Fastcampus Computer Science SCHOOL

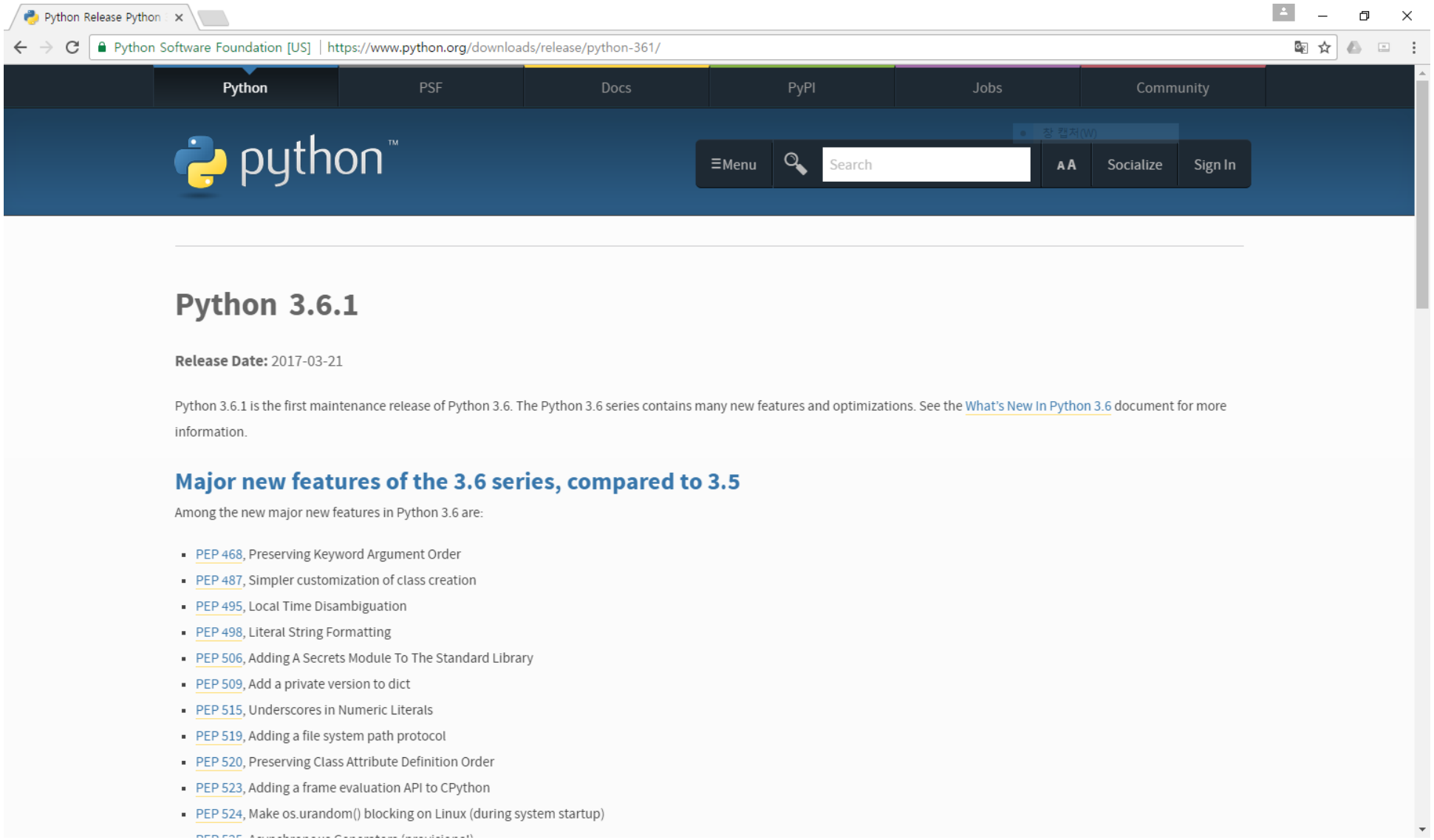
How to Install Python

How to Install Python

How to Install Python - Windows

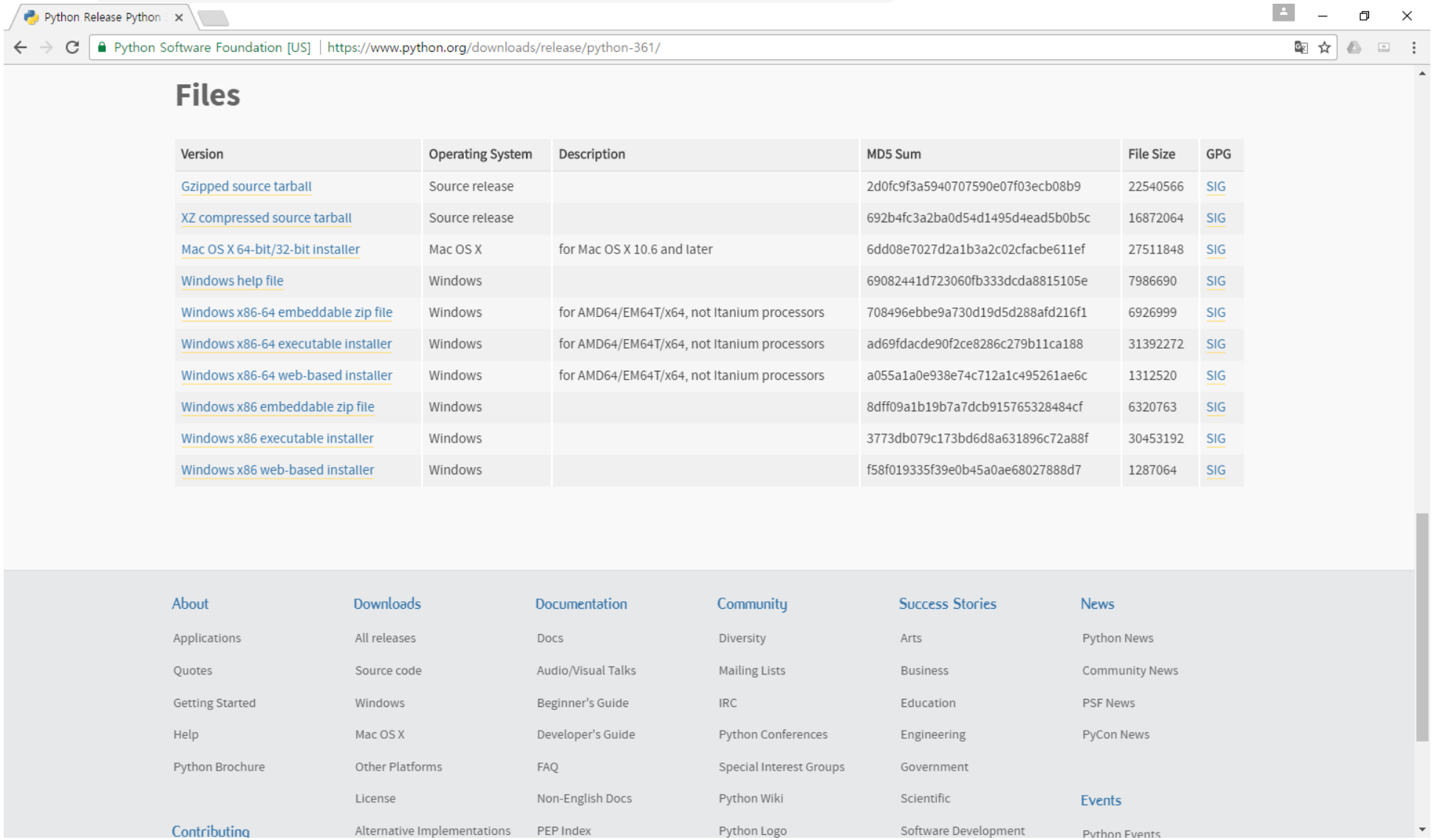
