

Livret d'activités pour la seconde professionnelle

Conformément au BO Spécial N°5 du 11 Avril 2019

Consolidation des acquis de cycle 4, bivalence, co-intervention, diversité des activités, travail expérimental ou numérique, algorithmique & programmation, démarche par compétences.



Jérôme Lenoir

SOMMAIRE

Fiche N°1 : Outil de calcul	page 3
Fiche N°2 : Loi d'Ohm	page 7
Fiche N°3 : Lois de la réfraction	page 10
Fiche N°4 : Audiences audiovisuelles	page 13
Fiche N°5 : Énergie cinétique	page 16
Fiche N°6 : Gestion d'un stock	page 19
Fiche N°7 : Dimensions d'une boîte de conserve	page 22
Fiche N°8 : Le jeu des couleurs	page 25
Fiche N°9 : Épures de charpente	page 28
Fiche N°10 : Fréquence d'apparition d'une lettre	page 31

Modules : Calculs commerciaux et financiers / Co-intervention

Niveau : Seconde professionnelle

TI-83 Premium CE



Outil de calcul

Jérôme LENOIR

Objectifs et capacités visées

- Traduire un programme de calcul à l'aide d'une fonction en Python.
- Calculer le montant d'un intérêt simple, d'une valeur acquise, d'un taux annuel de placement, d'une durée de placement (exprimée en jours, quinzaines, mois ou années), le montant du capital placé.

Compétences travaillées



Élaborer un algorithme



Mettre en œuvre un algorithme



Rendre compte d'un résultat et expliquer une démarche

Compétences et savoirs professionnels associés

- **Compétences**
 - Proposer des produits ou des services associés
 - Assurer la veille commerciale
- **Savoirs associés**
 - Outils d'aide à la vente
 - Prix, marges

Situation déclenchante

Lors des TP proposés en enseignement professionnel, une élève est amenée à calculer de manière récurrente différentes grandeurs financières comme le prix TTC connaissant le prix HT, un intérêt simple et la valeur acquise par un capital donné, etc. Pour gagner du temps, elle souhaite créer un outil qui lui permettrait de faire les calculs directement avec sa calculatrice.

Problématique

Pourriez-vous l'aider à réaliser cet outil de calcul ?

Proposer une méthode qui permettrait de répondre à la problématique



Appeler le professeur

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée

Modules : Calculs commerciaux et financiers / Co-intervention

Niveau : Seconde professionnelle

TI-83 Premium CE



Jérôme LENOIR

Fiche méthode

Préambule

L'objectif de cette première activité est de vous familiariser avec l'environnement Python et de réaliser votre premier script à l'aide de la **TI-83 Premium CE** connectée à un adaptateur Python ou la **TI83 Premium CE ÉDITION PYTHON**.

Un script est un programme qui peut contenir un certain nombre de fonctions. Dans notre exemple, nous allons créer en tout quatre fonctions qui vont renvoyer :

- le montant d'un prix TTC connaissant le prix HT (**variable PHT**) et le taux de TVA (**variable t**) ;
- le montant d'un intérêt simple et la valeur acquise par un capital (**variable C**) durant un certain nombre d'années (**variable n**) à un taux donné (**variable t**) ;
- le taux d'un placement connaissant le capital initial (**variable C**) et la valeur qu'il a acquise (**variable Va**) pendant un certain nombre d'années (**variable n**) ;
- la durée de placement pour qu'un capital (**variable C**) puisse atteindre une certaine valeur acquise (**variable Va**) lors d'un placement à un taux donné (**variable t**). On précisera dans cette fonction un argument complémentaire permettant d'indiquer si la capitalisation est quotidienne, par quinzaine, mensuelle ou annuelle



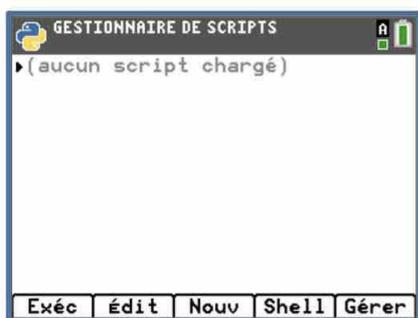
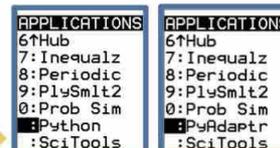
```
def PTTC(PHT, t):
```

```
def Is_Va(C, t, n):
```

```
def taux(Va, C, n):
```

```
def duree(Va, t, C, unite):
```

Pour accéder à l'application Python, appuyer sur les touches :



Nous allons créer un script Python appelé **OUTILS**. Pour cela, appuyer sur la touche **zoom** :

Le verrouillage alphanumérique est activé en majuscule comme l'indique le symbole ci-contre :



Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée

Modules : Calculs commerciaux et financiers / Co-intervention

Niveau : Seconde professionnelle

TI-83 Premium CE



Jérôme LENOIR

Fiche méthode

Algorithme et premier script

Nous allons ici étudier l'algorithme relatif à la fonction **duree** précédemment évoquée :

Fonction *duree* (*V_a*, *t*, *C*, *unite*) :

Si *unite* = « mois » alors
u ← 12
 Sinon si *unite* = « quinzaines » alors
u ← 24
 Sinon si *unite* = « jours » alors
u ← 360
 Sinon si *unite* = « années » alors
u ← 1
 Fin Si

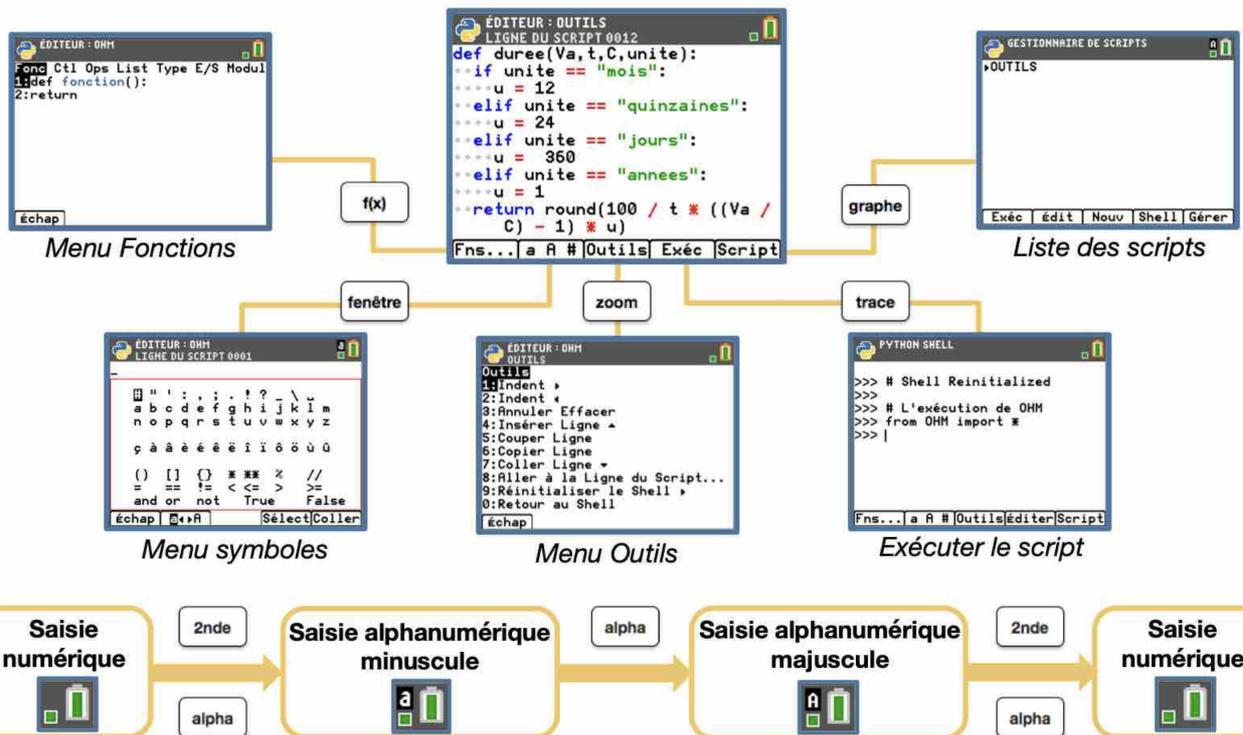
Renvoyer une valeur arrondie à l'unité du calcul $\frac{100 \times (\frac{V_a}{C} - 1)}{t} \times u$

Fin

```

PYTHON SHELL
à alpha
>>> duree(752,8,400,"annees")
11
>>> duree(613.85,5,600,"quinzaines")
11
>>> duree(355.83,2.5,350,"mois")
8
>>> duree(1000.82,3.2,1000,"jours")
9
>>> |
    
```

Utiliser l'environnement Python



Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
 © Texas Instruments 2019 / Photocopie autorisée

Modules : Calculs commerciaux et financiers / Co-intervention

Niveau : Seconde professionnelle

TI-83 Premium CE



Jérôme LENOIR

Fiche méthode

Remarques

Bon nombre d'instructions disponibles dans les sous-menus évoqués ci-dessus sont également accessibles via les (séquences de) touches suivantes :

- les bibliothèques qui regroupent les différentes instructions liées à la gestion des nombres aléatoires ou aux opérations mathématiques : math
- les instructions de comparaison et les opérateurs logiques : 2nde math
- les fonctions trigonométriques : trig
- En outre, vous avez la possibilité de saisir directement certaines instructions. À titre d'exemples :

```
Modul
1:math...
2:random...
```

Affecter 3 à la variable a



Tester si la valeur de la variable b vaut 5



Calculer le carré de la valeur de la variable a

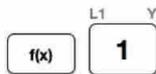


Calculer la valeur de la variable b à la puissance 4



Édition du script

On définit une fonction **PTTC** :



Les tabulations devant les lignes d'instructions sont appelées **indentations**. Elles sont indispensables à la saisie des scripts Python afin d'hierarchiser les instructions.

