

# Calculus



Matej Koreň  
Version 1.0  
Wed Apr 28 2021

# Class Index

## Class List

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

<b>calculator.CalculatorWindow (Class of main window )</b>	<b>4</b>
<b>calculator.History (Class of history )</b>	<b>5</b>
<b>calculator.ScrollLabel (Class of scrollable labels )</b>	<b>6</b>
<b>Math_lib_tests.TestAddition (Tests for addition function )</b>	<b>7</b>
<b>calc_tests.TestBasics (Tests for basic calculations with one kind of operators )</b>	<b>8</b>
<b>Math_lib_tests.TestConstants (Tests for constants in functions )</b>	<b>9</b>
<b>Math_lib_tests.TestCosine (Tests for cosine function )</b>	<b>10</b>
<b>Math_lib_tests.TestCotangent (Tests for cotangent function )</b>	<b>11</b>
<b>Math_lib_tests.TestDivision (Tests for division function )</b>	<b>12</b>
<b>calc_tests.TestErrors (Tests for error handling )</b>	<b>13</b>
<b>Math_lib_tests.TestFactorial (Tests for factorial function )</b>	<b>14</b>
<b>calc_tests.TestGoniometric (Tests for calculations of goniometric functions )</b>	<b>15</b>
<b>calc_tests.TestMultipleOperators (Test for calculations with multiple different operators )</b>	<b>16</b>
<b>Math_lib_tests.TestMultiplication (Tests for multiplication function )</b>	<b>17</b>
<b>calc_tests.TestParentheses (Tests for calculations with parentheses )</b>	<b>18</b>
<b>Math_lib_tests.TestPower (Tests for power function )</b>	<b>19</b>
<b>Math_lib_tests.TestRoot (Tests for root function )</b>	<b>20</b>
<b>Math_lib_tests.TestSine (Tests for sine function )</b>	<b>21</b>
<b>Math_lib_tests.TestSubtraction (Tests for subtraction function )</b>	<b>22</b>
<b>Math_lib_tests.TestTangent (Tests for tangent function )</b>	<b>23</b>

# File Index

## File List

Here is a list of all documented files with brief descriptions:

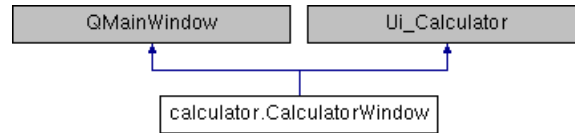
<b>calc.py (Processing script for computing the result of input )</b>	24
<b>calc_tests.py (Tests for calc.py )</b>	28
<b>calculator.py (Realization of graphical user interface )</b>	29
<b>Math_lib.py (Implementation of mathematical functions and constants )</b>	32
<b>Math_lib_tests.py (Tests for Math_lib.py )</b>	37

# Class Documentation

## calculator.CalculatorWindow Class Reference

Class of main window.

Inheritance diagram for calculator.CalculatorWindow:



### Static Public Attributes

- string **expression** = "0"  
*Variable containing current expression from input.*

---

### Detailed Description

Class of main window.

---

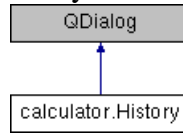
The documentation for this class was generated from the following file:

- **calculator.py**

## calculator.History Class Reference

Class of history.

Inheritance diagram for calculator.History:



### Public Member Functions

- `def __init__ (self, parent=None)`  
*Initialization of history window.*
- `def UiComponents (self)`  
*Adding widgets to history window.*
- `def clear_history (self)`  
*Clearing history of calculator.*

### Public Attributes

- `label`  
*label*
- `clear_button`  
*clear\_button*

---

### Detailed Description

Class of history.

---

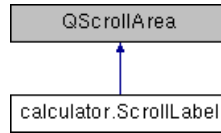
The documentation for this class was generated from the following file:

- `calculator.py`

## calculator.ScrollLabel Class Reference

Class of scrollable labels.

Inheritance diagram for calculator.ScrollLabel:



### Public Member Functions

- `def __init__ (self, *args, **kwargs)`  
*Initialization of scrollable part of history.*
- `def setText (self, text)`  
*Adding text to scrollable label.*

### Public Attributes

- `label`  
*label*

---

### Detailed Description

Class of scrollable labels.

