

단축키

명령모드(Command) 단축키

- a** : above (위에 셀 추가하기)
- b** : bottom (아래 셀 추가하기)
- d+d** : delete + delete (셀 삭제하기)
- shift + 'l'** : line (셀 줄번호 표시하기)
- shift + 'm'** : merger (셀 병합)

편집모드(Edit) 단축키

- control + Return** : 셀 실행
- shift + Return** : 셀 실행 후 아래 셀로 이동
- option + Return** : 셀 실행 후 아래 셀 삽입하며 이동
- shift + Tab** : 도움말
- Tab** : 자동 완성
- control + '/'** : 코드 주석 처리(해당 코드라인 가장 앞에 #을 넣는 기능)
- control + shift + '-'** : 셀 나누기 / shift + m 병합하기와 반대 기능

자료형 및 기본 문법

- ()** : 함수, 쿼리
- []** : 호출, 리스트
- { : }** : 딕셔너리

- ESC키**를 누르면 **명령(Command)모드**로 전환
- Return키**를 누르면 **편집(Edit)모드**로 전환

ESC키를 눌러 명령모드로 전환된 상태에서 'M'키를 누르면 Markdown 상태가 되어 제목이나 메모를 기록할 수 있다. (**M : Markdow**
n, Y : Code)

문자열, 리스트

sum(ABC) : ABC 리스트의 합을 반환.

min(ABC) : ABC 리스트에서 최소값 반환.

max(ABC) : ABC 리스트에서 최대값을 반환

int(ABC) : ABC의 자료형을 숫자(정수)로 형변환. 소숫점 이하는 내림.

float(ABC) : ABC의 자료형을 숫자(실수)로 형변환.

str(123) : 123의 자료형을 문자로 형변환.

type(ABC) : ABC의 자료형을 반환한다.

ABC = 123 : 123은 숫자(정수)

ABC = "123" : 123은 문자

print(secinfo[3][0]) : secinfo 리스트의 3번 인덱스 값의 0번 인덱스 값

print(secinfo[0][2]) : secinfo 리스트의 0번 인덱스 값의 2번 인덱스 값

print(secinfo[1][1]) : secinfo 리스트의 1번 인덱스 값의 1번 인덱스 값

ABC[2] = 120 : ABC 리스트의 2번 인덱스 값을 120으로 업데이트한다. **insert와 비교**

ABC[2:4] = [120,250] : ABC 리스트의 2번 인덱스 ~ 4번 인덱스 전까지 값을 120,250으로 업데이트한다.

ABC.remove(120) : ABC 리스트에서 가장 앞에서 발견된 120을 삭제한다.

del ABC[3] : ABC 리스트에서 3 인덱스 값을 삭제한다.

ABC.pop() : ABC 리스트에서 마지막 값을 반환 후 리스트에서 삭제한다.

ABC.pop(3) : ABC 리스트에서 3번 인덱스 값을 반환 후 리스트에서 삭제한다.

ABC[2:5] = [] : ABC 리스트의 2번 인덱스 ~ 5번 인덱스 전까지 값을 삭제한다.

ABC.append(300) : ABC 리스트의 마지막에 값 300을 덧붙인다.

ABC.insert(3,200) : ABC 리스트의 3번 인덱스에 값 200을 삽입한다. **update와 비교**

ABC.extend([135,270]) : ABC 리스트의 마지막에 [135,270] 리스트를 연장한다.

(리스트 + 리스트 포맷은 리스트 연산으로도 가능하다. **ABC = ABC + [135,270]**)

ABC.sort() : ABC 리스트의 값을 오름차순 오름차순 정렬한다. 1,2,3,4. (저장 X / 저장하려면 **ABC = ABC.sort()**)

ABC.sort(reverse = False) : ABC 리스트의 값을 오름차순 정렬한다. 1,2,3,4. 반드시 대문자로 쓴다. false로 입력하면 에러!!

