R3・confidential

Canon
production engineering headquarters
production apparatus component development center

CRAVIS-mini
Manual for script developers

	Date	Author	Revision contents
-	2016.06.01		New making
01	2017.12.06		Translation to English
02			
03			
04			
05			
06			
07			
08			
09			
10			

【 Background 】

This book purchases CRAVIS-mini and becomes the manual for the people that script development is in charge of mainly. There are contents wanting you to know it to a minimum to perform script development and explains it.

【 Contents 】

1. About a development environment

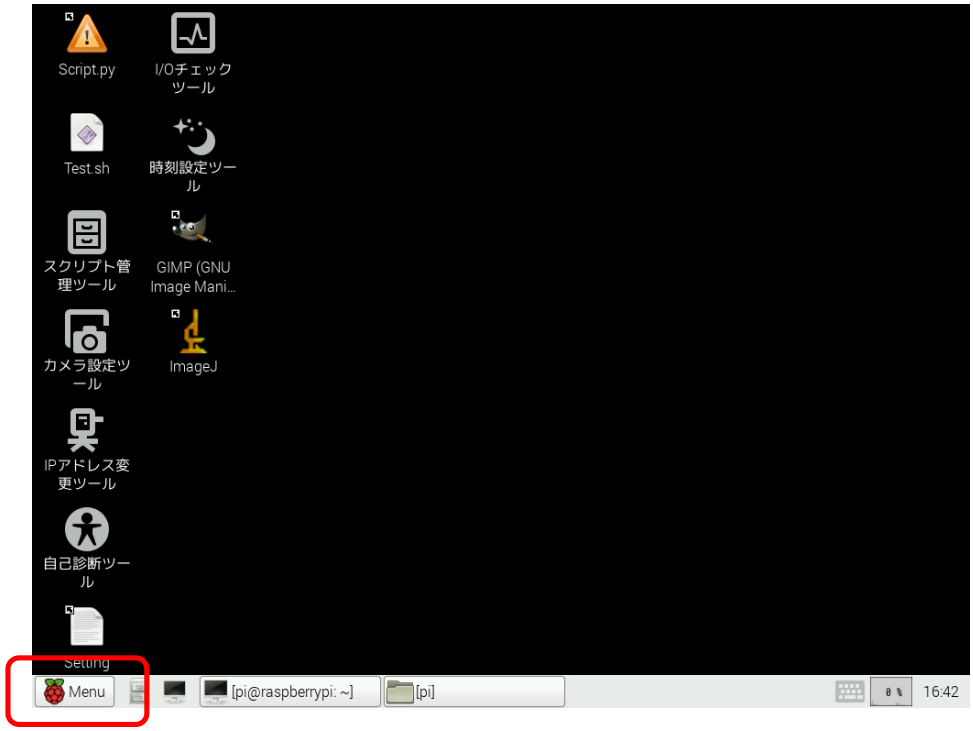
I perform the script development of CRAVIS-mini using a programming language Python.
In addition, I use OpenCV, Numpy, a library such as scipy for the numerical computation function in the script, an image processing function.
When the environment developing this script greatly separates you, it is outside, and there are two ways of the method encoding using method encoding in the "CRAVIS-mini" main body and "WindowsPC".
"CRAVIS-mini" is "WindowsPC" and, here, explains software for the development to use each.

1.1 Development environment in CRAVIS-mini ①～Geany

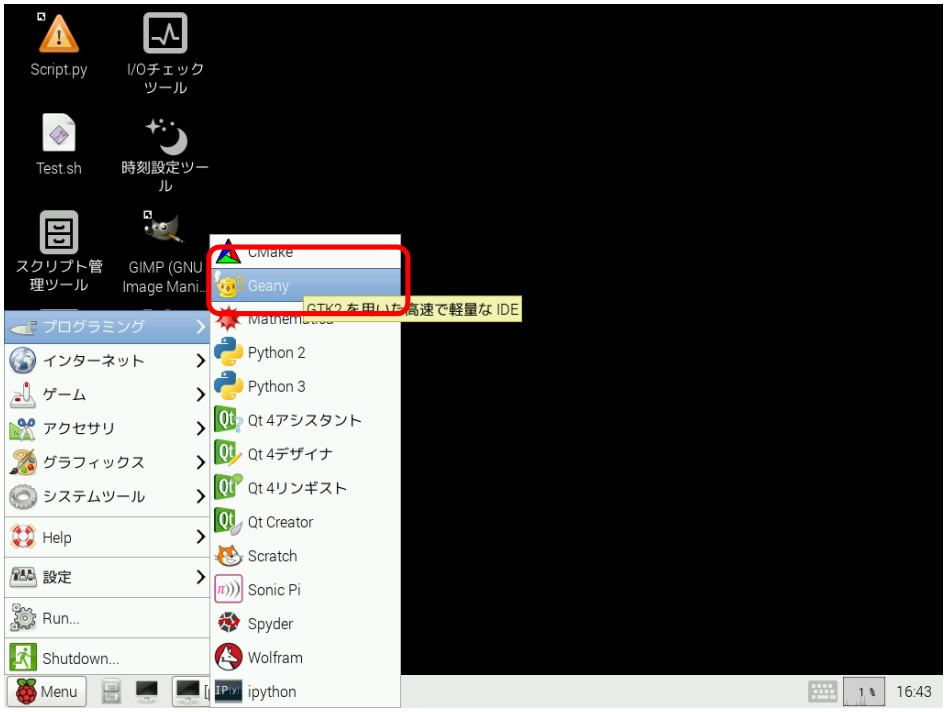
Geany is a text editor-based script development environment.
When I perform the simple alteration of the script, practice confirmation, I use it.

【 Start method of Geany 】

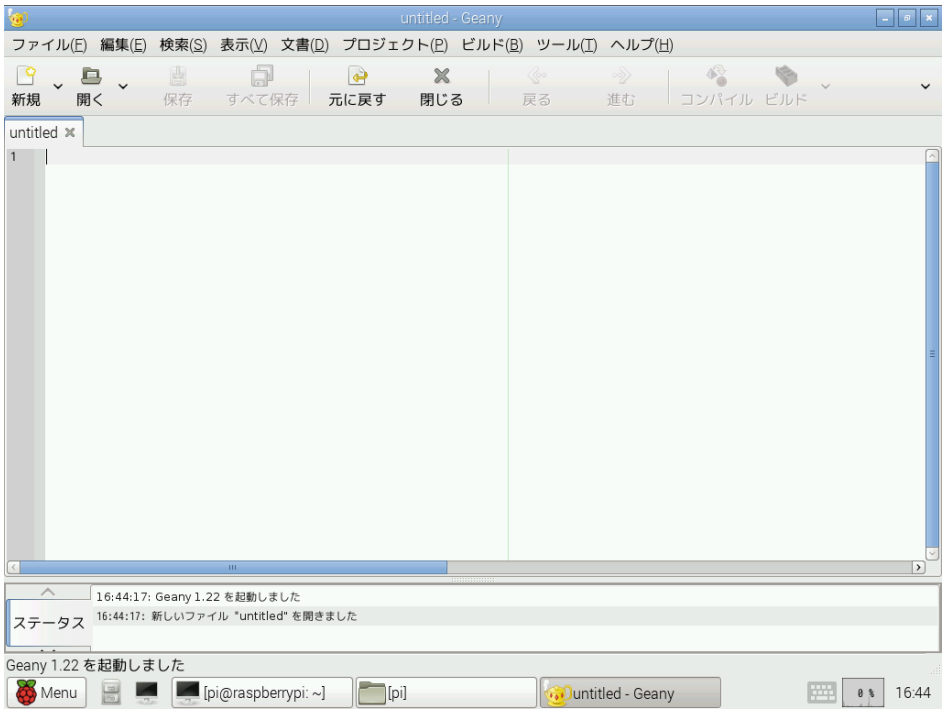
①I click "Menu button" in the Desktop.



②I click "Geany button" in the Start Menu.

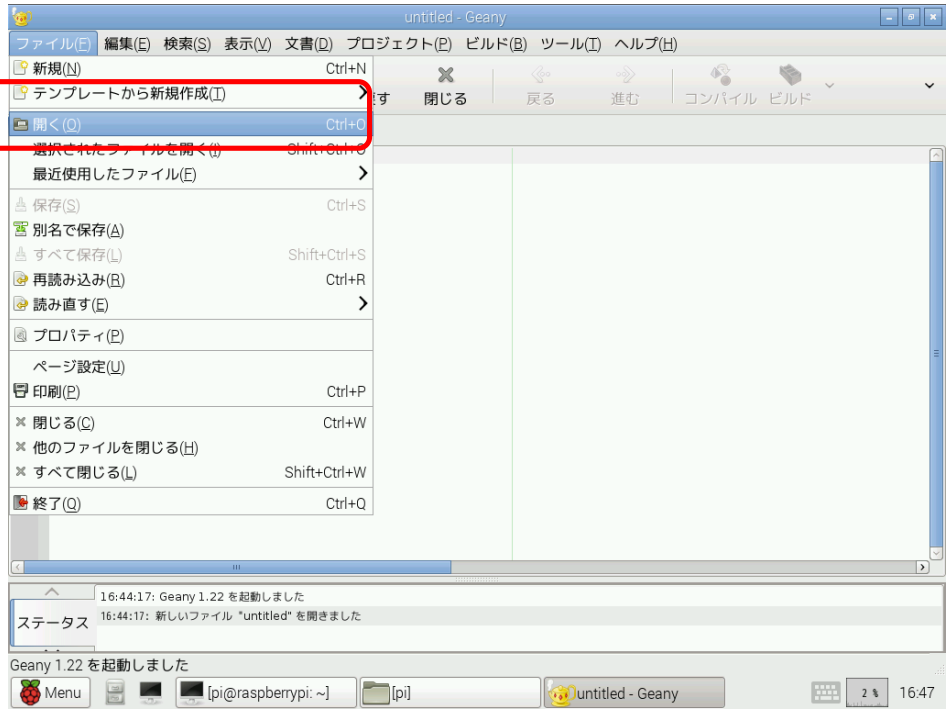


③Geany starts.

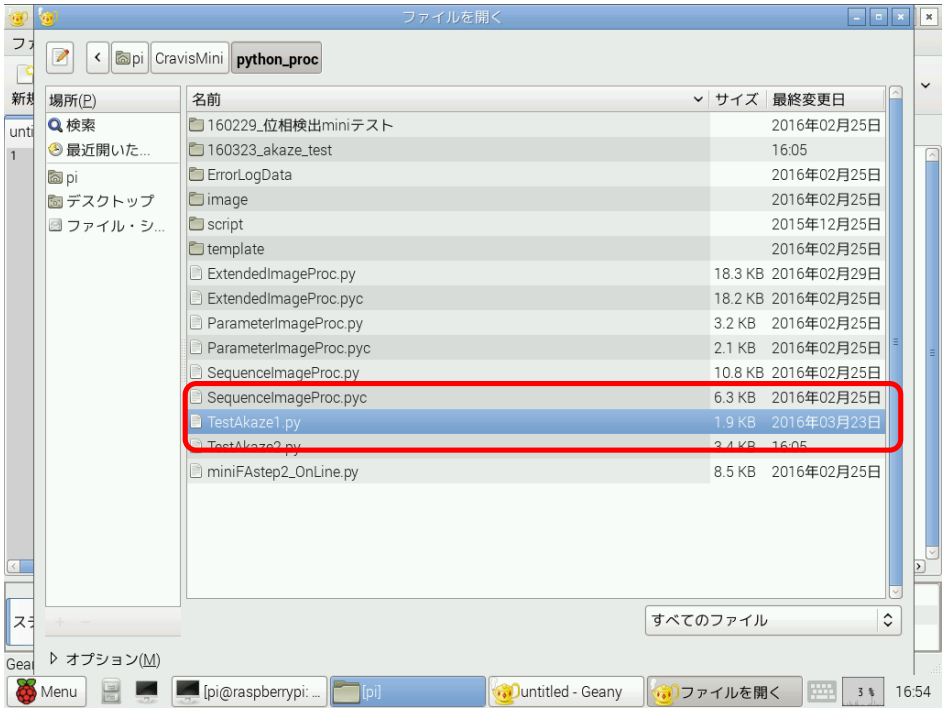


【 How to open script files in Geany 】

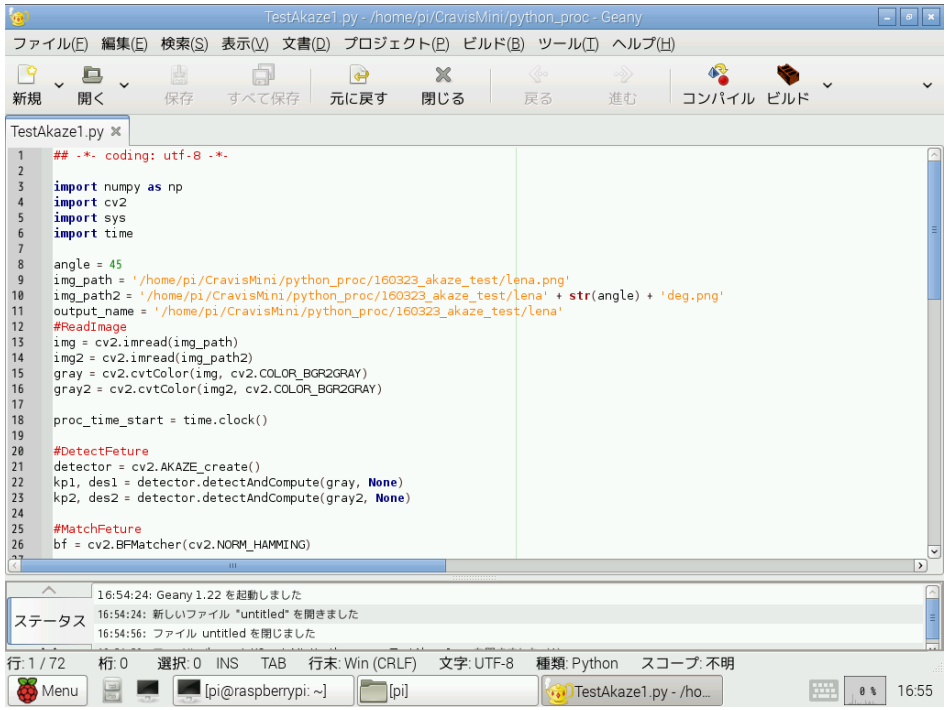
①I click "the Open button" in the File menu.



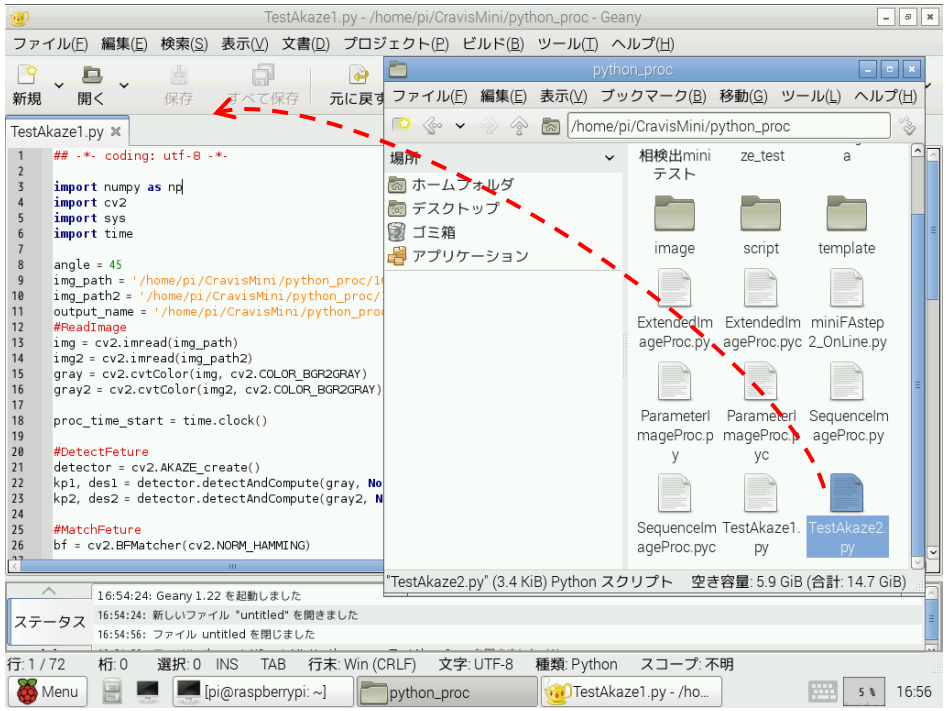
②I choose any script file (.py file).



③The script file which I chose on the text editor of Geany is opened.

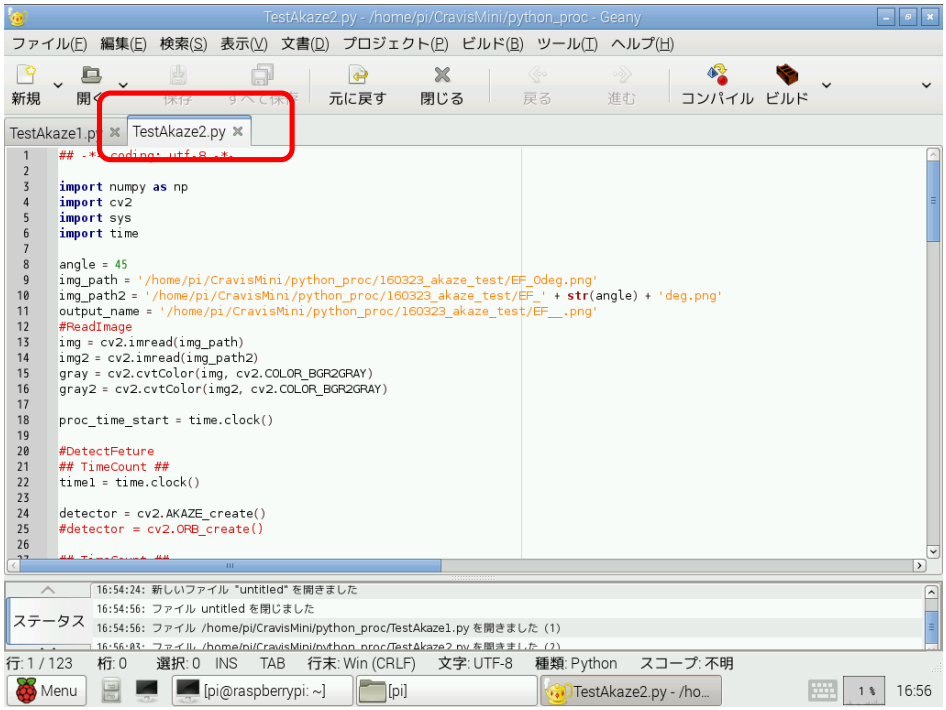


(supplement) Even drag & drop opens a script file with a script file on Geany.

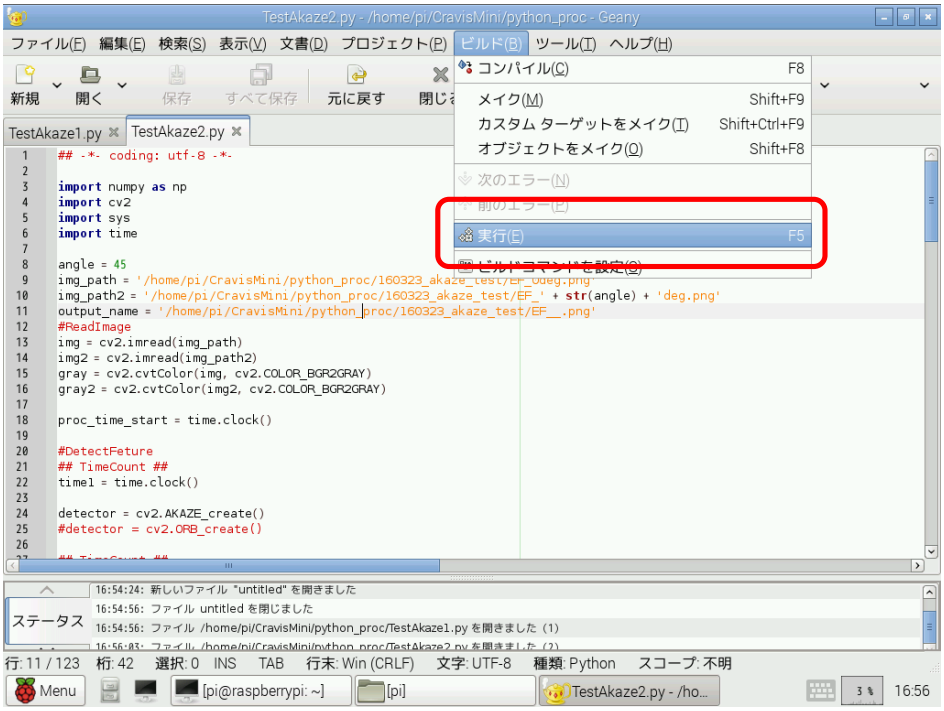


【 Script file practice in Geany 】

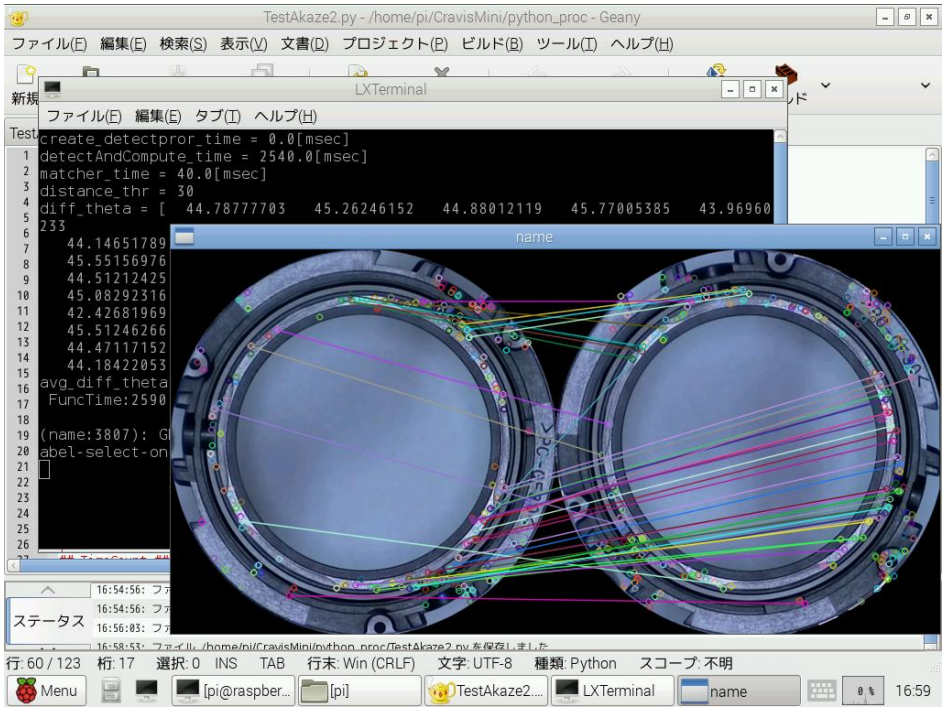
①I make the tab of the script file to carry out the front.



②I push "the practice button" of the build menu.



③A terminal is output, and a result is displayed.

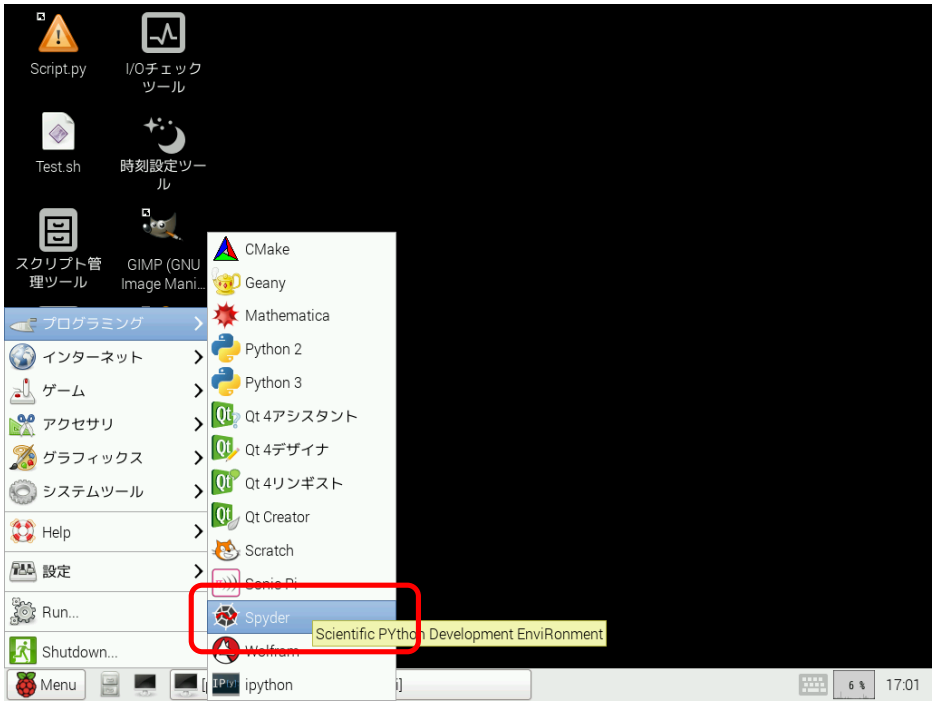


1.2 Development environment ②～ Spyder in CRAVIS-mini

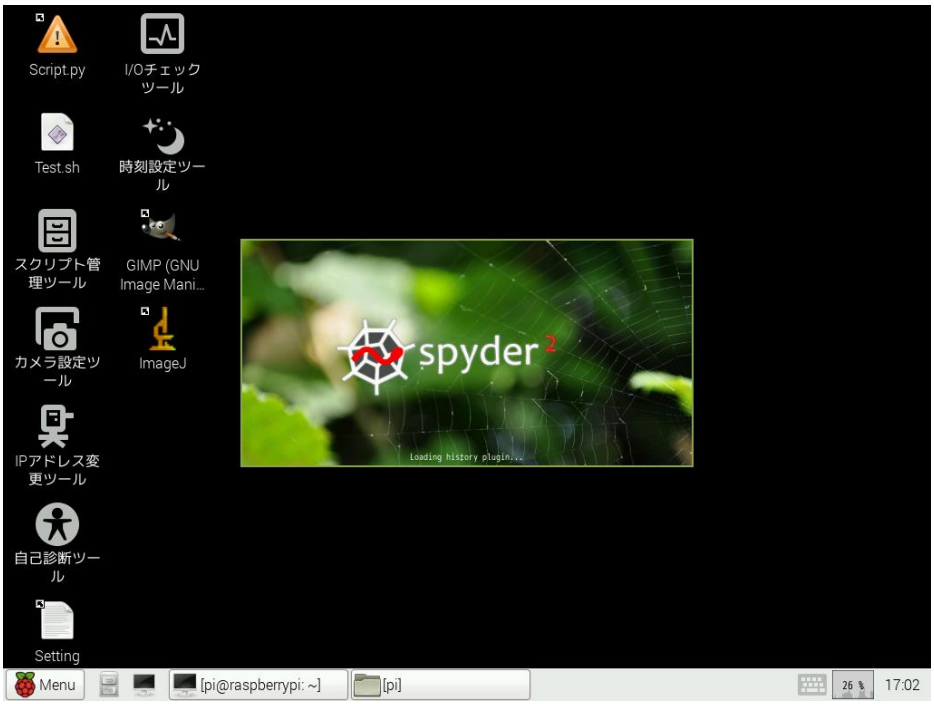
Spyder is overall development environment having the debugging tools of a variable view, the breakpoint that can confirm the contents of an editor and the variable with a syntax highlight and an automatic supplement function.
When I want to perform debug in greater detail, about application, I use it.

