참고 포스트

<http://engineerk.com/220689796935> (기초 개념)

<https://blog.naver.com/yuj325/221398730020> (PyCharm과 Anaconda 연동)

<http://ngee.tistory.com/130?category=596100> (파이썬 병렬 및 기본문법들)

<https://matplotlib.org/tutorials/introductory/usage.html#sphx-glr-tutorials-introductory-usage-py> (matplot tutorial)

<https://www.codecademy.com/> 코드 튜토리얼

<https://blog.naver.com/PostView.nhn?blogId=passion053&logNo=220745954177&parentCategoryNo=&categoryNo=27&viewDate=&isShowPopularPosts=false&from=postView> pycharm을 사용한 디버깅

<http://earthpy.org/earthpy-basemap-amazon.html> 여러 그림 예시

<https://oceanpython.org/> python for ocean

- octant - <https://github.com/hetland/octant>

- pyroms - <https://github.com/kshedstrom/pyroms>

- romslab - <https://github.com/rsoutelino/romslab>

- model2roms - <https://github.com/trondkr/model2roms>

- romstools - <https://github.com/trondkr/romstools>

- Python-for-ROMS - <https://github.com/dkobashi/Python-for-ROMS>

Anaconda

1. 여러 라이브러리를 통합관리할 수 있는 패키지
2. 여러 파이썬 버젼, 프로젝트별로 따로 관리 및 실행이 가능함.
3. conda 명령어 와 같은 형태로 사용 (터미널이나 명령 프롬프트에서)
4. conda list (설치된 패키지 리스트)
5. conda create –n 생성할가상환경명 –clone 원본가상환경명
6. window에서, python37 environments 생성 후 matplotlib과 basemap 설치

PyCharm

1. 대중적으로 많이 쓰이는 에디터
2. 유료가 좋지만, 무료인 커뮤니티 버젼도 좋음.
3. 문법 체크 및 프로젝트 관리 용이
4. GitHub 연동
5. Anaconda에서 environments를 만들고, 거기서 만들어진 envs의 python 실행파일을 PyCharm의 existing interpreter로 추가해주면 Pycharm과 Anaconda를 연동하여 사용할 수 있음.
6. 파이썬 3.7은 연동이 제대로 안되는 듯 하고, 파이썬 3.6으로만 연동이 제대로 되는 것으로 보임.

2018 – 12 -21

Basemap 설치 시, conda-forge에서 끌어와야 안정되는 듯.

Basemap \_\_init\_\_ 실행 시 pyproj.pyproj\_datadir 을 인식을 못함.

epsg 파일이 있는 곳을 직접 입력해주니까 됨

윈도우의 환경변수 문제라는 것 같은데, 내컴퓨터 – 설정 - ~~~ 이런식으로 시스템 환경변수 설정해줘도 먹통이었음

2021-06-28  
IPython : An enhanced interactive Python shell.  
NumPy : Support for large, multi-dimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays.  
SciPy : Library of algorithms and mathematical tools for the Python programming language.  
matplotlib : 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments.

miniconda 설치

conda config –-add channels conda-forge (conda에 많이 사용되는 다운로드 채널 추가)

conda create -n roms\_kyy python=3 ipython numpy scipy netCDF4 (roms\_kyy 독립환경 생성)

conda activate roms\_kyy (roms\_kyy 독립환경 활성화)

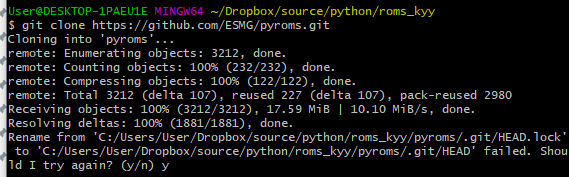
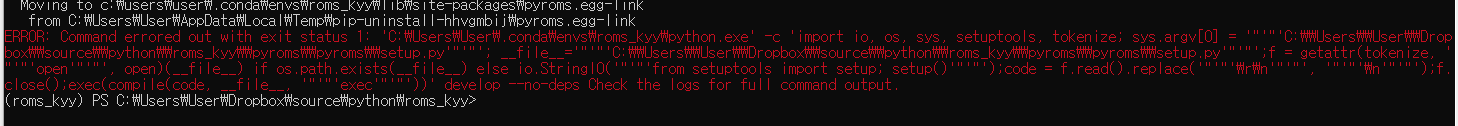
conda install matplotlib basemap basemap-data-hires nco (추가 라이브러리 설치)

conda install cftime lpsolve55 pip (추가 라이브러리 설치)  
conda info --env 로 현재 설치된 독립 환경들 볼 수 있음  
oceanspy? (그림 툴)