L'argument **unite** est une chaîne de caractère. Elle est donc à saisir entre guillemets.

```
ÉDITEUR : OUTILS
LIGNE DU SCRIPT 0003
def PTTC(PHT,t):
    return PHT * (1+t/100)
def Is_Va(C,t,n):
    Is = C * t/100 * n
    Va = C + Is
    return Is, Va
def taux(Va,C,n):
    return 100 / n * ((Va / C) - 1)
def duree(Va,t,C,unite):
    if unite == "mois":
        u = 12
    elif unite == "quinzaine":
        u = 24
    elif unite == "jours":
        u = 360
    elif unite == "annees":
        u = 1
    return round(100 / t * ((Va / C) - 1) * u)
Fns... | a | # | Outils | Exéc | Script
```

L'instruction **return** permet de renvoyer le résultat d'une fonction :



L'ensemble des instructions conditionnelles sont disponibles dans le sous-menu **Ctl** prévu

à cet effet : f(x) >

Pour exécuter le script, appuyer sur la touche trace puis sélectionner une des quatre fonctions dans la liste accessible à l'aide de la touche var

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée

Modules : Fonctions, électricité

TI-83 Premium CE

Niveau : Seconde professionnelle

**Loi d'Ohm**

Jérôme LENOIR

Objectifs et capacités visées

- Traduire un programme de calcul à l'aide d'une fonction en Python.
- Déterminer l'équation réduite d'une droite non parallèle à l'axe des ordonnées.
- Reconnaître une situation de proportionnalité et déterminer la fonction linéaire qui la modélise.
- Lire et représenter un schéma électrique.
- Mesurer l'intensité d'un courant électrique et la tension aux bornes d'un dipôle.

Compétences travaillées

- | | | |
|--|--|---|
| <div style="background-color: red; color: white; padding: 5px; border-radius: 5px; display: inline-block; margin-right: 10px;">C1</div> Rechercher, extraire et organiser des informations | <div style="background-color: blue; color: white; padding: 5px; border-radius: 5px; display: inline-block; margin-right: 10px;">C2</div> Élaborer un algorithme | <div style="background-color: green; color: white; padding: 5px; border-radius: 5px; display: inline-block; margin-right: 10px;">C3</div> Mettre en œuvre un algorithme |
| <div style="background-color: orange; color: white; padding: 5px; border-radius: 5px; display: inline-block; margin-right: 10px;">C4</div> Exploiter et interpréter les résultats obtenus afin de répondre à une problématique | <div style="background-color: purple; color: white; padding: 5px; border-radius: 5px; display: inline-block; margin-right: 10px;">C5</div> Rendre compte d'un résultat et expliquer une démarche | |

Situation déclenchante

En application d'une séance de cours sur les fonctions affines, l'enseignant de mathématiques & physique-chimie propose une séance de travaux pratiques en électricité sous forme de challenge. Il fournit à ses élèves un résistor dont il a caché la valeur de la résistance, et leur demande de déterminer cette valeur à l'aide d'une partie du matériel mis à leur disposition (sachant qu'un ohmmètre n'en fait pas partie).

Problématique**Pourriez-vous déterminer la valeur cachée de la résistance ?**

Proposer une méthode qui permettrait de répondre à la problématique

**Appeler le professeur**

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée

Modules : Fonctions, électricité

TI-83 Premium CE

Niveau : Seconde professionnelle



Fiche méthode

Jérôme LENOIR

Préambule

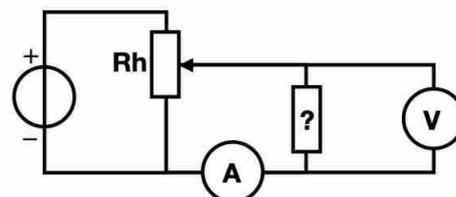
L'objectif de cette activité est double :

- développer chez l'apprenant une partie des capacités et connaissances du module d'électricité de manière transversale tout en promouvant la bivalence avec l'enseignement des mathématiques, comme le préconisent les nouveaux programmes de la voie professionnelle (BO spécial N°5 du 11 Avril 2019) ;
- utiliser l'algorithmique comme outil permettant de répondre à la problématique posée, une fois l'expérimentation scientifique réalisée.

Il n'est bien évidemment pas question ici de présenter le contenu de cette séquence dans sa globalité, mais plutôt de faire un focus sur le second point précédemment évoqué.

À titre indicatif, les tableaux ci-contre présentent les résultats de l'expérimentation réalisée par un des binômes ainsi que le schéma du dispositif électrique illustrant la situation.

Afin de prendre toutes les valeurs expérimentales en compte dans notre étude, nous utiliserons la méthode de Mayer. Cette méthode n'étant pas au programme de seconde professionnelle, il conviendra de ne pas la développer. Néanmoins elle permet de travailler les automatismes chez l'apprenant en réinvestissant par exemple le calcul de la moyenne de plusieurs valeurs numériques.



U (V)	1,02	2,01	3,03
I (mA)	3,11	6,15	9,24

U (V)	4,00	5,05	6,04
I (mA)	12,2	15,3	18,3

Algorithme

Nous allons donc créer un algorithme puis un script qui permet de trouver l'expression d'une fonction affine (sous la forme $y = m \cdot x + p$) passant par deux points de coordonnées respectives (a ; b) et (c ; d) dans un repère donné :

Fonction *equa_red* (a , b , c , d) :

```

Si a = c alors
    Afficher « Droite parallèle à (Oy) »
Sinon
    m ← (d - b) / (c - a)
    p ← b - m * a
Fin Si
Renvoyer m et p
Fin
    
```

```

PYTHON SHELL
>>> # Shell Reinitialized
>>>
>>> # L'exécution de OHM
>>> from OHM import *
>>> equa_red(1,3,5,-1)
(-1.0, 4.0)
>>> |
    
```

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
 © Texas Instruments 2019 / Photocopie autorisée



Modules : Fonctions, électricité

TI-83 Premium CE

Niveau : Seconde professionnelle



Fiche méthode

Jérôme LENOIR

Édition du script

La fonction `equa_red` détermine l'équation réduite de la droite passant par les points de coordonnées $(a ; b)$ et $(c ; d)$.

```

ÉDITEUR : OHM
LIGNE DU SCRIPT 0008
def equa_red(a,b,c,d):
    if a == c:
        print("Droite parallèle
à 0y")
    else:
        m = (d-b)/(c-a)
        p = b - m * a
    return m, p
  
```

Si les deux points étudiés ont la même abscisse, alors la droite est parallèle à l'axe des ordonnées.

Pour la droite passant par les deux points étudiés, on calcule la valeur : m du coefficient directeur et p de l'ordonnée à l'origine.

Conclusion

Le tableau ci-contre indique les valeurs moyennes calculées pour chacun des deux tableaux présentés en préambule. Une fois le script saisi, exécutez-le à l'aide de la touche `trace`, puis sélectionner la fonction `equa_red` à l'aide de la touche `var`.

U_m (V)	2,02	5,03
I_m (mA)	6,17	15,3

```

PYTHON SHELL
>>> # Shell Reinitialized
>>>
>>> # L'exécution de OHM
>>> from OHM import *
>>> equa_red(0.00617,2.02,0.0153
,5.03)
(329.6823658269442, -0.014140197
1522459)
>>> |
  
```

Dans notre cas, la valeur de l'ordonnée à l'origine est proche de zéro. Nous pouvons donc considérer que la fonction $U = f(I)$ peut être modélisée par une fonction linéaire de coefficient directeur ayant pour valeur (arrondie à l'unité) 330. Cette valeur correspond à la valeur en ohm de la résistance recherchée.

