

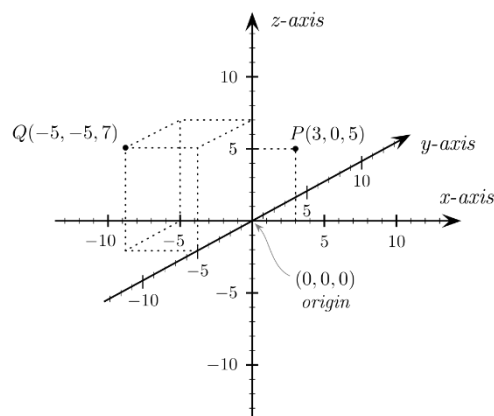
Machine Learning on Material Informatics
Development Environment For Machine Learning
20190312-Homework 01

TASK

Exercise 3: determine a point in which quadrant without using "if ...". Say (-3, 1, 3)

□ Quadrant:

I: + + +
II: + - +
III: - - +
VI: - + +
V: + + -
VI: + - -
VII: - - -
VIII: - + -



SOLUTION

```
#return from 1 to 8
```

```
def checkQuadrants(x,y,z) :
```

```
    returnuadrants = [8,-1,5,7,4,6,1,3,-1,2] # by magic
```

```
    sx = x/abs(x) # sign of x
```

```
    sy = y/abs(y) # sign of y
```

```
    sz = z/abs(z) # sign of z
```

```
    return    returnuadrants[(int)((((2*sx-3*sy+4*sz)+9)/2))] #magic
```

```
#mapping from the magic formula to the quadrants
```

```
#1 +++ 6
```

```
#2 +-+ 9
```

```
#3 --+ 7
```

```
#4 -++ 4
```

```
#5 +- - 2
```

```
#6 +-- 5
```

```
#7 --- 3
```

```
#8 -+- 0
```

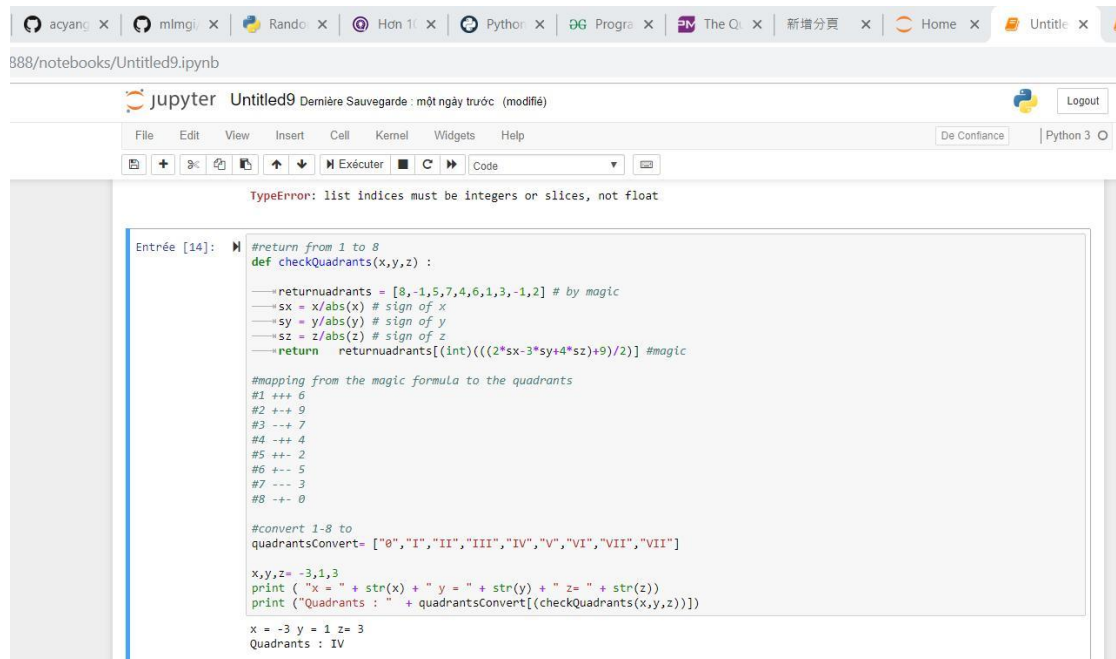
#convert 1-8 to

```
quadrantsConvert= ["0","I","II","III","IV","V","VI","VII","VIII"]
```

x,y,z= -3,1,3

```
print ( "x = " + str(x) + " y = " + str(y) + " z = " + str(z))
```

```
print ("Quadrants : " + quadrantsConvert[(checkQuadrants(x,y,z))])
```



The screenshot shows a Jupyter Notebook window titled 'Untitled9'. The interface includes a top bar with browser tabs and a menu bar with options like File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu bar is a toolbar with icons for running, saving, and other actions. The main area displays a code cell with the following Python code:

```
Entrée [14]: #return from 1 to 8
def checkQuadrants(x,y,z) :
    #return quadrants = [8,-1,5,7,4,6,1,3,-1,2] # by magic
    #sx = x/abs(x) # sign of x
    #sy = y/abs(y) # sign of y
    #sz = z/abs(z) # sign of z
    #return returnquadrants[(int)((((2*sx-3*sy+4*sz)+9)/2))] #magic

#mapping from the magic formula to the quadrants
#1 +++ 6
#2 +-+ 9
#3 --+ 7
#4 -++ 4
#5 +++ 2
#6 +-- 5
#7 --- 3
#8 -+- 0

#convert 1-8 to
quadrantsConvert= ["0","I","II","III","IV","V","VI","VII","VIII"]

x,y,z= -3,1,3
print ( "x = " + str(x) + " y = " + str(y) + " z = " + str(z))
print ("Quadrants : " + quadrantsConvert[(checkQuadrants(x,y,z))])

x = -3 y = 1 z= 3
Quadrants : IV
```

Below the code cell, a red error message is displayed: 'TypeError: list indices must be integers or slices, not float.' This indicates that the function 'checkQuadrants' is returning a float value, which cannot be used as an index for the 'quadrantsConvert' list.