【 Start method of Spyder 】

①I click "Spyder button" in the Start Menu.

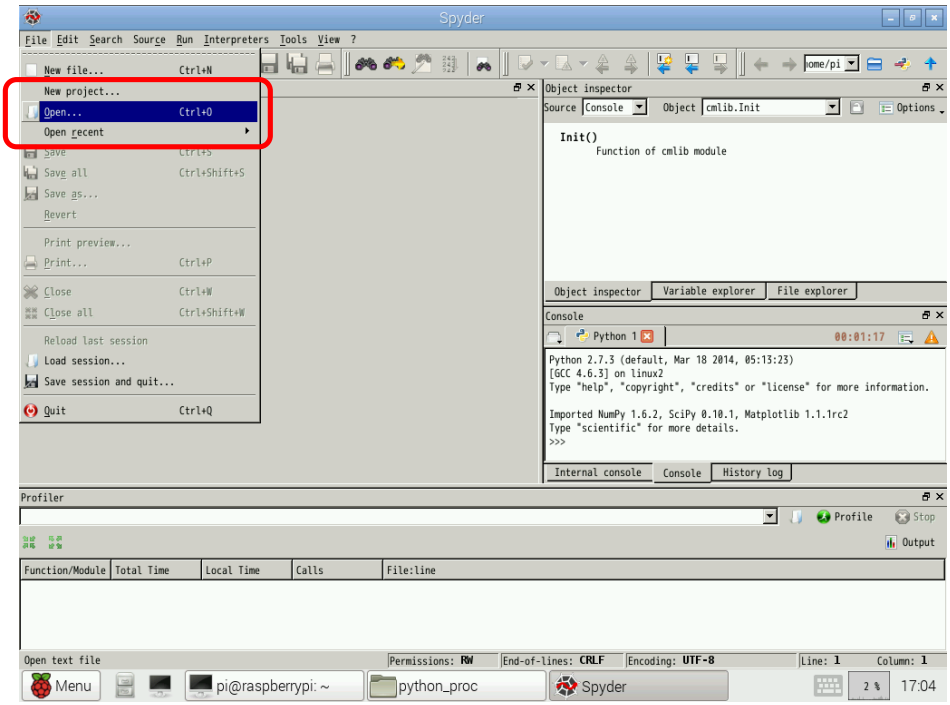


②Spyder starts.

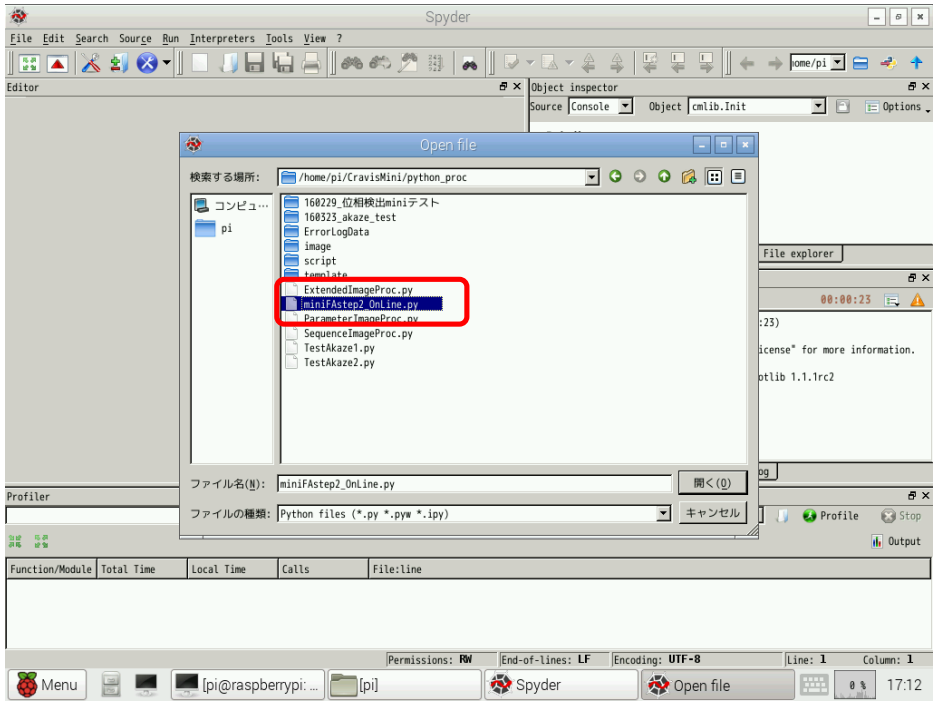


【 How to open script files in Spyder 】

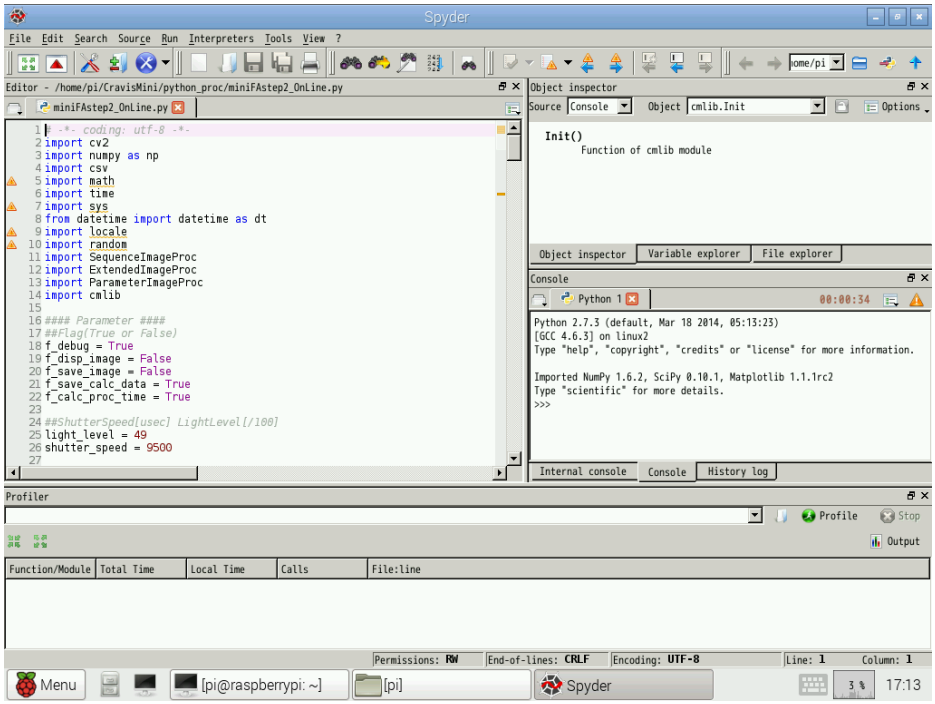
①I click "the Open file button" in the File menu.



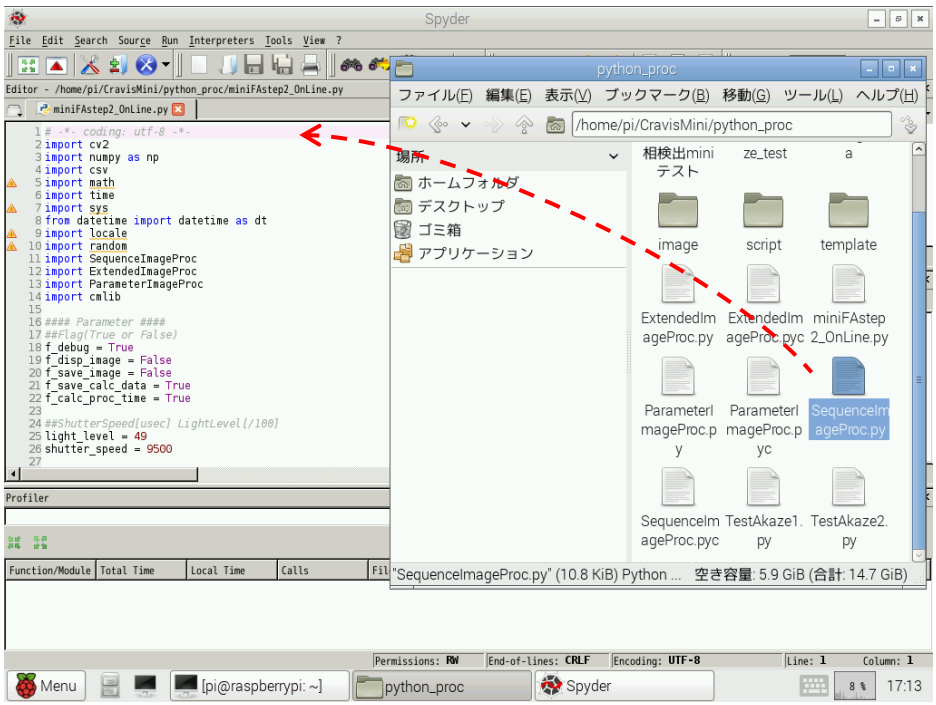
②I choose any script file (.py file).



③The script file which I chose on the text editor of Spyder is opened.

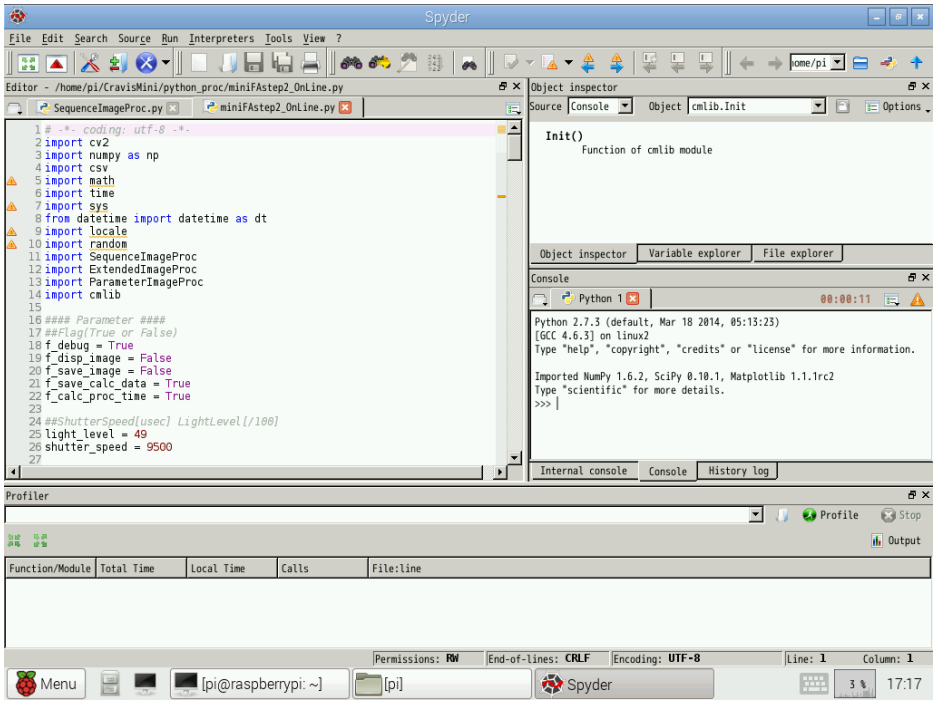


(supplement) Even drag & drop opens a script file on the text editor of Spder with a script file.

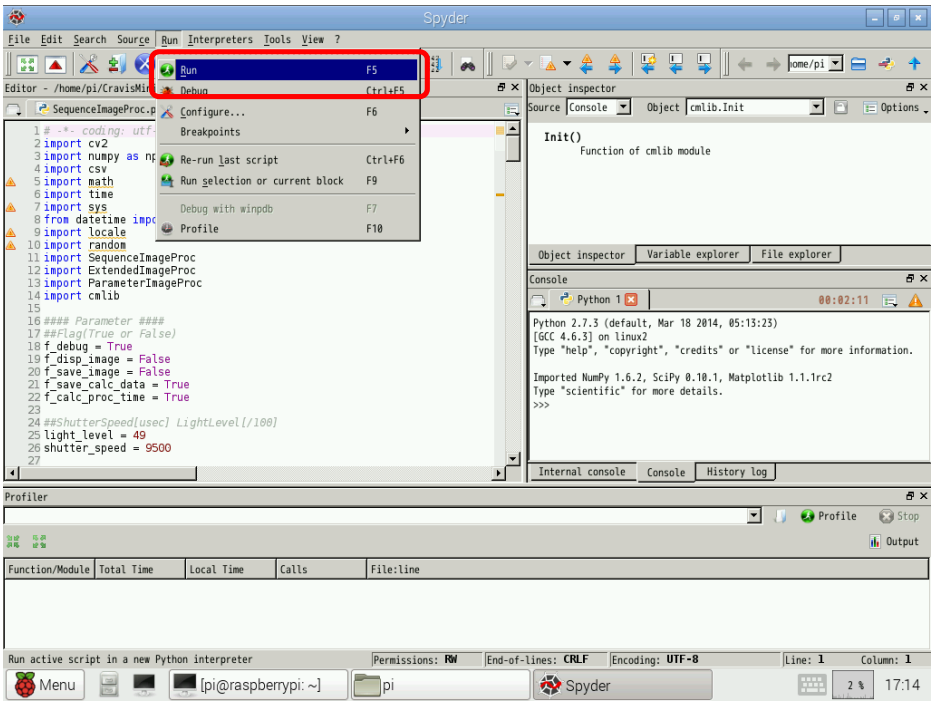


【 Script file practice in Spyder 】

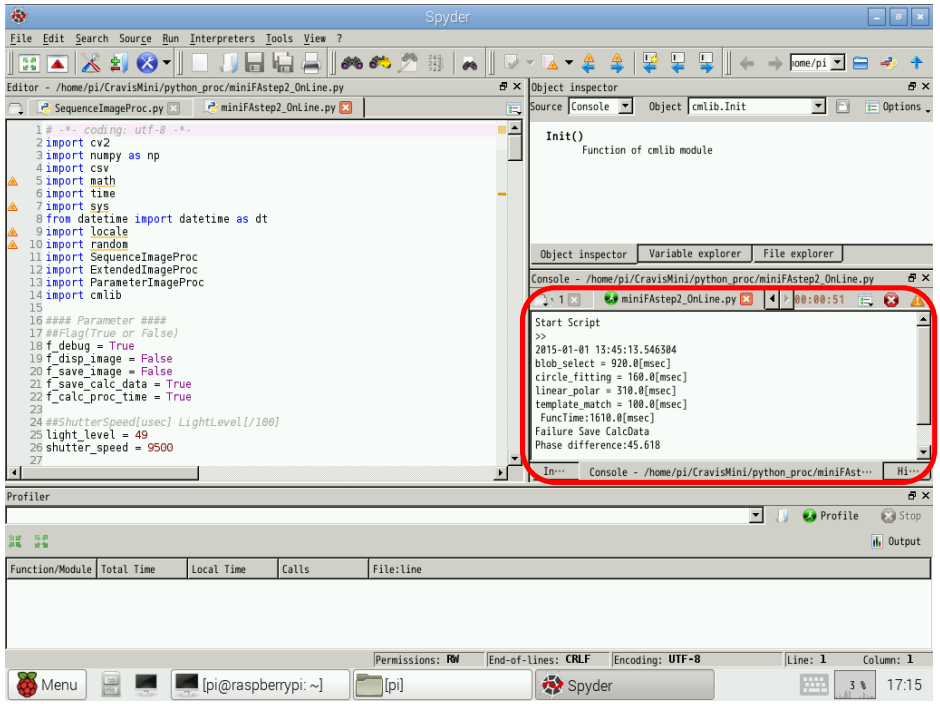
①I make the tab of the script file to carry out the front.



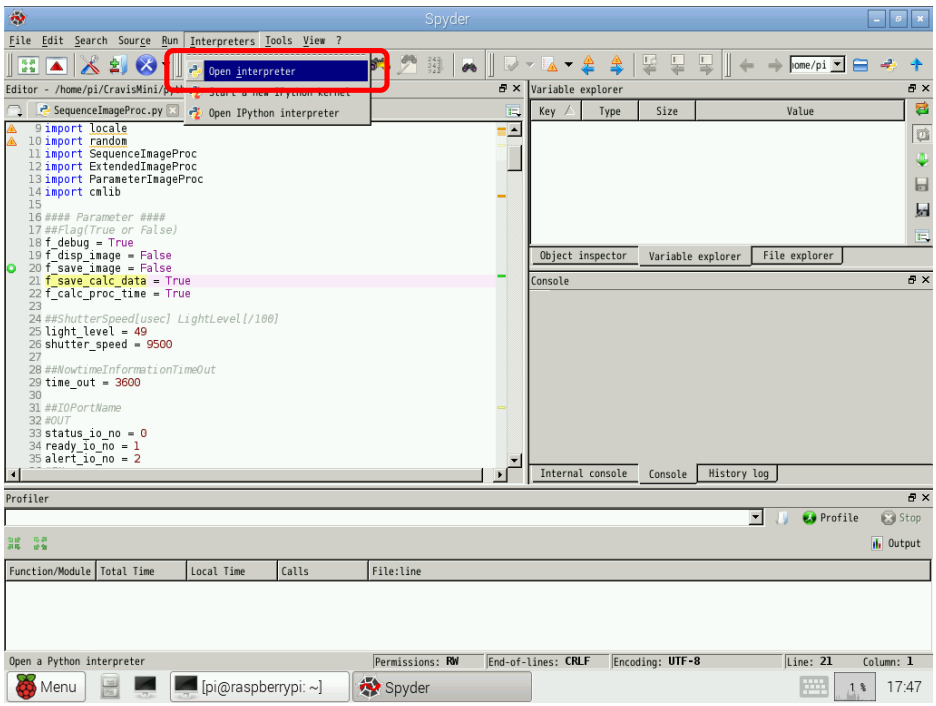
②I push "the Run button" of the Run menu.



③A result is displayed by Ipython console window.



(supplement) When I close Ipython console window, I may not execute a script.
You push "the Open interpreter button" from "the Interpreters button" of the menu, and please display Ipython console window on this occasion.

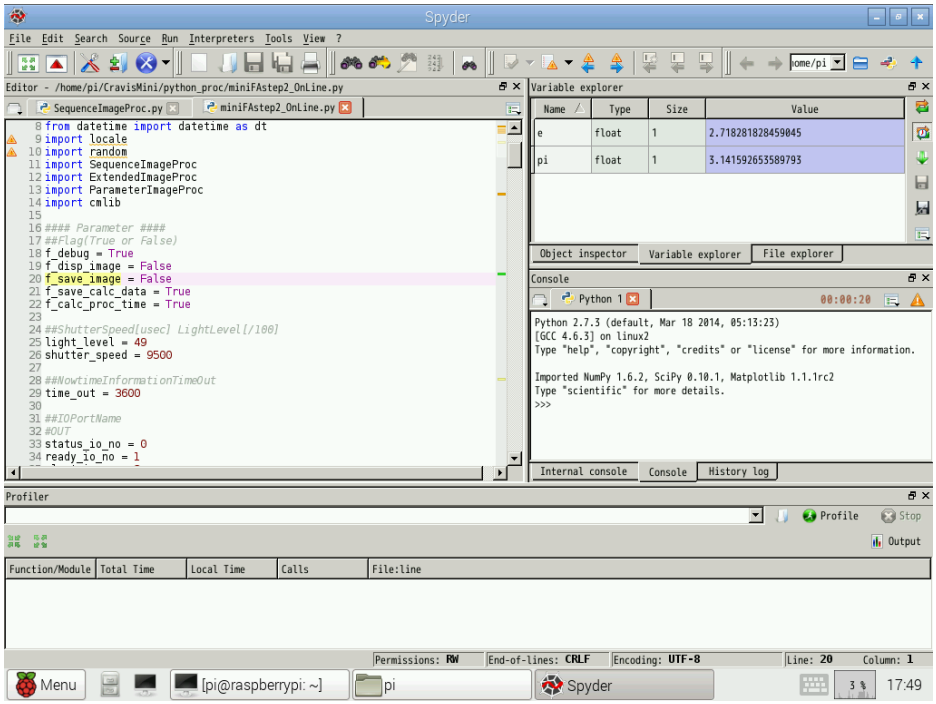


【 Script file debugging method in Spyder 】

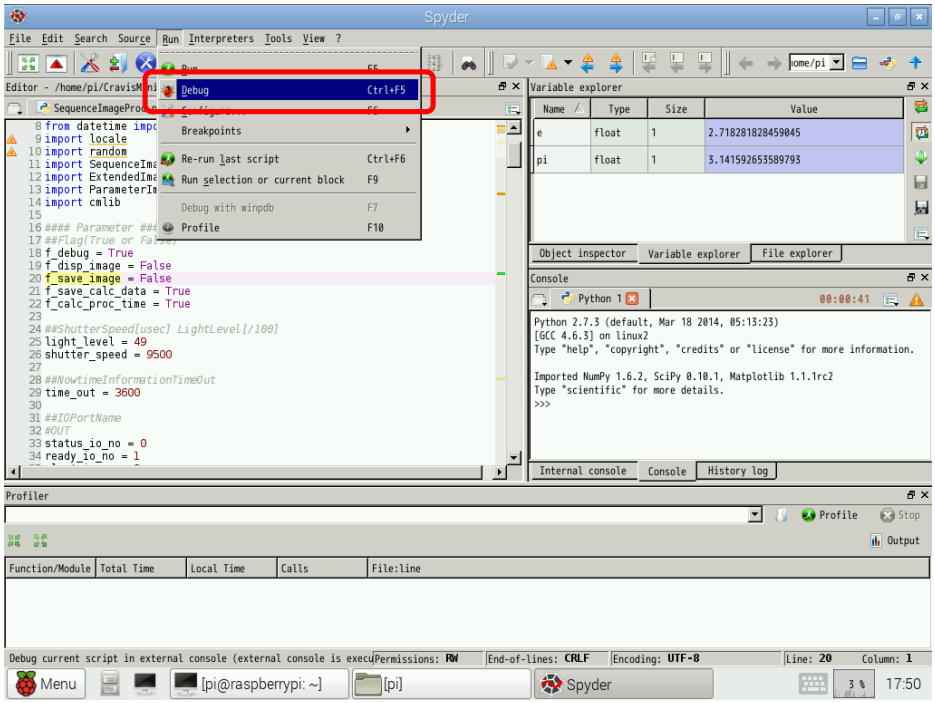
It is accompanied, and a debugging function called pdb can perform the detailed debugging of the script using pdb in Spyder. Here, I explain the usage of pdb and the variable view as a method of the debugging.

< Start method of pdb >

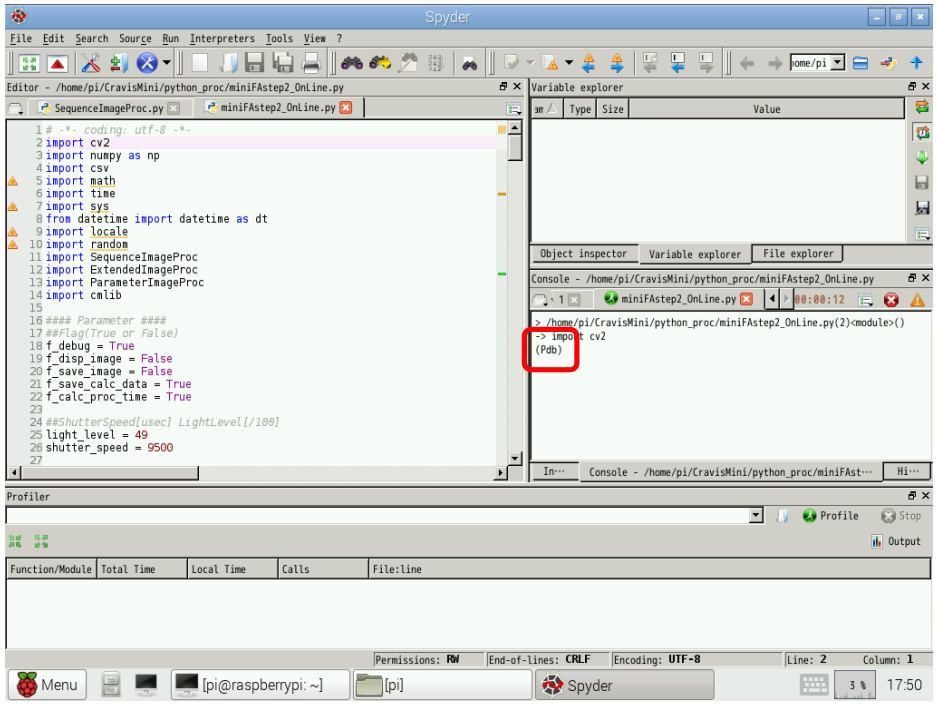
①I make the tab of the script file debugging the front.



②I push "the Debug button" of the Run menu.



③I am displayed after Ipython console window with pdb> in the line.



(supplement) Like a script practice method,
When I close Ipython console window, I may not execute a script.
You push "the Open interpreter button" from "the Interpreters button" of the menu, and please display Ipython console window on this occasion.

