

Projet python : collier de perles

Le but du projet est d'écrire un programme en turtle qui dessine un collier de perles.

0 Consignes de rendu

Ce projet individuel est à rendre pour le

Les séances
sont prévues pour travailler le projet.

Vous le mettrez dans le dossier ramassage sur une machine, et il sera ramassé automatiquement. En cas d'absence ou d'oubli, vous pouvez envoyer le projet par pronote, avant le soir à minuit.

Si vous n'avez qu'un seul fichier python, veillez bien à mettre vos nom et prénom dans le nom du fichier ! Si vous avez plus d'un fichier python, mettez-les dans un dossier, et archivez l'ensemble dans une archive zip (qui portera vos nom et prénom).

Le respect des consignes compte pour une part importante de la note du projet, mais tout ce qui n'est pas explicitement obligatoire est libre (style, ...). Les commentaires sont indispensables pour comprendre votre code : que dessine ce bloc de code ? À quoi sert cette variable ? Etc.

Tous les exercices sont à écrire sous forme de fonction, qui ne renvoient rien mais qui dessinent à l'écran.

Si vous souhaitez modifier les spécifications d'une fonction pour une *bonne* raison, expliquez bien dans le commentaire de votre code la raison de ce choix. Dans le doute, demandez à votre professeur(e).

*La lisibilité du code est également une part importante dans la notation : mettez des **commentaires**, utilisez des boucles et des variables pour ne pas recopier des valeurs partout, mettez des **commentaires**, utilisez des fonctions si cela vous paraît pertinent, et surtout mettez des **commentaires**. Et n'oubliez pas de **commenter** votre code.*

1 La signature (3)

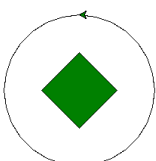
DEMA

Écrire une fonction signature qui ne prend pas d'argument, et qui « signe » votre dessin : la tortue doit dessiner les deux premières lettres de votre prénom et les deux premières lettres de votre nom de famille.

La couleur et le style sont libres, et cette fonction ne renvoie rien. Vous avez un exemple ci-contre.

Remarque : vous ne pouvez pas utiliser la fonction turtle qui permet d'afficher un texte.

2 Première perle (3)



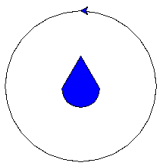
Écrire une fonction `perleA` qui dessine une perle avec un carré au centre. Elle doit prendre deux arguments : `rayon` et `couleur`. `rayon` est le rayon de la perle, et `couleur` est la couleur du carré (qui peut être donné comme une couleur python : soit avec un

texte, soit avec un triplet RVB). Le carré doit s'adapter à la taille de la perle et rester centré si la perle a une taille différente (donc prévoir de tout calculer en fonction de rayon).

Ici on a tracé `perleA(100, "green")`.

Remarque : dans l'exemple ci-dessus, la tortue commence en haut de la perle, tournée vers la gauche et revient à sa position et à son orientation de départ. Cela permettra de tracer plus facilement le collier plus tard. Vous pouvez choisir d'autres modes de fonctionnement, mais dans tous les cas précisez bien dans les commentaires où démarre la tortue et où elle finit, et choisissez *la même convention* pour toutes vos perles, cela facilitera la partie « collier ».

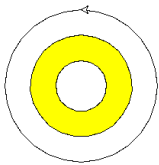
3 Deuxième perle (3)



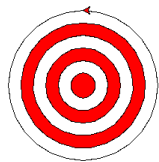
Écrire une fonction `perleB` qui dessine une perle avec une « goutte d'eau » au centre. Elle doit prendre deux arguments : rayon et couleur, comme la fonction précédente (taille et couleur de remplissage du dessin). Vous pouvez dessiner la perle de différentes manières.

On a tracé ici `perleB(100, "blue")`.

4 Troisième perle, variable (4)

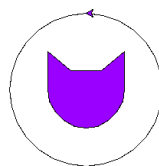


Écrire une fonction `perleC` qui dessine une perle avec des cercles concentriques, de couleurs alternées. Elle prendra trois arguments : rayon (comme les précédentes, le rayon de la perle), couleur (comme les précédentes) et `n` qui est le nombre de cercles à l'intérieur. Les couleurs intérieures doivent être alternées : blanc et la couleur.



Ci-contre, on a tracé `perleC(100, "yellow", 2)` et `perleC(100, "red", 2)`. Il est conseillé d'utiliser une boucle...

5 Une quatrième perle libre (3)



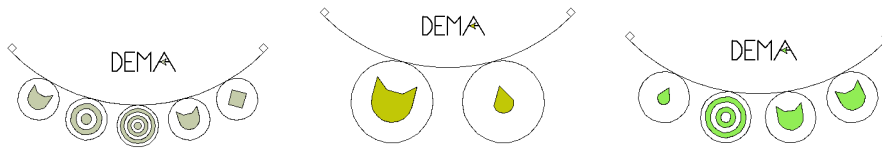
Écrire une fonction `perleD` qui dessine une perle avec à l'intérieur le motif de votre choix (ici une petite tête de chat). Vous pouvez choisir simple ou complexe...

Elle doit prendre en argument rayon et couleur : la perle doit être de ce rayon et la couleur principale celle indiquée.

Vous pouvez, si vous le souhaitez, rajouter d'autres couleurs (l'idée est que la couleur passée en argument soit « majoritaire »), des arguments supplémentaires, ...

Et le dessin à l'intérieur doit s'adapter à la taille de la perle, comme les autres perles.

6 Le collier de perles (4)



Écrire, enfin, la fonction `collier` qui va :

- Remettre la tortue à zéro,
- Dessiner un collier (la taille peut être fixe) avec des petits dessins aux extrémités (ils peuvent être simples, ici des carrés).
- Dessiner sur le collier un nombre aléatoire de perles (à vous de choisir quel aléatoire), d'une couleur aléatoire (mais la même pour toutes). Chaque perle est choisie aléatoirement parmi les quatre (ou plus?) perles disponibles. Pour la perle `C`, il faudra tirer au hasard le nombre `n` de cercles concentriques. La taille des perles doit s'adapter au nombre de perles (si il y en a plus, elles doivent être plus petites).
- À la fin, ajouter sa signature (grâce à la fonction du début).

Bonus

Vous pouvez décider de créer d'autres perles, comptées comme des bonus si elles sont un minimum intéressantes, et les intégrer à votre collier. Vous pouvez aussi agrémenter le collier de décorations supplémentaires, ...

Et n'oubliez pas de bien commenter votre code ! Une part de la note en tient compte !