

## Lucenext Calculator

Generated by Doxygen 1.9.1



<b>1 Namespace Index</b>	<b>1</b>
1.1 Packages	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 File Index</b>	<b>5</b>
3.1 File List	5
<b>4 Namespace Documentation</b>	<b>7</b>
4.1 src Namespace Reference	7
4.2 src.calculator Namespace Reference	7
4.2.1 Function Documentation	7
4.2.1.1 build_safe_ns()	7
4.2.1.2 eval_node()	8
4.2.1.3 evaluate()	8
4.2.1.4 tokenize()	9
4.3 src.gui Namespace Reference	9
4.3.1 Function Documentation	9
4.3.1.1 main()	9
4.3.1.2 resource_path()	10
4.4 src.math_lib Namespace Reference	10
4.4.1 Function Documentation	11
4.4.1.1 _snap_to_integer()	11
4.4.1.2 abs()	11
4.4.1.3 add()	11
4.4.1.4 arctan()	12
4.4.1.5 compute_e()	12
4.4.1.6 cos()	13
4.4.1.7 cotg()	13
4.4.1.8 div()	13
4.4.1.9 fact()	14
4.4.1.10 ln()	14
4.4.1.11 log()	14
4.4.1.12 mul()	15
4.4.1.13 nthroot()	15
4.4.1.14 pi()	16
4.4.1.15 power()	16
4.4.1.16 sin()	16
4.4.1.17 sqrt()	17
4.4.1.18 square()	17
4.4.1.19 sub()	17
4.4.1.20 sum()	18

4.4.1.21 tg()	18
4.5 src.stddev Namespace Reference	19
4.5.1 Function Documentation	19
4.5.1.1 calculate_stddev()	19
4.5.1.2 load_data()	19
4.5.2 Variable Documentation	20
4.5.2.1 data	20
4.5.2.2 stddev	20
4.6 src.test_math_lib Namespace Reference	20
4.6.1 Function Documentation	21
4.6.1.1 test_abs_sum()	21
4.6.1.2 test_add()	21
4.6.1.3 test_arctan_and_pi()	21
4.6.1.4 test_compute_e()	21
4.6.1.5 test_div()	21
4.6.1.6 test_fact()	22
4.6.1.7 test_ln_and_log()	22
4.6.1.8 test_mul()	22
4.6.1.9 test_nthroot()	22
4.6.1.10 test_sin_cos_tg_cotg()	22
4.6.1.11 test_sqrt()	22
4.6.1.12 test_square_and_power()	22
4.6.1.13 test_sub()	22
<b>5 Class Documentation</b>	<b>23</b>
5.1 CalculatorGUI Class Reference	23
5.1.1 Constructor & Destructor Documentation	24
5.1.1.1 __init__()	24
5.1.2 Member Function Documentation	24
5.1.2.1 _get_mapping()	24
5.1.2.2 change_base()	24
5.1.2.3 format_result()	25
5.1.2.4 on_button()	26
5.1.2.5 on_keypress()	26
5.1.2.6 show_help()	27
5.1.3 Member Data Documentation	27
5.1.3.1 after_equal	27
5.1.3.2 base_var	27
5.1.3.3 buttons	27
5.1.3.4 expr_var	27
5.1.3.5 last_result	28
5.1.3.6 master	28

5.1.3.7 result_var . . . . .	28
5.2 Parser Class Reference . . . . .	28
5.2.1 Detailed Description . . . . .	29
5.2.2 Constructor & Destructor Documentation . . . . .	29
5.2.2.1 __init__() . . . . .	29
5.2.3 Member Function Documentation . . . . .	29
5.2.3.1 advance() . . . . .	29
5.2.3.2 current() . . . . .	30
5.2.3.3 parse() . . . . .	30
5.2.3.4 parse_expression() . . . . .	30
5.2.3.5 parse_factor() . . . . .	30
5.2.3.6 parse_power() . . . . .	31
5.2.3.7 parse_term() . . . . .	31
5.2.4 Member Data Documentation . . . . .	31
5.2.4.1 last_ans . . . . .	31
5.2.4.2 pos . . . . .	31
5.2.4.3 tokens . . . . .	31
<b>6 File Documentation</b>	<b>33</b>
6.1 __init__.py File Reference . . . . .	33
6.2 calculator.py File Reference . . . . .	33
6.2.1 Detailed Description . . . . .	34
6.3 gui.py File Reference . . . . .	34
6.3.1 Detailed Description . . . . .	34
6.4 math_lib.py File Reference . . . . .	34
6.4.1 Detailed Description . . . . .	35
6.5 stddev.py File Reference . . . . .	36
6.5.1 Detailed Description . . . . .	36
6.6 test_math_lib.py File Reference . . . . .	36
<b>Index</b>	<b>39</b>



# Chapter 1

## Namespace Index

### 1.1 Packages

Here are the packages with brief descriptions (if available):