How to Install Python - Windows

python download link



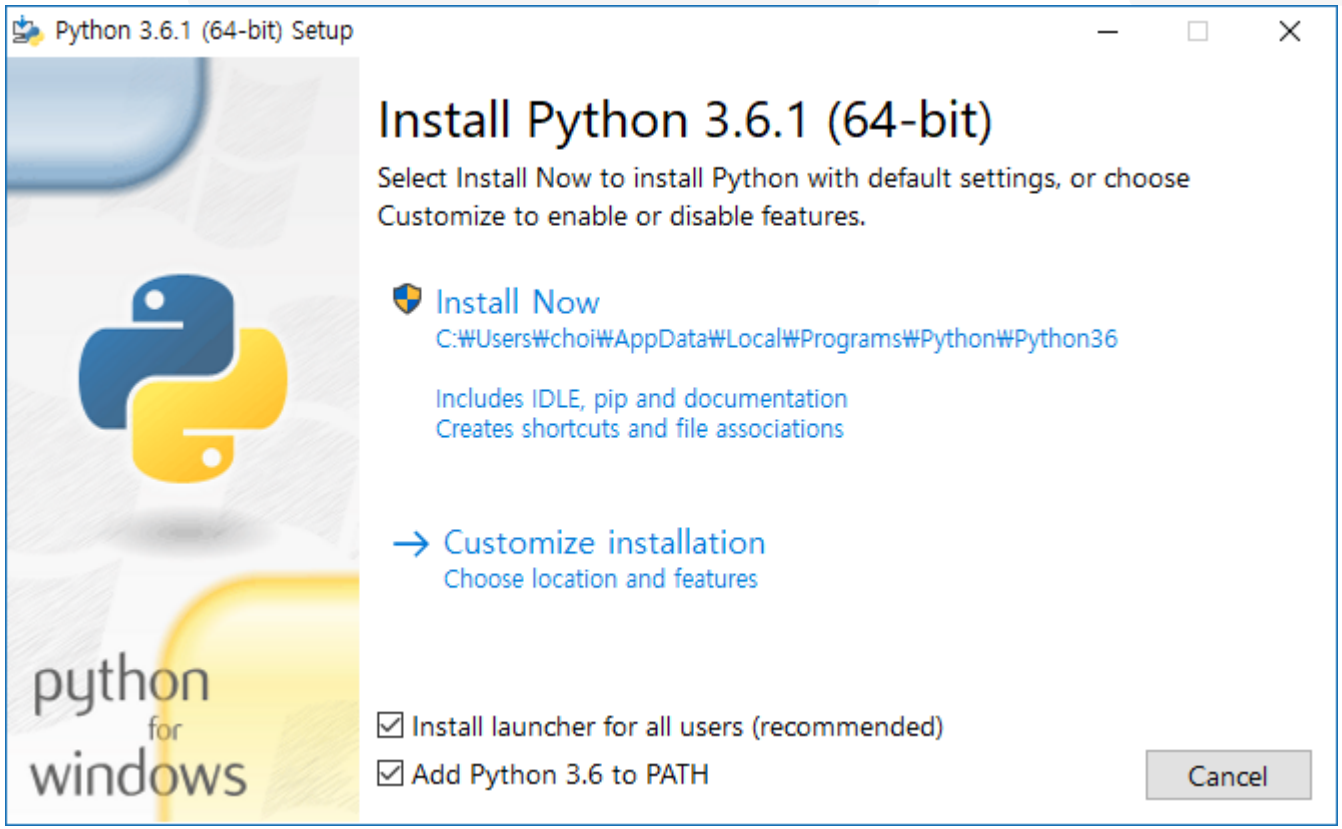
How to Install Python - Windows

click `windows x86-64 executable Installer`