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée



Modules : Fonctions, optique

TI-83 Premium CE

Niveau : Seconde professionnelle

**Loi de la réfraction**

Jérôme LENOIR

Capacités et connaissances visées

- Déterminer un angle limite de réfraction.
- Connaître les lois de la réfraction.
- Savoir que la réfringence d'un milieu est liée à la valeur de son indice de réfraction.
- Connaître la condition d'existence de l'angle limite de réfraction et du phénomène de réflexion totale.

Compétences travaillées

C1 Traduire des informations, des codages



C2 Élaborer un algorithme



C3 Mettre en œuvre un algorithme

Situation déclenchante

Dans le cadre de sa séquence d'optique, l'enseignant de mathématiques & physique-chimie demande à ses élèves de concevoir un outil qui leur permettrait de déterminer très rapidement :

- l'angle limite de réfraction dans une situation mettant en jeu deux milieux transparents homogènes dont les indices de réfraction sont connus ;
- l'indice de réfraction d'un milieu connaissant les angles d'incidence et de réfraction ainsi que l'indice de réfraction du milieu d'incidence.

Problématique**Peux-tu aider ces élèves à concevoir cet outil ?**

Proposer une méthode qui permettrait de répondre à la problématique



Appeler le professeur

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée



Modules : Fonctions, optique

TI-83 Premium CE

Niveau : Seconde professionnelle



Fiche méthode

Jérôme LENOIR

Proposition de démarche de résolution

Nous allons créer un algorithme puis un script regroupant deux fonctions qui vont répondre à la problématique.

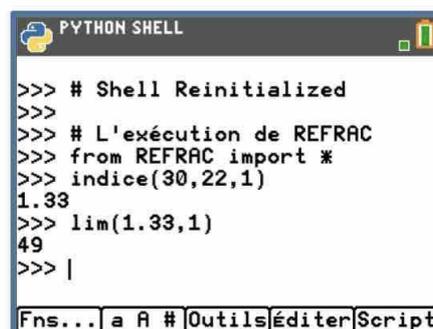
```
def lim(n1,n2):
```

```
def indice(i1,i2,n1):
```

Algorithme

```
Fonction lim( n1, n2 ) :
    Si n1 < n2 alors
        Renvoyer « Pas de réflexion totale »
    Sinon
        Calculer et renvoyer l'angle limite de réfraction
    Fin Si
Fin
```

```
Fonction indice ( i1, i2, n1 ) :
    Calculer et renvoyer n1 * sin( i1 ) / sin( i2 )
Fin
```

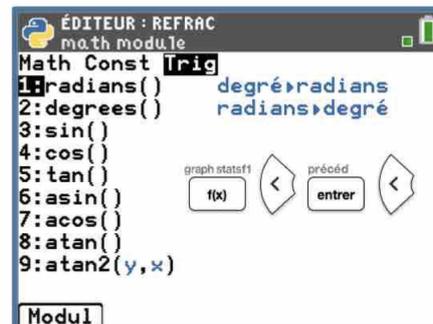


Remarques

Afin de créer ce script, nous aurons besoin d'utiliser les fonctions **SINUS (sin)** et **ARCSINUS (asin)**. L'unité d'angle de référence pour ces deux fonctions étant le radian, deux autres instructions sont nécessaires :

- **degrees()** qui permet de convertir une valeur initialement exprimée en radians vers une valeur en degrés ;
- **radian()** qui permet de convertir une valeur initialement exprimée en degrés vers une valeur en radians.

Ces différentes fonctions et instructions sont disponibles dans la bibliothèque **MATH** qu'il est donc impératif d'importer en début de script (à défaut d'importer une à une chacune d'entre elles) à l'aide de la séquence de touches ci-contre :



Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
 © Texas Instruments 2019 / Photocopie autorisée

Modules : Fonctions, optique

TI-83 Premium CE

Niveau : Seconde professionnelle



Fiche méthode

Jérôme LENOIR

Édition du script

```

ÉDITEUR : REFRAC
LIGNE DU SCRIPT 0002
from math import *

def lim(n1,n2):
  if n1 < n2:
    return "Pas de reflexion totale"
  else:
    return round(degrees(asin(n2/n1)))

def indice(i1,i2,n1):
  return round(n1*sin(radians(i1))/sin(radians(i2)),2)
  
```

Toute chaîne de caractère doit être saisie entre des guillemets :



L'instruction **round** renvoie une valeur arrondie à l'unité d'un nombre réel. Dans l'exemple ci-dessous, la fonction **arr** renvoie une valeur de π arrondie au centième.

```
def arr():
  return round(pi,2)
```

Conclusion

```

PYTHON SHELL
VARS : REFRAC
> indice()
lim()
  
```

Pour exécuter le script, appuyer sur la touche **trace** puis sélectionner une des deux fonctions dans la liste accessible à l'aide de la touche **var**.

Dans les cas exposés ci-contre :

- Le milieu de réfraction est l'air ($n=1$) lorsque l'angle d'incidence est de 20° dans le verre ($n=1,5$) pour un angle de réfraction est de 31° ;
- Il n'y a pas de réflexion totale lorsqu'un rayon lumineux passe de l'eau ($n=1,33$) au verre ($n=1,5$) ;
- L'angle limite de réfraction est de 42° lorsqu'un rayon lumineux passe du verre ($n=1,5$) à l'air ($n=1$).

```

PYTHON SHELL
>>> # Shell Reinitialized
>>>
>>> # L'exécution de REFRAC
>>> from REFRAC import *
>>> indice(20,31,1.5)
1.0
>>> lim(1.33,1.5)
'Pas de reflexion totale'
>>> lim(1.5,1)
42
>>> |
  
```

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée



Modules : Probabilités

TI-83 Premium CE

Niveau : Seconde professionnelle

**Audiences audiovisuelles**

Jérôme LENOIR

Objectifs et capacités visées

- Modifier une simulation donnée pour percevoir une version vulgarisée de la loi des grands nombres.
- Estimer la probabilité non triviale d'un événement à partir des fréquences.
- Faire preuve d'esprit critique face à une situation aléatoire simple.

Compétences travaillées

<div style="background-color: red; color: white; padding: 5px; border-radius: 5px; display: inline-block;">C1</div> Rechercher, extraire et organiser des informations	<div style="background-color: blue; color: white; padding: 5px; border-radius: 5px; display: inline-block;">C2</div> Élaborer un algorithme	<div style="background-color: green; color: white; padding: 5px; border-radius: 5px; display: inline-block;">C3</div> Mettre en œuvre un algorithme
<div style="background-color: orange; color: white; padding: 5px; border-radius: 5px; display: inline-block;">C4</div> Exploiter et interpréter les résultats obtenus afin de répondre à une problématique	<div style="background-color: purple; color: white; padding: 5px; border-radius: 5px; display: inline-block;">C5</div> Rendre compte d'un résultat et expliquer une démarche	

Situation déclenchante

Afin d'évaluer les audiences des émissions de télévision, Médiamétrie utilise un panel de 5000 foyers soit un total de 11 400 individus représentatifs de la population française. Un salarié de Médiamétrie souhaite évaluer la part d'audience obtenue par une émission en première partie de soirée.

Problématique**Peux-tu aider ce salarié à estimer cette part d'audience ?**

Proposer une méthode qui permettrait de répondre à la problématique

**Appeler le professeur**

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée



Modules : Probabilités

TI-83 Premium CE

Niveau : Seconde professionnelle



Fiche méthode

Jérôme LENOIR

Proposition de démarche de résolution

Nous allons créer un algorithme puis un script permettant de :

- générer un nombre entier aléatoire **nb** compris entre 0 à 11400, correspondant au nombre d'individus regardant l'émission de télévision étudiée ;
- simuler la situation en fonction de la taille **n** de l'échantillon étudié ;
- évaluer la part d'audience à l'aide des fréquences obtenues lors des différentes simulations en créant une fonction **media**.

```
nb = randint(0,11400)
```

```
def media(n)
```

Algorithme

$nb \leftarrow$ nombre aléatoire entier entre 0 et 11 400

Fonction $media(n)$:

$s \leftarrow 0$

Pour i allant de 1 à n

$a \leftarrow E(\text{nombre aléatoire} + nb/11\ 400)$

$s \leftarrow s + a$

Fin Pour

Renvoyer s/n

Fin

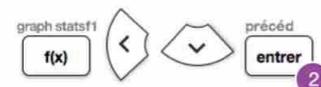
```

PYTHON SHELL
>>> # Shell Reinitialized
>>>
>>> # L'exécution de MEDIA
>>> from MEDIA import *
>>> nb
5802
>>> media(1000)
0.502
>>> |
  
```

Remarques

Le script que nous allons écrire nécessite d'importer deux bibliothèques (accessibles à l'aide des séquences de touches ci-contre) :

- la bibliothèque **RANDOM** qui regroupe les différentes instructions liées à la gestion des nombres aléatoires ;
- la bibliothèque **MATH** qui contient, entre autres, l'instruction **FLOOR** permettant d'extraire la partie entière d'un nombre.



