

## Tracer les fractales

### L'ensemble de Mandelbrot

Pour travailler sur l'ensemble de Mandelbrot, il faut travailler sur les nombres complexes. Python peut le faire nativement :

```
>>> z = complex(1, 2)  # Définition d'un nombre complexe
>>> z
(1+2j)                # j remplace i dans la notation
>>> z * z               # On peut faire des calculs
(-3+4j)
>>> abs(z)              # Calcul de la norme
2.23606797749979
```

**EXERCICE 1 :** Écrire une fonction `bornee` qui prend un nombre complexe `c` et un entier `iter_max` et qui renvoie un booléen indiquant si la suite associée à `c` a un terme dont le module dépasse 2 pour un rang inférieur à `iter_max`.

```
>>> bornee(complex(1, 1))
False
>>> bornee(complex(1, 0))
False
>>> bornee(complex(-1, 0))
True
>>> bornee(complex(-.2, .4))
True
>>> bornee(complex(-.4, .5))
True
>>> bornee(complex(-.4, .8))
False
>>> bornee(complex(.3, 0))
False
```

**EXERCICE 2 :** Compléter le code de la fonction `affiche` qui prend en entrée les valeurs `xmin`, `xmax`, `ymin`, `ymax`, `pasx` et `pasy` et qui affiche l'ensemble de Mandelbrot pour les points  $M(x;y)$  tels que  $xmin \leq x \leq xmax$  et  $ymin \leq y \leq ymax$ . Les valeurs de  $x$  partent de `xmin` et vont de `pasx` en `pasx`. Les valeurs de  $y$  vont de `ymax` et diminuent de `pasy` à chaque étape. Si le point appartient à l'ensemble on affiche '\*'. Sinon on affiche un espace ' '.

```
def afficher(xmin=-1.6, xmax=0.6, ymin=-1, ymax=1, pasx=.025, pasy=0.05):
    y = ...
    while ...:
        x = ...
        ligne = ""
        while ...:
            if bornee(...):
                ligne += ...
            else:
                ligne += ...
        x = ...
        print(ligne)
    y = ...
```

```
afficher(xmin=-1.6, xmax=0.6, ymin=-.9, ymax=.9, pasx=.025, pasy=0.05)
```

**EXERCICE 3 :** Pour voir un peu comment la suite  $(z_n)$  se comporte en fonction des valeurs de  $c$ , vous pouvez utiliser ces pages :

- <https://www.geogebra.org/m/BUVhcRSv#material/Npd3kBKn>
- <https://www.geogebra.org/m/BUVhcRSv#material/XQprvGbW>

**EXERCICE 4 :** Pour faire de vraies images, vous pouvez utiliser le fichier `mandelbrot_final.py` qui permet de générer des images en couleur. Vous pouvez explorer l'ensemble en choisissant les coordonnées du centre de l'image et le niveau de zoom. Vous pouvez même faire des animations. :w

Vous pouvez utiliser les touches suivants :

- “+” et “-” pour zoomer ou dézoomer.
- “↑” et “↓” pour doubler ou diviser par 2 la limite.
- “r” pour revenir au début.
- “s” pour enregistrer l'image.
- Vous pouvez zoomer sur l'image en sélectionnant un rectangle avec le clic gauche de la souris.

Vous pouvez aussi utiliser Xaos pour explorer l'ensemble et ensuite utiliser le programme Python pour faire de jolies images. Il faut afficher “l'état” pour connaître le centre et le niveau de zoom. Le centre est donné par `Vue` et `calcule_zoom(Taille)` pour le zoom.