

# Réciproque du théorème de Pythagore



**Résumé :** cette fiche porte sur la programmation de la réciproque du théorème de Pythagore.

**Mots-clés :** bibliothèque `turtle` ; réciproque du théorème de Pythagore

## Compétences visées

**Chercher :** « Observer, s'engager dans une démarche, expérimenter en utilisant éventuellement des outils logiciels ».

**Modéliser :** « Utiliser, comprendre, élaborer une simulation numérique » en partant d'une situation usuelle rencontrée fréquemment en mathématiques.

**Représenter :** « passer d'un mode de représentation à un autre » en transférant des calculs sur feuille à un code.

**Calculer :** « mettre en œuvre des algorithmes simples ».

## Situation déclenchante

Il s'agit d'une séance interne aux mathématiques dans laquelle les élèves doivent programmer la réciproque au théorème de Pythagore.

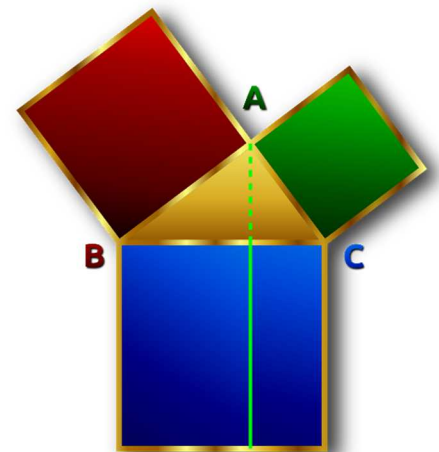


Image libre de droits d'après [Pixabay](#)

## Problématique

Quel programme pour indiquer si un triangle est rectangle ou non, connaissant ses trois longueurs ?



# Réciproque du théorème de Pythagore



## Scénario pédagogique

- **Avec la classe** : on pourra proposer beaucoup de situations impliquant la problématique, comme par exemple : « Le triangle 10 ; 12 ; 14 est-il rectangle ? », ce qui devrait faire émerger l'intérêt d'un script permettant d'indiquer si un triangle est rectangle ou non.
- **En groupe de 2 à 4 élèves pendant quelques minutes** : les élèves cherchent des stratégies pour répondre à la problématique. Beaucoup veulent passer par un aspect graphique (voir la fiche n°10, Le L). La réciproque du théorème de Pythagore n'émerge pas de suite pour certains groupes.
- **Avec la classe** : faire un point en listant les différentes propositions, en demandant finalement quelles sont les entrées (les trois longueurs du triangle), ce qui, par conséquent, est possible ou non. Par exemple, l'utilisation d'un dessin nécessite la trigonométrie et par conséquent de savoir que le triangle est rectangle.
- **En groupe de 2 à 4 élèves** : les élèves doivent faire une ébauche d'algorithme de la réciproque du théorème de Pythagore, avant de passer à l'implantation dans un langage ou un autre, après validation du professeur. Cette partie nécessite un temps assez long pour que les élèves s'approprient le script.
- **Mise en commun** : les élèves expliquent leurs démarches.
- **Pour les élèves les plus en avance** : il est possible de leur proposer un ou plusieurs prolongements possibles, décrit en [fin de fiche](#).
- **Difficultés rencontrées** :
  - prévoir qu'une donnée soit celle du côté le plus grand ;
  - mettre les données au carré.

Voici les visuels à l'issue des programmes :

en Scratch



avec la TI-83 Premium CE Edition Python



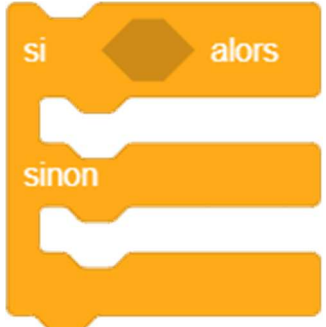
```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de T10_PYTH
>>> from T10_PYTH import *
>>> test(10,12,14)
'triangle non rectangle'
>>> |
```

