

Programming

프로그래밍, 코딩, 개발을 할 때 필요한 세가지.



Python

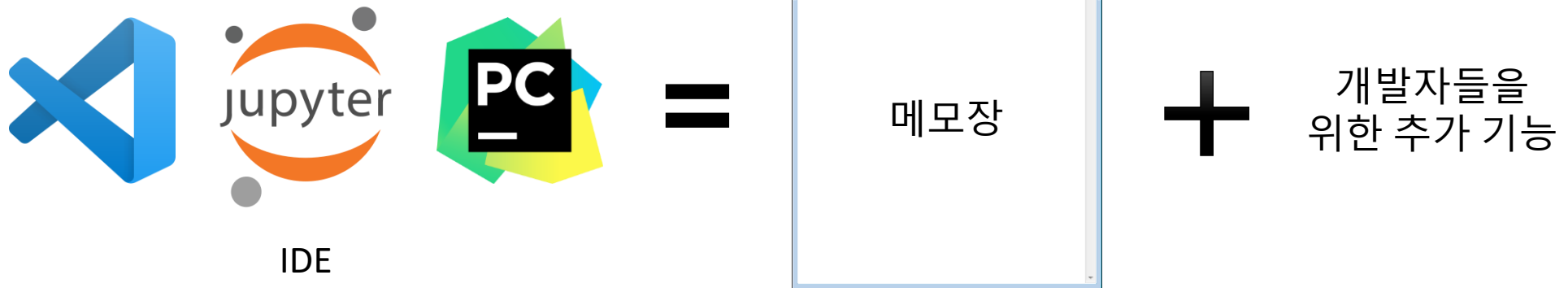


통합개발환경



가상환경

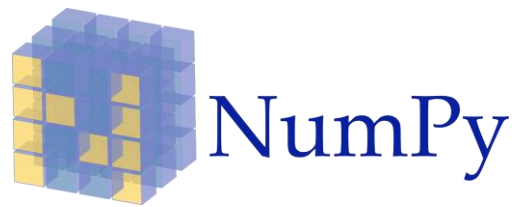
Integrated Development Environment



통합개발환경(IDE)란, 개발자들이 코드를 작성할 때 필요한 여러 기능들을 추가한 프로그램.
수업에서는, **Anaconda Jupyter Notebook**를 사용할 예정.

Virtual Environments

프로그래밍을 잘 하는 사람이 미리 만들어 둔 함수를 사용할 수 있다면?



pip



PyPI에 등록된 Package, 우리는 pip 명령어로 간단히 설치하여 사용 가능.

Ex. pip install tensorflow

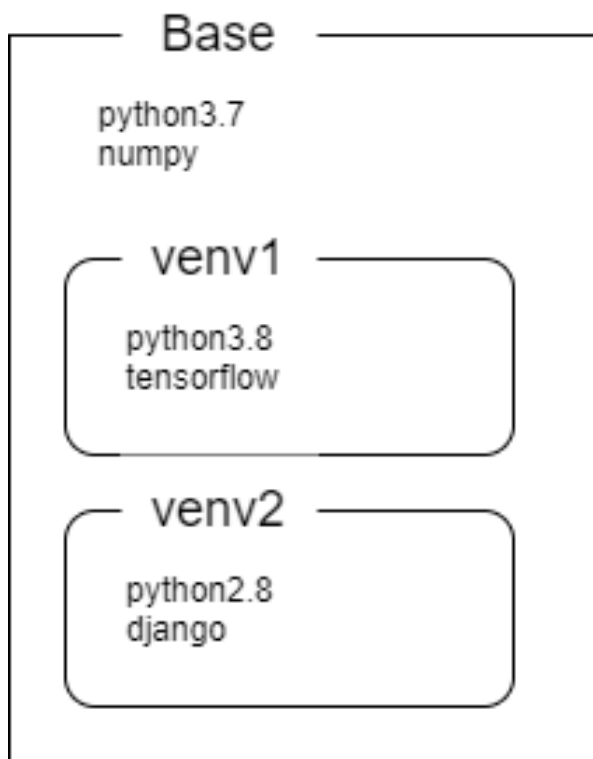
Virtual Environments

```
(base) ~ >>> conda list
# packages in environment at /Users/samueljames/opt/anaconda3:
#
# Name                                Version                                Build      Channel
_ipyw_jlab_nb_ext_conf               0.1.0                                py37_0
alabaster                             0.7.12                               py37_0
anaconda                             2020.02                              py37_0
anaconda-client                      1.7.2                                py37_0
anaconda-navigator                   1.9.12                               py37_0
anaconda-project                     0.8.4                                py_0
applaunchservices                    0.2.1                                py_0
appnope                              0.1.0                                py37_0
appscript                             1.1.0                               py37h1de35cc_0
argh                                  0.26.2                               py37_0
asn1crypto                           1.3.0                                py37_0
astroid                              2.3.3                                py37_0
astropy                              4.0                                  py37h1de35cc_0
atomicwrites                         1.3.0                                py37_1
attrs                                19.3.0                               py_0
autopep8                             1.4.4                                py_0
babel                                 2.8.0                                py_0
backcall                             0.1.0                                py37_0
backports                             1.0                                  py_2
backports.functools_lru_cache         1.6.1                                py_0
backports.shutil_get_terminal_size    1.0.0                               py37_2
backports.tempfile                   1.0                                  py_1
backports.weakref                     1.0.post1                            py_1
```