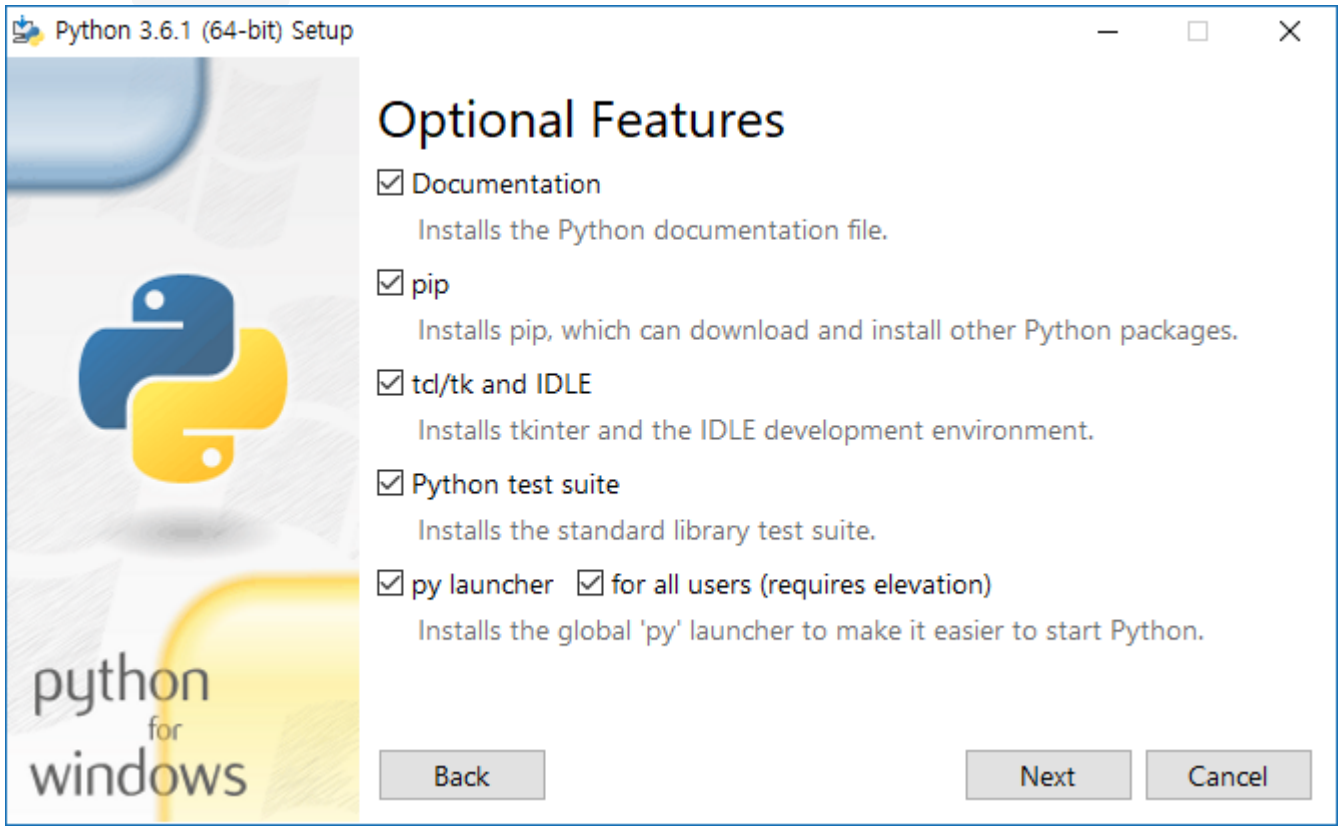
How to Install Python - Windows

enable `Add Python 3.6 to PATH` and click `Customize installation`



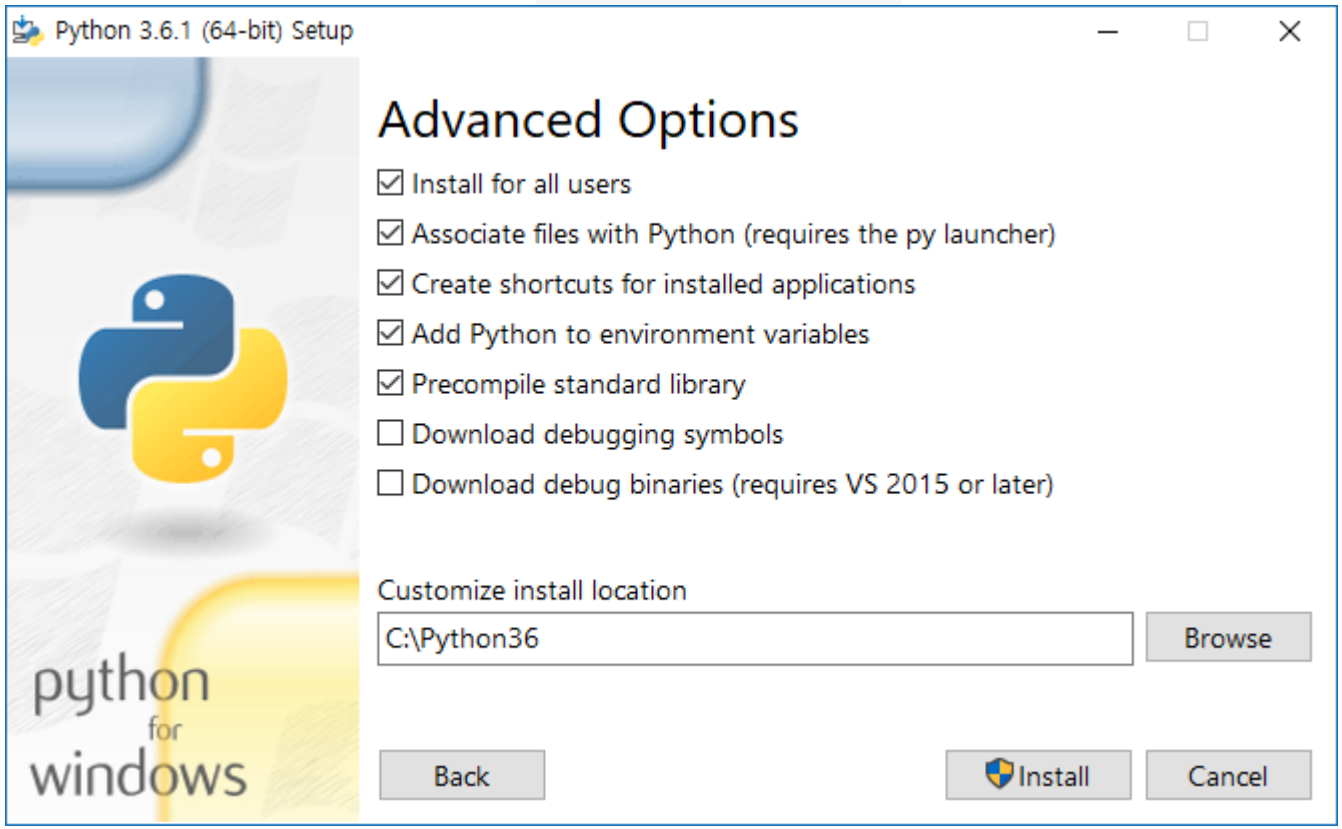
How to Install Python - Windows

