

Suwoni-Codelab

Python(파이썬) 기본 - 52. 가상환경 - venv, virtualenv, conda

📅 2018-03-21 | 📁 Python 기본

Python의 가상환경에 대하여 정리합니다.

아래 내용은 공부한 것을 정리하므로 틀린 내용이 포함되어 있을 수 있습니다.

1. Python에서의 가상 환경이란?

- 파이썬에서는 한 라이브러리에 대해 하나의 버전만 설치가 가능합니다.
- 여러개의 프로젝트를 진행하게 되면 이는 문제가 됩니다. 작업을 바꿀때마다 다른 버전의 라이브러리를 설치해야 합니다.
- 이를 방지하기 위한 격리된 독립적인 가상환경을 제공합니다.
- 일반적으로 프로젝트마다 다른 하나의 가상환경을 생성한 후 작업을 시작하게 됩니다.
- 가상환경의 대표적인 모듈은 3가지가 있습니다.
 - venv: Python 3.3 버전 이후 부터 기본모듈에 포함됨
 - virtualenv: Python 2 버전부터 사용해오던 가상환경 라이브러리, Python 3에서도 사용가능
 - conda: Anaconda Python을 설치했을 시 사용할 수 있는 모듈
 - pyenv: pyenv의 경우 Python Version Manger임과 동시에 가상환경 기능을 플러그인 형태로 제공

2. venv

- 안타깝지만 venv 모듈은 **pyenv의 버그**로 인해, 제가 현재 작성할 환경이 안됩니다.
 - <https://askubuntu.com/questions/488529/pyvenv-3-4-error-returned-non-zero-exit-status-1>
- pyenv환경이 아닌 분들을 위해 다른 링크로 대체합니다.
 - https://tutorial.djangogirls.org/ko/django_installation/

3. virtualenv

- virtualenv모듈을 사용하려면 pip 명령어로 모듈을 설치해야 합니다.

```
$ pip install virtualenv
```

- virtualenv로 가상환경을 생성합니다.

```
$ virtualenv 가상환경명
```

- 가상환경을 구동해줍니다.
- Mac 또는 리눅스에서 가상환경 구동 방법입니다.

```
$ source 가상환경명/bin/activate
```

- Windows 가상환경 구동방법입니다.

```
가상환경명/Scripts/activate
```

- 가상환경이 구동되면 터미널창의 프롬프트가 아래와 같이 변경됩니다.(터미널에 따라 약간 모양이 다를 수 있습니다.)

```
(가상환경명) $
```

- 이렇게 가상환경으로 변경된 상태에서 pip 명령어로 패키지를 install하면 별도의 격리된 공간에 라이브러리가 설치됩니다.
- 가상환경을 빠져나오는 명령어

```
$ deactivate
```

3. conda

- conda 역시 pyenv 환경에서는 정상적으로 가상환경이 격리되지 않았습니다. 아래 설명은 pyenv가 아닌 환경에서만 유효합니다.
- conda는 사실 가상환경만을 구성하는 모듈은 아닙니다.
- anaconda python의 정보를 확인하고, pip와 마찬가지로 패키지 인스톨 기능이 있으며 그리고 가상환경을 설치할수 있는 기능이 있습니다.
- conda 명령어로 가상환경을 만들어 보겠습니다.
- create 가 가상환경을 만드는 서브명령어이며 -n 이 이름을 명시하는 옵션입니다.

```
$ conda create -n 가상환경명
```

- 가상환경을 활성화 합니다.

```
$ source activate 가상환경명
```

- 가상환경을 비활성화 합니다.

```
$ source deactivate
```

4. pyenv

- 사실상 pyenv를 설치하게 되면 conda, venv는 잘 격리되지 않는 것 같습니다. 혹시 다른 정보나 틀린 정보있으면 제보 부탁드립니다.
- pyenv는 파이썬 버전관리 틀이면서 동시에 가상환경을 플러그인 형태로 제공합니다.
- pyenv 플러그인을 설치해보겠습니다.
- 먼저 pyenv-virtualenv git repository를 클론하고 bash_profile에 정보를 추가합니다.
- zsh 사용시 .bash_profile 대신에 .zshenv 을 입력합니다.

```
$ git clone https://github.com/pyenv/pyenv-virtualenv.git ~/.pyenv/plugins/pyenv-virtualenv
$ echo 'eval "$(pyenv virtualenv-init -)"' >> ~/.bash_profile
```

- 위에 내용을 적용하기 위해 아래명령어를 입력합니다.

```
$ source ~/.bash_profile
```

- 플러그인 설치가 완료되었습니다.
- pyenv에서 가상환경을 설정해봅니다.
- 파이썬인터프리터명 은 pyenv versions 에 나오는 설치되어있는 파이썬 인터프리터 명칭을 입력합니다.

```
$ pyenv virtualenv 파이썬인터프리터명 가상환경명
```

- 가상환경을 활성화 합니다.

```
$ pyenv activate 가상환경명
```

- 가상환경을 비활성화 합니다.

```
$ pyenv deactivate
```

- * 파이썬인터프리터명을 anaconda 시리즈로 지정할 시에는 conda activate 명령어로 가상환경 활성화가 가능합니다.

Python # 파이썬 # venv # conda # virtualenv

좋아요 공유하기 친구들이 무엇을 좋아하는지 알아보려면 가입하기

< Python(파이썬) 기본 - 53. Json 직렬화(Serialize) 및 역직렬화 Python(파이썬) 기본 - 51. pip - 패키지 매니저 >

6개 코딩인강, 492,000원
codeit - 코드잇

코딩의 탄탄한 기반을 다져 줄 6개 강
할인가, 492,000원. 7월 2일까지 할인

댓글 0건 suwoni-codelab.com

박종익 ▾

♡ 추천 Tweet f 공유

인기순 ▾



토론 시작

1등으로 댓글 달기

SUWONI-CODELAB.COM의 다른 댓글.

Python(파이썬) 기본 - 43. class 정리 - 정적메소드 @classmethod와 @staticmethod의 정리

댓글 한 건 • 일년 전



Dong-ryull Shin — * 파이썬에서는 다른언어와는 다르게 정적메소드임에도 불구하고 인스턴스에서도 접근이 가능합니다. 이 차이에 유의해야 합니다. 위의 문장들의 의미가 다음과 같은 의미인지 궁금합니다.* classmethod

Python(파이썬) 기본 - 40. itertools 모듈과 iterable에 유용한 내장함수

댓글 한 건 • 일년 전



D. Test — thousand_primes = islice((x for x in count() if is_prime(x)),1000) 는 0 부터 999까지의 수 중에 소수인 수를 반환하는 iterator가 아니라1000개의 소수를 반환하는 iterator인 듯 하네요.

05. 리눅스(CentOS) 개발 놀이터 만들기 - Samba 설치

댓글 5건 • 일년 전



안녕안 — 강의 잘 보고있습니다.올려주신대로 진행했는데 제 아이피네트 워크로 접속하면 아무 폴더도 생기질 않네요 ππ samba 폴더를 어떻게 해야 만들 수 있을까요?

03. 리눅스(CentOS) 개발 놀이터 만들기 - 몇가지 보안 조치

댓글 2건 • 일년 전



최수원 — CentOS 7 설치하실때 인프라로 설치를 하셨는지요? 그러면 semanage가 있어야 할듯 합니다.

© 2019 ♥ Suwon Choi

Powered by [Jekyll](#) | Theme - [NexT.Muse](#)