ABC.sort(reverse = True) : ABC 리스트의 값을 내림차순 정렬한다. 4,3,2,1. 반드시 대문자로 쓴다. true로 입력하면 에러!!

len(ABC) : ABC 리스트의 길이를 반환.

`ABC.index(120)` : ABC 리스트에서 값 120의 인덱스를 반환.

`ABC.count(120)` : ABC 리스트에서 값 120의 개수 세기.

`ABC.index("G")` : ABC에서 대문자 "G"의 인덱스를 반환.

`ABC.count("G")` : ABC에서 대문자 "G"의 개수 세기.

`ABC.upper()` : ABC의 모든 문자를 대문자로 바꾼다.

`ABC.lower()` : ABC의 모든 문자를 소문자로 바꾼다.

`ABC.replace("G","K")` : ABC에서 대문자 "G"를 대문자 'K'로 바꾼다.

`DEF = ABC.split("G")` : ABC를 대문자 "G"를 기준으로 n개로 분리한 값들을 DEF 리스트로 만든다.

`print("원숭이 엉덩이는 {} 빨가면 {} 사과는 {}".format(변수1, 변수2, 변수3))` : 변수와 문자열 여러개가 섞인 경우 출력 방법.

`ABC.strip()` : ABC 문자열 앞뒤의 공백(/n, /t 포함)을 제거한다. (문자열中间的 공백은 유지)

딕셔너리

`ABC["Key1"]` : ABC 딕셔너리의 "Key1"에 대응하는 "Value1"을 반환한다.

`ABC["Key1"] = "Value1_1"` : ABC 딕셔너리에 "Key1"이 있으면 Value값을 "Value1_1"로 업데이트한다.

`ABC["KeyA"] = "ValueA"` : ABC 딕셔너리에 "KeyA"가 없으면 {"KeyA ": "ValueA"}를 딕셔너리에 추가한다.

`del ABC["Key2"]` : ABC 딕셔너리의 "Key2"와 그에 대응하는 "Value2"를 삭제한다.

`dic.pop("Key2")` : ABC 딕셔너리에의 "Key2"에 대응하는 "Value2"를 반환 후 딕셔너리에서 삭제한다.

홀수값 뽑아내기

`ABC[0::2]` : 리스트 0번에서 시작. 등차는 2.

짝수값 뽑아내기

`ABC[1::2]` : 리스트 1번에서 시작. 등차는 2.

반복문으로 표현하는 방법은 다음 글 하단을 참조한다.

[2020/03/25 - \[개발자/Python\] - 홀수값, 짝수값](#)

함수

`import datetime as dt` (as dt는 빼도 되지만 이름을 선언하는게 코딩 하기 더 좋다.)

`datetime.datetime.now()`는 **통째로 외우자!**

`datetime` 모듈 이름을 `dt`라 정했다면, `dt.datetime.now()`로 쓸 수 있다.

`ABC.hour` : (ABC가 시간값일 때) 시(hour)를 정수로 출력.

`ABC.minute` : (ABC가 시간값일 때) 분(minute)을 정수로 출력.

`import pandas as pd`
`import d6stack` : 판다스 성능 향상.

`pd.DataFrame(ABC, columns=[' '])` : ABC 리스트를 데이터프레임으로 만들기.

`pd.DataFrame(zip(ABC,DEF), columns=[' ',' '])` : ABC, DEF 리스트를 데이터프레임으로 만들기.

(2개 이상의 리스트를 데이터프레임으로 만들려면 한 번에 `zip()`으로 묶어서 만들거나
각각 데이터프레임으로 만든 다음 `pd.concat`으로 합친다.)

`ABC.columns = ['aaa','bbb','ccc']` : ABC 데이터프레임에 `aaa`, `bbb`, `ccc`를 각 컬럼헤더로 넣기. (컬럼헤더를 못 넣고 데이터프레임을 만들었을 경우.)

