

## 파이썬 스크립트 생성

- 

Files

## Clusters

New ▼



Folder

## Notebooks

## Python 2



log

pid

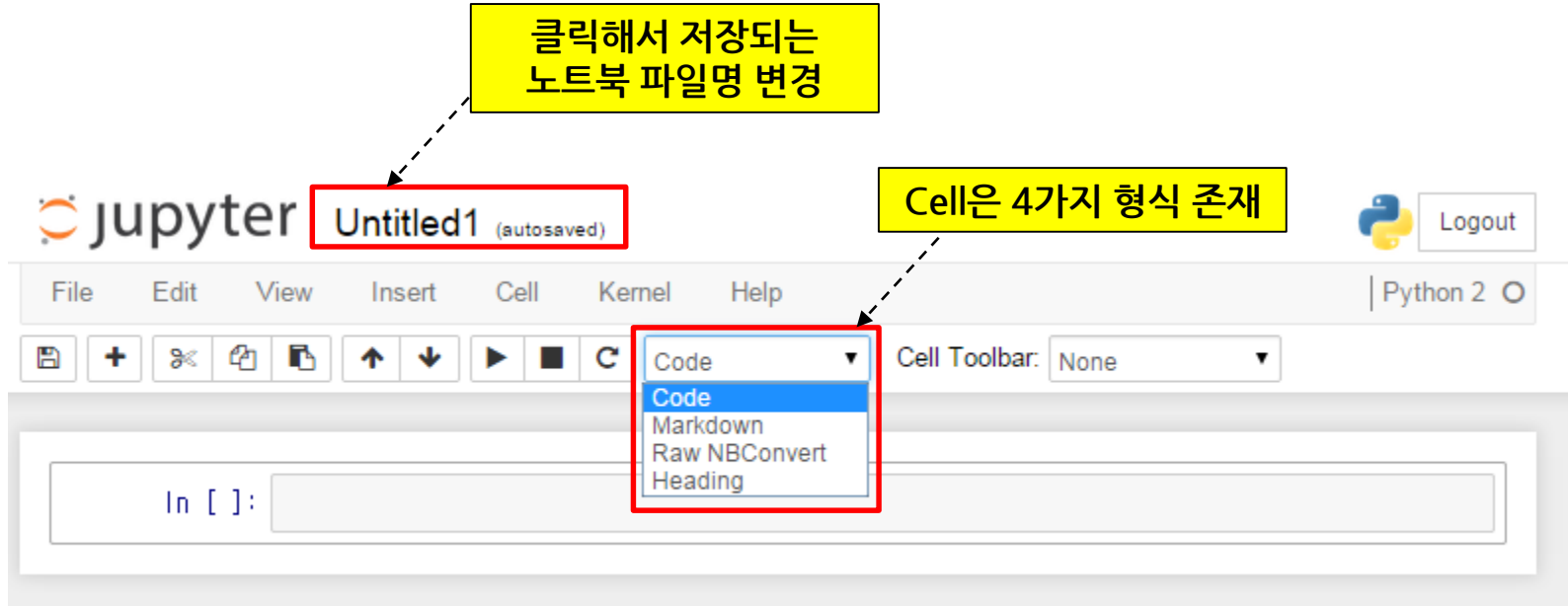
security



# IPython Notebook

## 파이썬 스크립트 생성

- Notebooks - Python 선택



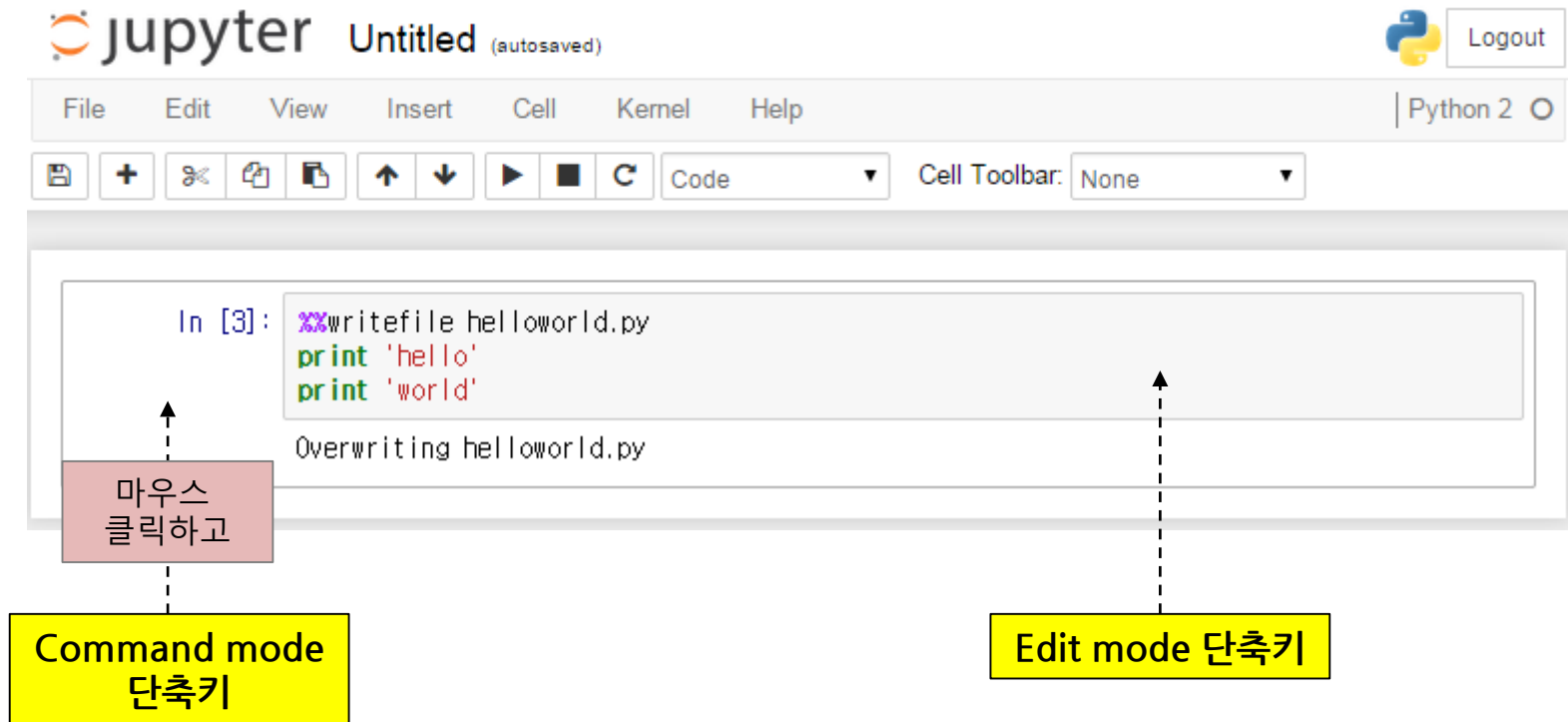
### Cell 종류 설명

- Code : 파이썬 코드
- Markdown : 일종의 마크업 언어. 엔하위키나 텀블러에서 사용 중
- Raw NBConvert : IPython 노트북 날 코딩(?)
- Heading : <html>의 <head> 태그와 유사한 기능

# IPython Notebook

## 단축키 (Shortcut)

- Edit Mode : 셀 안에서 사용하는 단축키
- Command Mode : 셀 경계에서 사용하는 단축키
- h 를 누르면 기본 단축키에 대해 어느 정도 나온다.



# IPython Notebook

## 단축키 (Shortcut) : Edit Mode 단축키

키보드	기능	키보드	기능
Tab	코드 자동완성, 들여쓰기(intent)	Ctrl-Left	한 단어 좌측 이동
Shift-Tab	툴팁	Ctrl-Right	한 단어 우측 이동
Ctrl-]	들여쓰기(intent)	Ctrl-Backspace	이전 단어 삭제
Ctrl-[	내어쓰기(detent)	Ctrl-Delete	이후 단어 삭제
Ctrl-a	전체 선택	Esc	command mode
Ctrl-z	undo	Ctrl-m	command mode
Ctrl-Shift-z	redo	Shift-Enter	run cell, 다음 셀 선택
Ctrl-y	redo	Ctrl-Enter	run cell
Ctrl-Home	셀 처음으로 이동	Alt-Enter	run cell, 다음 셀 삽입
Ctrl-End	셀 끝으로 이동	Ctrl-Shift--	셀 분할
		Ctrl-s	노트북 파일 저장

