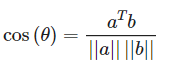
¿Qué ángulo forman los vectores v1 = [0,1] y v2 = [1,0]?

90 grados, porque norma\_v1 \* norma\_v2 \* np.cos(np.deg2rad(90)) da 0



angle = np.math.atan2(np.linalg.det([v0,v1]),np.dot(v0,v1))

print (np.degrees(angle))



aTb = np.dot(v1,v2)

= np.linalg.norm(v1) x np.linalg.norm(v1) x np.cos(np.deg2rad(Angulo))

|  |  |
| --- | --- |
|  |  |
|  |  |

Graphical user interface, application

Description automatically generated

Ambos nos dan 9.0 para 45 grados

Si el Angulo tiende a ‘0’ hay similitude entre ambos vectores/matrices

V1.T.dot(v2) = v1\_norma x v2\_norma x cos(Angulo)

np.cos(no.deg2rad(90)

import numpy as np

v0 = np.array([0,3])

v1 = np.array([3,3])

angle = np.math.atan2(np.linalg.det([v0,v1]),np.dot(v0,v1))

print (np.degrees(angle))

r/ 45

===============

**CALCULE ANGULO, 3 VECTORES**

import numpy as np

p0 = [3.5, 6.7]

p1 = [7.9, 8.4]

p2 = [10.8, 4.8]

v0 = np.array(p0) - np.array(p1)

v1 = np.array(p2) - np.array(p1)

**angle = np.math.atan2(np.linalg.det([v0,v1]),np.dot(v0,v1))**

print (np.degrees(angle))

R/

107

Con vectores en 3D, puede hacerlo de forma concisa utilizando el cinturón de herramientas [vg](https://github.com/lace/vg) . Es una capa ligera encima de numpy.

import numpy as np

import vg

vec1 = np.array([1, 2, 3])

vec2 = np.array([7, 8, 9])

vg.angle(vec1, vec2)

También puede especificar un ángulo de visión para calcular el ángulo a través de la proyección:

vg.angle(vec1, vec2, look=vg.basis.z)

O calcula el ángulo firmado a través de la proyección:

vg.signed\_angle(vec1, vec2, look=vg.basis.z)

[Hace un año](https://platzi.com/comentario/1960813/)

3

Text

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated