de Flexbox

Pour faire de la mise en page avec Flexbox, il faut :

1. Définir un conteneur.
2. Et placer à l'intérieur plusieurs éléments.

Un conteneur (*container* en anglais) est une balise qui peut renfermer d'autres balises.

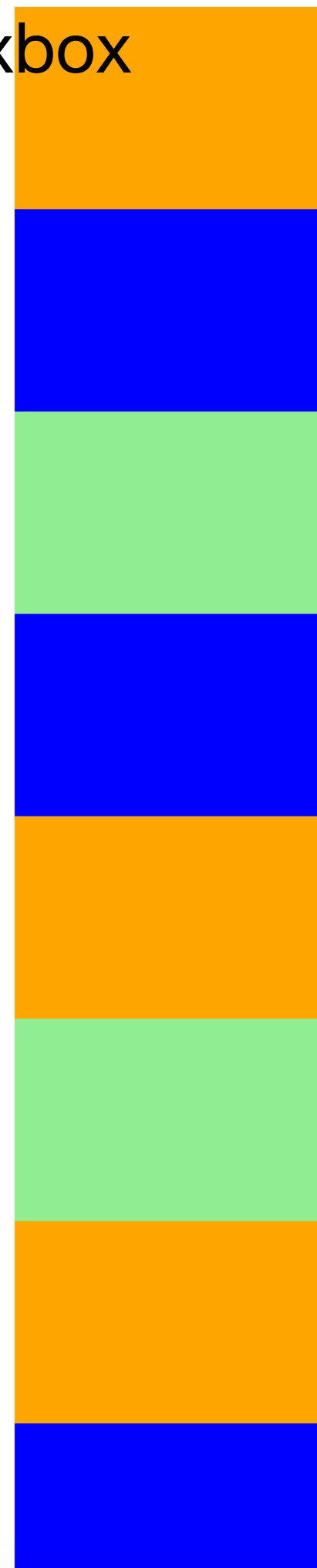
```
<div class="container">
  <div class="element element1"></div>
  <div class="element element2"></div>
  <div class="element element3"></div>
  <div class="element element2"></div>
  <div class="element element1"></div>
  <div class="element element3"></div>
  <div class="element element1"></div>
  <div class="element element2"></div>
  <div class="element element3"></div>
  <div class="element element2"></div>
  <div class="element element1"></div>
  <div class="element element3"></div>
</div>
```

```
.element {
  width: 150px;
  min-height: 100px;
}

.element1 {
  background-color : orange;
}

.element2 {
  background-color : blue;
}

.element3 {
  background-color : lightgreen;
}
```



Utilisation de Flexbox

La propriété `display:flex` va placer les blocs côte à côte.

```
<div class="container">
  <div class="element element1"></div>
  <div class="element element2"></div>
  <div class="element element3"></div>
  <div class="element element2"></div>
  <div class="element element1"></div>
  <div class="element element3"></div>
  <div class="element element1"></div>
  <div class="element element2"></div>
  <div class="element element3"></div>
  <div class="element element2"></div>
  <div class="element element1"></div>
  <div class="element element3"></div>
</div>
```

```
.container {
  display: flex;
}
```

Avec flexbox



Utilisation de Flexbox

Flexbox permet d'agencer ces éléments dans le sens que l'on veut. Avec **flex-direction**, on peut les positionner verticalement ou encore les inverser.

- row : organisés sur une ligne (par défaut) ;
- column : organisés sur une colonne ;
- row-reverse : organisés sur une ligne, mais en ordre inversé ;
- column-reverse : organisés sur une colonne, mais en ordre inversé.

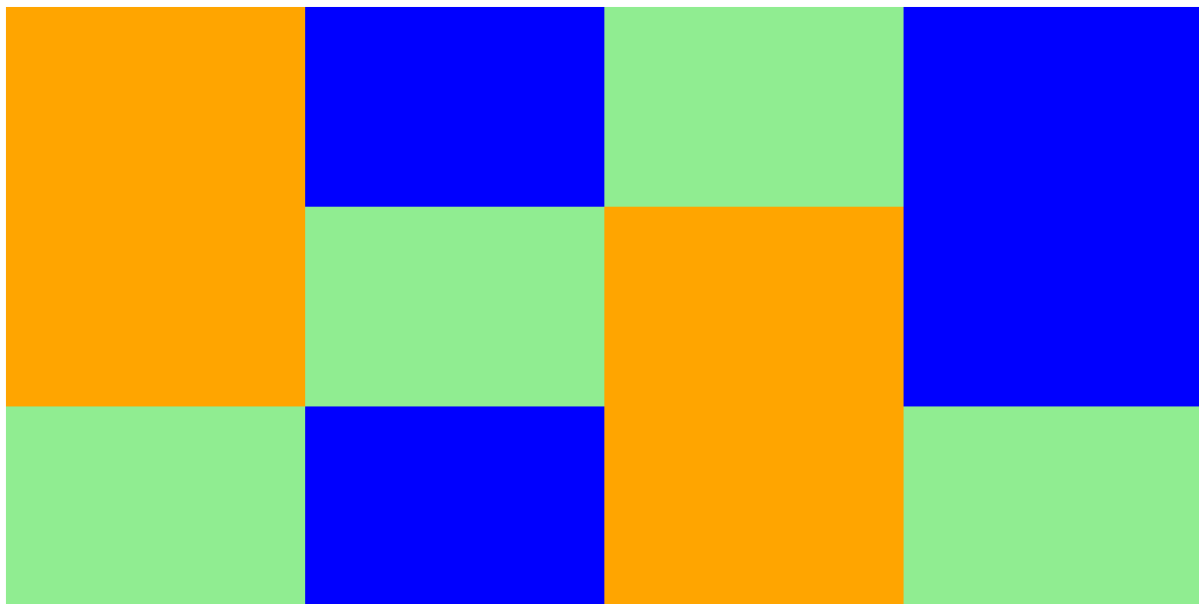
Par défaut, les blocs essaient de rester sur la même ligne s'ils n'ont pas la place. Vous pouvez demander à ce que les blocs aillent à la ligne lorsqu'ils n'ont plus la place, avec **flex-wrap**.

- nowrap : pas de retour à la ligne (par défaut) ;
- wrap : les éléments vont à la ligne lorsqu'il n'y a plus la place ;
- wrap-reverse : les éléments vont à la ligne, lorsqu'il n'y a plus la place, en sens inverse.

