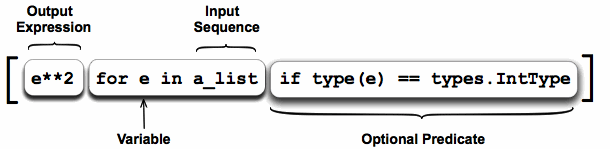
|  |
| --- |
| http://stackoverflow.com/ |
| http://nullege.com/ |
| <http://www.tutorialspoint.com/python/index.htm> |
| 책: 고성능 파이썬 프로그래밍 |
| <https://wiki.python.org/moin/ParallelProcessing> |
| https://github.com/vhf/free-programming-books/blob/master/free-programming-books.md |
| https://wiki.python.org/moin/IntegratingPythonWithOtherLanguages |

|  |
| --- |
| Python 2, 3 동시 설치 |
| Control panel이 잘 실행 되지 않으면,  이전 버전의 환경 변수가 삭제 되지 않아서 그렇다.  PYTHONHOME  PYTHONPATH  TCL\_LIBRARY  PYTHONHOME  C:\WinPython\python-3.5.1;  PYTHONPATH  C:\WinPython\python-3.5.1;C:\WinPython\python-3.5.1\DLLs;C:\WinPython\python-3.5.1\Lib;C:\WinPython\python-3.5.1\Lib\site-packages;  TCL\_LIBRARY  C:\WinPython\python-3.5.1\tcl\tcl8.6 |



**List Comprehensions**

|  |
| --- |
| * **import X** imports the module X, and creates a reference to that module in the current namespace. Or in other words, after you’ve run this statement, you can use *X.name* to refer to things defined in module X. * **from X import \*** imports the module X, and creates references in the current namespace to all *public* objects defined by that module (that is, everything that doesn’t have a name starting with “\_”). Or in other words, after you’ve run this statement, you can simply use a plain *name*to refer to things defined in module X. But X itself is not defined, so*X.name* doesn’t work. And if *name* was already defined, it is replaced by the new version. And if *name* in X is changed to point to some other object, your module won’t notice. * **from X import a, b, c** imports the module X, and creates references in the current namespace to the given objects. Or in other words, you can now use *a* and *b* and *c* in your program. |
| """  from myfuncTest import \*  print( myTestSum(2,30) )  print( myTestSum2(2,30) )  """  import myfuncTest  print( myfuncTest.myTestSum(2,30))  print( myfuncTest.myTestSum2(2,30)) |
| 특정 디렉토리에 있는 pyd import하기  import sys  sys.path.insert(0,'E:\hccho\makePYD\Debug')  import hccho\_test\_module |

**Call C++ DLL**

http://stackoverflow.com/questions/252417/how-can-i-use-a-dll-file-from-python

|  |
| --- |
| It's very easy to call a DLL function in Python. I have a self-made DLL file with two functions: addand sub which take two arguments.  add(a, b) returns addition of two numbers sub(a, b) returns substraction of two numbers  The name of the DLL file will be "demo.dll"  **Program:**  from ctypes import\* # give location of dll mydll = cdll.LoadLibrary("C:\\demo.dll") result1= mydll.add(10,1) result2= mydll.sub(10,1) print "Addition value:-"+result1 print "Substraction:-"+result2  **Output:**  Addition value:-11 Substraction:-9 |

**Call COM dll from Python**

|  |
| --- |
| Static library는 python에서 불러 사용할 수 없다.  Q. is it possible to import modules from .lib library to Python program (as simple as .dll)?  🡺  In theory, yes; in practice, probably not -- and certainly not as simply as a DLL. Static libraries are essentially just collections of object files, and need a full linker to correctly resolve all relocation references they may contain. It might be possible to take your static library and simply link its contents to form a shared library, but that would require that the static library had been built as position independent code (PIC), which is not guaranteed. |

|  |  |
| --- | --- |
| comtypes설치  <https://sourceforge.net/projects/comtypes/files/>  (easy\_install 로 설치한 경우는 CreateObject에서 오류 발생. 다운 받아 직접설치 해야…)  (sourceforge comtypes google검색하여 exe파일 다운 받음)  Com DLL은 반드시 regsvr32로 등록되어 있어야 함.   |  | | --- | | Test 대상 hcchoATLtest2Lib 밑에 interface AAA2 밑에 method aa22 |   from comtypes.client import CreateObject  from comtypes.client import GetModule  X = GetModule("E:\\hccho\\hcchoATLtest(VS2012)\\Release\\hcchoATLtest2.dll")  #X = GetModule( ('{2F0F0202-2E9C-46F5-8AEF-78D03BAD98C6}',1,0)) #<--이렇게 해도 됨  Y = CreateObject(X.AAA2, interface=X.IAAA2)  a=3.5  b=5.6  print(Y.aa22(a,b))  import ctypes  n=10  mydata\_tmp=ctypes.c\_double \* n  mydata = mydata\_tmp()  for i in range(0,n):  mydata[i]=i  mydata[5]=121.33  print("Before: ")  for i in range(0,n):  print(mydata[i])  Y.ArrayTest(n,mydata)  print("After: ")  for i in range(0,n):  print(mydata[i]) |
| def MathFintest():  from comtypes.client import CreateObject  from comtypes.client import GetModule    MF0 =GetModule("C:\\Windows\\mathfin.dll")  MF =CreateObject(MF0.OptFormula, interface=MF0.IOptFormula)      gr = MF.BSCall(42,40,0.5,0.1,0.2,0)  print(type(gr))  print (gr.Delta)    **dd = MF0.MFDate()**  print( type(dd))  dd.year=2016        return; |
| GetModule (TypeLib Id, Major Version, Minor Version)  GetModule은 이렇게 해도 OK, **1,0 🡨-version**  X = GetModule( ('{2F0F0202-2E9C-46F5-8AEF-78D03BAD98C6}',**1,0**))  # Interface의 uuid |
| Download and install [comtypes](https://pypi.python.org/pypi/comtypes)\*, put the Snippets module from Mark Cederholm in PYTHONPATH, and you're all set.  from snippets102 import GetLibPath, InitStandalone  from comtypes.client import GetModule, CreateObject  m = GetModule(GetLibPath() + "esriGeometry.olb")  **InitStandalone()**  p = CreateObject(m.Point, interface=m.IPoint)  p.PutCoords(2,3)  print p.X, p.Y  **InitStandalone() 🡨뭔지???** |
| import comtypes.gen.hcchoATLtest2Lib |

**Call Python From C++**

|  |
| --- |
| https://docs.python.org/3.5/extending/embedding.html |
| #include "python.h"  int main(int argc, char\* argv[])  {  wchar\_t \*program = Py\_DecodeLocale(argv[0], NULL);  if (program == NULL) {  fprintf(stderr, "Fatal error: cannot decode argv[0]\n");  exit(1);  }  Py\_SetProgramName(program); // optional but recommended  Py\_Initialize();  PyRun\_SimpleString("from time import time,ctime\n"  "print('Today is', ctime(time()))\n");  Py\_Finalize();    PyMem\_RawFree(program);  return 0;  } |
| Fatal Error 발생: path에 python 디렉토리 문제   * You just have to copy **Python's Lib/ directory** in your program's working director   PATH="C:\python33\_x64;%PATH%"  PYTHONPATH="C:\python33\_x64\DLLs;C:\python33\_x64\Lib;C:\python33\_x64\Lib\site-packages"  PYTHONHOME=C:\python33\_x64 |
| <http://stackoverflow.com/questions/49137/calling-python-from-a-c-program-for-distribution> |
|  |