1. 세가지의 다른 프로그래밍을 한다면,
필요 없는 패키지가 쌓여간다.
2. 현재 진행하고 있는 프로젝트에 필요한
패키지의 **이름과 버전**이 무엇인지 알 수가 없다.
3. 프로젝트 A에서는 Package X의 v2.0가
필요하고 프로젝트 B에서는 v1.0가 필요하다면,
v2.0를 삭제하고 v1.0를 재설치 해야 한다

=> 가상 환경의 등장 배경

Virtual Environments



Package를 저장하는 **독립된 공간**을 만들어,
다른 가상환경 끼리 **서로 영향을 미치지 않도록 한다**

Base에 설치된 Package는 모두가 공유
가상환경을 삭제하면 그 가상환경에서 설치한 모든 Package 삭제

“[pip freeze > requirements.txt](#)” 로, 현재 가상환경에 설치된 Package
이름과 버전을 텍스트 파일로 생성

“[pip install -r requirements.txt](#)”로 텍스트 파일에 저장된 Package를
저장된 버전에 맞게 설치

Windows has been detected and your computer has been shut down to prevent damage to your brain.

DRIVER_JIM_NOT_LESS_OR_EQUAL_WHATEVER_THAT_MAY_MEAN

If this is the first time you have seen this stop error screen, get used to it. You'll probably be seeing it quite a few times in the coming months. (especially Windows 9x users). If you think it'll help, you can try this:

Check to make sure the kettle is on. Tea or coffee should be served ASAP. If this is a new installation, ask your hardware or software manufacturer why they sold you the dodgy products, and if possible, get your money back.

If problems persist, take the cover off your computer and poke various boards with a sharp metal stick. Disable BIOS settings at random, and keep your fingers crossed. You may want to press f8 and enter Safe Mode, but there's no guarantee that'll work either. If all else fails, headbutt the monitor, and run around like a headless chicken.

Below is some unintelligible code, you can go to Microsoft.com and search for the strings but I doubt you'll find anything useful there.

Technical Information:

***STOP: 0x00000001 (0xFC10003F,0x00000002, 0x00000001, 0xf870f80a)

***PatMgr.sys - Address F870F90A base at F870F000, DateStamp 3B7DC5A7

HAVE A NICE DAY:FOXHOUND, NEMESIS:

Windows has been detected and your computer has been shut down to prevent damage to your brain.

DRIVER_JIM_NOT_LESS_OR_EQUAL_WHATEVER_THAT_MAY_MEAN

If this is the first time you have seen this stop error screen, get used to it. You'll probably be seeing it quite a few times in the coming months. (especially Windows 9x users). If you think it'll help, you can try this:

Check to make sure the kettle is on. Tea or coffee should be served ASAP. If this is a new installation, ask your hardware or software manufacturer why they sold you the dodgy products, and if possible, get your money back.

If problems persist, take the cover off your computer and poke various boards with a sharp metal stick. Disable BIOS settings at random, and keep your fingers crossed. You may want to press f8 and enter Safe Mode, but there's no guarantee that'll work either. If all else fails, headbutt the monitor, and run around like a headless chicken.

Below is some unintelligible code, you can go to Microsoft.com and search for the strings but I doubt you'll find anything useful there.

Technical Information:

***STOP: 0x00000001 (0xFC10003F, 0x00000002, 0x00000001, 0xf870f80a)

***PatMgr.sys - Address F870F90A base at F870F000, DateStamp 3B7DC5A7

HAVE A NICE DAY:FOXHOUND, NEMESIS:

Debugging

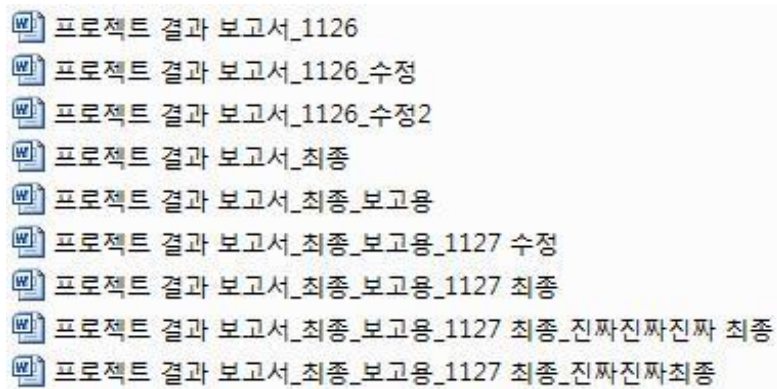
Error의 종류.