Les deux bibliothèques étant importées dans le script, nous allons maintenant saisir les différentes instructions relatives à l'algorithme précédent.

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée



Modules : Probabilités

TI-83 Premium CE

Niveau : Seconde professionnelle



Fiche méthode

Jérôme LENOIR

Édition du script

Ajouter un niveau d'indentation :



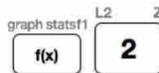
Pour supprimer un niveau d'indentation :



Pour i allant de 0 à n-1



Retourner le nombre correspondant à la moyenne des fréquences pour les n individus d'un panel.



```

ÉDITEUR : MEDIA
LIGNE DU SCRIPT 0011
from math import *
from random import *

nb = randint(0,11400)

def media(n):
    s=0
    for i in range(n):
        a=floor(random()+nb/11400)
        s+=a
    return s/n
    
```

Générer un nombre aléatoire entier entre deux valeurs :



Extraire la partie entière ...



... d'un nombre aléatoire décimal compris entre 0 et 1.



Incrémenter la variable s de la valeur a. Deux écritures sont possibles :

```

s += a
s = s + a
    
```

Conclusion

```

PYTHON SHELL
>>> # L'exécution de MEDIA
>>> from MEDIA import *
>>> media(10)
0.6
>>> media(100)
0.3
>>> media(1000)
0.319
>>> media(10000)
0.3453
>>> |
    
```

Pour exécuter le script, appuyer sur la touche `trace` puis sélectionner la fonction `media` dans la liste accessible à l'aide de la touche `var`.

Dans le cas exposé ci-contre, on peut estimer la part d'audience de l'émission étudiée aux alentours de 0,3453 soit environ 34,53 % de part de marché.

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus

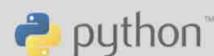


Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
 © Texas Instruments 2019 / Photocopie autorisée

Modules : Fonctions

TI-83 Premium CE

Niveau : Seconde professionnelle

**Énergie cinétique**

Jérôme LENOIR

Objectifs et capacités visées

- Traduire un programme de calcul à l'aide d'une fonction en Python.
- Rechercher un encadrement ou une valeur approchée d'une solution d'une équation du type $f(x) = c$ par balayage sur un intervalle donné (où c est un nombre réel donné et f une fonction du type $x \mapsto kx^2$ avec k un nombre réel donné)
- Calculer les images de nombres par une fonction.

Compétences travaillées

- | | | |
|---|---|---|
| <div style="background-color: #e91e63; color: white; padding: 5px; border-radius: 5px; display: inline-block; margin-bottom: 5px;">C1</div> Rechercher, extraire et organiser des informations | <div style="background-color: #0070c0; color: white; padding: 5px; border-radius: 5px; display: inline-block; margin-bottom: 5px;">C2</div> Élaborer un algorithme | <div style="background-color: #70ad47; color: white; padding: 5px; border-radius: 5px; display: inline-block; margin-bottom: 5px;">C3</div> Mettre en œuvre un algorithme |
| <div style="background-color: #ffc000; color: white; padding: 5px; border-radius: 5px; display: inline-block; margin-bottom: 5px;">C4</div> Exploiter et interpréter les résultats obtenus afin de répondre à une problématique | <div style="background-color: #6a3d9a; color: white; padding: 5px; border-radius: 5px; display: inline-block; margin-bottom: 5px;">C5</div> Rendre compte d'un résultat et expliquer une démarche | |

Situation déclenchante

Lorsqu'un véhicule est en mouvement, il possède une quantité d'énergie cinétique (exprimée en joule) qui dépend à la fois de sa masse (en kg) et de sa vitesse (en m/s) selon la formule :

$$E_c = \frac{1}{2} \times m \times v^2$$

Afin d'illustrer son cours de physique-chimie, une enseignante présente un exemple dans lequel elle évoque l'importance de limiter l'énergie cinétique de son véhicule (dont la masse est de 1,62 tonne) à 400 kJ lorsqu'elle rentre chez elle.

Problématique**Pourquoi l'enseignante a-t-elle limité cette valeur à 400 kJ ?**

Proposer une méthode qui permettrait de répondre à la problématique

**Appeler le professeur**

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée



Modules : Fonctions

TI-83 Premium CE

Niveau : Seconde professionnelle



Fiche méthode

Jérôme LENOIR

Préambule

Afin de simplifier sa rédaction en vue d'une exploitation par les élèves, le script que nous allons créer se limitera à l'étude d'une fonction strictement croissante sur un intervalle $[a ; b]$. En effet, la situation proposée étudie l'évolution de l'énergie cinétique d'un véhicule en fonction de sa vitesse, ce qui entre tout à fait dans cette optique pour une vitesse comprise entre 0 et 130 km/h.

Néanmoins, il est tout à fait envisageable de proposer le cas d'une fonction décroissante sur un intervalle donné dans un nouveau contexte qui pourrait être lié à l'enseignement professionnel dans le cadre de la co-intervention ou à des fins de différenciation pédagogique, de remédiation ou d'approfondissement dans le cadre du module d'accompagnement renforcé.

Proposition de démarche de résolution

Nous allons créer un algorithme puis un script :

- créant une fonction f correspondant à l'énergie cinétique du véhicule en fonction de sa vitesse v ;
- donnant un encadrement au $1/100^e$ de la solution de l'équation d'inconnue x , $f(x) = c$, c'est-à-dire l'antécédent du nombre c par la fonction f .

```
def f(v):
    return 0.5*1620*v**2
```

```
def antecedent(a,b,c)
```

Algorithme

L'algorithme suivant permet de résoudre l'équation $f(v) = c$ d'inconnue v où f est une fonction strictement croissante sur l'intervalle $[a ; b]$ et c un nombre réel appartenant à cet intervalle.

Fonction *antecedent*(a, b, c) :

Tant que $b - a > 0,01$

$m \leftarrow (a + b) / 2$

Si $f(m) > c$ alors

$b \leftarrow m$

Sinon

$a \leftarrow m$

Fin Si

Fin Tant que

Renvoyer a et b

Fin

```
PYTHON SHELL
>>> # Shell Reinitialized
>>>
>>> # L'exécution de EC
>>> from EC import *
>>> f(10)
81000.0
>>> antecedent(15,25,400000)
(22.216796875, 22.2265625)
>>> |
Fns... a A # Outils Éditer Script
```

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée



Modules : Fonctions

TI-83 Premium CE

Niveau : Seconde professionnelle



Fiche méthode

Jérôme LENOIR

Édition du script

La fonction **antecedent** détermine la solution de l'équation : $f(v) = c$ sur l'intervalle $[a; b]$.

La fonction f modélise l'évolution de l'énergie cinétique de la voiture de l'enseignante en fonction de sa vitesse.

```

ÉDITEUR : EC
LIGNE DU SCRIPT 0003
def f(v):
    return 0.5*1620*v**2

def antecedent(a,b,c):
    while (b - a) > 0.01:
        m = (a + b)/2
        if f(m) > c:
            b = m
        else:
            a = m
    return a,b
  
```

Les opérateurs de comparaison sont accessibles dans le sous-menu **Ops** :



La fonction renvoie la valeur des variables **a** et **b** représentant un encadrement de la valeur de la vitesse recherchée dans la situation.

Conclusion

Une fois le script saisi, exécutez-le à l'aide de la touche **trace**, puis sélectionner la fonction **antecedent** à l'aide de la touche **var**.

```

PYTHON SHELL
>>> # Shell Reinitialized
>>>
>>> # L'exécution de EC
>>> from EC import *
>>> antecedent(0,36.11,400000)
(22.21611328125, 22.22492919921875)
>>> 22.2*3.6
79.92000000000001
>>> |
  
```

En France, la vitesse autorisée variant entre 0 et 130 km/h (soit entre 0 et environ 36,11 m/s), nous utiliserons donc ces deux paramètres respectivement pour les arguments **a** et **b** de la fonction **antecedent**. L'argument **c** correspond quant à lui à la quantité d'énergie cinétique (exprimée en joule) donnée en énoncé, c'est-à-dire 400 000.

L'énergie cinétique de 400 kJ correspond à une vitesse d'environ 22,2 m/s, soit environ 80 km/h. L'enseignante a choisi cette valeur car elle correspond à la vitesse maximale autorisée sur les routes départementales françaises.

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée



Modules : Fonctions / Co-intervention

TI-83 Premium CE

Niveau : Seconde professionnelle

**Gestion des stocks**

Jérôme LENOIR

Objectifs et capacités visées

- Traduire un programme de calcul à l'aide d'une fonction en Python.
- Déterminer l'image ou des antécédents éventuels d'un nombre par une fonction définie sur un intervalle donné.
- Rechercher un extremum par balayage sur un intervalle donné.

Compétences travaillées

- | | | |
|--|--|---|
| <div style="background-color: red; color: white; border-radius: 50%; width: 30px; height: 30px; display: flex; align-items: center; justify-content: center; margin: 0 auto 10px auto;">C1</div> <p>Rechercher, extraire et organiser des informations</p> | <div style="background-color: blue; color: white; border-radius: 50%; width: 30px; height: 30px; display: flex; align-items: center; justify-content: center; margin: 0 auto 10px auto;">C2</div> <p>Élaborer un algorithme</p> | <div style="background-color: green; color: white; border-radius: 50%; width: 30px; height: 30px; display: flex; align-items: center; justify-content: center; margin: 0 auto 10px auto;">C3</div> <p>Mettre en œuvre un algorithme</p> |
| <div style="background-color: orange; color: white; border-radius: 50%; width: 30px; height: 30px; display: flex; align-items: center; justify-content: center; margin: 0 auto 10px auto;">C4</div> <p>Exploiter et interpréter les résultats obtenus afin de répondre à une problématique</p> | <div style="background-color: purple; color: white; border-radius: 50%; width: 30px; height: 30px; display: flex; align-items: center; justify-content: center; margin: 0 auto 10px auto;">C5</div> <p>Rendre compte d'un résultat et expliquer une démarche</p> | |

Compétences et savoirs professionnels associés

- **Pôle 1 : Gestion administrative des relations externes**
 - 1.1.4 : Évaluation & suivi des stocks
Compétences : Apprécier les stocks en quantité, en valeur et en qualité.
Savoirs associés : méthodes d'évaluation des stocks, inventaire physique et inventaire théorique.
- **Pôle 3 : Gestion administrative interne**
 - 3.1.1 : Collecte et recherche d'informations
Compétences : Exploiter la veille et mobiliser des techniques de recherche.
Savoirs associés : système d'information, nature et formes d'information, modes de communication, gestion électronique des données, concision, reformulation, précision lexicale, hiérarchie des informations.

Situation déclenchante

Lors de sa seconde période de formation en milieu professionnel, le tuteur d'une élève de seconde professionnelle lui demande de faire une analyse sur l'état des stocks de l'entreprise afin de préparer une présentation auprès de ses collègues. La fonction s ci-dessous, définie sur l'intervalle $[5 ; 45]$ modélise l'évolution des stocks :

$$s(n) = 0,08 n^3 - 5,4 n^2 + 84 n + 300 \quad \text{où } n \text{ représente le nombre de commandes passées.}$$

Problématique

Peux-tu aider cette élève à faire une analyse de l'état des stocks ?

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée



Modules : Fonctions / Co-intervention

TI-83 Premium CE

Niveau : Seconde professionnelle



Fiche méthode

Jérôme LENOIR

Proposition de démarche de résolution

Pour répondre à cette problématique, nous allons déterminer une valeur arrondie à l'unité des extremums de la fonction s sur l'intervalle $[5 ; 45]$ en créant un algorithme puis un script contenant deux fonctions :

- la fonction s indiquée dans l'énoncé et renvoyant le nombre de pièces en stock en fonction du nombre n de commandes passées. (L'expression de cette fonction pourra tout à fait être modifiée ultérieurement en fonction de la situation proposée) ;
- la fonction ext qui va déterminer une valeur arrondie à l'unité des extremums de la fonction s sur l'intervalle $[a ; b]$.

```
def s(n):
    return 0.08*n**3-5.4*n**2+84*n
    +300
```

```
def ext(a,b)
```

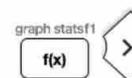
Algorithme

```
Fonction ext ( a , b )
  xmin ← a
  xmax ← b
  min ← s ( a )
  max ← s ( b )
  Tant que a ≤ b
    Si s ( a ) < min alors
      xmin ← a
      min ← s ( a )
    Fin Si
    Si s ( a ) > max alors
      xmax ← a
      max ← s ( a )
    Fin Si
    a ← a + 0,01
  Fin Tant que
  Renvoyer une valeur arrondie à l'unité de xmin, min, xmax, max
Fin
```

```
PYTHON SHELL
>>> # Shell Reinitialized
>>>
>>> # L'exécution de STOCK
>>> from STOCK import *
>>> s(25)
275.0
>>> ext(15,20)
(20, 460, 15, 615)
>>> |
```

Remarques

- Les opérateurs de comparaison et de logique sont regroupés dans un sous-menu **Ops** spécifique accessible à l'aide des touches :
- Les instructions de bouclage (for, while) et les instructions conditionnelles (if, then, elif, else) sont accessibles dans un sous-menu **Ctl** (contrôles) spécifique à l'aide de la séquence suivante :



Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée



Modules : Fonctions / Co-intervention

TI-83 Premium CE

Niveau : Seconde professionnelle



Fiche méthode

Jérôme LENOIR

Édition du script

On saisit la fonction **s** donnée en énoncé qui modélise l'évolution des stocks en fonction du nombre **n** de commandes passées.

```

ÉDITEUR : STOCK
LIGNE DU SCRIPT 0003
def s(n):
    return 0.08*n**3-5.4*n**2+84*n
    +300

def ext(a,b):
    xmin = a
    xmax = b
    min = s(a)
    max = s(b)
    while a <= b:
        if s(a) < min:
            xmin = a
            min = s(a)
        if s(a) > max:
            xmax = a
            max = s(a)
        a=a+0.01
    return round(xmin),round(min),
    round(xmax), round(max)
  
```

On initialise les valeurs **xmin**, **xmax**, **min** et **max**.

Tant que la valeur de la variable **a** est inférieure ou égale à **b**, on traite deux tests (instructions **if**) puis on incrémente de 0,01 la valeur de la variable **a**.

Pour chaque valeur que prend la variable **a**, on teste si son image par la fonction **s** est strictement inférieure à la valeur minimale enregistrée ou strictement supérieure à la valeur maximale enregistrée. On affecte alors, le cas échéant, la valeur de l'abscisse du minimum et du maximum de la fonction **s** sur l'intervalle **[a ; b]** respectivement aux variables **xmin** et **xmax**.

On renvoie les coordonnées arrondies à l'unité du minimum et du maximum de la fonction **s** sur l'intervalle **[a ; b]**.

Conclusion

Une fois le script saisi, exécutez-le à l'aide de la touche `trace`, puis sélectionner la fonction **ext** à l'aide de la touche

`var`.

```

PYTHON SHELL
>>> # Shell Reinitialized
>>>
>>> # L'exécution de STOCK
>>> from STOCK import *
>>> ext(5,45)
(35, 55, 10, 680)
>>> |
  
```

Le stock minimum était de 55 pièces pour un total de 35 commandes et le stock maximum était de 680 pièces pour un total de 10 commandes.

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée



Modules : Fonctions / Co-intervention

TI-83 Premium CE

Niveau : Seconde professionnelle

**Dimensions d'une boîte de conserve**

Jérôme LENOIR

Objectifs et capacités visées

- Traduire un programme de calcul à l'aide d'une fonction en Python.
- Dans le cadre de problèmes modélisés par des fonctions, résoudre une équation du type $f(x) = c$ (où c est un nombre réel donné (les fonctions cube, inverse, et $x \mapsto kx^2$ avec k un nombre réel donné pouvant être évoquées lors de la résolution de problèmes avec le domaine professionnel).
- Calculer les images de nombres par une fonction.
- Calculer des longueurs, des aires, des volumes dans les figures ou solides.

Compétences travaillées

- | | | |
|--|--|---|
| <div style="background-color: red; color: white; border-radius: 50%; width: 30px; height: 30px; display: flex; align-items: center; justify-content: center; margin: 0 auto 10px auto;">C1</div> <p>Rechercher, extraire et organiser des informations</p> | <div style="background-color: blue; color: white; border-radius: 50%; width: 30px; height: 30px; display: flex; align-items: center; justify-content: center; margin: 0 auto 10px auto;">C2</div> <p>Élaborer un algorithme</p> | <div style="background-color: green; color: white; border-radius: 50%; width: 30px; height: 30px; display: flex; align-items: center; justify-content: center; margin: 0 auto 10px auto;">C3</div> <p>Mettre en œuvre un algorithme</p> |
| <div style="background-color: orange; color: white; border-radius: 50%; width: 30px; height: 30px; display: flex; align-items: center; justify-content: center; margin: 0 auto 10px auto;">C4</div> <p>Exploiter et interpréter les résultats obtenus afin de répondre à une problématique</p> | <div style="background-color: purple; color: white; border-radius: 50%; width: 30px; height: 30px; display: flex; align-items: center; justify-content: center; margin: 0 auto 10px auto;">C5</div> <p>Rendre compte d'un résultat et expliquer une démarche</p> | |

Compétences et savoirs professionnels

- **C1** : Rechercher une information dans une documentation technique, en local ou à distance.
- **C2** : Formuler et transmettre des informations, communiquer sous forme écrite ou orale.
- **C4** : Interpréter et vérifier les données de définition de tout ou partie d'un ensemble chaudronné.
- **C5** : Préparer la fabrication de tout ou partie d'un ensemble chaudronné.
- **S21-2** : Coût des moyens matériels.
- **S23-1** : Terminologie des surfaces et volumes, décomposition des surfaces, développement par calcul de volumes simples.