<a href="#">src</a>	7
<a href="#">src.calculator</a>	7
<a href="#">src.gui</a>	9
<a href="#">src.math_lib</a>	10
<a href="#">src.stddev</a>	19
<a href="#">src.test_math_lib</a>	20





## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">CalculatorGUI</a> . . . . .	<a href="#">23</a>
<a href="#">Parser</a>	
Class to parse the tokenized input . . . . .	<a href="#">28</a>



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

<a href="#">__init__.py</a>	.....	33
<a href="#">calculator.py</a>	Calculator that evaluates mathematical expressions	33
<a href="#">gui.py</a>	GUI for the calculator	34
<a href="#">math_lib.py</a>	Library for mathematical operations	34
<a href="#">stddev.py</a>	Calculates the standard deviation of a series of numbers provided via standard input (stdin)	36
<a href="#">test_math_lib.py</a>	.....	36



## Chapter 4

# Namespace Documentation

### 4.1 src Namespace Reference

#### Namespaces

- [calculator](#)
- [gui](#)
- [math\\_lib](#)
- [stddev](#)
- [test\\_math\\_lib](#)

### 4.2 src.calculator Namespace Reference

#### Classes

- class [Parser](#)  
*Class to parse the tokenized input.*

#### Functions

- def [tokenize](#) (expr)  
*Function to tokenize the input expression.*
- def [eval\\_node](#) (node, ns)  
*Function to evaluate the AST node.*
- def [build\\_safe\\_ns](#) (last\_ans, base=10)  
*Function to build a safe namespace for the calculator.*
- def [evaluate](#) (expr, base=10)  
*Function to evaluate the expression.*

#### 4.2.1 Function Documentation

##### 4.2.1.1 build\_safe\_ns()

```
def src.calculator.build_safe_ns (  
    last_ans,  
    base = 10 )
```

Function to build a safe namespace for the calculator.

**Parameters**

<i>last_ans</i>	last answer used in the calculator
<i>base</i>	base for number conversion

**Returns**

dictionary of functions and constants

**4.2.1.2 eval\_node()**

```
def src.calculator.eval_node (
    node,
    ns )
```

Function to evaluate the AST node.

**Parameters**

<i>node</i>	AST node
<i>ns</i>	namespace for functions and constants

**Returns**

evaluated value of the node

**4.2.1.3 evaluate()**

```
def src.calculator.evaluate (
    expr,
    base = 10 )
```

Function to evaluate the expression.

**Parameters**

<i>expr</i>	input expression
<i>base</i>	base for number conversion (2, 8, or 10)

**Returns**

evaluated result as a string

#### 4.2.1.4 tokenize()

```
def src.calculator.tokenize (
    expr )
```

Function to tokenize the input expression.

##### Parameters

<i>expr</i>	input expression
-------------	------------------

##### Returns

generator of tokens

## 4.3 src.gui Namespace Reference

### Classes

- class [CalculatorGUI](#)

### Functions

- def [resource\\_path](#) (relative\_path)
- def [main](#) ()  
*Main entry point launching the calculator GUI.*

#### 4.3.1 Function Documentation

##### 4.3.1.1 main()

```
def src.gui.main ( )
```

Main entry point launching the calculator GUI.

Initializes the Tk root window, sets minimum and maximum sizes, and starts the main event loop.

##### Returns

None

#### 4.3.1.2 resource\_path()

```
def src.gui.resource_path (
    relative_path )
```

## 4.4 src.math\_lib Namespace Reference

### Functions

- def [add](#) (a, b)  
*Function to add two numbers.*
- def [sub](#) (a, b)  
*Function to subtract second number from first.*
- def [mul](#) (a, b)  
*Function to multiply two numbers.*
- def [div](#) (a, b)  
*Function to divide first number by second.*
- def [fact](#) (n)  
*Function to calculate factorial of a number.*
- def [compute\\_e](#) (precision=20)  
*Function to compute the value of Euler's number.*
- def [arctan](#) (x, precision=1e-17)  
*Function to calculate arcus tangens of x.*
- def [pi](#) (precision=1e-17)  
*Function to compute the value of pi.*
- def [square](#) (a)  
*Function to calculate the square of a number.*
- def [power](#) (a, b)  
*Function to calculate the power of a number.*
- def [sqrt](#) (a)  
*Function to compute the square root of a number.*
- def [nthroot](#) (a, n)  
*Function to compute the nth root of a number.*
- def [ln](#) (a, precision=1e-20)  
*Function to calculate the natural logarithm of a number.*
- def [log](#) (a, b)  
*Function to calculate the logarithm of a number with base b.*
- def [abs](#) (a)  
*Function to calculate the absolute value of a number.*
- def [\\_snap\\_to\\_integer](#) (val, precision)  
*Function to round result into more readable format.*
- def [sin](#) (x, precision=1e-17)  
*Function to calculate the sine of an angle in degrees.*
- def [cos](#) (x, precision=1e-17)  
*Function to calculate the cosine of an angle in degrees.*
- def [tg](#) (x, precision=1e-10)  
*Function to calculate the tangent of an angle in degrees.*
- def [cotg](#) (x, precision=1e-10)  
*Function to calculate the cotangent of an angle in degrees.*
- def [sum](#) (numbers)  
*Function to calculate the sum of a list of numbers.*



## 4.4.1 Function Documentation

### 4.4.1.1 `_snap_to_integer()`

```
def src.math_lib._snap_to_integer (
    val,
    precision ) [private]
```

Function to round result into more readable format.

