

🚫 TROUBLE SHOOTING 🚫 패키지 매니저 간 충돌

[JPttseenv] 현재 상황

터미널

Kernel - JPttseenv

해결 방법

하나의 패키지 매니저만 사용

가상환경 생성 시에 --no-pip 옵션 사용

환경 변수 관리

가상환경 삭제 및 재생성

해결 방법 실행

하나의 패키지 매니저만 사용

1) 가상 환경 생성

2) 가상 환경 목록 조회

3) 가상 환경 활성화

4) 터미널에서 패키지 확인

ipykernel 설치

설치된 패키지 확인

커널 생성

커널 목록 조회

Kernel Shutdown, jupyter notebook 재실행

커널 선택 후 노트에서 패키지 확인

[JPttseenv] 현재 상황

원인

패키지 매니저 간 충돌

파이썬 인터프리터로 설치된 패키지와 conda를 통해 설치된 패키지 간 버전 충돌 발생

해결

셸에서 가상환경 활성화하면 된다

!source activate JPttseenv

터미널

- JPttseenv 가상환경 활성화

```
source activate JPttseenv
```

- 설치된 패키지 확인

Kernel - JPttseenv

- JPttseenv 커널 선택
- 설치된 패키지 확인 : conda

현재 활성화된 가상환경에 의해 설치된 모든 패키지 나열

```
conda list
```

- 확인할 패키지 항목

	표시된 버전	필요한 버전
Cython	0.29.21	0.29.21
librosa	0.9.1	0.9.1
matplotlib	3.3.1	3.3.1
numpy	1.18.5	1.18.5
phonemizer	2.2.1	2.2.1
python	3.7	3.7.16
scipy	1.5.2	1.5.2
tensorboard	2.3.0	2.3.0
torch	1.6.0	1.6.0
torchvision	0.7.0	0.7.0
Unidecode	1.1.1	1.1.1

conda 를 통해 설치된 패키지를 보여줌

```
!conda list
```

- 확인할 패키지 항목

	표시된 버전	필요한 버전
❌ Cython	0.29.21	0.29.3
❌ librosa	0.9.1	-
matplotlib	3.3.1	3.5.2
❌ numpy	1.18.5	1.21.5
❌ phonemizer	2.2.1	-
❌ python	3.7	3.9.13
scipy	1.5.2	1.5.2
❌ tensorboard	2.3.0	-
❌ torch	1.6.0	-
❌ torchvision	0.7.0	-
❌ Unidecode	1.1.1	1.2.0

- 설치된 패키지 확인 : pip

파이썬 인터프리터 에 의해 설치된 모든 패키지 나열

주로 **pip** 을 통해 설치된 패키지를 보여줌

```
!{sys.executable} -m pip list
```

- 확인할 패키지 항목

	표시된 버전	필요한 버전
Cython	0.29.21	0.29.21
librosa	0.9.1	0.9.1
matplotlib	3.3.1	3.3.1
numpy	1.18.5	1.18.5
phonemizer	2.2.1	2.2.1
❌ python	3.7	-
scipy	1.5.2	1.5.2
tensorboard	2.3.0	2.3.0
torch	1.6.0	1.6.0

torchvision	0.7.0	0.7.0
Unidecode	1.1.1	1.1.1

해결 방법

하나의 패키지 매니저만 사용

- 두 패키지 매니저를 함께 사용하지 않는다
- 'conda'를 사용하여 가상환경을 관리하고, 해당 가상환경 내에서만 패키지를 설치하도록 한다
 - 'pip'을 통해 설치한 패키지를 'conda'를 통해 재설치하거나
 - 'pip'으로 설치한 패키지를 제거한 후 'conda'를 통해 패키지를 다시 설치한다

가상환경 생성 시에 --no-pip 옵션 사용

```
conda create -n 가상환경이름 --no-pip
```

- conda create 명령어를 사용하여 가상환경을 생성할 때 '--no-pip' 옵션을 추가하여 'pip'을 설치하지 않는다
- 이렇게 하면 'pip'이 없는 상태에서 가상환경이 생성되며, 이후 필요한 패키지는 'conda'로만 설치할 수 있다

환경 변수 관리

- 가상환경이나 Jupyter Notebook에서 사용하는 환경 변수가 일관성 있게 설정되어 있는지를 확인한다
- 'PATH' 등의 환경 변수가 다르게 설정되어 있을 경우에는 버전 충돌이 발생할 수 있다.
- 터미널에서 echo \$PATH 명령어를 통해 환경변수를 확인하고 필요에 따라 조정한다

가상환경 삭제 및 재생성

- 문제가 여전히 해결되지 않는 경우, 현재의 가상환경을 삭제하고, 새로운 가상환경을 생성하여 시작할 수 있다
- 이때 '--no-pip' 옵션을 사용하여 'pip'을 포함하지 않도록 하는 것도 고려

해결 방법 실행

하나의 패키지 매니저만 사용

1) 가상 환경 생성

```
conda create -n test python=3.7
```

2) 가상 환경 목록 조회

```
conda info --envs
```

```
(test) ljhp1004@S220:~$ conda info --envs
# conda environments:
#
base                  /home/ljhp1004/anaconda3
JPttsenv              /home/ljhp1004/anaconda3/envs/JPttsenv
test                  *  /home/ljhp1004/anaconda3/envs/test
ttsenv                /home/ljhp1004/anaconda3/envs/ttsenv
```