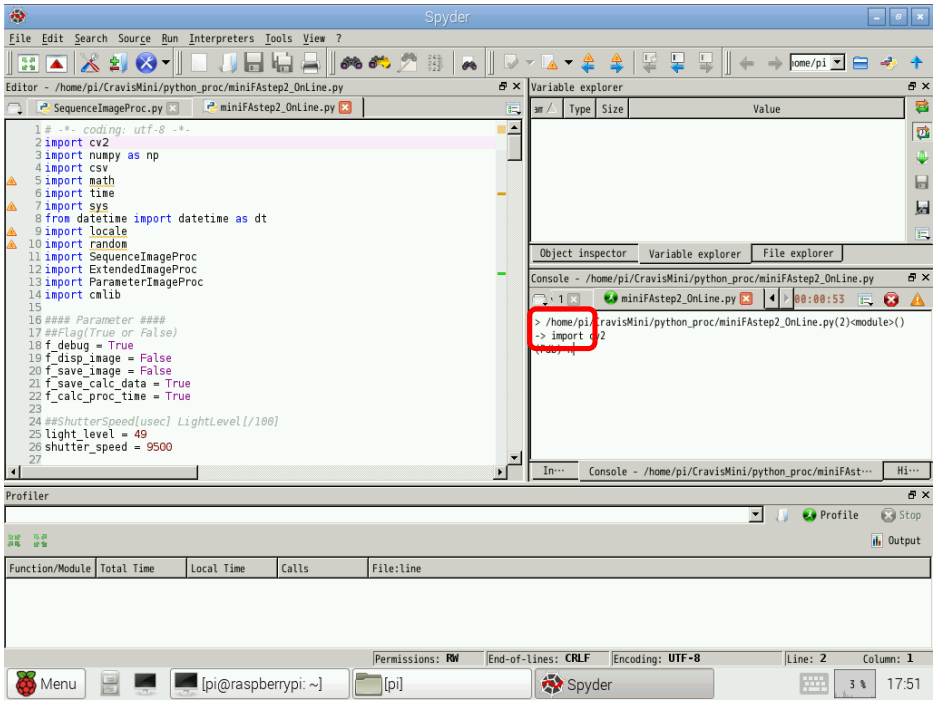
< Debugging method using pdb >

The debugging method of the script using pdb inputs a command on Ipython console window and performs it. Here, I explain "the step practice" that is a representative debugging method and "the debugging using the breakpoint".

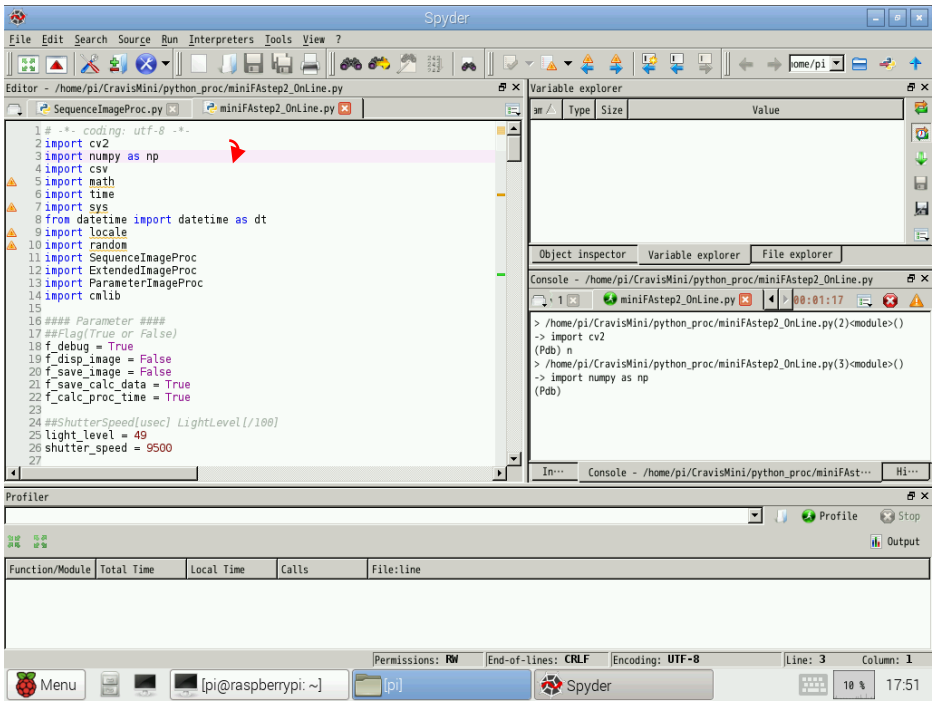
「 Step practice 」

The step practice is the method that I debug while carrying out the code of the script by one line.

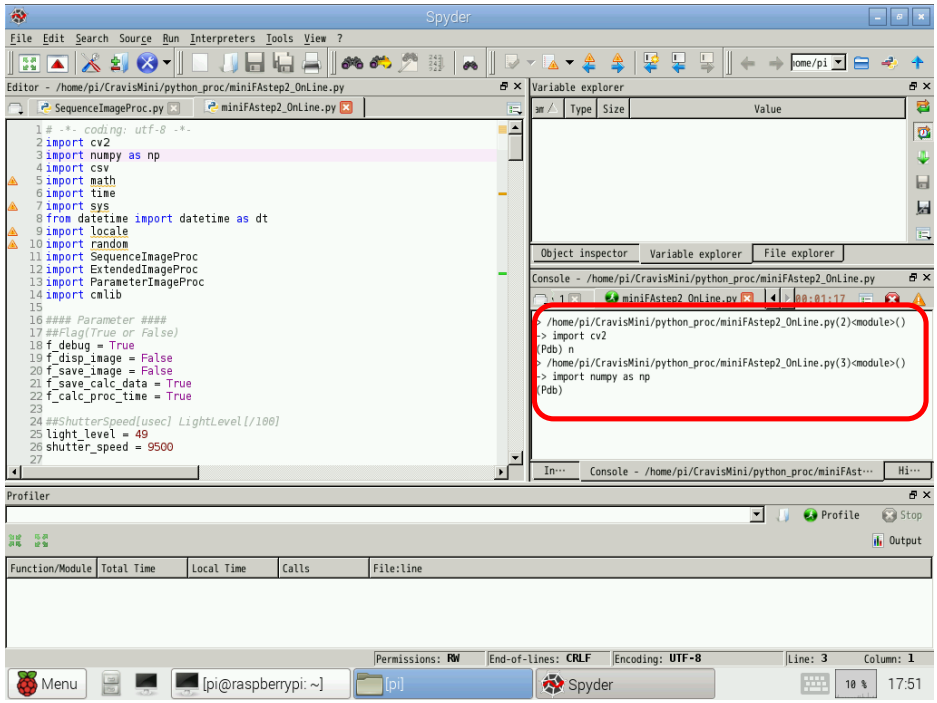
①I input with "n" behind pdb> on the Ipython console.



②I get out of the gray on an editor, and TA row goes down by one.



③I carry it out on Ipython console, and a result of the TA row is displayed.

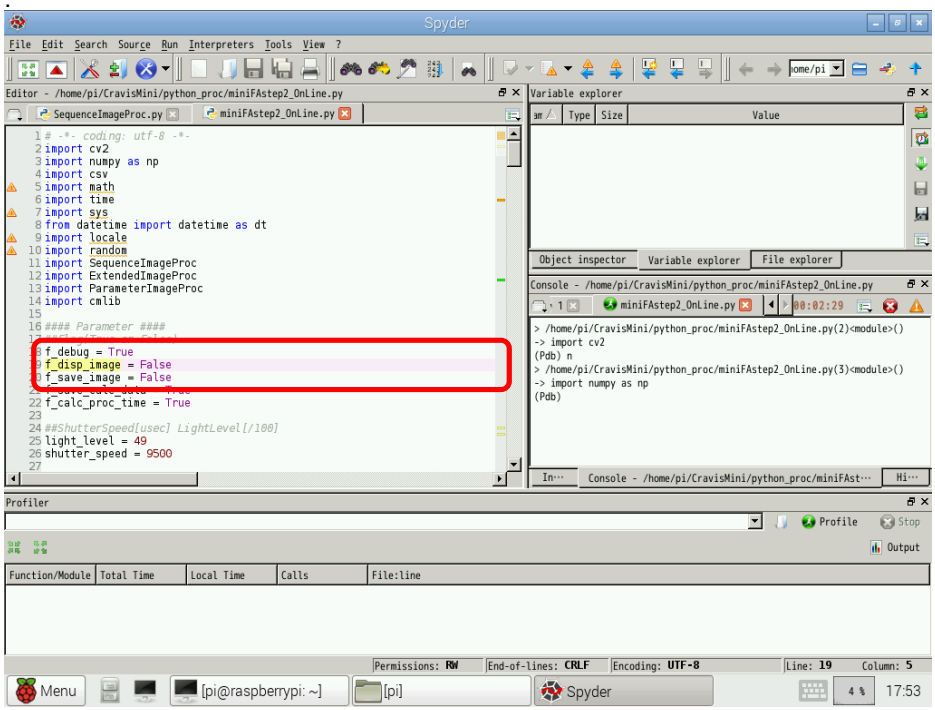


When I want to chase the practice result of each line on the script in detail, I use "the step practice".
I carry it out to a part to perform this "debugging using the breakpoint" to speak later, and to want to see in detail consecutively, and by one does how to use confirming a practice result from there.

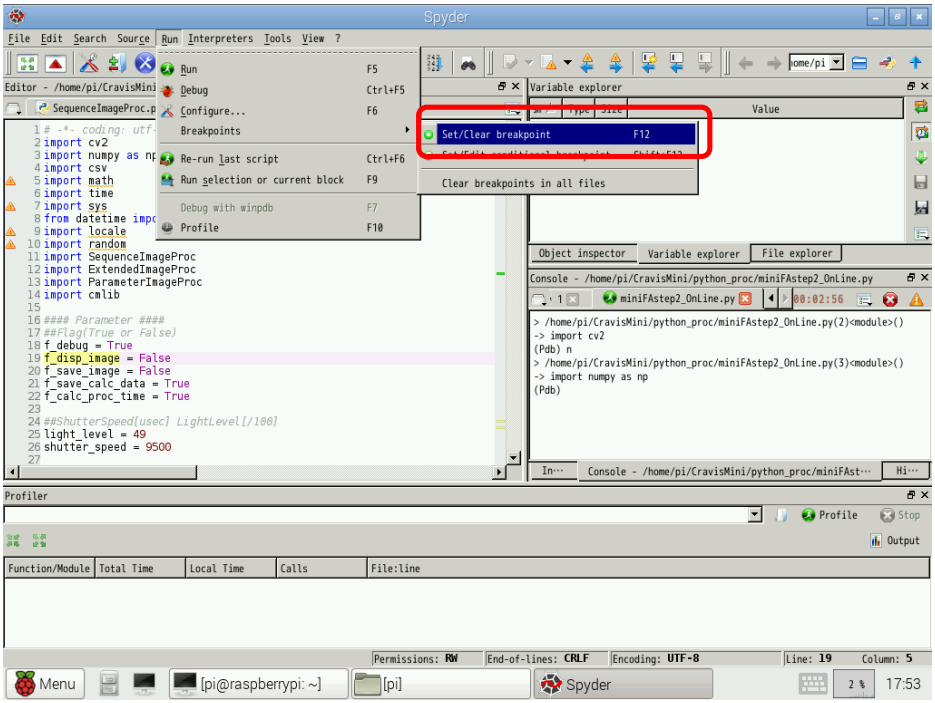
"The debugging using the breakpoint"

I call a point letting a program executing at debugging stop intentionally the break point. I can set a break point on Spyder. I can execute a break point consecutively.

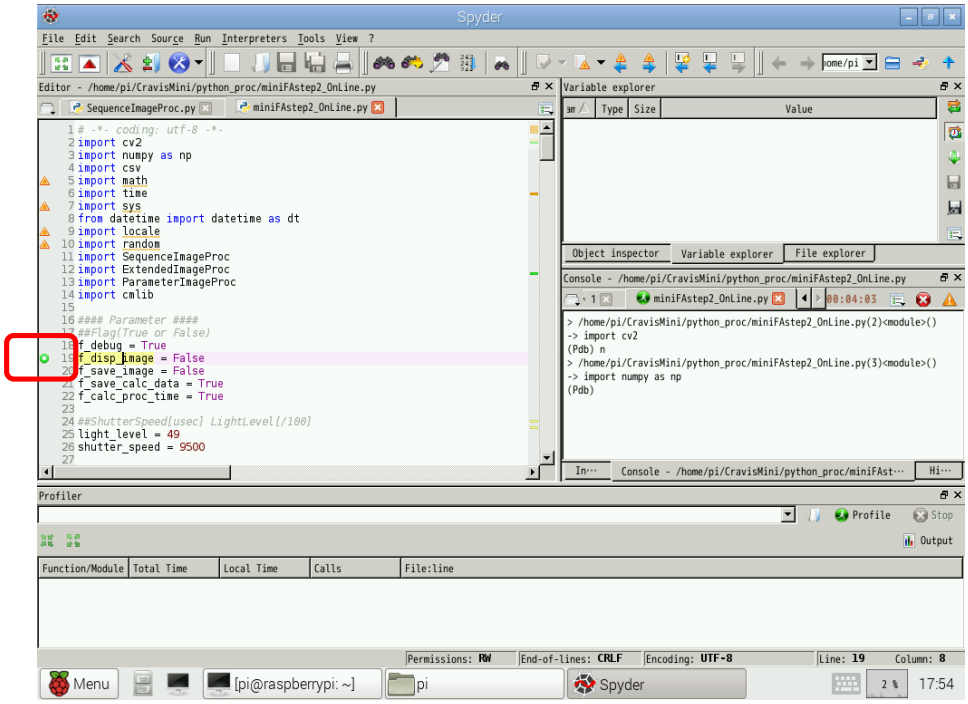
①I click a line (the line that wants to stop) that wants to set the break point in the editor.



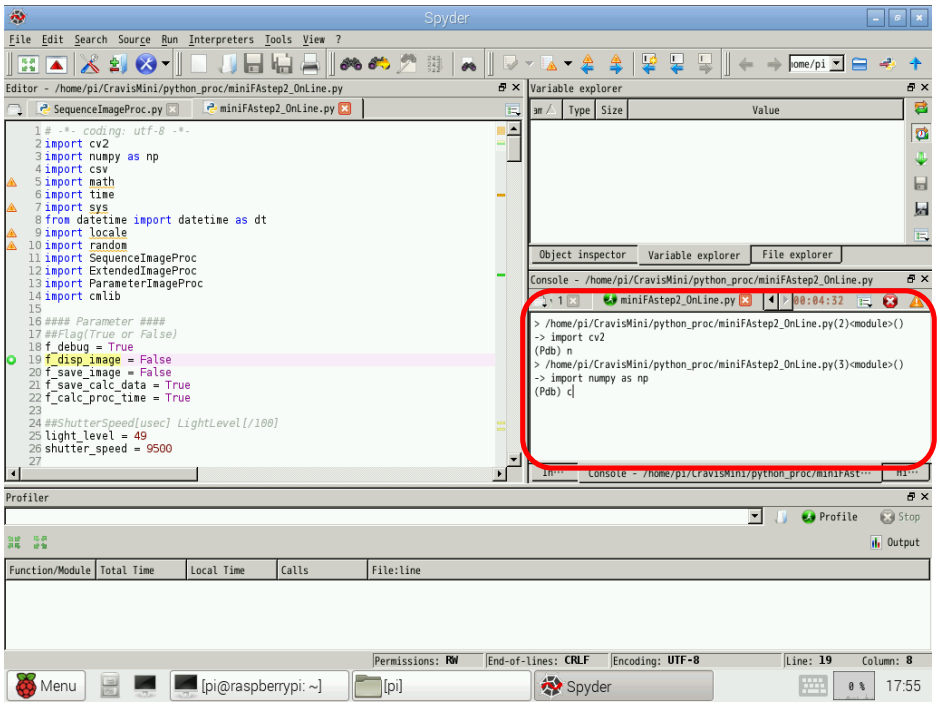
②I choose "Break points "→" Set/Clear break point" of the Run menu.
(when I want to remove a break point, I do work like to here.)



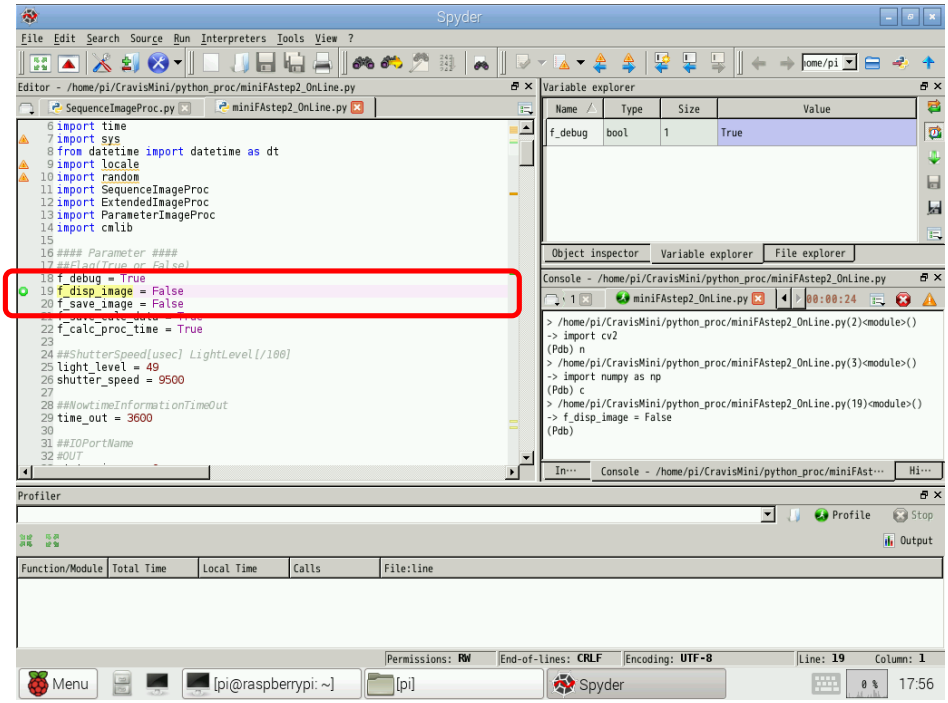
③I confirm that a break point (green point) was set on the editor.



④I start pdb and input with "c" behind pdb> on the Ipython console.

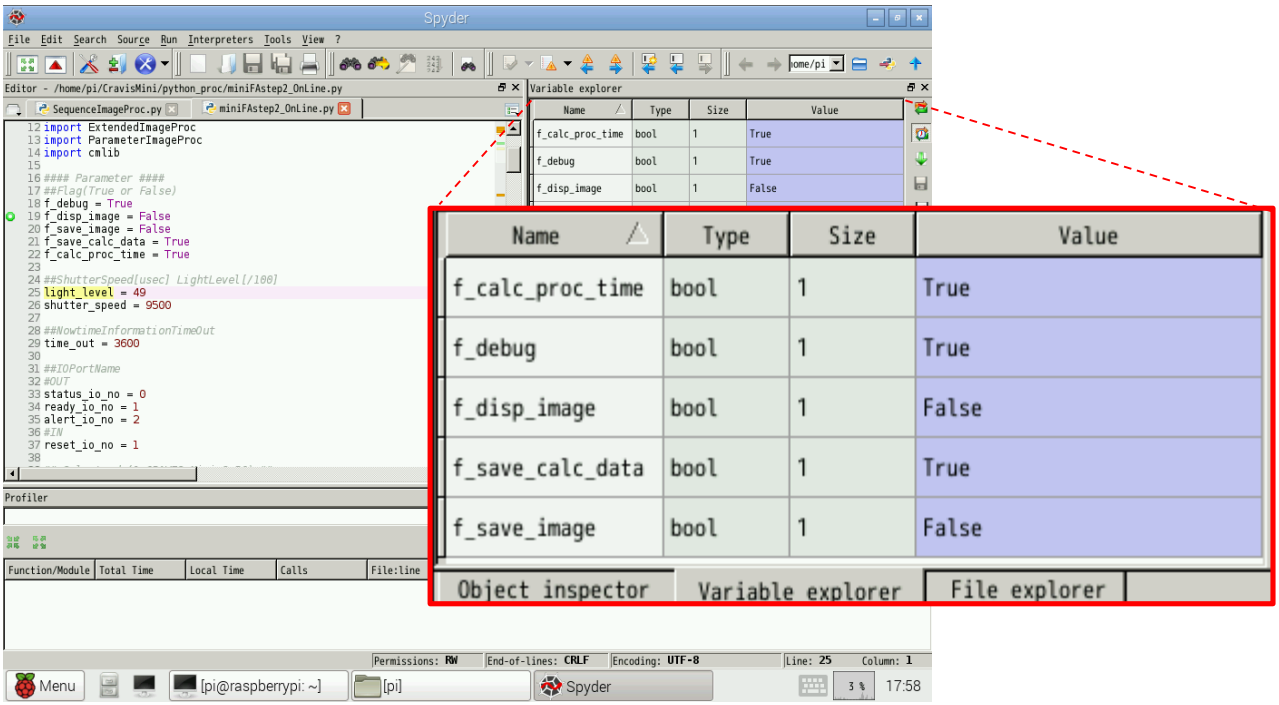


⑤I get out of the gray on an editor, and TA row moves to the breakpoint.
In addition, a practice result to there is displayed on Ipython console.

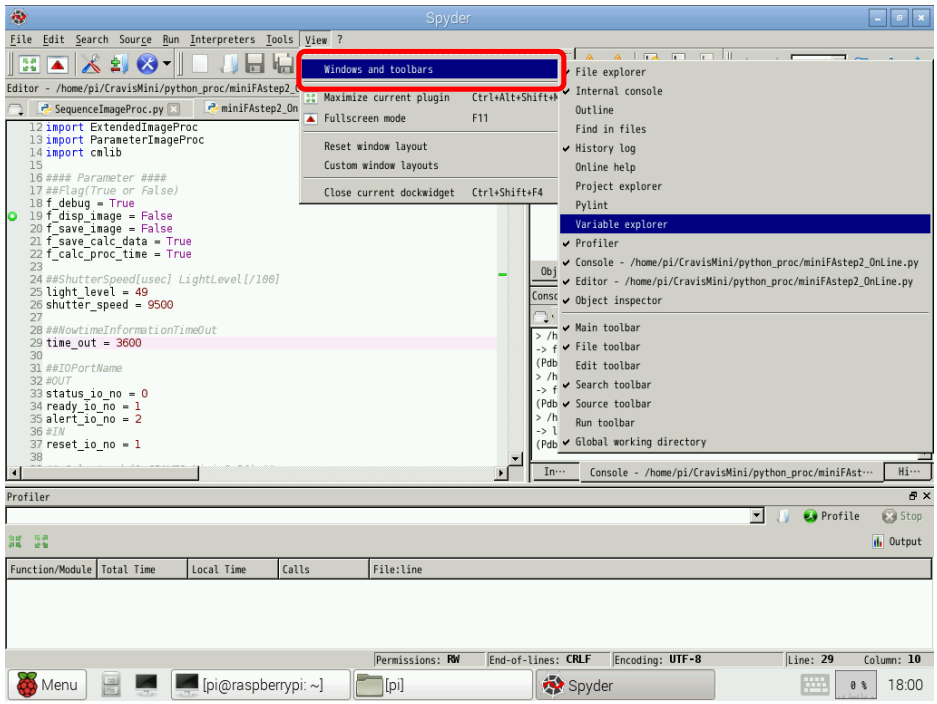


「"The usage of the Variavle explorer "」

The Variavle explorer can confirm what kind of value each variable in the script which I execute contains.



Because you may not be displayed at the time of start, you choose "Variavle explorer" among "View "→" Windows and toolbars" of the menu on this occasion, and please display the variavle explorer.



Development environment in WindowsPC

I can develop the script for the application of CRAVIS-mini in WindowsPC if I fix the development environment. Here, I explain environmental construction method and how to use about "Spyder" and "VisualStudio" which are a development environment on WindowsPC.

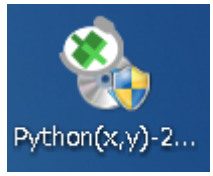
1.3 Development environment ①～ Spyder in WindowsPC

Spyder becomes the Windows version of the application that I explained in a development environment in CRAVIS-mini.

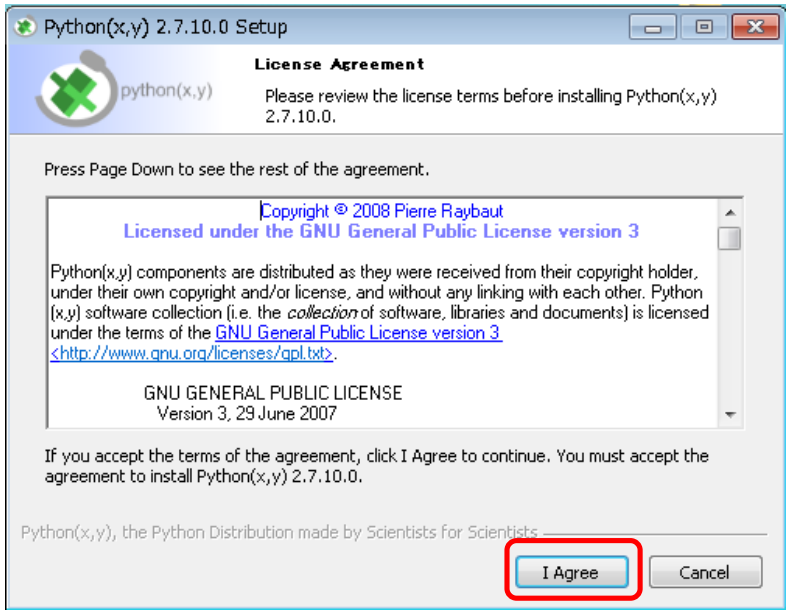
【 Installation method of Spyder 】

①I download "Python xy.exe" *.
https://python-xy.github.io/downloads.html

② I execute "Python xy.exe" and start installation.