Situation déclenchante

Une entreprise de métallurgie fabrique des boîtes de conserve en vue de fournir une grande marque de légumes. Le bureau d'étude cherche à minimiser la surface de métal nécessaire à la fabrication des deux modèles les plus vendus par la marque de volumes respectifs 212 et 850 cm³.

Pour une boîte de conserve de volume V (exprimée en cm³), la surface de métal nécessaire (exprimée en cm²) est donnée par la fonction S définie sur l'intervalle $[2 ; 10]$ par :

$$S(R) = \frac{2 \times V}{R} + 2\pi \times R^2 \quad \text{où } R \text{ représente le rayon, en cm, de la boîte de conserve.}$$

**Problématique**

Quelles seraient les dimensions des deux modèles pour minimiser la surface de métal nécessaire à leur fabrication ?

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée



Modules : Fonctions / Co-intervention

TI-83 Premium CE

Niveau : Seconde professionnelle



Fiche méthode

Jérôme LENOIR

Proposition de démarche de résolution

Pour répondre à cette problématique, nous allons créer un algorithme puis un script contenant trois fonctions :

- la fonction **hauteur** qui renvoie la hauteur de la boîte de conserve en fonction de son rayon (**variable R**) et de son volume (**variable V**) ;
- la fonction **surface** qui renvoie la surface de métal de la boîte de conserve en fonction de son rayon (**variable R**) et de son volume (**variable V**) ;
- la fonction **dimensions** qui renvoie le rayon et la hauteur de la boîte de conserve en fonction de son volume (**variable V**) et en utilisant le moins de métal possible.

```
def hauteur(R,V):
    return V/(pi*R**2)
```

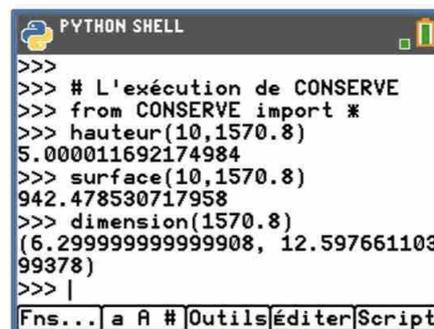
```
def surface(R,V):
    return (2*V)/R + 2*pi*R**2
```

```
def dimensions(V):
```

Algorithme

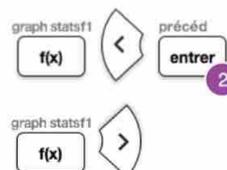
L'algorithme ci-dessous concerne la fonction **dimensions** :

```
Fonction dimensions ( V )
    i ← 2
    R ← 0
    s ← surface ( 2 , V )
    Tant que i < 10
        Si surface ( i , V ) < s alors
            s ← surface ( i , V )
            R ← i
        Fin Si
        i ← i + 0,01
    Fin Tant que
    Renvoyer R et hauteur ( R , V )
Fin
```



Remarques

- Dans notre cas de figure nous avons choisi un pas de 0,01 pour balayer l'intervalle [a ; b] mais il est tout à fait envisageable de modifier cette valeur ;
- Nous allons dans un premier temps importer la bibliothèque **MATH** en saisissant la séquence de touches suivante :
- Les instructions de bouclage et les instructions conditionnelles sont accessibles dans un menu spécifique à l'aide de la séquence suivante :



Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée

Modules : Fonctions / Co-intervention

TI-83 Premium CE

Niveau : Seconde professionnelle



Fiche méthode

Jérôme LENOIR

Édition du script

Cette fonction permet de calculer la surface de métal nécessaire à la confection d'une boîte de conserve connaissant son rayon (en cm) et son volume (en cm³).

Le rayon étant compris entre 2 et 10, on initialise la variable *i* à la valeur 2.

Pour chaque valeur *i* du rayon comprise entre 2 et 10 avec un pas de 0,01 on teste si l'image du nombre *i* par la fonction **surface** est bien strictement inférieure à la valeur de la variable *s*. Si c'est le cas, on affecte cette nouvelle valeur à la variable *s*.

```

ÉDITEUR : CONSERVE
LIGNE DU SCRIPT 0002
from math import pi

def hauteur(R,V):
    return V/(pi*R**2)

def surface(R,V):
    return (2*V)/R + 2*pi*R**2

def dimensions(V):
    i = 2
    R = 0
    s = surface(2,V)
    while i < 10:
        if surface(i,V) < s:
            s = surface(i,V)
            R = i
            i = i + 0.01
    return R, hauteur(R,V)
    
```

Cette fonction permet de calculer la hauteur d'un cylindre dont on connaît le rayon (en cm) et le volume (en cm³).

On utilise la variable **R** pour stocker toutes les valeurs successives que le rayon correspondant à la surface minimale de métal prend au cours de l'exécution du script. Elle est, en premier lieu, initialisée à 0.

Tant que la variable *i* n'a pas atteint 10, on incrémente sa valeur de 0,01.

Conclusion

Une fois le script saisi, exécutez-le à l'aide de la touche `trace`, puis sélectionnez la fonction **dimensions** à l'aide de la touche `var`.

```

PYTHON SHELL
>>>
>>>
>>> # L'exécution de CONSERVE
>>> from CONSERVE import *
>>> dimensions(850)
(5.129999999999933, 10.28097546657202)
>>> dimensions(212)
(3.229999999999974, 6.468162818675988)
>>> |
    
```

Pour utiliser le moins de matière première les boîtes de conserve doivent avoir les dimensions (arrondies au centième) suivantes :

- Un rayon de 5,13 cm et une hauteur de 10,28 cm pour le modèle de 850 cm³ ;
- Un rayon de 3,23 cm et une hauteur de 6,47 cm pour le modèle de 212 cm³.

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
 © Texas Instruments 2019 / Photocopie autorisée

Modules : Probabilités

TI-83 Premium CE

Niveau : Seconde professionnelle

**Le jeu des couleurs**

Jérôme LENOIR

Objectifs et capacités visées

- Modifier une simulation donnée pour percevoir une version vulgarisée de la loi des grands nombres.
- Écrire des fonctions permettant de simuler une expérience aléatoire, une répétition d'expériences aléatoires indépendantes.
- Estimer la probabilité non triviale d'un événement à partir des fréquences.
- Faire preuve d'esprit critique face à une situation aléatoire simple.

Compétences travaillées

- | | | |
|--|--|---|
| <div style="background-color: red; color: white; border-radius: 50%; width: 30px; height: 30px; display: flex; align-items: center; justify-content: center; margin-bottom: 5px;">C1</div> <p>Rechercher, extraire et organiser des informations</p> | <div style="background-color: blue; color: white; border-radius: 50%; width: 30px; height: 30px; display: flex; align-items: center; justify-content: center; margin-bottom: 5px;">C2</div> <p>Élaborer un algorithme</p> | <div style="background-color: green; color: white; border-radius: 50%; width: 30px; height: 30px; display: flex; align-items: center; justify-content: center; margin-bottom: 5px;">C3</div> <p>Mettre en œuvre un algorithme</p> |
| <div style="background-color: orange; color: white; border-radius: 50%; width: 30px; height: 30px; display: flex; align-items: center; justify-content: center; margin-bottom: 5px;">C4</div> <p>Exploiter et interpréter les résultats obtenus afin de répondre à une problématique</p> | <div style="background-color: purple; color: white; border-radius: 50%; width: 30px; height: 30px; display: flex; align-items: center; justify-content: center; margin-bottom: 5px;">C5</div> <p>Rendre compte d'un résultat et expliquer une démarche</p> | |

Situation déclenchante

Dans le cadre du module d'accompagnement renforcé, l'enseignant propose à ses élèves une activité sous forme de jeu. Dans une urne, il place 30 boules (indiscernables au toucher) de couleurs différentes : des rouges, des bleues et des jaunes. Après avoir expérimenté quelques tirages, les élèves ont pour consigne de créer un algorithme puis un script qui permet d'évaluer le nombre de boules de chaque couleur contenues dans l'urne.

