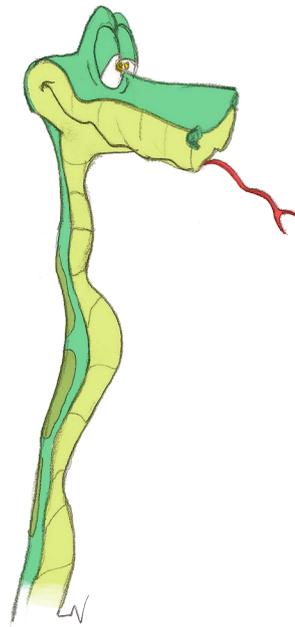




Recueil d'Exercices Python

version 1.4

Novembre 2023



Avant-propos

Pour joindre le compilateur des exercices :

✉ eric.berthomier@free.fr

Sources

- Les illustrations sont extraites du cours de Bob CORDEAU et Laurent POINTAL. Elles ont été réalisées par Hélène CORDEAU.
- Les exercices sont inventés ou piochés au gré de mes clics sur la toile. Je citerai notamment : TiProf : <http://tiprof.fr>

L'ennui ...

L'ennui est entré dans le monde par la paresse. - Jean de La Bruyère -

Aussi, afin d'éviter cette paresse, une fois fini l'ensemble des exercices, il est possible d'aller jouer aux Mathématiques Informatiques, en se rendant sur le site de mon cher ami Laurent Signac :

<https://deptinfo-ensip.univ-poitiers.fr/pydefis/>

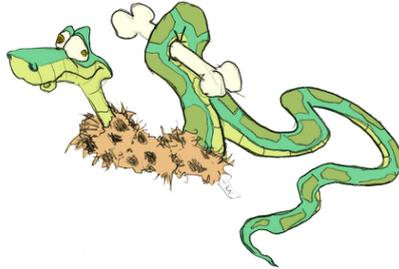
Versions

Ce recueil évolue en fonction des demandes et de mon temps ... toute contribution est la bienvenue.

| | | |
|----------------|---|-----------|
| Décembre 2014 | Version initiale | 1.0 |
| Janvier 2016 | Ajout d'exercices sur la récurrence les expressions régulières | 1.1 |
| Septembre 2017 | Ajout d'exercice CRobot (POO) introduction du mot clé super() | 1.2 |
| Janvier 2021 | Correction Expression Régulière Adresse IP | 1.2 / 1.3 |
| Novembre 2023 | Courbe du dragon - Ajout d'un lien | 1.4 |



Mise en bouche



Rappel des principales commandes



| | |
|--------------|---------------------------|
| Typage | type () |
| Affectation | = |
| Saisie | input () |
| Commentaires | # Ceci est un commentaire |

Avant de commencer...

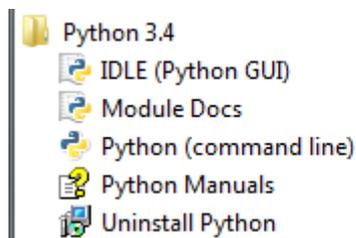
La réalisation des ces travaux pratiques nécessite l'installation de Python 3 et pour plus de commodités de Notepad++ (Windows) ou Scite (Linux).

- Python : <https://www.python.org/downloads/>
- Notepad++ : <http://notepad-plus-plus.org/fr/>
- Scite : <http://www.scintilla.org/SciTEDownload.html>

Langage Python (Windows)

L'interpréteur est lancé par l'item correspondant ajouté dans le menu Démarrer à l'issue de l'installation.

Par exemple :



À l'aide de NotePad++, saisir le script suivant et l'enregistrer sous le nom `hello.py` :

```
#! C:\Python34\python
# -*- coding: UTF-8 -*- pour spécifier le codage des caractères

print (5) # affichage de la valeur 5

a = 5
print (a) # affichage du contenu de a
print ("a = ", a) # affichage d'un message et du contenu de a
```

```

b = 5.50
print ("b = ", b) # affichage d'un message, du contenu de a et de b sur la même ligne

c = 5,50,14
print ("c = ",c)

texte="Mon texte"
print (texte)
print()

nom = input("Quel est votre nom : ")
print ("nom = ",nom)

# Affichage des types
print ()
print ("type de a: ",type(a))
print ("type de b: ",type(b))
print ("type de c: ",type(c))
print ("type de texte: ",type(texte))
print ()
print ("type de nom: ",type(nom))

```

Le chemin de l'exécutable Python est nécessaire pour que le script puisse désigner l'interpréteur qui doit exécuter ce dernier.

Remarque

✓ Une fois sauvé avec l'extension `.py`, le script s'exécute quand on clique sur son icône.

Il est aussi possible d'exécuter le script en ligne de commande :

```
.\hellow.py
```

Langage Python (Linux)

L'interpréteur est lancé en ligne de commande par l'exécution de :

```
python
```

Cette commande peut être modifiée en fonction de la version, que l'on veut exécuter.

À l'aide de votre éditeur favori, saisir le script suivant et l'enregistrer en `hello.py` :

```

#!/usr/bin/python3
# -*- coding: UTF-8 -*- pour spécifier le codage des caractères

print (5) # affichage de la valeur 5

a = 5
print (a) # affichage du contenu de a
print ("a = ", a) # affichage d'un message et du contenu de a

b = 5.50
print ("b = ", b) # affichage d'un message, et du contenu de b

c = 5,50,13
print ("c = ",c)

texte="Mon texte"
print (texte)
print()

nom = input("Quel est votre nom : ")
print ("nom = ",nom)

# Affichage des types
print ()
print ("type de a: ",type(a))
print ("type de b: ",type(b))
print ("type de c: ",type(c))
print ("type de texte: ",type(texte))
print ()
print ("type de nom: ",type(nom))

```

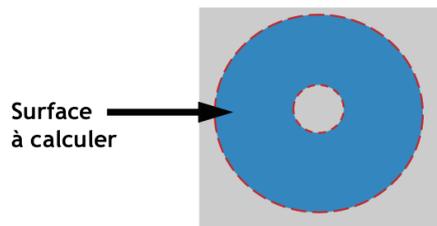
Le chemin de l'exécutable Python est nécessaire pour que le script puisse désigner l'interpréteur, qui est capable de l'exécuter.

Pour pouvoir exécuter le script, il faut en premier lieu rendre ce dernier exécutable.

```
chmod +x hello1.py
./hello1.py
```

Application directe du cours

1. Écrire un programme, qui définit 3 variables : une variable de type texte, une variable de type nombre entier, une variable de type nombre décimal et qui affiche leur type.
2. Affecter **dans une même ligne** les 3 variables précédemment définies.
3. Cet exercice est sans rapport avec le précédent ...
Écrire un programme qui, à partir de la saisie d'un rayon et d'une hauteur, calcule le volume d'un cône droit : $V = \frac{1}{3} \times \pi \times r^2 \times h$.
Comparer la précision de calcul avec votre calculatrice ou celle de l'ordinateur.
4. Une machine découpe dans une plaque, des disques circulaires de rayon r_{Ext} , percés d'un trou circulaire de rayon r_{Int} avec $r_{Int} < r_{Ext}$ et ne débordant pas du disque.
Quelle est la surface d'un disque découpé?