Fns... a A # Outils Éditer Script

## Réciproque du théorème de Pythagore



## Avec Scratch

Les briques de codes principales en Scratch pour ce programme	Explications Cette instruction permet de :	Traduction en langage Python sur la TI-83 Premium CE Edition Python
	Affecter une valeur à la variable a, 0 ici ;	<code>a=0</code>
	Mettre au carré la réponse demandée à l'utilisateur ; il n'existe pas de fonction puissance dans Scratch ;	<code>a**2</code> ** désigne la séquence puissance dans Python
	Créer une structure de contrôle conditionnelle.	<pre> if condition:     **instructions si vraie else:     **instructions si faux           </pre> L'indentation avec <code>**</code> est nécessaire pour définir les instructions à exécuter dans un bloc du code Python.

Une programmation possible est disponible sur le site de Scratch : [scratch.mit.edu/studios/27615196/](https://scratch.mit.edu/studios/27615196/)

## Réciproque du théorème de Pythagore



## Avec Python

Le code complet est à construire par les élèves.

Ce code est composé de deux fonctions, seule la première est attendue des élèves, la seconde permet une ouverture vers les triplets pythagoriciens :

- La fonction `test` de paramètres `a` ; `b` et `c` qui sont les trois longueurs du triangle à tester.

Les paramètres sont insérés dans la liste `li`, ce qui permet de les ordonner de façon croissante avec la fonction `li.sort()`.

Deux options pour le retour à l'utilisateur : soit par un message texte, soit par un retour sous forme de booléen `True/False`.

Cette deuxième option permet le réinvestissement de la fonction `test` dans la fonction suivante.

```

ÉDITEUR : T10_PYTH
LIGNE DU SCRIPT 0001
def test(a,b,c):
    li=[a,b,c]
    li.sort()
    a=li[0]
    b=li[1]
    c=li[2]
    if a**2+b**2==c**2:
        return True
    #return "triangle rectangle"
    else:
        return False
    #return "triangle non rectan
    gle"

```

- La fonction `triplet` de paramètres `a` et `n_rep` qui sont respectivement la longueur de départ et l'étendue à tester.

Ainsi, à partir d'une valeur (longueur) déterminée, tous les triplets entiers sont testés depuis `(a ; a ; a)` jusqu'à `(a ; a+n_rep ; a+n_rep)`. Le script s'arrête à la première valeur trouvée ou indique qu'il n'y a pas de tels triplets sur l'intervalle testé.

Cela est rendu possible grâce à la fonction `test`. Il est pratique de renvoyer un booléen.

A noter que dans la boucle `for`, il faut aller jusqu'à `n_rep+1` puisque la boucle commence à 0 et s'arrête à l'entier qui précède `n_rep+1`.

```

ÉDITEUR : T10_PYTH
LIGNE DU SCRIPT 0012
def triplet(a,n_rep):
    for i in range(n_rep+1):
        for j in range(n_rep+1):
            if test(a,a+i,a+j):
                return a,a+i,a+j
    return "pas trouvé"

```

Une programmation possible est disponible sur le site TI : [education.ti.com/fr/scratch-python](http://education.ti.com/fr/scratch-python)



# Réciproque du théorème de Pythagore



## Mode opératoire

Une fois le script exécuté, il faut appuyer sur la touche [var] : les deux fonctions définies dans le script apparaissent : `test` et `triplet`.

Par les flèches directionnelles, il faut sélectionner la fonction `test`, valider par `Ok`, puis rajouter les trois longueurs du triangle à tester en séparant les paramètres par une virgule.

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de T10_PYTH
>>> from T10_PYTH import *
>>> test(5,12,13)
Fns... a A # Outils Éditer Script
```

## Prolongements possibles

Voici des pistes pour les élèves les plus rapides ou qui ont envie de prolonger le travail :

- décliner un programme permettant d'indiquer en quel point le triangle est rectangle ;
- faire un algorithme de tri ;
- déterminer un triplet pythagoricien à partir d'un nombre et d'une étendue (script `triplet`) ;
- déterminer la mesure des angles dans le cas d'un triangle rectangle.