#### Parameters

<i>val</i>	number
<i>precision</i>	number of decimal places

#### Returns

rounded value

Used in some goniometric functions

### 4.4.1.2 `abs()`

```
def src.math_lib.abs (
    a )
```

Function to calculate the absolute value of a number.

#### Parameters

<i>a</i>	number
----------	--------

#### Returns

absolute value of a

### 4.4.1.3 `add()`

```
def src.math_lib.add (
    a,
    b )
```

Function to add two numbers.

**Parameters**

<i>a</i>	first number
<i>b</i>	second number

**Returns**

sum

**4.4.1.4 arctan()**

```
def src.math_lib.arctan (
    x,
    precision = 1e-17 )
```

Function to calculate arcus tangens of x.

**Parameters**

<i>x</i>	number
<i>precision</i>	number of decimal places

**Returns**

arctan of x

Computation using the Taylor series expansion.

**4.4.1.5 compute\_e()**

```
def src.math_lib.compute_e (
    precision = 20 )
```

Function to compute the value of Euler's number.

**Parameters**

<i>precision</i>	number of decimal places
------------------	--------------------------

Computation using the Taylor series expansion.

**Returns**

value of euler's number

#### 4.4.1.6 cos()

```
def src.math_lib.cos (
    x,
    precision = 1e-17 )
```

Function to calculate the cosine of an angle in degrees.

##### Parameters

<i>x</i>	angle in degrees
<i>precision</i>	number of decimal places

##### Returns

cosine of x

Computation using the Taylor series expansion.

#### 4.4.1.7 cotg()

```
def src.math_lib.cotg (
    x,
    precision = 1e-10 )
```

Function to calculate the cotangent of an angle in degrees.

##### Parameters

<i>x</i>	angle in degrees
<i>precision</i>	number of decimal places

##### Returns

cotangent of x

Computation using the sine and cosine functions.

#### 4.4.1.8 div()

```
def src.math_lib.div (
    a,
    b )
```

Function to divide first number by second.

##### Parameters

<i>a</i>	first number
<i>b</i>	second number

**Returns**

quotient

If b is 0, returns "Error".

**4.4.1.9 fact()**

```
def src.math_lib.fact (
    n )
```

Function to calculate factorial of a number.

**Parameters**

<i>n</i>	number
----------	--------

**Returns**

factorial of n

If n is negative, returns "Error".

**4.4.1.10 ln()**

```
def src.math_lib.ln (
    a,
    precision = 1e-20 )
```

Function to calculate the natural logarithm of a number.

**Parameters**

<i>a</i>	number
<i>precision</i>	number of decimal places

**Returns**

natural logarithm of a

Computation using the Taylor series expansion.

If a is less than or equal to 0, returns "Error".

**4.4.1.11 log()**

```
def src.math_lib.log (
    a,
    b )
```

Function to calculate the logarithm of a number with base b.

**Parameters**

<i>a</i>	number
<i>b</i>	base

**Returns**

logarithm of a with base b

**4.4.1.12 mul()**

```
def src.math_lib.mul (  
    a,  
    b )
```

Function to multiply two numbers.

**Parameters**

<i>a</i>	first number
<i>b</i>	second number

**Returns**

product

**4.4.1.13 nthroot()**

```
def nthroot (  
    a,  
    n )
```

Function to compute the nth root of a number.

**Parameters**

<i>a</i>	number
<i>n</i>	root

**Returns**

nth root of a

If a is negative and n is even, returns "Error".

**4.4.1.14 pi()**

```
def src.math_lib.pi (
    precision = 1e-17 )
```

Function to compute the value of pi.

**Parameters**

<i>precision</i>	number of decimal places
------------------	--------------------------

**Returns**

value of pi

Computation using the Machin-like formula.

**4.4.1.15 power()**

```
def src.math_lib.power (
    a,
    b )
```

Function to calculate the power of a number.

**Parameters**

<i>a</i>	base
<i>b</i>	exponent

**Returns**

a raised to the power of b

**4.4.1.16 sin()**

```
def src.math_lib.sin (
    x,
    precision = 1e-17 )
```

Function to calculate the sine of an angle in degrees.

**Parameters**

<i>x</i>	angle in degrees
<i>precision</i>	number of decimal places

**Returns**

sine of x

Computation using the Taylor series expansion.