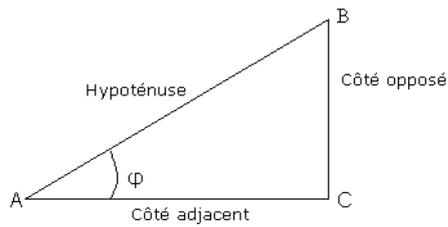


Guide :

- (a) trouver les données
- (b) effectuer les calculs
- (c) afficher le résultat

Application réfléchie

1. Écrire un programme qui affiche le type du résultat des instructions suivantes :
 - `a=3`
 - `a==3`
2. Écrire un programme, qui ajoute une chaîne de caractères à un nombre entier (Exemple la chaîne "le chat" et le nombre 3 pour donner le chat + 3).
3. Écrire un programme, qui donne la mesure de l'angle α d'un triangle rectangle, dont on saisit le côté opposé et l'hypoténuse.
Rappel : $\sin \varphi = \text{CoteOppose} / \text{Hypotenuse}$



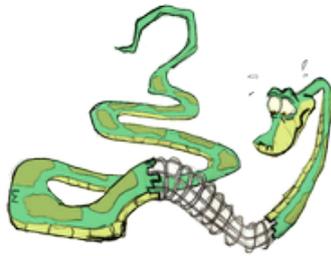
Application avancée

1. Écrire un programme qui réalise la saisie d'un nombre entier puis affiche la valeur ainsi saisie et son type. Essayer de dépasser la taille maximale des entiers (cf. cours C) avec Python 2 et 3. Expliquer.
2. Lors de la saisie d'un nombre par cast (`int (input())`) : indiquer une chaîne de caractères en lieu et place d'un nombre, rechercher comment éviter ce bug (aide : commande `try`)

Culture Générale

1. Quel autre mot courant de l'informatique est issu de la culture Monthy Python ?
2. Qu'est-ce qu'un radian ?

Conditions - Itérations - Répétitions



Rappel des principales commandes



| | |
|-----------------------|-------------------|
| Si | if (condition): |
| Sinon Si | elif (condition): |
| Sinon | else: |
| Pour | for variable in |
| Tant que | while condition : |
| Longueur d'une chaîne | len (chaîne) |

Application directe du cours

1. Écrire un programme `min_max.py`, qui demande de saisir 2 valeurs et qui affiche la plus petite des 2 valeurs.
2. Écrire un script `longueur_chaine.py`, qui demande de saisir 2 chaînes de caractères et qui affiche la plus grande des 2 chaînes (celle qui a le plus de caractères).
3. Écrire le script `fumeurs.py` vu en cours, en le complétant par des saisies.
4. Écrire le script `convertir.py`, qui effectue une conversion euros en dollars.
 - Le programme commencera par demander à l'utilisateur d'indiquer par un caractère 'E' ou '\$' la devise du montant qu'il va entrer.
 - Puis le programme exécutera une action conditionnelle de la forme :

```

if devise == 'E' :
    .....
elif devise == '$' :
    .....
else :
    ..... # affichage d'un message d'erreur
    
```

5. Écrire un programme, qui affiche 50 fois "Je dois ranger mon bureau" à l'aide de l'instruction `for`.
6. Chanson traditionnelle bretonne
La séquence d'instructions

```
n=10
print ("C'est dans %d ans je m'en irai j'entends le loup le renard chanter" % n)
```

permet d'afficher le message :

C'est dans 10 ans je m'en irai j'entends le loup le renard chanter

Écrire une boucle while qui permet d'afficher :

```
C'est dans 10 ans je m'en irai j'entends le loup le renard chanter
C'est dans 9 ans je m'en irai j'entends le loup le renard chanter
C'est dans 8 ans je m'en irai j'entends le loup le renard chanter
...
C'est dans 1 ans je m'en irai j'entends le loup le renard chanter
```

Dans un premier temps, on ne s'occupera pas de la faute d'orthographe de la dernière ligne.

Application réfléchie

- Écrire le script `multiple3.py` qui affiche en fonction d'une valeur saisie l'un des messages suivants :
 - "Ce nombre est pair"
 - "Ce nombre est impair, mais est multiple de 3"
 - "Ce nombre n'est ni pair ni multiple de 3"

Définition

 Un nombre est multiple de 3 si le reste de la division de ce nombre par 3 est égal à 0. Ce reste est nommé modulo

Syntaxe

 % permet d'obtenir le modulo en Python

- Différence entre `if` et `while`.
On considère les deux programmes suivants :

```
a = 7.5
if a > 3 :
    a = a-1
```

```
a = 7.5
while a > 3 :
    a = a-1
```

Comprendre, calculer et exécuter le programme, afin de donner la valeur de `a`.

- Écrire un programme qui affiche les nombres de 2 en 2 jusqu'à 100 avec un `for` puis avec un `while`.
- Écrire un programme qui affiche les tables de multiplications de 1 à 10.
Aide : utiliser une boucle imbriquée.

Application avancée

- Écrire un programme qui affiche un joli sapin de Noël, dont la taille est donnée par l'utilisateur.
Exemple pour une taille de 12 lignes :

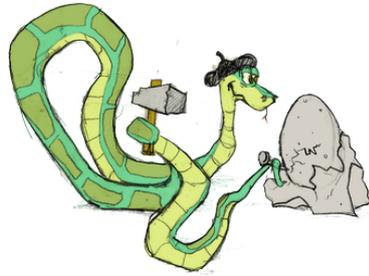
```
^
^^
^^^
^^^^
^^^^^
^^^^^^
^^^^^^^
^^^^^^^
^^^^^^^
^^^^^^^
^^^^^^^
^^^^^^^
```

```
      ^^^^^^^^
     ^^^^^^^^^
    ^^^^^^^^^^^
   ^^^^^^^^^^^^
  ^^^^^^^^^^^^^
 ^^^^^^^^^^^^^^
^________________^
```

Culture Générale

1. Comment se traduit en Python le `i++`, `i--` du C / C++ en Python ?

Utilisation d'un module : Turtle



Rappel des principales commandes



| | |
|-----------------|----------------------------------|
| Utiliser Turtle | <code>import turtle</code> |
| Efface tout | <code>reset()</code> |
| Aller à | <code>goto (x,y)</code> |
| Avancer | <code>forward (distance)</code> |
| Reculer | <code>backward (distance)</code> |
| Couleur | <code>color (couleur)</code> |
| Tourner gauche | <code>left (angle)</code> |
| Tourner droite | <code>right (angle)</code> |

Turtle nécessite l'installation de tk-inker : `apt install python-tk` ou `apt install python3-tk`

Application directe du cours

1. Écrire un programme `carre.py` qui trace un carré.
2. Écrire un programme, qui trace un triangle équilatéral (3 angles à 60°).
3. Écrire un programme, qui trace un hexagone (polygone à 6 côtés, angles interne à 120°).

Application réfléchie

1. Écrire un programme, qui trace un carré, puis un triangle.
Modifier ensuite votre programme pour dessiner n figures consécutives adjacentes.

