MyProject

Release 0.0.1

MyAuthor

Feb 01, 2025

CONTENTS:

Add your content using `reStructuredText` syntax. See the reStructuredText documentation for details.

CHAPTER

ONE

SPHINX TUTORIAL UPDATED

1.1 add module

1.1.1 Addition Module

$$a + b$$

This module provides a function to add two numbers.

`add.`**`add`**(*a*, *b*)

Returns the sum of a and b.

1.2 newtest module

`newtest.`**`area_of_ellipse`**(*a*, *b*)

Computes the area of an ellipse:

$$A = \pi a b$$

where: - A is the area, - a is the semi-major axis, - b is the semi-minor axis.

Parameters

- **`a`** – The semi-major axis length.
- **`b`** – The semi-minor axis length.

Returns

The area of the ellipse.

Return type

float

`newtest.`**`fibonacci`**(*n*)

Computes the nth Fibonacci number using Binet's formula:

$$F_n = \frac{1}{\sqrt{5}} \left(\left(\frac{1 + \sqrt{5}}{2} \right)^n - \left(\frac{1 - \sqrt{5}}{2} \right)^n \right)$$

where: - F_n is the nth Fibonacci number.

Parameters

`n` – The index of the Fibonacci sequence.

Returns

The nth Fibonacci number.

Return type

int

newtest.**logistic_growth**(*P0*, *r*, *t*, *K*)

Models the population growth according to the logistic growth model:

$$P(t) = \frac{KP_0}{P_0 + (K - P_0)e^{-rt}}$$

where: - $P(t)$ is the population at time t, - P_0 is the initial population, - r is the growth rate, - K is the carrying capacity of the environment.

Parameters

- **P0** – Initial population size.
- **r** – Growth rate.
- **t** – Time at which to evaluate the population.
- **K** – Carrying capacity.

Returns
The population at time t.

Return type
float

newtest.**logistic_map**(*x0*, *r*, *n*)

Simulates the logistic map:

$$x_{n+1} = rx_n(1 - x_n)$$

where: - x_{n+1} is the population at the next time step, - x_n is the population at the current time step, - r is the growth rate.

This is often used to model chaotic systems.

Parameters

- **x0** – Initial population size.
- **r** – Growth rate parameter.
- **n** – Number of iterations.

Returns
The population after n iterations.

Return type
float

newtest.**pendulum_period**(*length*, *g=9.81*)

Calculates the period of a simple pendulum:

$$T = 2\pi\sqrt{\frac{l}{g}}$$

where: - T is the period, - l is the length of the pendulum, - g is the acceleration due to gravity (default value is 9.81 m/s^2).

Parameters

- **length** – The length of the pendulum.
- **g** – The gravitational acceleration (default is 9.81 m/s^2).

Returns
The period of the pendulum.

Return type
float

newtest.**quadratic_formula**(*a*, *b*, *c*)

Solves the quadratic equation:

$$ax^2 + bx + c = 0$$

Using the quadratic formula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

where: - a, b, and c are the coefficients of the quadratic equation. - x is the solution to the equation.

Parameters

- **a** – Coefficient of x^2.
- **b** – Coefficient of x.
- **c** – Constant term.

Returns
A tuple containing the two solutions for x.

Return type
tuple of floats

newtest.**sum_of_squares**(*n*)

Computes the sum of squares of the first n integers:

$$S = 1^2 + 2^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$$

where: - S is the sum of squares. - n is the number of terms.

Parameters
n – The number of terms to sum.

Returns
The sum of squares.

Return type
float

1.3 pythagorus module

pythagorus.**area_of_circle**(*radius*)

Compute the area of a circle.

The formula is given by:

$$A = \pi r^2$$

where: - A is the area, - r is the radius.

Parameters
radius (*float*) – The radius of the circle.

Returns
The computed area.

Return type
float

pythagorus.**energy**(*mass*, *c*)

Computes energy using Einstein's equation:

$$E = mc^2$$

where: - E is energy, - m is mass, - c is the speed of light.

Parameters

- **mass** – The mass of the object.
- **c** – The speed of light.

Returns
The computed energy.

pythagorus.**force**(*mass*, *acceleration*)

Computes force using Newton's Second Law: .. math:: F = m cdot a

where: - F is the force, - m is the mass, - a is the acceleration.

Parameters

- **mass** – The object's mass.
- **acceleration** – The object's acceleration.

Returns
The computed force.

pythagorus.**kinetic_energy**(*mass*, *velocity*)

Computes kinetic energy using the formula:

$$KE = \frac{1}{2}mv^2$$

Parameters

- **mass** (*float*) – The object's mass (m).
- **velocity** (*float*) – The object's velocity (v).

Returns
The kinetic energy (KE).

Return type
float

pythagorus.**pythagoras**(*a*, *b*)

Computes the hypotenuse using the Pythagorean theorem:

$$c^2 = a^2 + b^2$$
$$(a + b)^2 = a^2 + 2ab + b^2$$

Parameters

- **a** – Side A length.
- **b** – Side B length.

Returns
Hypotenuse length.

pythagorus.**velocity**(*distance*, *time*)

Computes velocity using the equation:

$$v = \frac{d}{t}$$

Parameters

- **distance** (*float*) – Distance traveled (d).
- **time** (*float*) – Time taken (t).

Returns
Computed velocity (v).

Return type
float

1.4 subtract module

1.4.1 Subtraction Module

$$a - b$$

This module provides a function to subtract two numbers.

subtract.**subtract**(*a*, *b*)

Returns the difference of a and b.

PYTHON MODULE INDEX