Bibliothèques TP6

Exercice 1 Premières fonctions non natives

La plupart des fonctions mathématiques usuelles (cos, sin, exp, ln, etc) ne sont pas directement accessibles dans Python. Il faut les "importer" depuis une *bibliothèque* que l'on télécharge depuis Internet.

Pour importer une bibliothèque, taper en début de script, la commande :

```
1 import ma_bibliotheque
```

où `ma_bibliotheque` est le nom de la bibliothèque voulue.
Si vous travaillez sur votre ordinateur portable, la première fois que vous utilisez une bibliothèque, il faut la télécharger depuis Internet en tapant directement dans le shell :

```
1 pip install ma_bibliothèque
```

Cette étape de téléchargement n'est nécessaire qu'à la première utilisation de la bibliothèque. Ensuite, vous pouvez utiliser simplement la commande `import`.

Une fois cette bibliothèque téléchargée, on a accès aux fonctions qu'elle contient. Toutefois, il faut rappeler à Python d'aller chercher ces fonctions dans la bibliothèque concernée. Pour cela, si l'on souhaite utiliser une fonction `fun` issue de la bibliothèque `ma_bibliotheque`, il faut utiliser la syntaxe `ma_bibliotheque.fun`.

Par exemple, la bibliothèque dont le nom est `math` contient la fonction exponentielle `exp`. Pour calculer exp(5) on écrit donc `math.exp(5)`.

Q1 Importer la bibliothèque `math`. Elle contient la fonction `exp` et la fonction `sin`. Écrire une fonction Python `f` prenant en argument un réel x et renvoyant $\exp(\sin^2(x))$. On vérifiera que `f(12)` vaut approximativement 1,333.

Il peut être fastidieux de recopier le nom de la bibliothèque devant le nom des fonctions qu'on souhaite utiliser. On peut donc renommer la bibliothèque à l'aide d'un *alias* plus court. La syntaxe est la suivante :

```
1 import ma_bibliotheque as alias
```

pour utiliser la fonction `fun` issue de cette biblithèque, il suffit alors d'écrire `alias.fun`.

Les noms d'alias sont libres. Toutefois, certains noms se sont imposés par l'usage ; et il n'est pas recommandé de faire de fantaisie. Par exemple, la bibliothèque `numpy`, contenant elle aussi les fonctions mathématiques usuelles, s'importe généralement avec l'alias `np`. On écrit donc en préambule :
`import numpy as np`

Q2 Importer la bibliothèque `numpy` avec l'alias `np`. Cette bibliothèque contient la constante `pi` et la fonction racine carrée `sqrt` (pour l'anglais "square root"). Écrire une commande permettant d'accéder à la valeur de $\sqrt{\pi}$. On vérifiera que cette valeur vaut environ 1,772. Accéder ensuite à la valeur de $\ln(\pi) \simeq 1,1447$.
À retenir : en Python $\ln(x)$ s'écrit :

Remarque

Il est également possible de n'importer qu'une fonction en particulier depuis une bibliothèque avec la syntaxe :

```
1 from ma_bibliotheque import ma_fonction
```

Par exemple, si je n'ai besoin que de la fonction racine carrée issue de la bibliothèque `math`, il me suffit d'écrire : `from math import sqrt`. Dans ce cas, il n'est plus nécessaire de rappeler le nom de la bibliothèque pour utiliser la fonction. On pourra écrire directement `sqrt(5)` pour accéder à $\sqrt{5}$.
On peut aussi importer toutes les focntions d'une bibliothèque et ne plus avoir besoin de rappeler le nom de la bibliothèque ou son alias. Pour cela, on utilise la syntaxe :

```
1 from ma_bibliotheque import *
```

Par exemple, en écrivant `from math import *`, on peut ensuite écrire `sqrt(sin(5))` pour calculer $\sqrt{\sin(5)}$.

Toutefois, ces deux derniers usages ne sont pas recommandés. Même si cela peut paraître plus long, utiliser un alias permet de se rappeler dans quelle bibliothèque se trouve quelle fonction, et surtoût qu'il faut penser à importer cette bibliothèque !

Exercice 2 Un premier graphique

Pour tracer des graphiques on utilise la bibliothèque `matplotlib.pyplot`. L'alias usuel pour importer cette bibliothèque est `plt`. On peut donc retenir que pour utiliser cette bibliothèque on écrira en préambule :

```
1 import matplotlib.pyplot as plt
```

On dispose alors des commandes suivantes :

```
1 plt.plot(mes_absi,mes_ordo) # construit le graphique de mes_ordo en
    fonction de mes_absi
2 plt.show() # affiche le graphique précédent
```

Elles permettent de créer le graphique représentant les valeurs contenues dans une variable `mes_ordo` en fonction de celles contenues dans `mes_absi`.

Pour créer rapidement les valeurs des abscisses, on utilise la fonction `linspace` de la bibliothèque `numpy`. Il faudra donc penser à importer cette bibliothèque (mais ici il n'est pas nécessaire de réécrire `import numpy as np` puisqu'on l'a déjà fait à l'exercice 1 !).

La commande `np.linspace(a,b,n)` permet d'obtenir une liste de `n` valeurs régulièrement espacées commençant à `a` et se terminant à `b`.

Q1 Prédire ce que contient `np.linspace(0,1,5)`. Vérifier ensuite dans la console.

Pour obtenir les valeurs en ordonnées correspondantes aux valeurs contenues dans `mes_absi`, il suffit d'appeler une fonction numérique sur `mes_absi`.
Par exemple, si `mes_absi = np.linspace(0,1,5)` alors `np.cos(mes_absi)` contient les valeurs $\cos(x_i)$ pour toutes les valeurs x_i apparaissant dans `mes_absi`.

