

PYTHON 프로그래밍 실습

Anaconda and Jupyter notebook

2019-01-03



ANACONDA



아나콘다 설치

- 파이썬 패키지 관리를 쉽게 해 주는 도구
하나의 PC에 독립된 개발환경 구성 가능

- 설치

- 다운로드 url: <https://www.anaconda.com/download/>

\$ bash Anaconda3-2019.10-Linux-x86_64.sh

(1) 설치 시, 아래 화면 나오면 yes 입력

```
Do you accept the license terms? [yes|no]  
[no] >>> 
```

아나콘다 설치

(2) 아래의 화면에서는 엔터 입력 후, 진행

```
Anaconda3 will now be installed into this location:
/home/pirl/anaconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/home/pirl/anaconda3] >>> █
```

(3) 아래의 화면에서는 yes 입력

```
installation finished.
Do you wish the installer to initialize Anaconda3
in your /home/pirl/.bashrc ? [yes|no]
[no] >>> █
```

\$ source .bashrc (← PATH환경변수 로딩)

\$ conda update conda (← 최신 버전의 conda로 업데이트)

가상환경

■ 가상환경 만들기

\$ **conda create -n py27** 또는 **conda create --name py27**

가상환경 이름

(예) 파이썬 3.6을 사용하는 py36라는 이름의 가상환경 만들기

\$ **conda create -n py36 python=3.6**

- 가상환경 생성 위치: ~/anaconda3/envs/py36
- 가상환경 목록: \$ conda info --envs

(참고)

\$ conda create -n py36 python=3.6 **ananconda**

(ananconda로 할 경우, 아나콘다 기본패키지가 모두 설치 됨)

가상환경

- 가상환경 활성화

\$ conda activate py36

- 가상환경 해제

\$ conda deactivate

- 콘다 내부의 패키지 리스트

\$ conda list 또는 conda list -n py36

패키지 설치

- 패키지 설치 (conda install package_name)

\$ conda install python=2.7

\$ conda install scipy=0.15.0 curl -n py27

\$ **conda install pandas**

- 패키지 업데이트

- conda update package_name

- 패키지 삭제

\$ conda remove -n py27 package_name

- 가상환경 삭제

\$ conda remove -n py27 --all

커널 등록

- Py36 을 추가

```
$ conda create -n py36 python=3.6
```

```
$ source activate py36
```

```
$ conda install notebook ipykernel
```

```
$ ipython kernel install --user --name=py36
```

--name = [virtualenv](#)

