



1. Dans chaque cas, écrire les fonctions Python indiquées et tester ces fonctions en console.
 - a. Écrire une fonction Python permettant de calculer une vitesse v (en $km \cdot h^{-1}$) à partir d'une distance d parcourue (en km) et de la durée t écoulée (en h).
 - b. Écrire une fonction Python permettant de calculer la moyenne de deux nombres.
2. Le prix TTC (toutes taxes comprises) d'un article se calcule en additionnant le prix HT (hors taxe) et la TVA (taxe sur la valeur ajoutée).

➤ **Partie A**

Dans cette partie, on considère que le taux de TVA à appliquer est de 20%.

- a. Quelle opération permet de calculer le montant de la TVA (en €) à partir du prix HT (en €) ? Écrire une fonction Python TVA qui reçoit en argument un prix HT et qui renvoie le montant de cette TVA.
- b. Utiliser cette fonction pour calculer la TVA d'un article coûtant 17 € HT.
- c. Écrire une fonction Python TTC qui reçoit en argument un prix HT et qui renvoie le prix TTC. Votre fonction utilise-t-elle la fonction TVA précédente ? Si ce n'est pas le cas, adapter votre fonction.
- d. Utiliser cette fonction Python pour calculer le prix TTC d'un article coûtant 34€ HT.

➤ **Partie B**

Dans cette partie, le taux de TVA est variable.

- a. Ajouter un argument aux fonctions écrites dans la partie A pour qu'on puisse modifier le taux de TVA lors de l'appel aux fonctions.
- b. Utiliser ces fonctions pour calculer le prix TTC d'un article coûtant 25€ HT avec un taux de TVA à 5%.



Découvrir des syntaxes

3. Fonction avec valeur par défaut

- a. Saisir la fonction Python `double` ci-contre.
- b. Tester en console les saisies suivantes : `double(1)` ; `double(6)` et `double()`.
Indiquer, dans chaque cas, quelle valeur de x a été utilisée pour le calcul dans la fonction.

```
def double(x=3):  
    return 2*x
```

4. Fonctions à valeurs booléennes

Saisir la fonction Python `test_pos` ci-contre.

- a. Tester cette fonction avec différentes valeurs de x .
À quelle question cette fonction répond-t-elle ?
- b. Écrire une fonction Python qui reçoit en argument une valeur x et renvoie `True` si et seulement si x est compris dans l'intervalle $[-5; 5]$.
- c. Écrire une fonction Python qui reçoit en argument une valeur entière x et renvoie `True` si et seulement si x est divisible par 3.

```
def test_pos(x):  
    return x >= 0
```

Aide : La syntaxe Python `a%b` permet d'obtenir le reste de la division euclidienne de a par b .

Compléments



Cliquer ici pour accéder à
une fiche dynamique
des syntaxes élémentaires