SEA-PY -> 필요한 library들 목록 굉장히 많이 정리되어 있음  
  
C:\Users\User\Dropbox\source\python\roms\_kyy 아래에 pyroms 설치

pyroms 설치하려면 git 설치 필요, git 설치 후 git prompt로 저 위치에 pyroms 다운로드

git clone <https://github.com/ESMG/pyroms.git> 다음 명령어로 git prompt에서 설치

- 다음 에러 표출  
  
  
pyroms 설치 시 다음 에러 표출  
pip install -e pyroms/pyroms (git으로 패키지 받은 곳으로 가서 명령어 실행해야 함  


-> 설치 실패  
-> powershell 사용했으나 안됨  
-> powershell 관리자 권한으로 실행해서 받았으나 안됨  
-> pipwin 설치 후 pipwin으로 설치했으나 안됨

-> 위쪽 보니 error: Microsoft Visual C++ 14.0 is required. 라고 되어 있음

-> <https://cofs.tistory.com/388> 보고 설치 – 실패

-> 10 ~ recent version까지 redistributable package set 설치 -> 실패

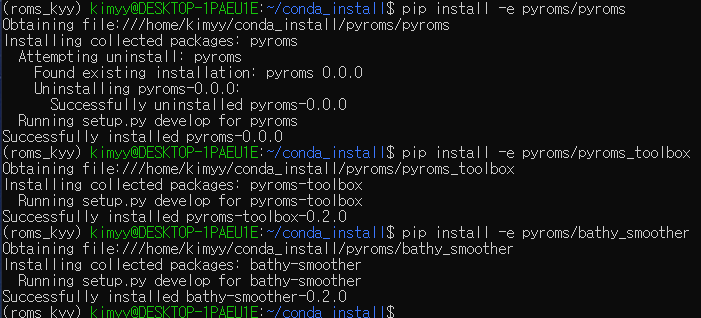
-> 17 visual studio build tools 설치 실패

-> 19 vs build tools 설치 실패

-> 재부팅 후 19 vs build tools는 설치되었으나 (vs 15 build tools는 여전히 설치 안됨) 다른 에러 표출 : error: extension 'pyroms.\_interp' has Fortran sources but no Fortran compiler found

윈도우 우분투에서 miniconda 설치하는 것으로 전략 변경  
wget <https://repo.anaconda.com/miniconda/Miniconda3-py39_4.9.2-Linux-x86_64.sh> 으로 받아서 설치