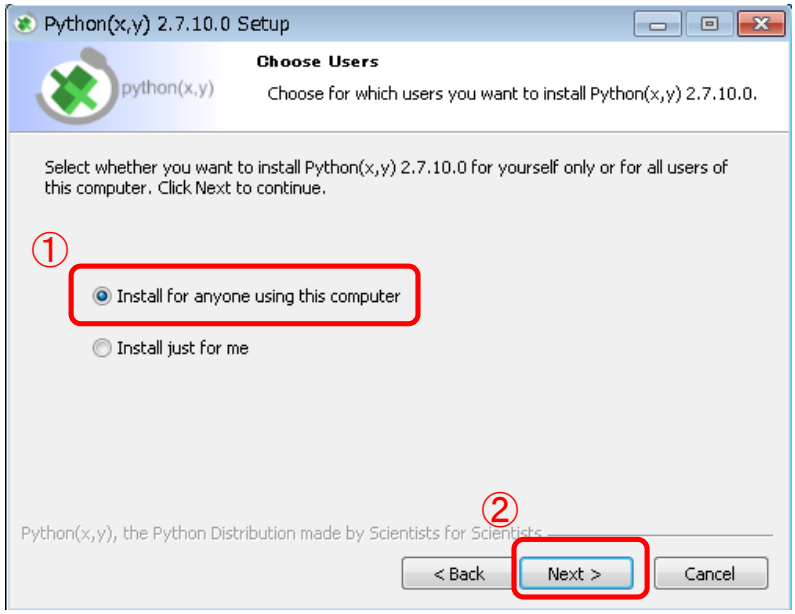


③I confirm the contents of the license and push "the I Agree button".

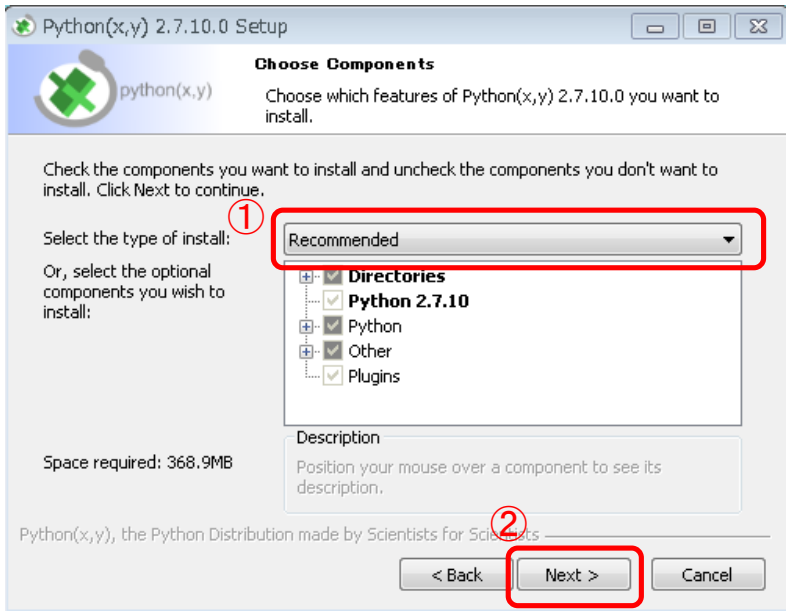


*With Python xy, it becomes the distribution that packed Python-related various free modules, tools including Spyder.
About the downloading installation of the software, please obey the instructions of each base, section.

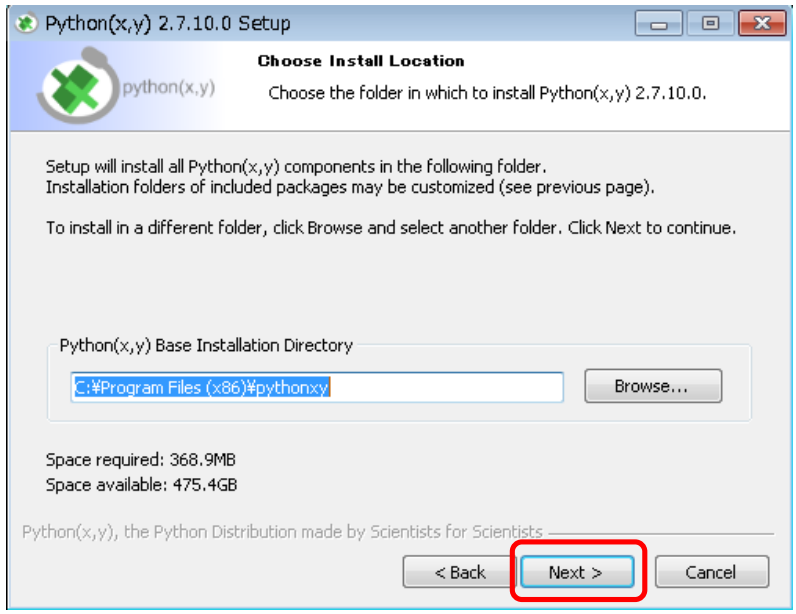
④I make the use of "Python xy" oneself-limited or show it to other users or set it.
I choose "Install for anyone using this computer" to set it with an exhibition in other users here.



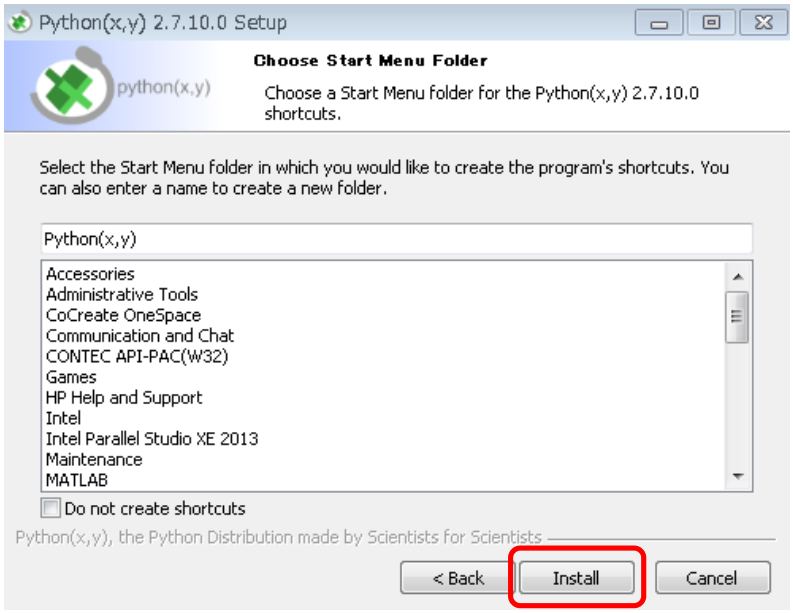
⑤I choose a module, a tool to install in "Python xy".
I choose "Recommended" and, without the module which I want to install, advance next especially.
I choose a module, a tool to install in "Python xy".
I choose "Recommended" and, without the module which I want to install, advance next especially.



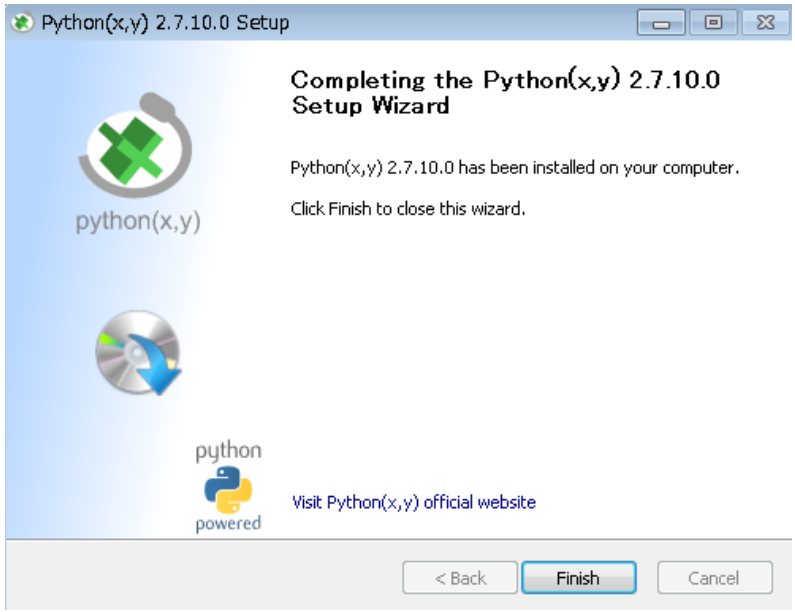
⑥I choose a directory installing "Python xy".
This just advances next without designation in particular, too.



⑦I choose a folder name making a short cut of "Python xy".
I just advance next, and this starts installation without designation in particular, too.



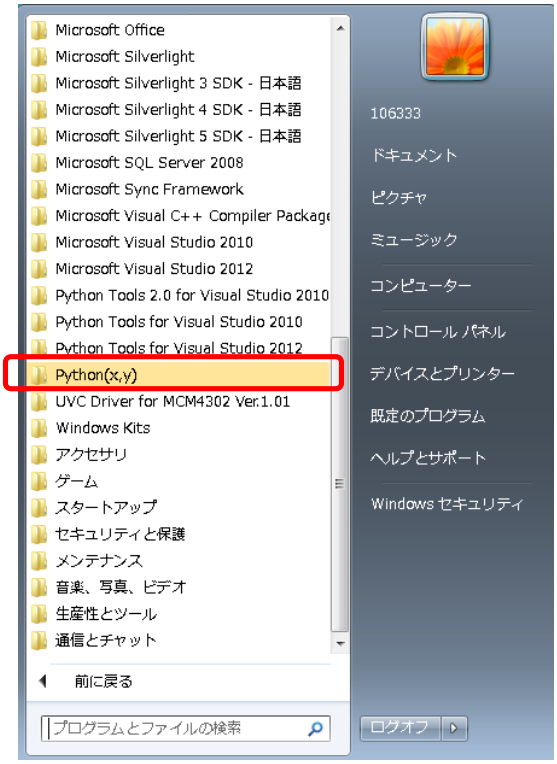
⑧I wait until installation is completed.

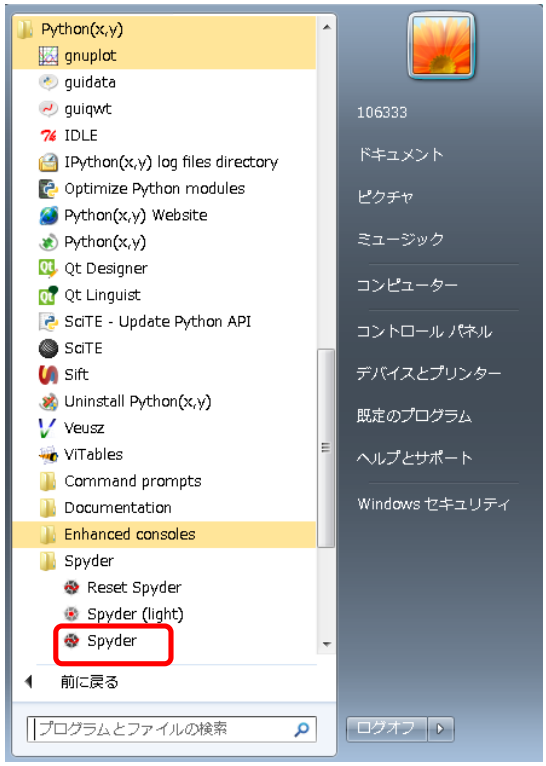


The above becomes the installation work.

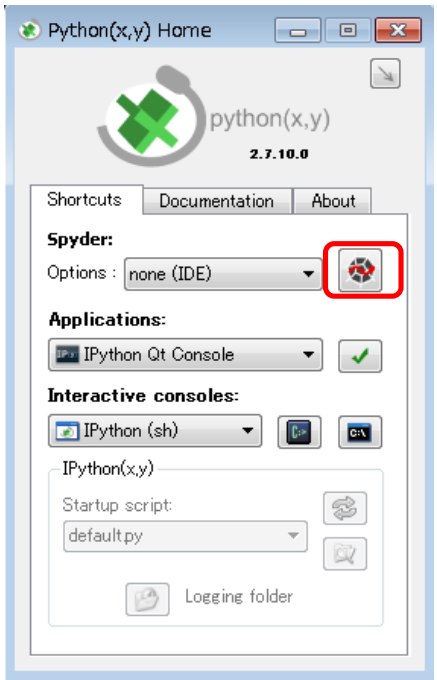
【 Start method of Spyder 】

①I click "Spyder button" in the Start Menu.





When I cannot execute "Spyder" from "Spyder button" in supplementary) Start Menu, You click "Python(x,y)" in the desktop top, and please click the icon of the red frame of the displayed window.



②Spyder starts.

【 How to open script files in Spyder 】

The basic operation becomes like spyder on CRAVIS-mini.
Please refer to your explanation.

【 Script file practice in Spyder 】

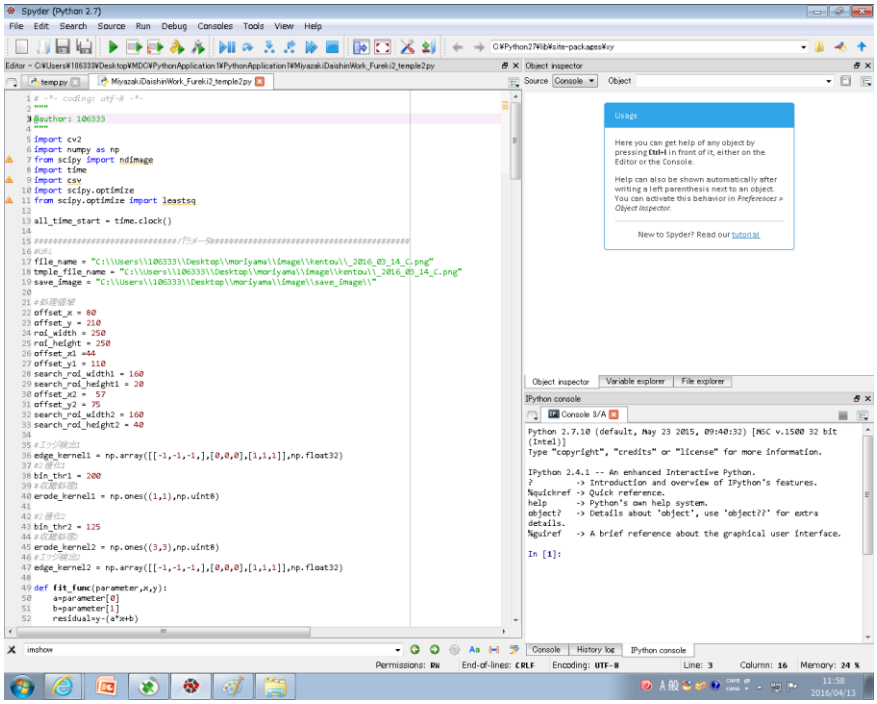
The basic operation becomes like spyder on CRAVIS-mini.
Please refer to your explanation.

【 Script file debugging method in Spyder 】

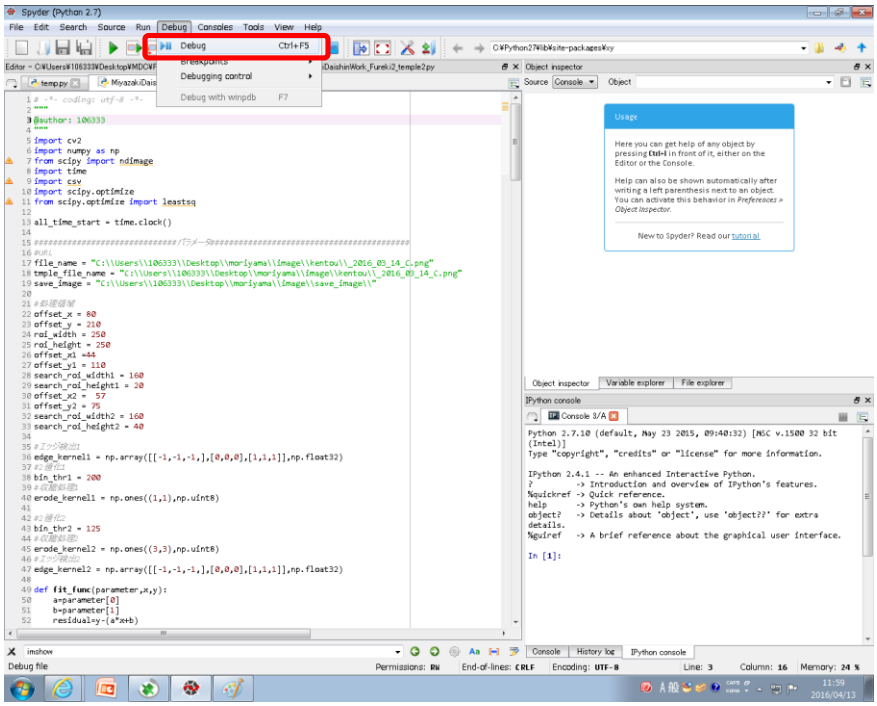
It is accompanied, and a debugging function called pdb can perform the detailed debugging of the script using pdb in Spyder of the Windows version. Here, I explain the usage of pdb and the Variavle explorer as a method of the debugging.

< Start method of pdb >

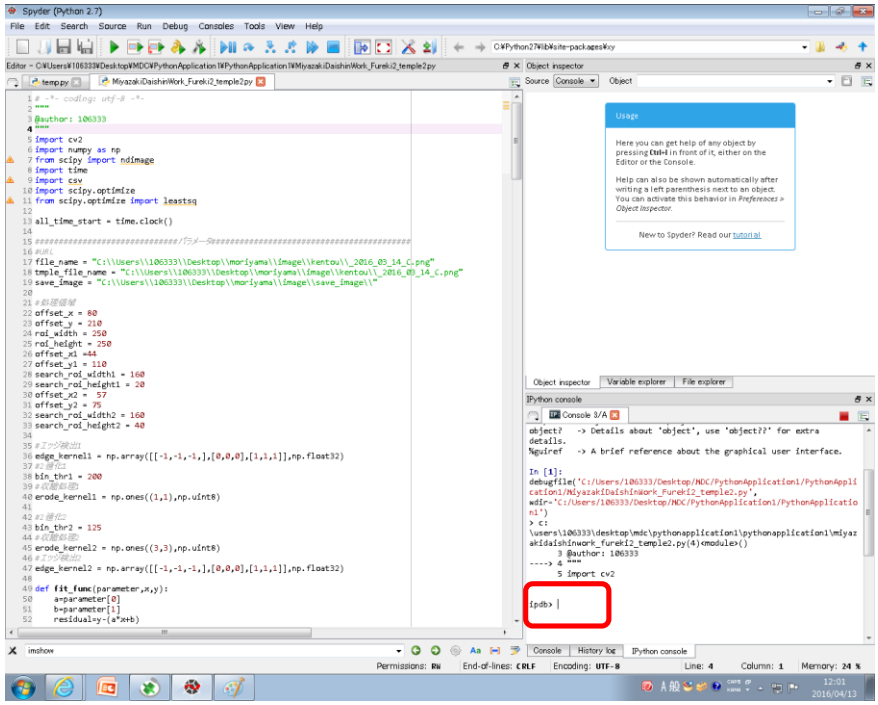
①I make the tab of the script file debugging the front.



②I push "the Debug button" of the Debug menu.



③I am displayed after Ipython console window with ipdb> in the line.



(supplement) Like a script practice method,
When I close Ipython console window, I may not carry out a script.
You push "the Open an IPython Console button" from "the Consoles button" of the menu, and please display Ipython console window on this occasion.

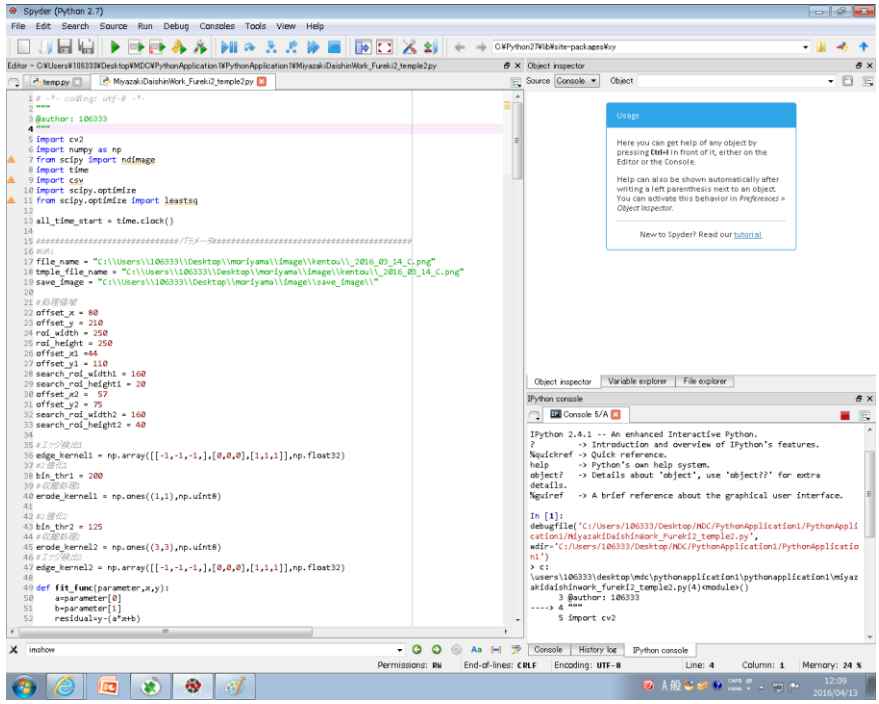
＜Debugging method using pdb＞

The debugging method of the script using pdb inputs a command on Ipython console window and performs it. Here, I explain "the step practice" that is a representative debugging method and "the debugging using the breakpoint".

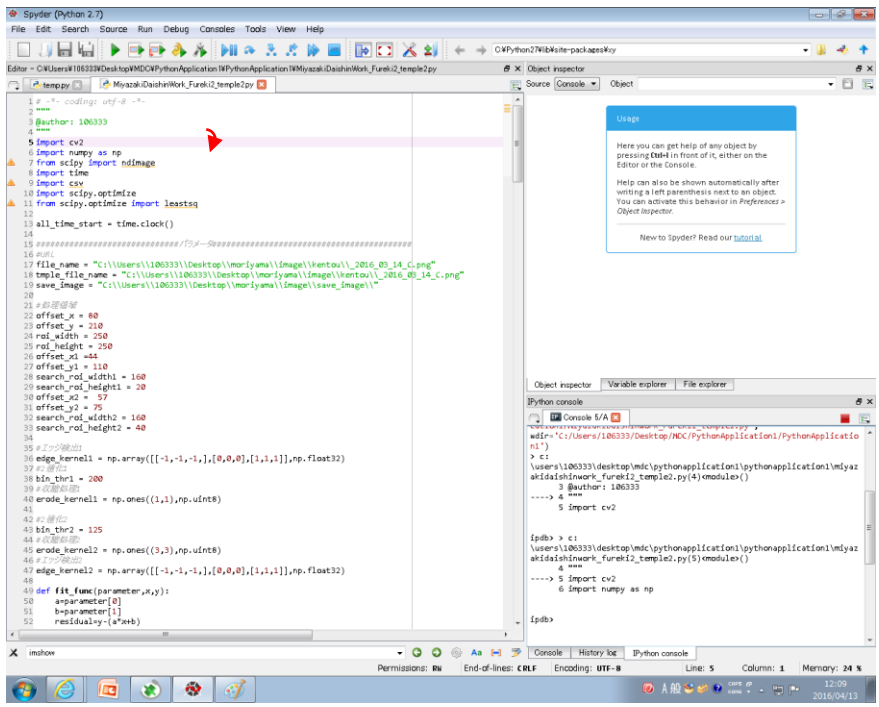
「Step practice」

The step practice is the method that I debug while execute the code of the script by one line.

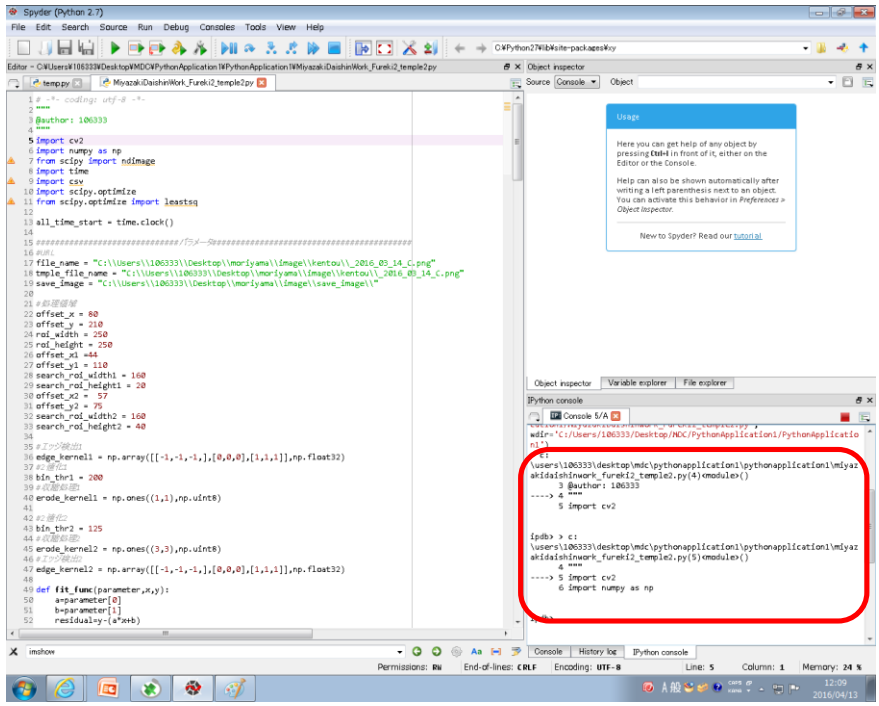
①I start pdbs and push the "ctrl" + "F10" button.



②I get out of the gray on an editor, and TA row falls.



③I execute on Ipython console, and a result of the TA row is displayed.