1. Syntax Error : 문법적 오류
2. Logical Error : 논리적 오류, 알고리즘 상 문제
3. Runtime Error : 실행 시에 발생 하는 오류

앞으로 만나게 될 오류들을 **대하는 자세와 해결할 방법**을 알고 있는 것이 중요.

=> **오류 메시지 확인 + 알고리즘 검토 + 도와줘요 구글!**

Git



Git란 Version Control System(VCS).

파일이 변할 때 마다 어떤 부분이 바뀌었는지 설명과 함께 저장하는 버전 관리 시스템.

Version이 변화하는 log는 local에 저장되어 있다. Local에 저장된 변화과정을 Remote로 push.

이러한 Git를 협업에 사용한다면, **누가 어떤 부분을 어떻게, 왜 바꾸었는지에 대한 이해를 쉽게 할 수 있다.** 또한, 같은 파일을 동시에 수정하더라도 **충돌을 잘 해결** 할 수 있도록 도와준다.

GitHub



Remote Repository를 만들어주는 서비스

CLI 환경인 git와 다르게 GitHub는 **GUI 환경**을 제공한다

협업에 좋은 기능들을 제공하여, 현재 개발을 하는 데에 있어서 필수적으로 익혀야 하는 Tool

Git Command

- git config <옵션> <설정 항목> <설정 값>

누가 git를 다루게 될지 등록하는 명령어.

```
(base) >>> git config --global user.name "4Moyede"
(base) >>> git config --global user.email "krkdgd9@gmail.com"
(base) >>> cat ~/.gitconfig
[user]
    name = 4Moyede
    email = krkdgd9@gmail.com
```

- Git init

Git Initialization. Version을 등록하기 전에 준비 상태를 만들어 준다.

- Git clone <URL>

Remote의 있는 git를 Local로 가져온다.

```
(base) >>> git clone https://github.com/4Moyede/4Moyede_Blog.git
Cloning into '4Moyede_Blog'...
remote: Enumerating objects: 244, done.
remote: Counting objects: 100% (244/244), done.
remote: Compressing objects: 100% (167/167), done.
remote: Total 244 (delta 93), reused 219 (delta 68), pack-reused 0
Receiving objects: 100% (244/244), 4.53 MiB | 3.34 MiB/s, done.
Resolving deltas: 100% (93/93), done.
(base) >>> █
```

Git Command

- git status

어떤 파일이 add나 commit이 안되어 있는지, 현재 상태를 출력

```
(base) >>> vi views/index.ejs
(base) >>> git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   views/index.ejs

no changes added to commit (use "git add" and/or "git commit -a")
(base) >>>
```

- git add <파일>

이번에 등록할 Version에 추가 혹은 변경할 파일 지정

- git commit -m <Commit 메시지>

Version을 등록할 때에 설명을 작성

```
no changes added to commit (use "git add" and/or "git commit -a")
(base) >>> git add views/index.ejs
(base) >>> git commit -m '[Mod] Blog Navi 임시 삭제'
[master b6a9b53] [Mod] Blog Navi 임시 삭제
 1 file changed, 2 insertions(+), 2 deletions(-)
(base) >>>
```


Git Command

- `git remote add <별칭> <URL>`

Remote Repository의 주소를 등록, 사용될 때는 별칭으로 사용

```
(base) >>> git remote add origin https://github.com/4Moyede/4Moyede_Blog.git
```

