

```
cd /home/event/discovery_piscine
mkdir cell12
cd cell12
mkdir ex02
cd ex02

cp /home/event/discovery_piscine/cell11/ex01/piscineweb.html piscineweb.html

vim piscineweb.html
i
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Piscine Web</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 écrit les règles CSS ici pour plus de lisibilité, au lieu de l'écrire en attribut
      // de l'élément visé, en mode inline (en ligne)
      // (pour toutes les regrouper au même endroit), et pour pouvoir appliquer plus
      // facilement un même style à plusieurs éléments à la fois
      // (grâce à l'utilisation de sélecteurs) sans avoir à dupliquer les déclarations CSS

      // cette façon d'appliquer du CSS à un élément est dite "interne" (on utilise une
      // feuille de style interne), car les règles CSS sont écrites dans le document html
      // ici, le sélecteur est h1
      // la règle "text-align: center;" sera appliquée à tous éléments h1 de ce document html
      // la ou les règle(s) appliquée(s) à l'(aux) élément(s) sélectionné(s) est (sont)
      // indiquée(s) entre accolades

      h1 {
        text-align: center;
      }
      // le sélecteur est div
      // le style sera appliqué à l'élément div
      // ce container contiendra les 2 images (elle les encapsulera)
      // nous pourrons ainsi aligner facilement les éléments de ce container (les images)
```

```

        // dans celui-ci

        div {
            text-align: center;
        }

        // on aligne les images (horizontalement) au milieu de la page

        // par défaut, les images sont des éléments de type inline, elles seront donc traitées
        // comme du texte : si un élément img se place après un autre élément img, comme c'est
        // le cas ici (voir plus bas), ce dernier se placera à droite du premier, à la suite
        // cela s'explique par le fait qu'un élément de type inline a une largeur égale à son
        // contenu
        // alors que les éléments de type block ont une largeur de 100% par défaut (ils prennent
        // toute la largeur disponible)

        // avec cette règle, on indique donc au contenu texte de la div
        // (les balises img donc, qui sont considérées comme du texte) de s'aligner au centre
    </style>        // on ferme la balise style
</head>
<body>
    <h1>Piscine Web</h1>
    <div>    // div : balise générique de type bloc ; commence sur une nouvelle ligne et s'étend
            // sur toute la largeur disponible par défaut
            // cette balise est souvent utilisée comme un container
            // (pour contenir d'autres éléments)
            // on peut ainsi appliquer des styles à ce container
            // pour qu'ils soient appliqués à
            // tous ses éléments enfants, redimensionner ce bloc, lui appliquer
            // une largeur et une hauteur précise, une couleur de fond (background), etc.

            
            

            // la balise img (image) indique une image
            // c'est une balise orpheline (pas besoin de balise fermante)
            // on indique l'URL de l'image que l'on souhaite afficher comme valeur
            // de l'attribut src (source)

```

```

// on peut aussi indiquer le chemin vers une image stockée localement
// si l'image est dans le même répertoire que la fichier html, qui la contient,
// on indique directement le nom du fichier
// sinon, on indique le chemin à partir du répertoire courant
// (chemin relatif, par exemple "assets/img/img1.jpeg")
// ou à partir de la racine (/home/event/discovery_piscine/img/img1.jpeg par exemple)
// (chemin absolu)

// l'attribut title contient le titre de l'image : il doit donner plus de contexte
// (ou des informations complémentaires)
// sur l'image ; il sera affiché lorsque l'utilisateur passera le curseur de sa souris
// sur l'image, dans une petite infobulle
// l'attribut alt (alternative text) fournit une description textuelle de l'image
// cet attribut est important pour l'accessibilité
// il décrit l'image aux utilisateurs qui ne peuvent pas la voir
// (le texte pourra être lu par un lecteur d'écran par exemple)
// ou dans les cas où l'image ne peut pas être chargée
// l'attribut title est facultatif mais l'attribut alt est OBLIGATOIRE,
// même s'il est laissé vide
// COMPORTEMENTS INNATENDUS POSSIBLES DANS CERTAINS CAS SI L'ATTRIBUT ALT EST ABSENT !
// il est aussi important d'indiquer la largeur et la hauteur originale de l'image
// (du fichier .jpg par exemple)
// dans les attributs width (pour la largeur) et height (pour la hauteur)
// même s'ils sont modifiés par la suite en CSS
// aller dans les propriétés de l'image pour connaître ses dimensions et les indiquer ici
</div> // on ferme le container
</body>
</html>

```

```

echap
:wq
enter

```

```

firefox piscineweb.html

```