**4.4.1.17 sqrt()**

```
def src.math_lib.sqrt (  
    a )
```

Function to compute the square root of a number.

**Parameters**

<i>a</i>	number
----------	--------

**Returns**

square root of a

If a is negative, returns "Error".

**4.4.1.18 square()**

```
def src.math_lib.square (  
    a )
```

Function to calculate the square of a number.

**Parameters**

<i>a</i>	number
----------	--------

**Returns**

square of a

**4.4.1.19 sub()**

```
def src.math_lib.sub (  
    a,  
    b )
```

Function to subtract second number from first.

**Parameters**

<i>a</i>	first number
<i>b</i>	second number

**Returns**

difference

**4.4.1.20 sum()**

```
def src.math_lib.sum (
    numbers )
```

Function to calculate the sum of a list of numbers.

**Parameters**

<i>numbers</i>	list of numbers
----------------	-----------------

**Returns**

sum of numbers

**4.4.1.21 tg()**

```
def src.math_lib.tg (
    x,
    precision = 1e-10 )
```

Function to calculate the tangent of an angle in degrees.

**Parameters**

<i>x</i>	angle in degrees
<i>precision</i>	number of decimal places

**Returns**

tangent of x

Computation using the sine and cosine functions.



## 4.5 src.stddev Namespace Reference

### Functions

- def `load_data` ()  
*Function to load data from standard input.*
- def `calculate_stddev` (data)  
*Function to calculate the standard deviation of a list of numbers.*

### Variables

- def `data` = `load_data`()  
*list for storing input*
- def `stddev` = `calculate_stddev`(data)

### 4.5.1 Function Documentation

#### 4.5.1.1 `calculate_stddev()`

```
def src.stddev.calculate_stddev (  
    data )
```

Function to calculate the standard deviation of a list of numbers.

##### Parameters

<i>data</i>	list of numbers
-------------	-----------------

##### Returns

standard deviation of data

#### 4.5.1.2 `load_data()`

```
def src.stddev.load_data ( )
```

Function to load data from standard input.

This function reads lines from standard input, splits them into individual numbers, converts them to floats, and stores them in a list. It handles invalid input. Stops reading when EOF is reached.

##### Returns

list of floats

## 4.5.2 Variable Documentation

### 4.5.2.1 data

```
data = load_data()
```

list for storing input

### 4.5.2.2 stddev

```
def stddev = calculate_stddev(data)
```

## 4.6 src.test\_math\_lib Namespace Reference

### Functions

- def [test\\_add](#) ()  
*Test addition functionality from [math\\_lib](#).*
- def [test\\_sub](#) ()  
*Test subtraction functionality.*
- def [test\\_mul](#) ()  
*Test multiplication functionality.*
- def [test\\_div](#) ()  
*Test division functionality.*
- def [test\\_fact](#) ()  
*Test factorial computation.*
- def [test\\_compute\\_e](#) ()  
*Test Euler's number approximation.*
- def [test\\_arctan\\_and\\_pi](#) ()  
*Test arctangent and pi approximation.*
- def [test\\_square\\_and\\_power](#) ()  
*Test squaring and exponentiation.*
- def [test\\_sqrt](#) ()  
*Test square root functionality.*
- def [test\\_nthroot](#) ()  
*Test n-th root computation.*
- def [test\\_ln\\_and\\_log](#) ()  
*Test natural logarithm and logarithm with custom base.*
- def [test\\_abs\\_sum](#) ()  
*Test absolute value and list summation.*
- def [test\\_sin\\_cos\\_tg\\_cotg](#) ()  
*Test trigonometric functions: sin, cos, tan, cotangent.*

## 4.6.1 Function Documentation

### 4.6.1.1 test\_abs\_sum()

```
def src.test_math_lib.test_abs_sum ( )
```

Test absolute value and list summation.

Tests abs for positive/negative numbers and sum over lists, including empty list

### 4.6.1.2 test\_add()

```
def src.test_math_lib.test_add ( )
```

Test addition functionality from [math\\_lib](#).

Verifies correct results for positive, negative, and floating-point additions

### 4.6.1.3 test\_arctan\_and\_pi()

```
def src.test_math_lib.test_arctan_and_pi ( )
```

Test arctangent and pi approximation.

Validates arctan result and compares custom pi approximation with math.pi

### 4.6.1.4 test\_compute\_e()

```
def src.test_math_lib.test_compute_e ( )
```

Test Euler's number approximation.

Compares custom approximation with Python's math.e constant

### 4.6.1.5 test\_div()

```
def src.test_math_lib.test_div ( )
```

Test division functionality.

Tests normal division, floating-point division, and division by zero

#### 4.6.1.6 test\_fact()

```
def src.test_math_lib.test_fact ( )
```

Test factorial computation.

Checks correct calculation for 0, positive integers and handling of negative inputs

#### 4.6.1.7 test\_ln\_and\_log()

```
def src.test_math_lib.test_ln_and_log ( )
```

Test natural logarithm and logarithm with custom base.