**C++를 이용하여 pyd module만들기**

|  |
| --- |
| /\*  . win32 프로젝트 - DLL/빈프로젝트  . 속성에서  . C/C++ -> 일반 -> 추가 디렉토리: C:\WinPython\python-3.5.1\include  . 링커 -> 일반 -> 추가 라이브러리 디렉토리: C:\WinPython\python-3.5.1\libs  . 링커 -> 입력 -> 추가 종속성: python35.lib  . 링커 -> 명령줄 -> 추가 옵션: /export:PyInit\_spam 🡨-c코드에서 정의된 함수  . 링커 -> 일반 -> 출력 파일: $(OutDir)\spam.pyd  \*/  pydfilename.pyd와 C++ 코드 상의 PyInit\_pydfilename 같아야 됨 |
| **// strlen, strlen2 🡨- 2개 함수를 만드는 예:**  #include<Python.h>  staticPyObject\*  spam\_strlen(PyObject\*self,PyObject\*args)  {  constchar\*str=NULL;  intlen=0;  if(!PyArg\_ParseTuple(args,"s",&str))  returnNULL;  len=strlen(str);  returnPy\_BuildValue("i",len);  }  staticPyObject\*  spam\_strlen2(PyObject\*self,PyObject\*args)  {  constchar\*str=NULL;  intlen=0;  if(!PyArg\_ParseTuple(args,"s",&str))  returnNULL;  len=strlen(str)+1;  returnPy\_BuildValue("i",len);  }  staticPyMethodDefSpamMethods[]={  {"strlen",spam\_strlen,METH\_VARARGS,"count string length."},  {"strlen2",spam\_strlen2,METH\_VARARGS,"count string length+1."},  {NULL,NULL,0,NULL}/\* Sentinel \*/  };  #ifPY\_VERSION\_HEX>=0x03000000  /\* Python 3.x code \*/  staticstructPyModuleDefspammodule={  PyModuleDef\_HEAD\_INIT,  "spam",/\* name of module \*/  "spam\_doc",/\* module documentation, may be NULL \*/  -1,/\* size of per-interpreter state of the module,  or -1 if the module keeps state in global variables. \*/  SpamMethods  };  PyMODINIT\_FUNC  PyInit\_spam(void)  {  returnPyModule\_Create(&spammodule);  }  #else  /\* Python 2.x code \*/  PyMODINIT\_FUNC  initspam(void)  {  (void)Py\_InitModule("spam",SpamMethods);  }  #endif |
| **Python27의 PyInt\_AsLong 🡺 3.5에서 PyLong\_AsLong으로 바뀜** |
| // PyArg\_ParseTuple: Python 자료형을 C의 자료형으로 변환  // Py\_BuildValue: C 자료형을 Python 자료형으로 변환  // PyArg\_ParseTuple(args, "s", &str)  // 설명: args에 있는 값을 "s", 즉 const char \*로 해석하고, str에 저장  // "i" ==> int, "f" ==> float, "l" => long,  // PyArg\_ParseTuple(args, "lls", &a,&b,&c); ==> "lls" 는 long, long, string으로 해석되고, 각각을 a,b,c에 저장한다. |
| void GetDataFromPyObject2d(PyObject\* A,double\*\*&X,int& n,int& m)  {  // 2차원 Python List로 부터 n by m data를 추출하여 X에 넣어준다.  // X에 대한 메모리 할당은 여기서 이루워지며, memory delete의 책임은 이 함수를 불러사용하는 사용자가 따로 해야함.  int i,j;  PyObject\* B;  n = PyList\_Size(A);  X = new double\* [n];  for(i=0;i<n;i++){  B = PyList\_GetItem(A,i);  m = PyList\_Size(B); // m은 항상 같아야 하는데...  X[i] = new double [m];  for(j=0;j<m;j++){  X[i][j] = PyFloat\_AsDouble(PyList\_GetItem(B,j));  }  }  }  void GetDataFromPyObject1d(PyObject\* A,double\*&X,int& n)  {  int i;  n = PyList\_Size(A);  X = new double [n];  for(i=0;i<n;i++){  X[i] = PyFloat\_AsDouble(PyList\_GetItem(A,i));  }  }  static PyObject \*  python\_argument\_pasing\_test\_hccho\_xxx(PyObject \*self, PyObject \*args)  {  int n,m;  int i,j;  PyObject\* A;  PyObject\* B;  PyObject\* Temp,\*Temp2;  if (!PyArg\_ParseTuple(args, "OO",&A,&B)){ // 약속된 2개의 argument를 분리해 낸다. "OO"는 Object 갯수,  return NULL;  }    double my\_sum=0.0;  // 분리해 낸 각각의 PyObject를 다시 parsing한다.  if (!PyList\_Check(A)){  return NULL;  }    // 약속에 따라, A는 2차원 List임을 알고 parsing  double\*\* X=NULL;  GetDataFromPyObject2d(A,X,i,j);  for(n=0;n<i;n++){  for(m=0;m<j;m++){  my\_sum += X[n][m];  }  }  Free2D(X,i);  //약속에 따라, B는 1차원 List임을 알고 parsing    double\* Y=NULL;  GetDataFromPyObject1d(B,Y,j);  double my\_sum2=0.0;  for(n=0;n<j;n++){  my\_sum2 += Y[n];  }  delete[] Y;  return Py\_BuildValue("[ff]", my\_sum,my\_sum2);  } |
| http://stackoverflow.com/questions/11713701/how-to-debug-c-extensions-for-python-on-windows  Pyd 프로젝트 debugging하기(2.7, 3.5 공통)   1. Libs 디렉토리 및에 있는 python35.lib(python27.lib)파일을 복사하여 python35\_d.lib(python27\_d.lib)를 만든다 링커-추가종속성에서 위의 lib를 명시적으로 지정하지 않아도 됨.pyconfig.에서 “\_DEBUG” define 여부에 따라 자동으로 연결해줌. 2. pyconfig.h 파일에서  #define Py\_DEBUG 이 부분을 comment처리한다. 3. Go to Project > Properties, then under Configuration Properties  - a. General - ensure that you are using the correct Character Set. For me it was Use **Multi-Byte Character Set**. Python 3 probably needs Use **Unicode Character Set**.   다른 dll파일은 디버깅하는 방식으로 디버깅하면 됨   * 속성-디버깅 – 명령: C:\WinPython\WinPython-32bit-3.5.2.2\python-3.5.2\python.exe * 연결: 예 * spyder실행 또는 python cmd창 * visual studio내에서 디버깅 시작(F5) * spyder내에서 test파일 실행   **특정 디렉토리에 있는 pyd import하기**  **import sys**  **sys.path.insert(0,'E:\hccho\makePYD\Debug')** |