click `Next`

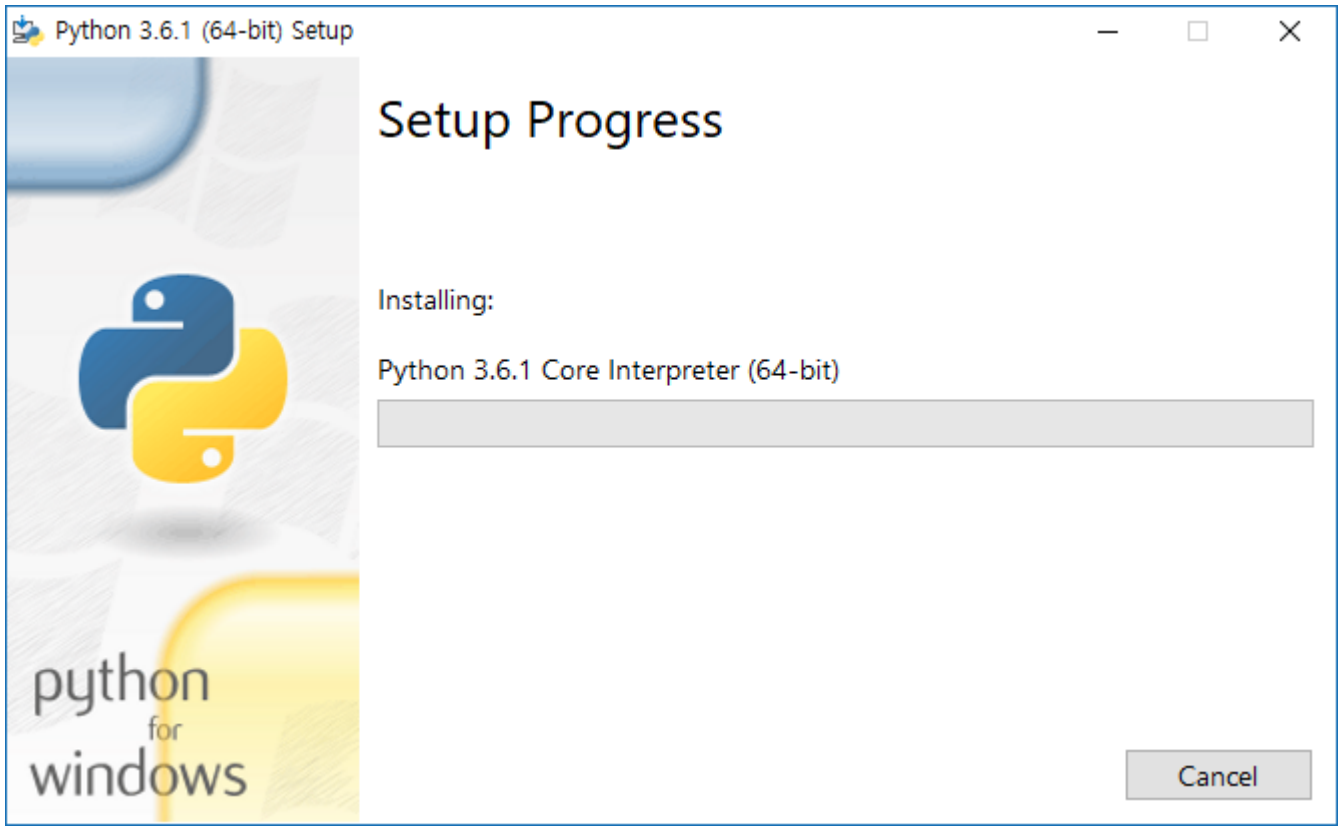


How to Install Python - Windows

Set install location to `C:\Python36`

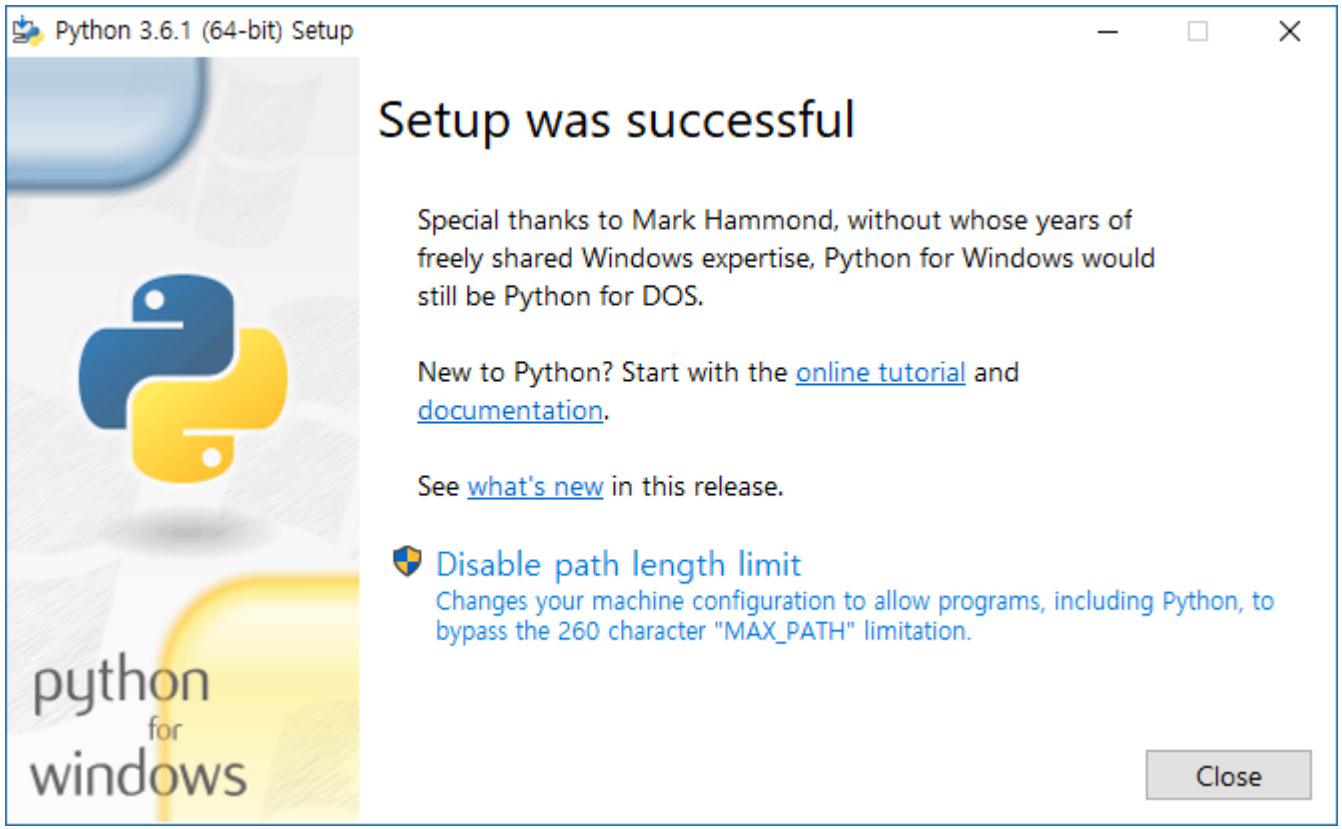