# IPython Notebook

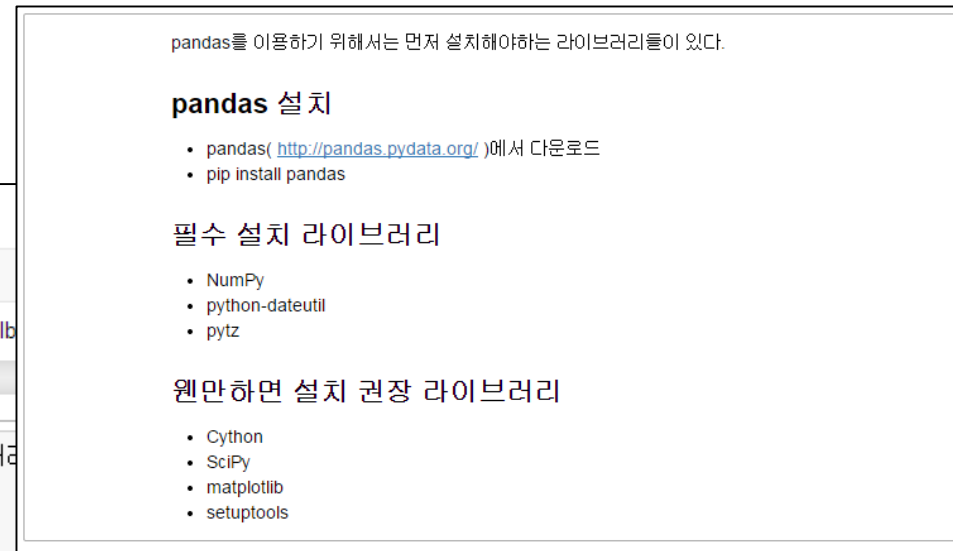
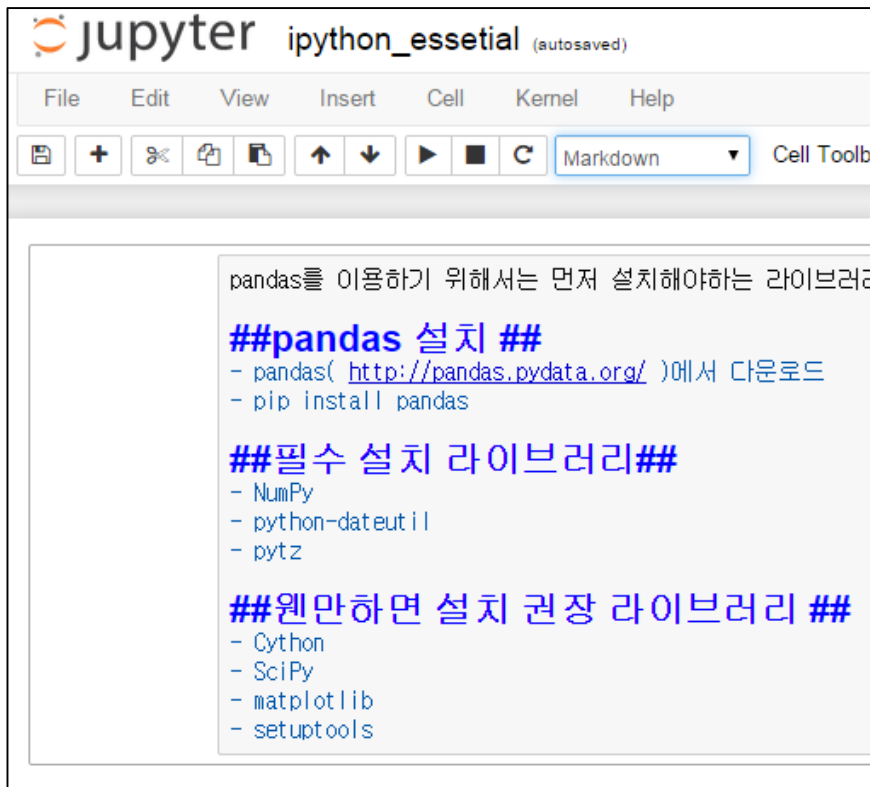
## 단축키 (Shortcut) : Command Mode 단축키

키보드	기능	키보드	기능
Enter	Edit mode 진입	c	셀 복사
Shift-Enter	run cell, 다음 셀 선택	Shift-v	위에 붙여넣기
Ctrl-Enter	run cell	v	아래 붙여넣기
Alt-Enter	run cell, 다음 셀 삽입	x	셀 잘라내기
y	Code 셀로 변환	dd	셀 삭제
m	Markdown 셀로 변환	z	undo 이전 삭제 복구
r	Raw NB 셀로 변환	Shift-m	아래 셀과 병합(merge)
1	Markdown : H1	s	노트북 저장
2	Markdown : H2	Ctrl-s	노트북 저장
3	Markdown : H3	l	해당 셀의 라인번호 토글
4	Markdown : H4	o	결과값 토글
5	Markdown : H5	Shift-o	결과값 토글(scrolling)
6	Markdown : H6	h	keyboard shortcuts
a	위에 셀 삽입	ii	interrupt kernel
b	아래 셀 삽입	00	restart kernel
k / 방향키 위	이전 셀로 이동	j / 방향키 아래	다음 셀로 이동