**Cython을 이용한 pyd파일 만들기  
(python 코드를 pyd로 변환)**

|  |
| --- |
| 1) 아래와 같이 작성한후 helloworld.pyx 파일로 저장하자  **print** "Hello World"  2) setup.py 파일을 만들고 아래와 같이 코딩  **from** **distutils.core** **import** setup  **from** **Cython.Build** **import** cythonize  setup( ext\_modules = cythonize("helloworld.pyx") )  3) 아래와 같이 실행하여 동적라이브러리 파일을 만들어보자.  python setup.py build\_ext --inplace  4) 이제 동적라이브러리가 만들어 졌으니 아래처럼 import하여 사용하면 끝~~~  >>> **import** **helloworld**  Hello World |

**Scicomp Pyd만들기**

|  |
| --- |
| 2.7용 scifinance\_py.h 에서 PyInt\_AsLong 🡺 PyLong\_AsLong 으로 바꾸어 새로운 파일을 만든다.  예를 들어, scifinance\_py35.h  scicomp에서 자동생성된 xxx\_py.cpp에서  FromPythonBound ==> FromPython  FromPythonFixed ==> FromPython  FromPythonFixed1 ==> FromPython |

|  |  |
| --- | --- |
| **2.7** | **3.5** |
|  |  |
| 링커-명령줄 – 추가옵션  /export:initAC2LVstep2 | /export:PyInit\_AC2LVstep2\_35 |
| #if PY\_MAJOR\_VERSION >= 3  static struct PyModuleDef AC2LVstep2\_pymodule = {  PyModuleDef\_HEAD\_INIT,  "AC2LVstep2",  "AC2LVstep2 option",  -1,  AC2LVstep2\_methods,  NULL,  NULL,  NULL,  NULL  };  PyMODINIT\_FUNC  PyInit\_AC2LVstep2(void)  {  PyObject\* p = PyModule\_Create(&AC2LVstep2\_pymodule);  return SciFinancePython::PostInitModule(p);  }  #else  PyMODINIT\_FUNC  initAC2LVstep2()  {  PyObject\* p = Py\_InitModule("AC2LVstep2", AC2LVstep2\_methods);  SciFinancePython::PostInitModule(p);  }  #endif | #if PY\_MAJOR\_VERSION >= 3  static struct PyModuleDef AC2LVstep2\_pymodule = {  PyModuleDef\_HEAD\_INIT,  "AC2LVstep2",  "AC2LVstep2 option",  -1,  AC2LVstep2\_methods,  NULL,  NULL,  NULL,  NULL  };  PyMODINIT\_FUNC  PyInit\_AC2LVstep2\_35(void)  {  PyObject\* p = PyModule\_Create(&AC2LVstep2\_pymodule);  return SciFinancePython::PostInitModule(p);  }  #else  PyMODINIT\_FUNC  initAC2LVstep2()  {  PyObject\* p = Py\_InitModule("AC2LVstep2", AC2LVstep2\_methods);  SciFinancePython::PostInitModule(p);  }  #endif |

**QuantLib**

|  |
| --- |
| <http://www.smileofthales.com/build-quantlib-for-python/> |
| <http://blog.naver.com/nomore_bet/220624978039>  <http://www.lfd.uci.edu/~gohlke/pythonlibs/>   * 여기서 버전에 맞게 [QuantLib\_Python-1.6.1-cp35-none-win32.whl](javascript:;)   를 다운 받는다.  C:\Quantlib>pip install QuantLib\_Python-1.6.1-cp35-none-win32.whl  이렇게 하면, C:\WinPython\python-3.5.1\Lib\site-packages\QuantLib  가 생기면서, 설치됨 |

**잡동사니**

|  |
| --- |
| **Input 박스**  http://stackoverflow.com/questions/2963263/how-can-i-create-a-simple-message-box-in-python  import ctypes # An included library with Python install.  def Mbox(title, text, style):  return ctypes.windll.user32.MessageBoxA(0, text, title, style)  respond = Mbox('Your title', 'Your text', 4)  print (respond) |
| **Message box**  import easygui  easygui.msgbox("This is a message!", title="simple gui") |
| import ctypes  MessageBox = ctypes.windll.user32.MessageBoxW  MessageBox(None, 'Hello', 'Window title', 0) |
| Warning  import warnings  warnings.filterwarnings("ignore") |

**Parallel Python**

|  |
| --- |
| http://bryan7.tistory.com/438  C:\WinPython\WinPython-32bit-3.5.1.2\python-3.5.1\Lib\site-packages>python ppserver.py -p 35000 -i 172.21.101.38 -s "mysecret" <-----client ip(즉,자기 자신)  ppservers = ("172.21.101.38:35000", )  job\_server = pp.Server(ncpus, ppservers=ppservers,secret="mysecret")  C:\Users\영업\Downloads\pp-1.6.4.4\examples>python sum\_primes.py 1 <----- 1대신 0도 가능, 옵션 없으면, local에서만 계산  ppservers = ("172.21.101.38", )  python ppserver.py -a  ==> 안됨  ppservers = ("172.21.101.38", )  python ppserver.py -a  ==> 안됨  ppservers = ("172.21.101.38", )  python ppserver.py -p 60000  ==> 안됨  ppservers = ("172.21.101.38:60000", )  python ppserver.py -p 60000  ==> 작동함(마지막에 프로세스 찾을 수 없다는 메시지 나옴) |
| <https://wiki.python.org/moin/ParallelProcessing> 🡸 python parallel 정리된 사이트 |

|  |
| --- |
| json파일 위치:  C:\WinPython\WinPython-32bit-3.5.1.2\settings\.ipython\profile\_default\security |
| ipcontroller-client.json  ipcontroller-engine.json  ipcluster start –n 2 🡨cmd창에서 실행하면 json파일 생성됨. 실행할 때마다 갱신 |
| http://ipython.org/ipython-doc/stable/parallel/ |
| Controller PC에서 **ipcontroller.exe --ip=”\*”** 를 실행하면, ipcontroller-client.json, ipcontroller-engine.json 2개의 파일이 생긴다. 참고로, ipcontroller\_config.py 파일을 통해 c**.**HubFactory**.**ip **=** '\*' 지정하는 방법도 있다.   * Ipcontroller.exe를 실행할 때 마다 다른 json파일이 생긴다. 실행할 때마다 client pc로 복사해야 함. * Ipcontroller –ip=192.168.1.16 * HubFactory.ip = ‘192.168.1.16’  1. ipcontroller-client.json 는 client PC로 옮긴다 2. ipcontroller-engine.json 파일은 여러대의 엔진 PC로 옮긴다. 3. 각각의 엔진 pc에서 ipengine.exe를 실행한다.   **ipengine** 또는 **ipcluster engines –n 5** 🡨여러 개의 process 띄우기 |
| 이런 단계를, ipcontroller\_config.py, ipcluster\_config.py를 통해 일괄로 할 수 있다. |
| Local PC에서 엔진을 설치하고 test하는 방법: ipcluster start –n 4  from ipyparallel import Client  rc = Client()  rc.ids  [0,1,2,3] 🡨--- ipcluster start -n 4 를 통해 4개의 엔진을 생성했기 때문    Jupyter notebook의 “IPython Cluster” 탭을 통해서도 가능 |
| Controller와 engine간의 python 버전은 일치해야 됨. |

