

Exercice 18 : Maison

A l'aide des fonctionnalités du module `turtle` (voir page 4) :

- Ecrire une fonction qui dessine un carré de largeur 100.
Indications :
 - Dessiner le carré en tournant dans le sens des aiguilles d'une montre
 - La tortue démarre par défaut horizontalement et vers la droite
- Ecrire une fonction qui dessine un triangle équilatéral de base horizontale de largeur 100. Dessiner le triangle en tournant dans le sens contraire des aiguilles d'une montre.
- Ecrire un programme utilisant ces fonctions pour dessiner une maison

Exercice 19 : Fonction triangle

- Ecrire une fonction qui dessine un triangle de taille et de couleur données en paramètres
- Ecrire un programme Python qui demande à l'utilisateur la longueur du côté, la couleur du côté et appelle la fonction ci-dessus.

Exercice 20 : Fonction filtre ligne

Écrivez en python une fonction `filtre_ligne` qui prend en entrée une chaîne de caractères et l'imprime à l'écran sous réserve qu'elle fasse moins de n caractères de long. La chaîne de caractères et la valeur de n sont passés en paramètres :

Si elle fait plus de n caractères, la fonction écrit seulement la ligne "TROP LONG !".

Exercice 21 : Diviseur

Ecrire une fonction qui retourne vrai si d est diviseur de x , d et x étant donnés en paramètres.

Exemple : `diviseur(5, 10)` retourne vrai
`diviseur(5, 11)` retourne faux

Exercice 22 : Fonction arrondi bancaire

- Ecrire une fonction `decimale_n` qui retourne la n ème décimale d'un nombre réel x , n et x étant passés en paramètres.
- Écrire en python une fonction `arrondi_pair` qui prend en entrée un nombre entier positif n et qui retourne le nombre pair immédiatement supérieur ou égal à n .

Par exemple :

`arrondi_pair(5)` retourne 6
`arrondi_pair(6)` retourne 6
`arrondi_pair(7)` retourne 8

- La méthode dite *Méthode d'arrondissement au pair le plus proche, arrondi bancaire* ou encore *arrondi statistique* fonctionne suivant le principe suivant :
 - On précise la décimale d à laquelle le nombre doit être arrondi et la décimale dd qui suit la décimale d .
 - Si dd est inférieur ou égal à quatre, alors la décimale d reste inchangée.
 - Alors que si dd est six ou plus, la décimale est augmentée d'une unité.
 - Enfin si dd est le chiffre cinq lui-même suivi par des chiffres différents de zéro, on fait un arrondi pair.

Cette méthode est employée afin d'éliminer le biais qui surviendrait en arrondissant à chaque fois par excès les nombres dont le dernier chiffre est cinq.

Par exemple pour un arrondi au centième :

3.2464 est arrondi à 3.25
3.54398 est arrondi à 3.54
3.2356 est arrondi à 3.24
3.245 est arrondi à 3.24 parce que 4 est pair
3.715 est arrondi à 3.72 parce que 1 est impair

Ecrire une fonction `arrondi_bancaire` qui calcule l'arrondi bancaire à 2 chiffres après la virgule d'un nombre positif à 3 décimales. Cette fonction utilisera la fonction `arrondi_pair`.

Indication : une technique consiste à faire les calculs en entier puis à repasser en réel. On peut aussi utiliser la fonction `round` de Python

Exercice 23 : Fonction C_n^p

Je voudrais comparer mes chances de gagner au loto ou au keno.

- Ecrire une fonction `Cnp` qui calcule C_n^p le nombre de combinaisons de p éléments parmi n .
- Ecrire le programme qui calcule le nombre de combinaisons possibles au loto (6 numéros sur 49 possibles) et au Keno (10 numéros sur 70 possibles) à l'aide de la fonction `Cnp` écrite précédemment et affiche le jeu pour lequel il ya le moins de combinaisons possibles.

Exercice 23 bis : Equation d'une droite

$$Y = ax + b$$

Ecrire une fonction qui, à partir des abscisses et ordonnées de deux points connus de cette droite, calcule et retourne le coefficient directeur a de la droite ainsi que son ordonnée à l'origine b .

