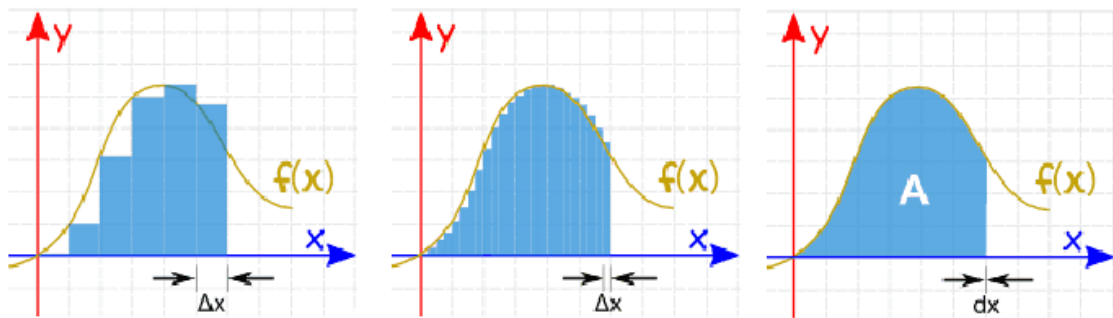


# **DERIVADAS EN PHYTON**



**Alejandro Sánchez Moreno**

**2Bach-E**

## **ÍNDICE**

1. **Explicación de derivadas**
2. **Código de la derivada en  
python**
3. **Fotos del código**
4. **Resultado de la derivada**

# **1.Explicación de la derivada**

En matemáticas, la derivada de una función mide la rapidez con la que cambia el valor de dicha función matemática, según cambie el valor de su variable independiente. La derivada de una función es un concepto local, es decir, se calcula como el límite de la rapidez de cambio media de la función en cierto intervalo, cuando el intervalo considerado para la variable independiente se torna cada vez más pequeño. Por ello se habla del valor de la derivada de una función en un punto dado.

El concepto de derivada es uno de los conceptos básicos del Análisis matemático. Los otros son los de integral indefinida, integral definida, sucesión; sobre todo, el concepto liminar de límite. Este es usado para la definición de cualquier tipo de derivada y para la integral de Riemann, sucesión convergente y suma de una serie y la continuidad. Por su importancia, hay un antes y después de tal concepto que biseca las matemáticas previas, como el Álgebra, la Trigonometría o la Geometría Analítica, del Cálculo. Según Einstein, el mayor aporte que se obtuvo de la derivadas fue la

posibilidad de formular diversos problemas de la física mediante ecuaciones diferenciales .

## **2.CÓDIGO DE LA DERIVADA**

```
# -*- coding: utf-8 -*-  
import numpy as np  
import matplotlib.pyplot as plt  
  
t = np.linspace(0, 4*np.pi, 100)  
y = np.cos(t) # Función original  
kdy = -np.sin(t) # Derivada simbólica  
dy = np.diff(y)/np.diff(t) # Derivada numérica  
  
plt.plot(t, y, label='FO')  
plt.plot(t, kdy, 'o', label='DS')  
plt.plot(t[1:], dy, label='DN')  
  
plt.legend()  
plt.show()
```

### **3.FOTOS DE LA DERIVADA**

```
# -*- coding: utf-8 -*-  
import numpy as np  
import matplotlib.pyplot as plt  
  
t = np.linspace(0, 4*np.pi, 100)  
y = np.cos(t) # Función original  
kdy = -np.sin(t) # Derivada simbólica  
dy = np.diff(y)/np.diff(t) # Derivada numérica  
  
plt.plot(t, y, label='FO')  
plt.plot(t, kdy, 'o', label='DS')  
plt.plot(t[1:], dy, label='DN')  
  
plt.legend()  
plt.show()
```

### **4.RESULTADO DE LA DERIVADA**

