

```
cd /home/event/discovery_piscine
mkdir cell16
cd cell16
mkdir ex06
cd ex06
```

```
cp /home/event/discovery_piscine/cell15/ex05/tab.html responsive.html
```

```
vim responsive.html
```

```
i
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Responsive</title>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>

      // on sélectionne l'élément html et l'élément body
      // ATTENTION à bien séparer html et body par une virgule
      // pour indiquer que l'on sélectionne l'élément html ET l'élément body
      // (pour que les règles suivantes soient appliquées à html ET à body)
      // ne pas confondre avec html body qui signifie "l'élément body DANS l'élément html"
      // car cela appliquera les règles suivantes seulement à l'élément body dans ce cas

      html, body {

        // on indique une hauteur de 100% pour indiquer que l'on veut travailler avec
        // toute la hauteur de la fenêtre
        // cela est nécessaire car la hauteur de html et de body n'est pas définie par défaut
        // (elle s'adapte au contenu normalement)
        // sans cette règle sur html et body, la règle height: 100% sur table (plus bas)
        // ne saura pas à quelle hauteur
        // se référer (car height: 100% veut dire 100% de la hauteur de l'élément parent)
        // et donc, si la hauteur de body n'est pas définie, son enfant table n'aura pas
```

```
// de référence pour la hauteur
// remarque : cela n'est pas nécessaire d'indiquer width: 100% pour body et html
// car, comme ces éléments sont de type block, ils occupent déjà par défaut
// toute la largeur disponible du conteneur parent (et le
// conteneur parent de html est la fenêtre du navigateur)

    height: 100%;
}

body {

    // on supprime les marges par défaut
    // (la plupart des navigateurs ajoutent des petites marges extérieure (margin)
    // autour de l'élément body)
    // on la remet donc ainsi à 0 si tel est le cas)

    margin: 0;
}

table {

    // on indique cette propriété pour que les bordures fassent partie des 100%
    // de la largeur et de la hauteur
    // sans cette propriété, la hauteur sera de 100% + 2px
    // (1px en haut et 1px en bas),
    // idem pour la largeur
    // il y aurait donc débordement du tableau de la fenêtre

    box-sizing: border-box;
    border-collapse: collapse;
    border: 1px solid black;

    // on indique que le tableau devra couvrir 100% de la hauteur de body,
    // qui couvre 100% de la hauteur de html
    // (voir plus haut), qui couvre lui-même 100% de la hauteur de la fenêtre
    // (voir plus haut)

    height: 100%;
```

```

        // on indique que le tableau devra couvrir 100% de la largeur de body,
        // qui couvre (par défaut) 100% de la largeur de html, qui couvre lui-même
        // (par défaut) 100% de la largeur de la fenêtre

        width: 100%;

    }

    td {

        vertical-align: middle;

        // on indique cette propriété ici aussi
        // pour que les bordures fassent partie des 50% de la largeur et de la hauteur
        // (la largeur serait de 50% + 1px sinon)

        box-sizing: border-box;
        width: 50%;

    }

    tr td:nth-child(1) {
        background-color: darkkhaki;
        border-right: 1px solid grey;
    }

    tr td:nth-child(2) {
        background-color: lightpink;
        border-left: 1px solid grey;
    }
}
</style>
</head>
<body>
    <table>
        <tr>
            <td>First cell</td>
            <td>Second cell</td>
        </tr>
    </table>

```

```
        </body>  
</html>
```

```
echap  
:wq  
enter
```

```
firefox responsive.html
```