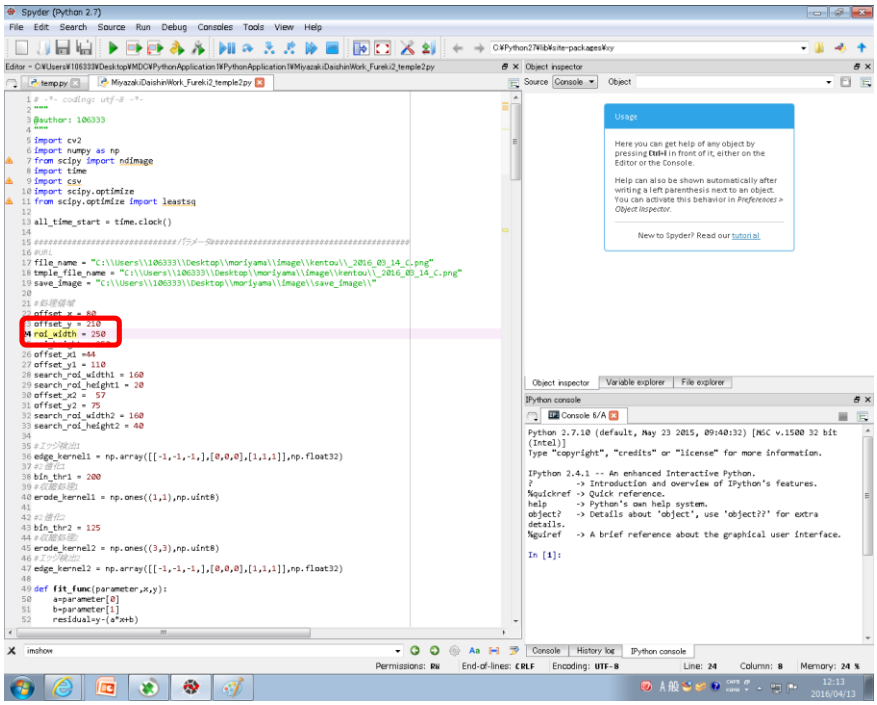


When I want to chase the practice result of each line on the script in detail, I use "the step practice".
I execute a part to perform this "debugging using the breakpoint", and to want to see in detail consecutively, and by one does how to use confirming a practice result from there.

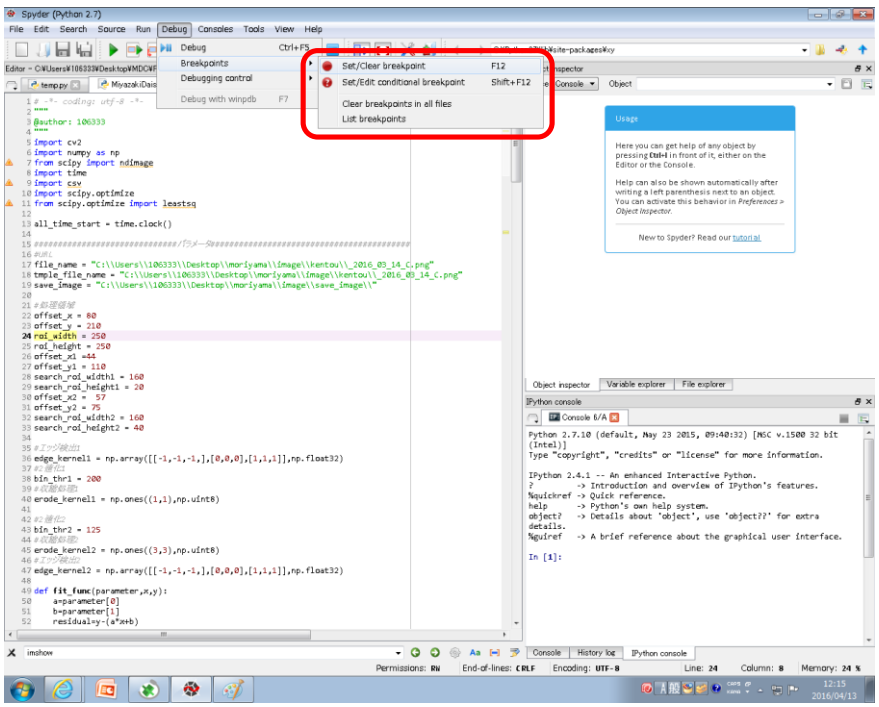
「 Debugging using the breakpoint 」

I call a point letting a program executing at debugging stop intentionally the break point. I can set a break point on Spyder. I execute a break point consecutively.

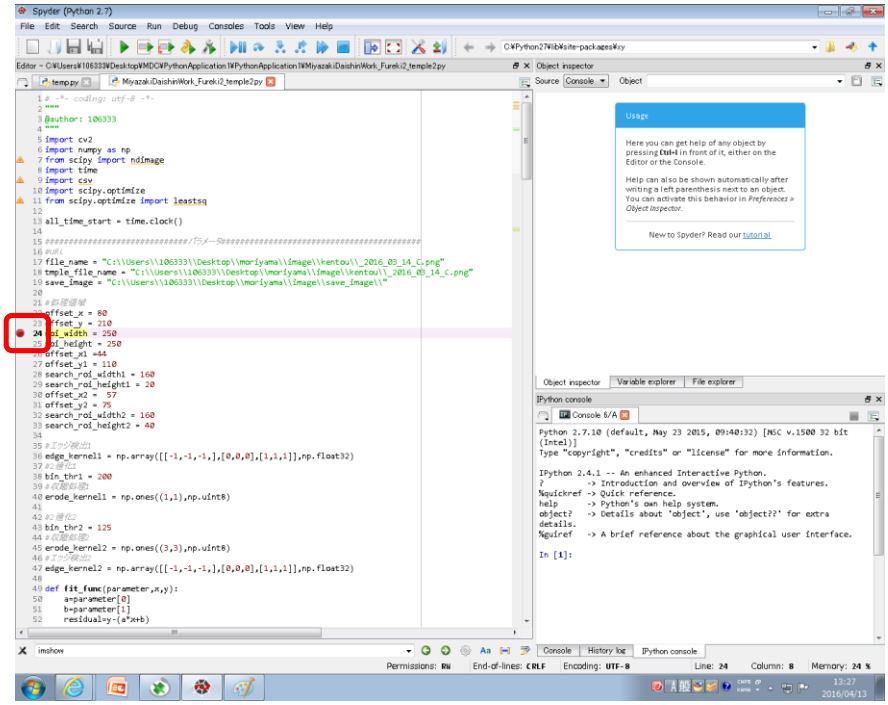
①I start pdb and click a line (the line that wants to stop) that wants to set the break point in the editor.



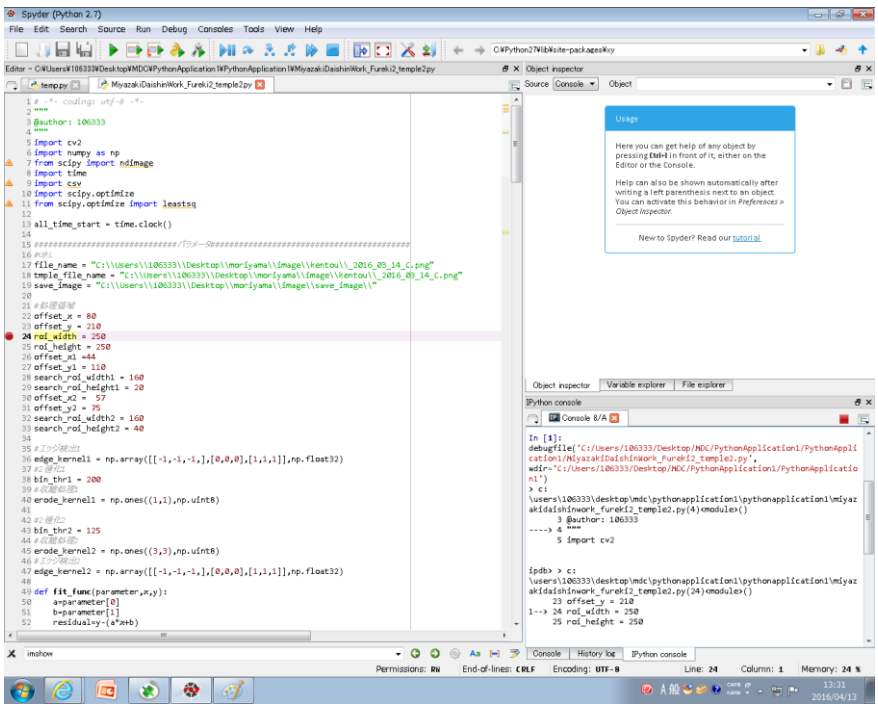
②I choose "Break points "→" Set/Clear break point" of the Debug menu.
(when I want to remove a break point, I do work like to here.)



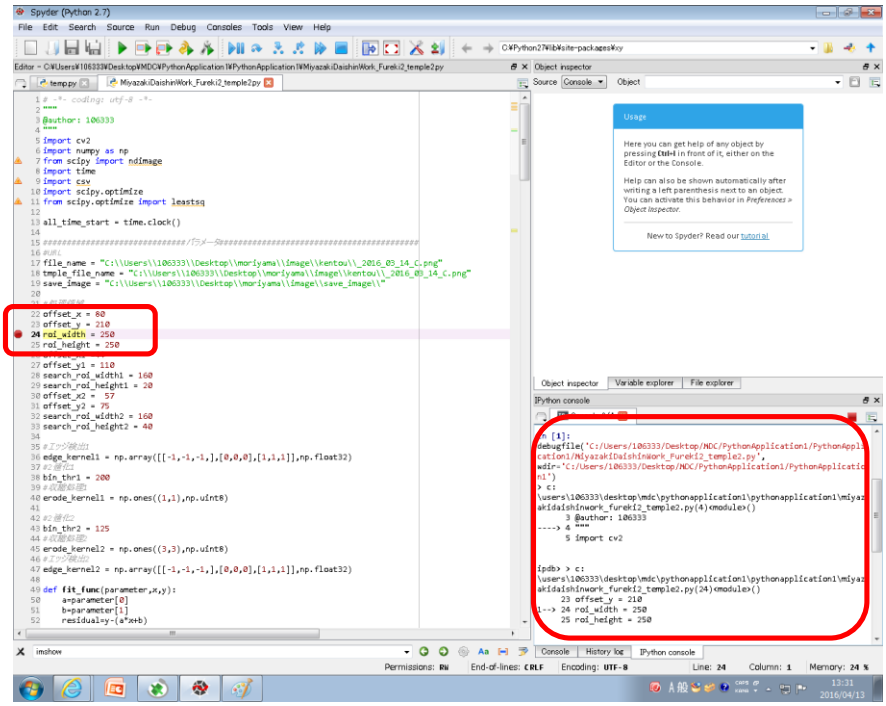
③I confirm that a break point (failing mark) was set on the editor.



④I push the "ctrl" + "F10" button.

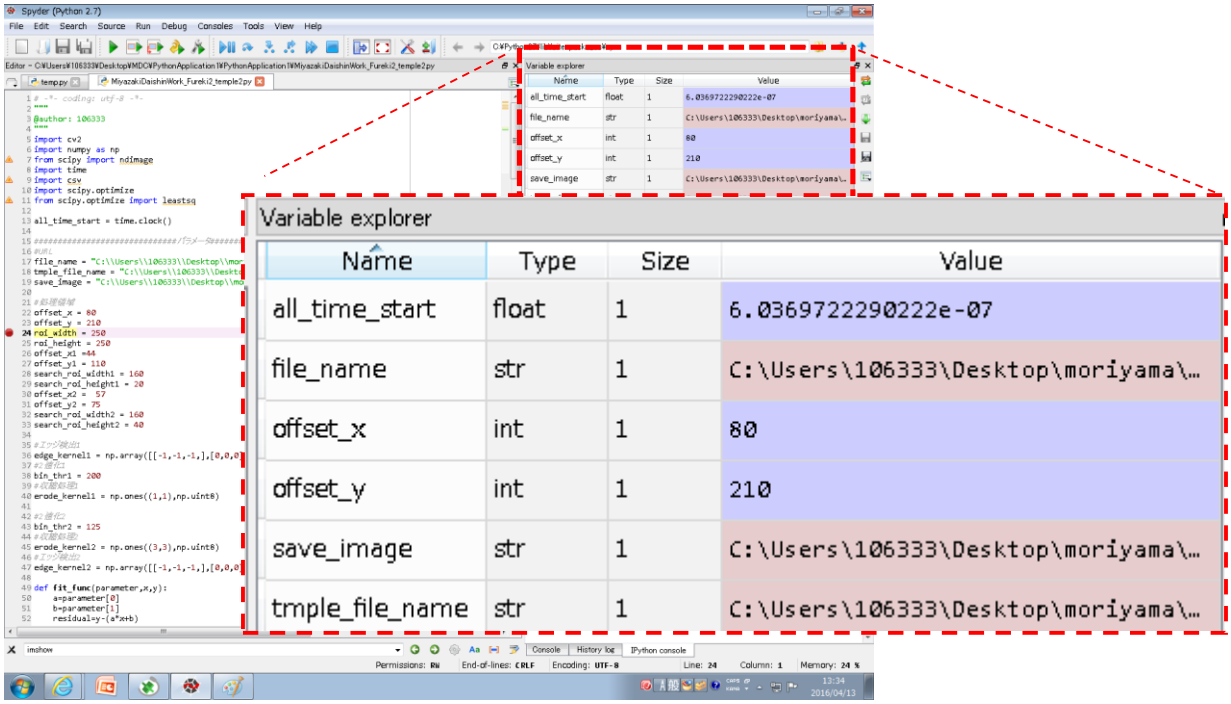


⑤I get out of the gray on an editor, and TA row moves to the breakpoint.
In addition, a practice result to there is displayed on Ipython console.

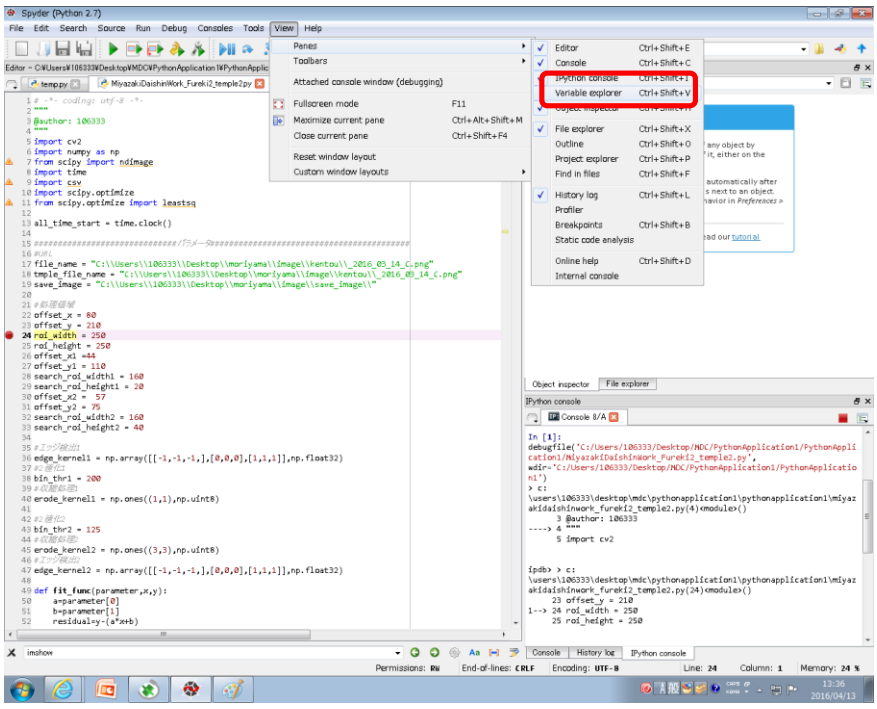


「"The usage of the Variavle explorer "」

The Variavle explorer can confirm what kind of value each variable in the script which I execute contains.



Because you may not be displayed at the time of start, you choose "Variavle explorer" among "View "→" Panes" of the menu on this occasion, and please display the variavle explorer.

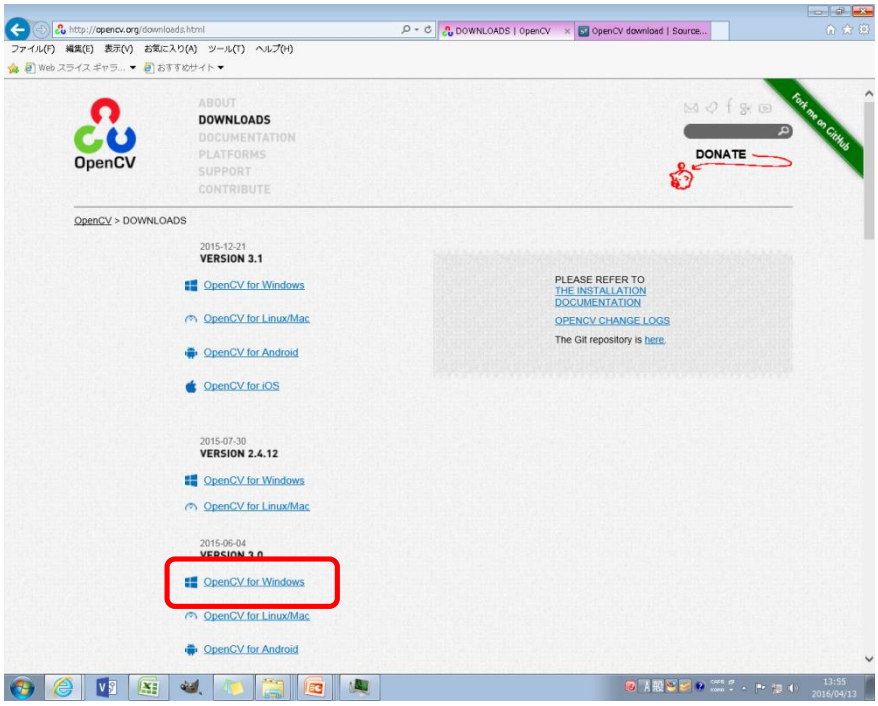


1.3.2 OpenCV

OepnCV is a library forming the basis of the image processing in the application of CRAVIS-mini.
When I use this library on Windows, it is necessary to confirm ver of the CRAVS-mini body, and to install a library of same ver.
I install OpenCV3.0 in an example this time and work.

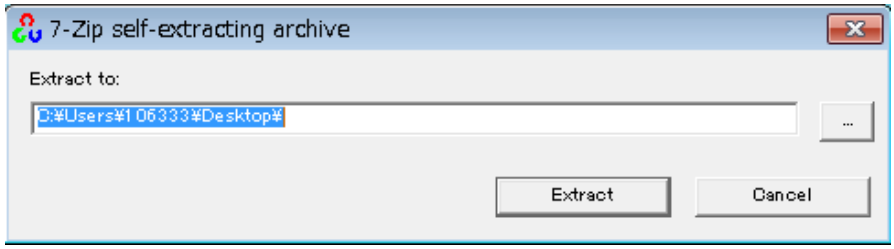
【 Installation method of OpenCV 】

①I download "OpenCV-3.0.0.exe" *.
http://opencv.org/downloads.html

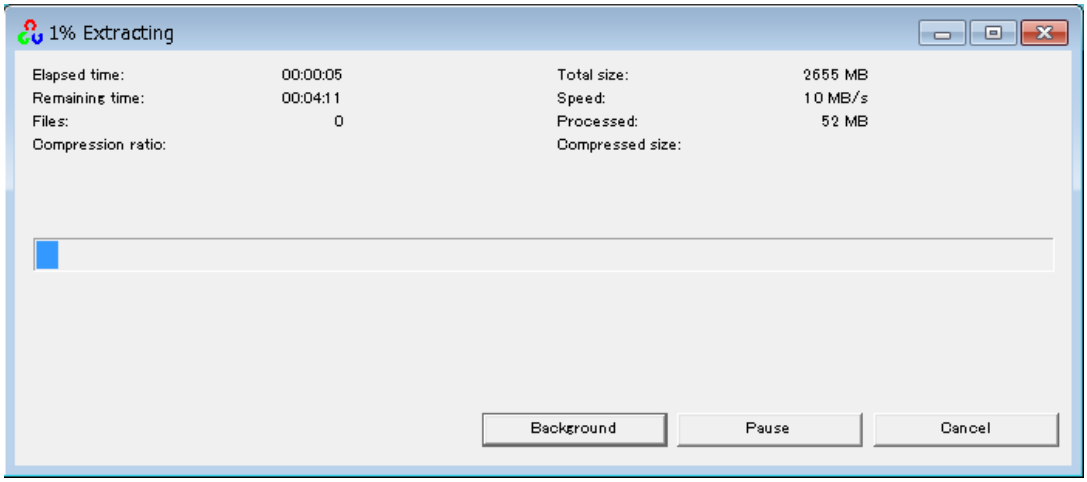


② I execute "OpenCV-3.0.0.exe" and start installation.

③I choose thawing of the file.



④I wait until a file is defrosted.



⑤The ..opencvbuildpython2.7x86cv2.pyd file in the defrosted file
C: Please copy it in Python27Libsite-packages.

*Please copy .pyd in the folder of x86 with the 64bit machine.

1.4 Development environment ②～ VisualStudio in WindowsPC

1.4.1 Python tools for VisualStudio

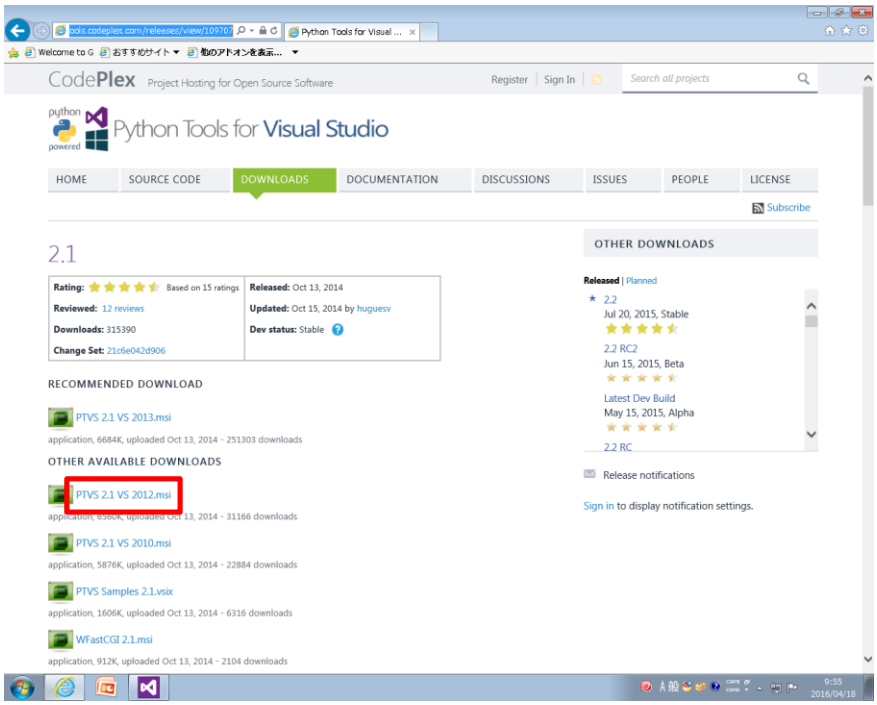
Python tools for VisualStudio is the tool which becomes able to develop application of Python using software development environment VisualStudio of MicroSoft company.
By the explanation of this manual, I explain "Python tools for VisualStudio2.1" for the cause in "VisualStudio2012".

【 Installation method of Python tools for Visual Studio 】

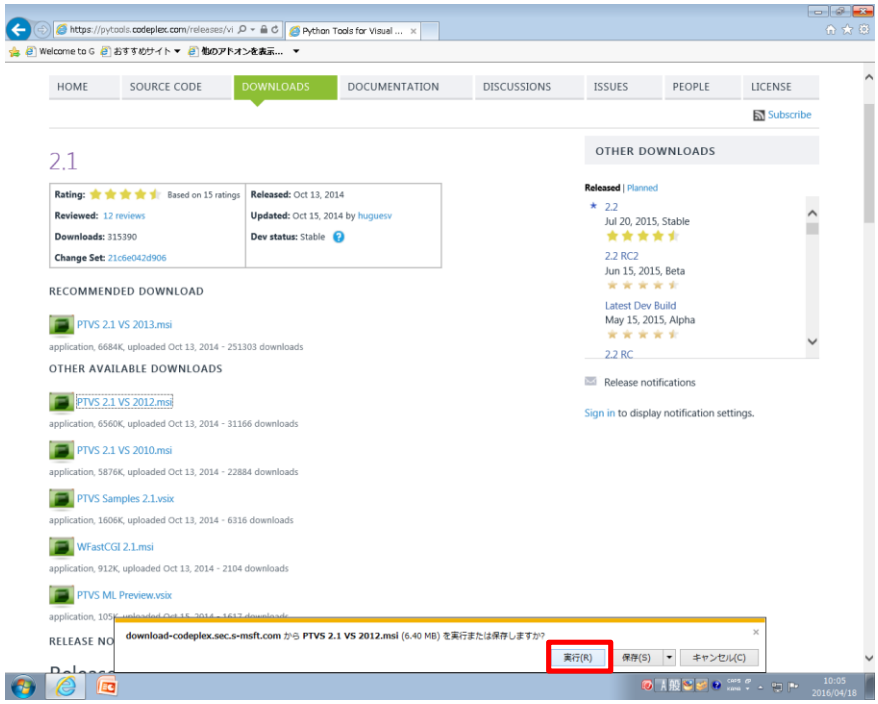
①The following preparations are necessary to use Python tools for VisualStudio.
- Installation of Python xy
- Installation of OpenCV3.0
- Installation of VisualStudio2013
- The work below installation this of versions more than .NetFramework4.0 stimulates that the preparations mentioned above were completed on a premise.