Remarque

On pourra donc retenir que pour tracer le graphe d'une fonction numérique `f` entre les abscisses `a` et `b` avec `n` points de tracé on peut écrire, après avoir importé les bibliothèques adéquates :

```
absi = np.linspace(a,b,n)
ordo = f(absi)
plt.plot(absi,ordo)
plt.show()
```

Attention, la syntaxe `f(absi)` consiste à appeler une fonction numérique `f` avec pour argument une *liste de valeurs*. Cela n'est possible que si pour définir `f` on a utilisé des fonctions issues de la bibliothèque `numpy` (et les opérations usuelles). La bibliothèque `math` n'est donc pas adaptée au tracé de fonctions (on peut l'oublier !).

Q2 Tracer le graphe de la fonction $g : x \longmapsto (x+1)/(x+2)$ sur $[-1,8\ ;\ 10]$ en utilisant 1000 points de tracé.

Q3 Tracer le graphe de la fonction $h : x \longmapsto \ln(5-x)$ sur $[0,10]$ avec 200 points de tracé. Votre graphe est-il bien sur l'intervalle $[0,10]$? Qu'a fait automatiquement (ou pas) Python ?

Remarque

Enfin, de nombreuses commandes existent pour ajouter des éléments aux graphiques, par exemple :

```
plt.grid() # Affiche une grille
plt.title("Mon titre")
plt.xlabel("Titre des abscisses")
plt.ylabel("Titre des ordonnés")
```

Ce sont loin d'être les seules, et il ne faut pas hésiter à consulter la documentation de la bibliothèque `matplotlib.pyplot` en ligne. Tapez les mots clés correspondant à ce que vous souhaitez faire dans un moteur de recherche, vous trouverez très rapidement la syntaxe appropriée et des exemples.

Toutes ces commandes sont surtoût utiles lorsqu'on manipule des données réelles, en physique ou en biologie (et en TIPE). Sauf indication contraitre, on ne demandera pas de les utiliser en mathématiques lorsqu'on trace des graphes de fonctions.

Exercice 3 Fonctions usuelles, symétries, translations

Actualisez la console (bouton "Refresh") : les commandes compilées précédemment ont donc été oubliées par la console.

Q1 Définir une fonction Python `f` correspondant à la fonction $f : x \longmapsto \sqrt{x} - \frac{x}{2}$. Tracer ensuite le graphe de f sur $[0,4]$ avec 100 points de tracé et en affichant une grille.

Q2 Définir une fonction Python correspondant à la fonction $g : x \longmapsto f(x) + 2$. Tracer alors sur le même dessin les graphes de f et de g sur $[0,4]$ avec 100 points de tracé. On tracera le graphe de f en noir et celui de g en vert, et on affichera une grille.

Remarque

Pour préciser la couleur d'un graphe, utilisez la syntaxe :

```
plt.plot(absi, ordo, 'z')
```

en remplaçant `z` par : `b` pour bleu, `r` pour rouge, `g` pour vert, `k` pour noir, `y` pour jaune, etc.

Q3 Reprendre la question précédente mais avec g définie par $g : x \longmapsto f(x) - 2$. De manière générale, comment obtient-on le graphe de $x \longmapsto f(x) + K$ en fonction de celui de f ?

Q4 Définir une fonction Python correspondant à la fonction $h : x \longmapsto f(x - 2)$. Tracer alors sur le même dessin les graphes de f et de h sur des ensembles de votre choix. On tracera le graphe de f en noir et celui de h en jaune, et on affichera une grille. Reprendre cette question avec $h : x \longmapsto f(x+2)$ en modifiant si besoin les ensembles de tracés pour visualiser correctement le dessin. De manière générale, comment obtient-on le graphe de $x \longmapsto f(x + K)$ en fonction de celui de f ?

Q5 Définir une fonction Python correspondant à la fonction $mf : x \longmapsto -f(x)$. Tracer alors sur le même dessin les graphes de f et de mf sur des ensembles appropriés. On tracera le graphe de f en noir et celui de mf en rouge, et on affichera une grille. Reprendre cette question avec la fonction $f^{\text{bis}} : x \longmapsto \ln(x)$. De manière générale, comment obtient-on le graphe de $x \longmapsto -f(x)$ en fonction de celui de f ?

Q6 Dans cette question, on utilise la fonction $p : x \longmapsto e^x$. Définir la fonction p en Python, ainsi qu'une fonction correspondant à $\tilde{p} : x \longmapsto p(-x)$. Tracer alors sur le même dessin les graphes de p et de $\tilde{p}$ sur $[-2, 2]$. On tracera le graphe de p en noir et celui de $\tilde{p}$ en rouge, et on affichera une grille. Reprendre cette question avec la fonction $p^{\texttt{bis}} : x \longmapsto \sin(x^3)$. De manière générale, comment obtient-on le graphe de $x \longmapsto f(-x)$ en fonction de celui de f ?

Q7 Que se passe-t-il lorsque f est paire ? impaire ?

Q8 En appliquant les règles ci-dessus, tracez à la main les graphes de fonctions suivantes. Vérifiez ensuite votre résultat en utilisant Python.

1. $f_1 : x \longmapsto x^3 + 1$
2. $f_2 : x \longmapsto \ln(x + 1)$
3. $f_3 : x \longmapsto e^{-x}$
4. $f_4 : x \longmapsto -\dfrac{1}{x - 2}$