Includes valid and invalid input handling for ln and log

#### 4.6.1.8 test\_mul()

```
def src.test_math_lib.test_mul ( )
```

Test multiplication functionality.

Includes tests with integers, negative numbers and floating-point values

#### 4.6.1.9 test\_nthroot()

```
def src.test_math_lib.test_nthroot ( )
```

Test n-th root computation.

Tests for positive/negative inputs, even/odd roots, and invalid inputs

#### 4.6.1.10 test\_sin\_cos\_tg\_cotg()

```
def src.test_math_lib.test_sin_cos_tg_cotg ( )
```

Test trigonometric functions: sin, cos, tan, cotangent.

Validates results for known angles and handles undefined cases

#### 4.6.1.11 test\_sqrt()

```
def src.test_math_lib.test_sqrt ( )
```

Test square root functionality.

Checks for perfect square, irrational square root and negative number handling

#### 4.6.1.12 test\_square\_and\_power()

```
def src.test_math_lib.test_square_and_power ( )
```

Test squaring and exponentiation.

Verifies squaring a number and raising to various powers

#### 4.6.1.13 test\_sub()

```
def src.test_math_lib.test_sub ( )
```

Test subtraction functionality.

Validates subtraction for various positive and negative values including floats

## Chapter 5

# Class Documentation

### 5.1 CalculatorGUI Class Reference

#### Public Member Functions

- def `__init__` (self, `master`)  
*Constructor for the `CalculatorGUI` class.*
- def `show_help` (self)  
*Opens a help window with usage instructions.*
- def `on_keypress` (self, event)  
*Handles keyboard input mapping.*
- def `on_button` (self, char)  
*Processes button presses from UI or keyboard.*
- def `format_result` (self, res)  
*Formats numeric results according to selected base and type.*
- def `change_base` (self)  
*Enables/disables calculator buttons based on current base.*

#### Public Attributes

- `master`
- `after_equal`
- `expr_var`
- `result_var`
- `last_result`
- `base_var`
- `buttons`

#### Private Member Functions

- def `_get_mapping` (self)  
*Provides mapping from display symbols to evaluation tokens.*

## 5.1.1 Constructor & Destructor Documentation

### 5.1.1.1 `__init__()`

```
def __init__ (
    self,
    master )
```

Constructor for the [CalculatorGUI](#) class.

#### Parameters

<i>master</i>	The Tkinter root window.
---------------	--------------------------

Initializes window properties, state variables, layouts, buttons, display labels, and keyboard bindings.

## 5.1.2 Member Function Documentation

### 5.1.2.1 `_get_mapping()`

```
def _get_mapping (
    self ) [private]
```

Provides mapping from display symbols to evaluation tokens.

Returns a dictionary mapping button labels (e.g., '×','sin') to the corresponding expression strings for the evaluator.

#### Returns

A dict mapping UI labels to evaluator tokens.

### 5.1.2.2 `change_base()`

```
def change_base (
    self )
```

Enables/disables calculator buttons based on current base.

Activates only valid digit buttons for binary/octal, always enables operators and control keys, and resets display.

#### Returns

None

### 5.1.2.3 `format_result()`

```
def format_result (
    self,
    res )
```

Formats numeric results according to selected base and type.

**Parameters**

<i>res</i>	The result to format (int or float).
------------	--------------------------------------

Converts ints to binary/octal/decimal strings, floats to fixed precision (10 decimals, trimmed).

**Returns**

A string representation of the result.

**5.1.2.4 on\_button()**

```
def on_button (
    self,
    char )
```

Processes button presses from UI or keyboard.

**Parameters**

<i>char</i>	The label of the button pressed.
-------------	----------------------------------

Handles insertion of characters/functions, evaluation on '=', clear/backspace, ANS insertion, and state resets.

**Returns**

None

**5.1.2.5 on\_keypress()**

```
def on_keypress (
    self,
    event )
```

Handles keyboard input mapping.

**Parameters**

<i>event</i>	The Tkinter keyboard event.
--------------	-----------------------------

Maps keys (digits, operators, Enter, Backspace, Delete) to corresponding button actions, validating against current base.



**Returns**

None

**5.1.2.6 show\_help()**

```
def show_help (
    self )
```

Opens a help window with usage instructions.

Creates a scrollable Toplevel window listing all operations, parameters, and notes.

**Returns**

None

**5.1.3 Member Data Documentation****5.1.3.1 after\_equal**

```
after_equal
```

**5.1.3.2 base\_var**

```
base_var
```

**5.1.3.3 buttons**

```
buttons
```

**5.1.3.4 expr\_var**

```
expr_var
```

#### 5.1.3.5 last\_result

last\_result

#### 5.1.3.6 master

master

#### 5.1.3.7 result\_var

result\_var

The documentation for this class was generated from the following file:

- [gui.py](#)

## 5.2 Parser Class Reference

Class to parse the tokenized input.