|  |
| --- |
| Ipython Notebook  <http://freeprog.tistory.com/3> 🡨notebook server만들기  <http://star.mit.edu/cluster/docs/latest/plugins/ipython.html>  <http://goodtogreate.tistory.com/entry/IPython-Notebook->설치방법  http://activisiongamescience.github.io/2016/04/19/IPython-Parallel-Introduction/  그냥 ipython notebook이 아니라, jupyter notebook 서버 설치에 관해 찾아보아야 함. |
| **ipython profile create mynbserver 🡨-생성. 이 때, 생성되는 파일은 모든 내용이 comment처리되어 있음.**  **수정해야 됨**  **ipython notebook --profile=mynbserver 🡨서버 실행. cmd창의 현재 디렉토리가 홈으로 설정됨** |
| **Notebook profile디렉토리 확인: ipython profile locate default** |
| **한대의 PC에서 cluster start 시키고 example을 돌리는 것은 돌아감. 문제는 여러대의 PC를 활용하는 것인데, 아직까지는 성공하지 못함.** |
| Easily distributing a parallel IPython Notebook on a cluster **http://twiecki.github.io/blog/2014/02/24/ipython-nb-cluster/** |

|  |
| --- |
| <https://github.com/tritemio/PyBroMo/wiki/Howto-setup-an-IPython-cluster> **Howto setup an IPython cluster** Antonino Ingargiola edited this page on 18 Jul 2014 · [8 revisions](https://github.com/tritemio/PyBroMo/wiki/Howto-setup-an-IPython-cluster/_history)  Pages 3  * [**Home**](https://github.com/tritemio/PyBroMo/wiki) * [**Howto remote access on Window 7**](https://github.com/tritemio/PyBroMo/wiki/Howto-remote-access-on-Window-7) * [**Howto setup an IPython cluster**](https://github.com/tritemio/PyBroMo/wiki/Howto-setup-an-IPython-cluster)  Clone this wiki locally   [**Clone in Desktop**](https://windows.github.com/)  This is a wiki page for the***[PyBroMo](http://tritemio.github.io/PyBroMo/)***software  This is a quick howto on the setup of an [IPython](http://ipython.org/) cluster. For more info see the official IPython docs: [Using IPython for parallel computing](http://ipython.org/ipython-doc/dev/parallel/index.html).  Before starting you need to install IPython. The easiest way is to get it through a scientific python distribution, like [Anaconda](https://store.continuum.io/cshop/anaconda/). Parallel computing on a single machineMethod 1 Launch the notebook server and, from the cluster tab, start 4 engines. Method 2 Open a terminal (cmd.exe) and type:  ipcluster start -n 4 Parallel computing on many machines (Windows 7) Reference from IPython docs:   * [Starting the controller and engines on different hosts](http://ipython.org/ipython-doc/dev/parallel/parallel_process.html#starting-the-controller-and-engines-on-different-hosts)   Here we configure 2 machines, one **controller-host** that launch the simulation and one **slave-host**that performs the computation. This procedure can be extended to multiple "slave" machine just repeating this same configuration.  **NOTE for Windows:** All the commands must be pasted in a **cmd.exe** terminal. Setup the controller Only the first time we need to create an IPython profile.  ipython profile create --parallel --profile=parallel  This command copies a new set of configuration files in IPYTHONDIR/profile\_parallel, where IPYTHONDIR is usually a folder named .ipython in the user home folder (C:\Users\username\). These files can be customized to change the default behavior, if needed.  Now, each time we want to start a parallel computation we begin starting the controller:  ipcontroller --profile=parallel --ip=169.232.130.141  where the address is the **controller ip address**.  This command creates a file ipcontroller-engine.json that contains the connection info that the other machines need in order to connect to the controller. The file is located in IPYTHONDIR/profile\_parallel/security.  We need to copy ipcontroller-engine.json to the computation machine. To automate this step I like to link the IPython folder into a Dropbox folder so that all the configuration files are automatically copied/updated on the different machines. Setup the "slave" machine Also on the machine in which we run the computation it's useful to create a profile (only the first time), with the same command as before:  ipython profile create --parallel --profile=parallel  A new set of configuration files is created in IPYTHONDIR/profile\_parallel.  We can start a computation engine with the ipengine command, specifying the path of the ipcontroller-engine.json file:  ipengine --profile=parallel --file=C:\Data\user\software\Dropbox\ipython\profile\_parallel\security\ipcontroller-engine.json  or, we can write the file name in the configuration file so we don't need to write it every time. To do so, edit the file ipengine\_config.py found in the previously created profile folder (IPYTHONDIR/profile\_parallel). Find the line:  #c.IPEngineApp.url\_file = u''  remove the trailing # and write the ipcontroller-engine.json path, in our example:  c.IPEngineApp.url\_file = u'C:\Data\user\software\Dropbox\ipython\profile\_parallel\security\ipcontroller-engine.json'  Now to launch an engine simply type:  ipengine --profile=parallel  It is suggested to launch as many engine as the number of cores. To launch a second engine open a new terminal and type again the command, and so on.  To add another machine for computation just repeat the previous steps. Launching the simulation Once the cluster is started (either in a single machine or on multiple machines) we are ready to launch a simulation.  On the controller machine start an **IPython QtConsole** or an **IPython notebook** using the profile parallel:  ipython qtconsole --profile=parallel  or  ipython notebook --profile=parallel  Then do:  from IPython.parallel import Client  rc = Client()  rc.ids  the last command should print the number of engines that were started.  Alternatively, if you have a QtConsole or Notebook already started without the profile parallel, you can simply specify the path of the file that contains the clients (engines) information. This file is ipcontroller-client.json (not -engines as before!) and is located in the profile folder.  **NOTE**: This trick is used by the **PyBroMo** notebooks so you don't need to restart the notebook server after you launch the cluster. |
| https://github.com/tritemio/PyBroMo/wiki/Howto-setup-an-IPython-cluster  **Controller PC에서 할 일**  ipython profile create --parallel --profile=parallel  ipcontroller --profile=parallel --ip=172.21.101.40   * C:\WinPython\WinPython-32bit-3.5.2.2\settings\.ipython\profile\_parallel\security * 디렉토리에 새로 생성되는 ipcontroller-engine.json 파일을 엔진 PC의 특정 디렉토리에 복사한다.   **Engine PC에서 할 일**  ipython profile create --parallel --profile=parallel (매번 할 필요는 없음)  ipengine --profile=parallel --file=Y:\TeamMember\hccho\ipcontroller-engine.json  ipengine\_cofig.py파일을 편집하여   |  | | --- | | c.IPEngineApp.url\_file = u'Y:\TeamMember\hccho\ipcontroller-engine.json' |   이렇게 하면,   1. ipengine --profile=parallel 2. 또는 Jupyter Note Book “IPython Clusters” 탭에서 엔진 개수를 설정하여 엔진을 열수도 있다. 이때 security 디렉토리에 json파일이 생기는데, 영향을 주지는 않는 듯. “IPython Clusters” 탭에서 보이는 profile은 위의 ipython profile create를 통해서 만든 것들이 보임.   **Client를 실행할 때, profile 이름을 “default”로 하지 말고, “parallel”로 해야 함.**  **rc = Client(profile = "parallel")**   * **TEST 성공** |