# IPython Notebook

## Cell : Markdown

- Markdown language 기반으로 Document 작성 (Wikipedia처럼 작성 방식)
- 코드에 대한 설명이 가능하다.
- 수학수식 표현도 가능 (Mathjax)



내용을 적고  
Ctrl+Enter

# IPython Notebook

## Cell : Markdown

- 제목(Head 태그)

# 텍스트

# 하나를 쓰면 HTML의 <h1> 태그

# 두개를 쓰면 HTML의 <h2> 태그. 최대 6개까지 쓸 수 있다.

- 리스트 (li 태그)

- 번호 없는 리스트 : -텍스트 , +텍스트, \*텍스트

- 번호 있는 리스트 : 숫자.텍스트

- 기울인 글씨(Italic 태그) : \_텍스트\_ , \*텍스트\*

- 굵은 글씨(Strong 태그) : \_\_텍스트\_\_ , \*\*텍스트\*\*

- 인용(cite) : >텍스트

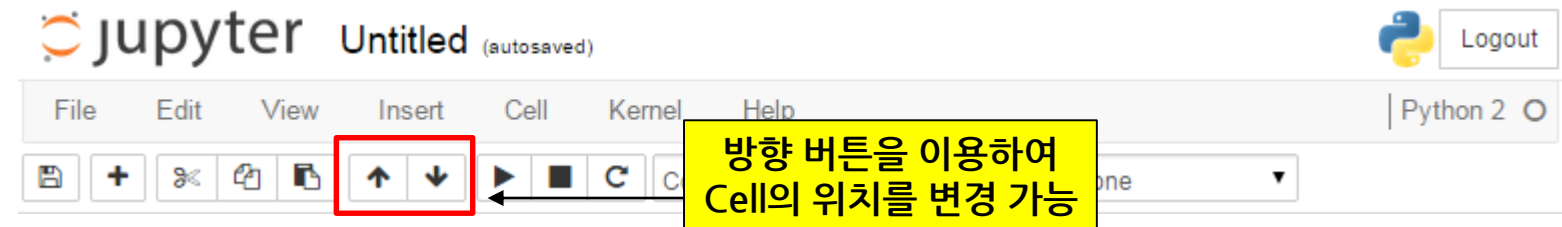
- 인라인 링크 : [텍스트](링크)

- 수평선(hr 태그) : --- , \*\*\* , \_\_\_\_

# IPython Notebook

## Cell : Code

- 파이썬의 코드를 각 셀에 원하는 만큼을 작성하여 실행하면 메모리 상에 반영된다.
- 위, 아래 위치가 달라도 실행한 스텝 번호(In [번호])가 높을 수록 최근에 수행된 영역이다.



The Fibonacci numbers are the sequence of numbers  $\{F_n\}_{n=1}^{\infty}$  defined by the linear recurrence equation

$$F_n = F_{n-1} + F_{n-2}$$

with  $F_1 = F_2 = 1$ , and conventionally defining  $F_0 = 0$ .

```
In [3]: def fibonacci(n):  
        if n < 1:  
            return 1  
        else:  
            return fibonacci(n-1)+fibonacci(n-2)
```

```
In [4]: print fibonacci(3)
```

5



# IPython Notebook

## Magic Keyword (매직 명령어)

- 자주 사용하는 IPython 매직 명령어
  - `%(명령어)` : 셀 내의 라인 독립 실행
  - `%%(명령어)` : 셀 내의 전체 내용 관련 실행

매직 명령어	기능
<code>%quickref</code>	IPython의 빠른 도움말 표시
<code>%magic</code>	모든 매직 함수에 대한 상세 도움말 출력
<code>%debug</code>	최근 예외 트레이스백의 하단에서 대화형 디버거로 진입
<code>%hist</code>	명령어 입력(그리고 선택적 출력) history 출력
<code>%pdb</code>	예외가 발생하면 자동으로 디버거로 진입
<code>%run script.py</code>	IPython 내에서 파이썬 스크립트 실행
<code>%timeit statement</code>	statement를 여러차례 실행한 후 평균 실행 시간을 출력. 매우 짧은 시간 안에 끝나는 코드의 시간을 측정할 때 유용
<code>%%writefile 파일명</code>	filename인 파일을 생성
<code>%save 파일명 from-to</code>	IPython의 이미 실행된 Step번호 구간대 명령어를 파일로 저장