**Problématique****Es-tu capable de relever le défi en créant ce script ?**

Proposer une méthode qui permettrait de répondre à la problématique

**Appeler le professeur**

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée



Modules : Probabilités

TI-83 Premium CE

Niveau : Seconde professionnelle



Fiche méthode

Jérôme LENOIR

Proposition de démarche de résolution

Nous allons créer un algorithme puis un script permettant de :

- générer trois nombres aléatoires entiers et complémentaires de 30 (**b**, **r**, **j**), correspondant respectivement au nombre de boules bleues, rouges et jaunes contenues dans l'urne ;
- évaluer la fréquence d'apparition de chacune des couleurs à l'aide de la fonction **urne** ;
- simuler la situation en fonction de la taille **n** de l'échantillon étudié pour une couleur donnée en paramètre.

La variable **nb** est utilisée pour stocker le nombre de boules de couleur bleue, rouge ou jaune en fonction du choix de l'utilisateur.

```
b = randint(0,30)
r = randint(0,(30-b))
j = 30 - b - r
```

```
>>> urne(1000,"jaune")
```

Algorithme

```
b ← nombre aléatoire entier entre 0 et 30
r ← nombre aléatoire entier entre 0 et (30 - b)
j ← 30 - b - r
```

Fonction *urne*(*n*, *couleur*) :

```
s ← 0
Si couleur = « bleu » alors
    nb ← b
Sinon
    Si couleur = « rouge » alors
        nb ← r
    Sinon
        nb ← j
    Fin Si
Fin Si
```

```
Fin Si
Pour i allant de 0 à n-1
    a ← E(nombre aléatoire + nb/30)
    s ← s + a
Fin Pour
Renvoyer s/n
```

Fin

```
PYTHON SHELL
>>> # L'exécution de URNE
>>> from URNE import *
>>> b
14
>>> j
8
>>> r
8
>>> urne(1000,"jaune")
0.258
>>> |
Fns... a A # Outils Éditer Script
```

Remarques

- L'équiprobabilité n'étant pas une condition sinequanone dans le contexte de cette activité, la méthode choisie pour générer les nombres entiers aléatoires **b**, **r** et **j** a été simplifiée ;

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée



Modules : Probabilités

TI-83 Premium CE

Niveau : Seconde professionnelle

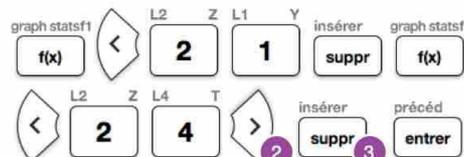


Fiche méthode

Jérôme LENOIR

- dans un script, il est possible d'importer la totalité d'une bibliothèque comme cela a souvent été le cas dans les activités précédentes, mais il est également envisageable d'importer une instruction en particulier. L'exemple ci-contre permet l'import de l'instruction **randint** contenue dans la bibliothèque **RANDOM**.

```
from random import randint
```



Édition du script

Instruction conditionnelle :

- Si ... **if**
- Sinon Si ... **elif**
- Sinon ... **else:**



La méthode utilisée est identique à celle que nous avons développée dans la fiche intitulée **Audiences audiovisuelles**.

```
ÉDITEUR : URNE
LIGNE DU SCRIPT 0003
from math import *
from random import *

b = randint(0,30)
r = randint(0,(30-b))
j = 30 - b - r

def urne(n,couleur):
    s=0
    if couleur == "bleu":
        nb = b
    elif couleur == "rouge":
        nb = r
    else:
        nb = i
    for i in range(n):
        a=floor(random()+nb/30)
        s=s+a
    return s/n_
```

On importe les bibliothèques **math** et **random**

Remarque importante : les trois variables **b**, **r** et **j** sont générées en amont de la fonction **urne**. Ceci permettra d'expérimenter en aval sur la valeur de l'argument **n** (nombre de tirages) tout en gardant la même valeur générée pour ces trois variables (tant que le script ne sera pas à nouveau exécuté).

L'argument **couleur** est une variable de type « chaîne de caractère, il est donc nécessaire de la saisir entre guillemets :



Conclusion

Pour exécuter le script, appuyer sur la touche **trace** puis sélectionner la fonction **urne** dans la liste accessible à l'aide de la touche **var**.

```
PYTHON SHELL
>>> urne(100,"rouge")
0.16
>>> urne(1000,"rouge")
0.144
>>> urne(10000,"rouge")
0.1369
>>> urne(100000,"rouge")
0.13372
>>> r
4
>>> |
```

Dans l'exemple exposé, on peut s'apercevoir que lorsqu'on augmente le nombre de tirages, les fréquences d'apparition des boules rouges tendent à se stabiliser vers une valeur approximative de 0,134. On peut donc estimer que l'urne contient 4 boules rouges (soit 13,4 % d'un total de 30 boules).

(À noter qu'en fin d'étude il est possible de vérifier cette valeur en appelant la variable **r** en appuyant sur la touche : **var**).

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée

Modules : Fonctions / Co-intervention

TI-83 Premium CE

Niveau : Seconde professionnelle



Épures de charpente

Jérôme LENOIR

Capacités, connaissances et objectifs visés

- Traduire un programme de calcul à l'aide d'une fonction en Python.
- Calculer des longueurs.
- Réciproque du théorème de Pythagore.

Compétences travaillées



Élaborer un
algorithme



Mettre en œuvre
un algorithme



Exploiter et interpréter les résultats obtenus
afin de répondre à une problématique

Tâches, compétences et savoirs professionnels associés

• TÂCHE DE RÉALISATION : Activité de préparation (contrôle et réception de l'existant)

T1 : Réceptionner et analyser les données techniques concernant l'ouvrage.

T3 : Contrôler les dimensions, la géométrie, la qualité des supports.

T4 : Effectuer les croquis et prendre en compte les côtes nécessaires à la préfabrication.

• COMPÉTENCES PROFESSIONNELLES VISEES :

C1.12 : Caractériser les pièces et composants constitutifs (dimensions).

C1.42 : Relever les caractéristiques dimensionnelles et géométriques.

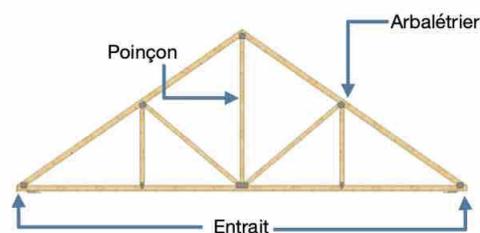
• SAVOIRS ASSOCIÉS :

S2.13 : Représentation des ouvrages (définition géométriques, surfaciques et volumiques).

S2.23 : Cotation de fabrication (cotes de fabrication).

Situation déclenchante

Un élève préparant un baccalauréat professionnel Technicien Constructeur Bois réalise sa période de formation en entreprise de charpente industrielle durant laquelle il doit fabriquer l'épure de fermes dites « en M » pour plusieurs dossiers. Afin de vérifier rapidement que les épures sont bien conformes, il souhaite réaliser un outil permettant de vérifier que l'entrait et le poinçon sont bien perpendiculaires, connaissant les différentes longueurs dans chaque cas.



Problématique

Peux-tu l'aider à créer cet outil ?

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>

© Texas Instruments 2019 / Photocopie autorisée



Modules : Fonctions / Co-intervention

TI-83 Premium CE

Niveau : Seconde professionnelle



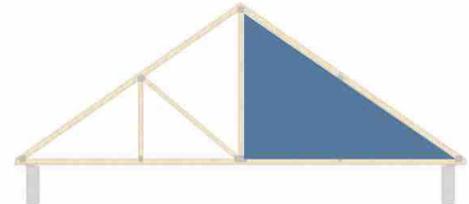
Fiche méthode

Jérôme LENOIR

Proposition de démarche de résolution

Pour répondre à cette problématique, nous allons créer un algorithme puis un script permettant de tester si le triangle interne formé par le demi-entrait, le poinçon et l'arbalétrier de la charpente est rectangle à l'aide d'une fonction nommée **rect**.

```
def rect(a,b,c):
```



Algorithme

```

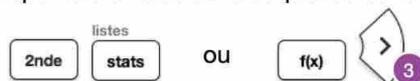
Fonction rect ( a , b , c )
  L ← [ a , b , c ]
  Trier les éléments de la liste L par ordre croissant
  a ← L [ 0 ]
  b ← L [ 1 ]
  c ← L [ 2 ]
  Si  $c^2 / (a^2 + b^2) = 1$  alors
    Renvoyer « Triangle rectangle »
  Sinon
    Renvoyer « Triangle non rectangle »
  Fin Si
Fin
  