How to Install Python - Windows

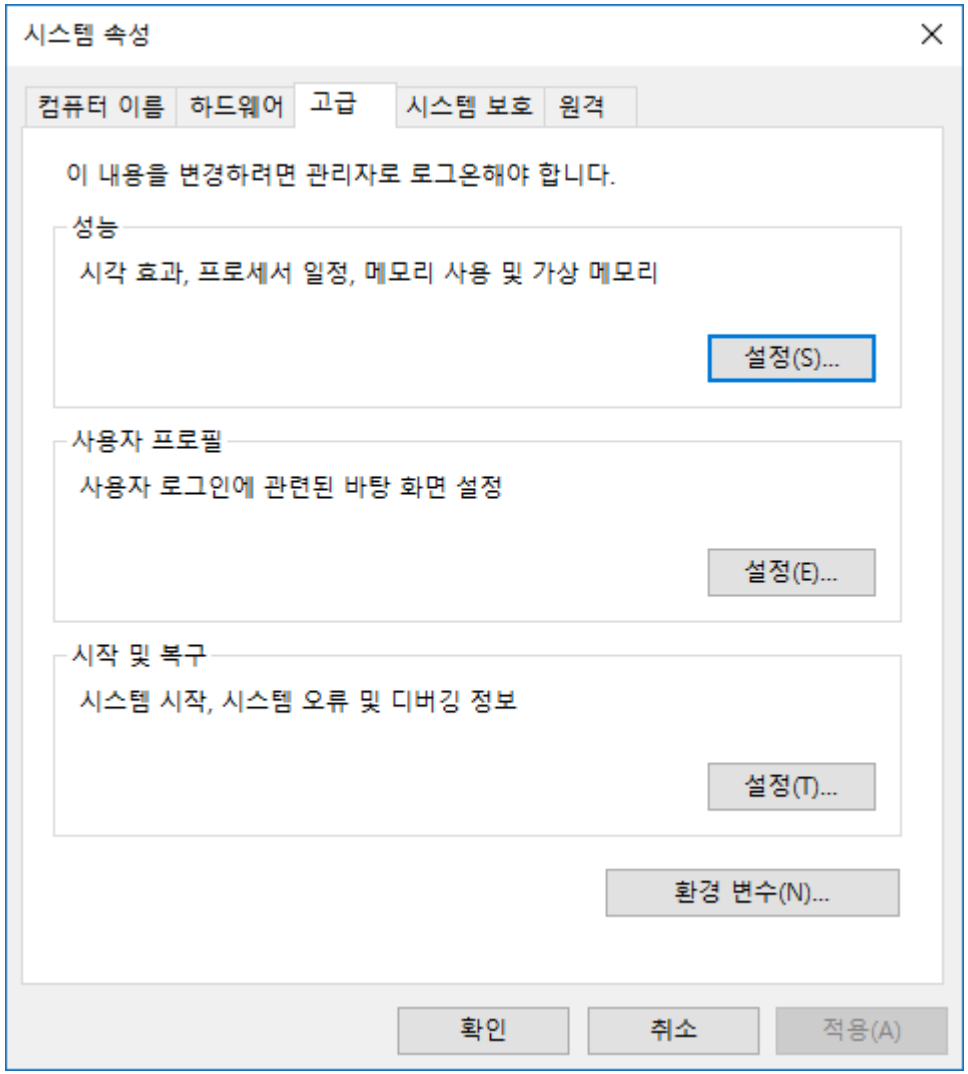


How to Install Python - Windows

Installation Completed

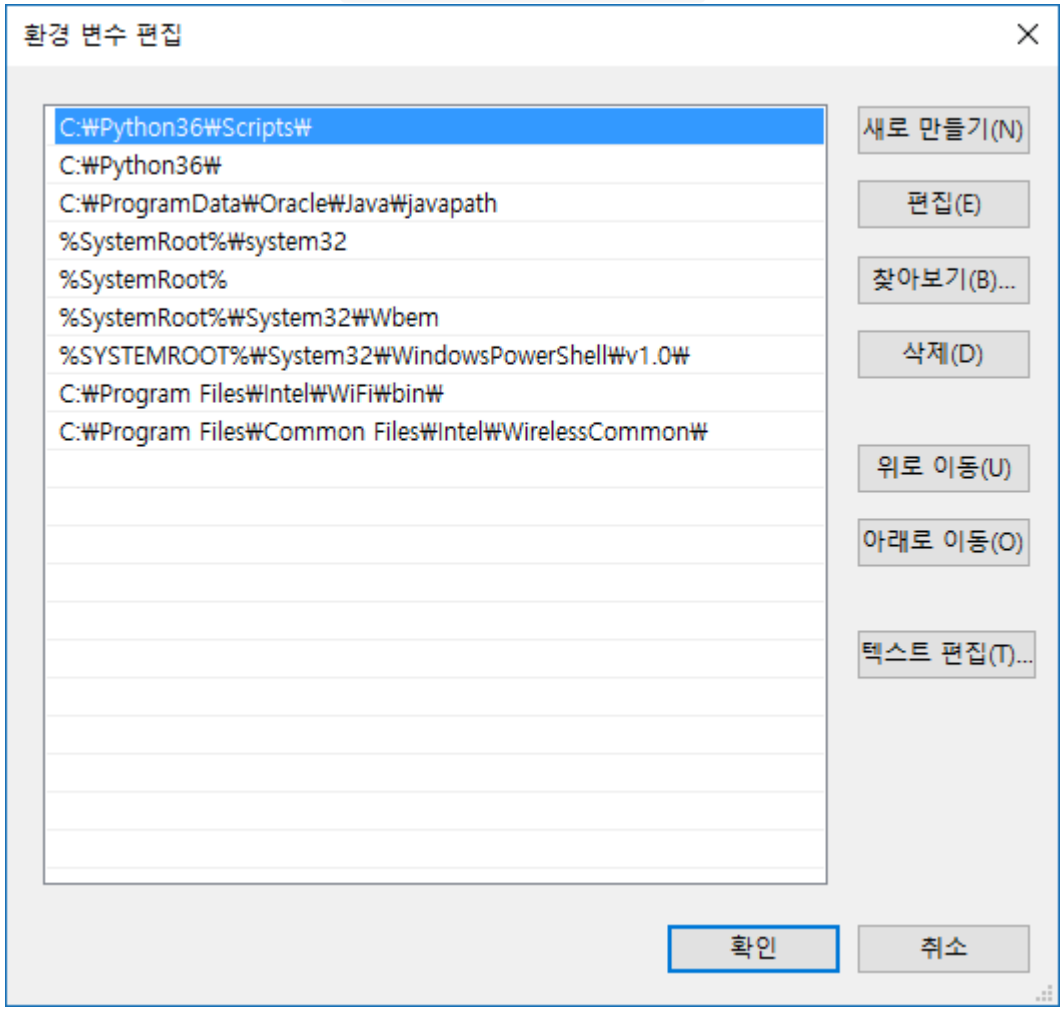


How to Install Python - Windows