※ 커널 제거

```
$ jupyter kernelspec uninstall virtualenv
```

※ 커널 확인

```
$ jupyter kernelspec list
```


가상환경 Export and Import

- Export

```
$ conda env export -n env_name > environment.yml
```

- Import

```
$ conda env create -f environment.yml
```



JUPYTER NOTEBOOK



Jupyter notebook 서버 설치 및 실행

- 설치 (아나콘다를 설치하지 않고 주피터 노트북 설치)

```
$ sudo pip install jupyter
```

- 실행

```
$ jupyter notebook
```

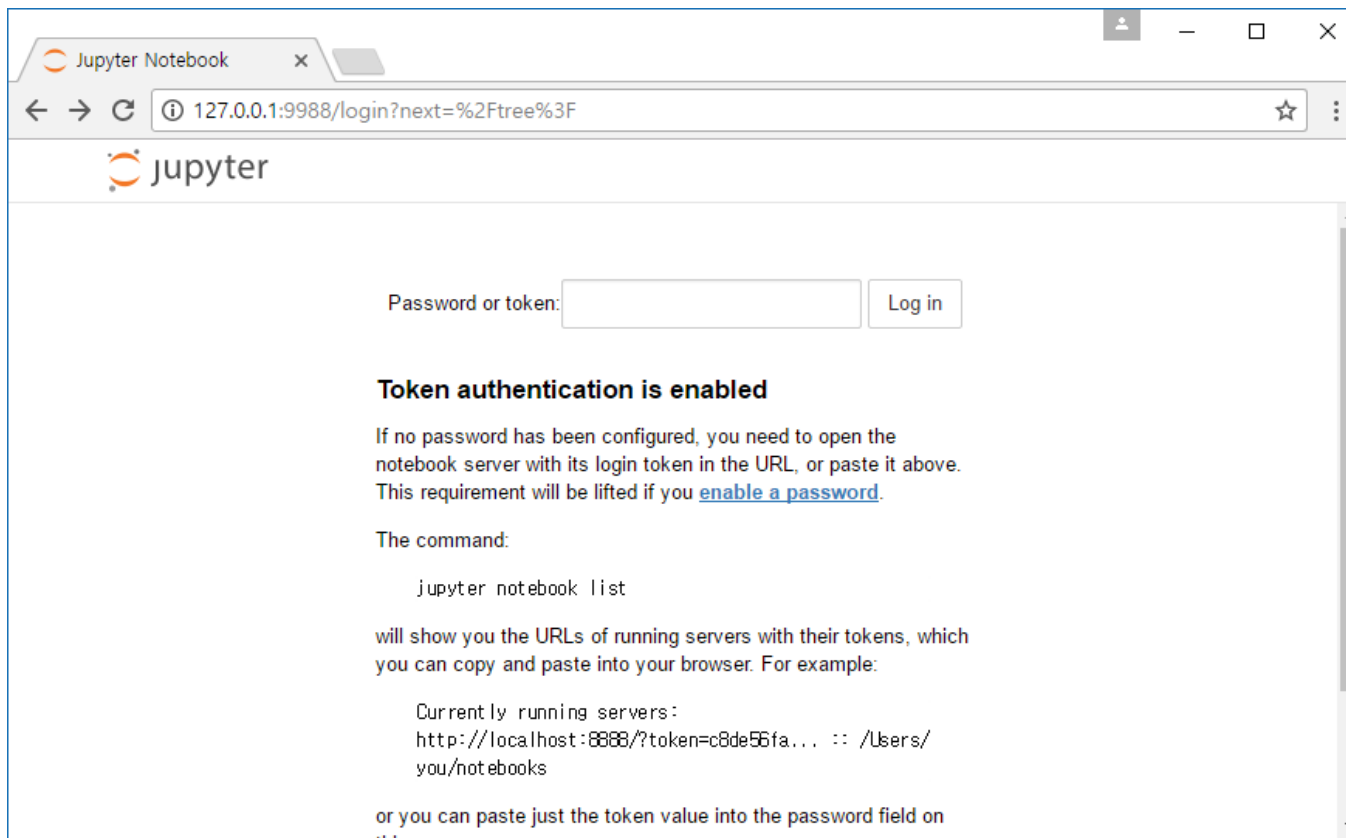
※ 기본적으로 실행된 위치를 기준으로 디렉터리 구조를 보여줌

- 웹 브라우저를 켜고, 아래 주소 입력

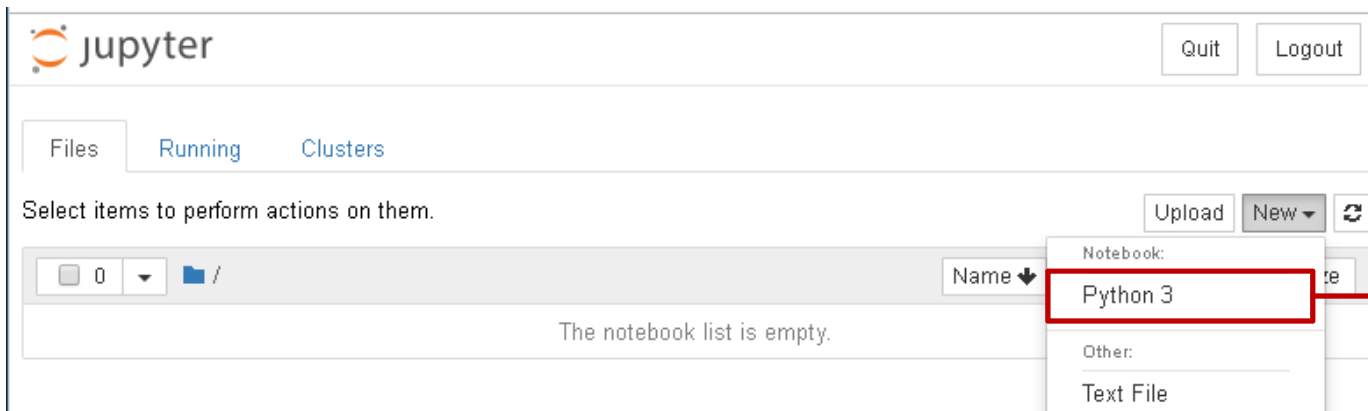
<http://127.0.0.1:8888>

Jupyter notebook 실행

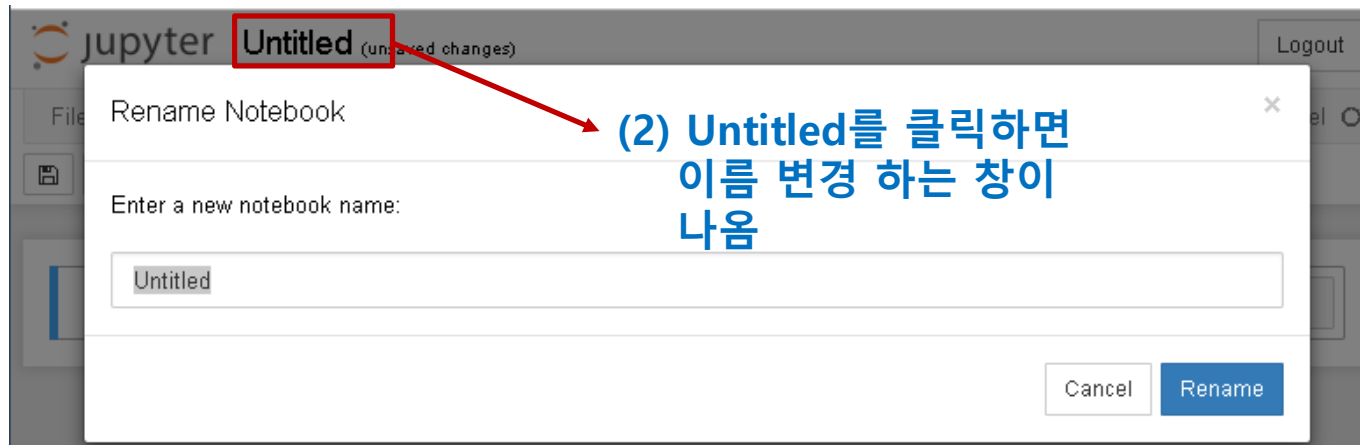
- 아래의 패스워드 입력 창에 token을 복사하여 입력하자.