Utilisation de Flexbox



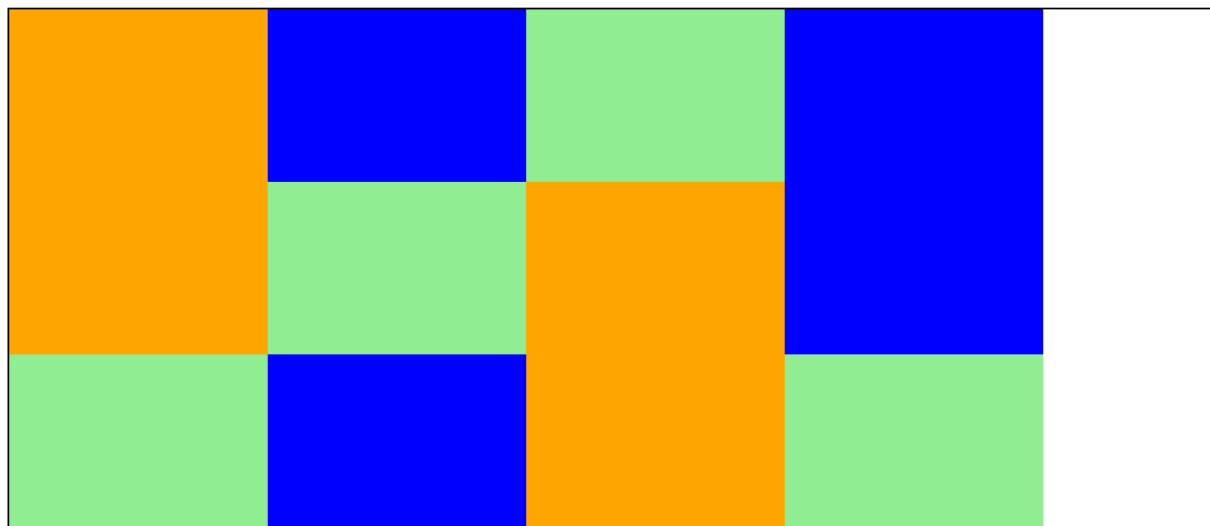
```
.container {  
  display: flex;  
  width: 650px;  
}
```



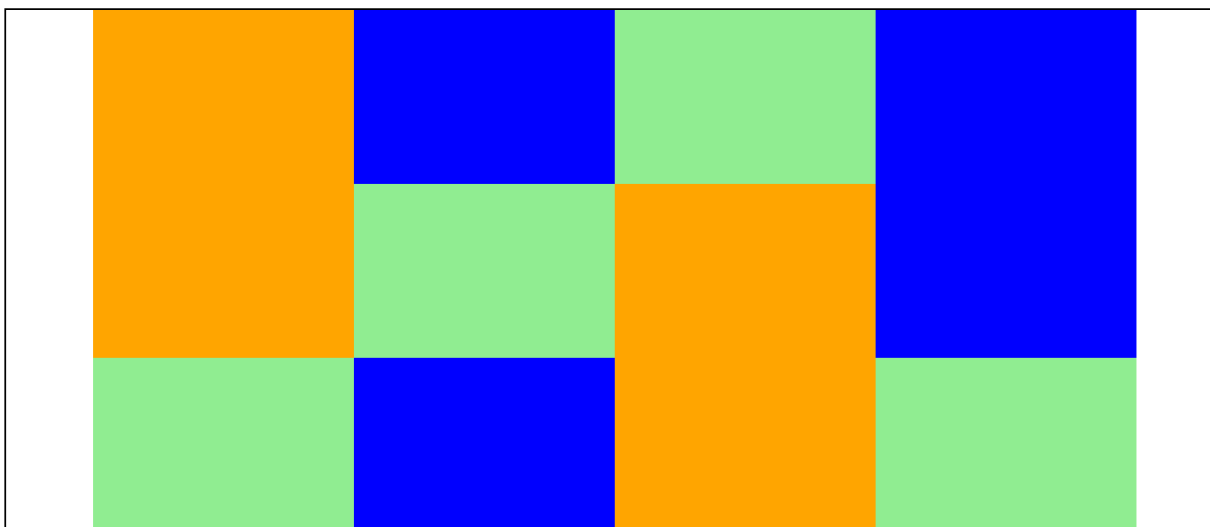
```
.container {  
  display: flex;  
  width: 650px;  
  flex-wrap: wrap;  
}
```

Pour changer l'alignement de l'axe principal, on utilise **justify-content** :

- **flex-start**: alignés au début (par défaut) ;
- **flex-end**: alignés à la fin ;
- **center**: alignés au centre ;
- **space-between**: les éléments sont étirés sur tout l'axe (il y a de l'espace entre eux) ;
- **space-around**: idem, les éléments sont étirés sur tout l'axe, mais ils laissent aussi de l'espace sur les extrémités.



```
.container {  
  display: flex;  
  width: 700px;  
  flex-wrap: wrap;  
  border: 1px solid #000;  
}
```



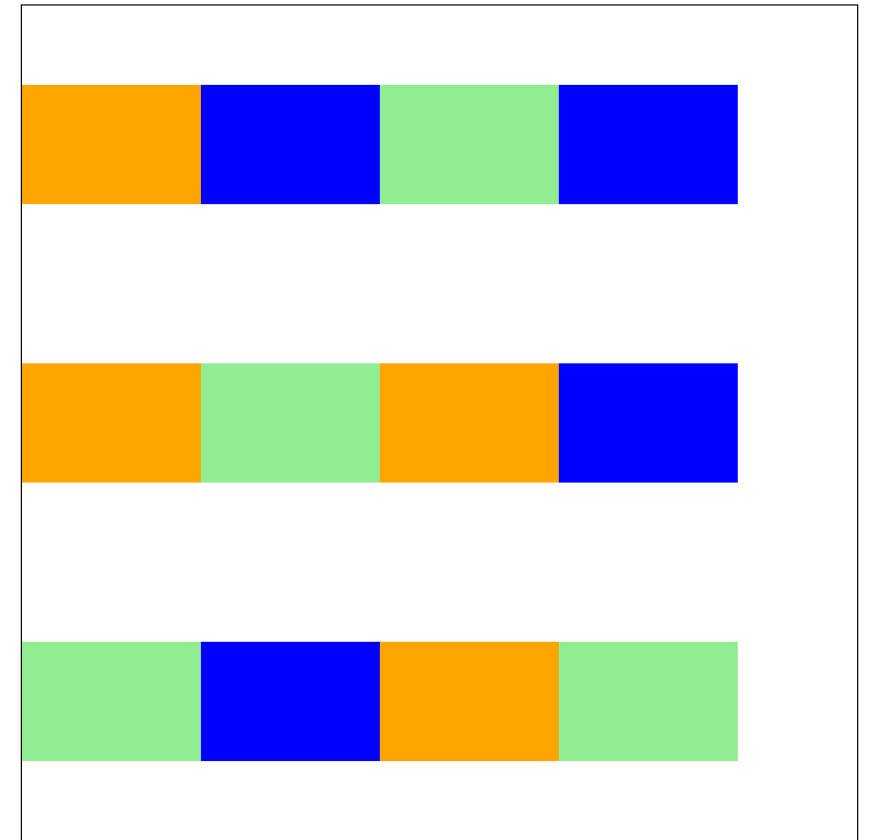
```
.container {  
  display: flex;  
  width: 700px;  
  flex-wrap: wrap;  
  justify-content: center;  
  border: 1px solid #000;  
}
```

