

# Solutions des exercices du polycopié de Python

Romain Tavenard

## Exercice 2.1

```
import math

def aire_equi(l):
    base = l
    hauteur = l * math.sin(math.pi / 3)
    return base * hauteur / 2

print(aire_equi(1.))
# [Sortie] 0.4330127018922193
```

## Exercice 3.1

```
def argmax(liste):
    i_max = None
    # On initialise elem_max à une valeur
    # qui n'est clairement pas le max
    elem_max = liste[0] - 1
    for i, elem in enumerate(liste):
        if elem > elem_max:
            i_max = i
            elem_max = elem
    return i_max

print(argmax([1, 6, 2, 4]))
# [Sortie] 1
```

## Exercice 4.1

```
def compte_prefix(s, prefix):
    compteur = 0
    for mot in s.split():
        if mot.startswith(prefix):
            compteur += 1
    return compteur

print(compte_prefix("la vie est belle au bord du lac", "la"))
```

```
# [Sortie] 2
```

### Exercice 5.1

```
s = "la vie est belle c'est la vie"
d = {}
for mot in s.split():
    d[mot] = d.get(mot, 0) + 1
print(d)
# [Sortie] {"c'est": 1, 'la': 2, 'belle': 1, 'est': 1, 'vie': 2}
```

### Exercice 6.1

```
import os

def nb_lignes(nom_fichier):
    n = 0
    fp = open(nom_fichier, "r")
    for ligne in fp.readlines():
        n += 1
    return n

def nb_lignes_repertoire(repertoire):
    for nom_fichier in os.listdir(repertoire):
        if nom_fichier.endswith(".txt"):
            nom_complet_fichier = os.path.join(repertoire, nom_fichier)
            n = nb_lignes(nom_complet_fichier)
            print(nom_complet_fichier, n)

nb_lignes_repertoire(".")
```

### Exercice 7.1

```
def bissextile(annee):
    if annee % 4 == 0 and annee % 100 != 0:
        return True
    elif annee % 400 == 0:
        return True
    else:
        return False

def nb_jours(mois, annee):
    if mois in [1, 3, 5, 7, 8, 10, 12]:
        return 31
    elif mois in [4, 6, 9, 11]:
        return 30
    else: # Mois de février
        if bissextile(annee):
            return 29
```

```

        else:
            return 28

def lendemain(jour, mois, annee):
    if jour < nb_jours(mois, annee):
        return jour + 1, mois, annee
    elif mois < 12: # Dernier jour du mois mais pas de l'année
        return 1, mois + 1, annee
    else: # Dernier jour de l'année
        return 1, 1, annee + 1

# Tests de la fonction bissextile
print(bissextile(2004)) # True car divisible par 4 et non par 100
print(bissextile(1900)) # False car divisible par 100 et non par 400
print(bissextile(2000)) # True car divisible par 400
print(bissextile(1999)) # False car divisible ni par 4 ni par 100

# Tests de la fonction nb_jours
print(nb_jours(3, 2010)) # Mars : 31
print(nb_jours(4, 2010)) # Avril : 30
print(nb_jours(2, 2010)) # Février d'une année non bissextile : 28
print(nb_jours(2, 2004)) # Février d'une année bissextile : 29

# Tests de la fonction lendemain
print(lendemain(12, 2, 2010)) # 13, 2, 2010
print(lendemain(28, 2, 2010)) # 1, 3, 2010
print(lendemain(31, 12, 2010)) # 1, 1, 2011

```