**Pandas & DB**

|  |
| --- |
| params = urllib.parse.quote\_plus('PROVIDER=MSDataShape;Data PROVIDER=MSDASQL;dsn=POSIM;uid=;pwd=;')  engine = create\_engine("mssql+pyodbc:///?odbc\_connect=%s" % params) |
| import pymysql  import pandas as pd  BabyData=[('Bob', 968), ('Jessica', 155), ('Mary', 77), ('John', 578), ('Mel', 973)]  df = pd.DataFrame(data = BabyData, columns=['Names', 'Births'])  print(df)  conn2 = pymysql.connect(host='172.21.101.40', user='root', passwd='12345', db='posim')  psql.write\_frame(df, con=conn2, name='hcchotest', if\_exists='append', flavor='mysql') |
| conn = pyodbc.connect('DRIVER={MySQL ODBC 3.51 Driver};SERVER=localhost;DATABASE=posim;UID=root;PWD=12345;') |
| BabyData=[('Bob', 968), ('Jessica', 155), ('Mary', 77), ('John', 578), ('Mel', 973)]  df = pd.DataFrame(data = BabyData, columns=['Names', 'Births'])  print(df)  conn2 = pymysql.connect(host='172.21.101.40', user='root', passwd='12345', db='posim')  df.to\_sql(name='hcchotest',con=conn2,if\_exists='append', flavor='mysql',index=False) |
| sqlalchemy의 create\_engine과 pandas는 아직 맞지 않음   * Pandas 0.19.2 에서는 create\_engine으로 해야만 됨 |
|  |
| import pymysql  import pandas as pd  # connect to mysql  conn = pymysql.connect(host='172.21.102.25', port=3306, user='otcadmin', password='0tcr00t',charset='utf8')  # select database  conn.select\_db('otctrading')  # cursor  cur = conn.cursor()  # query  sql = "select \* from xxxxx”  cur.execute(sql)  # fetch result  rows = cur.fetchall()  # turn table result into dataframe  df=pd.DataFrame(list(rows)) |
| **Pandas 0.19.2 방식**  SQLAlchemy, MySQL\_python(mysqlclient 3.5용), pymysql, wheel 설치/업데이트 해야 함.  import pandas as pd  from sqlalchemy import create\_engine  x1=[3,4,5]  x2=[*'aa'*,*'ccc'*,*'df'*]  x3 =[1971,1975,2000]  z=pd.DataFrame(data=list(zip(x1,x2,x3)),columns=[*'Address'*,*'names'*,*'Births'*])  engine = create\_engine(*'mysql://root:12345@172.21.101.40'*)  engine.execute(*"USE POSIM"*)  z.to\_sql(name = *"hcchotest"* ,con = engine, if\_exists=*'append'*, index=False)  engine.dispose() |

|  |
| --- |
| **dataframe만들기**  import pandas as pd  x1=[3,4,5]  x2=['aa','ccc','df']  x3 =[1971,1975,2000]  df=pd.DataFrame(data=list(zip(x1,x2,x3)),columns=['age','name','year'])  (\*\*\* zip은 list를 tuple로 묶어서 새로운 list를 만들어준다)  df.ix[df['name']=='aa', 'year' ] = 1972 # data수정  #row를 추가할 때는 새로운 dataframe을 만들어서 추가  df=df.append(pd.DataFrame([[7,'xx',2002]],columns=['age','name','year']))  **df.head() 🡨-첫 다섯개 row**  **df.head(10) 🡨- 10개 row**  **df.columns 🡨-field명**  **df.ix[2] 🡨-2번째 행(칼럼이름 포함)**  **df.values 🡨---- list를 return**  **df.values[2] 🡨-칼럼이름 없이 2번째 행**  **df.ix[n:m] 🡨- n부터 m-1까지**  **df.ix[:] 🡨전체**  **df.index = df.name 🡨-‘name’ 칼럼을 index로 지정. 이름만 지정할 때는 z.index.name="myindx"**  **df.ix[‘aa’] 🡨- index로 지정된 name이 ‘aa’인 행**  **df.ix[i,j] 🡨-df table의 i행,j열 (0-based index)**  **df.ix[: , 3] 🡨-3번째 열**  len(z) 🡨-행 개수  len(z.columns) 🡨열 개수 |
| **df = pd.DataFrame({'a':[1,3,5,7,4,5,6,4,7,8,9],**  **'b':[3,5,6,2,4,6,7,8,7,8,9]})**  **print(df)**  **A=df['a'].values.tolist()**  **A[1]=100**  **print(df)** |
| **C:\Users\영업\Desktop\001.jpg**  **dict형으로부터 dataframe을 만들 때는 data형태가 list야 함.** |
| |  |  | | --- | --- | |  | **loc:** only work on index **iloc:** work on position **ix:** You can get data from dataframe without it being in the index **at:** get scalar values. It's a very fast loc **iat:** Get scalar values. It's a very fast iloc |   **A=[[2,3,4],[3,3,5],[5,63,4],[3,4,5]]**  **A1=['a1','a2','a3']**  **A2=['x1','x2','x3','x4']**  **df=pd.DataFrame(data=A,index=A2,columns=A1)**  **print(df)**  **df.loc['x1']**  **df.iloc[0]**  **df.iloc[0,1] 🡨좌표로 접근**  **df.iloc[0:1,0:2]**  **print(df.at['x1','a1'])**  **print(df.at[0,1])** |
| Dataframe filtering  sql = "SELECT \* FROM posimresult"  df = psql.read\_sql(sql, conn)  **df =df[df['SimDate'] == datetime.date(2016, 7, 28)]**  **#df =df[df['SimId'] == '4theta\_SD\_AH\_PDIV\_20160714']**  print(df) |
| import pandas as pd  zerofilestr= 'Y:\TeamMember\hccho\PythonTest\AH5\data\Zero160804.xlsx'  zerofile = pd.ExcelFile(zerofilestr)  valdateser = 42586  dg=zerofile.parse('Ks') #Excel Sheet지정 ==> dataframe  dg=dg[['days','rate']] # days, rate 2개 column만 축출햐여 dataframe만들기. []한쌍이면 series  dg['yf']=dg['days']/365. # yf column 추가  dg['rate']=dg['rate']/100. # rate column 수정  dg['dates']=valdateser+dg['days'] # dates column 추가 |
| def **Do\_SQL**(DbConnecInfo\_ForData, sql):  import pymysql  import pandas.io.sql as psql  conn = pymysql.connect(host = DbConnecInfo\_ForData[*'IP'*], user=DbConnecInfo\_ForData[*'user'*], passwd=DbConnecInfo\_ForData[*'pw'*], db=DbConnecInfo\_ForData[*'dbname'*])  df = psql.read\_sql(sql, conn)  conn.close()  return df |