Exercice 24 : Symbole de la radioactivité

On dispose des fonctions "turtle" (voir fonctions de [turtle](#) page 4) suivantes :

```
from turtle import *
```

```
def square(length) :  
    forward(length)  
    left(90)  
    forward(length)  
    left(90)  
    forward(length)  
    left(90)  
    forward(length)  
    left(90)
```

```
def sector(radius, angle) :  
    forward(radius)  
    left(90)  
    circle(radius, angle)  
    left(90)  
    forward(radius)  
    left(180-angle)
```



Ecrire un programme qui dessine le logo de la radioactivité.

Exercice 24 bis : Etoile

A l'aide des fonctions du module [turtle](#) (voir fonctions de [turtle](#) page 4) :

- Ecrire une fonction Python qui trace un triangle équilatéral, la longueur, l'épaisseur et la couleur des côtés étant données en paramètres. Vous pouvez simplement modifier la fonction `triangle` écrite dans l'exercice 18.

Indications : vous pouvez utiliser les fonctions `color`, `width` du module [turtle](#).

- Utiliser cette fonction pour dessiner l'étoile ci-contre.

Indications :

- La tortue tourne par défaut dans le sens des aiguilles d'une montre
- Si l = longueur d'un côté des triangles, la hauteur d'un triangle est égale à :
 $h = l * \sin(\pi/3)$
- Si l = longueur d'un côté des triangles, le rayon du cercle circonscrit est égal à :
 $r = l / 2. / \cos(\pi/6)$
- Vous pouvez utiliser les fonctions `down()`, `up`, `seth`, `setx`, `sety`.



Exercice 51 : Problèmes bancaires

1) Placement

Soit S une somme placée en chaque début d'année sur un compte rémunéré au taux annuel de t pour cent. La valeur finale F obtenue après n années est calculée par la formule :

$$F = S ((1 + t/100) + (1 + t/100)^2 + (1 + t/100)^3 + \dots + (1 + t/100)^n)$$

Écrire un programme qui permette de calculer :

- a) la valeur à terme pour un dépôt annuel constant. La durée du placement est donnée en nombres d'années, le taux est donné en pourcentage.
- b) le nombre d'années nécessaires pour obtenir un capital au moins égal à une somme donnée, avec dépôt annuel constant.

2) Emprunt

Un établissement bancaire propose des emprunts à ses clients avec remboursement mensuel constant. Il offre également la possibilité de faire des remboursements anticipés d'une partie de l'emprunt. Un remboursement anticipé peut être fait à n'importe quel moment mais doit être au moins égal à 3% du capital initial emprunté.

Quand il y a un remboursement anticipé, on suppose que le montant des mensualités n'est pas modifié mais que, a fortiori, le nombre de mensualités peut diminuer. Étant donné une somme donnée à rembourser (cette somme comprend le capital emprunté plus le montant des intérêts) et un nombre d'années de remboursement, écrire un programme qui calcule le montant mensuel du remboursement et qui offre chaque mois la possibilité d'avoir un remboursement anticipé. Pour cela il faut :

- vérifier que le montant du remboursement anticipé vérifie la règle des 3%
- mettre à jour le solde à rembourser
- ne plus se baser sur le nombre de mensualités pour l'itération mais tester à chaque fois si le solde à rembourser n'est pas nul.

Exercice 52 : Jeu de l'oie

Écrire un programme qui simule le déplacement d'un joueur sur un jeu de l'oie et compte le nombre de coups joués.

Le joueur est caractérisé par une place, comprise entre 0 et 66, qui situe sa position sur le parcours. A chaque tour de jeu, la nouvelle place du joueur est déterminée après le jet de 2 dés DE1 et DE2, sachant que les règles suivantes sont appliquées :

- il faut arriver exactement sur la case 66 sinon on recule,
- un diabolin sur toutes les cases multiples de 13, sauf au 39, fait reculer de 3 cases,
- une oie toutes les 9 cases sauf au 63 double le déplacement,
- une tête de mort au 58 renvoie au départ,
- un cheval à la case 41 envoie directement au 66.

