

```

# TP1 BCP1B 22-23.py

# TP1

## exo 2

# q1
# x4 vaut 200

# q2
y1,y2 = 1,4
y3, y4 = y1 + y2/2, y3**2
# ce code renvoie un message d'erreur car y3 n'est pas encore défini avant qu'on
# parle de y3**2
# il faudrait faire
y1,y2 = 1,4
y3 = y1 + y2/2
y4 = y3**2

#q3
# z4 vaut 1

## exo 3

# q1
a = 5
b = 5

##
# q2
a = a-1 # on enlève un diamant dans A
b = b+1 # on ajoute un diamant dans B
##

# q3
# on compile plusieurs fois de suite les cellules précédentes

## exo 4

# q1
# non

# q3
# non
# on dit que la variable c est locale : elle n'a de sens qu'au sein de la définition
# de la fonction, pas en dehors

## exo 5

# q1 et q2
def f(x,y):
    num1 = x-y**2
    denom1 = 2/x + 1/(2*y)
    num2 = 3*x**2
    denom2 = x**2 + 2*y**2 + 2
    return num1/denom1 + num2/denom2

# q3
def g(x,y):
    return 3*x**2 + y**2/2

# q4
# dans la console on teste, par exemple, que g(-2,4) renvoie bien 20

```

```

# q5
def h(x,y):
    t = g(x,y)
    a = (t+1)/(t**2 + 1)
    b = (x + t**3)/(y**2+1)
    return a,b

# q6
# à la question 5, il suffit de demander à Python de calculer une fois g(x,y). En
particulier, le code suivant n'est pas le meilleur :
def h_mauvais(x,y):
    a = (g(x,y)+1)/(g(x,y)**2 + 1)
    b = (x + g(x,y)**3)/(y**2+1)
    return a,b

## exo 6

# q1 et q2
def aire(a,b,c):
    s = (a+b+c)/2
    produit = s * (s-a) * (s-b) * (s-c)
    return produit**(1/2)

# on a bien aire(3,4,5) = 6 = 4*3/2 = hauteur * largeur /2

# q3
# on a un message d'erreur car la quantité sous la racine est négative. En fait, il
n'existe pas de triangle de côtés 2, 1 et 5.

# q4
# ce sont des inégalités triangulaires car elles indiquent que la somme des longueurs
de 2 des côtés d'un triangle est supérieur à la longueur du troisième côté.

# q5
# N'hésitez pas à venir me proposer une démonstration.
# On pourra aussi consulter, à titre culturel, la page Wikipédia concernant la
formule de Héron :
# https://en.wikipedia.org/wiki/Heron%27s\_formula

```