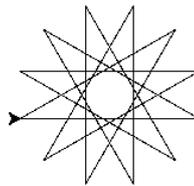


2. Écrire un programme, qui trace un carré puis un triangle, qui grossissent au fur et à mesure.



Application avancée

1. Écrire un programme, qui trace un cercle (non parfait), sans utiliser la fonction `circle` de Turtle.
2. Écrire un programme, qui trace une étoile.



3. Étudier le code suivant.

```
#!/usr/bin/python3
# -*- coding: UTF-8 -*-

import turtle

turtle.setup (800, 600) # Taille du canevas

wn = turtle.Screen () # La fenêtre d'écran de Turtle
wn.bgcolor ("lightblue") # Couleur de fond de la fenêtre
wn.title ("Python Turtle") # Définit un titre

tess= turtle.Turtle() # Objet "Turtle"
tess.shape ("turtle") # Une tortue au lieu d'un triangle
tess.color ("green") # Tortue verte

tess.goto(200,200)
tess.right (90)
tess.forward(100)
tess.left (90)
tess.forward(50)

caroline= turtle.Turtle()
caroline.shape ("turtle")
caroline.color ("black")
caroline.forward (30)
caroline.right (90)
caroline.forward(100)
caroline.left (90)
caroline.forward(50)

wn.exitonclick() # On attend un clic sur la croix
```

Culture Générale

1. Turtle est un module de Python, mais s'inspire d'un langage bien plus ancien. Quel est il?

Les Listes



Rappel des principales commandes



| | |
|----------------------------------|--------------------------------|
| Supprimer un élément d'une liste | <code>remove()</code> |
| Ajouter un élément d'une liste | <code>append()</code> |
| Trier une liste | <code>sort()</code> |
| Inverser l'ordre d'une liste | <code>reverse()</code> |
| Pour | <code>for variable in</code> |
| Tant que | <code>while condition :</code> |
| Longueur d'une chaîne | <code>len (chaîne)</code> |

Préambule

Dans l'ensemble des ces exercices, la liste suivante sera utilisée :

- lapin
- chat
- chien
- chiot
- dragon
- ornithorynque

Application directe du cours

1. Écrire un programme `liste_animaux.py`, qui initialise la liste et qui affiche l'ensemble des éléments.
2. Afficher la liste de manière inversée.
3. Afficher la liste de manière triée.
4. Ajouter (`append`) l'élément `troll` dans la liste, puis supprimer l'ensemble des animaux domestiques. Afficher le résultat. Afin de réaliser la suppression, on créera une liste des animaux domestiques.

Application réfléchie

1. Écrire un programme `liste_chaine.py`, qui donne le nombre de caractères de chaque élément de la liste.

Exemple : lapin possède 5 caractères.

Dans les deux exercices suivants, on considèrera un tableau initialisé avec 10 valeurs aléatoires. Le but des exercices est de dire, si la valeur saisie par l'utilisateur est dans la liste ou non.

Afin de vous aider, voici le début des deux programmes.

```
#!/usr/bin/python3
# -*- coding: UTF-8 -*-

import random

tableau_jeu=[]

# Initialisation d'une liste de 10 éléments
for i in range (0,10):
    tableau_jeu.append (random.randint (1,10))
```

2. Recherche séquentielle dans une liste non triée.
Le programme parcourt la liste des valeurs en la comparant une à une à la valeur cherchée, et sort de la boucle :
 - soit quand il a parcouru toute la liste sans trouver la valeur
 - soit quand la valeur a été trouvée
 Si la valeur a été trouvée, le programme annonce "Gagné", sinon, le programme annonce "Perdu".
3. Recherche séquentielle dans une liste triée.
Le programme trie les valeurs tirées au sort, puis parcourt la liste des valeurs en la comparant une à une à la valeur cherchée. Il sort de la boucle :
 - soit quand il a parcouru toute la liste sans trouver la valeur
 - soit quand la valeur lue dans la liste est supérieure à la saisie
 Si la valeur a été trouvée, le programme annonce "Gagné", sinon, le programme annonce "Perdu".
4. Jeu de cartes - Une couleur
 - Créer un programme `jeu_de_cartes.py`, qui crée une liste avec l'ensemble des cartes soit : 1,2,...10,V,D,R sans spécifier la couleur. On se permettra une saisie manuelle des valeurs des cartes.
 - Contrôler le contenu de votre liste en l'affichant
5. Jeu de cartes complet Créer une nouvelle liste avec les couleurs, afficher l'ensemble du jeu de cartes.
Exemple : R de Coeur.

Application avancée

2 Jeux de Cartes

- Créer un programme `jeu_de_cartes2.py`, qui crée deux listes avec l'ensemble des cartes (soit 2 * 52 cartes).
Exemple : "1Coeur" pour As de Coeur.
- Contrôler le contenu de vos 2 listes en les affichant
- On tire 10 cartes au hasard dans chaque jeu. Faire afficher les 10 cartes tirées pour chacun des 2 jeux.
- Trouver les cartes identiques dans les 2 tirages.

Scrabble©

En compétition de Scrabble©, le tirage des 7 lettres est dit valide quand celui-ci comporte au moins 2 voyelles¹.

Écrire un programme qui tire les lettres dans un sachet constitué des lettres suivantes (simplifié par rapport au vrai jeu) et qui retire les lettres jusqu'à ce que le jeu soit correct.

On ne vous demande qu'un seul tirage valide donc pas de prise en compte des lettres utilisées dans un précédent tirage ...

Instructions : On affichera les tirages incorrects.

- E : 15 lettres
- A : 9 lettres
- I : 6 lettres
- N : 6 lettres
- O : 6 lettres
- R : 6 lettres
- S : 6 lettres
- T : 6 lettres

Aide : Utiliser une liste d'éléments et les fonctions qui lui sont associées ...

Faux ami

Problématique de la copie

La copie d'une liste par égal peut faire penser que la copie et son original sont deux éléments dissociés. Ce n'est hélas pas le cas.

```
#!/usr/bin/python
```

```
a=[1,2,3]
b=a
b.append(4)
print("a: ", a)
print("b: ", b)
```

```
('a: ', [1, 2, 3, 4])
('b: ', [1, 2, 3, 4])
```

Pour réaliser une copie différenciée de l'originale, il faut soit utiliser le module `copy` soit faire la copie à la main.