`ABC.rename(columns = {'aa':'A','bb':'B'})`:ABC 데이터프레임의 컬럼헤더를 `aa`, `bb`에서 `A`, `B`로 변경하기.
파이썬에는 `inplace`라는 옵션을 가지는 함수가 있다. `inplace=False` : 저장 안 함. `inplace=True` 저장함.
`rename` 함수의 `inplace` 기본값은 `False`로 저장하려면 다시 변수에 넣어주거나 `.rename({ },inplace=True)` 옵션을 준다.

`ABC.index = ['aaa','bbb','ccc']` : ABC 데이터프레임에 `aaa`, `bbb`, `ccc`를 각 인덱스로 넣기. (로우헤더.)

`pd.concat([ABC,DEF,GHI], axis = 1)` : ABC,DEF,GHI 데이터프레임을 하나로 합치기. (axis = 0 : 위아래로, axis = 1 : 좌우로)

`ABC.head(2)` : ABC 데이터프레임에서 첫 2줄만 보기.

`ABC.tail(2)` : ABC 데이터프레임에서 끝 2줄만 보기.

`ABC.columns` : ABC 데이터프레임의 컬럼헤더 보기.

`ABC['컬럼헤더']` : ABC 데이터프레임에서 컬럼헤더가 '컬럼헤더'인 것만 보기.

`ABC.loc[ABC.컬럼명 == 값]` : ABC 데이터프레임을 제한한다. ABC 데이터프레임의 컬럼명의 값만 뽑는다.

`pd.read_csv("../../../ABC.csv")` : ABC.csv를 판다스로 불러오기. (한글 포함 파일 에러 발생시, `encoding='ms949'` 또는 `'cp949'` 옵션을 준다.)

`pd.read_csv("../../../ABC{}.csv".format(i))` : ABCi.csv를 판다스로 불러오기. (ABC1, ABC2, ... 이렇게 파일이 있을 경우 반복문을 통해 불러올 수 있다. 단, 중간에 비는 번호가 있을 경우를 대비해 `try except`로 예외처리를 해줘야한다.)

`ABC.to_csv("./filename.csv", index=False, encoding="UTF-8")` : AB데이터프레임을 `filename.csv` 로 저장한다. (판다스가 자동으로 인덱스를 1열에 삽입하여 저장하기 때문에 일반적으로 , `index+False` 옵션은 항상 주는 것이 좋다.)
(내보낼 수 있는 옵션은 csv 외에도 clipboard, dict, excel, html, json, latex, markdown, numpy, pickle, records, sql,string 등 다양하다.) (파일 저장시 누적해서 저장되지 않는다. 파일을 덮어쓰기 한다.)

`pd.read_sql_query("Query", engine)` : "Query"를 이용해 판다스로 불러올 수 있다. 접속은 `engine`을 이용한다. (테이블에서 원하는 것만 불러올 수 있다.)

`pd.read_sql_table(table_name = ABC, con = engine)` : 테이블 이름이 ABC인 것을 `engine`을 이용해 판다스로 불러오기. (테이블 전체를 불러온다.)

`ABC.to_sql(name = "Hello", con = engine, index = False)` : ABC 데이터프레임을 `engine`을 이용해 DB에 'Hello'라는 테이블 이름으로 저장한다.

`d6tstack.utils.pd_topsql(df = ABC, uri = "engine 만들 때 사용하는 URL", table_name = "Hello")` : ABC 데이터프레임을 "engine 만들 때 사용하는 URL"을 이용해 "Hello"라는 테이블 이름으로 저장한다.

`pd.json_normalize()` : JSON 데이터를 데이터프레임으로 만든다.

`pd.read_json()` : JSON 데이터를 데이터프레임으로 만든다.

`import psycopg2` : PostgreSQL 클라이언트 모듈.

`import pymysql` : MySQL, MariaDB 클라이언트 모듈.

`from sqlalchemy import create_engine` : 데이터베이스 종류, 주소, 이름 등을 이용해 접속하는 엔진을 만들어준다.