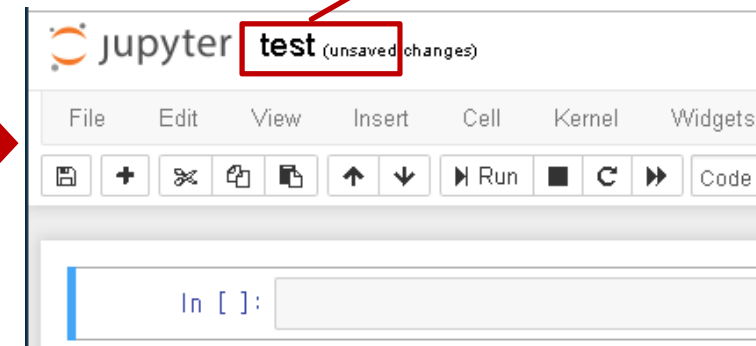
Jupyter notebook 실행



(1) New → Python 3 선택



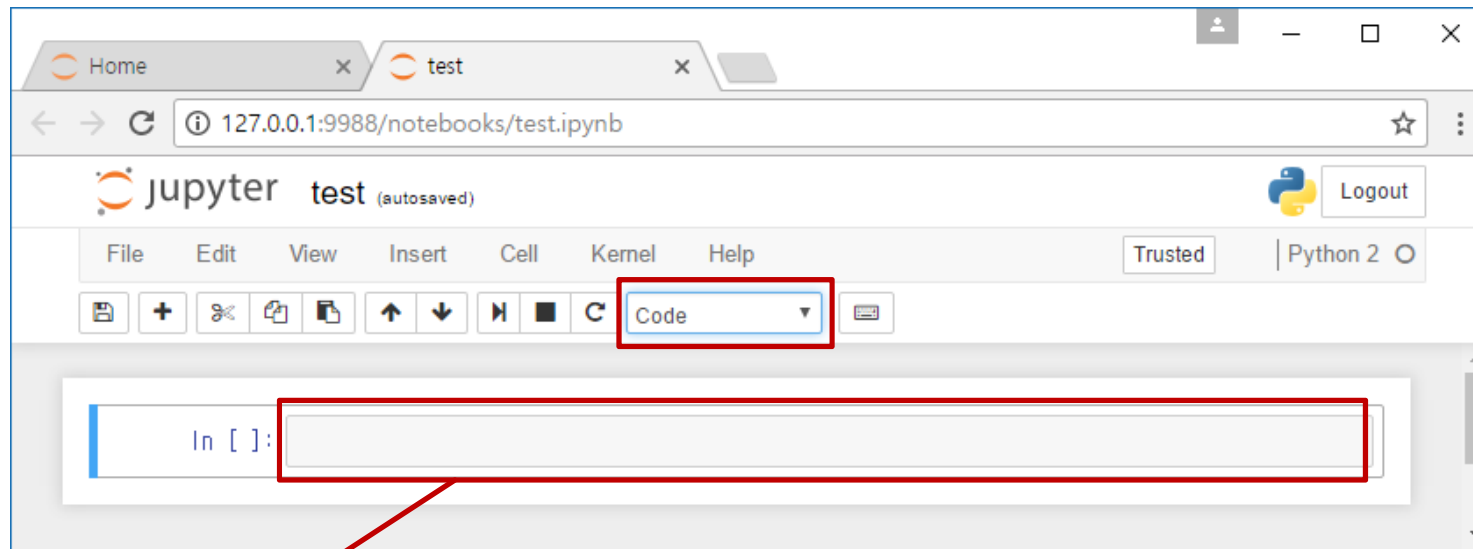
(2) Untitled를 클릭하면
이름 변경 하는 창이
나옴



(3) 이름 변경됨

Cell type: Code

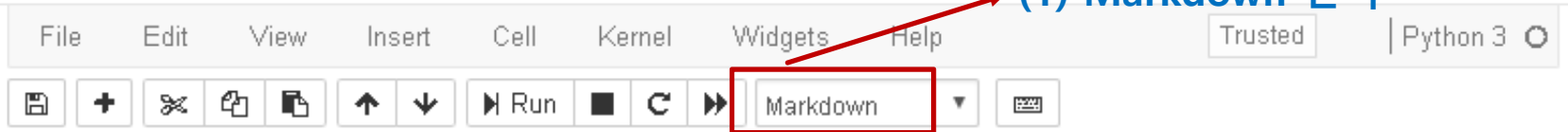
- Code는 Cell에 Python 코드를 입력하는 모드



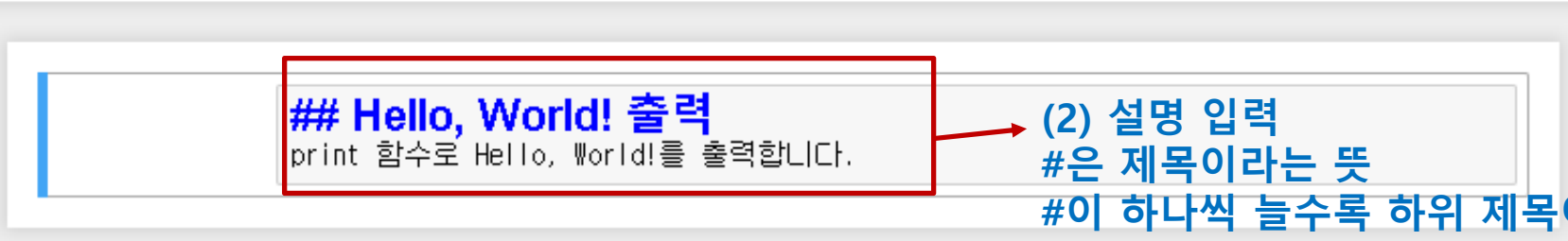
셀

Cell type: Markdown


(1) Markdown 선택



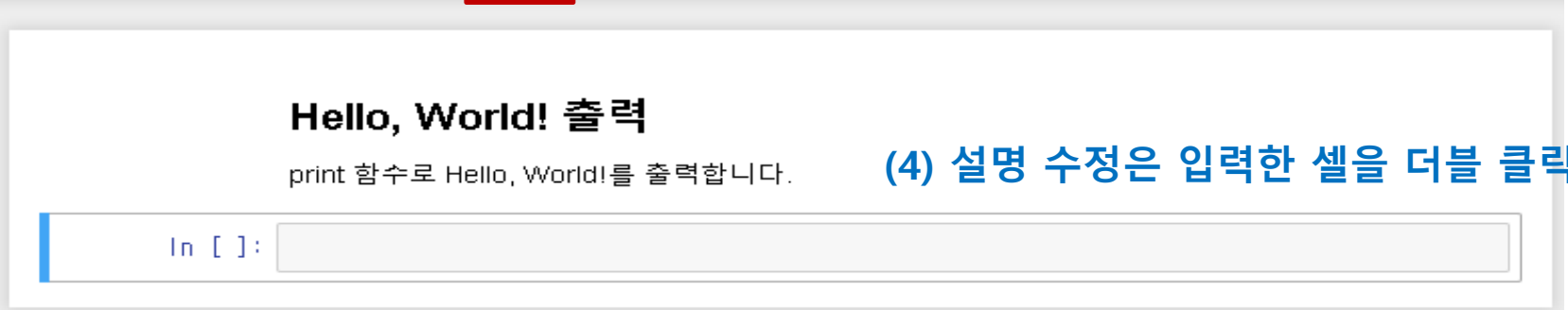
(2) 설명 입력
#은 제목이라는 뜻
#이 하나씩 늘수록 하위 제목이 됨