Copie manuelle

```
#!/usr/bin/python
```

```
a=[1,2,3]
b=[]
for i in a:
    b.append(i)
b.append(4)
print("a: ", a)
print("b: ", b)
```

```
('a: ', [1, 2, 3])
('b: ', [1, 2, 3, 4])
```

Module `copy`

1. Cette règle est valable pour les 16 premiers tirages

```
#!/usr/bin/python
import copy

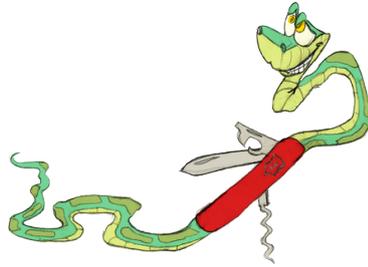
a=[1,2,3]
b=copy.deepcopy(a)
b.append (4)
print ("a: ", a)
print ("b: ", b)
```

```
('a: ', [1, 2, 3])
('b: ', [1, 2, 3, 4])
```

Culture Générale

1. En informatique, qu'est ce qu'un TROLL ?

Fonctions



Rappel des principales commandes



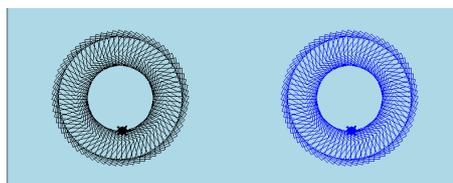
| | |
|---------------------------|--|
| Définition d'une fonction | <code>def nom_fonction (param1, param2, ...):</code> |
| Retour d'une valeur | <code>return valeur</code> |
| Vrai | <code>True</code> |
| Faux | <code>False</code> |

Préambule

Dans l'ensemble des exercices ci-dessous, il vous est demandé d'écrire des fonctions ... mais aussi le programme, qui lui est associé afin de les tester !

Application directe du cours

1. `tortue_carre.py` : écrire une fonction `dessine_carre`, qui fait tracer un carré de 50 pixels de côté à la tortue.
2. Modifier le programme précédent de la façon suivante :
 - (a) la tortue dessine un carré
 - (b) la tortue tourne de 5°
 - (c) la tortue avance de 5 pixels
 - (d) la tortue dessine un carré
 Et ceci **72 fois**.
 Nommer le `tortue_carre_tournant.py`
3. Modifier le programme précédent de manière à faire des lunettes ...



4. `pair_impair.py` : écrire 2 fonctions :
 - `pair (nbre)`, qui renvoie True, si le nombre est Pair
 - `impair (nbre)`, qui renvoie True, si le nombre est Impair
 Les nombres sera demandé à l'utilisateur. Le résultat attendu est : "La fonction Pair retourne True pour la valeur 2"

5. `mini_maxi.py` : écrire 2 fonctions :
 - `mini (a,b)` qui renvoie le minimum entre a et b
 - `maxi (a,b)` qui renvoie le maximum entre a et b
 Les 2 nombres a et b seront demandés à l'utilisateur.

Application réfléchie

1. Modifier le programme `tortue_carre_tournant.py` en `tortue_carre_couleur.py`, pour que la couleur du carré soit passée en paramètre à la fonction `dessine_carre`.
2. Modifier le programme `tortue_carre_couleur.py` en `tortue_triangle_couleur.py`, pour qu'en lieu et place d'un carré, ce soit un triangle équilatéral, qui soit dessiné.

Application avancée

1. `tortue_carre_france.py` : Reprendre le programme `tortue_carre_couleur.py` et créer un liste des trois couleurs du drapeau français, afin de dessiner les carrés avec ces 3 couleurs.
2. `palyndrome.py` : Créer une fonction qui indique, si une chaîne de caractères est ou non un palyndrome ¹.

Aide

 La commande `list(chaine)` permet de transformer une chaîne de caractères en liste de caractères.

La commande `"".join(list)` permet de transformer une liste en chaîne de caractères.

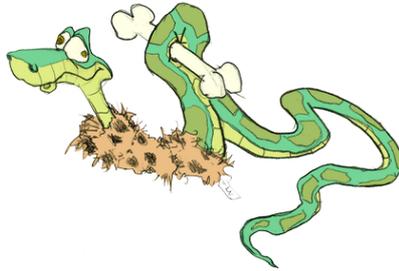
-
3. `pendu.py` : Créer un programme de Pendu.

Culture Générale

1. Qu'est ce qui différencie une procédure d'une fonction? Citer un langage de programmation, qui différencie ces deux termes dans sa syntaxe.

1. Un palyndrome est un mot qui peut s'écrire dans les deux sens, exemple : radar

Fichiers et module OS



Rappel des principales commandes



| | |
|------------------------|-----------------|
| Ouvrir un fichier | open |
| Fermer un fichier | close |
| Lire dans un fichier | read / readline |
| Écrire dans un fichier | write |

Application directe du cours

1. Écrire un programme `lire_fichier.py`, qui lit le fichier `loremipsum.txt` et l'affiche.
2. `temperature.py` : Créer un fichier avec un mot par ligne à partir de la liste suivante : `"chaud", "froid", "tempéré", "glacial", "brûlant"`.
3. `temperature_add.py` : Ajouter à ce fichier les équivalents anglais : `"hot", "cold", "moderate", "icy", "ardent"`
4. Lister le contenu du répertoire courant.

Application réfléchie

1. `dico.py` : Lire le fichier `dico.txt` et en ressortir les palindromes.

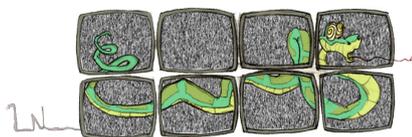
Aide

👉 `mot = mot.replace("\n", "")` permet de supprimer le retour à la ligne d'une chaîne.

Culture Générale

1. Qu'est ce que Lorem Ipsum ?
2. En terme SSI, quel peut être l'intérêt de la lecture d'un dictionnaire ?

Programmation Orientée Objet



Astuce sur les modules

Il est possible de tester les modules écrits directement en utilisant `if __name__ == "__main__":`.

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

import math

def Aire_Cercle (rayon):
    return (math.pi * rayon * rayon )

def Perimetre_Cercle (rayon):
    return (2 * math.pi * rayon)

if __name__ == "__main__":
    r = int (input ("Votre rayon : "))
    print ("Aire : %f" % (Aire_Cercle (r)))
    print ("Perimetre : %f" % (Perimetre_Cercle (r)))
```

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

from module import *

r = int (input ("Votre rayon : "))
print ("Aire : %f" % (Aire_Cercle (r)))
print ("Perimetre : %f" % (Perimetre_Cercle (r)))
```

Application directe du cours

Attention

⚠ Aucune création ou héritage de classe n'est demandé pour les exercices de l'"Application directe du cours"

1. `dir_tortue.py` : À l'aide de la commande `dir`, lister l'ensemble des propriétés de l'objet `Turtle`.
2. `3tortues.py` : Écrire un programme qui créer 3 tortues et qui leur fait effectuer des déplacements et des figures identiques, mais en partant d'une position différente.
 - Le déplacement sera implémenté sous la forme d'une fonction s'appliquant à un objet de type `tortue`.
 - La figure fera elle aussi l'objet d'une fonction s'appliquant à un objet de type `tortue`.

| Nom de la tortue | Position de départ |
|------------------|--------------------|
| angelo | 50,50 |
| donatello | 100,100 |
| raphael | 150,150 |

Application réfléchie

- Créer une classe Cercle (fichier CCercle.py) qui se définit par son rayon.
La classe CCercle aura 2 méthodes, qui permettront de calculer l'aire ($\pi \times r^2$) et le périmètre ($2 \times \pi \times r$).
Écrire un programme cercle.py qui instancie cette classe et utilise ses méthodes.
- Créer une classe Domino (fichier CDomino.py) qui se définit par la valeur des points sur ce dernier.
La classe CDomino aura 2 méthodes :
 - affiche_points qui affiche le domino
 - valeur qui donne la somme des points du domino
 Écrire un programme domino.py qui instancie cette classe et utilise ses méthodes :
 - Créer 2 dominos [2,6] et [4,3] et leur appliquer les méthodes
 - Créer une liste de 7 dominos [6,valeur de 1 à 6] qui les affiche puis indique la somme de l'ensemble.