### Public Member Functions

- def `__init__` (self, [tokens](#), [last\\_ans](#)=0)  
*Constructor for the [Parser](#) class.*
- def [current](#) (self)  
*Function to get the current token.*
- def [advance](#) (self)  
*Function to advance to the next token.*
- def [parse](#) (self)  
*Function to parse the expression.*
- def [parse\\_expression](#) (self)  
*Function to parse the expression.*
- def [parse\\_term](#) (self)  
*Function to parse the term.*
- def [parse\\_power](#) (self)  
*Function to parse power.*
- def [parse\\_factor](#) (self)  
*Function to parse the factor.*

## Public Attributes

- [tokens](#)
- [pos](#)
- [last\\_ans](#)

### 5.2.1 Detailed Description

Class to parse the tokenized input.

### 5.2.2 Constructor & Destructor Documentation

#### 5.2.2.1 `__init__()`

```
def __init__ (
    self,
    tokens,
    last_ans = 0 )
```

Constructor for the [Parser](#) class.

Parameters

<i>tokens</i>	list of tokens
<i>last_ans</i>	last answer used in the calculator

### 5.2.3 Member Function Documentation

#### 5.2.3.1 `advance()`

```
def advance (
    self )
```

Function to advance to the next token.

Increments the position of the current token.

#### 5.2.3.2 `current()`

```
def current (
    self )
```

Function to get the current token.

##### Returns

current token

#### 5.2.3.3 `parse()`

```
def parse (
    self )
```

Function to parse the expression.

Parses the entire expression and returns the abstract syntax tree (AST).

##### Returns

AST node representing the expression

#### 5.2.3.4 `parse_expression()`

```
def parse_expression (
    self )
```

Function to parse the expression.

Handles addition and subtraction.

##### Returns

AST node representing the expression

#### 5.2.3.5 `parse_factor()`

```
def parse_factor (
    self )
```

Function to parse the factor.

Handles parentheses, numbers, identifiers, and unary minus.

##### Returns

AST node representing the factor

### 5.2.3.6 `parse_power()`

```
def parse_power (
    self )
```

Function to parse power.

Handles exponentiation.

#### Returns

AST node representing power

### 5.2.3.7 `parse_term()`

```
def parse_term (
    self )
```

Function to parse the term.

Handles multiplication and division.

#### Returns

AST node representing the term

## 5.2.4 Member Data Documentation

### 5.2.4.1 `last_ans`

```
last_ans
```

### 5.2.4.2 `pos`

```
pos
```

### 5.2.4.3 `tokens`

```
tokens
```

The documentation for this class was generated from the following file:

- [calculator.py](#)



## Chapter 6

# File Documentation

### 6.1 `__init__.py` File Reference

#### Namespaces

- [src](#)

### 6.2 `calculator.py` File Reference

Calculator that evaluates mathematical expressions.

#### Classes

- class [Parser](#)  
*Class to parse the tokenized input.*

#### Namespaces

- [src.calculator](#)

#### Functions

- def [tokenize](#) (expr)  
*Function to tokenize the input expression.*
- def [eval\\_node](#) (node, ns)  
*Function to evaluate the AST node.*
- def [build\\_safe\\_ns](#) (last\_ans, base=10)  
*Function to build a safe namespace for the calculator.*
- def [evaluate](#) (expr, base=10)  
*Function to evaluate the expression.*

### 6.2.1 Detailed Description

Calculator that evaluates mathematical expressions.

Date

2025-04-29

## 6.3 gui.py File Reference

GUI for the calculator.

### Classes

- class [CalculatorGUI](#)

### Namespaces

- [src.gui](#)

### Functions

- def [resource\\_path](#) (relative\_path)
- def [main](#) ()  
*Main entry point launching the calculator GUI.*

### 6.3.1 Detailed Description

GUI for the calculator.

This script creates a graphical user interface for a calculator using the Tkinter library.

Date

2025-04-29

## 6.4 math\_lib.py File Reference

Library for mathematical operations.

### Namespaces

- [src.math\\_lib](#)



