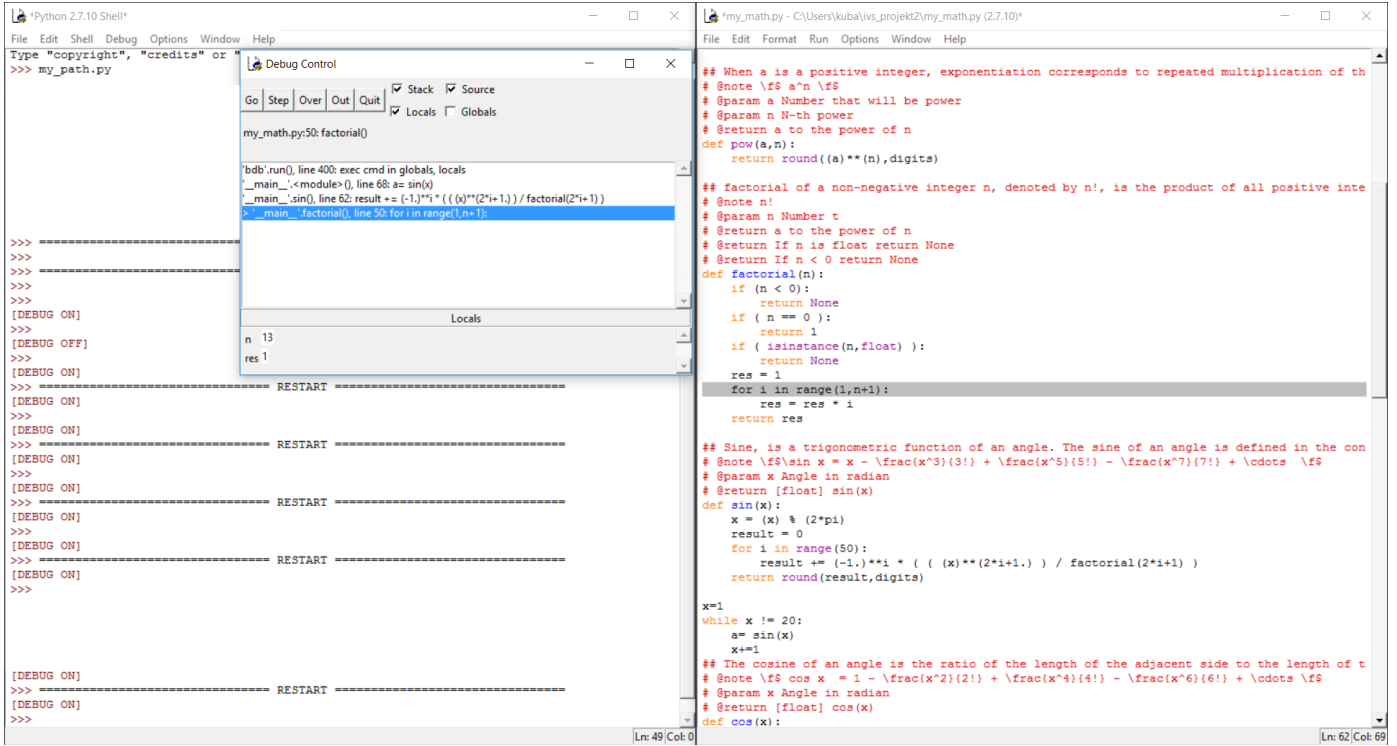


1 Debugging

In these pictures, we can see debugging factorial function.



The image displays a Python IDE with two windows. The left window, titled 'Python 2.7.10 Shell', shows a shell prompt and a debug console. The right window, titled 'my_math.py - C:\Users\kuba\ivs_projekt2\my_math.py (2.7.10)*', shows the source code of a module named 'my_math'.

Left Window: Python 2.7.10 Shell

```
>>> Type "copyright", "credits" or
>>> my_path.py
```

Debug Console:

```
my_math.py:51: factorial()

bdb> run(), line 400: exec cmd in globals, locals
_main_<module>(). line 68: a= sin(x)
_main_<sin0, line 62: result += (-1)**i * ((x)**(2*i+1)) / factorial(2*i+1)
> _main_<factorial0, line 51: res = res * i
```