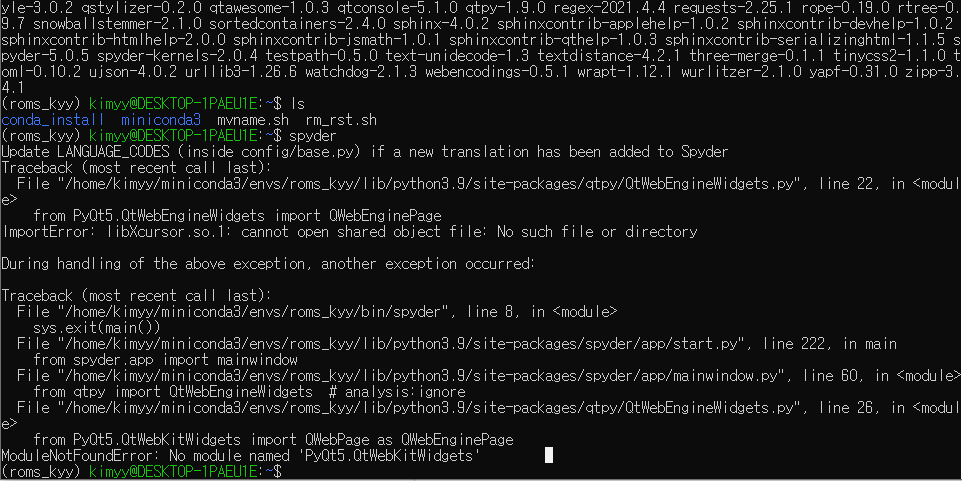
.bashrc 아래에 conda 환경이 올라와 있음

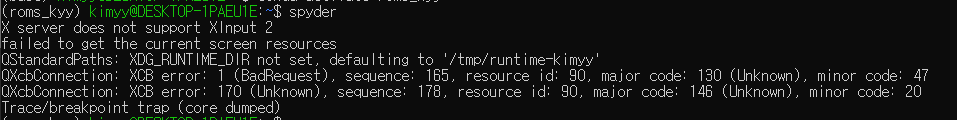
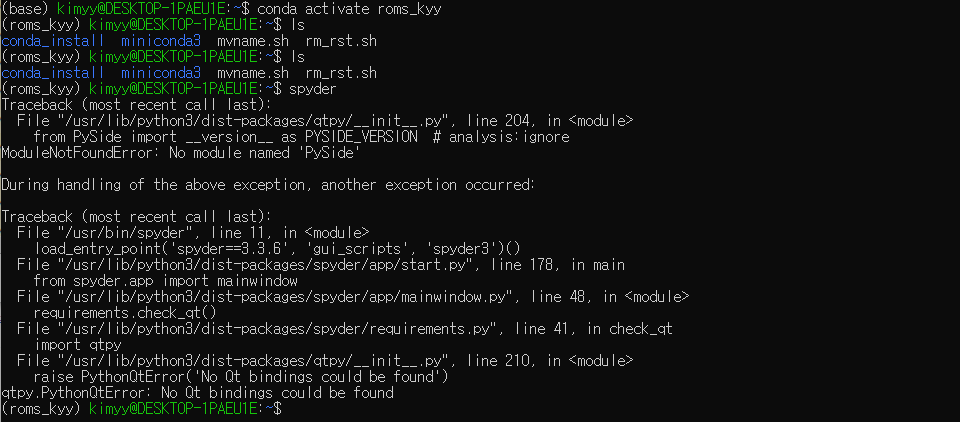
* 그냥 실행할 시 http 에러가 뜨면서 conda-forge에 접근이 안됨. 다른 미러링 사이트로 변경 -> 아닐거같음
* ubuntu를 관리자 권한으로 실행시키면 에러 안뜨고 진행
* git으로 pyroms 다운 및 설치 pip install -e pyroms/pyroms 진행
* fortran compiler 없음 에러 뜸
* sudo apt install gfortran 실행 시   
  Failed to fetch http://security.ubuntu.com/ubuntu/pool/main/l/linux/linux-libc-dev\_5.4.0-42.46\_amd64.deb 404 Not Found [IP: 91.189.88.142 80]  
  에러뜸
* apt-get update -> permission denied와 함께 제대로 안됨
* apt-get update && apt-get install alien 역시 안됨
* sudo 안해서 그럼. sudo apt-get update 하면 잘됨, 그 후 sudo apt install gfortran도 잘됨
* pyroms 설치 완료!
* 
* pip list 로 현재 설치된 모듈 목록 볼 수 있음

2021-06-29

pip install spyder

* 설치는 되는데 에러남



* sudo apt-get install libxcursor1
* libXss 관련 에러가 또 뜸, xwindow를 아예 설치해야될 것 같음
* sudo apt-get install ubuntu-desktop (2기가) , startx --> 에러남
* Xming이 있어야됨
* sudo vi /etc/ssh/ssh\_config 후 ForwardAgent, ForwardX11, ForwardX11Trusted, Port 22 앞의 주석 제거
* 위의 파일에서 XauthLocation /usr/bin/xauth, Protocol 2 도 아래에 추가
* .bashrc에 export DISPLAY=localhost:0 추가
* xclock도 작동하고, spyder를 실행했을 때 splash창까지 뜸, 근데 spyder가 실행이 안됨
* 
* 무언가 multiple window 관련 에러로 추정, VcXsrv 설치해보기로 함
* sudo apt-get install libxcursor-dev also libegl1-mesa libegl1 으로 spyder 구동에 필요한 모듈들 설치
* bashrc에   
  DISPLAY=:0  
  export DISPLAY  
  LIBGL\_ALWAYS\_INDIRECT=1  
  Export LIBGL\_ALWAYS\_INDIRECT  
  추가
* VcXsrv의 Xlaunch 를 이용해 GUI 할 수 있도록 실행. multiple windows 선택 및 Display number는 0으로 설정
* 전보다 에러가 없어지긴 했는데 다음과 같은 에러 출현  
  
