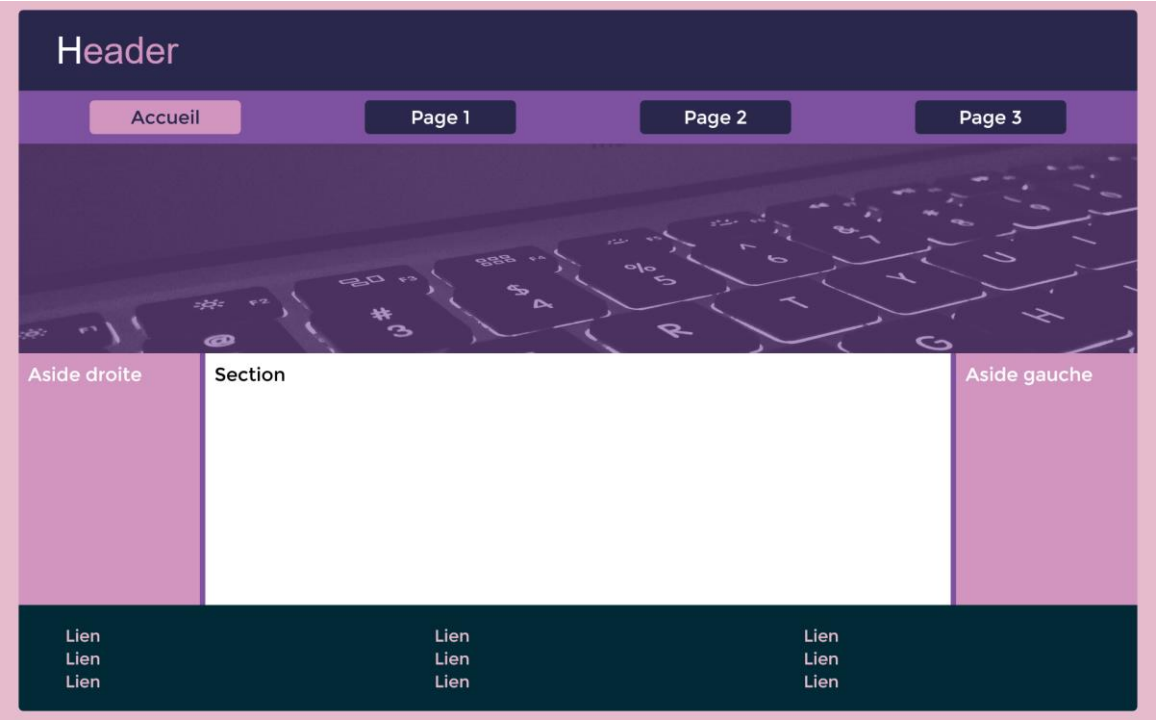
# TP N° 12

## ÉNONCÉ :

Soit la page web de TP précédent réalisée en deux versions : version flexbox et grid (les fichiers de travail sont disponibles dans les dossiers ***version flexbox*** et ***version grid***).




Ce TP consiste à rendre adaptable (au sens de responsive web design) cette page.

## Version grid :

**1.** La page à une largeur fixe exprimée en pixels (960px), elle ne s'adapte donc pas du tout à la taille de la fenêtre du navigateur. Modifiez le code CSS de la page afin de la rendre fluide : largeur exprimée en pourcentage mais bornée (ex : 1200px maximum).

***Étape 1 : On modifie les dimensions du conteneur pour le rendre flexible.***

```
.page {
        width: 100%;
        max-width: 1200px;
        margin: auto;
        display: grid;
        grid-template-columns: 160px auto 160px;
        grid-template-rows: auto auto 180px auto auto;
        grid-template-areas:    "header header header"
                                "menu menu menu"
                                "hero hero hero"
                                "aside1 content aside2"
                                "footer footer footer";
        border-radius: 4px;
        overflow: hidden;
}
```

***Étape 2 : Modification de l'image en bandeau***

```
.home div.hero {
          grid-area: hero;
          width: 100%;
          height: 180px;
          background-image: url("img.jpg");
          position: relative;
          background-size: cover;
          background-position: center center;
}
```

**2.** L'objectif va être de rendre l'interface adaptable (responsive). Préparez les media queries permettant de gérer trois largeurs d'écran :