---

### Member Function Documentation

**def calculator.ScrollLabel.setText ( *self*, *text*)**

Adding text to scrollable label.

#### Parameters

<i>text</i>	Text to be added
-------------	------------------

---

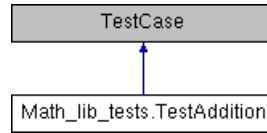
The documentation for this class was generated from the following file:

- `calculator.py`

## Math\_lib\_tests.TestAddition Class Reference

Tests for addition function.

Inheritance diagram for Math\_lib\_tests.TestAddition:



### Public Member Functions

- **def testInteger** (self)  
*Tests for addition of integers.*
- **def testFloat** (self)  
*Tests for addition of floats.*

---

### Detailed Description

Tests for addition function.

---

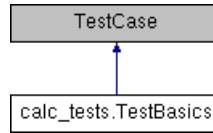
The documentation for this class was generated from the following file:

- **Math\_lib\_tests.py**

## calc\_tests.TestBasics Class Reference

Tests for basic calculations with one kind of operators.

Inheritance diagram for calc\_tests.TestBasics:



### Public Member Functions

- **def testOneNumber (self)**  
*Test for the return of a single number.*
- **def testAdd (self)**  
*Test for the addition.*
- **def testSub (self)**  
*Test for the subtraction.*
- **def testMultiplication (self)**  
*Test for the multiplication.*
- **def testDivision (self)**  
*Test for the division.*
- **def testPower (self)**  
*Test for the power of a given number.*
- **def testRoot (self)**  
*Test for the square root of a given number.*
- **def testFactorial (self)**  
*Test for the factorial of a given number.*

---

### Detailed Description

Tests for basic calculations with one kind of operators.

---

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

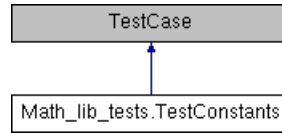
- **calc\_tests.py**



## Math\_lib\_tests.TestConstants Class Reference

Tests for constants in functions.

Inheritance diagram for Math\_lib\_tests.TestConstants:



### Public Member Functions

- `def testPi (self)`  
*Tests for the constant "pi".*
- `def testE (self)`  
*Tests for the constant "e".*

---

### Detailed Description

Tests for constants in functions.

---

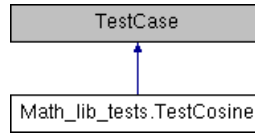
The documentation for this class was generated from the following file:

- `Math_lib_tests.py`

## Math\_lib\_tests.TestCosine Class Reference

Tests for cosine function.

Inheritance diagram for Math\_lib\_tests.TestCosine:



### Public Member Functions

- `def testInteger (self)`  
*Tests for cosine function of an integer.*
- `def testFloat (self)`  
*Tests for cosine function of a float.*

---

### Detailed Description

Tests for cosine function.

---

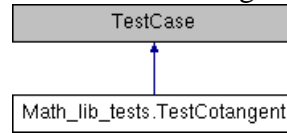
The documentation for this class was generated from the following file:

- `Math_lib_tests.py`

## Math\_lib\_tests.TestCotangent Class Reference

Tests for cotangent function.

Inheritance diagram for Math\_lib\_tests.TestCotangent:



### Public Member Functions

- `def testNotDefined (self)`  
*Tests for exceptions with the cotangent of its undefined values.*
- `def testInteger (self)`  
*Tests for cotangent of an integer.*
- `def testFloat (self)`  
*Tests for cotangent of a float.*

---

### Detailed Description

Tests for cotangent function.

---

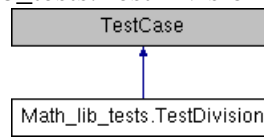
The documentation for this class was generated from the following file:

- `Math_lib_tests.py`

## Math\_lib\_tests.TestDivision Class Reference

Tests for division function.