②I access the following URL.
https://pytools.codeplex.com/releases/view/109707

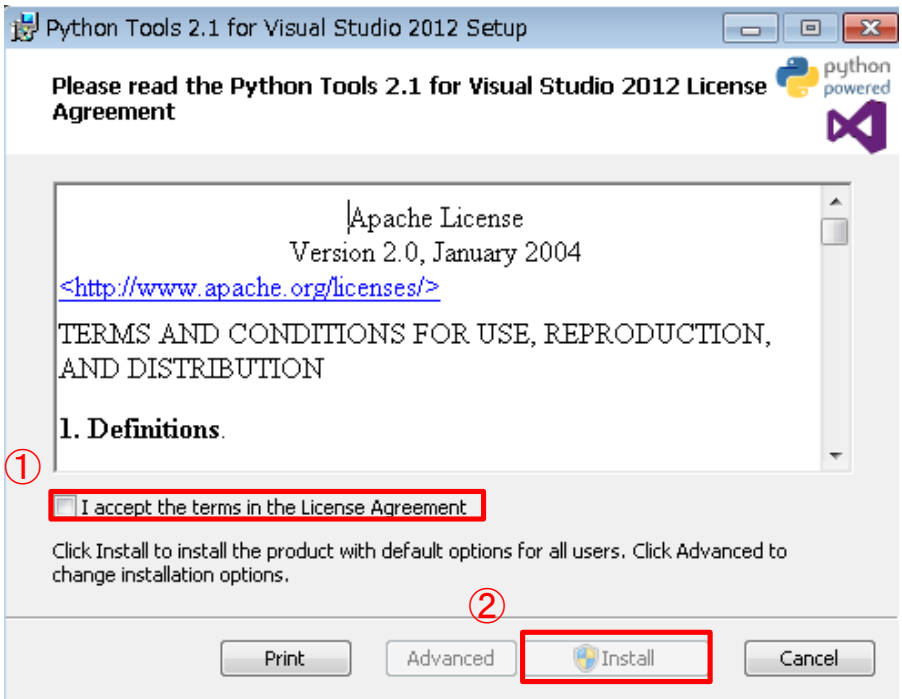
③I click "PTVS2.1 VS2012.msi" in the HP.



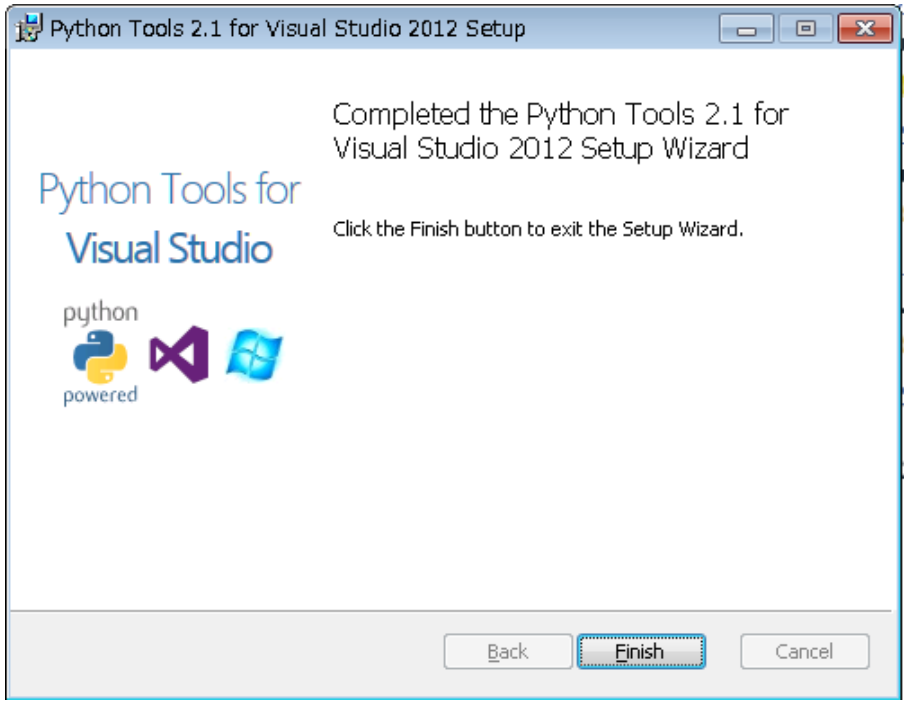
④I push "the practice button" to carry out installation of PTVS2.1 VS2013.msi.



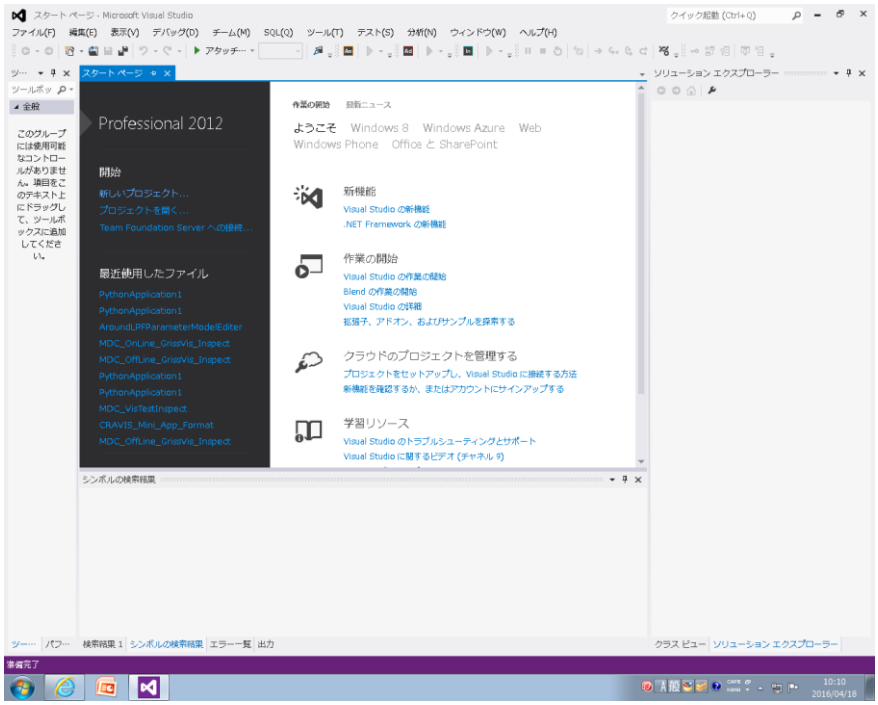
⑤I confirm the contents of the license and check it in check box and push "the Install button".



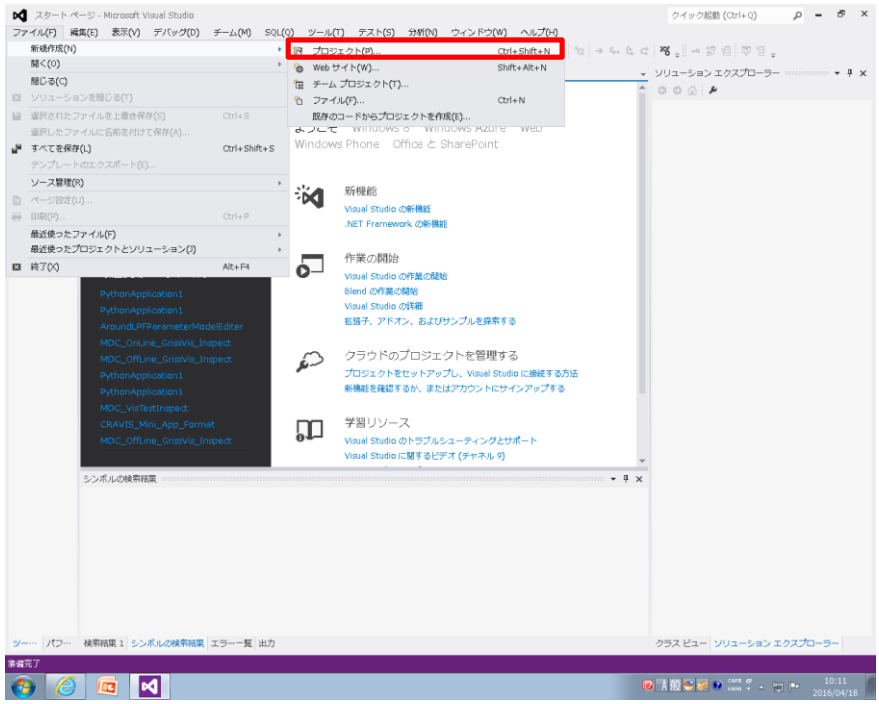
⑥I wait until installation is completed.



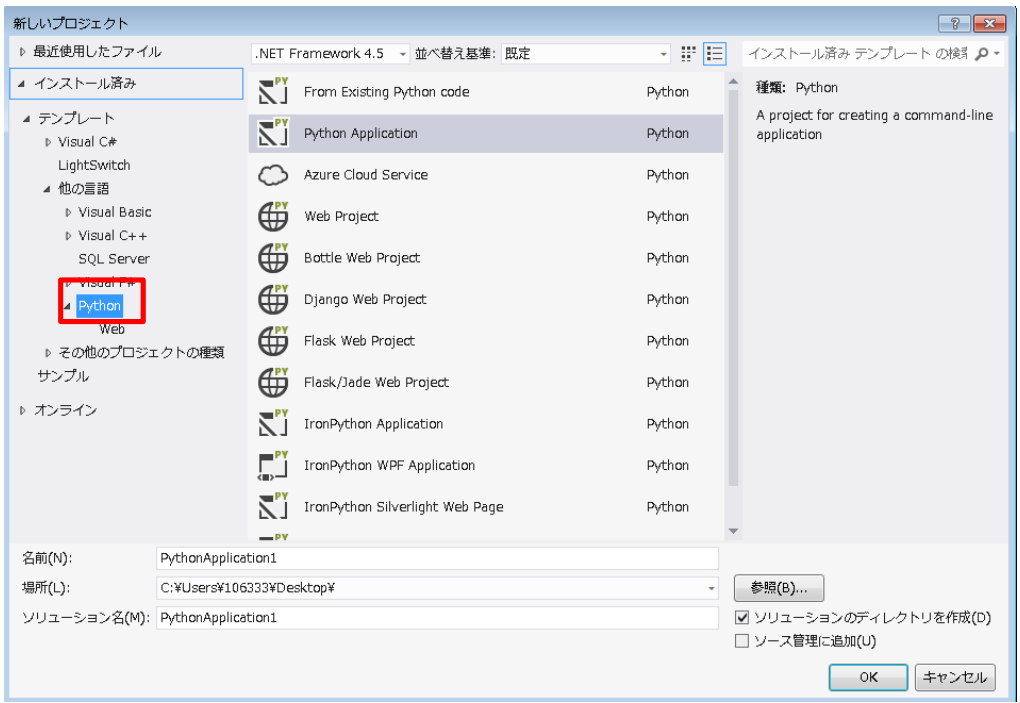
⑦I stand and put up "Visual Studio2012".



⑧I choose "file "→" new making "→" project".

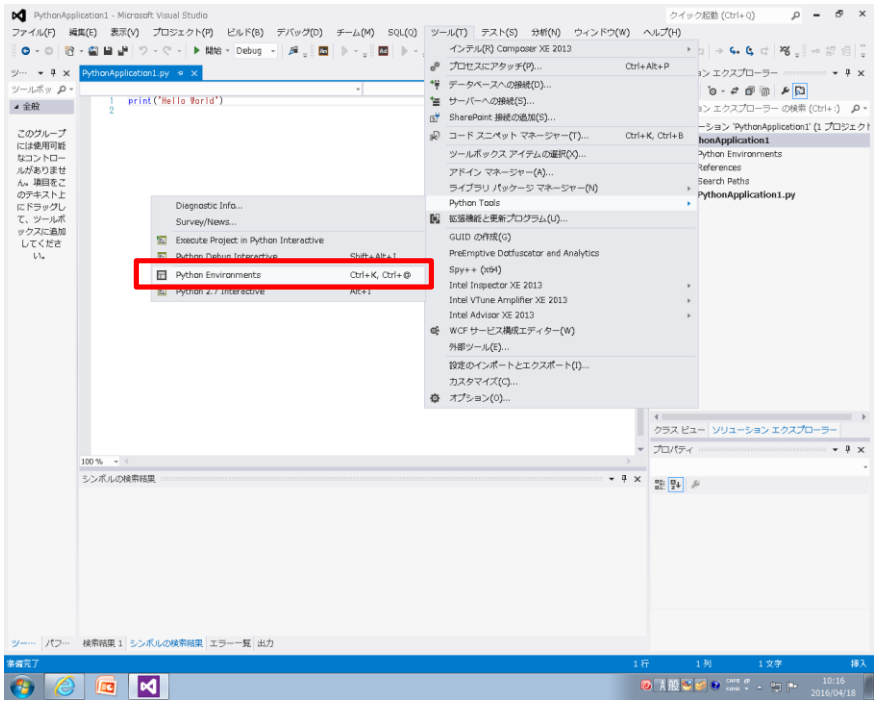


⑨If a template comes to be able to choose "Python", it is installation work completion.

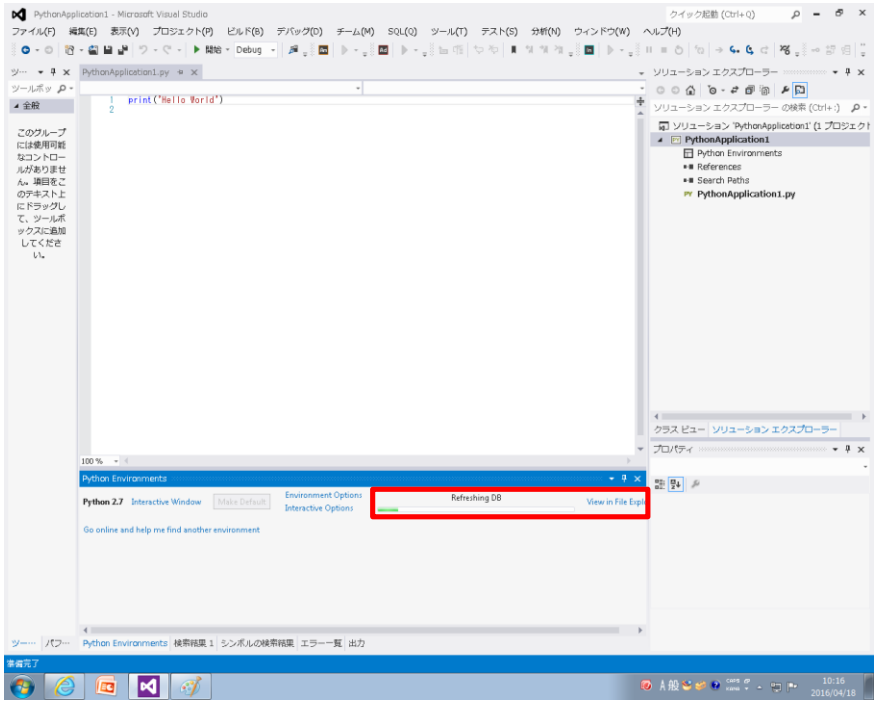


(supplement) It is accompanied setup method VisualStudio to use IntelliSense of python in VisualStudio the IntelliSense function that I display the list of of the member of the function of the development environment using, a parameter hint, candidate input and support.
I explain the setup method that I can use this IntelliSense for in python.

①I choose "tool "→" Python Tools "→" Python Environments".



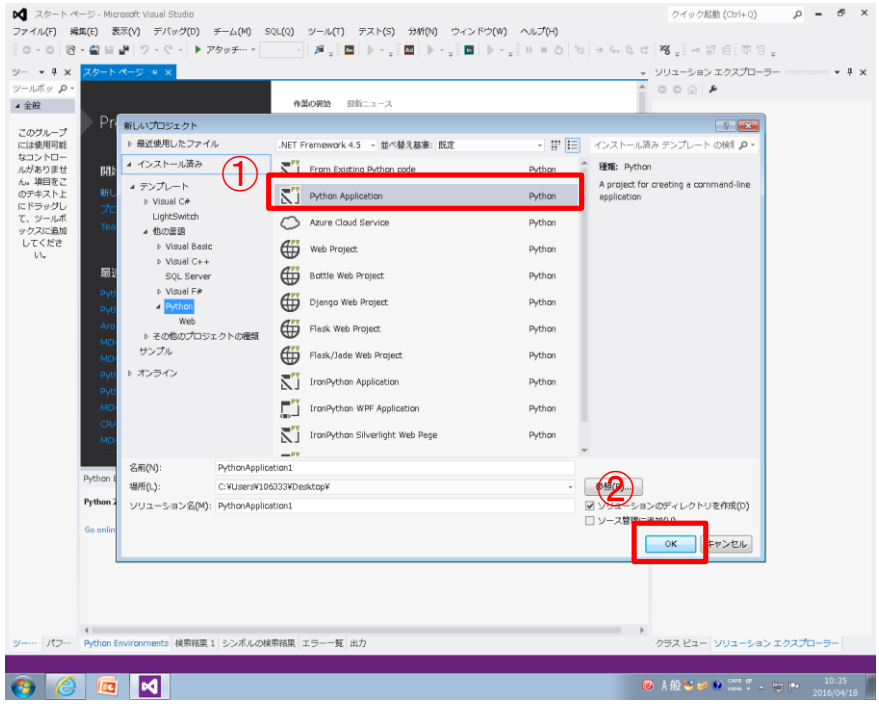
②I wait for a progress bar of "RefreshingDB" of "Python Environments" to become 100%.



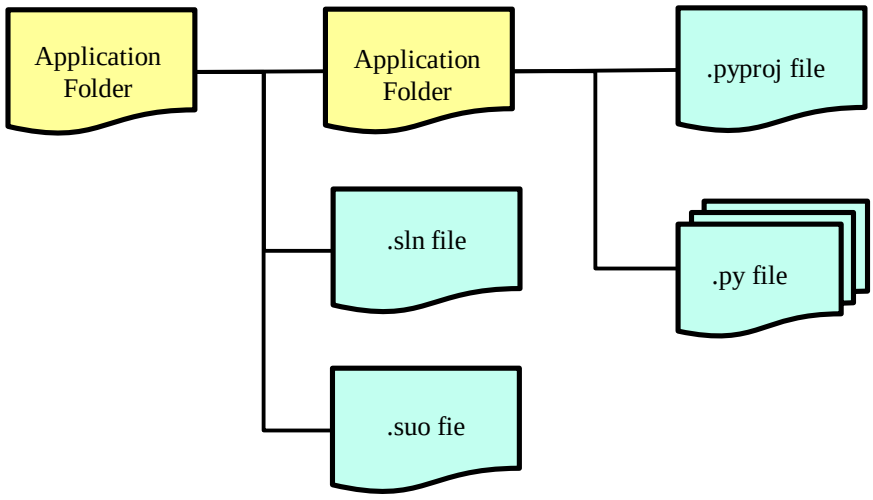
【 About file constitution of the application of Visual Studio 】

When I build application of Python in VisualStudio, it is necessary to understand file constitution of the application on VisualStudio. Here, I explain file constitution of the application on VisualStudio.

①I make new application of pyhon in VisualStudio.



②The constitution of the contents is as follows when I confirm a folder of the python application that I made.



A file becoming important in this folder is two of follows.
It is a file managing the application in one in ".sln file" ... VisualStudio.
When I want to edit application, I exctute.
It becomes the script file of each Python constituting ".py file" ... application.
By the application of VisualStudio, I can give one application plural .py files.

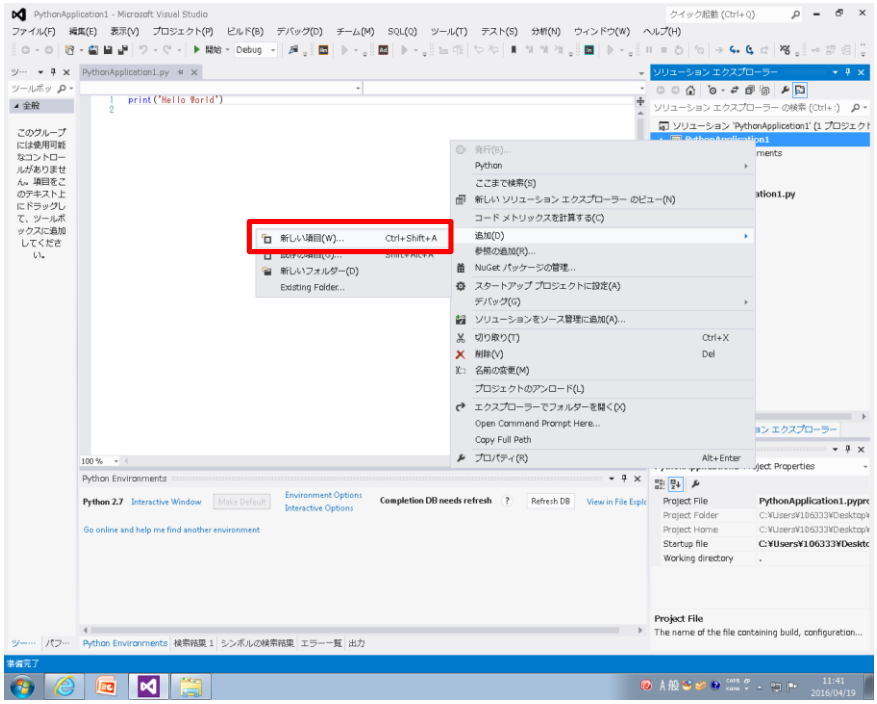
The application editing of Python on VisualStudio starts ".sln file" and edites ".py file".

【 Method to make a script file for application of Visual Studio newly 】

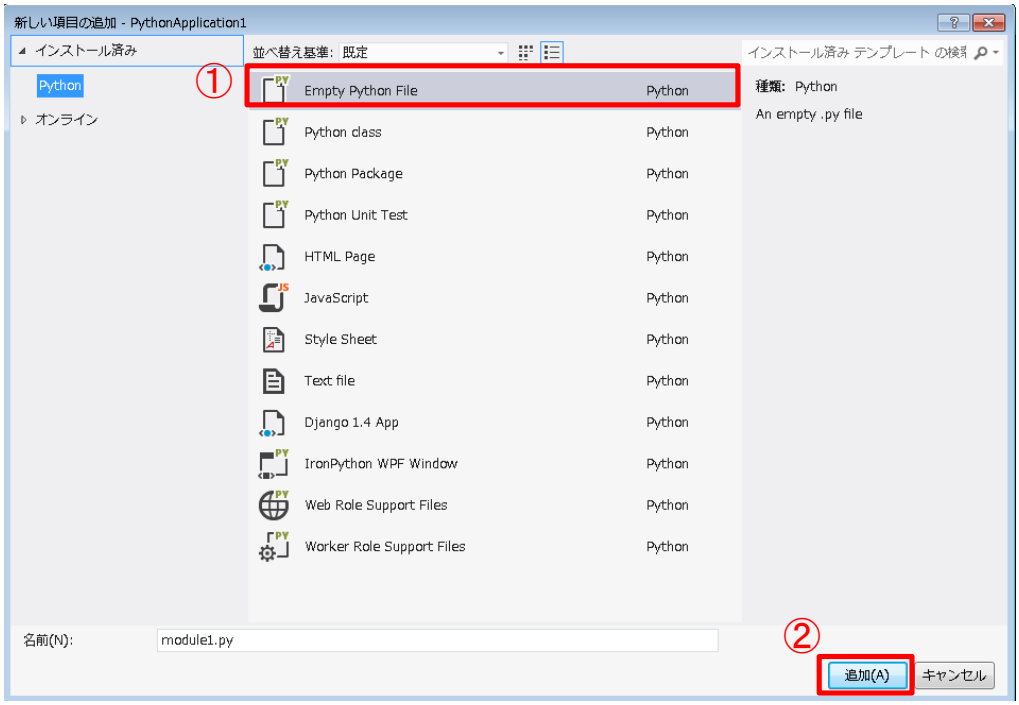
I explain a method to make .py file for application on VisualStudio newly.

①I open up an application file (.sln file) of VisualStudio.

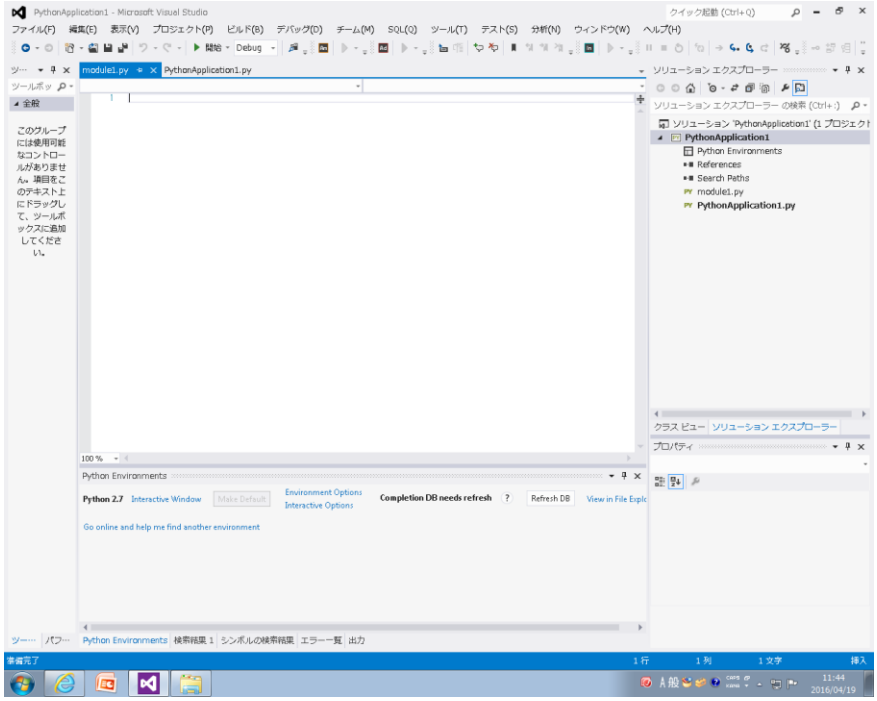
②Of the solution Explorer PY I click the right button on an icon and choose "additional "→" new item".



③I choose "Empty Python File" and input any file name and push "the Add button".



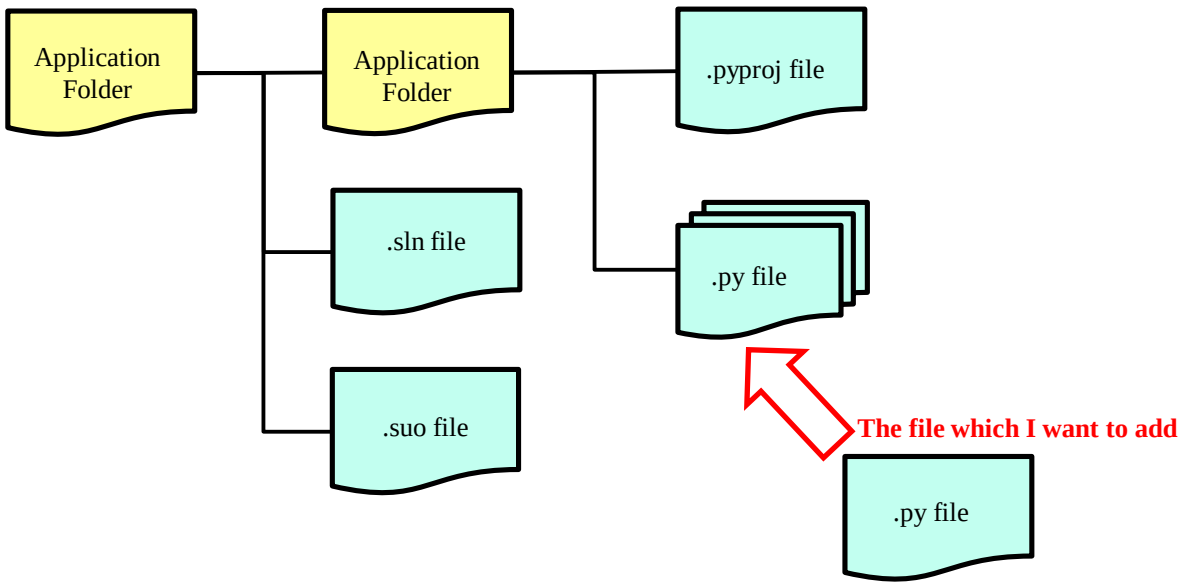
④New py file is added.