**a.** Pour un viewport d'une largeur supérieure à 960px (interface par défaut) ;

**b.** Pour un viewport d'une largeur comprise entre 576px et 960px ;

**c.** Pour un viewport d'une largeur inférieure à 576px.

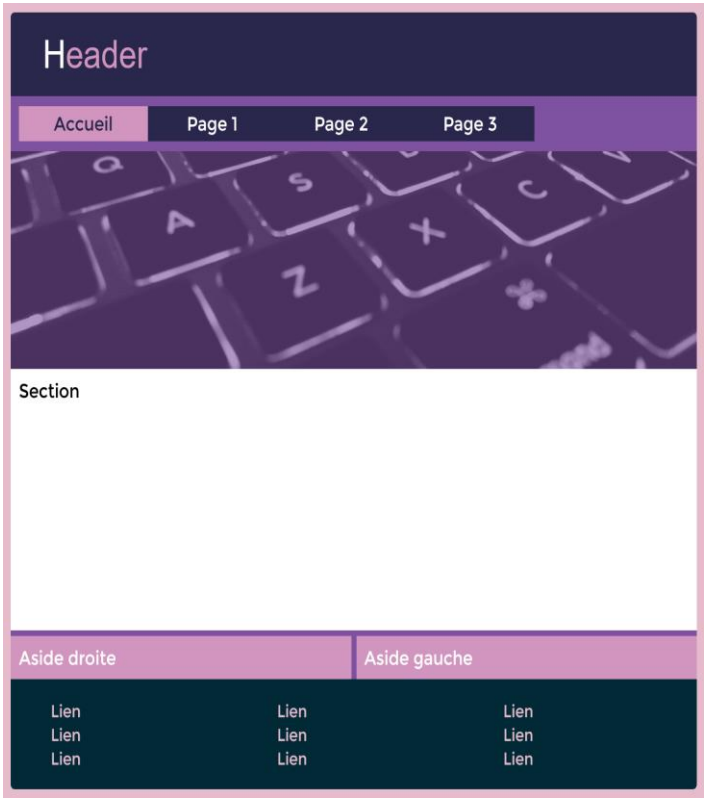
**NB :** Nous n'allons pas prévoir de **media queries** pour la taille par défaut, mais uniquement pour les deux autres largeurs d'écran. Il suffira alors d'apporter les modifications entre l'interface par défaut et la largeur d'écran définie par les media queries.

```
/* Paramètres pour les écrans avec une largeur inférieure à 576px */
@media screen and (max-width: 575px) {

}

/* Paramètres pour les écrans d'une largeur comprise entre à 577px et 960px */
@media screen and (min-width: 576px) and (max-width: 960px) {
}
```

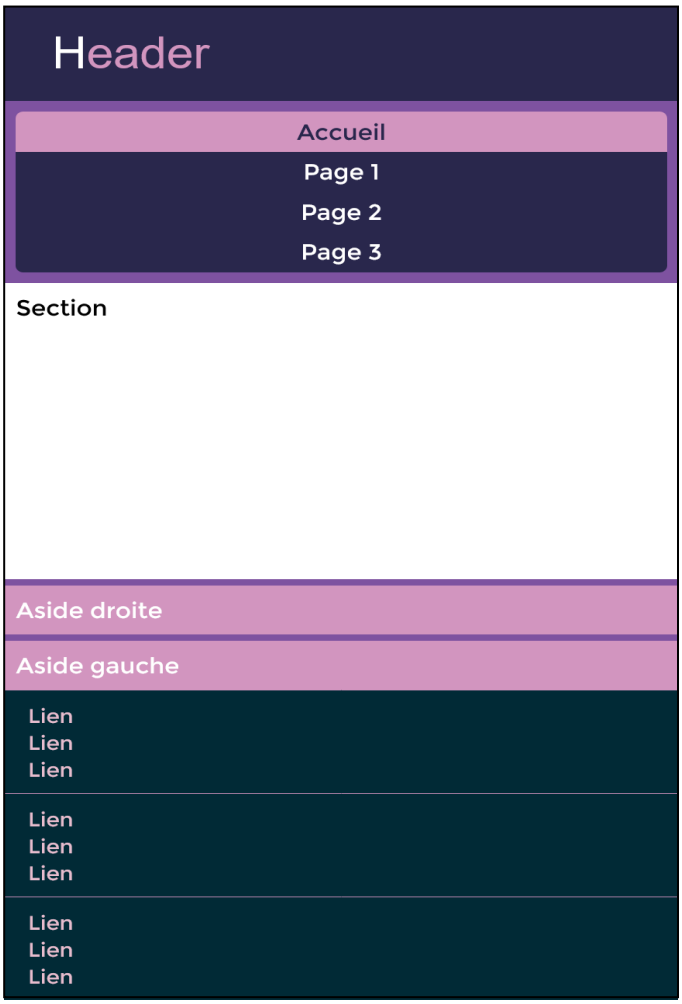
***3.*** Ajustez les paramètres de l'interface afin qu'elle corresponde à l'image ci-dessous pour les écrans ayant une largeur comprise entre 576px et 960px.




```
@media screen and (min-width: 576px) and (max-width: 960px) {
        /* Modification de la grille pour repositionner les éléments */
        .page {
                grid-template-columns: 1fr 1fr;
                grid-template-rows: auto auto 180px auto auto auto;
                grid-template-areas: "header header"
                                     "menu menu"
                                     "hero hero"
                                     "content content"
                                     "aside1 aside2"
                                     "footer footer";
        }
        /* Modification du menu */
        nav.main-nav ul {
                justify-content: flex-start;
        }
```