Application avancée

Jeu de Cartes

- CJeuDeCartes.py : Créer un classe CJeuDeCartes qui crée un jeu de 52 cartes.
 - Une carte sera composée d'une valeur et d'une forme (coeur, carreau, pique, trèfle).
 - Les méthodes suivantes seront implémentées :
 - nom_carte : affiche le nom d'une carte de manière littérale, ex. Valet de Coeur.
 - battre : mélange le jeu de cartes (on utilisera la fonction shuffle du module random).
 - afficher : permet d'afficher le jeu de cartes.
 - tirer : permet d'extraire une carte du jeu.
 - Vous ferez un jeu de test, qui exécutera les points suivants :
 - Créer un jeu de cartes
 - Mélanger le jeu de cartes (battre)
 - Afficher le jeu de cartes mélangé
 - Tirer une à une toutes les cartes et les afficher

Robot

Cet exercice est adapté d'un devoir surveillé écrit par Mlle Imene Sghaier.

1. `CRobot.py` : Créer une classe `CRobot` qui crée un robot défini par
 - Les caractéristiques suivantes
 - Type
 - SN : Numéro de Série
 - Orientation
 - Status (En service, Hors service, En réparation)
 - Les méthodes suivantes
 - Constructeur `Robot ()` ou `Robot (type, sn)`
 - `getType ()` : retourne le type de robot
 - `getSN ()` : retourne le numéro de série du robot
 - `getOrientation ()` : retourne l'orientation du robot
 - `getStatus ()` : retourne le status du robot
 - `setOrientation (...)` : définit l'orientation du robot
 - `setEtat(...)` : définit l'état du robot
 - `tourner(...)` : tourne le robot d'un 1/4 de tour
 - `afficher()` : affiche les informations du robot
 - orientation est un attribut de type entier qui désigne l'orientation du robot.
1 : NORD, 2 : EST, 3 : SUD, 4 : OUEST
 - tourner permet de tourner le robot, par défaut vers la gauche
 - afficher permet d'afficher l'état, l'orientation, le numéro de série et le type du robot.

Instancier cette classe sur un tableau de 4 robots en utilisant l'ensemble des fonctions pour au moins l'un d'eux.
2. `CRobotMobile.py` : Créer un classe `CRobotMobile` qui
 - hérite de `CRobot`
 - se caractérise en plus avec les attributs entiers `abs` et `ord` qui définissent la position du `Robot-Mobile`
 - possède une méthode `avancer (...)` qui permet d'avancer le Robot selon son orientation
 - si on avance de `x` vers l'Est, l'abscisse augmente de `x`
 - si on avance de `x` vers l'Ouest, l'abscisse diminue de `x`
 - si on avance de `x` vers le Nord, l'ordonnée augmente de `x`
 - si on avance de `x` vers le Sud, l'ordonnée diminue de `x`
 - possède une méthode `affichePosition()` qui affiche la position (coordonnées).
 - (a) Écrire un constructeur sans argument de la classe `CRobotMobile`
 - (b) Écrire un constructeur à quatre arguments (`type`, `sn`, `abs`, `ord`) de la classe `CRobotMobile`
 - (c) Redéfinissez la méthode `affiche tout` en utilisant celle de la classe mère et la méthode `affichePosition()`
 - (d) Écrire un programme qui teste votre `CRobotMobile` en lui appliquant les actions suivantes :
 - i. Tourner vers l'Est
 - ii. Avancer de 4 vers l'Ouest
 - iii. Avancer de 6 vers le Nord
 - iv. Avancer de 14 vers l'Est
 - v. Reculer de 8 vers le Sud
 - (e) Proposer une amélioration de gestion pour l'orientation du robot ...

Culture Générale

1. Une méthode de modélisation s'appuie sur l'objet ou inversement. Quelle est elle ?

Les Expressions Régulières



Lectures

Lors du parcours de certains sites Python sur les expressions régulières notamment, vous trouverez des syntaxes avec un suffixe `r`. Afin de bien comprendre ce que ce suffixe produit, voici un exemple et son interprétation.

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-

import re

print (r"\tBonjour, \n et \n au revoir")
print ("\n--\n")
print ("\tBonjour, .* \n et \n au revoir")
```

```
\tBonjour, \n et \n au revoir

--

    Bonjour, .* \n et
au revoir
```

Avec `r`

Le caractère `\` est protégé de toute interprétation :

- `\t` reste `\t`
- `\\n` reste `\\n`
- `\\n` reste `\\n`

Sans `r`

Le caractère `\n` est pas protégé de toute interprétation :

- `\t` devient une tabulation
- `\\n` devient `\n`
- `\\n` devient un retour chariot

Aide

Expressions régulières

Le site <http://pythex.org/> vous aidera dans la création et la correction de vos expressions régulières, n'hésitez pas à l'utiliser.

Saisie

La lecture d'une chaîne de caractères peut entraîner la prise en compte du retour chariot de fin de saisie, il est possible d'utiliser ce petit code pour enlever ce dernier.

Syntaxe

```
clear = chaine.rstrip('\r\n')
```

La chaîne `clear` contient alors la chaîne `chaine` nettoyée de tout caractères de retour à la ligne.

Exemple

```
#!/usr/bin/python3
# -*- coding :utf-8 -*-

chaine="coucou\n"

print ("Initial : " + chaine)
print ("-----")

clear = chaine.rstrip('\r\n')