【 Method to add an existing script file to application of Visual Studio 】

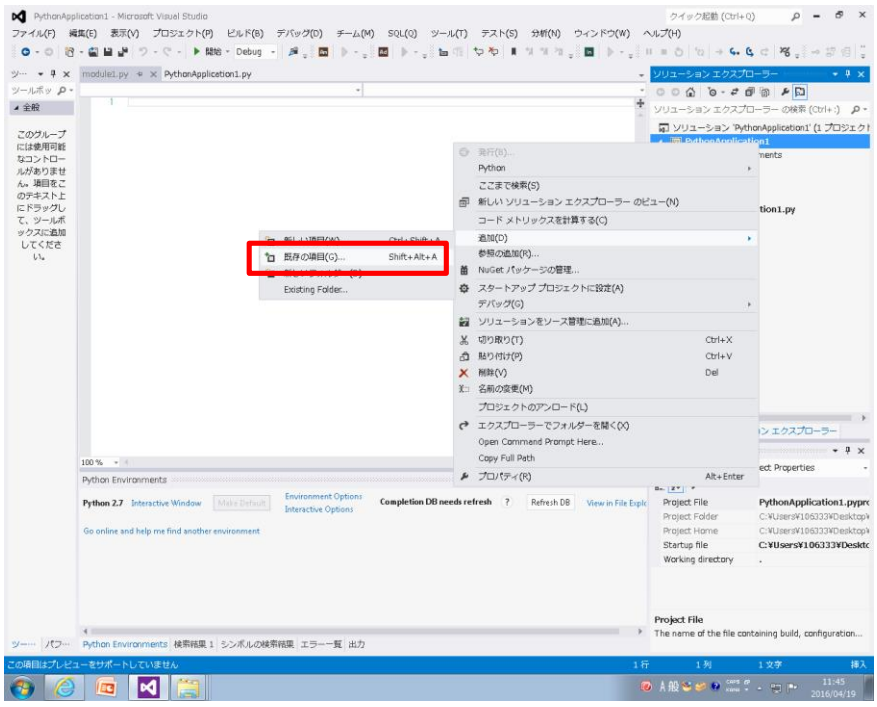
I explain a method to add existing .py file to on VisualStudio.

①I locate the .py file which I want to add in the folder of the following application.

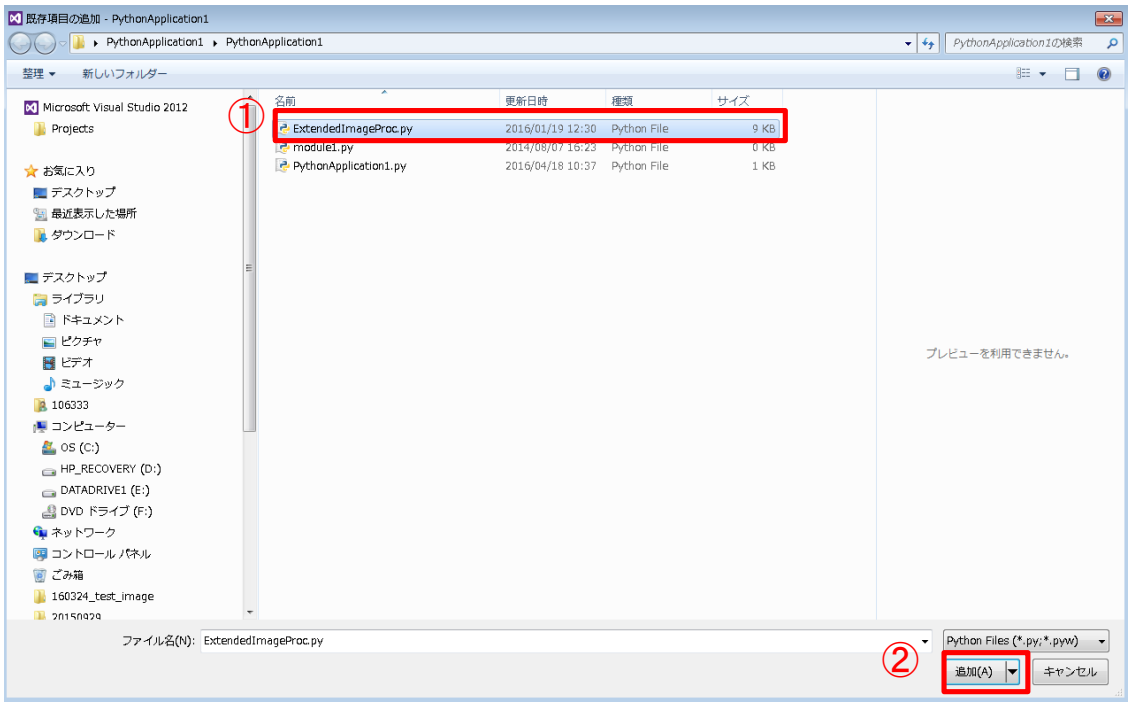


②I open up an application file (.sln file) of VisualStudio.

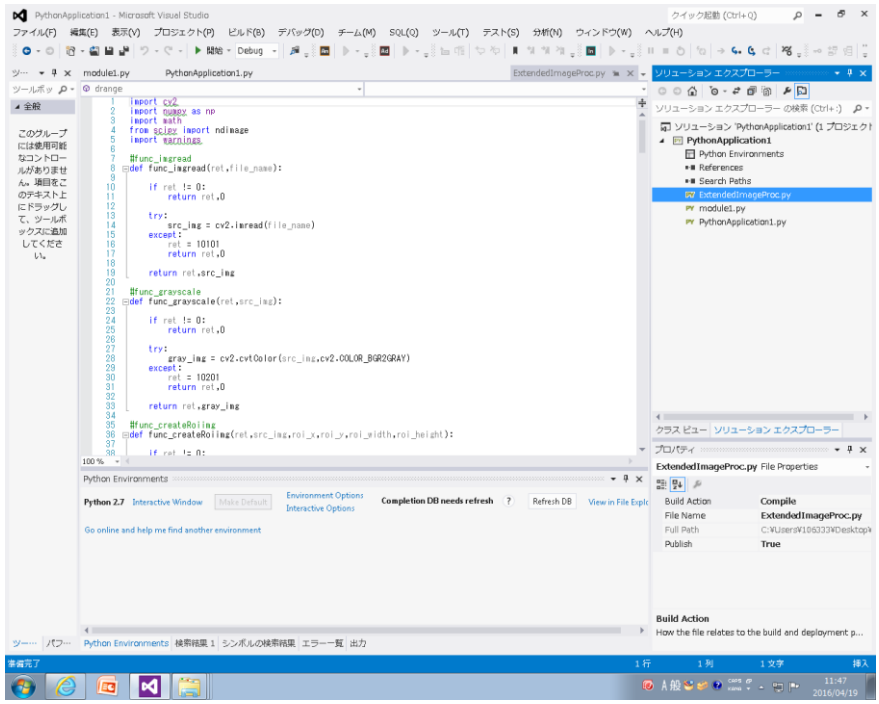
③Of the solution Explorer PY I click the right button on an icon and choose "additional "→" existing item".



④I choose the .py file which I want to add and push ″the Add button″.



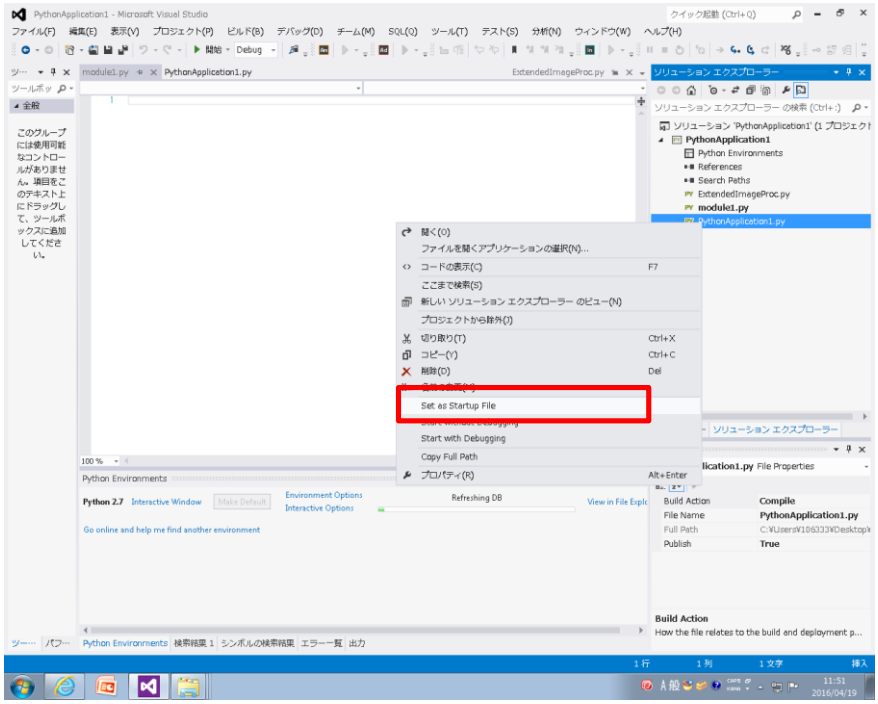
⑤The .py file which I added is opened.



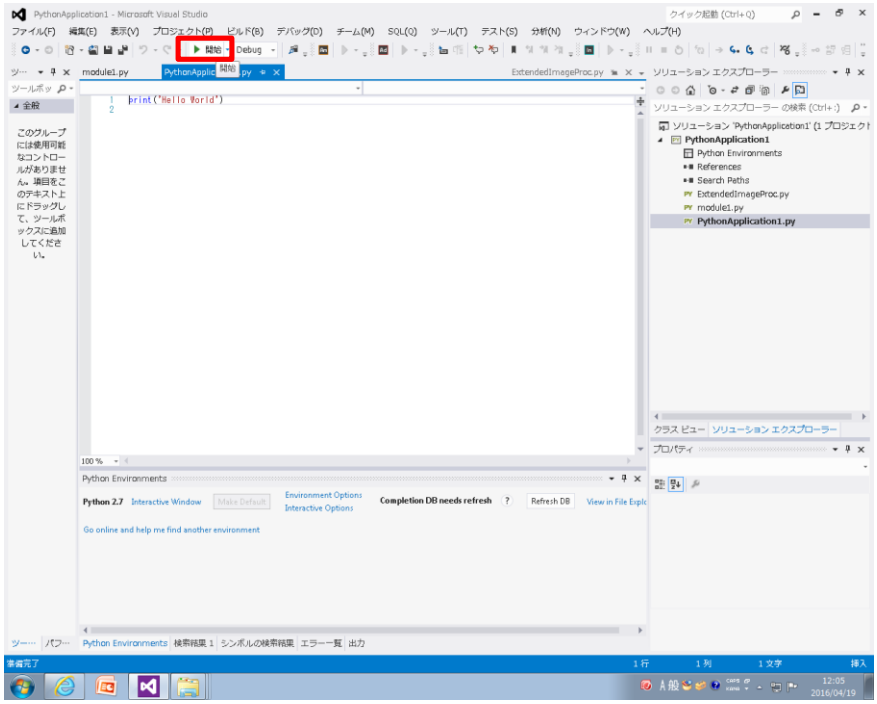
【 Method to carry out a script file in Visual Studio 】

I explain a method to carry out .py file on VisualStudio.

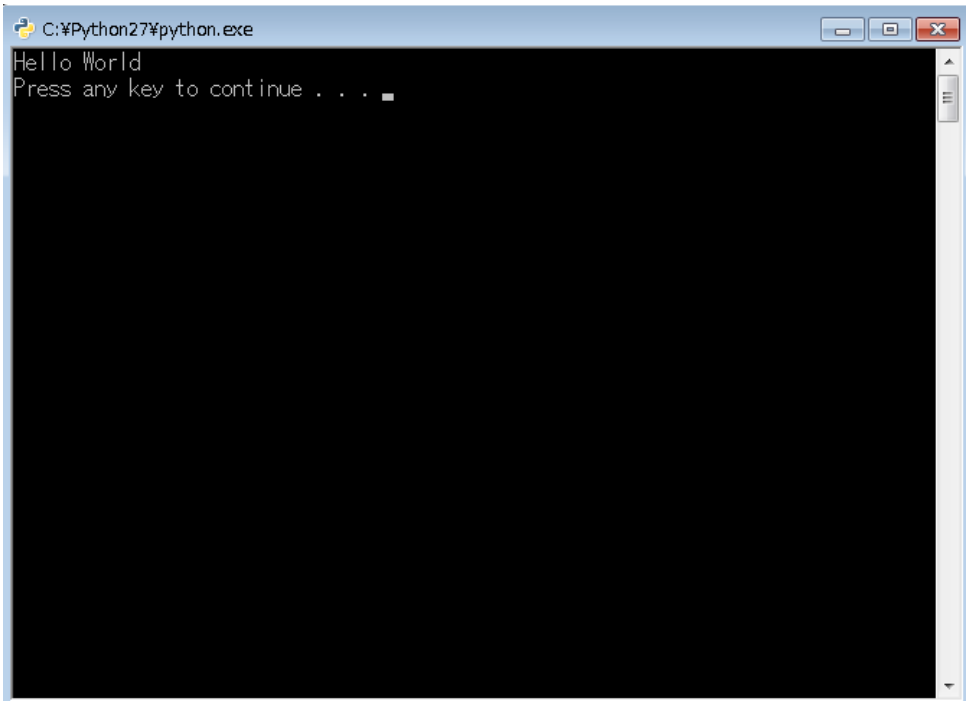
①I click the right button on the .py file name that I want to execute the solution Explorer, I choose "Set as Startup File".



②I push "the start button" and execute.



③A terminal window is displayed, and a result is output.



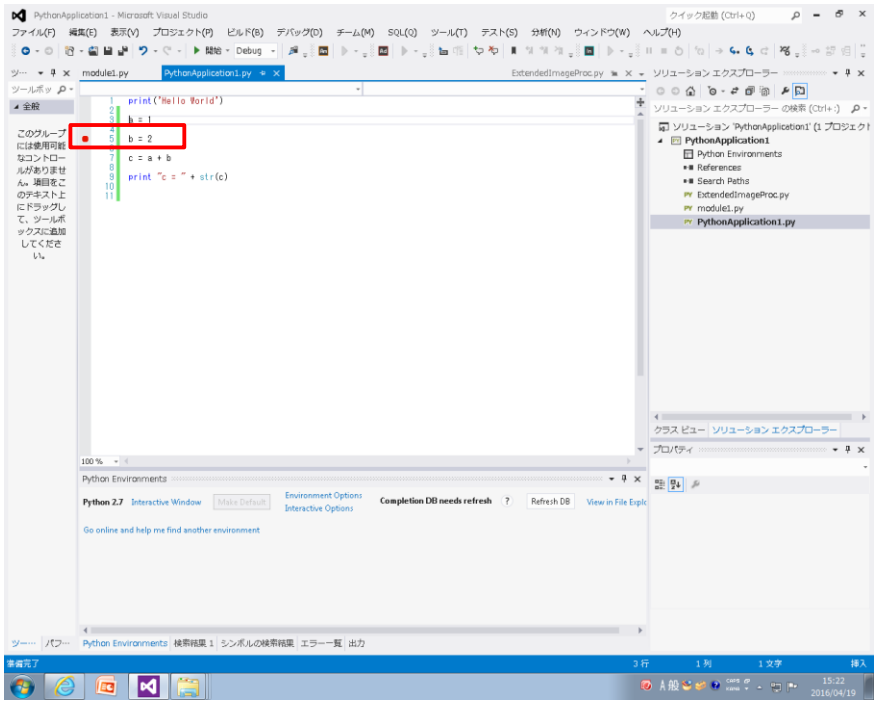
【 Script file debugging method of Visual Studio 】

VisualStudio comprises a powerful debugging tool for the debugging of the script file. "A debugging method using Blake Point" that is representative how to use explains "a watch function" in detail here.

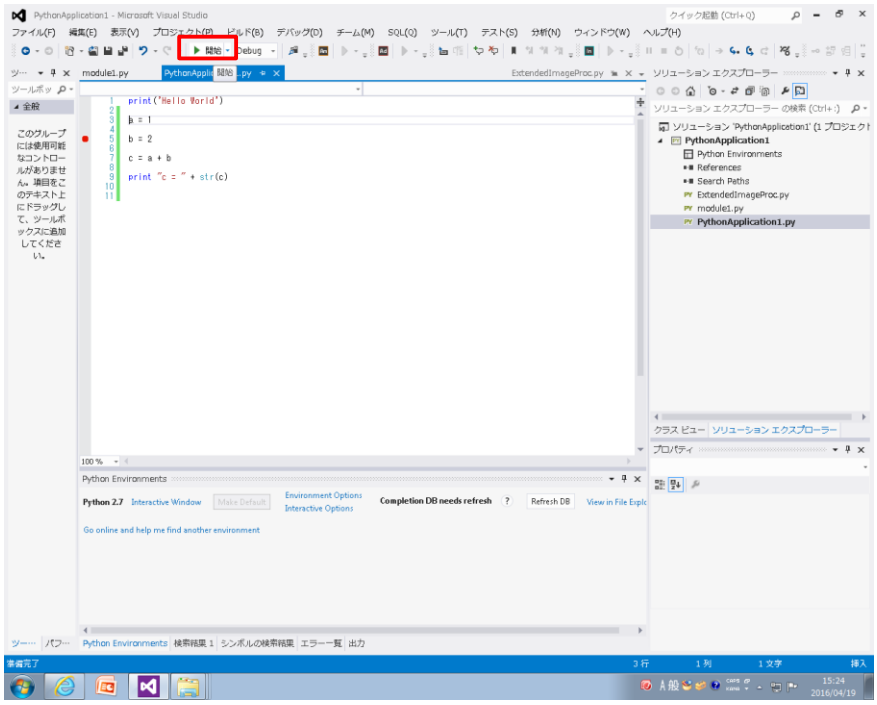
「 Debugging using the breakpoint 」

①I open the solution file .sln file for the application from VisualStudio.

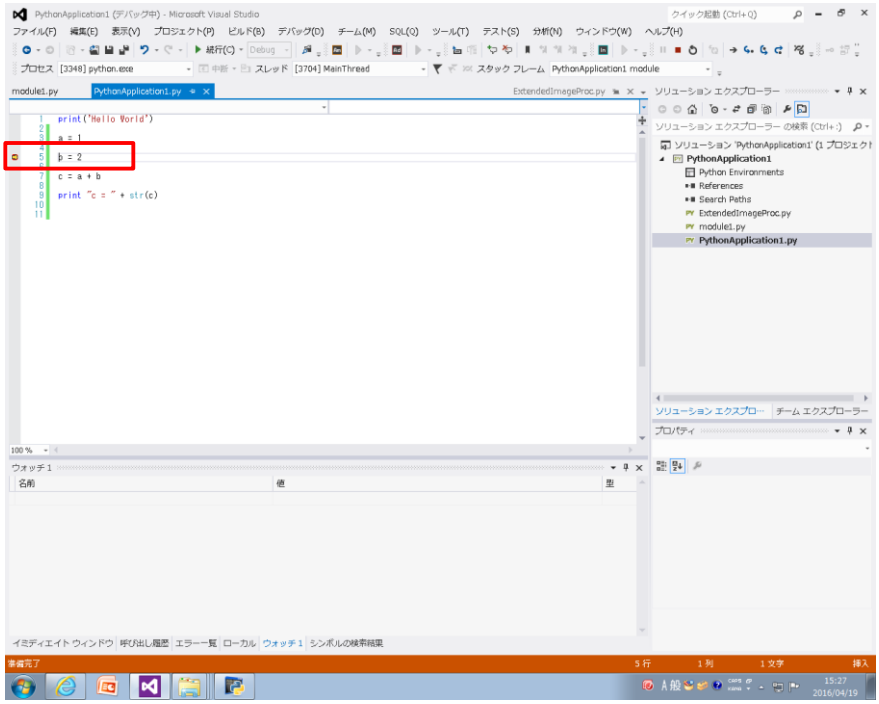
②I click the side of the line where I want to put a break point.



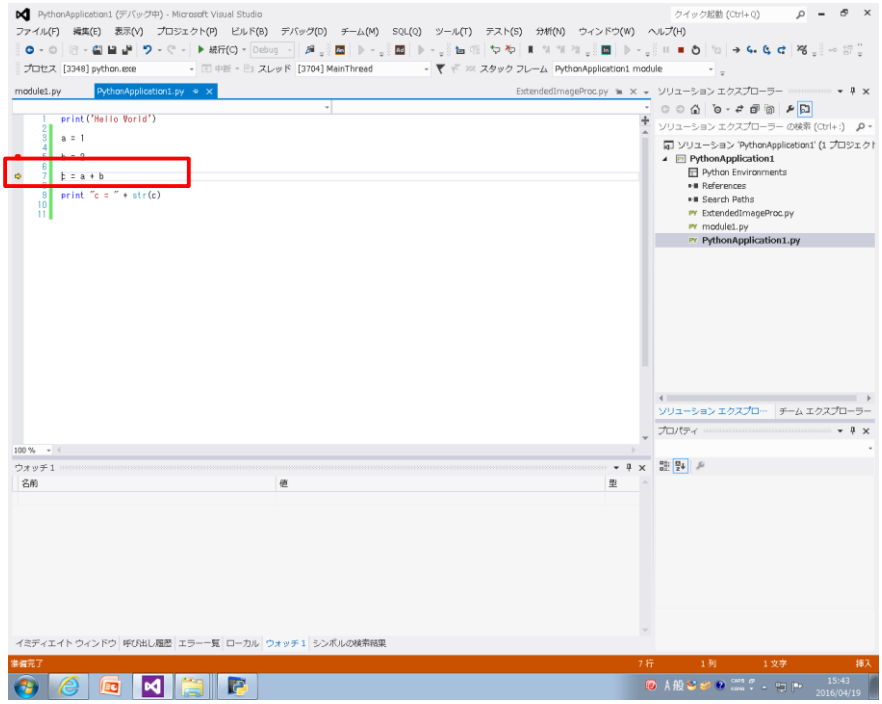
③I push "the start button".



④Debugging is carried out and stops at the failing mark position.



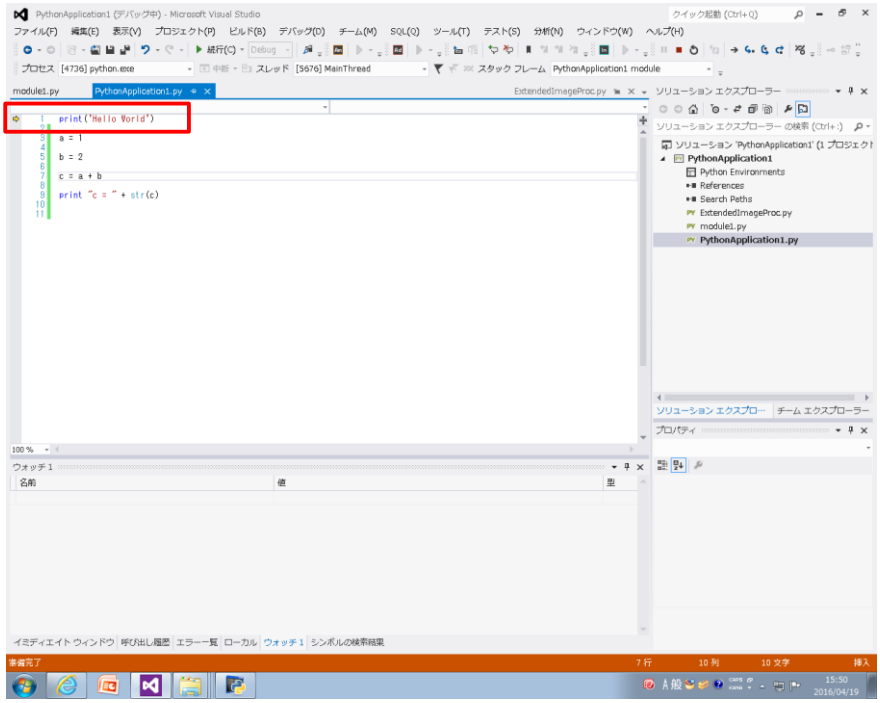
⑤By one line comes by step practice by pushing "the F10 key".
I can execute again from a failing mark position by pushing "the F5 key".