|  |
| --- |
| **File 읽기**  import pandas as pd  IVfile = "Y:\TeamMember\hccho\PythonTest\AH5\data\Iv160804SdExc.xlsx"  inputfile = pd.ExcelFile(IVfile)  df=inputfile.parse('Hs')  df = df.set\_index(['Mat'])  df=df/100.  df.index.name='0' |
| **Dictionary 🡺 dataframe**  A = {'101HSCEI': 0.6825, 'XIN9I101': 0.35099999999999998, 'SX5E101': 0.46250000000000002, '101XIN9I': 0.35099999999999998, 'SPXSX5E': 0.62590000000000001, '101SX5E': 0.46250000000000002, 'XIN9IHSCEI': 0, 'SPXXIN9I': 0, 'HSCEISPX': 0.3866, 'XIN9ISPX': 0, 'HSCEISX5E': 0.51770000000000005, 'HSCEI101': 0.6825, 'SPXHSCEI': 0.3866, 'HSCEIXIN9I': 0, 'SX5EHSCEI': 0.51770000000000005, 'SX5ESPX': 0.62590000000000001, 'XIN9ISX5E': 0.2258, 'SX5EXIN9I': 0.2258, '101SPX': 0.37530000000000002, 'SPX101': 0.37530000000000002}  **df= pd.DataFrame(list(A.items()),columns=['Date', 'DateValue'])** |
| **Pivot Table**  **Index 지정 후, unstack()** |
| **my\_list = list(df.values)**  **my\_list2 = []**  **for i in range(len(my\_list)):**  **if my\_list[i][0] - Xldate <= 112:**  **my\_list2.append(my\_list[i])**  **else:**  **my\_list2.append([7\*int((my\_list[i][0] - Xldate)/7.0) + Xldate, my\_list[i][1]])**  **dg = pd.DataFrame(my\_list2,columns= ['Date','Value'])**  **dg=dg.groupby('Date').sum()**  **dg=dg.reset\_index() 🡨-새로운 index를 만듬** |
| df[*'Date'*] = df[*'Date'*].apply(lambda x: x if x - Xldate <= 112 else 7\*int((x - Xldate)/7.0) + Xldate)  df= df.groupby(*'Date'*).sum() |
| Sort 예  import pandas as pd  x1=[3,7,5]  x2=['aa','ccc','df']  x3 =[1971,1975,2000]  df=pd.DataFrame(data=list(zip(x1,x2,x3)),columns=['age','name','year'])  print(df)  dg=df.sort\_values(['age'],ascending=[True])  print(dg) # df는 바뀌지 않음.  df.sort\_values(['age'],ascending=[True], **inplace = True**) # df가 바뀜 |
| Data Sample  import pandas as pd  a1=[1,90,180,270,360,540,720,900,1080]  a2=[1.474,1.494,1.521,1.581,1.604,1.654,1.739,1.784,1.83]  df=pd.DataFrame(data=list(zip(a1,a2)),columns=['days','rates'])  print(df) |
| 모든 행 또는 열 출력, 출력 숫자 자리수 설정  pandas.set\_option('display.max\_columns', None)  pandas.set\_option('display.max\_rows', None)  pd.set\_option('precision', 5) |

**IP address**

|  |
| --- |
| import socket  s = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)  s.connect(("gmail.com",80))  print(s.getsockname()[0])  s.close() |
| import socket  print([(s.connect(('8.8.8.8', 53)), s.getsockname()[0], s.close()) for s in [socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)]][0][1]) |
| def **get\_local\_ip**():  import socket  return [(s.connect((*'8.8.8.8'*, 53)), s.getsockname()[0], s.close()) for s in [socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)]][0][1] |

**lmfit을 사용한 Optimization**

|  |
| --- |
| <http://cars9.uchicago.edu/software/python/lmfit/index.html> |
| Levenberg-Marquardt도 가능 |
| # create data to be fitted  x = np.linspace(0, 15, 301)  data = (5. \* np.sin(2 \* x - 0.1) \* np.exp(-x\*x\*0.025) + np.random.normal(size=len(x), scale=0.2) )  # define objective function: returns the array to be minimized  #def fcn2min(params, x, data):  # """ model decaying sine wave, subtract data"""  # amp = params['amp']  # shift = params['shift']  # omega = params['omega']  # decay = params['decay']  # model = amp \* np.sin(x \* omega + shift) \* np.exp(-x\*x\*decay)  # return model - data  def fcn2min(params, x, data):  """ model decaying sine wave, subtract data"""  v = params.valuesdict()  model = v['amp'] \* np.sin(x \* v['omega'] + v['shift']) \* np.exp(-x\*x\*v['decay'])  return model - data      # create a set of Parameters  params = Parameters()  params.add('amp', value= 10, min=0)  params.add('decay', value= 0.1)  params.add('shift', value= 0.0, min=-np.pi/2., max=np.pi/2)  params.add('omega', value= 3.0)  # do fit, here with leastsq model  minner = Minimizer(fcn2min, params, fcn\_args=(x, data))  kws = {'options': {'maxiter':10}}  result = minner.minimize()  # calculate final result  final = data + result.residual  # write error report  report\_fit(result)  # try to plot results  try:  import pylab  pylab.plot(x, data, 'k+')  pylab.plot(x, final, 'r')  pylab.show()  except:  pass |

**File Server만들기**

|  |
| --- |
| IIS(인터넷 정보 서비스) 관리자 |
| “프로그램 제거 또는 변경” – Windows기능 사용 -🡪 “인터넷정보 서비스” 추가 |
| 제어판 – 관리도구 –IIS 관리자 |
| 사이트 -> 웹사이트 추가 |
| 가상디렉토리를 추가할 수 있다. 이때 “C:\inetpub\wwwroot”에 있는 web.config 파일은 없애야 함 |
| MIME형식에 필요한 파일의 확장자 추가해야 됨. ( .\* application/octet-stream ) |
| 오른쪽버턴 🡪 웹사이트 관리 -> 고급설정 |
|  |
| http://172.21.101.40/Data/simple.dat |

**Visual Studio Project 파일**

|  |
| --- |
| xxx.vcxproj 파일과 cpp, h 파일만 있으면 됨  (sdf, suo, sin파일 필요 없음) |
| Vcxproj 파일 이름을 바꾸면 출력 파일도 맞게 수정됨 |

**XLW**

|  |
| --- |
| XLRegistration::Arg ConcatArgs[] = {  { "string1", "First string", "XLF\_OPER" },  { "string2", "Second string", "XLF\_OPER" }  }; // argument 이름, argumrnt 설명, argument type  XLRegistration::XLFunctionRegistrationHelper registerConcat(  "xlConcat", "Concat", "Concatenate two strings", "xlw Example", ConcatArgs, 2);  // 함수이름, 엑셀에 노출되는 함수 이름, 함수 설명, Library(엑셀 함수 마법사에서 category "xlw Example"), argument(위에서 선언한 XLRegistration::Arg ), argument갯수 |
|  |