print ("Final : " + clear)
print ("-----")
```

```
Initial : coucou
-----
Final : coucou
-----
```

Application directe du cours

1. Un nombre est composés d'une suite de chiffres. Vérifier que la chaîne saisie est bien celle d'un nombre.
2. Une plaque d'immatriculation est composée de 2 lettres majuscules, un tiret ("-"), 3 chiffres, un tiret ("-") et enfin de 2 lettres majuscules. Vérifier que la chaîne saisie est bien celle d'une plaque d'immatriculation.
3. Une adresse IPv4 est composée de 4 nombres entre 0 et 255 séparés par des `.`. Vérifier que la chaîne saisie est bien celle d'une adresse IP (Merci à Eric Martin pour m'avoir signalé une erreur).
4. Une adresse mail est composée de caractères alphanumériques suivis de `@` suivis d'un `.` et d'un nom de domaine. Vérifier que la chaîne saisie est bien celle d'une adresse mail.
5. Identifier les expressions reconnues par les expressions régulières données ci-dessous. Pour se faire, libre à vous de programmer, tester le site ou d'effectuer les tâches manuellement sachant que le mieux est tout de même de comprendre ce que fait chacune de ces expressions.

Cet exercice est extrait du site <http://home.gna.org/unix-initiation/website/node190.html>

| Numéro | Chaîne | Numéro | Chaîne |
|--------|------------------------|--------|-----------------------------|
| a | abc | b | zzzz xx |
| c | abcdef | d | 123456 7890 abcaziuz |
| e | yyyy | f | xyz stuv abc |
| g | xx abcxxxxxxxxxx | h | xAb* 12345 |
| i | xAB* 45678 | j | 98745 xAb* 23654 |
| k | abc\$!k; | l | 567 |
| m | 5666777 | n | 57 |
| o | Suite... du paragraphe | p | Suite... de l'histoire |
| q | la suite... | r | Suite.. au prochain numero. |

| Numéro | Expressions Régulières | Numéro | Expressions Régulières |
|--------|---------------------------|--------|-------------------------|
| 1 | c\$ | 2 | c\\$ |
| 3 | ^abc | 4 | abc\$ |
| 5 | ^abc\$ | 6 | ^abc. |
| 7 | 45 | 8 | ^56[67] |
| 9 | .56[67] | 10 | x[Aa][Bb] |
| 11 | x[^Aa] | 12 | [Aa][^b] |
| 13 | abcd* | 14 | 566?7 |
| 15 | [r-v] | 16 | 56*7* |
| 17 | 56+7+ | 18 | 56??? |
| 19 | 566?7 | 20 | 987 789 |
| 21 | abc [def] | 22 | *[Aa][Bb].*12.* |
| 23 | *12.*[Aa][Bb] | 24 | *[Aa]b.*12.*12.*[Aa]b.* |
| 25 | *([Aa]b.*12.*12.*[Aa]b).* | 26 | abc[def][m-x]* |
| 27 | ^[Ss]uite\\.\\.\\.* | | |

Application réfléchie

1. Modifier le code suivant pour rechercher le contenu de texte compris entre <Erreur></Erreur> dans le fichier vandamme.txt en ne prenant pas en compte les espaces inutiles.

```
#!/usr/bin/python3
# -*- coding: UTF-8 -*-

import re

# Chaîne de caractères sur laquelle on va travailler
S = 'Bugger all down here on earth!'

print ("\nCas 1\n----\n")

# Expression régulière visant à matcher le texte
patt = re.compile('Bugger all\s*(.*)here on (.*)!')

# Match
mobj = patt.match(S)

# Affichage du résultat dans le cas où un pattern correspond
if mobj:
    print (mobj.group(1))
    print (mobj.group(2))
else:
    print ("Not find")

# Attention match analyse la chaîne à partir de son début

print ("\nCas 2\n----\n")

# Modification de l'expression régulière visant à matcher le texte
patt = re.compile('all\s*(.*)here on (.*)!')
```

```

# Match
mobj = patt.match(S)

# Affichage du résultat dans le cas où un pattern correspond
if mobj:
    print (mobj.group(1))
    print (mobj.group(2))
else:
    print ("Not find")

# On doit donc le corriger en

print ("\nCas 3\n----\n")

