solutions_2

January 29, 2024

1 Solutions exercices 2

`Import math` nécessaire pour utiliser certains outils math dans les fonctions ci-dessous.

> Note importante: il y a plusieurs variations possibles de code qui fonctionne. Prenez le temps de tester toutes les possibilités pour vous assurer que chaque type d'entrée aura la réponse espérée.

Retour aux questions

2 Question 1

2.0.1 a)

```
def quad(nombre):
    # on peut aussi affecter le résultat dans une variable locale et retourner la variable
    resultat = 4*nombre
    return resultat

print('Q1a) :',quad(9))
```

2.0.2 b)

```
def cinetique(masse, vitesse):
    # On peut aussi retourner directement le résultat dans le return
    reponse = 1/2 * masse * vitesse**2
    return reponse

print('Q1b) : Une Ford Fiesta de 1300 Kg qui roule à 50 km/h (donc 13.89 m/s) dépenserait', rou
```

2.0.3 c)

```
def distance(x_1, y_1, x_2, y_2):
    distance_entre_2 = math.sqrt((x_2 - x_1)**2 + (y_2 - y_1)**2)
    return distance_entre_2

print('Q1c) : La distance entre le point (2,3) et le point (5,-1) est ', distance(2,3,5,-1))
```

2.0.4 d)

```
def arc_cercle(rayon, angle):
    return angle/360 *2*math.pi* rayon
print("Q1d) : L'arc engendré par un rayon de 4 et un angle de 130 est de :", round(arc_cercle(4
```

2.0.5 e)

```
def max_5(n1, n2, n3, n4, n5):
    val_max = max(n1, n2, n3, n4, n5)
    return val_max
n1= 4
n2=-10
n3= 50
n4=-100
n5=9.5
print("Q1e) : le max entre les nombres", n1, ',',n2, ',',n3, ',', n4, ',',n5, "est: ", max_5(n1
```

```
print()
print()
```

2.0.6 f)

```
def hauteur(heure):
    h = 5.9*math.sin(math.pi * heure /6 -1.97) +6.9
    return h

temps_journee = 6
print("Q1f) La hauteur de la marrée pour", temps_journee,"heure est",hauteur(temps_journee), "m
```

2.0.7 g)

Aucun corrigé fourni pour cette question! À vous de présenter une solution logique :)

3 Question 2

3.0.1 a)

```
def pair_impair(x):
    if (x % 2 == 0) :
        return 'pair'
    else:
        return 'impair'
x= 19
print('Q2a) : avec x=',x, ', on obtient', pair_impair(x))
```

3.0.2 b)

```
def temperature_corporelle(chiffre, unites):
    resultat = ''
    if (unites == 'F'):
        if chiffre > 100.4:
            resultat = 'fièvre'
        else:
            resultat= 'pas de fièvre'
    elif (unites == 'C'):
        if chiffre > 38:
            resultat = 'fièvre'
        else:
            resultat= 'pas de fièvre'
    else:
        print("Le format de l'unité de mesure entré n'est pas bon")
        resultat = 'inconnu'

    return resultat

print("Q2b) : un patient qui fait 39 Celsius de température a comme résultat: ", temperature_c
```

3.0.3 c)

```
def quadratique(a,b,c):
    determinant = b**2 - 4*a*c
    zero_1 = None
    zero_2 = None
    if determinant < 0 :
        #inutile de changer la valeur des variables zero_1 et zero_2, qui restent à None
        print("Il n'y a pas de zéros puisque le déterminant est négatif")
    elif determinant == 0:
        # Si le déterminant est 0, on a juste une réponse (on a en fait une réponse double, ma
        print("Il n'y a qu'un seul zéro")
        zero_1 = (-b)/(2*a)
    else:
        print("Il y a 2 zéros")
        zero_1 = (-b + math.sqrt(determinant))/(2*a)
        zero_2 = (-b - math.sqrt(determinant))/(2*a)

    return (zero_1, zero_2)

print("pour -1,8,4:", quadratique(-1,8,4))
```

3.0.4 d) i)

```
def imposition(salaire):
    impot_a_payer = 0
    if (salaire >= 0 and salaire <= 51780):
```

```
        impot_a_payer = 14/100 * salaire
    #Dans le elif, on peut ajouter and salaire >51780, c'est syntaxiquement correct, mais inut
    elif (salaire < 103545):
        impot_a_payer = 19/100 * salaire
    #idem au elif précédent
    elif (salaire < 126000):
        impot_a_payer = 24/100*salaire
    elif (salaire >=126000):
        impot_a_payer = 25.75/100*salaire
    else:
        print("Le salaire entré n'est pas valide")

    return impot_a_payer

s= 100000
print("Un salaire de", s, "$ sera soumis à", imposition(s), "$ d'impôts")
```

3.0.5 d) ii)

```
def imposition2(salaire):
    impot_a_payer = 0
    '''
    Truc! pour éviter de changer les chiffres partout quand les impôts changeront,
    il est préférable de les déclarer à un seul endroit. Ceci est une optimisation FACULTATIVE
    '''
    limite_1 = 51780
    limite_2 = 103545
    limite_3 = 126000
    pourc_1 = 14/100
    pourc_2 = 19/100
    pourc_3 = 24/100
    pourc_4 = 25.75/100

    if (salaire >= 0 and salaire <= limite_1):
        impot_a_payer = pourc_1 * salaire
    elif (salaire < limite_2):
        impot_a_payer = pourc_2 * (salaire - limite_1) + pourc_1 * limite_1
    #idem au elif précédent
    elif (salaire < limite_3):
        impot_a_payer = pourc_3 * (salaire - limite_2) + pourc_2 * (limite_2 - limite_1) + pou
    elif (salaire >= limite_3):
        impot_a_payer = pourc_4 * (salaire - limite_3) +
        pourc_3 * (limite_3 - limite_2) + pourc_2 * (limite_2 - limite_1) + pourc_1 * limite_1
    else:
        print("Le salaire entré n'est pas valide")

    return impot_a_payer
```

```
s= 100000
print("Q2d)ii) : Un salaire de", s, "$ sera soumis à", imposition2(s), "$ d'impôts")
```

3.0.6 Q2e)

```
#Pour mettre une valeur par défaut, on met le paramètre = la valeur par défaut
def force(F=None, m=None, a=None):
    #Vérifier que F est la seule valeur non déclarée
    if (F == None and m != None and a != None):
        F = m*a
        print("La force F dans un système qui a une masse de", m, "kg qui accélère à", a, "m/s
    elif (F !=None and m == None and a != None):
        m = F/a
        print("La masse dans un système qui a une force de", F, "Newtons et qui accélère à", a
    elif (F !=None and m != None and a == None):
        a = F/m
        print("L'accélération dans un système qui a une force de", F, "Newtons et qui a une mas
    else:
        print("Il manque trop de paramètres ou bien ceux-ci ne sont pas numériques.")

force(15, 4) #force de 15, masse de 4
force(F=15, a=4) #force de 15, accélération de 4
```

Il est important de tester plusieurs valeurs pour s'assurer que la fonction traite tous les cas.

[ ]: