

Fonctions

1 Définition, Appel

1.1 Définition d'une fonction

Une fonction est un bloc d'instructions qui a reçu un nom, dont le fonctionnement dépend d'un certain nombre de paramètres (les arguments de la fonction) et qui renvoie un résultat.

Syntaxe:

```
def nom(param1,param2,param3):  
    bloc d'instructions  
    return sortie 1, sortie 2
```

- nom est le nom que l'on donne à la fonction.
Le nom d'une fonction doit être court et significatif...
- param1,param2,param3,... sont les paramètres de la fonction.
Il n'y a aucune restriction sur le nombre de paramètres qu'une fonction doit avoir. Si la fonction n'a aucun paramètre, on écrit: def nom()
- Il est CRUCIAL de respecter l'indentation, c'est-à-dire le retrait (à droite), par rapport à **def**
*Quand on revient au même niveau d'indentation que la ligne débutant par **def**, on sort de la fonction.*

Exemple: écrire deux fonctions python:

- (1) double(x) qui calcule le double d'un réel x
- (2) hypo(a,b) qui renvoie l'hypothénuse d'un triangle rectangle de côtés a et b

1.2 Appel d'une fonction

Syntaxe.

```
nom(valeur1,valeur2,...)
```

- nom est le nom de la fonction appelée.
- valeur1 est la valeur donnée au premier paramètre de la fonction, valeur2, la valeur donnée au second paramètre de la fonction, et ainsi de suite.
L'ordre dans lequel on donne les valeurs des paramètres lors de l'appel de la fonction doit être LE MÊME que celui dans lequel on a donné les paramètres lors de la définition de cette fonction!!
- L'appel d'une fonction n'est pas une instruction. Il doit donc être inclus dans une instruction (affectation, print(), ou autre...)

Exemple:

```
y=double(5); print(y)  
print(double(2))
```

```
hypo(1,2)!=hypo(2,1)
```

Que renvoie l'ordinateur ? Justifier.

2 Variables locales et globales

Variable globale: Les variables définies en dehors de toute fonction sont dites **globales**.

Toute modification d'une variable globale effectuée dans le corps d'une fonction n'a d'effet qu'à l'intérieur de la fonction. Une fois sortie de la fonction, la variable globale reprend la valeur qu'elle avait avant la création de la fonction.

Les variables globales ne peuvent pas être modifiées durablement par une fonction, à moins que l'on utilise le mot-clé `global`

Variable locale: les variables définies dans le bloc d'instructions qui compose la fonction sont dites **locales** à cette fonction. Elles sont créées à l'appel et détruites à la fin de l'exécution de la fonction.

- Toute référence à une variable locale en dehors de la fonction où elle est définie amène une erreur.
- Il est possible d'utiliser les mêmes noms de variables locales dans des fonctions différentes.
- Les variables locales peuvent avoir le même nom que certaines variables globales du programme sans interférer.

exemple:

```
def double(x):
    y=2*x
    return y

def carre(x):
    z=x*x
    return z

def plusgrand(x):
    if y>=z:
        return "le double est plus grand que le carre"
    return "le carre est plus grand que le double"
```

On exécute le programme et on appelle `plusgrand(3)` dans la console. Que renvoie l'ordinateur? Justifier et corriger si nécessaire.

Exemple: Quel(s) programme(s) affiche(nt) 9 ?

1.

```
def f(x) :
    s=x**2
    return s
f(3)
print(s)
```

2.

```
def f(x) :
    s=x**2
    return s
s=f(3)
print(s)
```

3.

```
def f(x) :
    x**2
    return x
print(f(3))
```

3 Exercices

Exercice 1 :

1. Montrer que: pour tous réels x et y ,

$$\max(x, y) = \frac{(x + y) + |x - y|}{2} \text{ et } \min(x, y) = \frac{(x + y) - |x - y|}{2}.$$

2. Écrire deux fonctions qui calculent le maximum et le minimum de deux réels quelconques.
sans utiliser les fonctions `max()` et `min()`

Exercice 2 :

1. Écrire une fonction qui prend en entrée un entier n , qui renvoie n si n est pair et $n - 1$ si n est impair.
2. Écrire une fonction `signe(x)` qui, étant donné un réel x renvoie le signe de x , i.e. -1 si x est strictement négatif, 0 si x est nul et 1 si x est strictement positif.

Exercice 3 :

1. Écrire une fonction qui, étant donné un entier n , renvoie `True` si n est pair, et `False` sinon.
On pourra penser au reste par la division euclidienne par 2.
2. Écrire une fonction qui, étant donné deux entiers n et m , renvoie `True` si les deux entiers ont la même parité, et `False` sinon.
3. Pourrait-on se passer de l'instruction `if` dans les questions précédentes ?

Exercice 4 Fonction rampe

La fonction rampe est définie comme suit :

$$R(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases}$$

Écrire une fonction `Rampe(x)` qui renvoie la valeur de la fonction rampe en `x`

Exercice 5 Puissance réelle

Soit $\alpha \in \mathbb{R}$ et x un réel strictement positif. On définit la puissance réelle x^α par :

$$x^\alpha = e^{\alpha \ln x}$$

Écrire une fonction `puissance_reelle` (*sans utiliser l'opérateur `**`*) qui, étant donnés deux réels x et y :

- renvoie un message d'erreur si $x \leq 0$,
- calcule et renvoie $x^y = e^{y \ln x}$ si $x > 0$.

Tester dans la console : `puissance_reelle(2,3) == 2**3`

Que renvoie l'ordinateur ? Justifier .