Locals:

```
i 3
n 13
res 2
```

Right Window: my_math.py

```
File Edit Format Run Options Window Help

## When a is a positive integer, exponentiation corresponds to repeated multiplication of th
# @note \f$ a^n \f$
# @param a Number that will be power
# @param n N-th power
# @return a to the power of n
def pow(a,n):
    return round((a)**(n),digits)

## factorial of a non-negative integer n, denoted by n!, is the product of all positive inte
# @note n!
# @param n Number t
# @return a to the power of n
# @return if n is float return None
# @return if n < 0 return None
def factorial(n):
    if (n < 0):
        return None
    if ( n == 0 ):
        return 1
    if ( isinstance(n,float) ):
        return None
    res = 1
    for i in range(1,n+1):
        res = res * i
    return res

## Sine, is a trigonometric function of an angle. The sine of an angle is defined in the con
# @note \f$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots \f$
# @param x Angle in radian
# @return [float] sin(x)
def sin(x):
    x = x % (2*pi)
    result = 0
    for i in range(50):
        result += (-1)**i * ( (x)**(2*i+1) ) / factorial(2*i+1)
    return round(result,digits)

x=1
while x != 20:
    a= sin(x)
    x+=1

## The cosine of an angle is the ratio of the length of the adjacent side to the length of t
# @note \f$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots \f$
# @param x Angle in radian
# @return [float] cos(x)
def cos(x):
    result = 0
```

The image shows a Python 2.7.10 Shell and a Python 2.7.10 IDE. The IDE window on the right displays the source code of a script named `my_math.py`. The script defines a `pow` function, a `factorial` function, and a `sin` function. The `factorial` function is currently being debugged. The Shell window on the left shows the execution of the script, with a debug console window open. The debug console shows the function being called with arguments `(4, 24)` and the result being calculated as `24`.

[illegible][illegible]

The screenshot shows a Python IDE with a debug session. The main window displays the source code of a module named `my_math.py`. The code includes functions for calculating powers, factorials, and trigonometric functions like sine and cosine. The left sidebar shows the 'Debug Control' panel with a stack trace. The bottom status bar indicates the current line and column.

Source Code (my_math.py):

```

## When a is a positive integer, exponentiation corresponds to repeated multiplication of the
# @note \f$a^n = 1 \f$
# @param N Number that will be power
# @param n N-th power
# @return a to the power of n
def pow(a,n):
    return round((a)**(n),digits)

## factorial of a non-negative integer n, denoted by n!, is the product of all positive inte
# @note n!
# @param N Number t
# @return a to the power of n
# @return If n is float return None
# @return If n < 0 return None
def factorial(n):
    if (n < 0):
        return None
    if ( n == 0 ):
        return 1
    if ( isinstance(n,float) ):
        return None
    res = 1
    for i in range(1,n+1):
        res = res * i
    return res

## Sine, is a trigonometric function of an angle. The sine of an angle is defined in the con
# @note \f$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \f$
# @param x Angle in radian
# @return [float] sin(x)
def sin(x):
    x = (x) % (2*pi)
    result = 0
    for i in range(50):
        result += (-1)**i * ( (x)**(2*i+1.) ) / factorial(2*i+1)
    return round(result,digits)

x=1
while x != 20:
    a= sin(x)
    x+=1

## The cosine of an angle is the ratio of the length of the adjacent side to the length of t
# @note \f$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots \f$
# @param x Angle in radian
# @return [float] cos(x)
def cos(x):
    result = 0

```

Debug Control Panel:

Go Step Over Out Quit Stack Source Locals Globals

my_math.py:51: factorial()

bdb.run(), line 400: exec cmd in globals, locals
 main<.<module>(), line 68: a= sin(x)
 main<.<sin(), line 62: result += (-1)**i * ((x)**(2*i+1)) / factorial(2*i+1)
 > _main_<.<factorial(), line 51: res = res * i

Locals

i 9
 n 15
 res 40320

Ln: 49 Col: 0

The screenshot displays two side-by-side windows from a Python IDE.

- Left Window:** Titled "Python 2.7.10 Shell". It shows a file named `my_path.py`. The main script contains several `>>>` prompts followed by `[DEBUG ON]`, `[DEBUG OFF]`, and `[RESTART]` commands. A `Debug Control` window is open over the script, showing tabs for Go, Step, Over, Out, Quit, Stack, Source, Locals, and Globals. The `Locals` tab is active, displaying variables `i` with value 12, `n` with value 15, and `res` with value 39916800. The background code includes comments like "`#note <module>()`" and "`#main_<sin()>`".
- Right Window:** Titled "Python 2.7.10 Shell". It shows a file named `my_math.py`. The code defines functions `factorial(n)` and `sin(x)`. The `factorial` function calculates the factorial of a non-negative integer `n`. The `sin` function uses Taylor series expansion to approximate the sine of an angle `x` in radians. Comments explain the trigonometric identity used for `sin(x)` and the series for `cos(x)`.