`engine = create_engine`

`import request, bs4` : 웹 크롤링을 위해 HTML을 불러와 예쁘게 담아낸다.

`from selenium import webdriver`

`driver.get(url)` : url로 이동

`driver.page_source` : 현재 페이지의 HTML 태그를 저장한다. `requests.get(url).text`와 같다.

`driver.find_element_by_xpath(xpath).click()` : 클릭.

`driver.find_element_by_xpath(xpath).send_keys('ABC')` : 'ABC'를 입력.

`driver.find_element_by_xpath(xpath).send_keys(Keys.ENTER)` : return 입력. (반드시 ENTER를 대문자로 입력해야한다.)

Selenium Keys

더보기

`action1 = driver.find_element_by_xpath(xpath1)`

`action2 = driver.find_element_by_xpath(xpath2)`

`ActionCahins.move_to_element(action1).click(action2).perform()` : 마지막 `perform()`에 의해 순차적으로 실행.

```
# example

from selenium.webdriver.common.action_chains import ActionChains

coupangUrl = 'http://www.coupang.com'
driver.get(coupangUrl)

mainMenu = '//*[@id="header"]/div/a'
subMenu = '//*[@id="gnbAnalytics"]/ul[1]/li[4]/a'
subMenu2 = '//*[@id="gnbAnalytics"]/ul[1]/li[4]/div/div/ul/li[6]/a'
targetMenu = '//*[@id="gnbAnalytics"]/ul[1]/li[4]/div/div/ul/li[6]/div/ul/li[6]/a'
main = driver.find_element_by_xpath(mainMenu)
sub = driver.find_element_by_xpath(subMenu)
sub2 = driver.find_element_by_xpath(subMenu2)
target = driver.find_element_by_xpath(targetMenu)
ActionChains(driver).move_to_element(main).move_to_element(sub).move_to_element(sub2).click(target).perform()
```

`time.sleep(3)` : `import time` 필요.

`driver.implicitly_wait(3)` : 셀레니움 드라이버의 함수

`driver.close()` : 종료.

`import smtplib`

`from email.message import EmailMessage`

import pickle

pw = "password"

pickle.dump(pw, open('pw.pickle', 'wb')) : pw.pickle이란 파일에 데이터를 입력한다. 파일이 없다면 새로 생성. (write byte)

pw = **pickle.load**(open('pw.pickle', 'rb')) : pw.pickle이란 파일에서 데이터를 읽어온다. (read byte)

```
#로그인 아이디 / 앱 비밀번호 (설정 -> 보안 -> 앱 비밀번호)
myid = getpass.getpass() # 이메일 아이디를 넣는다.
mypw = getpass.getpass() # 앱 비밀번호를 넣는다.

pickle.dump(myid, open('id.pickle', 'wb')) # 내 이메일 ID를 id.pickle로 저장.
pickle.dump(mypw, open('pw.pickle', 'wb')) # 내 앱 비밀번호를 pw.pickle로 저장.

# 이후 위에 로그인 아이디 / 앱 비밀번호는 주석처리하고 pickle로 불러와 사용하기만 하면 된다.
userid = pickle.load(open('id.pickle', 'rb'))
userpw = pickle.load(open('pw.pickle', 'rb'))

# smtp를 이용해 gmail에 로그인
smtp_gmail.login(userid, userpw)
```

시각화(표, 그래프, 차트)

```
import matplotlib.pyplot as plt
from matplotlib import rc
import seaborn as sns
%matplotlib inline

rc('font', family='AppleGothic')
plt.rcParams['axes.unicode_minus'] = False
```

경고 무시

```
import warnings
warnings.filterwarnings(action="ignore")
```

디렉토리 구별하기

os.path로 만든 디렉토리가 아닌 경우 디렉토리는 반드시 다음과 같이 정해져야한다

/Users/path (X)

/Users/path/ (O) 이렇게 끝나야 검색이 된다. 혹시 안 될 경우 뒤에 / 를 붙여보자. 된다면 os.path를 이용해 / 가 안 붙어도 작동하게 만들어준다.

