

Instruction conditionnelle - Corrigé

NUMWORKS

1 Exercice

Écrire une fonction `vabsolue(x)` qui prend un réel en argument et renvoie sa valeur absolue.

1.1 Analyse de l'énoncé

La valeur absolue d'un nombre x , notée $|x|$, est sa distance à zéro. La valeur absolue d'un réel positif est donc ce nombre et celle d'un nombre négatif est son opposé.

En formalisant on peut écrire :

- $|x| = x$ si x est positif
- $|x| = -x$ si x est négatif

On souhaite écrire une fonction qui prend un réel x en argument et qui renvoie sa valeur absolue. Il va donc falloir effectuer un test sur le signe de x pour donner le résultat.

1.2 Résolution

On voit ici apparaître une instruction conditionnelle qui distingue deux cas :

- si x est positif, renvoyer x
- sinon, renvoyer $-x$

Le cas $x = 0$ peut être inclus dans l'un ou l'autre des cas. On peut donc écrire au choix : `if x > 0` ou bien `if x >= 0`.

```
def vabsolue(x):  
    if x > 0:  
        return x  
    else:  
        return -x
```

On écrira donc la fonction suivante :

```
deg PYTHON
def mediane(liste):
    liste=sorted(liste)
    n=len(liste)
    if n%2==1:
        return liste[int((n-1)/2)]
    else:
        v_1=liste[int(n/2-1)]
        v_2=liste[int((n/2))]
        return (v_1+v_2)/2
```

FIGURE 1 – image 3

```
deg PYTHON
>>> from exercice import *
>>> vabsolue(5)
5
>>> vabsolue(-5)
5
>>> vabsolue(0)
0
>>> |
```

Dans la console on peut tester cette fonction.

1.3 Remarque pour finir

La fonction valeur absolue est déjà présente dans Python. Il suffit d'écrire `abs()` pour l'appeler.

```
>>> abs(-4)
4
```

2 Un autre exercice à traiter

Écrire une fonction `mediane` qui prend une série de nombres de taille quelconque en argument et qui renvoie la médiane de la série.

2.1 Corrigé

2.2 Quelques remarques sur le corrigé

La commande `sorted(liste)` renvoie la liste des valeurs classées par ordre croissant :

```
>>> sorted([5,3,4])  
[3,4,5]
```

Le caractère `%` (modulo) donne le reste de la division euclidienne. Ainsi `effectif % 2` teste si `effectif` est pair ou impair.

```
>>> 5 % 2  
1
```

L'indice d'un élément d'une liste doit être entier. C'est pour cela que l'on utilise `int` (*integer* = entier en anglais) entre les crochets :

```
>>> int(4.0)  
4
```

Pour sélectionner le k -ième élément d'une liste L , on écrit $L[k-1]$. En effet, le premier indice d'une liste est 0 en Python. On sélectionne ici l'élément placé en $(n+1)/2$ dans `liste_valeurs`, le 3ème élément pour une liste à 5 éléments :

```
>>> liste = [1,4,7,9,10]  
>>> n = 5  
>>> liste[int((n-1)/2)]  
7
```