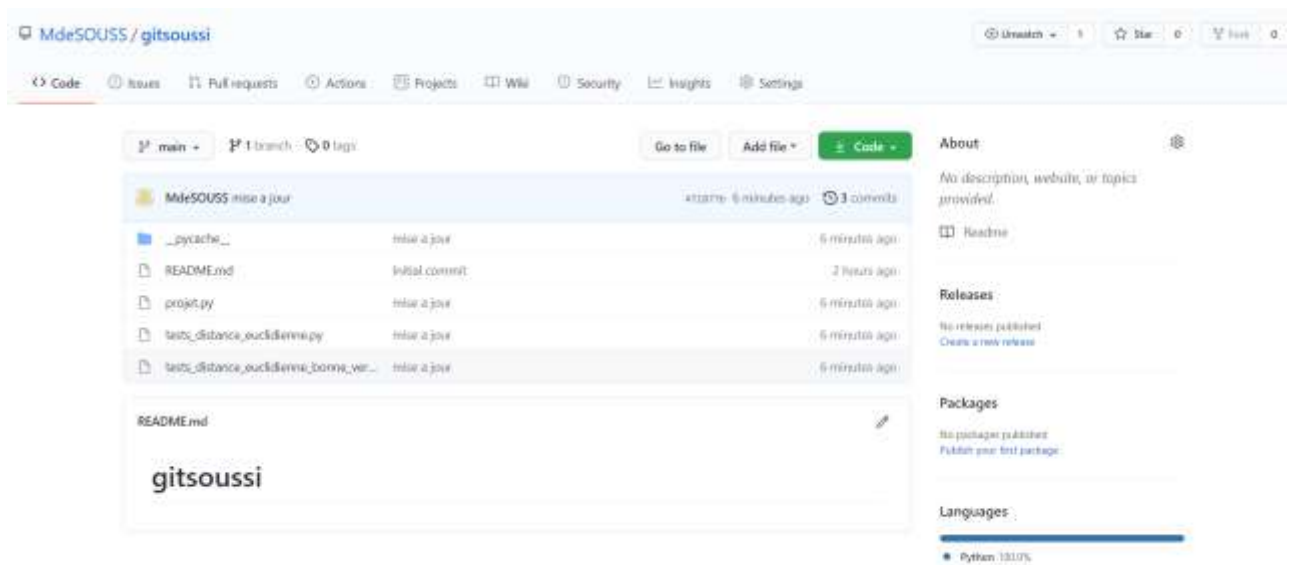


Monde de l'entreprise



Sauvegarde projet GIT sur Github



Lien du github : [MdeSOUSS/gitsoussi \(github.com\)](https://github.com/MdeSOUSS/gitsoussi)

Commandes pour mettre a jour sur le site :

```
souss@DESKTOP-ENE79D3 MINGW64 ~/Desktop/MathiasSOUSSIGIT/gitsoussi (main)
$ git add
Nothing specified, nothing added.
hint: Maybe you wanted to say 'git add .'
hint: Turn this message off by running
hint: "git config advice.addEmptyPathspec false"

souss@DESKTOP-ENE79D3 MINGW64 ~/Desktop/MathiasSOUSSIGIT/gitsoussi (main)
$ git add *

souss@DESKTOP-ENE79D3 MINGW64 ~/Desktop/MathiasSOUSSIGIT/gitsoussi (main)
$ git commit -a -m "mise a jour"
[main 471877b] mise a jour
4 files changed, 79 insertions(+), 1 deletion(-)
create mode 100644 __pycache__/projet.cpython-37.pyc
create mode 100644 tests_distance_euclidienne.py
create mode 100644 tests_distance_euclidienne_bonne_version.py

souss@DESKTOP-ENE79D3 MINGW64 ~/Desktop/MathiasSOUSSIGIT/gitsoussi (main)
$ git push -u origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 1.59 KiB | 813.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MdeSOUSS/gitsoussi.git
01b5261..471877b main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.

souss@DESKTOP-ENE79D3 MINGW64 ~/Desktop/MathiasSOUSSIGIT/gitsoussi (main)
$ |
```

Lors du début de ce Tp je 'avais pas compris ce qu'il fallait faire et j'ai donc créer un programme a base de if et de else pour tester la validité de la fonction. Environ 1 heure avant la fin du temp règlementaire pour rendre le tp un camarade m'a fait remarquer mon erreur et donc il ya deux fichier test dont l'un avec la bonne méthode, la méthode TDD

Programme Main :

```
# Créé par souss, le 24/04/2021 en Python 3.7
import numpy as np
point_a = np.array((0,0))
point_b = np.array((1,1))

def distance_euclidienne(a,b):
    distance = np.linalg.norm(a-b)
    return distance

print(distance_euclidienne(point_a,point_b))
```

Test avec TDD :

```
# Créé par souss, le 30/04/2021 en Python 3.7
import numpy as np
import math
from projet import distance_euclidienne
from random import randint
import unittest
class test_distance_euclidienne(unittest.TestCase):
    def setUp(self):
        point_a = np.array((randint(0,10000),randint(0,10000)))
        point_b = np.array((randint(0,10000),randint(0,10000)))
        point_c = np.array((randint(0,10000),randint(0,10000)))
    def PremierTest(self):
        self.assertEqual(0,distance_euclidienne(point_a,point_a))
    def DeuxièmeTest(self):
        self.assertGreaterEqual(distance_euclidienne(point_a,point_b),0)
    def TroisièmeTest(self):

self.assertEqual(distance_euclidienne(point_a,point_b),distance_euclidienne
(point_b,point_a))
    def QuatrièmeTest(self):
        ab = distance_euclidienne(point_a,point_b)
        ac = distance_euclidienne(point_a,point_c)
        bc = distance_euclidienne(point_b,point_c)
        somme = ab + bc
        self.assertLessEqual(ac,somme)
    def Derniertest(self):
        self.assertAlmostEqual(distance_euclidienne((0,1),(1,0)),sqrt(2))
if __name__ == '__main__':
    unittest.main()
```

Test BARBARE :

```
# Créé par souss, le 24/04/2021 en Python 3.7
import numpy as np
from projet import distance_euclidienne
from random import randint

Bilan = True

point_a = np.array((randint(0,10000),randint(0,10000)))
point_b = np.array((randint(0,10000),randint(0,10000)))
point_c = np.array((randint(0,10000),randint(0,10000)))
if distance_euclidienne(point_a,point_a) == 0:
    Bilan = True
else :
    Bilan = False
if Bilan == False:
    print("echec du test")

if distance_euclidienne(point_b,point_b) == 0:
    Bilan = True
else :
    Bilan = False
if Bilan == False:
    print("echec du test")

if distance_euclidienne(point_a,point_b) == 0 or
distance_euclidienne(point_a,point_b) > 0:
    Bilan = True
else :
    Bilan = False
if Bilan == False:
    print("echec du test")

ab = distance_euclidienne(point_a,point_b)
ac = distance_euclidienne(point_a,point_c)
bc = distance_euclidienne(point_b,point_c)
somme = ab + bc
if ac < somme or ac == somme :
    Bilan = True
else :
    Bilan = False
if Bilan == False:
    print("echec du test")
```