

Correction du DS 4

Julien REICHERT

Exercice 1

La fonction `v`, qui s'inspire de la fonction `vectorize` du module `numpy`, prend en argument une fonction `f` (a priori prenant en argument un nombre, mais ce n'est pas nécessaire) et retourne une fonction `g` qui peut appliquer `f` directement si son argument n'est pas itérable ou appliquer `f` à tous les éléments de son argument et renvoyer le tableau des résultats sinon.

Cas où la fonction marche :

- `f` attend un nombre et on donne un nombre à `g` ;
- `f` attend un nombre et on donne une liste de nombres à `g` ;
- `f` attend une liste et on donne une liste de listes à `g` ;
- `f` attend une chaîne de caractères et on donne une chaîne de caractères à `g`.

Cas où la fonction ne marche pas :

- `f` attend une séquence et on donne un nombre à `g` ;
- `f` attend une liste et on donne une liste de nombres à `g` ;
- `f` attend un nombre et on donne une liste de listes à `g` ;
- `f` attend un nombre (nécessairement) et on donne une chaîne de caractères à `g`.

Exercice 2

Cf. cours, adaptation par la suite.

Exercice 3

On s'inspire de l'exercice 1, volontairement placé ici.

```
def decale(fi, ci):
    def g(x):
        return fi(x) - ci
    return g

# Pour les fans des one-liners : decale = lambda fi, ci : lambda x : fi(x) - ci
```

Exercice 4

```
def newton(f, df, x0, eps, c): # autre version : return newton(decale(f, c), df, x0, eps)
    xn = x0
    fonction = decale(f, c)
    xnplusun = xn - fonction(xn) / df(xn)
    while abs(xnplusun - xn) > eps:
        xn = xnplusun
        xnplusun = xnplusun - fonction(xnplusun) / df(xnplusun)
    return xnplusun
```

Exercice 5

```
def derive(l):
    rep = []
    for i in range(1, len(l)):
        rep.append(l[i] * i)
    return rep
```

Exercice 6

```
def couleur(case1, case2):
    return (ord(case2[0]) - ord(case1[0]) + ord(case2[1]) - ord(case2[0])) % 2 == 0
```

Exercice 7

```
def etapes(n):
    rep = 0
    while n > 0:
        if n % 2 == 0:
            n //= 2
        else:
            n -= 1
        rep += 1
    return rep
```