```

```

PYTHON SHELL
>>> # Shell Reinitialized
>>>
>>> # L'exécution de CHARPENT
>>> from CHARPENT import *
>>> rect(6,8,10)
'Triangle rectangle'
>>> rect(6,8,11)
'Triangle non rectangle'
>>> |
  
```

Remarques

- Dans un premier temps, nous devons trier les variables **a**, **b**, **c** par ordre croissant à l'aide d'une liste. Un sous-menu spécifique à leur édition est disponible à l'aide de la séquence de touches :



- L'utilisation de variables flottantes peut s'avérer être difficile lorsque l'on souhaite tester l'égalité de deux expressions numériques. Pour pallier cette difficulté, nous allons comparer une valeur arrondie à 10^{-10} du quotient de c^2 par $a^2 + b^2$ à la valeur 1. Du point de vue du contexte, cela est tout à fait justifiable du fait des conditions de réalisation préconisées par le DTU 31.1 en termes de tolérance sur les longueurs de bois utilisées.

```

ÉDITEUR : CHARPENT
Fonc Ctl Ops List Type E/S Modul
1: [ ]
2: list(séquence)
3: len()
4: max()
5: min()
6: .append(x)
7: .remove(x)
8: .insert(indice,x)
9: sum()
0: sorted()
Échap
  
```

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée



Modules : Fonctions / Co-intervention

TI-83 Premium CE

Niveau : Seconde professionnelle



Fiche méthode

Jérôme LENOIR

Édition du script

On affecte alors les trois valeurs de la liste **L** aux variables **a**, **b** et **c** de manière à respecter :
 $a \leq b \leq c$

On calcule une valeur arrondie à 10^{-10} du rapport $\frac{c^2}{a^2 + b^2}$ puis on compare cette valeur à 1.

On conclut alors sur le fait que le triangle étudié est rectangle ou non.

```

ÉDITEUR : CHARPENT
LIGNE DU SCRIPT 0001
def rect(a,b,c):
    L = [a,b,c]
    L.sort()
    a = L[0]
    b = L[1]
    c = L[2]
    if round(c**2/(a**2+b**2),10)
    ==1:
        return "Triangle rectang
        le"
    else:
        return "Triangle non rec
        tangle"
    
```

On crée une liste appelée **L** contenant les valeurs correspondant aux trois longueurs à tester :



Les valeurs contenues dans la liste **L** sont triées par ordre croissant :



Conclusion

Une fois le script saisi, exécutez-le à l'aide de la touche **trace** , puis sélectionner la fonction **rect** à l'aide de la touche **var** .

```

PYTHON SHELL
>>> # Shell Reinitialized
>>>
>>> # L'exécution de CHARPENT
>>> from CHARPENT import *
>>> rect(4.31,1.76,5.16)
'Triangle non rectangle'
>>> rect(15.60,9.36,12.48)
'Triangle rectangle'
>>> |
    
```

Dans le cas des deux épures testées :

- La première présente un défaut de longueur sur au moins un des éléments ;
- La seconde épure est quant à elle correcte.

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
 © Texas Instruments 2019 / Photocopie autorisée

Modules : Statistique

TI-83 Premium CE

Niveau : Seconde professionnelle

**Fréquence d'apparition d'une lettre**

Jérôme LENOIR

Objectifs et capacités visées

- Traduire un programme de calcul à l'aide d'une fonction en Python.
- Recueillir et organiser des données statistiques.

Compétences travaillées

C1 Rechercher, extraire et organiser des informations



C2 Élaborer un algorithme



C3 Mettre en œuvre un algorithme



C4 Exploiter et interpréter les résultats obtenus afin de répondre à une problématique



C5 Rendre compte d'un résultat et expliquer une démarche

Situation déclenchante

Pour déterminer la langue utilisée dans un texte donné, certains moteurs de recherche calculent la fréquence d'apparition de certaines lettres de l'alphabet puis les comparent à des fréquences théoriques propres à chaque langue.

Problématique

Peux-tu créer un script permettant de déterminer la fréquence d'apparition de chaque lettre dans un texte donné ?

Proposer une méthode qui permettrait de répondre à la problématique



Appeler le professeur

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée



Modules : Statistiques

TI-83 Premium CE

Niveau : Seconde professionnelle



Fiche méthode

Jérôme LENOIR

Proposition de démarche de résolution

Nous allons créer un algorithme puis un script contenant deux fonctions :

- une première fonction nommée **f_lettre** qui renvoie la fréquence d'apparition de la lettre (**variable l**) dans un texte donné en argument (**variable t**) ;
- une seconde fonction nommée **freq** qui renvoie, pour chaque lettre de l'alphabet, la fréquence d'apparition dans un texte donné (**variable t**).

```
def f_lettre(t,l)
```

```
def freq(t)
```

Algorithmes

Fonction *f_lettre(t, l)* :

long ← longueur du texte *t*
cpt ← nombre d'occurrences de la lettre « *l* » dans le
 texte « *t* »
 Renvoyer lettre et *cpt/long*

Fin

Fonction *freq(t)* :

alpha ← « abcdefghijklmnopqrstuvwxyz »
 Pour chacune des lettres de la variable *alpha*
 Afficher *f_lettre(t, l)*
 Pause
 Fin Pour

Fin

```
PYTHON SHELL
alpha
>>> f_lettre("programmation en p
ython","p")
('p', 0.08695652173913043)
>>> freq("programmation en pytho
n")
('a', 0.08695652173913043)

('b', 0.0)

('c', 0.0)

Fns... a A # Outils Éditer Script
```

Remarques

- Python est un langage de programmation qui différencie les lettres minuscules des lettres majuscules. Aussi, il est important de respecter la casse dans la saisie des variables, c'est à dire de prêter une attention particulière à l'utilisation des majuscules et des minuscules. Néanmoins, dans notre cas de figure, il est tout à fait envisageable de modifier la variable **alpha** en travaillant avec des majuscules ou des symboles.
- L'instruction ci-contre permet de déterminer le nombre de caractères correspondant à la valeur de la variable **lettre** contenus dans la variable **texte**. Cette instruction n'étant pas directement accessible dans un sous-menu, il conviendra de la saisir en toutes lettres.

```
alpha = "ABCDEFGHJKLMNO
PQRSTUVWXYZ"
```

```
texte.count(lettre)
```

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
 © Texas Instruments 2019 / Photocopie autorisée



Modules : Statistiques

TI-83 Premium CE

Niveau : Seconde professionnelle



Fiche méthode

Jérôme LENOIR

- Dans l'algorithme précédent, la variable **long** comptabilise également les caractères d'espace contenus dans le texte (**variable t**) à étudier. Pour éviter que ces caractères n'influent sur le calcul des différentes fréquences, il est possible d'utiliser une variable **espace** qui comptera leur nombre d'occurrences dans le texte étudié.

```
cpt = t.count(' ')
espace = t.count(" ")
return l, cpt/(long-espace)
```

Édition du script

On crée une variable **alpha** de type chaîne de caractères regroupant l'ensemble des lettres minuscules de l'alphabet.

L'instruction **input()** est dans notre cas utilisée pour faire une pause durant l'affichage des différentes fréquences en invitant l'utilisateur à appuyer sur la touche :



```
ÉDITEUR : FLETTRE
LIGNE DU SCRIPT 0005
def f_lettre(t,l):
    long = len(t)
    cpt = t.count(l)
    return l, cpt/long

def freq(t):
    alpha = "abcdefghijklmnopqrs
            tuvwxvz"
    for l in alpha:
        print(f_lettre(t,l))
    input()

Fns... a A # Outils Exéc Script
```

On détermine le nombre de caractères de la variable **t** :



La fonction **f_lettre** renvoie la lettre mise en argument et la fréquence d'apparition correspondante dans la variable **t**.

Pour chacune des lettres minuscules de la variable **alpha**, on affiche le résultat de la fonction **f_lettre**.

Conclusion

```
PYTHON SHELL
@ ALPHA

>>> # Shell Reinitialized
>>>
>>> # L'exécution de F_LETTRE
>>> from F_LETTRE import *
>>> freq("python est disponible
sur la ti83 premium ce")
('a', 0.02272727272727273)
('b', 0.02272727272727273)

Fns... a A # Outils Éditer Script
```

Une fois le script saisi, exécutez - le à l'aide de la touche **trace**, puis sélectionnez la fonction **freq** à l'aide de la touche **var**.

Dans le texte testé ici, la fréquence d'apparition des lettres « a » et « b » est de 0,027 ce qui représente 2,7 % du texte.

À noter qu'il suffit d'appuyer sur la touche **entrer** pour voir apparaître une nouvelle lettre et la fréquence d'apparition correspondante.

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Ce document est mis à disposition sous licence Creative Commons <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>
© Texas Instruments 2019 / Photocopie autorisée