Pour changer l'alignement de l'axe secondaire, on utilise :

1. align-items

- S'applique aux éléments individuels sur chaque ligne.
- Gère l'alignement des éléments sur l'axe secondaire, dans leur propre ligne.

```
.container {  
  display: flex;  
  width: 700px;  
  height: 700px;  
  flex-wrap: wrap;  
  border: 1px solid #000;  
  align-items: center;  
}
```

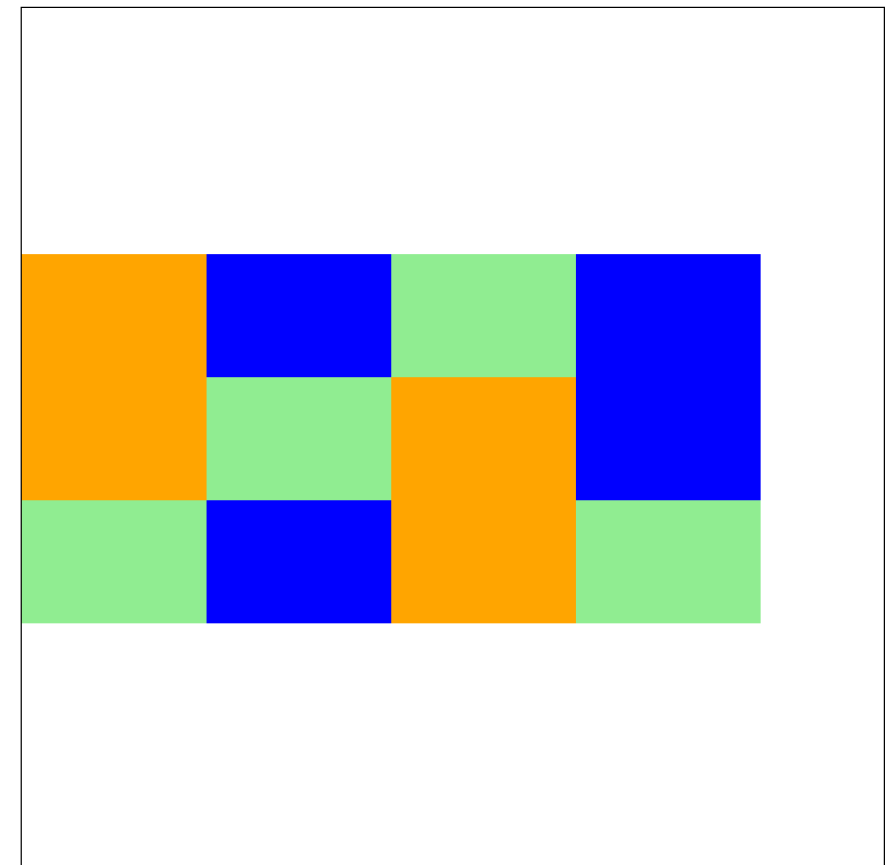


flex-start: alignés au début (par défaut) ;
flex-end: alignés à la fin ;
center: alignés au centre ;

2. align-content

- S'applique aux lignes elles-mêmes dans l'espace global du conteneur.
- Gère l'alignement des **lignes flexibles** sur l'axe transversal.

```
.container {  
  display: flex;  
  width: 700px;  
  height: 700px;  
  flex-wrap: wrap;  
  border: 1px solid #000;  
  align-content: center;  
}
```



```
<header>Bloc HEADER</header>
<div class="main-container">
  <nav>Bloc NAV
    <ul>
      <li> Lien 1 </li>
      <li> Lien 2</li>
      <li> Lien 3 </li>
    </ul>
  </nav>
  <section>
    <h1> Bienvenue sur la page 1 </h1>
    <p> bla bla bla </p>
  </section>
</div>
<footer>Bloc FOOTER</footer>
```

Bloc HEADER

Bloc NAV

- Lien 1
- Lien 2
- Lien 3

Bienvenue sur la page 1
bla bla bla

Bloc FOOTER

```

<header>Bloc HEADER</header>
<div class="main-container">
  <nav>Bloc NAV
    <ul>
      <li> Lien 1 </li>
      <li> Lien 2</li>
      <li> Lien 3 </li>
    </ul>
  </nav>
  <section>
    <h1> Bienvenue sur la page 1 </h1>
    <p> bla bla bla </p>
  </section>
</div>
<footer>Bloc FOOTER</footer>