# Modification de l'expression régulière visant à matcher le texte
patt = re.compile('.*all\s*(.*)here on (.*)!')

# Match
mobj = patt.match(S)

# Affichage du résultat dans le cas où un pattern correspond
if mobj:
    print (mobj.group(1))
    print (mobj.group(2))
else:
    print ("Not find")

```

Exemple, la saisie de la chaîne : <Erreur> Ceci est une erreur </Erreur> retournera Ceci est une erreur.

Application avancée

1. Analyser le fichier `script.txt` pour n'isoler que les commandes (entre crochets). On conservera les crochets à l'affichage.
Puis, dans un second temps, l'erreur pour chaque commande, c'est à dire le 3ème champs séparé d'un | si le second champs est erreur. **On affichera uniquement les erreurs différentes des autres.**
 - (a) Écrire un programme qui lit le fichier
 - (b) Isoler les différentes syntaxes à analyser
 - (c) Créer l'expression régulière ou les expressions régulières susceptibles de répondre au problème.
 - (d) Tester votre expression sur un échantillon de lignes
 - (e) Intégrer votre expression régulière dans la lecture du fichier



| Anchors | | Sample Patterns | | | | |
|---|--------------------------------|---|--|--|---|---|
| ^ | Start of line + | ([A-Za-z0-9-]+) | Letters, numbers and hyphens | | | |
| \A | Start of string + | (\d{1,2}\V\d{1,2}\V\d{4}) | Date (e.g. 21/3/2006) | | | |
| \$ | End of line + | ([^\s]+(?:\.(jpg gif png))\.\2) | jpg, gif or png image | | | |
| \Z | End of string + | (^[1-9]{1}\$ ^[1-4]{1}[0-9]{1}\$ ^50\$) | Any number from 1 to 50 inclusive | | | |
| \b | Word boundary + | (#[A-Fa-f0-9]{3}([A-Fa-f0-9]{3})?) | Valid hexadecimal colour code | | | |
| \B | Not word boundary + | ((?=\d)(?=[a-z])(?=[A-Z]).{8,15}) | 8 to 15 character string with at least one upper case letter, one lower case letter, and one digit (useful for passwords). | | | |
| \< | Start of word | (\w+@[a-zA-Z_]+?\.[a-zA-Z]{2,6}) | Email addresses | | | |
| \> | End of word | (\</?[^>]+\>) | HTML Tags | | | |
| Character Classes | | Note <i>These patterns are intended for reference purposes and have not been extensively tested. Please use with caution and test thoroughly before use.</i> | | | | |
| \c | Control character | | | | | |
| \s | White space | | | | | |
| \S | Not white space | | | | | |
| \d | Digit | | | | | |
| \D | Not digit | | | | | |
| \w | Word | | | | | |
| \W | Not word | | | | | |
| \xhh | Hexadecimal character hh | | | | | |
| \Oxxx | Octal character xxx | | | | | |
| POSIX Character Classes | | Quantifiers | Ranges | | | |
| [[:upper:]] | Upper case letters | * | 0 or more + | . | Any character except new line (\n) + | |
| [[:lower:]] | Lower case letters | *? | 0 or more, ungreedy + | (a b) | a or b + | |
| [[:alpha:]] | All letters | + | 1 or more + | (...) | Group + | |
| [[:alnum:]] | Digits and letters | ++ | 1 or more, ungreedy + | (?...) | Passive Group + | |
| [[:digit:]] | Digits | ? | 0 or 1 + | [abc] | Range (a or b or c) + | |
| [[:xdigit:]] | Hexadecimal digits | ?? | 0 or 1, ungreedy + | [^abc] | Not a or b or c + | |
| [[:punct:]] | Punctuation | {3} | Exactly 3 + | [a-q] | Letter between a and q + | |
| [[:blank:]] | Space and tab | {3,} | 3 or more + | [A-Q] | Upper case letter + between A and Q + | |
| [[:space:]] | Blank characters | {3,5} | 3, 4 or 5 + | [0-7] | Digit between 0 and 7 + | |
| [[:cntrl:]] | Control characters | {3,5}? | 3, 4 or 5, ungreedy + | \n | nth group/subpattern + | |
| [[:graph:]] | Printed characters | Special Characters | | Note <i>Ranges are inclusive.</i> | | |
| [[:print:]] | Printed characters and spaces | \ | Escape Character + | Pattern Modifiers | | |
| [[:word:]] | Digits, letters and underscore | \n | New line + | g | Global match | |
| Assertions | | \r | Carriage return + | i | Case-insensitive | |
| ?= | Lookahead assertion + | \t | Tab + | m | Multiple lines | |
| ?! | Negative lookahead + | \v | Vertical tab + | s | Treat string as single line | |
| ?<= | Lookbehind assertion + | \f | Form feed + | x | Allow comments and white space in pattern | |
| ?!= or ?<! | Negative lookbehind + | \a | Alarm | e | Evaluate replacement | |
| ?> | Once-only Subexpression | [\b] | Backspace | U | Ungreedy pattern | |
| ?() | Condition [if then] | \e | Escape | Metacharacters (must be escaped) | | |
| ?() | Condition [if then else] | \N{name} | Named Character | ^ | [| . |
| ?# | Comment | String Replacement (Backreferences) | | \$ | { | * |
| Note <i>Items marked + should work in most regular expression implementations.</i> | | \$n | nth non-passive group | \$ | \ | + |
| | | \$2 | "xyz" in /^(abc(xyz))\$/ | (| | ? |
| | | \$1 | "xyz" in /^(?:abc)(xyz)\$/ | < | > | |
| | | \$` | Before matched string | | | |
| | | \$' | After matched string | | | |
| | | \$+ | Last matched string | | | |
| | | \$& | Entire matched string | | | |
| | | \$_ | Entire input string | | | |
| | | \$\$ | Literal "\$" | | | |

Available free from AddedBytes.com

La Récursivité



Application directe du cours

1. Écrire un programme qui calcule la factorielle de n .

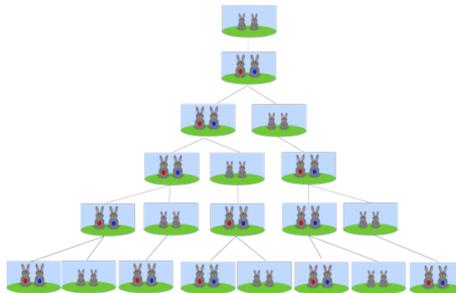
En mathématiques, la factorielle d'un entier naturel n est le produit des nombres entiers strictement positifs inférieurs ou égaux à n . La factorielle est notée !.¹

Exemple : $6! = 6 * 5 * 4 * 3 * 2 * 1$

La factorielle est utilisée pour dénombrer le nombre de permutations possibles de n éléments.

2. Écrire un programme qui calcule les n premiers éléments de la suite de fibonacci.

La suite de Fibonacci est une suite d'entiers dans laquelle chaque terme est la somme des deux termes qui le précèdent. Elle commence généralement par les termes 0 et 1 (parfois 1 et 1) et ses premiers termes sont : 0, 1, 1, 2, 3, 5, 8, 13, 21, etc.²



Application réfléchie

1. Division Euclidienne.

Tant qu'il nous reste dans a une quantité suffisante pour prendre b , on retranche b de a , c'est-à-dire

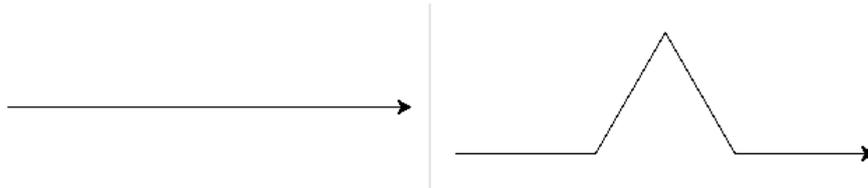
1. <https://fr.wikipedia.org/wiki/Factorielle>

2. https://fr.wikipedia.org/wiki/Suite_de_Fibonacci

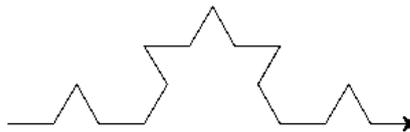
qu'on prend une fois de plus b de a et donc le quotient augmente d'une unité. Lorsqu'on ne peut plus retrancher b de a (parce que $a < b$) alors le reste de la division euclidienne est a .

2. Flocon de von Koch.

Le flocon de von Koch s'obtient en partant d'un segment que l'on partage en 3, la partie du milieu est remplacée par un triangle équilatéral comme ceci :



- Écrire une fonction `ligne` qui construit cet élément.
- Écrire une fonction `triangle` qui construit cet élément (équilatéral).
- Remplacer les `forward` de la fonction `triangle` par un tracé de triangle comme défini précédemment, vous devriez obtenir :



- Étudier le code suivant.

```
#!/usr/bin/python2
# -*- coding: UTF-8 -*-

# programme principal
from turtle import *

def segment (l,n):
    if n==0:
        # tracer jusqu'en B
        forward (l)
    else:
        # dessiner un triangle de équilatéral longueur l/3
        segment (l/3,n-1)
        left (60)
        segment (l/3,n-1)
        right (120)
        segment (l/3,n-1)
        left (60)
        segment (l/3,n-1)

def flocon (l,n):
    segment (l,n)
    right(120)
    segment (l,n)
    right(120)
    segment (l,n)

etape=int(raw_input("Donnez le nombre d'étapes (n) : "))
taille=float(raw_input("Donnez la longueur du côté initial (l) : "))