## Functions

- def `add` (a, b)  
*Function to add two numbers.*
- def `sub` (a, b)  
*Function to subtract second number from first.*
- def `mul` (a, b)  
*Function to multiply two numbers.*
- def `div` (a, b)  
*Function to divide first number by second.*
- def `fact` (n)  
*Function to calculate factorial of a number.*
- def `compute_e` (precision=20)  
*Function to compute the value of Euler's number.*
- def `arctan` (x, precision=1e-17)  
*Function to calculate arcus tangens of x.*
- def `pi` (precision=1e-17)  
*Function to compute the value of pi.*
- def `square` (a)  
*Function to calculate the square of a number.*
- def `power` (a, b)  
*Function to calculate the power of a number.*
- def `sqrt` (a)  
*Function to compute the square root of a number.*
- def `nthroot` (a, n)  
*Function to compute the nth root of a number.*
- def `ln` (a, precision=1e-20)  
*Function to calculate the natural logarithm of a number.*
- def `log` (a, b)  
*Function to calculate the logarithm of a number with base b.*
- def `abs` (a)  
*Function to calculate the absolute value of a number.*
- def `_snap_to_integer` (val, precision)  
*Function to round result into more readable format.*
- def `sin` (x, precision=1e-17)  
*Function to calculate the sine of an angle in degrees.*
- def `cos` (x, precision=1e-17)  
*Function to calculate the cosine of an angle in degrees.*
- def `tg` (x, precision=1e-10)  
*Function to calculate the tangent of an angle in degrees.*
- def `cotg` (x, precision=1e-10)  
*Function to calculate the cotangent of an angle in degrees.*
- def `sum` (numbers)  
*Function to calculate the sum of a list of numbers.*

### 6.4.1 Detailed Description

Library for mathematical operations.

Date

2025-04-29

## 6.5 stddev.py File Reference

Calculates the standard deviation of a series of numbers provided via standard input (stdin).

### Namespaces

- [src.stddev](#)

### Functions

- def [load\\_data](#) ()  
*Function to load data from standard input.*
- def [calculate\\_stddev](#) (data)  
*Function to calculate the standard deviation of a list of numbers.*

### Variables

- def [data](#) = load\_data()  
*list for storing input*
- def [stddev](#) = calculate\_stddev(data)

### 6.5.1 Detailed Description

Calculates the standard deviation of a series of numbers provided via standard input (stdin).

It processes the input, calculates the mean, sum of squares, and the standard deviation. The result is printed to stdout. The script expects input in the form of whitespace-separated numbers (multiple lines allowed).

Date

2025-04-28

## 6.6 test\_math\_lib.py File Reference

### Namespaces

- [src.test\\_math\\_lib](#)

## Functions

- def `test_add ()`  
*Test addition functionality from `math_lib`.*
- def `test_sub ()`  
*Test subtraction functionality.*
- def `test_mul ()`  
*Test multiplication functionality.*
- def `test_div ()`  
*Test division functionality.*
- def `test_fact ()`  
*Test factorial computation.*
- def `test_compute_e ()`  
*Test Euler's number approximation.*
- def `test_arctan_and_pi ()`  
*Test arctangent and pi approximation.*
- def `test_square_and_power ()`  
*Test squaring and exponentiation.*
- def `test_sqrt ()`  
*Test square root functionality.*
- def `test_nthroot ()`  
*Test n-th root computation.*
- def `test_ln_and_log ()`  
*Test natural logarithm and logarithm with custom base.*
- def `test_abs_sum ()`  
*Test absolute value and list summation.*
- def `test_sin_cos_tg_cotg ()`  
*Test trigonometric functions: sin, cos, tan, cotangent.*



# Index

- `__init__`
    - CalculatorGUI, [24](#)
    - Parser, [29](#)
  - `__init__.py`, [33](#)
  - `_get_mapping`
    - CalculatorGUI, [24](#)
  - `_snap_to_integer`
    - src.math\_lib, [11](#)
- abs
  - src.math\_lib, [11](#)
- add
  - src.math\_lib, [11](#)
- advance
  - Parser, [29](#)
- after\_equal
  - CalculatorGUI, [27](#)
- arctan
  - src.math\_lib, [12](#)
- base\_var
  - CalculatorGUI, [27](#)
- build\_safe\_ns
  - src.calculator, [7](#)
- buttons
  - CalculatorGUI, [27](#)
- calculate\_stddev
  - src.stddev, [19](#)
- calculator.py, [33](#)
- CalculatorGUI, [23](#)
  - `__init__`, [24](#)
  - `_get_mapping`, [24](#)
  - `after_equal`, [27](#)
  - `base_var`, [27](#)
  - `buttons`, [27](#)
  - `change_base`, [24](#)
  - `expr_var`, [27](#)
  - `format_result`, [24](#)
  - `last_result`, [27](#)
  - `master`, [28](#)
  - `on_button`, [26](#)
  - `on_keypress`, [26](#)
  - `result_var`, [28](#)
  - `show_help`, [27](#)
- change\_base
  - CalculatorGUI, [24](#)
- compute\_e
  - src.math\_lib, [12](#)
- cos
  - src.math\_lib, [12](#)
- cotg
  - src.math\_lib, [13](#)
- current
  - Parser, [29](#)
- data
  - src.stddev, [20](#)
- div
  - src.math\_lib, [13](#)
- eval\_node
  - src.calculator, [8](#)
- evaluate
  - src.calculator, [8](#)
- expr\_var
  - CalculatorGUI, [27](#)
- fact
  - src.math\_lib, [14](#)
- format\_result
  - CalculatorGUI, [24](#)
- gui.py, [34](#)
- last\_ans
  - Parser, [31](#)
- last\_result
  - CalculatorGUI, [27](#)
- ln
  - src.math\_lib, [14](#)
- load\_data
  - src.stddev, [19](#)
- log
  - src.math\_lib, [14](#)
- main
  - src.gui, [9](#)
- master
  - CalculatorGUI, [28](#)
- math\_lib.py, [34](#)
- mul
  - src.math\_lib, [15](#)
- nthroot
  - src.math\_lib, [15](#)
- on\_button
  - CalculatorGUI, [26](#)
- on\_keypress
  - CalculatorGUI, [26](#)