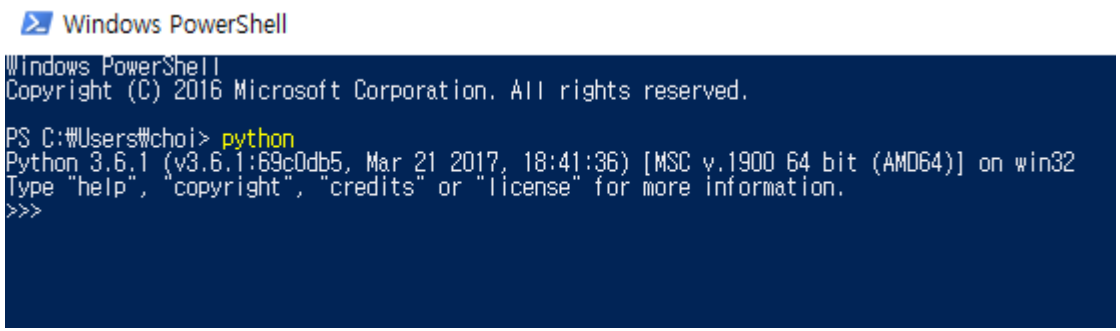
How to Install Python - Windows

Check PATH `C:\Python36`



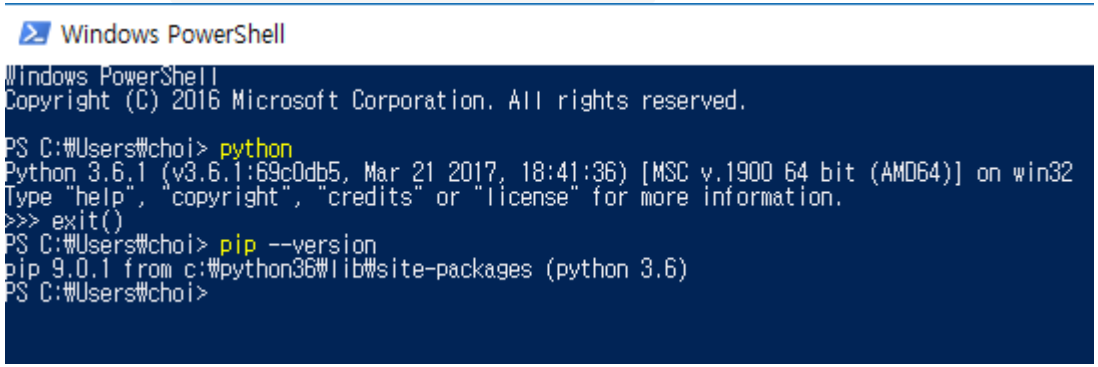
How to Install Python - Windows

It worked!