```

```

header {
  background-color: black;
  color: white;
  text-align: center;
  width:100%;
  height: 100px;
}

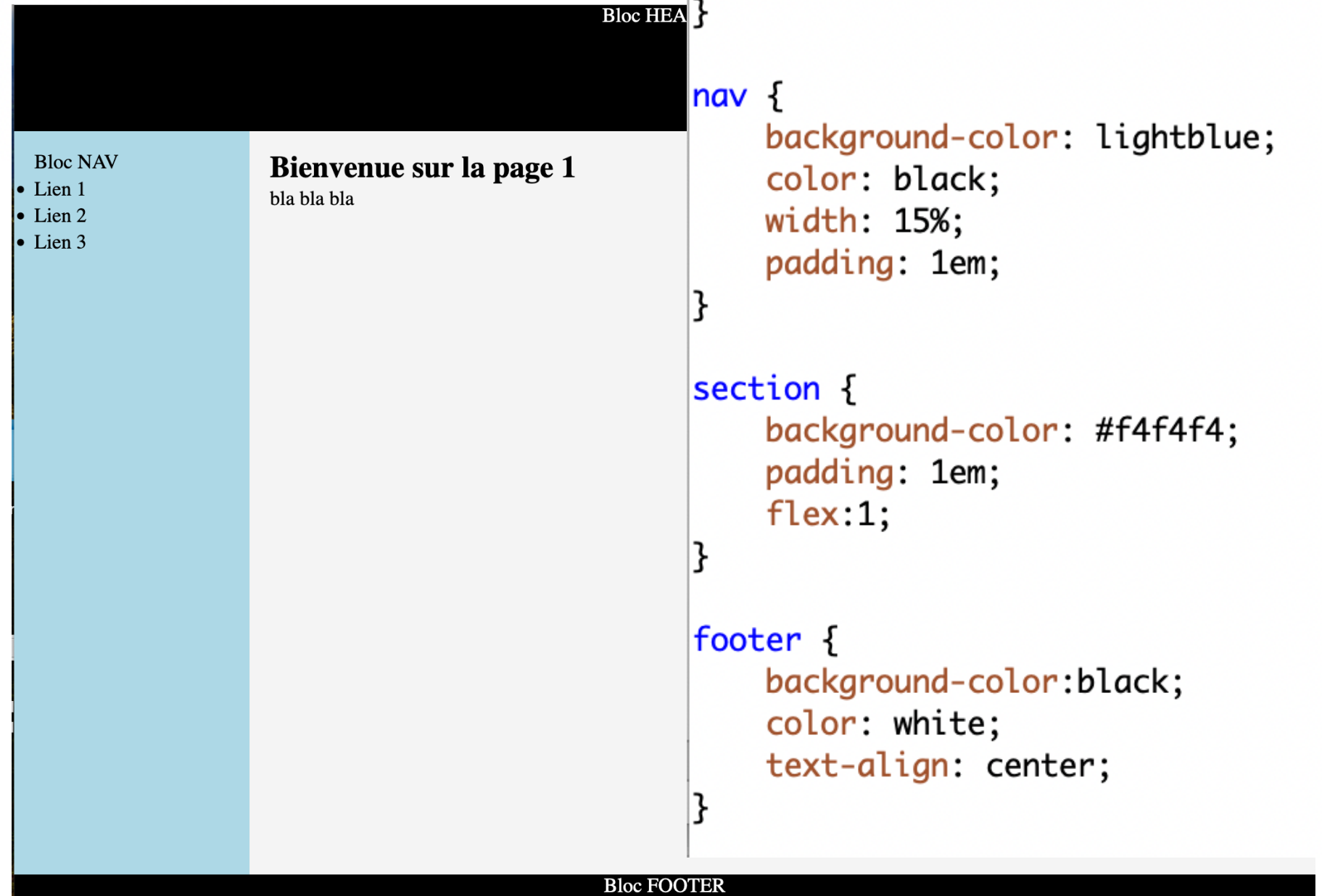
.main-container {
  display: flex;
  height: 80vh;
}

nav {
  background-color: lightblue;
  color: black;
  width: 15%;
  padding: 1em;
}

section {
  background-color: #f4f4f4;
  padding: 1em;
  flex:1;
}

footer {
  background-color:black;
  color: white;
  text-align: center;
}

```



L'unité **vh** est une unité de mesure relative en CSS qui signifie "**Viewport Height**" (hauteur de la fenêtre d'affichage).
1vh = 1% de la hauteur du viewport.

flex:1 signifie tous les éléments se partagent l'espace disponible

.item1 { flex: 1; }
.item2 { flex: 2; } signifie prend 2 fois plus de place que item1

```
header {
  background-color: black;
  color: white;
  text-align: center;
  width: 100%;
  height: 100px;
}

.main-container {
  display: flex;
  height: 80vh;
}

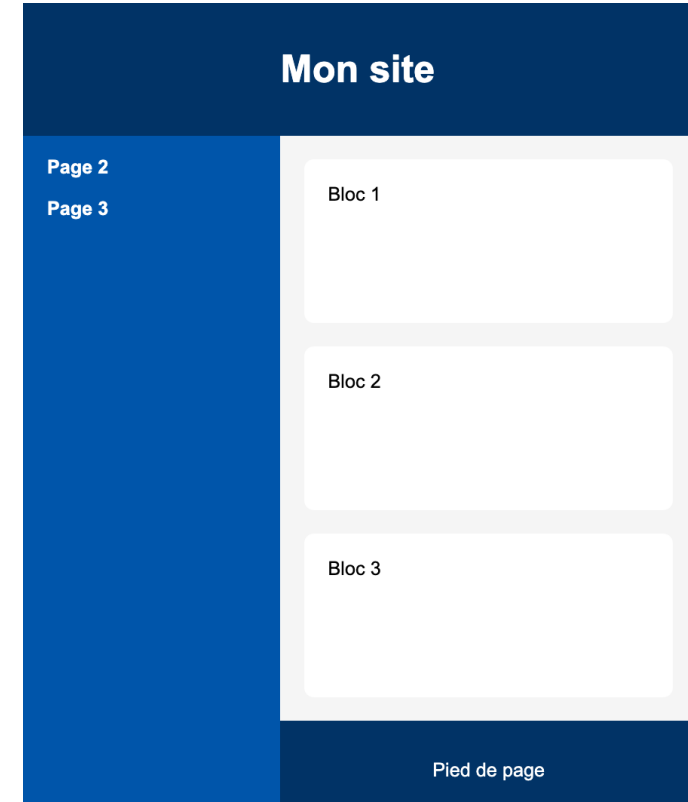
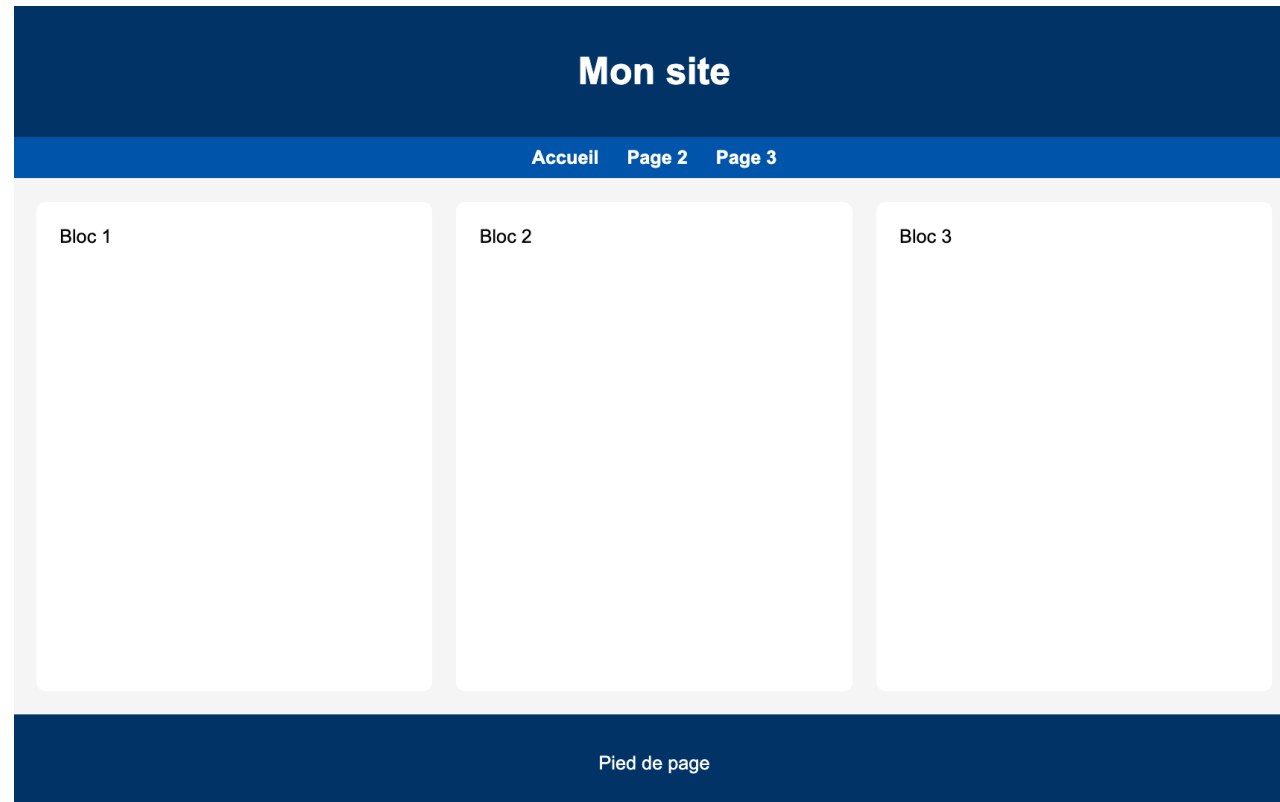
nav {
  background-color: lightblue;
  color: black;
  width: 15%;
  padding: 1em;
}

section {
  background-color: #f4f4f4;
  padding: 1em;
  flex: 1;
}

footer {
  background-color: black;
  color: white;
  text-align: center;
}
```