3) 가상 환경 활성화

```
source activate test
```

4) 터미널에서 패키지 확인

```
conda list
```

```
(test) ljhp1004@S220:~$ conda list
# packages in environment at /home/ljhp1004/anaconda3/envs/test:
#
# Name                        Version                        Build      Channel
_libgcc_mutex                 0.1                            main
_openmp_mutex                 5.1                             1_gnu
ca-certificates               2023.12.12                     h06a4308_0
certifi                       2022.12.7                      py37h06a4308_0
ld_impl_linux-64              2.38                           h1181459_1
libffi                        3.4.4                           h6a678d5_0
libgcc-ng                     11.2.0                         h1234567_1
libgomp                       11.2.0                         h1234567_1
libstdcxx-ng                  11.2.0                         h1234567_1
ncurses                       6.4                             h6a678d5_0
openssl                       1.1.1w                          h7f8727e_0
pip                           22.3.1                         py37h06a4308_0
python                        3.7.16                         h7a1cb2a_0
readline                      8.2                             h5eee18b_0
setuptools                    65.6.3                         py37h06a4308_0
sqlite                        3.41.2                         h5eee18b_0
tk                             8.6.12                         h1ccaba5_0
wheel                         0.38.4                         py37h06a4308_0
xz                            5.4.5                          h5eee18b_0
zlib                          1.2.13                         h5eee18b_0
```

ipykernel 설치

```
conda list ipykernel
```

설치된 패키지 확인

```
conda list
```

#	Name	Version	Build	Channel
	_libgcc_mutex	0.1	main	
	_openmp_mutex	5.1	1_gnu	
	backcall	0.2.0	pyhd3eb1b0_0	
	ca-certificates	2023.12.12	h06a4308_0	
	certifi	2022.12.7	py37h06a4308_0	
	debugpy	1.5.1	py37h295c915_0	
	decorator	5.1.1	pyhd3eb1b0_0	
	entrypoints	0.4	py37h06a4308_0	
	ipykernel	6.15.2	py37h06a4308_0	
	ipython	7.31.1	py37h06a4308_1	
	jedi	0.18.1	py37h06a4308_1	
	jupyter_client	7.4.9	py37h06a4308_0	
	jupyter_core	4.11.2	py37h06a4308_0	
	ld_impl_linux-64	2.38	h1181459_1	
	libffi	3.4.4	h6a678d5_0	
	libgcc-ng	11.2.0	h1234567_1	
	libgomp	11.2.0	h1234567_1	
	libsodium	1.0.18	h7b6447c_0	
	libstdcxx-ng	11.2.0	h1234567_1	
	matplotlib-inline	0.1.6	py37h06a4308_0	
	ncurses	6.4	h6a678d5_0	
	nest-asyncio	1.5.6	py37h06a4308_0	
	openssl	1.1.1w	h7f8727e_0	
	packaging	22.0	py37h06a4308_0	
	parso	0.8.3	pyhd3eb1b0_0	
	pexpect	4.8.0	pyhd3eb1b0_3	
	pickleshare	0.7.5	pyhd3eb1b0_1003	
	pip	22.3.1	py37h06a4308_0	
	prompt-toolkit	3.0.36	py37h06a4308_0	
	psutil	5.9.0	py37h5eee18b_0	
	ptyprocess	0.7.0	pyhd3eb1b0_2	
	pygments	2.11.2	pyhd3eb1b0_0	
	python	3.7.16	h7a1cb2a_0	
	python-dateutil	2.8.2	pyhd3eb1b0_0	
	pyzmq	23.2.0	py37h6a678d5_0	
	readline	8.2	h5eee18b_0	
	setuptools	65.6.3	py37h06a4308_0	
	six	1.16.0	pyhd3eb1b0_1	
	sqlite	3.41.2	h5eee18b_0	
	tk	8.6.12	h1ccaba5_0	
	tornado	6.2	py37h5eee18b_0	
	traitlets	5.7.1	py37h06a4308_0	
	wcwidth	0.2.5	pyhd3eb1b0_0	
	wheel	0.38.4	py37h06a4308_0	
	xz	5.4.5	h5eee18b_0	
	zeromq	4.3.5	h6a678d5_0	
	zlib	1.2.13	h5eee18b_0	

커널 생성

```
python -m ipykernel install --user --name test --display-name "test"
```

커널 목록 조회

```
jupyter kernelspec list
```

```
(test) ljhp1004@S220:~$ jupyter kernelspec list
Available kernels:
  jpttsenv    /home/ljhp1004/.local/share/jupyter/kernels/jpttsenv
  test       /home/ljhp1004/.local/share/jupyter/kernels/test
  ttsenv     /home/ljhp1004/.local/share/jupyter/kernels/ttsenv
  python3    /home/ljhp1004/anaconda3/envs/test/share/jupyter/kernels/python3
```

Kernel Shutdown, jupyter notebook 재실행

| unbuntu 터미널에서, jupyter notebook 터미널이 아님!

- `ctrl + c` : Kernel Shutdonw
- `jupyter notebook`

커널 선택 후 노트에서 패키지 확인

```
!conda list
```

- 결과 : packages in environment at /home/ljhp1004/anaconda3