* export XDG\_RUNTIME\_DIR='/mnt/c/Program Files/VcXsrv'  
  export RUNLEVEL=3 까지 추가해주니 XDG\_RUNTIME\_DIR 관련 에러도 없어짐  
  근데 Trace/breakpoint trap (core dumped) 에러가 남아 있음
* pip install spyder로 했었는데, 그게 아닌 sudo apt install spyder로 바꿔서 설치
* 혹시 몰라 pip uninstall spyder도 해줌
* 많은 에러 남
* 
* sudo spyder 해도 마찬가지
* sudo apt remove spyder로 지움
* 다시 pip install spyder로 사용, export RUNLEVEL=3 주석함. 똑같이 에러
* VcXsrv 에서 Disable access control 체크해봄 -> 똑같이 에러
* Display number를 0이 아닌 -1로 해봄 -> 똑같이 에러
* conda install pyopengl 해봄 -> 똑같이 에러
* pip uninstall spyder를 하고, conda install spyder로 바꿔봄 -> 똑같은 에러
* VcXsrv 로그를 보면, 역시 뭔가 창 관련 에러가 맞는듯 해 보임.
* export DISPLAY=$(cat /etc/resolv.conf | grep nameserver | awk '{print $2}'):0 로 bashrc 내 DISPLAY 설정을 바꿔봤는데 안됨
* RUNLEVEL을 1이나 4로 바꿔봤는데 되지 않음
* On another note, pip install spyder installs the missing dependencies causing the issues in the other comments. -> pip를 사용한 install은 오류를 불러일으킬 수 있음
* 혹시 몰라서 사람들이 쓰는 것으로 보이는 spyder 3.2.8을 설치해봄 -> 오류
* spyder-test라는 완전히 새로운 virtual environment를 생성하고, 거기서 matplotlib과 scipy만 설치 후 다른 환경변수 설정한 다음 실행 -> 오류
* 그냥 ROMS 서버의 linux에서 miniconda -spyder는 실행 잘됨
* window 용 miniconda에 spyder 설치해서 실행하는것도 잘됨 (pyroms는?)
* full anaconda 설치로 선회. full anaconda도 spyder 실행 잘됨 (pyroms는?)
* 이전에 miniconda로 설치되어있던 roms\_kyy 환경 지우고, 새로 설치, conda install numpy scipy netCDF4 matplotlib basemap basemap-data-hires nco cftime lpsolve55 pip spyder 로 다 설치 + proj4, conda 추가(21.06.30)
* conda install -c conda-forge fortran-compiler 로 포트란 컴파일러도 설치
* pyroms 설치됨, 근데 pyroms/pyroms\_toolbox 설치과정 중 average 코드가 설치가 안됨
* fortran-compiler 지우고, visual studio build tool도 지움
* conda install m2w64-gcc 로 gcc 설치
* conda install -c msys2 m2w64-gcc-fortran 로 gfortran 설치
* 이렇게 gcc를 바꾸면 Visual C++ 14.0 이 필요하다고 나옴
* 다시 visual studio 2019 설치 및 pyroms 들 재설치 -> 전부 완료  
  vim도 설치. create .bat files를 체크해야 powershell에서도 vim을 쓸 수 있음. 그대로 쓰면 vim으로만 실행할 수 있으므로, copy C:\Windows\vim.bat C:\Windows\vi.bat 를 해서 vi로도 실행되게 바꿔야 함  
  anaconda powershell에서는 폴더를 지울 때 rm -Recurse -Force 와 같이 지워야 함

<https://www.git-scm.com/book/ko/v2/> 내용 참고

git   
시간순으로 프로젝트의 스냅샷을 저장  
거의 모든 명령을 로컬에서 실행 (네트워크에 접속하고 있지 않아도 커밋 가능)  
Git으로 무얼 하든 Git 데이터베이스에 데이터가 추가됨. 되돌리거나 데이터를 삭제할 방법은 없음. 커밋하고나면 데이터를 잃어버리기 어려움  
Modified -> Staged -> Committed -> push  
Modifed : 수정만하고 로컬 데이터베이스에 커밋하지 않은 것

Staged : 커밋할 것이라고 표시한 것  
Committed : 로컬 데이터메이스에 안전하게 저장됨  
Push : 리모트 데이터베이스에 커밋된 파일 올림

origin : remote repository의 별칭

git config로 설정 내용을 확인하고 변경할 수 있음

**~/.gitconfig**, ~/.config/git/config : 현재사용자에게만 적용되는 설정. git config --global 옵션으로 이 파일을 읽고 쓸 수 있음. 특정 사용자의 모든 저장소 설정에 적용