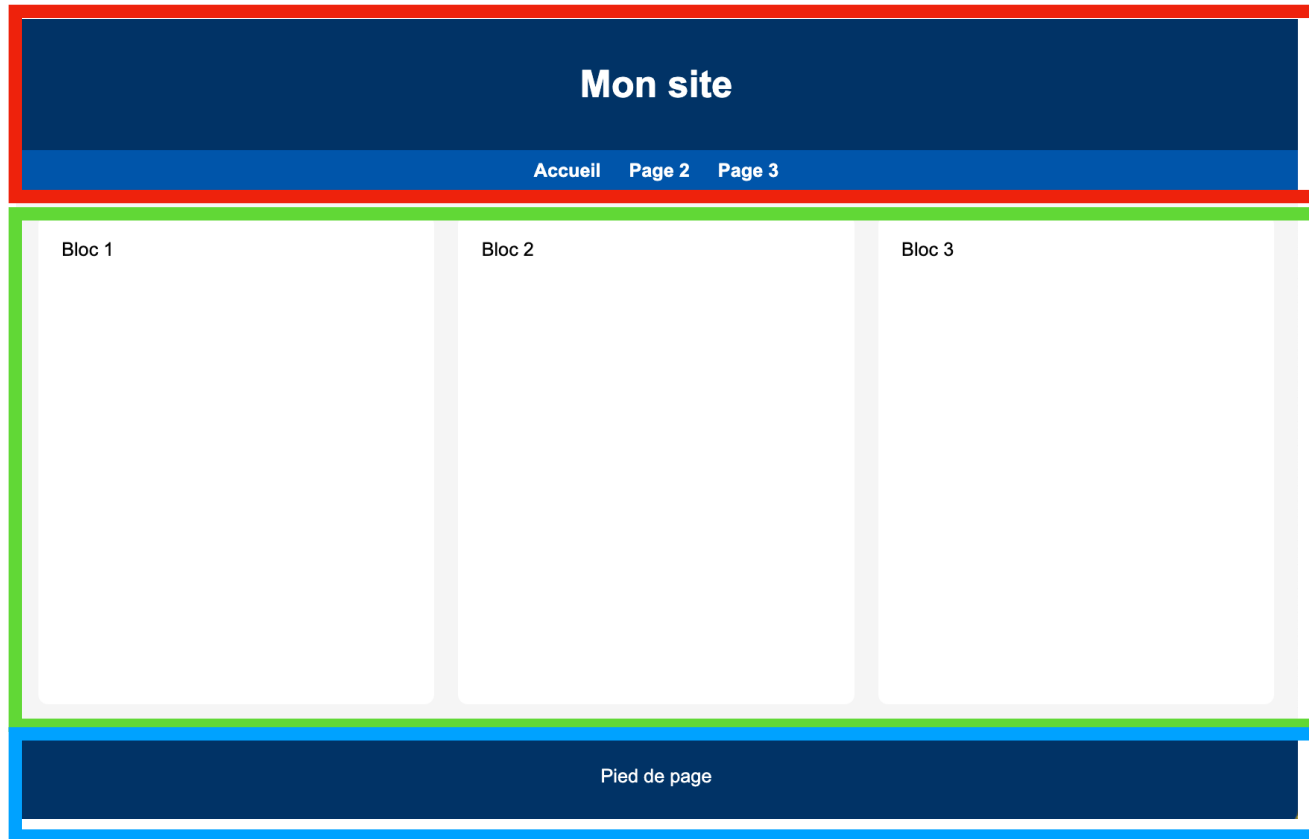
TP1 : Positionnement

Affichage attendu selon la taille de l'écran



Pour réaliser ceci : même fichier HTML

HTML



Mon site

- [Accueil](#)
- [Page 2](#)
- [Page 3](#)

