

DS 4

Informatique pour tous, première année

Julien REICHERT

Exercice 1 : Expliquer ce que fait la fonction suivante, en décrivant notamment l'argument et la valeur de retour. On supposera que `numpy` aura été importé avec l'alias `np`.

```
def v(f):
    """Pour information, iterable retourne un booléen
    qui détermine si son argument peut être parcouru."""
    def g(X):
        l = []
        if not iterable(X):
            return f(X)
        for x in X:
            l.append(f(x))
        return np.array(l)
    return g
```

Exercice 2 : Écrire en Python une fonction `newton(f, df, x0, eps)` qui détermine une approximation d'un zéro de la fonction `f`, en s'appuyant sur sa dérivée `df`, à partir du point `x0`. L'approximation doit être à `eps` près du zéro, et on accepte que pour des fonctions tordues la méthode échoue.

Exercice 3 : Écrire en Python une fonction `decale` qui prend en entrée une fonction `fi` simulant une fonction réelle f à valeurs dans \mathbb{R} ainsi qu'un nombre `ci` et qui retourne une fonction simulant la fonction réelle $x \mapsto f(x) - c$, sachant que `ci` représente le nombre c .

Exercice 4 : Adapter le code de l'exercice 2 pour que la méthode de Newton détermine une approximation d'un antécédent par la fonction en argument d'un nombre quelconque (qui est donc un argument supplémentaire). On pourra se contenter de faire apparaître les ajouts et modifications par rapport à la fonction de l'exercice 2, en y balisant clairement les lignes.

Exercice 5 : Écrire une fonction `derive(l)` qui prend en argument une liste `l` représentant un polynôme P , de sorte que pour tout indice valide `i`, l'élément `l[i]` corresponde au coefficient en X^i de $P(X)$, et qui retourne une liste représentant le polynôme P' suivant la même convention.

Par exemple, l'appel `derive([5, 2, 6, 3, 4])` renverra `[2, 12, 9, 16]`.

Exercice 6 : Écrire en Python une fonction `couleur(case1, case2)` qui prend en entrée deux chaînes de caractères représentant chacune une case d'un échiquier et qui détermine si les cases sont de la même couleur. La représentation d'une case se fait en donnant sa colonne, entre "A" et "H", et sa ligne, entre 1 et 8. La case "A1" est noire, et toute voisine orthogonale d'une case est d'une couleur différente.

Par exemple, l'appel `couleur("B3", "D4")` renverra `False`.

Exercice 7 : Écrire en Python une fonction `etapes(n)` qui prend en entrée un entier naturel `n` et qui détermine le plus petit nombre d'étapes nécessaires pour atteindre l'entier `n` depuis l'entier 0, une étape revenant à ajouter un ou à doubler.

Par exemple, l'appel `etapes(6)` renverra 4, les étapes pouvant être 0 -> 1 -> 2 -> 3 -> 6, avec trois incrémentation et une duplication.