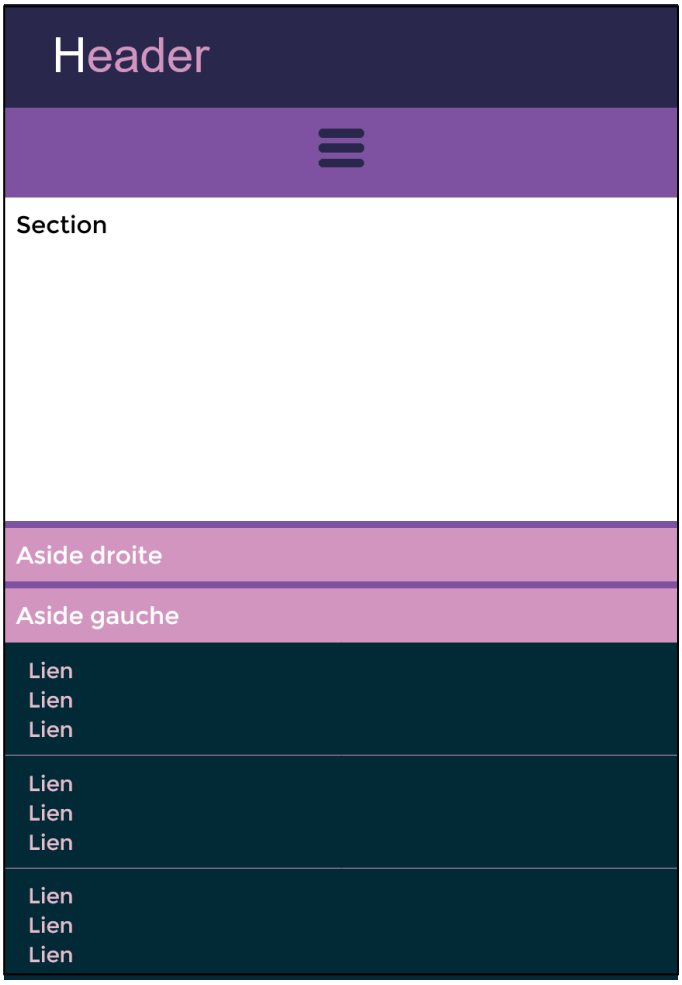
```
        nav.main-nav a {
                border-radius: 0px;
        }
        nav.main-nav li:first-of-type {
                border-radius: 5px 0 0 5px;
        }
        nav.main-nav li:last-of-type {
                border-radius: 0 5px 5px 0;
        }
        /* Modification des bordures pour les deux éléments aside */
        aside.main_aside_left,  aside.main_aside_right {
                border: 5px solid #7e52a0;
        }
        aside.main_aside_left {
                border-width: 5px 3px 0 0;
        }
        aside.main_aside_right {
                border-width: 5px 0 0 3px;
        }
}
```