Bloc 1
Bloc 2
Bloc 3

Pied de page

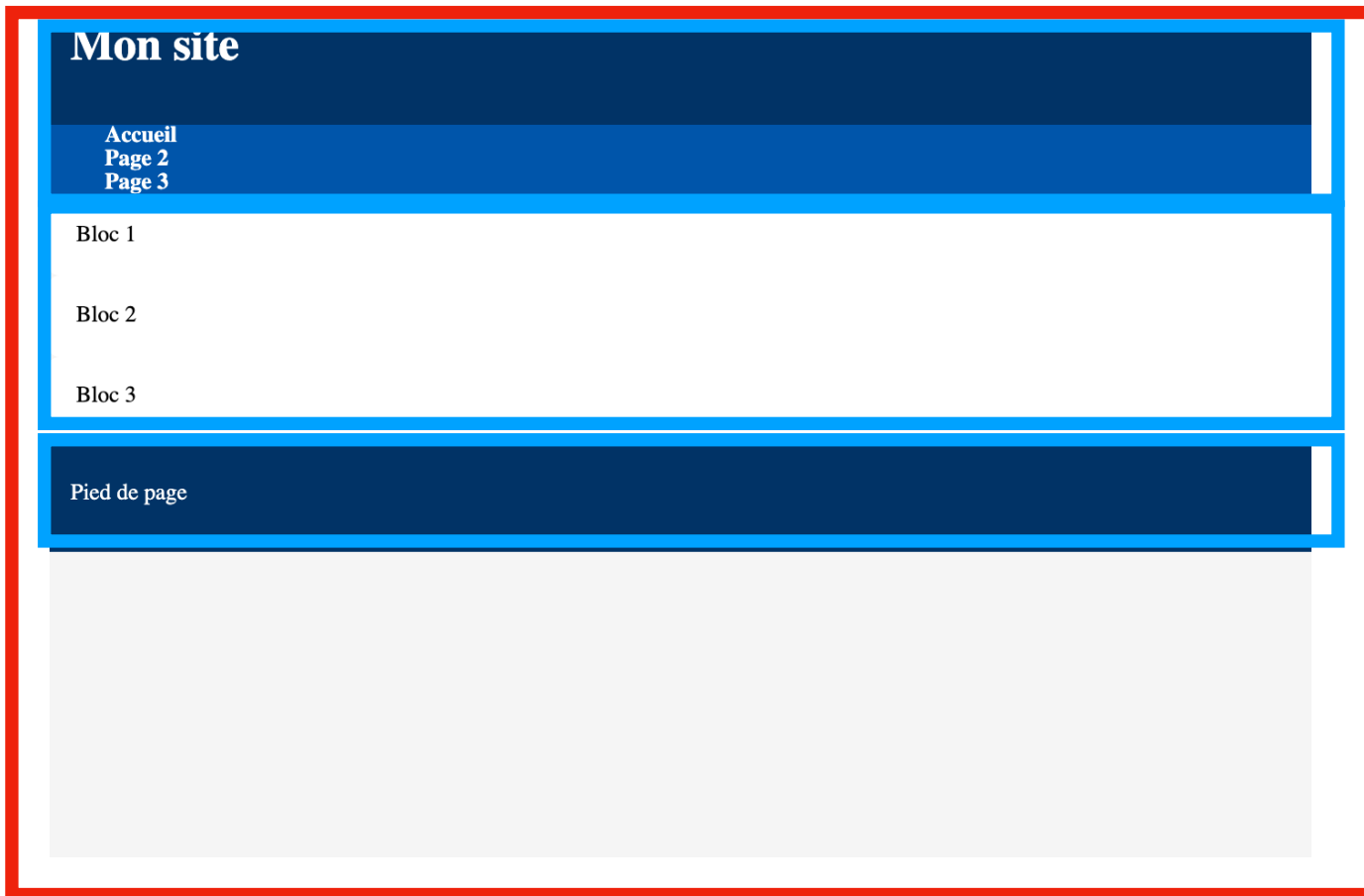
Rendu HTML sans CSS

```
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Mini-site</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
<header>
  <div class="bandeau">
    <h1>Mon site</h1>
  </div>
  <nav>
    <ul>
      <li><a href="index.html">Accueil</a></li>
      <li><a href="page2.html">Page 2</a></li>
      <li><a href="page3.html">Page 3</a></li>
    </ul>
  </nav>
</header>

<main>
  <section class="bloc">Bloc 1</section>
  <section class="bloc">Bloc 2</section>
  <section class="bloc">Bloc 3</section>
</main>

<footer>
  <p>Pied de page</p>
</footer>
</body>
</html>
```

