

# Funciones Matemáticas: La Biblioteca `math`

Materia: Algoritmos y Programación

Prof. D.Sc. BARSEKH-ONJI Aboud

Facultad de Ingeniería  
Universidad Anáhuac México

30 de octubre de 2025

# Agenda

1. Introducción a la Biblioteca `math`
2. Funciones Esenciales
3. Tarea Práctica

# ¿Qué es la Biblioteca math?

## ¿Por qué la necesitamos?

Los operadores básicos de Python ('+', '-', '\*', '/', '\*\*') son excelentes para aritmética simple. Pero, ¿qué pasa si necesitamos calcular la raíz cuadrada, el seno de un ángulo o un logaritmo?

## Una Caja de Herramientas Matemáticas

La biblioteca `math` es un módulo incorporado en Python que nos da acceso a un conjunto de funciones y constantes matemáticas avanzadas, estandarizadas y optimizadas.

# ¿Qué es la Biblioteca math?

## Examples

¿Cómo se usa? El comando `import` Para poder usar las funciones de esta biblioteca, **primero debemos importarla** al inicio de nuestro programa.

```
1 import math
```

```
2
```

# Sintaxis: Cómo Llamar a una Función

## Usando la Notación de Punto (')

Una vez que hemos importado la biblioteca, usamos la sintaxis `math.nombre_funcion()` para llamar a la función que necesitamos.

### Ejemplo Correcto

```
1 import math
2
3 raiz = math.sqrt(25)
4 print(raiz) # Imprime 5.0
5
```

### Error Común

```
1 # Incorrecto:
2 # Python no conoce una funcion
3 # llamada 'sqrt' por si sola.
4 raiz = sqrt(25)
5
6 # NameError: name 'sqrt' is not
   defined
7
```

# Agenda

1. Introducción a la Biblioteca `math`
2. Funciones Esenciales
3. Tarea Práctica

# Constantes Fundamentales

## Valores Predefinidos

La biblioteca `math` nos da constantes matemáticas universales para que no tengamos que escribirlas a mano (¡y con mayor precisión!).

### `math.pi`

El número Pi ( $\pi$ ), la relación entre la circunferencia de un círculo y su diámetro.

```
1 import math
2 print(math.pi)
3 # Imprime: 3.141592653589793
4
```

### `math.e`

El número de Euler ( $e$ ), la base de los logaritmos naturales.

```
1 import math
2 print(math.e)
3 # Imprime: 2.718281828459045
4
```

## Ejemplo de Uso: Área de un Círculo

```
1 import math
2 radio = 5
3 area = math.pi * (radio ** 2)
4 print(f"El area es: {area}")
5 # Imprime: El area es: 78.5398...
6
```



# Funciones de Potencia y Logaritmos

## Raíz Cuadrada: `math.sqrt()`

Calcula la raíz cuadrada de un número. Es más rápido y preciso que usar `x ** 0.5`.

```
1 import math
2 # Raíz cuadrada de 81
3 raiz = math.sqrt(81)
4 print(raiz) # Imprime: 9.0
5
```

## Potencia: `math.pow()`

Eleva un número `x` a la potencia `y`. Similar a `x ** y`, pero `pow()` siempre devuelve un flotante.

```
1 import math
2 # 3 elevado a la 4
3 potencia = math.pow(3, 4)
4 print(potencia) # Imprime: 81.0
5
```

# Funciones de Potencia y Logaritmos

## Logaritmos: `math.log ()`

- `math.log (x)`: Logaritmo natural (base  $e$ ).
- `math.log10 (x)`: Logaritmo base 10.
- `math.log (x, base)`: Logaritmo en cualquier base.

## Exponencial: `math.exp ()`

Calcula  $e^x$ .

```
1 import math
2 # e elevado a la 1
3 resultado = math.exp(1)
4 print(resultado) # Imprime:
5                 2.71828...
```

## Redondeo: `floor()`, `ceil()` y `trunc()`

### Controlando los Decimales

Estas funciones nos permiten "limpiar" números flotantes, pero lo hacen de formas muy diferentes.

