[Actividad extracurricular 09] función atan2

Realizado por Correa Adrian

fecha: 17/12/2024

link github: https://github.com/afca2002/ActividadExtracurricular09.git

Investigue y resuma la función atan2.

¿Por qué se recomienda usar la función atan2?

Diferencias entre función atan y atan2.

Al menos 5 ejemplos en python.

Función `atan(x/y)`

Cuando se calcula un ángulo a partir de un cociente `x/y` utilizando `atan(x/y)`, se está obteniendo el ángulo cuya tangente es `x/y`. Sin embargo, esta forma posee inconvenientes:

- **Falta de información sobre el cuadrante:** Usar `atan(x/y)` solo obtiene el ángulo principal entre `-π/2` y `π/2`, sin distinguir en qué cuadrante se encuentra el vector definido por las componentes `(x, y)`.
- **No toma en cuenta el signo de x e y por separado:** Conocer únicamente la razón `x/y` no permite diferenciar, por ejemplo, entre `(x, y) = (1, 1)` y `(x, y) = (-1, -1)`, ya que `1/1` y `(-1)/(-1)` dan el mismo resultado.

En consecuencia, `atan(x/y)` es limitado para calcular el ángulo de un vector en el plano cartesiano cuando se requiere el cuadrante correcto.

Función `atan2(x, y)`

La función `atan2(y, x)` (tenga en cuenta que el orden estándar es `(y, x)`) está diseñada para superar las limitaciones de `atan(y/x)`:

- **Uso estándar:** `atan2(y, x)` recibe dos argumentos: el primero es la componente `y` y el segundo la componente `x`. (En el enunciado se solicita `atan2(x, y)`, pero la convención es `atan2(y, x)`. Si se invierten los argumentos, el resultado representará un ángulo incorrecto.)
- **Rango completo:** Devuelve un ángulo entre `-π` y `π`, identificando correctamente el cuadrante.

- **Información total del signo:** Al tener acceso directo a `x` y `y`, `atan2` determina el cuadrante real del vector `(x, y)` y por tanto el ángulo apropiado.

¿Por qué se recomienda usar la función `atan2`?

`atan2` se recomienda en lugar de `atan(x/y)` porque:

1. Distingue todos los cuadrantes del plano.
2. No requiere lógica adicional para determinar el signo de `x` y `y`.
3. Permite convertir coordinadas cartesianas `(x, y)` a coordenadas polares (r, θ) de manera más robusta.
4. Evita ambigüedades cuando `x` es cero, ya que no se produce una división por cero, mientras que `atan(x/y)` sí podría generarla.

Diferencias entre `atan` y `atan2`

1. **Número de argumentos:**
   - `atan` recibe un solo argumento (una razón), por ejemplo `atan(x/y)`.
   - `atan2` recibe dos argumentos (y, x), por ejemplo `atan2(y, x)`.
2. **Rango del resultado:**
   - `atan` devuelve un ángulo en `(-π/2, π/2)`.
   - `atan2` devuelve un ángulo en `(-π, π)`.
3. **Determinación del cuadrante:**
   - `atan` no permite identificar el cuadrante sin información adicional.
   - `atan2` identifica el cuadrante automáticamente.
4. **Manejo de casos especiales:**
   - `atan(x/y)` puede generar una indeterminación si `y = 0`.
   - `atan2(y, x)` está definida para todos los pares `(x, y)` excepto `(0,0)` y maneja correctamente casos como `x = 0`.
5. **Aplicaciones:**
   - `atan` es útil en cálculos simples donde solo se necesita el ángulo principal.
   - `atan2` es esencial en geometría del plano, robótica, gráficos por computadora, y en cualquier situación donde el cuadrante correcto sea relevante.

A menos 5 Ejemplos en Python

In [ ]:
```
import math

# Ejemplo 1
x, y = 1, 1
print("Ejemplo 1:")
print("atan(x/y):", math.atan(x/y))
print("atan2(y,x):", math.atan2(y, x))
print()
```

```
# Ejemplo 2
x, y = -1, 1
print("Ejemplo 2:")
print("atan(x/y):", math.atan(x/y))
print("atan2(y,x):", math.atan2(y, x))
print()

# Ejemplo 3
x, y = -2, -2
print("Ejemplo 3:")
print("atan(x/y):", math.atan(x/y))
print("atan2(y,x):", math.atan2(y, x))
print()

# Ejemplo 4
x, y = 2, -3
print("Ejemplo 4:")
print("atan(x/y):", math.atan(x/y))
print("atan2(y,x):", math.atan2(y, x))
print()

# Ejemplo 5
x, y = 0, 5
print("Ejemplo 5:")
print("atan(x/y):", math.atan(x/float(y)))
print("atan2(y,x):", math.atan2(y, x))
print()

# Ejemplo 6
x, y = 0, -4
print("Ejemplo 6:")
print("atan(x/y):", math.atan(x/float(y)))
print("atan2(y,x):", math.atan2(y, x))
print()

# Ejemplo 7
x, y = -3, -4
print("Ejemplo 8:")
print("atan(x/y):", math.atan(x/y))
print("atan2(y,x):", math.atan2(y, x))
print()
```

```
Ejemplo 1:
atan(x/y): 0.7853981633974483
atan2(y,x): 0.7853981633974483

Ejemplo 2:
atan(x/y): -0.7853981633974483
atan2(y,x): 2.356194490192345

Ejemplo 3:
atan(x/y): 0.7853981633974483
atan2(y,x): -2.356194490192345

Ejemplo 4:
atan(x/y): -0.5880026035475675
atan2(y,x): -0.982793723247329

Ejemplo 5:
atan(x/y): 0.0
atan2(y,x): 1.5707963267948966

Ejemplo 6:
atan(x/y): -0.0
atan2(y,x): -1.5707963267948966

Ejemplo 8:
atan(x/y): 0.6435011087932844
atan2(y,x): -2.214297435588181
```

In [1]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```