Inheritance diagram for Math\_lib\_tests.TestDivision:



### Public Member Functions

- **def testDivByZero (self)**  
*Tests for division by zero.*
- **def testDivByOne (self)**  
*Tests for division by one.*
- **def testDivBySameNum (self)**  
*Tests for division by same number.*
- **def testFloat (self)**  
*Tests for division by same float.*

---

### Detailed Description

Tests for division function.

---

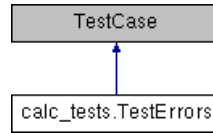
The documentation for this class was generated from the following file:

- **Math\_lib\_tests.py**

## calc\_tests.TestErrors Class Reference

Tests for error handling.

Inheritance diagram for calc\_tests.TestErrors:



### Public Member Functions

- `def testErrors (self)`  
*Tests for Math errors.*

---

### Detailed Description

Tests for error handling.

---

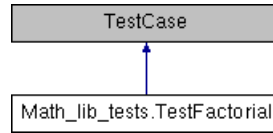
The documentation for this class was generated from the following file:

- `calc_tests.py`

## Math\_lib\_tests.TestFactorial Class Reference

Tests for factorial function.

Inheritance diagram for Math\_lib\_tests.TestFactorial:



### Public Member Functions

- **def testNotNaturalNumber** (self)  
*Tests for exceptions with the factorial of not a natural number.*
- **def testFactOfOne** (self)  
*Test for the factorial of one.*
- **def testInteger** (self)  
*Test for the factorial of an integer.*

---

### Detailed Description

Tests for factorial function.

---

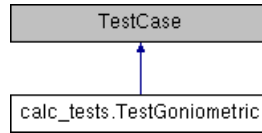
The documentation for this class was generated from the following file:

- **Math\_lib\_tests.py**

## calc\_tests.TestGoniometric Class Reference

Tests for calculations of goniometric functions.

Inheritance diagram for calc\_tests.TestGoniometric:



### Public Member Functions

- **def testSine (self)**  
*Test for the sine value of a given number (in degrees)*
- **def testCosine (self)**  
*Test for the cosine value of a given number (in degrees)*
- **def testTangent (self)**  
*Test for the tangent value of a given number (in degrees)*
- **def testCotangent (self)**  
*Test for the cotangent value of a given number (in degrees)*

---

### Detailed Description

Tests for calculations of goniometric functions.

---

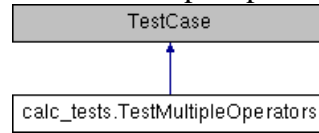
The documentation for this class was generated from the following file:

- **calc\_tests.py**

## calc\_tests.TestMultipleOperators Class Reference

Test for calculations with multiple different operators.

Inheritance diagram for calc\_tests.TestMultipleOperators:



### Public Member Functions

- **def testBasicOperators (self)**  
*Test for calculations with different simple operators.*
- **def testAdvancedOperators (self)**  
*Test for calculations with different advanced operators.*

---

### Detailed Description

Test for calculations with multiple different operators.

---

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

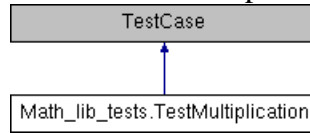
- **calc\_tests.py**



## Math\_lib\_tests.TestMultiplication Class Reference

Tests for multiplication function.

Inheritance diagram for Math\_lib\_tests.TestMultiplication:



### Public Member Functions

- **def testMultByZero (self)**  
*Tests for multiplication by zero.*
- **def testMultByOne (self)**  
*Tests for multiplication by one.*
- **def testInteger (self)**  
*Tests for multiplication by integer.*
- **def testFloat (self)**  
*Tests for multiplication by float.*

---

### Detailed Description

Tests for multiplication function.

---

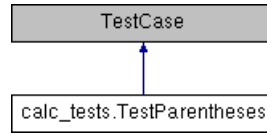
The documentation for this class was generated from the following file:

- **Math\_lib\_tests.py**

## calc\_tests.TestParentheses Class Reference

Tests for calculations with parentheses.

Inheritance diagram for calc\_tests.TestParentheses:



### Public Member Functions

- **def testBasicOperators** (self)  
*Tests for calculations with parentheses and different simple operators.*
- **def testAdvancedOperators** (self)  
*Tests for calculations with parentheses and different advanced operators.*

---

### Detailed Description

Tests for calculations with parentheses.

---

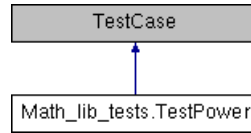
The documentation for this class was generated from the following file:

- **calc\_tests.py**

## Math\_lib\_tests.TestPower Class Reference

Tests for power function.

Inheritance diagram for Math\_lib\_tests.TestPower:



### Public Member Functions

- **def testNotNaturalNumber** (self)  
*Tests for exception with power of not a natural number.*
- **def testIntegerByInteger** (self)  
*Tests for integer's power of integer.*
- **def testFloatByInteger** (self)  
*Tests for float's power of integer.*

---

### Detailed Description

Tests for power function.

---

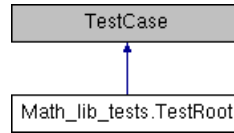
The documentation for this class was generated from the following file:

- **Math\_lib\_tests.py**

## Math\_lib\_tests.TestRoot Class Reference

Tests for root function.

Inheritance diagram for Math\_lib\_tests.TestRoot:



### Public Member Functions

- **def testNotNaturalNumber** (self)  
*Tests for exception with root of not a natural number.*
- **def testNegativeNumber** (self)  
*Tests for exception with root of a negative number.*
- **def testInteger** (self)  
*Tests for integer's root of integer.*
- **def testFloat** (self)  
*Tests for float's root of integer.*

---

### Detailed Description

Tests for root function.

---

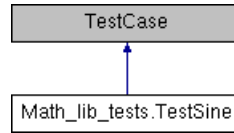
The documentation for this class was generated from the following file:

- **Math\_lib\_tests.py**

## Math\_lib\_tests.TestSine Class Reference

Tests for sine function.

Inheritance diagram for Math\_lib\_tests.TestSine:



### Public Member Functions

- `def testInteger (self)`  
*Tests for sine of an integer.*
- `def testFloat (self)`  
*Tests for sine of a float.*

---

### Detailed Description

Tests for sine function.

---

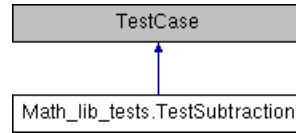
The documentation for this class was generated from the following file:

- `Math_lib_tests.py`

## Math\_lib\_tests.TestSubtraction Class Reference

Tests for subtraction function.

Inheritance diagram for Math\_lib\_tests.TestSubtraction:



### Public Member Functions

- `def testInteger (self)`  
*Tests for subtraction of integers.*
- `def testFloat (self)`  
*Tests for subtraction of floats.*

---

### Detailed Description

Tests for subtraction function.

---

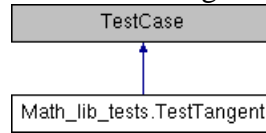
The documentation for this class was generated from the following file:

- `Math_lib_tests.py`

## Math\_lib\_tests.TestTangent Class Reference

Tests for tangent function.

Inheritance diagram for Math\_lib\_tests.TestTangent:



### Public Member Functions

- `def testNotDefined (self)`  
*Tests for exceptions with the tangent of its undefined values.*
- `def testInteger (self)`  
*Tests for tangent of an integer.*
- `def testFloat (self)`  
*Tests for tangent of a float.*

---

### Detailed Description

Tests for tangent function.

---

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

- `Math_lib_tests.py`

# File Documentation

## calc.py File Reference

Processing script for computing the result of input.

### Functions

- def **calc.get\_pars** (str eq)  
*get\_pars*
- def **calc.rewrite** (list eq, int index, insert)  
*rewrite*
- def **calc.to\_list** (str eq)  
*to\_list*
- def **calc.check\_empty** (str eq)  
*check\_neg*
- def **calc.type\_check** (str num)  
*type\_check*
- def **calc.check\_gon** (eq)  
*check\_gon*
- def **calc.findC** (str eq, str sign)  
*findC*
- def **calc.defloat** (num)  
*defloat*
- def **calc.calculate** (str eq)  
*calculate*
- def **calc.evaluate** (str eq)  
*evaluate*

### Variables

- list **calc.signs** = ["+", "-", "\*", "/", "^", "â¸", "!",]

---

## Detailed Description

Processing script for computing the result of input.