.git/config : .Git 디렉토리에 있고, 특정 저장소(현재 작업 중인 프로젝트) 에만 적용. --local 옵션 사용시 이 파일 사용하도록 지정 (기본)  
설정은 로컬일수록 우선  
  
git config --global user.name “Yong-Yub KIM”  
git config --global user.email kimyy308@snu.ac.kr   
git config --global init.defaultBranch main (로컬 브랜치는 master, 웹은 main이라 달라서 불편한데 한번 main으로 바꾸면 신경쓰지 않아도 됨)  
git config --global alias.ci commit : git ci 만으로도 commit할 수 있게 alias  
git config --global alias.unstage 'reset HEAD --' : unstage로 변경하는 명령을 alias

git config --global alias.last 'log -1 HEAD' : 최근 커밋 쉽게 확인  
  
**git config --list** 로 설정 내용 확인 가능  
  
git help 로 도움말. git help config 라면 config에 대한 도움말

**git init** 프로젝트의 디렉토리로 이동 후, 실행하면 .git이라는 하위 디렉토리를 만듦  
  
**git clone** <https://github.com/kimyy308/SSH_KHOA.git> : 웹상의 저장소를 Clone 하는 것. [user@server:path/to/repo.git](mailto:user@server:path/to/repo.git) 와 같이 SSH 프로토콜을 사용할 수도 있음  
  
**git status** : 현재 git 상태 확인. tracked -> 커밋 명령을 실행하면 커밋될 파일들, untracked -> 커밋 명령 실행해도 커밋되지 않는 파일들

**git add** 파일이름 : modified 상태의 파일을 stage에 올림 (커밋 전 단계)

.gitignore : 이 파일안에 적힌 파일들은 git이 관리하지 않음. 자세한 설명은 책 참고  
**git commit** : 커밋. 첫 줄에 적고싶은 메시지를 적고 저장해야 커밋이 됨  
git commit -m ‘적고싶은 메시지’ : 인라인 메시지와 함께 커밋

git commit -a : modified 파일들이 자동으로 stage에 추가된 후 commit 됨  
git commit --amend : 기존의 커밋을 재작성 (덮어쓰기)  
사용 예시 (파일 하나 빠뜨렸을 때, 기존의 잘못된 커밋을 새로운 커밋으로 덮어쓰기. 한 개의 커밋으로 기록됨)  
$ git commit -m 'initial commit'

$ git add forgotten\_file

$ git commit --amend  
  
git rm : tracked 상태의 파일을 삭제. --cached 옵션을 주면 파일은 유지, -f 옵션을 주면 워킹 디렉터리 내 파일까지 삭제 ex ) git rm log/\\*.log (와일드카드 사용 시 \ 붙여야)

git mv : 파일 이름 변경 (mv -> git rm -> git add)

**git log** : 커밋 히스토리 조회. 가장 최근의 커밋이 가장 먼저 나옴. 커밋의 SHA-1 체크섬, 저자 이름, 저자 이메일, 커밋 날짜, 메시지를 보여줌.   
-p 옵션을 걸면 각 커밋의 diff 결과를 보여줌   
-2 옵션을 걸면 최근 두 개의 결과만 조회   
--pretty=format 포맷맞춘 결과만 조회   
--since=2.weeks 최근 2주 내 커밋만 조회  
이외에도 옵션 매우 많음

git diff : unstaged 상태인 변경 부분들 비교. git diff --cached 나 --staged는 staged 상태인 파일 변경부분 비교

git reset HEAD <파일이름> : staged 파일을 unstaged로 변경  
git checkout -- <파일이름> : 수정했던 파일을 워킹 디렉토리에 처음 checkout한 내용으로 되돌림  
  
**git remote** : 현재 프로젝트에 등록된 리모트 저장소 확인. 저장소를 clone하면 origin 이라는 리모트 저장소가 자동으로 등록됨. -v 단축이름과 URL을 함께 볼 수 있음

**git remote add <단축이름> <url>** : 기존 워킹디렉토리에 새 리모트 저장소를 쉽게 추가 가능. 한번 추가하면 url 대신 단축이름을 사용할 수 있음

git fetch <remote> : 로컬에는 없지만, 리모트 저장소에는 있는 데이터를 모두 가져옴. 그러면 리모트 저장소의 모든 브랜치를 로컬에서 접근할 수 있음. 데이터를 가져오긴 하나 자동으로 merge는 안하므로, 수동으로 해야 함

