

# TP n°2 : les fonctions en Python.

---

## 1. Documentation des fonctions en Python et un peu d'arithmétique.

---

### Activité 1.

---

On considère la fonction python suivante :

```
def est_divisible_par_2(nombre):
    """déterminer si un nombre entier est divisible par 2
    nombre : un entier (int)
    return : True si le nombre est divisible par 2, False sinon.

    >>> est_divisible_par_2(8)
    True
    >>> est_divisible_par_2(5)
    False
    >>> est_divisible_par_2(1.8)
    "Votre nombre n'est pas un entier."
    if nombre%2==0 :
        return True
    else :
        return False
```

La fonction est documentée par ce que l'on nomme une Docstring ainsi que trois tests.

Recopier et exécuter cette fonction pour savoir si elle vérifie les trois tests de la docstring.

Modifier la fonction pour que les trois tests soient passés avec succès.

### Activité 2

---

Compléter la fonction suivante pour qu'elle réponde à sa docstring.

```
def est_divisible(n,p):
    """détermine si un nombre entier n est divisible par le nombre entier p
    n: un entier (int)
    p: un entier (int)
    return : True si le nombre n est divisible par p, False sinon.

    >>> est_divisible(8,2)
    True
    >>> est_divisible(15,3)
    True
```

```
>>> est_divisible(6,4)
False
>>> est_divisible(6,4.5)
"Vos nombres ne sont pas deux entiers."
'''
```

## Activité 3

---

On désire à présent se servir de la fonction créée à l'activité 2 pour créer une nouvelle fonction permettant de déterminer si un nombre entier est un nombre premier.

```
def est_premier(nombre):
    ''' déterminer si le nombre donné est un nombre premier
    nombre : int
    return : True si le nombre est premier, False sinon
    >>> est_premier(5)
    True
    >>> est_premier(8)
    False
    >>> est_premier(1)
    False
    >>> est_premier(15.8)
    "Votre nombre n'est pas un entier"
    '''
```

Pour déterminer si le nombre  $N$  est premier, on vérifiera simplement qu'il n'est pas divisible par tous les entiers compris entre 2 et  $N-1$ .

Compléter la fonction.

## 2. On travaille un peu avec les chaînes de caractères.

---

Compléter la fonction suivante pour qu'elle réponde à sa docstring.

## Activité 4

---

```
def inverse(message):  
    ''' inverse une chaîne de caractères  
    message : type str  
    return : retourne le message inversé (str ou int )  
  
    >>> inverse("azerty")  
    "ytreza"  
    >>> inverse("non")  
    "non"  
    '''
```

## Activité 4 ++

---

Modifier la fonction précédente pour qu'elle traite aussi bien les chaînes de caractères. La valeur retournée sera du même type que la valeur passée en paramètre.

## 3. On revient sur la notion de conversion d'un entier dans une base donnée.

---

### Activité 5 : mini projet

---

Ecrire une fonction permettant de donner l'écriture d'un entier  $n$ , donné en base 10, dans une base donnée. On utilisera pour cela l'algorithme des divisions successives présenté en cours.

La fonction sera documentée par une docstring. Elle retournera une chaîne de caractères.