- `git push <별칭> <branch>`

Local에 등록한 Version을 Remote로 보낸다

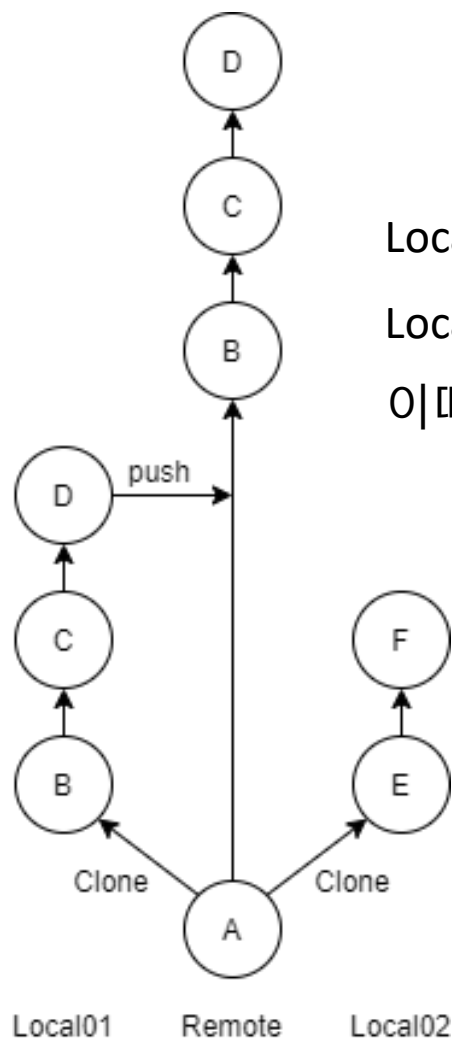
-u 옵션을 사용하면 pull을 한번 하고 진행하여 commit이 꼬이는 것을 방지해준다

```
(base) >>> git push origin main
```

- `git log`

현재 등록된 Version들의 History를 차례대로 보여준다

Git



Local01, Local02에서 각각 clone

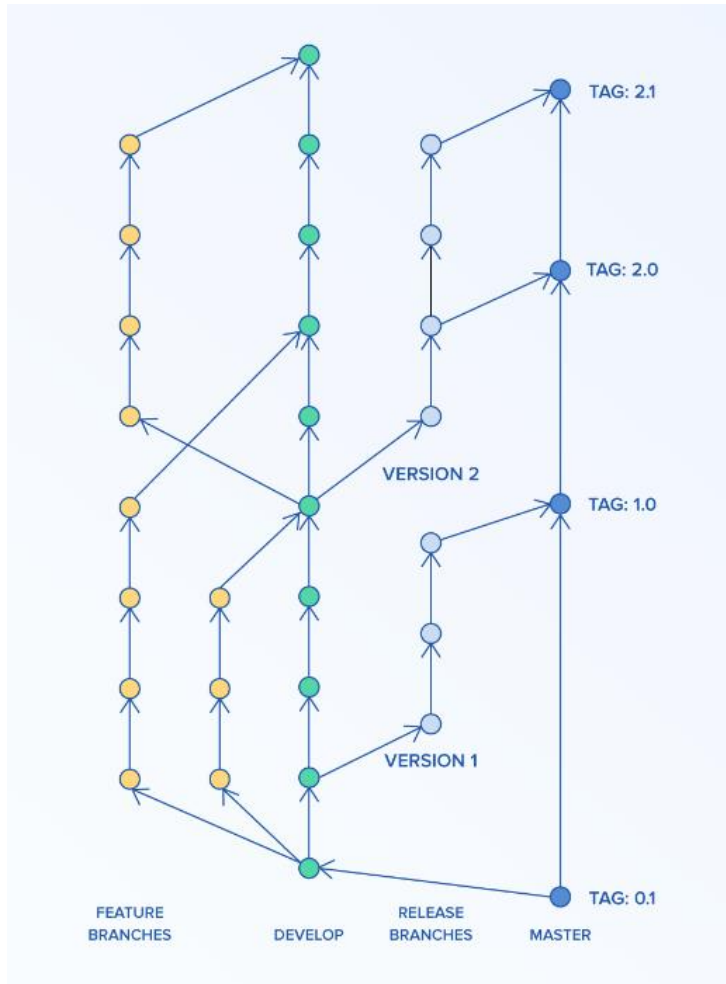
Local01에서 먼저 push를 한다면, Remote의 Commit 순서는 A -> B -> C -> D

이때, Local02에서 A -> E -> F 순서의 log를 push 하려고 한다면 에러

Local02에서 Push하기 전에, Pull을 먼저 한다면

Git가 Local02에서의 순서를 A -> B -> C -> D -> E -> F 로 만들어준다

Branch



분기점에서 다른 Branch가 만들어지면, 다른 공간
다른 Branch에 이름이 같은 파일이 수정되어도 영향이 없다

하나의 Branch는 하나의 목적으로 생성한다
두 개의 Branch가 Merge 될 때에, 충돌 사항이 있는지
체크하고 충돌하는 부분을 어떻게 처리할지 물어본다

Git Command

- git checkout <Branch명>

Branch를 이동, -b 옵션을 사용하면 생성하면서 이동할 수 있다.

```
(base) >>> git checkout -b test  
Switched to a new branch 'test'
```

- git merge <branch>

현재 Branch에 Local내의 다른 Branch를 병합

- git pull <별칭> <branch>

현재 Branch에 Remote Branch를 병합

```
(base) >>> git pull origin master  
From https://github.com/4Moyede/4Moyede_Blog  
* branch          master      -> FETCH_HEAD  
Already up to date.
```