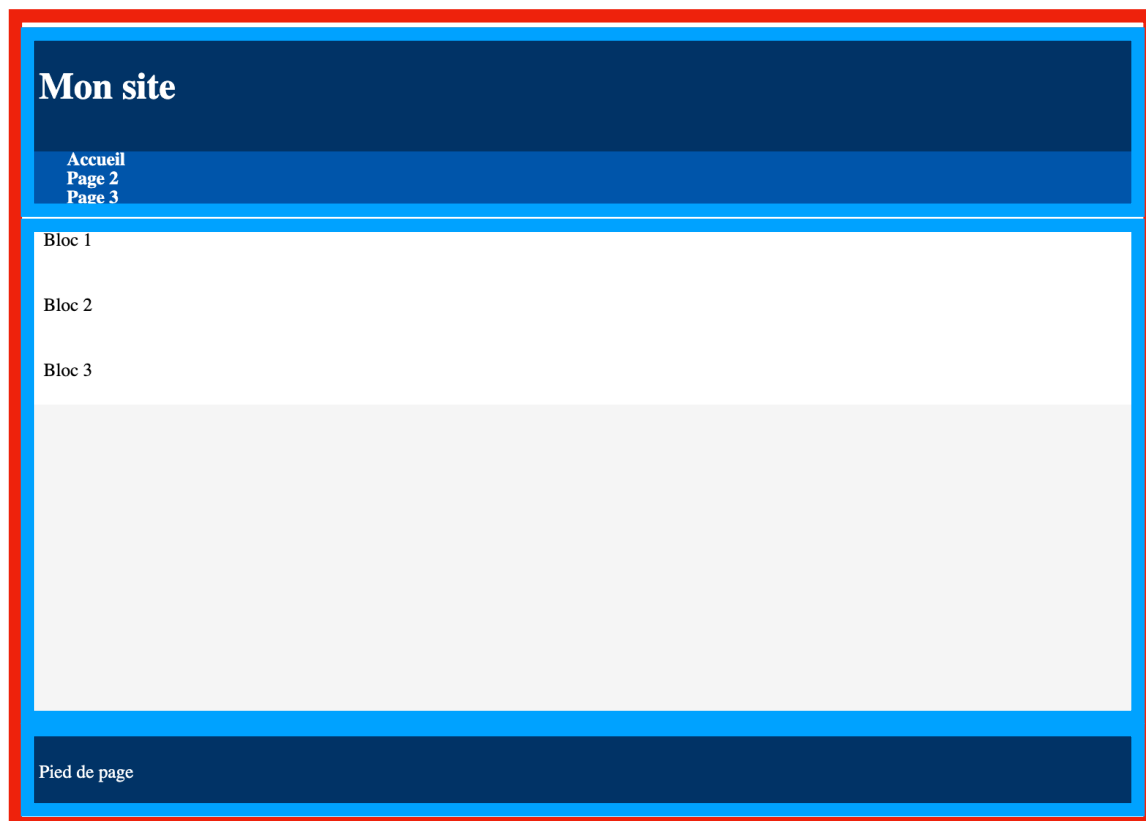
HTML



Balise body : sera le container flex
Avec un alignement vertical (column)

```
body {  
  margin: 0;  
  background: #f5f5f5;  
  min-height: 100vh;  
  display: flex;  
  flex-direction: column;  
}
```

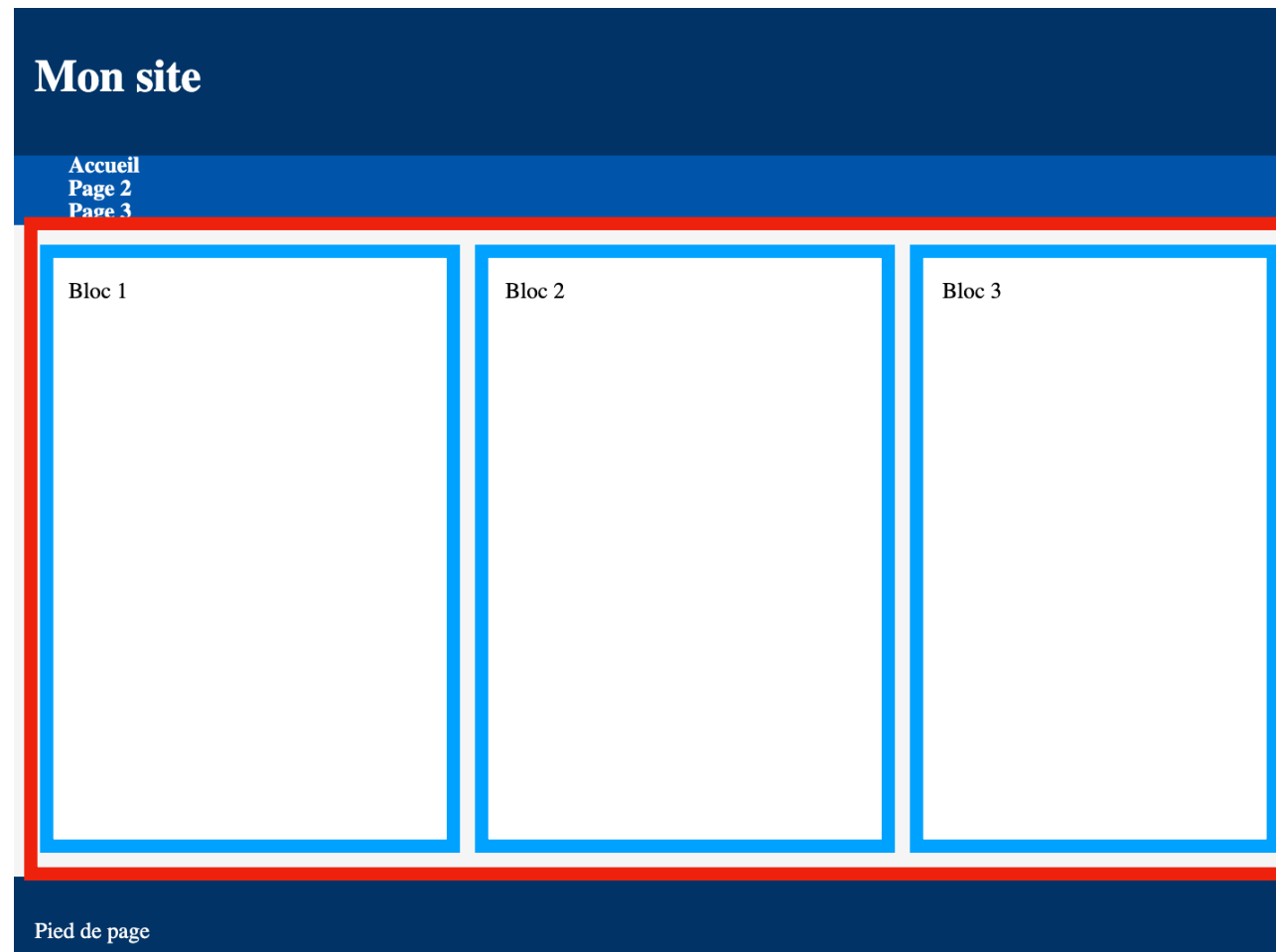
Balises header,
main et
footer
Avec son css pour les couleurs



Ici, la balise main remplit tout
l'espace restant grâce à la
propriété flex=1

```
main {  
  flex: 1;  
}
```