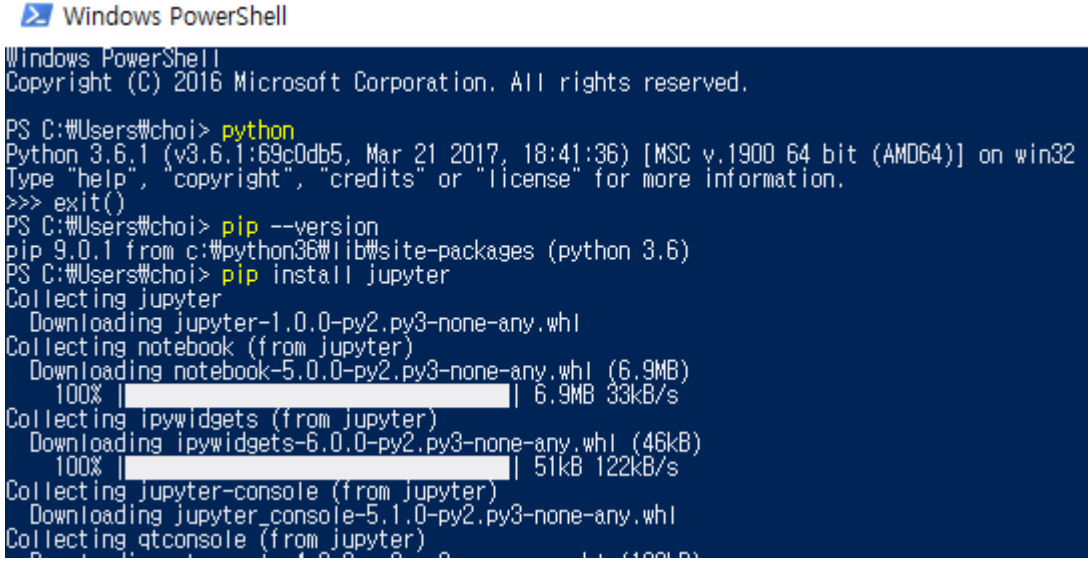
How to Install Python - Windows

check `$ pip --version`



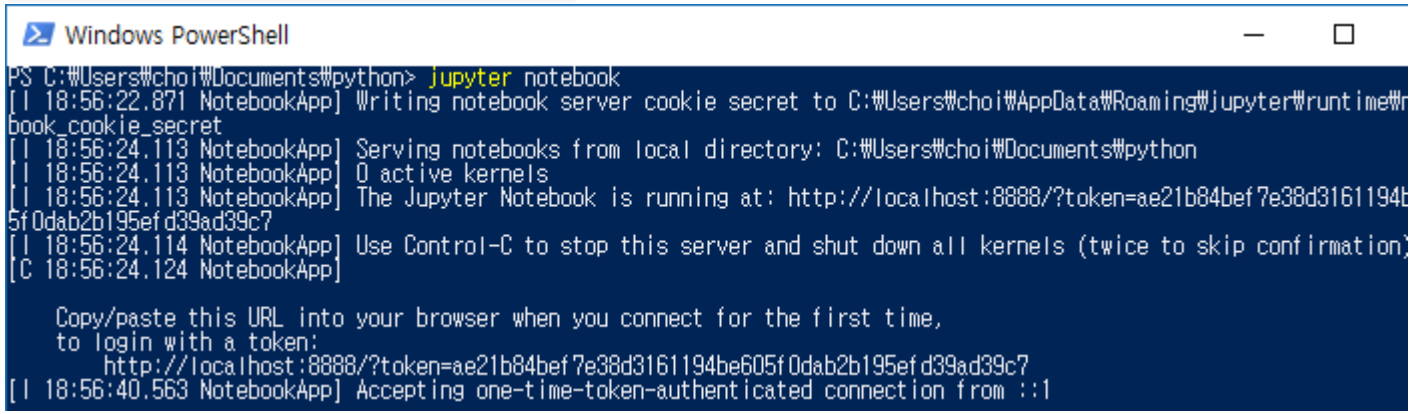
How to Install Python - Windows

`$ pip install jupyter` to install jupyter notebook



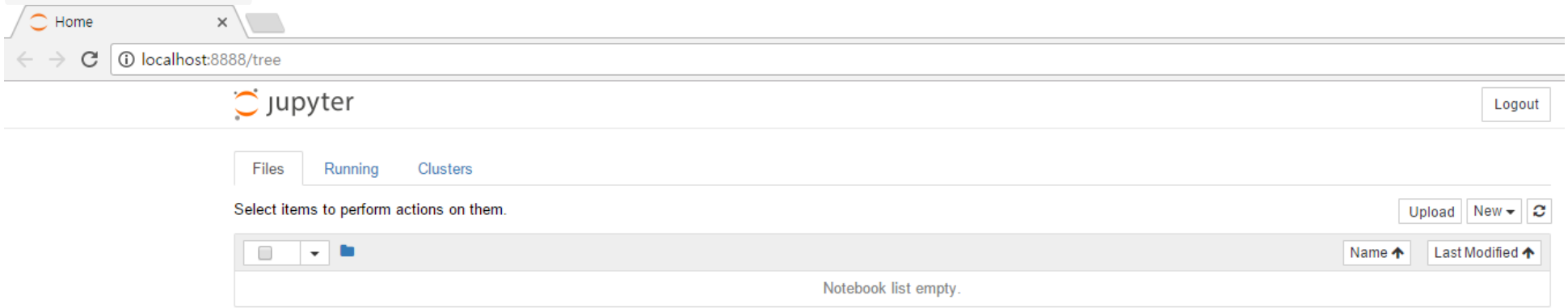
How to Install Python - Windows

`$ jupyter notebook` to run jupyter notebook



How to Install Python - Windows

```
Complete!
```



How to Install Python - MacOS, Linux(Ubuntu)

How to Install Python - Linux(Ubuntu)

```
$ sudo apt-get install python3.6.1
```

OR

Python Version Management - MacOS, Linux(Ubuntu)

Pyenv, Virtualenv

Pyenv - MacOS

Homebrew for MacOS

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

- Linux의 apt-get 처럼 패키지 관리를 할 수 있도록 도와주는 매니저
- apt-get 대신 brew를 입력하면 동일하게 사용가능

Pyenv - MacOS

```
$ xcode-select --install
```

```
$ brew update
```

```
$ brew install openssl readline xz
```

```
$ brew install pyenv
```

```
$ echo 'eval "$(pyenv init -)"' >> ~/.bashrc
```

```
$ exec $SHELL
```

Pyenv - Linux(Ubuntu)

Ubuntu

```
$ git clone https://github.com/pyenv/pyenv.git ~/.pyenv
```

```
$ apt-get install -y make build-essential libssl-dev zlib1g-dev
libbz2-dev libreadline-dev libsqlite3-dev wget curl llvm
libncurses5-dev xz-utils tk-dev
```

```
$ echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.bashrc
$ echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.bashrc
$ echo 'eval "$(pyenv init -)"' >> ~/.bashrc
```

```
$ exec $SHELL
```

Pyenv Installation check

`$ pyenv`

```
$ pyenv
pyenv 1.0.10
Usage: pyenv <command> [<args>]

Some useful pyenv commands are:
   commands    List all available pyenv commands
   local       Set or show the local application-specific Python version
   global      Set or show the global Python version
   shell       Set or show the shell-specific Python version
   install     Install a Python version using python-build
   uninstall   Uninstall a specific Python version
   rehash      Rehash pyenv shims (run this after installing executables)
   version     Show the current Python version and its origin
   versions    List all Python versions available to pyenv
   which       Display the full path to an executable
   whence      List all Python versions that contain the given executable

See `pyenv help <command>' for information on a specific command.
For full documentation, see: https://github.com/pyenv/pyenv#readme
```

Virtualenv - MacOS

```
$ brew install pyenv-virtualenv
```

```
$ echo 'eval "$(pyenv virtualenv-init -)"' >> ~/.bashrc
```

```
$ exec $SHELL
```

Virtualenv - Linux

```
$ git clone https://github.com/pyenv/pyenv-virtualenv.git $(pyen
```

```
$ echo 'eval "$(pyenv virtualenv-init -)"' >> ~/.bashrc
```

```
$ exec $SHELL
```

Create New version for specific project

install python with pyenv

```
$ pyenv install --list
```

```
$ pyenv install 3.6.1
```

```
$ pyenv versions
```

```
$ pyenv virtualenv 3.6.1 firstvenv
```

```
$ pyenv versions
```

```
$ pyenv activate firstenv
```

Install jupyter - MacOS, Linux

```
$ pip install jupyter
```

```
$ pip list
```

```
$ jupyter notebook
```

If these process doesn't work

- Try https://c9.io/
- Try https://repl.it/

Enjoy Python!