## Redondeo: floor(), ceil() y trunc()

### `math.floor(x)` (Piso)

Redondea un número **hacia abajo** al entero más cercano.

```
1 import math
2 x = 9.8
3 print(math.floor(x)) # Imprime: 9
4
```

### `math.ceil(x)` (Techo)

Redondea un número **hacia arriba** al entero más cercano.

```
1 import math
2 x = 9.2
3 print(math.ceil(x)) # Imprime: 10
4
```

## Redondeo: floor(), ceil() y trunc()

### `math.trunc(x)` (Truncar)

Simplemente **corta los decimales**, sin redondear. Para números positivos, es igual a `floor()`.

```
1 import math
2 print(math.trunc(9.8)) # Imprime: 9
3 print(math.trunc(-3.7)) # Imprime:
   -3
```

### `round(x)` (Función Nativa)

Redondea al entero más cercano (el .5 lo redondea al par más cercano). **No requiere** `import math`.

```
1 # No se necesita importar math
2 print(round(9.8)) # Imprime: 10
3 print(round(9.2)) # Imprime: 9
4
```

# Trigonometría: `sin()`, `cos()`, `tan()`

## Funciones Trigonómicas

La biblioteca `math` incluye todas las funciones trigonométricas estándar.

- `math.sin(x)`
- `math.cos(x)`
- `math.tan(x)`
- Y también las inversas: `math.asin(x)`, `math.acos(x)`, `math.atan(x)`

# Trigonometría: `sin()`, `cos()`, `tan()`

## ¡ADVERTENCIA! Python usa Radianes

Todas las funciones trigonométricas en Python (y la mayoría de los lenguajes de programación) esperan que los ángulos estén medidos en **RADIANES**, no en Grados. Este es el error más común al usar estas funciones.

```
1 import math
2
3 angulo_rad = math.pi / 2
4 resultado = math.sin(angulo_rad)
5 print(resultado)
6
```

# Conversión de Ángulos: radians() y degrees()

## La Solución Fácil

Para evitar tener que hacer las conversiones a mano (como  $\text{grados} * \text{math.pi} / 180$ ), la biblioteca `math` nos da dos funciones de ayuda.

### `math.radians(grados)`

Convierte un ángulo de grados a radianes.

```
1 import math
2 angulo_grados = 90
3 angulo_radianes = math.radians(
4     angulo_grados)
5 print(angulo_radianes)
6 # Imprime: 1.5707... (que es pi/2)
7
```

### `math.degrees(radianes)`

Convierte un ángulo de radianes a grados.

```
1 import math
2 angulo_radianes = math.pi
3 angulo_grados = math.degrees(
4     angulo_radianes)
5 print(angulo_grados)
6 # Imprime: 180.0
7
```



# Agenda

1. Introducción a la Biblioteca `math`
2. Funciones Esenciales
3. Tarea Práctica

# Tarea: Calculadora de Triángulo Rectángulo

## Objetivo

Escribir un programa que utilice la biblioteca `math` para resolver un triángulo rectángulo. El usuario debe proveer el valor de los dos catetos.

# Tarea: Calculadora de Triángulo Rectángulo

## Instrucciones

Tu programa debe hacer lo siguiente:

1. **Importar** la biblioteca `math`.
2. **Pedir** al usuario la longitud del **Cateto A** (un número).
3. **Pedir** al usuario la longitud del **Cateto B** (un número).
4. **Calcular y mostrar** la longitud de la **Hipotenusa**.
  - Pista: Teorema de Pitágoras ( $c = \sqrt{a^2 + b^2}$ ). Usa `math.sqrt()`.
5. **Calcular y mostrar** los dos ángulos agudos (Ángulo A y Ángulo B) en **GRADOS**.
  - Pista: Usa una función trigonométrica inversa como `math.atan()` (arco tangente).
  - Pista:  $\tan(A) = \text{Opuesto/Adyacente} = a/b$ .
  - Pista: ¡Recuerda convertir el resultado de radianes a grados usando `math.degrees()`!