# IPython Notebook

## Magic Keyword (매직 명령어)

- 자주 사용하는 IPython 매직 명령어
  - `%(명령어)` : 셀 내의 라인 독립 실행
  - `%%(명령어)` : 셀 내의 전체 내용 관련 실행

```
In [1]: %pdb
```

Automatic pdb calling has been turned ON

```
In [6]: %%writefile fibonacci.py
def fibonacci(n):
    if n < 1:
        return 1
    else:
        return fibonacci(n-1)+fibonacci(n-2)
```

Writing fibonacci.py

```
In [10]: from fibonacci import *
```

```
In [11]: fibonacci(10)
```

```
Out [11]: 144
```

```
In [4]: %timeit fibonacci(10)
```

10000 loops, best of 3: 123 µs per loop

# IPython Notebook

## Magic Keyword (매직 명령어) : TIP

- (리눅스/유닉스만) %%writefile으로 스크립트 저장 후 %run으로 실행하려면  
파이썬 스크립트에 한글이 포함될 때는 #!(샤방) 파이썬 패스, 언어 타입 정의 필요하다.

```
#!/usr/bin/python
```

```
#-*- coding: utf-8
```

```
In [13]: %%writefile fibonacci.py
#!/usr/bin/python
#-*-coding:utf-8
#피보나치 수열 함수
def fibonacci(n):
    if n < 1:
        return 1
    else:
        return fibonacci(n-1)+fibonacci(n-2)

#피보나치 수열 실행부 출력
print '피보나치 f(10)=', fibonacci(10)
```

Overwriting fibonacci.py

```
In [14]: %run fibonacci.py
```

피보나치 f(10)= 144

# IPython Notebook

## 운영체제와 함께 사용하기

- IPython에서 운영체제의 자체 명령어를 실행하기 위해서는 **!(명령어)**로 수행

```
In [1]: !pwd
```

```
/home/seen/.ipython/profile_2025
```

```
In [2]: !ls
```

```
2nd.ipynb      fibonacci.pyc  nohup.out
3rd.ipynb      helloworld.py  pandas.ipynb
Untitled.ipynb  ipython_config.py  pid
Untitled1.ipynb ipython_console_config.py  security
Untitled2.ipynb ipython_kernel_config.py  startup
cluster.ipynb   ipython_notebook_config.py  static
cluster_run.ipynb log                      untitled.txt
fibonacci.py    matplotlib.ipynb
```

```
In [3]: !tail nohup.out
```

```
        u'output_type': u'stream',
        u'stream': u'stdout',
        u'text': u'File `cluster.py` exists. Overwrite (y/[N])? y\n'}
[W 00:06:22.542 NotebookApp] Notebook cluster.ipynb is not trusted
[I 00:08:32.918 NotebookApp] Saving file at /Untitled.ipynb
[I 00:08:58.384 NotebookApp] Saving file at /Untitled.ipynb
[I 00:09:19.128 NotebookApp] Creating new notebook in
[I 00:09:19.657 NotebookApp] Kernel started: 7f0b4ee8-e376-44ff-a48a-448bd38dddf2
[I 00:09:39.630 NotebookApp] Saving file at /Untitled2.ipynb
[I 00:10:52.405 NotebookApp] Kernel restarted: 7f0b4ee8-e376-44ff-a48a-448bd38dddf2
```

# IPython Notebook

## 운영체제와 함께 사용하기

- **!(명령어)**로 수행된 출력결과를 IPython의 데이터로 사용 가능

In [2]:

```
!ls
```

```
2nd.ipynb      fibonacci.pyc      nohup.out
3rd.ipynb      helloworld.py      pandas.ipynb
Untitled.ipynb  ipython_config.py  pid
Untitled1.ipynb ipython_console_config.py security
Untitled2.ipynb ipython_kernel_config.py startup
cluster.ipynb   ipython_notebook_config.py static
cluster_run.ipynb log                untitled.txt
fibonacci.py    matplotlib.ipynb
```

In [4]:

```
list = !ls
```

In [9]:

```
for file in list:
    print file
```

```
2nd.ipynb
3rd.ipynb
Untitled.ipynb
Untitled1.ipynb
Untitled2.ipynb
cluster.ipynb
cluster_run.ipynb
fibonacci.py
fibonacci.pyc
helloworld.py
ipython_config.py
ipython_console_config.py
ipython_kernel_config.py
ipython_notebook_config.py
log
```