HTML



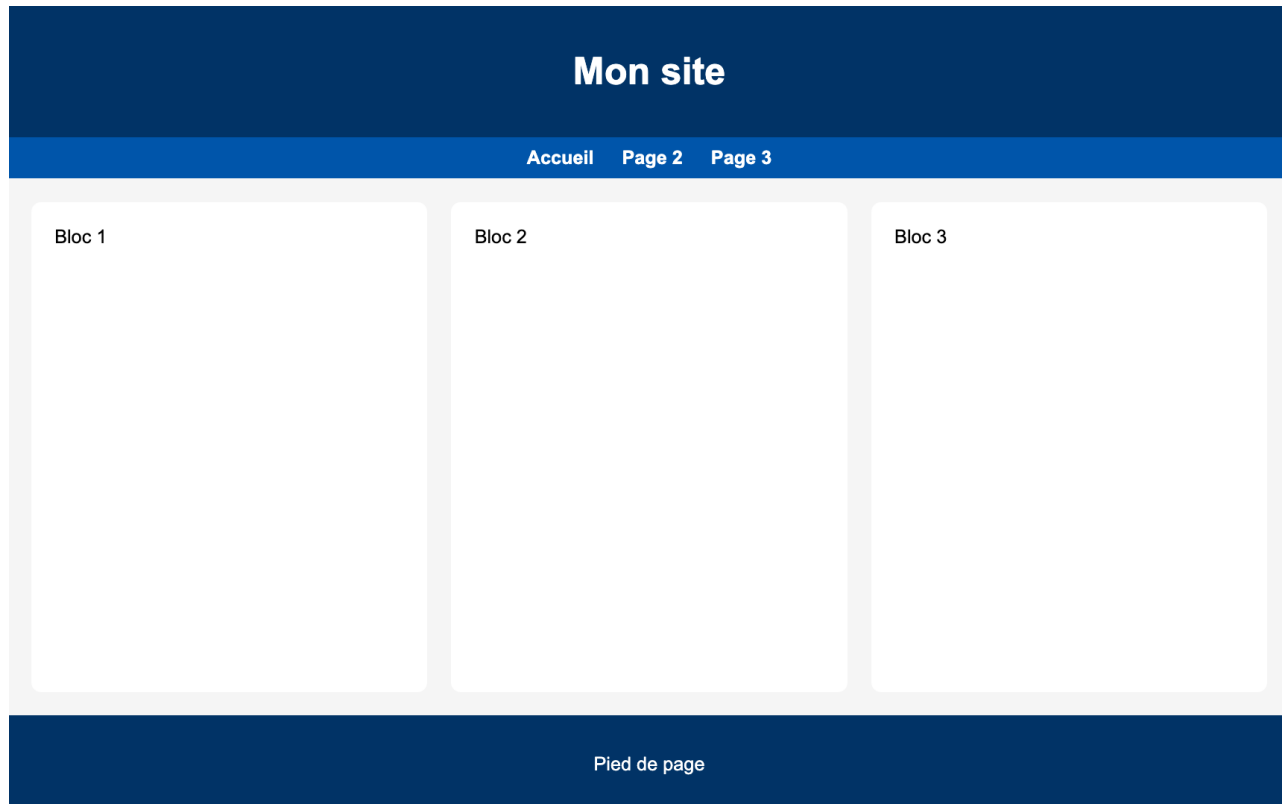
Balise main : sera un 2^e container flex
Avec un alignement horizontal (row)

Balises section de classe bloc

```
main {
  display: flex;
  flex-direction: row;
  align-items: stretch;
  justify-content: center;
  gap: 20px; /*pour définir l'espace entre les blocs */
  padding: 20px;
  flex: 1;
}

.bloc {
  background: white;
  padding: 20px;
  border-radius: 8px; /*pour arrondir les angles*/
  flex:1;
}
```

HTML



Reste du CSS pour centrer les titres...

```
.bandeau {
    background-color: #003366;
    color: white;
    padding: 1em;
}

nav {
    background: #0055aa;
}

nav ul {
    list-style: none;
    margin: 0;
    display: flex;
    justify-content: center;
    gap: 1.5em;
}

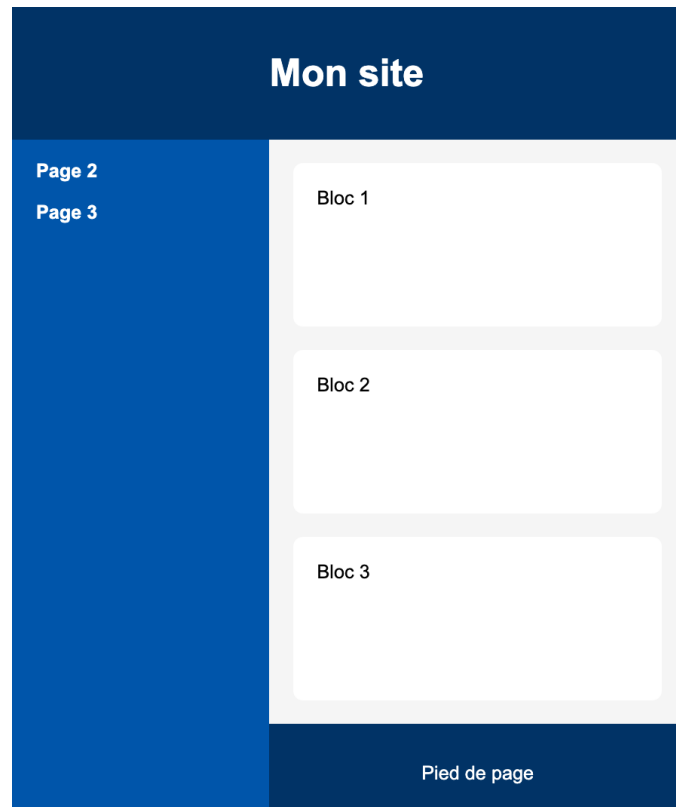
nav a {
    color: white;
    text-decoration: none;
    font-weight: bold;
}

footer {

    background: #003366;
    color: white;
    padding: 1em;
}
```

HTML

Pour un écran <1024px



Utilisation d'une requête média pour définir le nouveau positionnement à ajouter à la suite du fichier css

```
@media (max-width: 1024px) {  
  header {  
    display: flex;  
    flex-direction: column;  
  }  
  
  .bandeau {  
    position: sticky;  
    top: 0;  
    z-index: 10;  
  }  
  
  nav {  
    display: flex;  
    flex-direction: column;  
    width: 220px;  
    background: #0055aa;  
    min-height: calc(100vh - 70px); /* hauteur totale - bandeau */  
    position: fixed;  
    top: 70px; /* hauteur du bandeau */  
    left: 0;  
    padding-top: 1em;  
  }  
  
  nav ul {  
    flex-direction: column;  
    align-items: flex-start;  
    gap: 1em;  
    padding-left: 1.5em;  
  }  
  
  main {  
    margin-left: 220px; /* espace pour le menu */  
    display: flex;  
    flex-direction: column;  
    padding: 20px;  
  }  
  
  footer {  
    margin-left: 220px;  
  }  
}
```