

Fonctions : exercices

Si vous n'avez pas encore réussi à installer PYTHON chez vous, vous pouvez utiliser, en attendant :

- Jupyter en ligne : <http://jupyter.org/try>
- Python Tutor, qui permet en plus de bien comprendre comment cela fonctionne : <http://www.pythontutor.com/visualize.html>



Avant de traiter ces exercices, il faut avoir :

1. Relu le cours sur les tuples et les fonctions ;
2. terminé les parties 1, 2, 3 et 4 du TP n° 4 sur les fonctions.

Rappel : La syntaxe pour définir une fonction est :

PYTHON

```
def nom_fonction(arg1, arg2, ...):
    """Chaîne de documentation."""
    # instructions
    # à
    # réaliser
    return valeur
```

Par exemple :

PYTHON

```
import math

def volume_sphere(r):
    """Volume d'une sphère de rayon `r`."""
    return (4/3) * math.pi * r**3
```

On peut ensuite faire un *appel* à la fonction :

PYTHON

```
volume_soleil = volume_sphere(696342)
print(volume_soleil)
```

EXERCICE 1 *Application directe du cours*

Définir en PYTHON la fonction `multiplier(a, b)` qui multiplie ses deux arguments.

EXERCICE 2 *Quelques fonctions mathématiques*

Définir en PYTHON les fonctions suivantes :

1. $f : x \mapsto \frac{x}{1+x^2}$
2. $g : x \mapsto \ln x + x^{x+1}$
3. $h : x \mapsto \sin(\pi[x])$
4. $u : (x, y) \mapsto x^{-y}$
5. $v : (x, y, z) \mapsto (x+y) \times z$
6. $w : (x, y, z, t) \mapsto (y-x, t-z)$
7. $k : ((x, y), (z, t)) \mapsto (y-x, t-z)$

EXERCICE 3 *Quelques fonctions avec des booléens*

On reprend ici l'exercice n° 6 du TP n° 3 en utilisant des fonctions. Définir en PYTHON :

1. une fonction `div_par_5(n)` qui vérifie si `n` est divisible par 5, c'est-à-dire qui renvoie le booléen `True` si `n` est divisible par 5 et le booléen `False` sinon ;
2. une fonction `multiples(m, n)` qui vérifie si les entiers `m` et `n` sont tels que l'un est multiple de l'autre ;
3. une fonction `meme_signe(m, n)` qui vérifie si les entiers `m` et `n` sont de même signe ;
4. une fonction `tous_meme_signe(m, n, p)` qui vérifie si les trois entiers `m`, `n` et `p` sont de même signe. *On réutilisera la fonction `meme_signe`.*