**OpenMP**

|  |
| --- |
| #include <omp.h>  #pragma omp parallel for schedule(guided) |
| 프로젝트 – 속성 – C/C++ - 언어 - OpenMP지원 – 예 |
| #include "stdafx.h"  #include <iostream>  #include <omp.h>  using namespace std;  #define DIM 200    void main()  {  double x[DIM];  double y[DIM];  int i;  double xy;  double dot\_product = 0;    for(i=0; i<DIM; i++) {  x[i] = i+10.0;  y[i] = DIM-i;  }    // private(xy): 변수 xy를 각 thread가 공유하지 않는 변수이다. reduction(+:dot\_product); dot\_product의 합은 각 thread가 각각 계산하여 나중에 더하라.  **#pragma omp parallel for private(xy) reduction(+:dot\_product)**  for(i=0; i<DIM; i++) {  xy = x[i]\*y[i];  dot\_product += xy;  }    cout<< "dot product=" << dot\_product << endl;  } |
| #include <stdio.h>  #include <omp.h>  #include <time.h>    **void** main()  {      int max=100000;      int i, j;      int isPrime = 0;      int num\_prime = 0;        clock\_t begin = clock();        #pragma omp parallel for reduction(+:num\_prime) private(isPrime, j) **schedule(guided, 8)**  **for**(i=2; i<max; i++) {            isPrime = 1;  **for**(j=2; j<i; j++) {  **if**(i%j == 0) {                  isPrime = 0;  **break**;              }          }    **if**(isPrime)              num\_prime++;      }        clock\_t end = clock();      printf("연산 시간 %f ms\n", (end-begin)\*1000.0f/CLOCKS\_PER\_SEC);      printf("1과 %d 사이의 소수는 %d개\n", max, num\_prime);    } |
| OpenMP 에서 지원하는 schedule() directive 는 다음과 같다.  **1. schedule(static, chunk갯수)**  모든 블록들의 크기를 같게 나누고 round robin으로 순차적으로 scheduling하며 처리한다.  thread 의 scheduling에 대한 overhead가 적다.  **2. schedule(dynamic, chunk갯수)**  chunk들을 작게 나누어 각각의 thread 에 할당해 주고 작업이 끝난 thread 에게는 새로운 chunk를 할당해 작업한다.  scheduling에 대한 overhead가 크다.  **3. schedule(guided, chunk갯수)**  처음에는 큰 블록으로 일을 나누어 처리하고 점점 블록의 크기를 줄인다.  이 떄 block의 크기는 chunk의 크기보다 작지 않은 크기가 할당된다.  하지만 불행히도 어떤 scheduling이 가장 효율적인지는 알고리즘에 따라 다르기 때문에 알 수 없다.  알고리즘의 특성 그리고 trial and error로 적합한 scheduling 방법을 찾아야 한다. |

|  |
| --- |
| Eclipse Committers Mars  http://kaizen8501.tistory.com/16  **Eclipse에서 Python 개발 환경 구축하기**  **1. Python 설치**  Python은 아래 경로에서 다운로드 할 수 있습니다.  <http://www.python.org/download/>  위 경로에서 현재 사용하고 있는 OS에 맞는 Python을 설치 하면 되며, Python의 버전은 3.x 버전과 2.x 버전이 있으므로 원하시는 버전을 다운로드 하시면 됩니다. 저는 호환성 관계로 Python 2.x 버전을 설치 하였습니다.  **2. PyDev 설치 및 Eclipse 환경 설정**  **(Eclipse Neo3는 다운받아, 압축을 풀어 원하는 디렉토리로 옮기면 됨. 설치 방식이 아님)**  Eclipse에서 Python 개발 환경을 구축하기 위해서는 PyDev 플러그인을 설치 해야 합니다. PyDev는 이름 그대로 python 개발 환경을 위한 플러그인 이며, 설치 방법은 아래와 같습니다  Help -> Install New Software PyDev 사이트를 추가 하면 됩니다. ( Location은 http://pydev.org/updates )  http://cfile5.uf.tistory.com/image/262F1843529AA29A08AB8A  Eclipse에서 PyDev 프로젝트를 생성하여 Python을 사용하기 위해서는 PyDev Interpreter와 설치된 Python의 Link가 필요 합니다. 즉 Eclipse의 PyDev에게 사용 가능한 Python Interpreter가 무엇인지 지정을 해야 합니다.  설정 방법은 아래와 같습니다.  Window->Preferences->PyDev->Interpreter-Python Interpreter 에서 New Button을 선택하여 앞에서 설치한 Python의 설치 경로를 지정 합니다. (저는 Python 2.7을 설치한 관계로 Python 2.7이 설치되어 있는 경로를 지정 하였습니다. )  http://cfile1.uf.tistory.com/image/27438143529AA29C039F02  http://cfile3.uf.tistory.com/image/22632443529AA29C3AD9DB  위와 같이 설정 하면, PyDev에서 설치된 Pythoon의 Library 및 DLL, site-packages들을 자동으로 Link 합니다.  http://cfile25.uf.tistory.com/image/244FB343529AA29D3E41E7  4. Python Project 만들기  File->New->Other->PyDev 를 선택하여 아래와 같이 PyDev Project를 선택 하여 Hello world Project를 생성 합니다.  http://cfile5.uf.tistory.com/image/2120584B529AAAA509609C  앞에서 미리 한 설정 때문인지, Project type 및 Grammar Version은 자동으로 설정 되어 있습니다. 만약 원하는 설정이 아닌 경우에는 아래와 같이 설정 해야 합니다.  http://cfile3.uf.tistory.com/image/2407034B529AAAA611C16E  http://cfile23.uf.tistory.com/image/2659574B529AAAA61F6A03  http://cfile24.uf.tistory.com/image/276EDD4B529AAAA719DCBA  저는 임시로 Main Module을 만들었으며, 정상적으로 Module이 생성 되면 아래와 같이 Hello\_world.py가 생성되면서 간단한 Main 템플릿이 생성 됩니다.  http://cfile6.uf.tistory.com/image/221B614B529AAAA70B34F6  hello\_world.py에 pass를 지우고 print 'hello world'를 입력하고 Ctrl + F11을 누르면 hell\_world.py를 실행 할 수 있습니다.  Ctrl + F11을 누르면 아래와 같이 Save and Launch 창이 출력 되며, 실행 하고 싶은 Python 모듈을 선택합니다.  http://cfile24.uf.tistory.com/image/272A464B529AAAA805A1EC  Python 모듈이 정상적으로 실행 되면 아래와 같이 Eclipse의 Console 창에 hello world가 출력 됩니다.  http://cfile2.uf.tistory.com/image/2250164B529AAAA821E0B6  지금까지 Eclipse 환경에서 Python 프로젝트를 생성하고 간단한 프로그램을 구동하는 예제를 설명 하였습니다. 본 자료가 개발 하시는데 도움이 되었으면 좋겠습니다. |