- parse
  - Parser, [30](#)
- parse\_expression
  - Parser, [30](#)
- parse\_factor
  - Parser, [30](#)
- parse\_power
  - Parser, [30](#)
- parse\_term
  - Parser, [31](#)
- Parser, [28](#)
  - \_\_init\_\_, [29](#)
  - advance, [29](#)
  - current, [29](#)
  - last\_ans, [31](#)
  - parse, [30](#)
  - parse\_expression, [30](#)
  - parse\_factor, [30](#)
  - parse\_power, [30](#)
  - parse\_term, [31](#)
  - pos, [31](#)
  - tokens, [31](#)
- pi
  - src.math\_lib, [15](#)
- pos
  - Parser, [31](#)
- power
  - src.math\_lib, [16](#)
- resource\_path
  - src.gui, [9](#)
- result\_var
  - CalculatorGUI, [28](#)
- show\_help
  - CalculatorGUI, [27](#)
- sin
  - src.math\_lib, [16](#)
- sqrt
  - src.math\_lib, [17](#)
- square
  - src.math\_lib, [17](#)
- src, [7](#)
  - src.calculator, [7](#)
    - build\_safe\_ns, [7](#)
    - eval\_node, [8](#)
    - evaluate, [8](#)
    - tokenize, [8](#)
  - src.gui, [9](#)
    - main, [9](#)
    - resource\_path, [9](#)
  - src.math\_lib, [10](#)
    - \_snap\_to\_integer, [11](#)
    - abs, [11](#)
    - add, [11](#)
    - arctan, [12](#)
    - compute\_e, [12](#)
    - cos, [12](#)
    - cotg, [13](#)
    - div, [13](#)
    - fact, [14](#)
    - ln, [14](#)
    - log, [14](#)
    - mul, [15](#)
    - nthroot, [15](#)
    - pi, [15](#)
    - power, [16](#)
    - sin, [16](#)
    - sqrt, [17](#)
    - square, [17](#)
    - sub, [17](#)
    - sum, [18](#)
    - tg, [18](#)
  - src.stddev, [19](#)
    - calculate\_stddev, [19](#)
    - data, [20](#)
    - load\_data, [19](#)
    - stddev, [20](#)
  - src.test\_math\_lib, [20](#)
    - test\_abs\_sum, [21](#)
    - test\_add, [21](#)
    - test\_arctan\_and\_pi, [21](#)
    - test\_compute\_e, [21](#)
    - test\_div, [21](#)
    - test\_fact, [21](#)
    - test\_ln\_and\_log, [22](#)
    - test\_mul, [22](#)
    - test\_nthroot, [22](#)
    - test\_sin\_cos\_tg\_cotg, [22](#)
    - test\_sqrt, [22](#)
    - test\_square\_and\_power, [22](#)
    - test\_sub, [22](#)
  - stddev
    - src.stddev, [20](#)
  - stddev.py, [36](#)
  - sub
    - src.math\_lib, [17](#)
  - sum
    - src.math\_lib, [18](#)
  - test\_abs\_sum
    - src.test\_math\_lib, [21](#)
  - test\_add
    - src.test\_math\_lib, [21](#)
  - test\_arctan\_and\_pi
    - src.test\_math\_lib, [21](#)
  - test\_compute\_e
    - src.test\_math\_lib, [21](#)
  - test\_div
    - src.test\_math\_lib, [21](#)
  - test\_fact
    - src.test\_math\_lib, [21](#)
  - test\_ln\_and\_log
    - src.test\_math\_lib, [22](#)
  - test\_math\_lib.py, [36](#)
  - test\_mul
    - src.test\_math\_lib, [22](#)
  - test\_nthroot

- [src.test\\_math\\_lib, 22](#)
- [test\\_sin\\_cos\\_tg\\_cotg](#)
  - [src.test\\_math\\_lib, 22](#)
- [test\\_sqrt](#)
  - [src.test\\_math\\_lib, 22](#)
- [test\\_square\\_and\\_power](#)
  - [src.test\\_math\\_lib, 22](#)
- [test\\_sub](#)
  - [src.test\\_math\\_lib, 22](#)
- [tg](#)
  - [src.math\\_lib, 18](#)
- [tokenize](#)
  - [src.calculator, 8](#)
- [tokens](#)
  - [Parser, 31](#)