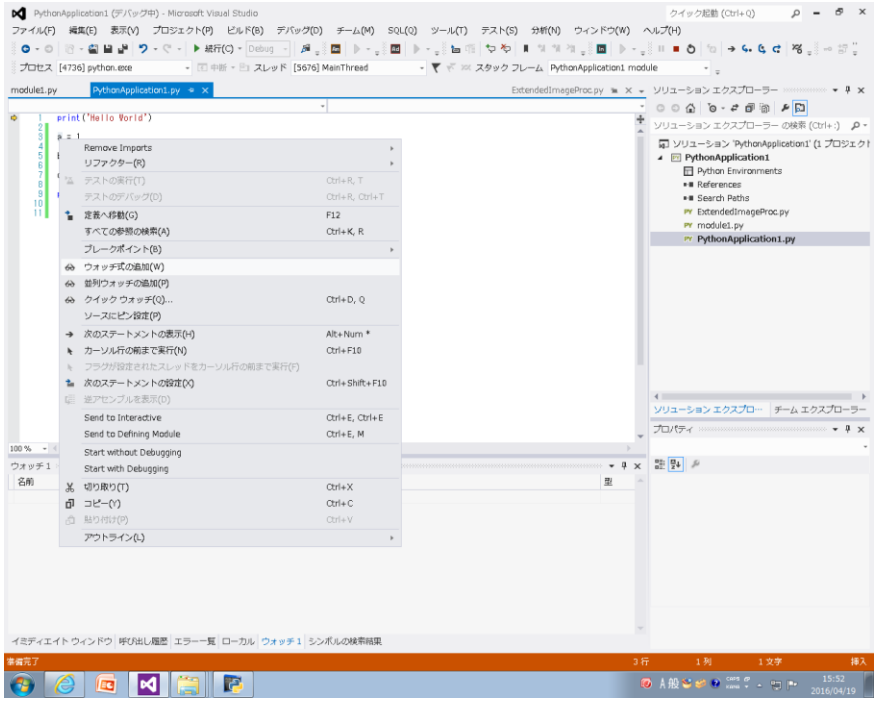
「 About a watch function 」

"A watch function" is the function that can identify the contents of the variable that I defined at the time of debugging.
I explain the usage of this function.

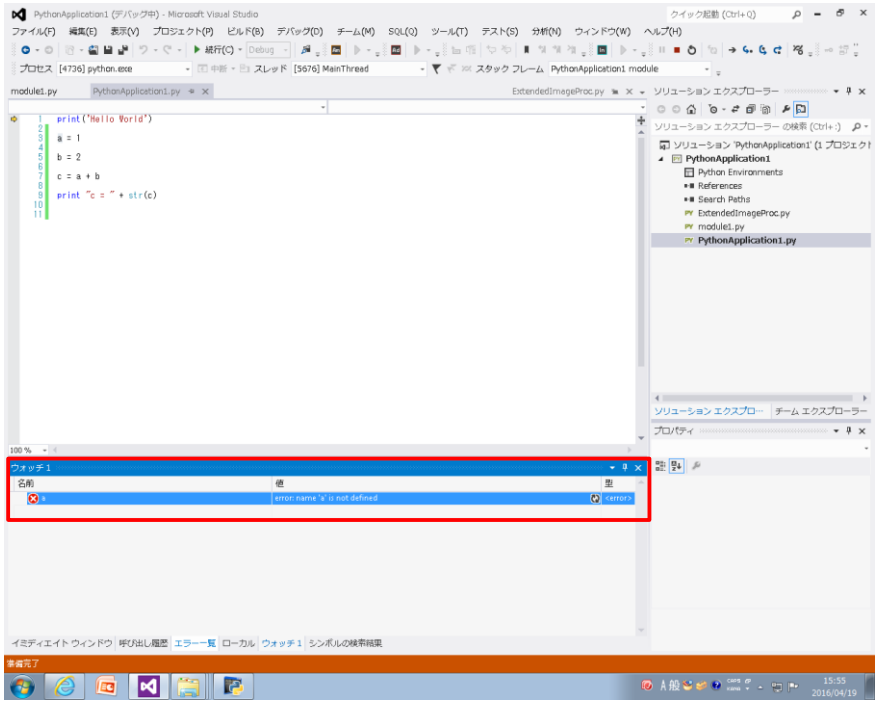
①I push "the F10 key" and execute debugging. I stop in the first first line.



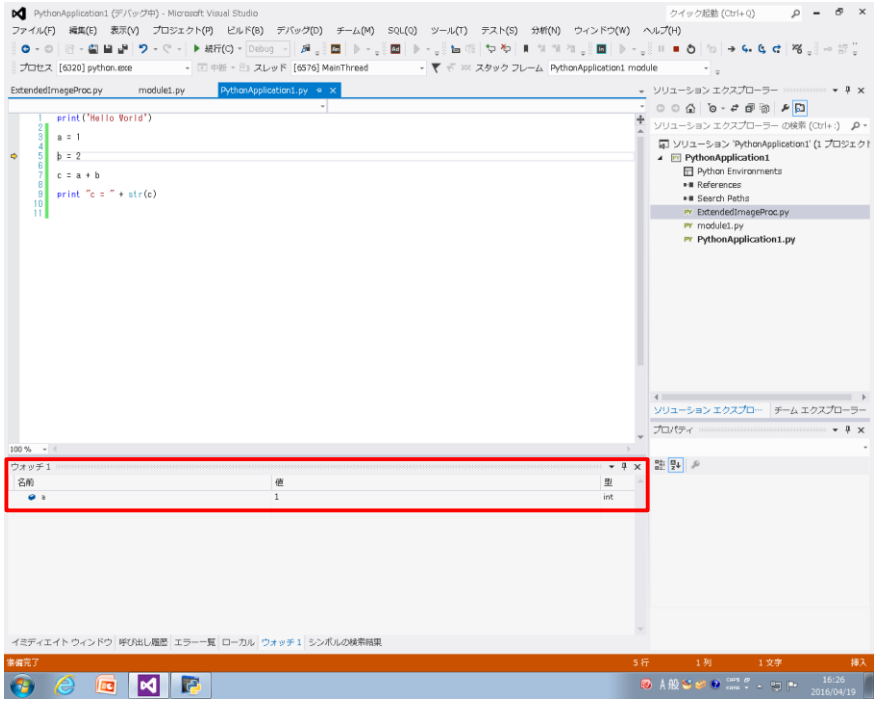
②Double-click on the variable that wants to identify the contents, and perform a right-click; and in "a watch type more ". I choose it.



③"A watch window" stands up, and the variable that I registered is displayed.



④A value is input when I pass the line of the variable that I set and can confirm it.



2. About OpenCV,NumPy,SciPy

2.1 Summary

For script development for the application of CRAVIS-mini, I use an image processing library and the numerical computation library which are a freeware. Here, I explain the summary of various libraries.

OpenCV (Open Source Computer Vision Library)

OpenCV becomes image processing, the image analysis library of the open source which Intel Corporation developed.
I cope with wide languages such as C++,Python,Java,MATLAB. In addition, it supports Windows, Mac OS X, Linux, Android as a platform.

NumPy

NumPy becomes the expansion module to perform numerical computation effectively in Python.
I add the support of the multi-demensional array (I can express a vector or a matrix) of the with a pattern to perform effective numerical computation to Python and offer a large-scale high-level mathematics function library to operate them.

SciPy

SciPy becomes the open source library of the scientific tool for Python.
Numpy is built in the basics and provides statistics, optimization, integral calculus, linear algebra, Fourier transform, a signal, image processing, genetic algorithm, various modules including the official development assistance (the differential equation that it is the way it goes) Solver.

2.2 List of functions

Function introduction of OpenCV

I introduce a part of the function of OpenCV.
When I use a function of OpenCV in a program, I attach "cv2." before a function name.
The version of OpenCV which I can use in CRAVIS-mini becomes OpenCV 3.0.

Function name	Explanation
imread、imwrite	I perform reading, the preservation of the picture file.
cvtColor	I convert the color space of the image.
threshold、 adaptiveThreshold	I perform threshold processing.
bilateralFilter、 GaussianBlur、medianBlur	I smooth an image. (bilateral filter, Gaussian filter, median filter)
dilate、erode	I perform the expansion, the shrinkage of the image.
morphologyEx	I perform morphology conversion such as an opening, the closing of the image.
Sobel、Canny、Laplacian	I calculate an edge image, a differential calculus image. (Sobel, Canny, Laplacian)
moments	I calculate the moment of the image.
findContours	I detect the outline of the image.
HoughLines、HoughCircles	I detect a straight line, Japanese yen by Hough transformation.
absdiff	I calculate the absolute value for the difference between the image.
countNonZero	I count the number of the pixels that are not zero.
resize	I change the size of the image.
waitKey	There is key input or waits until the time when I appointed it passes.
destroyAllWindows	I close all the windows produced in OpenCV.
imshow	I open the window and display an image.
line、circle	I draw a line, Japanese yen to a window.
putText	I draw a letter to a window.

Function introduction of NumPy

I introduce a part of the function of NumPy.
When I use a function of NumPy in a program, I attach "np." or "numpy." before a function name.

Function name	Explanation
sum、prod、mean、average、median、min、max、std、var	Average, the median, the minimum, the maximum, standard deviation, the dispersion that are available for the sum, the product, an arithmetical mean, a heaviness charge account
add、substract、multiply、divide、negative	Addition, subtraction, multiplication and division operation every element
reciprocal、power	Reciprocal number, involution
histogram、histogram2d、histogramdd	One-dimensional histogram, two-dimensional histogram, three-dimensional histogram
sin、cos、tan、arcsin、arccos、arctan、arctan2、sinh、cosh、tanh、arcsinh、arccosh、arctanh	Trigonometric function
around、floor、ceil、fix、trunc	I round it and handle it
degrees、radians、deg2rad、rad2deg、unwrap	Conversion of radian and the degree, conversion to -2 π - 2 π range
exp、log、log10	Index, logarithm
dot、cross	The scalar product, cross product
interp	Linear primary interpolation
linalg	Linear algebra (eigenvalue, characteristic vector, norm, solution, inverse matrix of the linear equation)
random	Random number generation
sort、argsort、lexsort、msort	Sort
argmax、argmin、nanargmax、argmin、nonzero、argwhere、where	Search
count_nonzero	Count the number of non-zero
all、any、isfinite、isinf、isnal、isneginf、isposinf、logical_and	Sequence, contents check of the matrix
logical_and、logical_or、logical_not、logical_xor	Boolean operation (available for array)
allclose、array_equal、array_equiv、greater、greater_equal、less、less_equal、equal、not_equal	Comparison
bitwise_and、bitwise_or、bitwise_xor、invert、left_shift、right_shift	Bit operation every element

Function introduction of NumPy

I introduce a part of the function of NumPy.
When I use a function of NumPy in a program, I attach "np." or "numpy." before a function name.

Function name	Explanation
amax、amin、ptp、nanmax、nanmin	The maximum along the axis, the minimum, maximum - minimum, the maximum except the indefinite, the minimum
corrcoef、cov	Coefficient of correlation, covariance line
fft	Disintegration Fourier conversion
trapez	Numerical value integral calculus in trapez
real、imag、angle、coonj	Complex number (real number, imaginary number, corner, complex conjugate)

Function introduction of SciPy

I introduce a part of the function of SciPy.
I attach "scipy." before a function name after having described it with "import scipy"
in the beginning when I use a function of SciPy in a program.

Function name	Explanation
cluster	The k-mean method, cohesion type hierarchical cluster ring
constants	The physical fixed number (the fixed number such as velocity of light, a gravitational constant, the Avogadro's number)
fftpack	Fourier transform
integrate	Numerical value integral calculus
interpolate	Interpolation (hyperspace, multinomial expression, spline)
io	Input and output (matlab, matrix market, wav, arff, netcdf)
linalg	Linear algebra
misc	Other functions
ndimage	Simple image processing, filter processing
odr	Orthogonality distance recurrence
optimize	Optimization such as the conjugate incline method, the Newton method
signal	I enfold it and am signal processed B spline, a wavelet, various filters
sparse	Sparse line processing
spatial	High-speed search handling of kd- tree, Delaunay triangulation
distance	Distance between vectors, distance line
special	Special functions such as Bessel function, gamma function, error function, the Legendre function
stats	Statistics functions such as density function, a share rank point, the moment

3. About cmlib

3.1 Summary

When I take a step about camera, illumination, I/O, communication in CRAVIS-mini, I use cmlib which is a control library for Python of the in-house production. cmlib includes the following function.

・Initialization (camera, illumination, I/O, communication)

・Camera setting (shutter speed)

・Camera photography

・Illumination control

・I/O control (4 input, 4 output)

・The image log output

・The movement log output

・RS-232C transmission and reception

・MC protocol transmission and reception (Ethernet, RS-232C)

3.2. List of functions

The lists of functions of cmlib are as follows. I attach "cmlib." before a function name and use it after having declared "import cmlib" when I use a function of cmlib.

Function name	Return value	Argument	Explanation
Init	type : int (Error code)	None	I make a state to initialize camera and I/O, and to be usable. I let the default value of the LightOn function reflect the illumination light control value that I set with a camera setting tool.
OutOK	type : int (Error code)	None	I turn on "OUT1/OK" of I/O and turn off other output signals.
OutNG	type : int (Error code)	None	I turn on "OUT2/NG" signal of I/O and turn off other output signals.
OutStatus	type : int (Error code)	None	I turn on "OUT0/Status" signal of I/O and turn off other output signals.
OutClear	type : int (Error code)	None	I turn off all output signals of I/O. [ex: with all output signals OFF] ret = cmlib.OutClear()
SetIO	type : int (Error code)	int, int (Signal number) Signal number : 0=OUT0/Status 1=OUT1/OK 2=OUT2/NG 3=OUT3 レベル : 0 : OFF 1 : ON	I set the output signal of the number that I appointed in a designated level. [ex: in OUT3 ON] ret = cmlib.SetIO(3, 1)
WaitTrigger	type : int (Error code)	None	I wait till "IN0/TRIG" of I/O becomes ON (battery level Low) without doing anything.
WaitIO	type : int (Error code)	int (Signal number) Signal number : 0=IN0/TRIG 1=IN1 2=IN2 3=IN3	I wait till the input signal of the number that I appointed of I/O becomes ON (battery level Low) without doing anything.
WaitIOInv	type : int (Error code)	int (Signal number) Signal number : 0=IN0/TRIG 1=IN1 2=IN2 3=IN3	I wait till the input signal of the number that I appointed of I/O becomes OFF (battery level High) without doing anything.
GetIO	type : int (Error code) 、Lv) Lv : 1 : ON (Battery level Low) 0 : OFF (Battery level High)	int (Signal number) Signal number : 0=IN0/TRIG 1=IN1 2=IN2 3=IN3	I read the signal level of the input signal of the number that I appointed of I/O. Because a plus and minus of the logic is reversed to a real battery level as for battery level Low, "0" as for "1" in battery level High, please be careful.

Function name	Return value	Argument	Explanation
GrabFrameEx	type : int, ndarray (error code、Image data)	None	I photograph it with a camera and return a photographed image. [ex:photographs it and stores it in im] ret, im = cmlib.GrabFrameEx()
ReadAdjustedValueEx	type : int, int, int, int, int (error code、X、Y、width、height)	None	I return a camera setting tool for the X initial point of the inspection domain, Y initial point, width, the high domain number (1-10) that I set. [an example takes a rectangle domain level of domain 2] ret,X2,Y2,W2,H2 = cmlib.ReadAdjustedValueEx(2)
LightOn	type : int (Error code)	int (Light control level)	I let illumination emit light with the light control level that I appointed (0-100). When there is not an argument, I let you emit light with the light control level that you set with a camera setting tool. (0 turns it on with the maximum lights out, 100) [an example turns it on in light control level 50%] ret = cmlib.LightOn(50)
LightOff	type : int (Error code)	None	I zero a light control level of the illumination and let you turn off the light.
ShutterSpeedSet	type : int (Error code)	int (Shutter time)	I set shutter speed in a designated value. It is ret = cmlib.ShutterSpeedSet(2000) (a unit: [μs], a range: 1-70,000) [an example: at shutter time 2000μs]
ImageLog	type : int (Error code)	ndarray (Image Data)	I store image log to an image log server. When an image log server does not exist, I cannot use it.
ImageLogUsb	type : int (Error code)	ndarray (Image Data)	I store image log in USB memory connected to in CRAVIS-mini.
CmLog	type : int (Error code)	unsigned char* (Character string)	I output the character string that I appointed in movement log.
SerialInit	type : int (Error code)	None	I initialize the RS-232C communication, and RS-232C communication makes the state that there is.
SerialTx	type : int (Error code)	unsigned char*, (Character string)	I output the character string that I appointed in RS-232C. Newline code "CRLF" is inserted in the end of the character string. CRAVIS-mini "is Uncompressing Linux to RS-232C port at the time of power supply injection and reboot... I output a message called done, booting the kernel. In RS-232C connection, correspondence in defiance of the message mentioned above is necessary.
SerialRx	type : int (Error code)	int, int (data length、Time-out time)	It stores character string of the designated data head (a unit: [byte]) sent in RS-232C in a variable. I output an error when it passes with time-out time (a unit: [sec], the maximum: 2,147,483,657) when I appointed it. It is ret,data = cmlib.SerialRx(8,10) [8byte waits an example in time-out ten seconds]

Function name	Return value	Argument	Explanation
McpEtherInit	type : int (Error code)	None	I initialize the MC protocol (Ethernet) communication and make the state that can communicate MC protocol (Ethernet).
McpEtherTx	type : int (Error code)	unsigned short int*、int (data、address)	I send the data which I appointed to the address that I appointed in MC protocol (Ethernet).
McpEtherRx	型 : int、unsigned shor int* (Error code、data)	int、int (data、address))	I receive the data of the data length which I appointed from the address that I appointed in MC protocol (Ethernet) (the number of WORD).
McpSerialInit	type : int (Error code)	None	I initialize the MC protocol (RS-232C) communication and make the state that can communicate MC protocol (RS-232C).
McpSerialTx	type : int (Error code)	unsigned short int*、int (data、address))	I send the data which I appointed to the address that I appointed in MC protocol (RS-232C). CRAVIS-mini "is Uncompressing Linux to RS-232C port at the time of power supply injection and reboot... I output a message called done, booting the kernel. In RS-232C connection, correspondence in defiance of the message mentioned above is necessary.
McpSerialRx	type : int、unsigned shor int* (Error code、data)	int、int (data、address)	I receive the data of the data length which I appointed from the address that I appointed in MC protocol (RS-232C) (the number of WORD).

3.3. List of error codes

The function of cmlib outputs the following error codes as a return value.

Error code (decimal number)	Function name	Function name	Fatality degree 0:no problem 1:reboot 2:Refer to a developer
0	-	Normalcy	-
100001	Init	Camera initialization error	2
200002		GPIO initialization error	2
100003		Camera gain setting error	1
200004		Configuration file reading error	2
200101	OutOK	Exception error	2
200201	OutNG	Exception error	2
200301	OutStatus	Exception error	2
200401	OutClear	OK/OUT1 signal exception error	2
200402		NG/OUT2 signal exception error	2
200403		Status/OUT0 signal exception error OUT3 signal exception error	2
200404		Status/OUT0 signal exception error OUT3 signal exception error	2
501	SetIO	Parameter abnormality	0
200502		Abnormal termination	2
200701	GrabFrameEx	Exception error	2
100702		Camera gain setting error exception error	1
200801	LightOn	Exception error	2
802		Parameter error	0
201101	LightOff	Exception error	2
1201	ShutterSpeedSet	Parameter abnormality	0
1301	ImageLog	The initialization error that a log server is not found in	0
201401	SerialInit	Transmission error	2
201501	SerialTx	Reception time-out	2
1601	SerialRx	Abnormal termination	0
201602		A designated number is out of a range	2
1702	ReadAdjustedValueEx	A designated number is out of a range	0
1703		指The initialization opening error that there are no domain data of the designated number in	0
201901	McpEtherInit	Initialization closing error	2
201902		Transmission error	2
202001	McpEtherTx	Transmission error	2
202002		Open error	2
202003		Closing error	2
202101	McpEtherRx	Reception error	2
202102		Open error	2
202103		Closing error	2

Error code (decimal number)	Function name	Function name	Fatality degree 0:no problem 1:reboot 2:Refer to a developer
2201	WaitIO	Parameter error	0
202202		Abnormal termination	2
2301	GetIO	Parameter error	0
202302		Abnormal termination	2
202401	McpSerialInit	Ialization error	2
202501	McpSerialTx	Transmission error	2
202601	McpSerialRx	Reception error	2
2701	ImageLogUsb	The parameter error that note is prohibited in the USB storage which a USB storage is not connected to	0
2702		A USB storage is put under ban of a note	0
2901	WaitIOInv	Parameter error	0
202902		異Abnormal termination	2

4. About a sample script

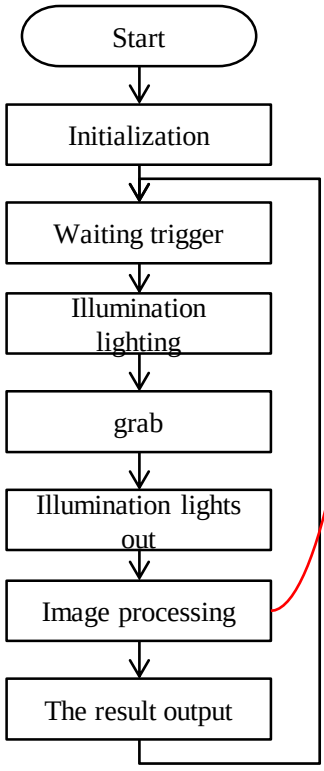
As help of the script development for the application of CRAVIS-mini,
I show the sample script in the production engineering headquarters.
This sample script is the model file which becomes the frame in developing application and the user reorganizes this and can use it.

4.1 Structure of the sample script

The sample script provides it in the form of a script file and one parameter file which became four structure to make each function of the application plain.
The image becomes like the chart below.

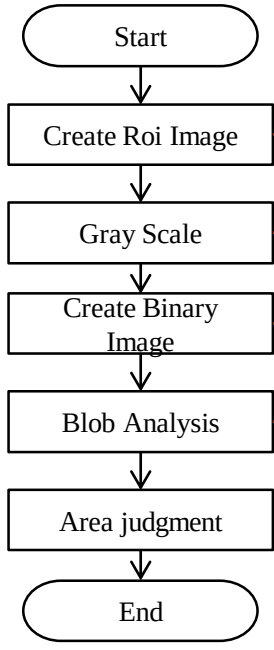
① main_sequence .py

A sequence file of the whole system. This file call for an image processing flow function.



② image_proc _sequence.py

The file of the image processing flow function.This file call for an image processing library.



③ image_proc _component.py

The function group file that gathered up each image processing library (include OpenCV).I give an exception handling.

Ex)gray scale

```
#func_grayscale
def func_grayscale(ret,src_img):

    if ret != 0:
        return ret,0

    try:
        gray_img
= cv2.cvtColor(src_img,cv2.COLOR_BGR2GRAY)
    except:
        ret = 10201
        return ret,0

return ret,gray_img
```

I wrap it by an exception handling

Management with the error code

④ parameter _class.py

I described a parameter to use by application
A class file.

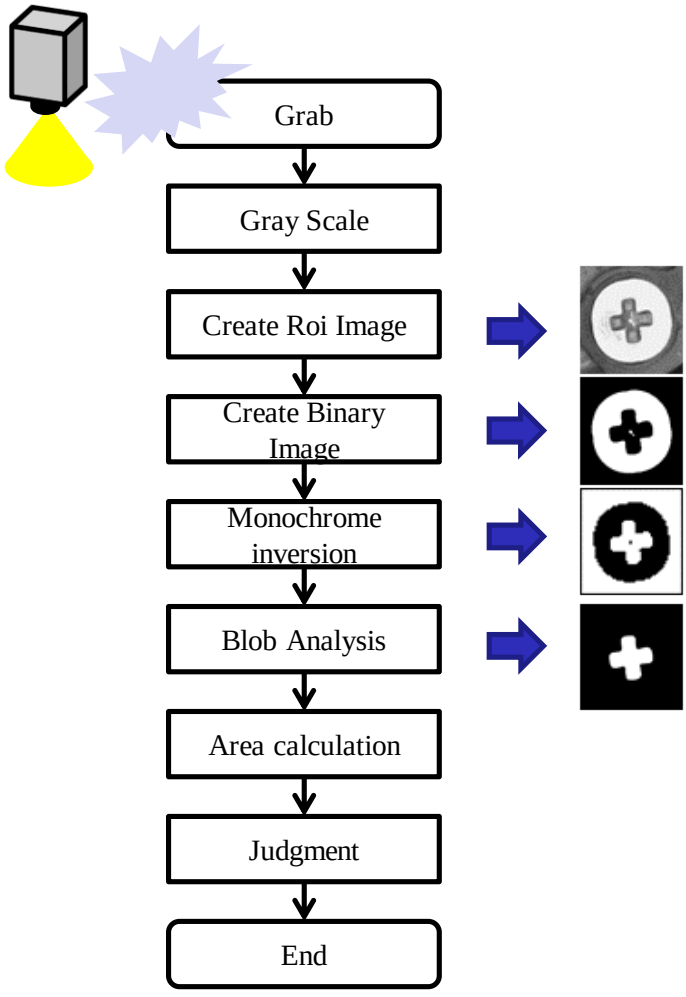
⑤ Parameter.ini

The parameter file that the value of the real parameter is described.
At the time of the parameter change, I change the value of this file.

4.2 Example introduction

I introduce an example of the application that I made using a sample script.
The example contents calculate the size (area) of the cross hole of the screw.

The flow of the image processing is as follows.
I cut only the domain that at first is necessary for a judgment to have possibilities to take time when I bet processing on the whole photography domain.
When I perform screw かじりの judgment, I count the number of the pixels of the white of the cross hole in the image which I binalized.
At first I binalize it after having cut an inspection domain.
I pull out only a cross hole for quantity of characteristic by Blob (lump of the pixel) extraction and perform area calculation afterwards.
I let you bite it and compare it with the threshold (user designation) of the judgment and perform the judgment that bites the head (cross hole) of the screw, or is not, and it is the inspection end.
I show a flow in the chart below.
In addition, I comment on a file of the application that I made using a sample script after the following page.



5. Instructions about the use of the software

The software of Canon FA development center (the following, our center) or the third party is included in software incorporated in CRAVIS-mini (the following, this plane).

A copyright belongs to us, and our software and documents to accompany are protected by Copyright Act, an international treaty article and other governing laws.

In addition, I use the software module of the open source which a third party owns a copyright at current opportunity and distributes. GNU General Public License v2 (following, GPL), GNU Lesser General Public License v2.1 (following, LGPL) or a software module catching the application of other licensing agreement is included in those parts.

As for the license readout of each software module, please see iFolder (201500990/01_ instruction manual, drawing, specifications /OSS/OSS license summary .pdf). To the software module of the open source, there is the thing which a thing demanding that I enable the acquisition of the source code of the module as a condition to distribute the software module of the practice form to and the copyright holder oblige to license indication. One set of source code of the software module of the open source displays it in iFolder (201500990/01_ instruction manual, drawing, less than specifications /OSS/source/) and is available from this. GPL, LGPL and other licensing agreement are listed in a source code.