**Package설치 & upgrade**

|  |
| --- |
| 새로 설치하고 싶은 package가 있을 때  cmd창에서 pip install package\_name  예를 들어, pip install py2exe, pip install pyinstaller  easy\_install 패키지이름   * > **python setup.py install** * pip install [package] --upgrade   easy\_install --upgrade SomePackage  **여기서** [**http://www.lfd.uci.edu/~gohlke/pythonlibs/**](http://www.lfd.uci.edu/~gohlke/pythonlibs/) **wheel을 다운받는다**  **pip install 패키지이름.whl** |

**CUDA Programming**

|  |
| --- |
| <http://www.nvidia.com/object/cuda_home_new.html> 🡺 CUDA Zone 🡺 Training |
| * C/C++ using CUDA C |
| Learn more by:   * Reading the [CUDA C Programming Guide](http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html) * Reading the [CUDA C Best Practices Guide](http://docs.nvidia.com/cuda/cuda-c-best-practices-guide/index.html) * Watching the many hours of [recorded sessions](http://www.gputechconf.com/gtcnew/on-demand-gtc.php) from the gputechconf.com site. * d.Participating in trainings provided at conferences, such as [Supercomputing](http://supercomputing.org/), [International Supercomputing](http://www.isc-events.com/), [GPU Technology Conference](http://www.gputechconf.com/page/home.html), any may others. * Browsing [here](https://developer.nvidia.com/get-started-cuda-cc) for more learning opportunities. |
| <http://haanjack.github.io/cuda/2016-03-27-cuda-prog-model/> |
| https://llpanorama.wordpress.com/cuda-tutorial/ |

Visual Studio Project 생성

|  |
| --- |
| * Win32 Consol 프로제트 생성 * 프로젝트 -> 사용자 지정 빌드 -> Cuda 선택. 없으면 파일(cuda.rules)를 찾아서… * cu 파일을 생성하고, 속성 -> 항목형식 -> Cuda C/C++ * 프로젝트 속성 -> 추가종속성 -> cudart.lib 추가 |

**Rabbitmq + Celery**

|  |
| --- |
| Rabbitmq를 설치하기 전에 Erlang을 먼저 설치해야 됨. Erlang의 실행파일 경로가 path에 설정되어야 함. |
| * path문제로 실패 |

**Machine Learning**

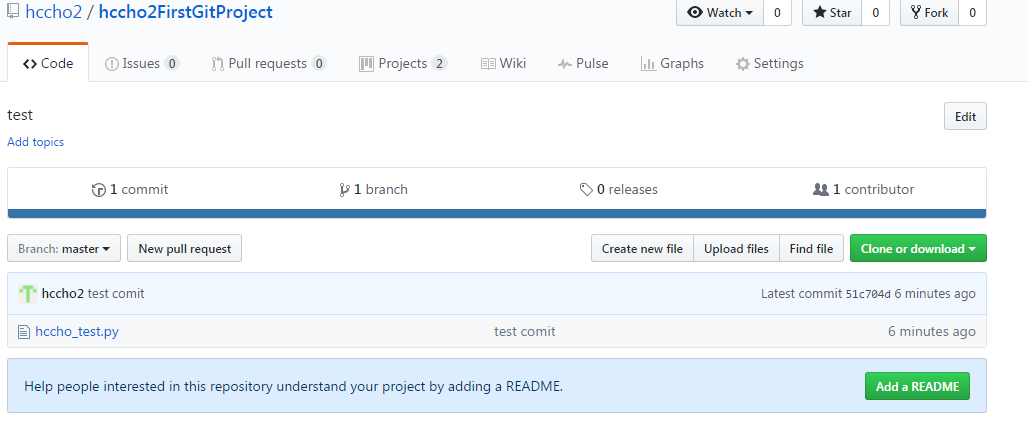
|  |  |
| --- | --- |
| https://youtu.be/BS6O0zOGX4E?list=PLlMkM4tgfjnLSOjrEJN31gZATbcj\_MpUm | 모두를 위한 딥러닝: 김성훈 |
| http://neuralnetworksanddeeplearning.com/ |  |
| <http://ddanggle.github.io/ml/ai/cs/2016/07/16/LearningHowToCodeNeuralNetworks.html>  Welch Labs  https://www.youtube.com/watch?v=bxe2T-V8XRs | 뉴럴네트워크 코드 짜는 법 배우기 본문 내용중에 있는 링크들이 아주 휼륭함. |
| https://tensorflowkorea.gitbooks.io/tensorflow-kr/ | Tensorflow 한글 tutorial MNIST를 비롯한 설명이 있음. |
| CNN( Im2col, col2im )  https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/making\_faster.html |  |
| tensorflow설치  http://goodtogreate.tistory.com/entry/GPU-TensorFlow-on-Window-10-TensorFlow-GPU%EB%B2%84%EC%A0%84-%EC%9C%88%EB%8F%84%EC%9A%B010-%EC%84%A4%EC%B9%98 |  |
| <http://aikorea.org/blog/rnn-tutorial-1/> | Recurrent Neural Network |
| <http://www.wildml.com/2015/09/implementing-a-neural-network-from-scratch/>  https://github.com/dennybritz/nn-from-scratch/blob/master/nn-from-scratch.ipynb | **IMPLEMENTING A NEURAL NETWORK FROM SCRATCH IN PYTHON – AN INTRODUCTION** |

**Anaconda - TensorFlow**

|  |
| --- |
| http://agiantmind.tistory.com/172 |
| Anaconda 설치 => TensorFlow wheel 파일 다운받아 pip로 설치 |

**GitHub**

|  |  |
| --- | --- |
| 디렉토리 변경(이동): cd E:\hccho |  |
| 디렉토리를 먼저 만들고.  Git command창에서 그 디렉토리로 이동  >git init  🡺 .git디렉토리가 생성된다  파일을 만든후 등록하기:  >git add 파일명  Commit을 통해 버전관리를 위한 스냅샷 생성  >git commit –m “test commit” 🡺 디렉토리 같은 개념(?)  >git remote add origin https://github.com/hccho2/hccho2FirstGitProject.git  >git push 🡺 id, pw를 물어보는 창이 뜬다 | 파일을 추가하면 git add부터 반복 |
|  |  |
|  |  |



**C++ simple DLL & VBA call**

|  |  |
| --- | --- |
| 링커-입력-모듈정이파일 def파일명 추가  C/C++ -고급 – 호출 규칙 \_\_stdcall (/Gz) |  |
| #define DllExport extern "C" \_\_declspec(dllexport)  DllExport double KimHT(double x,double y); | a.h |
| #include "a.h"  double KimHT(double x,double y)  {  return x+y;  } | a.cpp |
| LIBRARY MySimpleDLL  EXPORTS  KimHT @1 | a.def |
| Declare Function KimHT Lib "E:/hccho\ForHT/MySimpleDLL/Release/MySimpleDLL.dll" (ByVal x As Double, ByVal y As Double) As Double |  |