**Author**

Matej Hložek, xhloze02, PyJaMa's

**Date**

March/April/May 2021

---

**Function Documentation****def calc.calculate (str eq)**

calculate

The main function to process given equation

**Parameters**

<i>eq</i>	String to calculate
-----------	---------------------

**Returns**

A result

**def calc.check\_empty (str eq)**

check\_neg

Checks if input variable does include empty string or an negative number and repairs it

**Parameters**

<i>eq</i>	A iterable variable to be checked (string or list)
-----------	--

**Returns**

A repaired (if needed) list without empty strings and wrongly written negative numbers

**def calc.check\_gon ( eq)**

check\_gon

Checks if there are goniometric functions in input variable and calculates them

**Parameters**

<i>eq</i>	A list to check
-----------	-----------------

**Returns**

List without goniometric functions

**def calc.defloat ( num)**

defloat

Check if the number can be retyped to integer

**Parameters**

<i>num</i>	A number to be checked (float or integer)
------------	---

**Returns**

Either unchanged float number or integer number

**def calc.evaluate (str *eq*)**

evaluate

connecting function for GUI

**Parameters**

<i>eq</i>	String to calculate
-----------	---------------------

**Returns**

a result or an error string

**def calc.findC (str *eq*, str *sign*)**

findC

Finds first occurrence of given variable

**Parameters**

<i>eq</i>	A iterable variable where the character will be searched (string or list)
<i>sign</i>	A variable to be searched (string)

**Returns**

An index of occurrence of character

**def calc.get\_pars (str *eq*)**

get\_pars

Gets indexes of the largest couple of parentheses

**Parameters**

<i>eq</i>	A string to get indexes from
-----------	------------------------------

**Returns**

[index of 1st, index of 2nd] parentheses or Error string

**def calc.rewrite (list *eq*, int *index*, *insert*)**

rewrite

Reconstructs the list with result of calculation

**Parameters**

<i>eq</i>	A list to reconstruct
<i>index</i>	Index where should the variable "insert" should be inserted (integer)
<i>insert</i>	A variable to be inserted onto desired index

**Returns**

A updated list with result of calculation

**def calc.to\_list (str *eq*)**

to\_list

Converts string into list of numbers and operators

**Parameters**

<i>eq</i>	A string to make a list from
-----------	------------------------------

### Returns

A list of numbers and operators

**def calc.type\_check (str *num*)**

type\_check

Checks type of input number and converts it into adequate type

### Parameters

<i>num</i>	A number to convert (string)
------------	------------------------------

### Returns

A number (either float or integer depending on input number)

---

## Variable Documentation

**list calc.signs = ["+", "-", "\*", "/", "^", "â", "!"]**

### Parameters

<i>signs</i>	Global array with sign characters
--------------	-----------------------------------

## calc\_tests.py File Reference

Tests for **calc.py**.

### Classes

- class **calc\_tests.TestBasics**  
*Tests for basic calculations with one kind of operators.*
- class **calc\_tests.TestGoniometric**  
*Tests for calculations of goniometric functions.*
- class **calc\_tests.TestMultipleOperators**  
*Test for calculations with multiple different operators.*
- class **calc\_tests.TestParentheses**  
*Tests for calculations with parentheses.*
- class **calc\_tests.TestErrors**  
*Tests for error handling.*

### Functions

- def **calc\_tests.test\_ok** (str test\_name)  
*Test control.*

---

## Detailed Description

Tests for **calc.py**.

### Author

Matej Hložek, xhloze02, PyJaMa's

### Date

March 2021

## calculator.py File Reference

Realization of graphical user interface.