**git pull** **<remote> <branch>** : 리모트 저장소 브랜치에서 데이터를 가져올 뿐만 아니라 자동으로 로컬 브랜치와 Merge 시킬 수 있음. 가장 중요. 작업 시작 전에 remote 서버와 동기화 시켜줘야 함

**git push <remote> <branch>** : upstream 저장소에 push. 리모트 저장소에 쓰기 권한이 있고, clone하고 난 이후 아무도 upstream 저장소에 push하지 않았을 때만 사용할 수 있음. 다른 사람이 push 한 후에 push 하려면 안됨. 먼저 다른 사람이 작업한 것을 가져와서 merge한 후에 push 가능

git remote show <remote> : 리모트 저장소의 구체적인 정보 확인  
git remote rename <단축이름> <바꿀이름> : 리모트 저장소 이름 변경

git remote rm <remote> : 리모트 저장소 삭제

git -a v1.4 -m “메시지” : annotated 태그 작성

git show : 태그 정보 및 커밋 정보 모두 확인  
예전 커밋에 대해서도 태그 가능. git log --pretty=oneline 후,   
git tag -a v1.2 9fceb02 이런식으로 명령의 끝에 체크섬 명시해서 태그하면 됨

git push <branch> <태그 이름> 태그를 리모트 서버에 전송  
  
git branch <branch이름> : 브랜치 만들기

git branch -d <branch이름> : 브랜치 삭제  
**git checkout <branch이름>** : 다른 브랜치로 이동

예시 :   
git checkout testing 으로 testing 브랜치로 이동 후

git commit -a -m ‘made a change’ 로 커밋  
git checkout master 로 다시 기존 브랜치로 돌아옴  
이렇게 되면 임시로 수정한 파일을 커밋하고, 이전에 작업하던 시점으로 워킹디렉토리 파일들을 되돌려서 다시 작업을 진행할 수 있음. 이후 master로 다시 커밋을 하면, 분리되어 히스토리가 진행되고 이후에 Merge를 통해 branch를 합칠 수 있음

git log --oneline --decorate --graph --all 와 같은 형태로 브랜치와 커밋의 히스토리를 깔끔하게 볼 수 있음

git checkout <master> 후 git merge <hotfix> : hotfix가 최근 branch인데, master를 hotfix가 가리키는 커밋으로 이동시키는 방식으로 merge. 이후 git branch -d hotfix로 master만 남기면 hotfix는 없애고, master를 갱신할 수 있음.

만약 서로 다른 길을 가고 있던 branch들을 merge 시킬 경우, 별도의 커밋(Merge 커밋)이 생기며, 현재 checkout 중인 branch가 Merge 커밋을 가리키도록 이동한다. 만약 다른 분기가 같은 부분을 수정하면 git은 merge하지 못하고 conflict message를 출력한다. git status로 자세한 사항 확인 가능. 자세한 사항은 mergetool 부분 참고  
  
파일명을 변경한다던지, 삭제한다던지 할 때엔 mv나 rm이 아닌 git mv 나 git rm을 하고 commit을 한 다음 적용해야 remote에도 제대로 적용이 됨  
  
KISTI에 git 깔고 동기화한다음 수정해서 재동기화 예제  
**git init  
git config --global user.name “Yong-Yub KIM”  
git config --global user.email** [**kimyy308@snu.ac.kr**](mailto:kimyy308@snu.ac.kr)

**git config --global init.defaultBranch main**

**git config --global alias.unstage 'reset HEAD --'**

**git remote add ssh https://github.com/kimyy308/SSH\_KHOA.git  
git remote -v  
git pull ssh main**

**git status  
vi test\_KISTI  
git add test\_KISTI  
git status  
git commit  
git push ssh main**

편집기에서 자동 포맷팅이 아닌, git에 올릴 때 자동 포맷팅

https://www.daleseo.com/python-black/

2021.06.30.

powershell에서 Start-Process spyder 로 spyder 백그라운드 실행이 가능

git 의 stage에 안올라가는 hostfile을 만들고, 그 hostfile로부터 컴퓨터 정보를 읽어서 작업

spyder에선 처음에 new file로 임시 파일을 만들고 -> 저장해야 내가 원하는 폴더 안에 파일 생김

ctrl +1 로 한 줄이나 블록 주석

“”” “”” 로 여러 문장 주석

이렇게 두 가지 중 하나의 형태로만 선언되어야 함

원클래스이름.set\_abc(선언된클래스이름, 값)

선언된클래스이름.set\_abc(값)

opendrift는 ROMS 모델을 열 때 xarray 안의 open\_dataset을 사용해서 읽는 듯. 여긴 수직 ROMS 격자 설정(Vstretching)이 제대로 안되어 있는 것으로 보임. depth.sdepth 를 고쳐야 될 것으로 보임.