***4.*** Ajustez les paramètres de l'interface afin qu'elle corresponde à l'image ci-dessous pour les écrans ayant une largeur inférieure à 576px.



```
@media screen and (max-width: 575px) {
        body {
                margin: 0;
        }
        .page {
                border-radius: 0px;
                grid-template-columns: 1fr;
                grid-template-rows: repeat(6, auto);
                grid-template-areas: "header "
                                     "menu"
                                     "content"
                                     "aside1"
                                     "aside2"
                                     "footer";
        }
        nav.main-nav ul {
                flex-direction: column;
        }
        nav.main-nav a {
                border-radius: 0px;
        }
        nav.main-nav li:first-of-type a {
                border-radius: 5px 5px 0 0;
        }
        nav.main-nav li:last-of-type a {
                border-radius: 0 0 5px 5px;
        }
        .home div.hero {
                background: none;
                display: none;
        }
        aside.main_aside_left, aside.main_aside_right {
                border: 5px solid #7e52a0;
                border-width: 5px 0 0 0;
        }
        footer.main_footer ul {
                margin: 0;
                padding: .5em 1em;
                border-top: 1px solid  #d295bf;
                width: 100%;
        }
}
```

***5.*** Modifiez le code afin d'ajouter un bouton hamburger qui permettra d'afficher le menu pour les écrans ayant une largeur inférieure à 576px. Le bouton hamburger sera créé à partir du label d'une case à cocher qui déterminera si le menu est visible ou pas.




***Étape 1 : Mise en place du bouton***

Le bouton hamburger va être créé à partir du label associé à la checkbox. Pour associer un label à une checkbox, il est nécessaire d'ajouter un id à l'input et un for correspondant au label. Ainsi, il sera possible de cocher la case en cliquant sur le label en plus de la case. Le span ajouté à l'intérieur du label sera utilisé pour créer le trait central du bouton.

```
<nav class="main-nav">
        <input type="checkbox" class="btn-hamburger" id="btn-hamburger">
        <label for="btn-hamburger" class="btn-hamburger-label">
                <span></span>
        </label>
        <div>
                <ul>
                        <li><a href="#" class="active">Accueil</a></li>
                        <li><a href="#">Page 1</a></li>
                        <li><a href="#">Page 2</a></li>
                        <li><a href="#">Page 3</a></li>
                </ul>
        </div>
</nav>
```

### *Étape 2 : Mise en forme du bouton*

Nous commençons par masquer la case à cocher d'origine. Puis nous allons créer le bouton hamburger. Celui-ci se compose de trois traits. Les trois traits sont créés à partir du span présent dans le label et des deux pseudo-éléments ::before et ::after.

```
/* Masquage de la case à cocher */
input.btn-hamburger {
        display: none;
}
/* Création des trois traits du bouton de menu */
.btn-hamburger-label {
        width: 30px;
        height: 30px;
        margin: 10px auto 10px;
        padding-top: 2px;
        display: flex;
        justify-content: space-between;
        flex-direction: column;
        cursor: pointer;
}
.btn-hamburger-label span, .btn-hamburger-label::before,.btn-hamburger-
label::after {
        content: "";
        display: block;
        width: 100%;
        height: 7px;
        margin: 2px 0;
        background-color: #29274c;
        border-radius: 10px;
}
```

### *Étape 3 : Animation du menu*

L'état affiché ou masqué du menu est déterminé par la pseudo-classe :checked de l'élément input. Dans notre exemple, allons donc afficher l'élément div contenant notre menu en manipulant sa hauteur. Le problème est que CSS ne permet pas d'effectuer une transition entre height: 0 et height: auto. Nous allons donc utiliser max-height en lui attribuant une valeur qui doit être supérieure à la hauteur maximale du menu ouvert.

```
input.btn-hamburger {
        display: none;
}
input.btn-hamburger ~ div {
        transition: all 1s;
        overflow: hidden;
        max-height: 0px;
}
input.btn-hamburger:checked ~ div {
        max-height: 200px;
}
```

6. Prévoyez une version pour l'impression centrée sur le contenu et dépourvue des éléments suivants : menu, image hero, asides, footer.

```
@media print {
        .page {
                grid-template-columns: 1fr ;
                grid-template-rows: auto auto;
                grid-template-areas: "header"
                                     "content";
        }
}
```

## Version flexbox :

En exploitant les fichiers de travail de la version flexbox faites le nécessaire pour avoir le même résultat.