### Classes

- class **calculator.CalculatorWindow**  
*Class of main window.*
- class **calculator.ScrollLabel**  
*Class of scrollable labels.*
- class **calculator.History**  
*Class of history.*

### Functions

- def **calculator.\_\_init\_\_** (self)  
*Initialization of main window.*
- def **calculator.keyPressEvent** (self, event)  
*Connecting keys on keyboard to the functions.*
- def **calculator.show\_input** (self)  
*Formatting expression and showing it as input.*
- def **calculator.zero\_and\_space** (self)  
*Preparing expression for some functions.*
- def **calculator.add\_to\_expression** (self, add)  
*Adds operation to expression.*
- def **calculator.digit\_pressed** (self, digit)  
*Adding digit to input.*
- def **calculator.decimal\_pressed** (self)  
*Adding decimal point to input.*
- def **calculator.function\_pressed** (self, funct)  
*Adding function to input.*
- def **calculator.paren\_pressed** (self, paren)  
*Adding parentheses to input.*
- def **calculator.root\_pressed** (self)  
*Adding root to input.*
- def **calculator.trig\_pressed** (self, funct)  
*Adding trigonometric function to input.*
- def **calculator.clear\_pressed** (self)

*Resetting state of calculator.*

- **def calculator.del\_pressed (self)**  
*Deleting last action.*
- **def calculator.equals\_pressed (self)**  
*Calculating expression and showing result.*
- **def calculator.help\_window (self)**  
*Displaying help message.*
- **def calculator.history\_window (self)**  
*Displaying history.*

## Variables

- string **calculator.history** = "  
int **calculator.lparen** = 0  
*lparen*
- int **calculator.rparen** = 0  
*rparen*
- int **calculator.equals** = 0  
*equals*
- **calculator.expression**  
*expression*

---

## Detailed Description

Realization of graphical user interface.

### Author

Marek Tiss, xtissm00, PyJaMa's

### Date

March/April 2021

---

## Function Documentation

**def calculator.\_\_init\_\_ ( self)**

Initialization of main window.

Setting name and logo of main window

```
def calculator.add_to_expression ( self, add)
```

Adds operation to expression.

#### Parameters

<i>add</i>	Which operation should be added
------------	---------------------------------

```
def calculator.digit_pressed ( self, digit)
```

Adding digit to input.

#### Parameters

<i>digit</i>	Which digit was pressed
--------------	-------------------------

```
def calculator.function_pressed ( self, funct)
```

Adding function to input.

#### Parameters

<i>funct</i>	Which function was pressed
--------------	----------------------------

```
def calculator.paren_pressed ( self, paren)
```

Adding parentheses to input.

#### Parameters

<i>paren</i>	Which parenthesis was pressed
--------------	-------------------------------

```
def calculator.trig_pressed ( self, funct)
```

Adding trigonometric function to input.

#### Parameters

<i>funct</i>	Which function was pressed
--------------	----------------------------

---

## Variable Documentation

```
calculator.history = "
```

Global variable that tracks history of inputs/outputs

## Math\_lib.py File Reference

Implementation of mathematical functions and constants.

### Functions

- `def Math_lib.add (x, y)`  
*Addition.*
- `def Math_lib.sub (x, y)`  
*Subtraction.*
- `def Math_lib.mult (x, y)`  
*Multiplication.*
- `def Math_lib.div (x, y)`  
*Division.*
- `def Math_lib.pow (x, n)`  
*Exponentiation.*
- `def Math_lib.root (x, n)`  
*Root.*
- `def Math_lib.fact (x)`  
*Factorial.*
- `def Math_lib.sin (x)`  
*Sine.*
- `def Math_lib.cos (x)`  
*Cosine.*
- `def Math_lib.tan (x)`  
*Tangent.*
- `def Math_lib.cotg (x)`  
*Cotangent.*

### Variables

- `float Math_lib.pi = 3.14159265359`
- `float Math_lib.e = 2.71828182846`
- `int Math_lib.d_digits = 11`

---

### Detailed Description

Implementation of mathematical functions and constants.



**Author**

Marek Tiss, xtissm00, PyJaMa's

**Date**

March 2021

---

**Function Documentation****def Math\_lib.add ( x, y)**

Addition.

Adds 2 numbers together (x+y)

**Parameters**

x	First number
y	Second number

**Returns**

Sum of x and y

**def Math\_lib.cos ( x)**

Cosine.

Calculates cosine of number x (adjacent/hypotenuse)

**Parameters**

x	Number in degrees
---	-------------------

**Returns**

Cosine of x

**def Math\_lib.cotg ( x)**

Cotangent.