exitonclick()
```

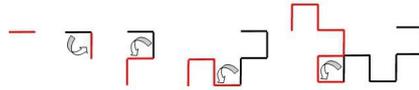
Application avancée

Fractale du dragon

Remarque

✓ 26/11/2023 - La courbe du dragon a fait l'objet d'un article sur mon site <https://ericberthomier.fr/spip.php?article308>, n'hésitez pas à vous y référer pour mieux comprendre ce programme.

La courbe du dragon se construit ainsi :



- Si $t = 0$, l'ordinateur doit dessiner une ligne. C'est la base (ou l'initiateur). La longueur a peu d'importance. On définit la longueur une fois avec s .
- Sinon, si $t > 0$: $\text{Dragon}(t) = \text{Dragon}(t-1) \overrightarrow{\text{Dragon}(t-1)}$. C'est la règle de récursivité (ou le générateur). L'ordinateur doit dessiner une courbe de dragon avec profondeur de récursion $t-1$. Cela donne :

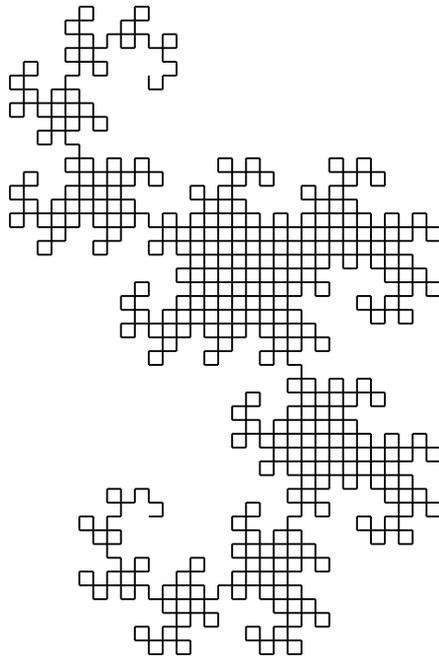
- Dessiner $\text{Dragon}(t-1)$
- Tourner à gauche (90°)
- Dessiner $\text{Dragon}(t-1)$

Il y a un petit problème : on ne peut pas dessiner $\text{Dragon}(t-1)$ exactement de la même façon les deux fois. En effet, le premier $\text{Dragon}(t-1)$ est dessiné vers l'extérieur en partant du milieu de $\text{Dragon}(t)$. Ensuite on tourne de 90° . Le deuxième $\text{Dragon}(t-1)$ est dessiné à l'inverse du milieu de $\text{Dragon}(t)$ vers l'extérieur. Pour que les deux $\text{Dragon}(t-1)$ soit représentée de la même façon, le deuxième $\text{Dragon}(t-1)$ doit être dessiné en miroir. Cela veut dire que tous les angles (a) sont en miroir et (b) doivent être dessinés dans l'ordre inverse.

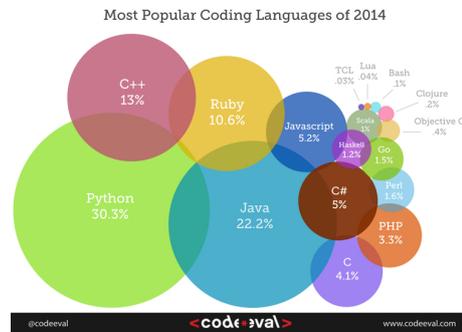
L'astuce consiste à donner un signe qui indique le sens ($vz = 1$ veut dire « + », $vz = -1$ veut dire « - »). On dessine d'abord un $\text{Dragon}(t-1)$ avec signe positif ($vz = 1$). Ensuite on tourne de 90° et dessinons un $\text{Dragon}(t-1)$ avec signe négatif ($vz = -1$).

- Dessiner $\text{Dragon}(t-1)$ signe (+)
- Tourner à gauche ($vz \cdot 90^\circ$)
- Dessiner $\text{Dragon}(t-1)$ signe (-)

Écrire le programme.



Far Away ...



Préambule

Aucun cours n'est associé à ces exercices ...

Les exercices précédent vous ont donné les bases nécessaires pour un apprentissage approfondi du Python, vous trouverez donc des codes à comprendre, étudier et enfin à adapter.

Map / Dictionaries

Cette fonctionnalité appelée aussi Table de Hashage permet d'associer un champs à un autre champs.
Exemple : eric → berthomier

```
#!/usr/bin/python3
# -*- coding: UTF-8 -*-

favorite_sports = {
    'Ralph Williams' : 'Football',
    'Michael Tippett' : 'Basketball',
    'Edward Elgar' : 'Baseball',
    'Rebecca Clarke' : 'Netball',
    'Ethel Smyth' : 'Badminton',
    'Frank Bridge' : 'Rugby'
}

print(favorite_sports['Rebecca Clarke'])

del favorite_sports['Ethel Smyth']
print(favorite_sports)

favorite_sports['Ralph Williams'] = 'Ice Hockey'
print(favorite_sports)

print ("Sport favori de Rebecca Clarke : ",favorite_sports.get('Rebecca Clarke', "Non trouvé"))
print ("Sport favori de Eric Berthomier : ",favorite_sports.get('Eric Berthomier', "Non trouvé"))

if "Eric Berthomier" in favorite_sports:
    print ("Eric Berthomier est dans la liste")
else:
    print ("Eric Berthomier n'est pas dans la liste")
```

Exercice ¹

- Choisissez 5 mots de la langue française et créez un dictionnaire qui associe à chacun de ces mots sa traduction en anglais.
- Ajoutez une entrée au dictionnaire de la question précédente (un nouveau mot et sa définition).
- Écrivez une fonction `ajoute(mot1, mot2, d)` qui prend en argument un mot en français, sa traduction en anglais et ajoute ces deux mots dans le dictionnaire `d` uniquement si `mot1` n'est pas une clé du dictionnaire.
- Écrivez une fonction qui affiche à l'écran toutes les valeurs correspondant aux clés qui sont dans votre dictionnaire (ici, tous les mots en anglais qui apparaissent dans votre dictionnaire).
Indication : on exécute une boucle `for` sur tous les éléments de `d.keys()` et l'on renvoie pour chacun la valeur qui lui est associée.
- Écrivez une fonction `supprime(car, d)` qui prend en argument un caractère `car` et un dictionnaire `d` et supprime du dictionnaire toutes les entrées correspondant à des clés qui commencent par la lettre `c`.

CSV

Ce module permet de lire / créer un fichier CSV (Comma-separated values).

```
#!/usr/bin/python2
# -*- coding: UTF-8 -*-

import csv

# Transformation du fichier CSV en Dictionnaire
file_dico = csv.DictReader(open("exemple.csv"))

# Affichage du dictionnaire
print (file_dico)

# Affichage du contenu du dictionnaire
for row in file_dico:
    print(row['français'], row['anglais'])
```

```
français,anglais
souris,mouse
cameleon,chameleon
chien,dog
vache,cow
epee ,sword
licorne,unicorn
aigle,eagle
dragon,dragon
nain,dwarf
dragon,smaug
noyau,kernel
epee ,stormbringer
tigre,tiger
```

Exercice

Modifier le code précédent pour créer un fichier `c.csv` ne contenant que les mots français commençant par `c`. On prendra soin à entourer de guillemets les champs pour les protéger.

Tk

Ce module permet de créer des interfaces graphiques à l'aide de composants appelés Widgets.

1. Extrait de la page personnelle de Victor Poupet <http://www2.lirmm.fr/~poupet/>

```
#!/usr/bin/python3
# -*- coding: UTF-8 -*-

from tkinter import *

def traceDroite():
    #- Tracé d'une ligne dans le canevas c
    # coordonnées de la ligne
    fenetre.create_line(10,190,190,10,width=2,fill="green")

#----- Programme principal -----#
#- Création de la fenetre :
fen = Tk()

# création des widgets (composants) :
fenetre = Canvas(fen,bg='grey',height=200,width=200)

fenetre.pack(side=LEFT)

btn1 = Button(fen,text='Quitter',command=fen.quit)
btn1.pack(side=BOTTOM)

btn2 = Button(fen,text='Tracer une ligne',command=traceDroite)
btn2.pack()

fen.mainloop() # boucle en attente d'événements
fen.destroy() # destruction (fermeture) de la fenêtre
```

Exercice

Modifier le code précédent pour ajouter un bouton qui crée une ligne de taille, de position et de couleur aléatoire.

PythonPlotting (Non présent en standard)

Ce module permet de créer des graphes.

```
#!/usr/bin/python3
# -*- coding: UTF-8 -*-

import numpy as np
import pylab as pl

# Création d'un tableau de valeur en x
x = [1, 2, 3, 4, 5]

# Création d'un tableau de valeur en y en adéquation avec les x
y = [1, 4, 9, 16, 25]

# Utilisation de pylab pour le tracé en mémoire
pl.plot(x, y)

# Affichage du résultat
pl.show()
```

Exercices

1. Modifier le code précédent pour ajouter une nouvelle courbe.

Culture Générale

1. Un langage de programmation porte le même nom qu'une pierre précieuse, quel est il?