디렉토리 목록만 리스트로 추출

```
## 둘 중 아무거나 하나 사용

# 입력 받은 디렉토리에서 디렉토리 목록만 리스트로 추출 (os.path.isfile)
def findDir(pwd):
    directory = []
    for eachDir in os.listdir(pwd):
        if os.path.isfile(eachDir):
            continue
        directory.append(eachDir)

    return directory

# 입력 받은 디렉토리에서 디렉토리 목록만 리스트로 추출 (os.path.isdir)
def findDir(pwd):
    directory = []
    for eachDir in os.listdir(pwd):
        fullDir = os.path.join(pwd, eachDir)
        if os.path.isdir(fullDir):
            directory.append(eachDir)

    return directory
```

findDir(basedir)

```
['0_zero',
 '9_nine',
 '6_six',
 '4_four',
 '2_two',
 '7_seven',
 '5_five',
 '1_one',
 '3_three',
 '8_eight']
```

디렉토리 목록을 전체 디렉토리 리스트로 추출

```
## 둘 중 아무거나 하나 사용
# 입력 받은 디렉토리에서 디렉토리만 전체 디렉토리 리스트로 추출 (os.path.isfile)
def findFullDir(pwd):
    directory = []
    for eachDir in os.listdir(pwd):
        if os.path.isfile(eachDir):
            continue
        directory.append(os.path.join(pwd, eachDir))

    return directory

# 입력 받은 디렉토리에서 디렉토리만 전체 디렉토리 리스트로 추출 (os.path.isdir)
def findFullDir(pwd):
    directory = []
    for eachDir in os.listdir(pwd):
        fullDir = os.path.join(pwd, eachDir)
        if os.path.isdir(fullDir):
            directory.append(fullDir)

    return directory
```

findFullDir(basedir)

```
['/Users/          /kopo_python_dataset/MNIST/trainingSet/0_zero',
 '/Users/          /kopo_python_dataset/MNIST/trainingSet/9_nine',
 '/Users/          /kopo_python_dataset/MNIST/trainingSet/6_six',
 '/Users/          /kopo_python_dataset/MNIST/trainingSet/4_four',
 '/Users/          /kopo_python_dataset/MNIST/trainingSet/2_two',
 '/Users/          /kopo_python_dataset/MNIST/trainingSet/7_seven',
 '/Users/          /kopo_python_dataset/MNIST/trainingSet/5_five',
 '/Users/          /kopo_python_dataset/MNIST/trainingSet/1_one',
 '/Users/          /kopo_python_dataset/MNIST/trainingSet/3_three',
 '/Users/          /kopo_python_dataset/MNIST/trainingSet/8_eight']
```

파일 구별하기

파일 목록만 리스트로 추출

```
# 입력 받은 디렉토리에서 파일 목록만 리스트로 추출 (os.path.isfile)
def findFile(pwd):
    file = []
    for eachFile in os.listdir(pwd):
        if os.path.isfile(eachFile):
            file.append(eachFile)

    return file
```

파일 목록을 전체 디렉토리 리스트로 추출

```
# 입력 받은 디렉토리에서 파일만 전체 디렉토리 리스트로 추출 (os.path.isfile)
def findFileFullDir(pwd):
    directory = []
    for eachFile in os.listdir(pwd):
        if os.path.isfile(eachFile):
```

```
directory.append(os.path.join(pwd, eachFile))
```

```
return directory
```

DB 접속 엔진 관련 (v1.1.2 updated 20.05.27)

PrintImport.py
0.00MB

모듈에 대한 다양한 목록은 아래 글의 가장 하단 부분을 참고.

[2020/04/30 - \[개발자/용어... 그 외의 것들...\] - pip, brew 명령어](#)