Calculates cotangent of number x (adjacent/opposite)

**Parameters**

x	Number in degrees
---	-------------------

**Returns**

cotangent of x

**Exceptions**

<i>ValueError</i>	if x is 180*I, while I is whole number
-------------------	--

**def Math\_lib.div ( x, y)**

Division.

Divides number x by number y (x/y)

**Parameters**

x	Dividend
---	----------

<i>y</i>	Divisor
----------	---------

#### Returns

Quotient of x and y

#### Exceptions

<i>ZeroDivisionError</i>	if y = 0
--------------------------	----------

**def Math\_lib.fact ( x)**

Factorial.

Calculates factorial of number x (x!)

#### Parameters

<i>x</i>	Number
----------	--------

#### Returns

Factorial of x

#### Exceptions

<i>ValueError</i>	if x isn't natural number
-------------------	---------------------------

**def Math\_lib.mult ( x, y)**

Multiplication.

Multiplies numbers x and y (x\*y)

#### Parameters

<i>x</i>	First number
<i>y</i>	Second number

#### Returns

Product of x and y

**def Math\_lib.pow ( x, n)**

Exponentiation.

Raises number x to the power of n (x^n)

#### Parameters

<i>x</i>	Base number
<i>n</i>	Exponent

#### Returns

X to the power of n

#### Exceptions

<i>ValueError</i>	if n isn't natural number
-------------------	---------------------------

**def Math\_lib.root ( x, n)**

Root.

Calculates n-th root of number x

#### Parameters

<i>x</i>	Base number
<i>n</i>	Order of the root

### Returns

N-th root of x

### Exceptions

<i>ValueError</i>	if n isn't natural number
<i>ValueError</i>	if x is negative and n is even

**def Math\_lib.sin ( x)**

Sine.

Calculates sine of number x (opposite/hypotenuse)

### Parameters

x	Number in degrees
---	-------------------

### Returns

Sine of x

**def Math\_lib.sub ( x, y)**

Subtraction.

Subtracts number y from number x (x-y)

### Parameters

x	First number
y	Second number

### Returns

Difference of x and y

**def Math\_lib.tan ( x)**

Tangent.

Calculates tangent of number x (opposite/adjacent)

### Parameters

x	Number in degrees
---	-------------------

### Returns

Tangent of x

### Exceptions

<i>ValueError</i>	if x is $90 + 180 \cdot I$ , while I is whole number
-------------------	--

---

## Variable Documentation

**Math\_lib.d\_digits = 11**

Number of decimal digits:

**Math\_lib.e = 2.71828182846**

Constant e

**Math.lib.pi = 3.14159265359**

Constant pi

## Math\_lib\_tests.py File Reference

Tests for **Math\_lib.py**.

### Classes

- class **Math\_lib\_tests.TestAddition**  
*Tests for addition function.*
- class **Math\_lib\_tests.TestSubtraction**  
*Tests for subtraction function.*
- class **Math\_lib\_tests.TestMultiplication**  
*Tests for multiplication function.*
- class **Math\_lib\_tests.TestDivision**  
*Tests for division function.*
- class **Math\_lib\_tests.TestPower**  
*Tests for power function.*
- class **Math\_lib\_tests.TestRoot**  
*Tests for root function.*
- class **Math\_lib\_tests.TestFactorial**  
*Tests for factorial function.*
- class **Math\_lib\_tests.TestSine**  
*Tests for sine function.*
- class **Math\_lib\_tests.TestCosine**  
*Tests for cosine function.*
- class **Math\_lib\_tests.TestTangent**  
*Tests for tangent function.*
- class **Math\_lib\_tests.TestCotangent**  
*Tests for cotangent function.*
- class **Math\_lib\_tests.TestConstants**  
*Tests for constants in functions.*

### Functions

- def **Math\_lib\_tests.test\_ok** (str test\_name)  
*Test control.*

---

## Detailed Description

Tests for **Math\_lib.py**.

### Author

Matej Hložek, xhloze02, PyJaMa's

### Date

March 2021