(3) 설명 적용을 위해 재생 버튼을 누름

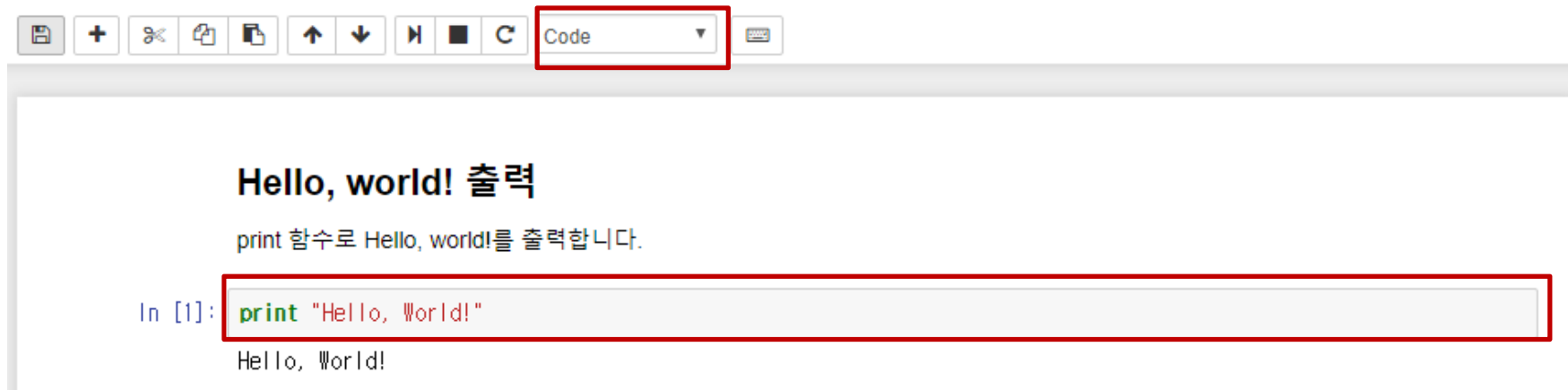


(4) 설명 수정은 입력한 셀을 더블 클릭!!



파이썬 코드 입력 및 실행

- Cell에 아래와 같이 한 줄 코드를 입력한다.



■ 실행

(1) 툴바 아이콘에서 재생버튼을 누름

(2) 또는, **Ctrl + enter**를 입력

코드가 실행되고 그 아래에 결과가 표시된다.

코드, 코드의 실행 결과, 코드에 대한 설명을 한 번에 저장할 수 있어 체계적이 기록이 가능

단축키

- **Ctrl + enter**

- Cell에 표현식을 입력하고 ctrl + enter 입력 시, 현재 셀이 실행됨

- **Shift + enter**

- 현재 셀 실행되고, 다음 셀이 활성화

- **Alt + enter**

- 현재 셀 실행되고, 아래 셀 삽입

- **Ctrl + s**

- 현재 jupyter notebook에 저장되었다는 표시가 남음

단축키

- 들어쓰기: Tab
- 내어쓰기: Shift + Tab
- 주석처리, 주석풀기: Ctrl + /
- 줄번호: Cell 에서 Ctrl-m 으로 명령(Command) 모드로 들어간 후
에 L 을 치면 줄 번호가 toggle되어 보인다.



OS COMMAND



디렉터리 만들고 이동

- magic command에는 line(%)과 cell(%%)로 지정해서 처리할 수 있음
- **%**(magic command)를 이용해서 현재 위치 및 디렉터리 생성 및 이동

```
In [13]: %pwd
```

```
Out[13]: u'/home/pirl'
```

```
In [14]: %mkdir TEST
```

```
In [15]: %cd TEST
```

```
/home/pirl/TEST
```

파이썬 파일 생성 및 실행

- **%%writefile** (magic command)를 이용해서 현재 위치에 add.py 생성하고 조회
- **%run** (magic command)을 이용해서 파이썬 모듈 실행

```
In [16]: %%writefile add.py  
  
def add(x,y):  
    return x + y  
  
print add(4,4)  
  
Writing add.py
```

```
In [17]: %ls  
  
add.py
```

```
In [19]: %run add.py  
  
8
```

파이썬 파일 실행하기

%loadpy (magic commad)를 이용해서 python 파일을 로드하고 실행하면 결과가 나옴

```
In [ ]: %loadpy add.py
```



```
In [ ]: # %load add.py  
  
def add(x,y):  
    return x + y  
  
print add(3,4)
```



```
In [26]: # %load add.py  
  
def add(x,y):  
    return x + y  
  
print add(3,4)
```

7