Les valeurs de DE1 et DE2 pourront être tirées au hasard. Certains déplacements permettront de rebondir (par exemple sur une case multiple de 9 avec un tirage de dés valant 9).

Le Python et la tortue

Le module `turtle` de Python est un ensemble d'outils permettant de dessiner à l'aide d'instructions simples.

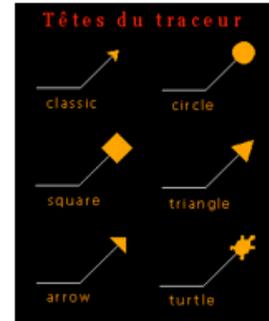
Pour utiliser les fonctions du module `turtle`, il faut importer le module :

```
from turtle import *
```

Vous trouverez ci-dessous quelques fonctions du module `turtle`. En mode standard, la tortue est placée au centre de la fenêtre graphique, elle se dirige vers la droite et tourne dans le sens des aiguilles d'une montre.

- `reset()` On efface tout et on recommence
- `goto(x, y)` Aller à l'endroit de coordonnées x et y (le point de coordonnées 0, 0 est situé au centre de la fenêtre graphique)
- `forward(distance)` Avancer d'une distance donnée
- `backward(distance)` Reculer
- `up()` Relever le crayon (pour pouvoir avancer sans dessiner)
- `down()` Abaisser le crayon (pour pouvoir recommencer à dessiner)
- `color(couleur)` Couleur peut être une chaîne prédéfinie ('red', 'blue', 'green', etc.)
- `left(angle)` Tourner à gauche d'un angle donné (exprimé en degré)
- `right(angle)` Tourner à droite
- `setheading(x)` Dirige directement dans une direction donnée par rapport à la fenêtre
- `width(épaisseur)` Choisir l'épaisseur du tracé
- `begin_fill()` Débute le remplissage d'un contour fermé à l'aide de la couleur sélectionnée
- `end_fill()` arrête le remplissage commencé par `begin_fill`
- `write(texte)` texte doit être une chaîne de caractères délimitée avec des " ou des '
- `circle(x)` cette commande trace un cercle complet de rayon x en partant de la position et de l'orientation actuelle du traceur, en tournant vers la gauche, et revient donc à la position initiale juste avant le tracé.
- `circle(x,y)` cette variante fait la même chose que `circle(x)` mais se limite à dessiner le cercle seulement sur un arc de y.
- `bgcolor(couleur)` permet de définir le fond de la fenêtre
- `exitonclick()` permet de sortir du programme par un simple `click` dans la fenêtre

- `speed(v)` vitesse d'avancement de la tortue durant le traçage. Attention : 0 signifie qu'aucune animation n'aura lieu (donc c'est en quelque sorte la vitesse la plus rapide). Mais vous pouvez aussi utiliser une échelle allant de 1 à 10 : 1 étant la vitesse la plus lente, 10 la plus rapide.
- `tracer(0)` permet de désactiver l'option d'"animation" de `turtle` et de lancer la production du dessin en une seule fois à la fin.
- `shape(traceur)` cette fonction permet de choisir la forme de son traceur. La forme par défaut est celle nommée 'classic'. Dans l'illustration ci-dessous, vous pouvez voir les 6 types différents de traceur. Pour ma part, je trouve sympathique l'image de la petite tortue, c'est pourquoi je mettrais souvent au début des exemples de ce didacticiel l'instruction suivante : `shape('turtle')`.



Exercice 13 : Opérations sur deux entiers

- a) Ecrire un programme qui saisit deux entiers et un code opération (+, -, * ou /) et qui affiche le résultat de cette opération sur ces entiers.
- b) Ecrire une fonction à 3 paramètres : deux entiers et un code opération (+, -, * ou /) et qui retourne le résultat de cette opération sur ces entiers.

Exemple : la fonction a en paramètre les entiers 12 et 5, le code opération '+' et affiche 17