옛날 예제 때 나는 scipy.io.netcdf\_file 을 사용해서 nc파일들을 읽었었음

pyroms는 pyroms에서 io를 읽어들어와서 그걸 사용해 nc 파일들을 읽음. io 내에서는 netCDF4를 사용

pyroms io 및 grid를 사용하면 될 듯?  
Vtrans라는게 있는데, Vtransform이 아니고 Vstretching임. Vstretching>=2 면 Vtransform은 당연히 2이므로, Vstretching에 따라서만 구분하도록 s\_coordinate\_2, 4, 5 이런식으로 vgrid.py안에 짜놨음

get\_lonlat (utility 내) : 인덱스를 넣으면 lon, lat을 반환하도록 해놓음   
get\_ij :반대

find\_nearestgridpoints : lon lat 넣으면 가장 가까운 점 반환

get\_nc\_var : simple 하게 파일 이름과 변수명으로 변수 읽기

KeyError : ‘PROJ\_LIB’ 에러가 남 -> proj4 깔았는데 해결 안됨. conda 깔고 위치 명시해줌(espg 파일이 있는 위치) - 해결

pyroms 에서 \_iso 를 import 하려는데 에러남. (fortran file 인데?)

from . import operation # 현재 패키지에서 operation 모듈을 가져옴

pyroms 사용 시, 설치하면 pyd 파일들이 생김. 이게 아닌 그냥 clone 한 폴더로 작업해서 헷갈리고 있었음

폴더1/폴더2/폴더3/파일 을 git으로 push 하면, git에서 한번에 표출하는데, 폴더1/파일 을 push 해주면 그냥 바로 나오는 것을 보니 문제 없어보임

basemap에서 proj를 할 때 matplotlib의 dedent를 불러와야 하는데, matplotlib 3.3 부터는 dedent를 없애버렸음. basemap은 19년?20년?부터 지원 종료됨. cartopy가 대안 모듈로서 제시되고 있음. 근데 pyroms가 basemap을 쓰고 있는데...?

파일 경로 적을 때 역슬래쉬 하면 제대로 인식 못함. 슬래쉬로 폴더 구분해줘야할 듯  
  
  
  
-은주  
모듈 사용 시 권장 버젼이 있는지 확인하고, 권장 버젼을 쓰는 것이 좋음  
pyroms에서 추천하는 버젼을 쓰는 것이 좋음.  
  
맽랩이랑 달리 라이브러리들이 다 개별적으로 만들어서 뿌리는거니까 python 버전이 최신화 했다고해서 금방 따라잡는 것도 아니고 그러다보면 못따라오는 버전들이랑 충돌이 있어버려서 얼마전에 기덕이 basemap 깔려고했을 때 버전문제 있어서 아예 환경 다시 잡았던 기억이 이씀  
내 환경에서 설치된게 basemap 1.2.0

mpl 3.2.2

python 3.7.0

-현근이형  
파이썬 계열중에서 아웃풋 퀄리티는 pyngl 가 제일 좋은듯  
즈변엔 iris 인가 쓰는 분도 잇더라  
예제 많은거 따지면

Pyngl metpy iris matplotlib  
Pyngl은 ps로 저장한담에 ps2epsi 프로그램으로 eps 변환하면 주변 트리밍도 되고 고퀄로 깔끔하게 나옴  
Ps2epsi는 용량 좀 큰거 빼면 여백 지워주고 그림 확대해도 안깨지고  
[진현근형] [오전 10:00] Jpg 처리할거면

[진현근형] [오전 10:00] 이미지매직 쓰면되는데

[진현근형] [오전 10:00] 리눅스 기본 내장프로그램인ㄷ0

[진현근형] [오전 10:00] convert -trim 인풋 아웃풋  
Eps자체가 맥에 최적화 된거같음