PYTHON SESSION

By Edu
@ 이영준 고병욱

Date _ 2018.03.24

CONTENTS

01 Python 기초

클래스(Class)

공통된 목적으로 사용하기 위한 변수와 함수(메소드)를 하나의 틀로 묶은 것

새로운 자료형을 만들 수 있음

자체적인 메소드를 정의할 수 있음

01 Python 기초

클래스(Class)

클래스를 사용하는 이유

자칫 복잡해질 수 있는 프로그램의 구조를 클래스 단위로 묶어 간결하게 나타낼 수 있음

데이터를 효율적으로 관리할 수 있는 틀을 제공

01 Python 기초

클래스(Class)

클래스 정의하기

```
class 클래스명:
        (변수)

        def 메소드1():
                (실행)

        def 메소드2():
                (실행)
```

* 클래스 내부에서 사용할 변수와 메소드를 지정할 수 있음

01 Python 기초

클래스(Class)

클래스 정의 예시 코드

```
class data_analysis:
        blank = []

        def mean(self, a,b,c):
                _mean = (a+b+c)/3
                return _mean
```

01 Python 기초

클래스(Class)

클래스 정의 예시 코드

```
class data_analysis:
        blank = []

        def mean(self, a,b,c):
                _mean = (a+b+c)/3
                return _mean

test = data_analysis()
test.mean(1,2,3) → 2.0 모니터에 출력
```

01 Python 기초

클래스(Class)

클래스 정의 예시 코드

```
class data_analysis:
        blank = []

        def mean(self, a,b,c):
                _mean = (a+b+c)/3
                return _mean

test = data_analysis()
print(test.blank) → [] 모니터에 출력
```

01 Python 기초

클래스(Class)

객체, 인스턴스

객체: 클래스에 의해 만들어진 실체

ex) 이전 슬라이드의 test는 객체임

인스턴스: 객체를 클래스와의 관계로 부를 때 사용

ex) test 객체는 data_analysis 클래스의 인스턴스

01 Python 기초

클래스(Class)

변수의 타입은 해당 변수의 클래스를 나타냄

print(type(123)) → <class 'int'>

01 Python 기초

클래스(Class)

생성자

인스턴스를 만들 때 자동으로 실행되는 메소드

__init__()

01 Python 기초

클래스(Class)

생성자 코드 예시

```
class data_analysis:
        def __init__(self, a, b, c):
                self.a = a
                self.b = b
                self.c = c

        blank = []

        def mean(self):
                _mean = (self.a+self.b+self.c)/3
                return _mean
```

01 Python 기초

클래스(Class)

self.변수명: 인스턴스변수

인스턴스변수는 클래스 내 다른 메소드에서 참조 가능

메소드의 인자가 존재한다면 첫 번째 인자는 self를 넣어줘야 함 (인자가 없으면 해당되지 않음)

self는 만들어질 각각의 인스턴스를 지칭함

test = data_analysis()에서 test가 클래스의 self에 해당됨

01 Python 기초

연습문제

다음 클래스를 완성해주세요

```
class stock_analysis:
        def __init__(self,code):
                ...
        def close_mean(self):
                ...
        def close_variance(self):
                ...
        def close_std(self):
                ...
        def volume_mean(self):
```

01 Python 기초

연습문제

1. 종목명을 인자로 받아주세요
2. __init__에서 인스턴스 변수를 만들어주세요
self.latest_close, _self.latest_open
_self.latest_low, _self.latest_high
3. 종가 평균, 분산, 표준편차, 종가5일이동평균을 리턴하는 메소드를 만들어주세요
4. 파일명/경로가 잘못되었다면 “파일명 혹은 경로가 잘못되었습니다”를 출력해주세요

*5일이동평균은 날짜를 key, 해당일의 MA5를 value로 하는 딕셔너리를 리턴해주세요

01 Python 기초

클래스(Class)

클래스 상속

기존 클래스의 변수와 메소드를 가져와 원하는 부분만 수정해 클래스를 만들 수 있음

만들고자 하는 클래스와 비슷한 클래스를 상속하여 효율적으로 코딩할 수 있음

class 클래스명(상속받는클래스명):
    (수정)

01 Python 기초

클래스(Class)

클래스 상속 코드 예시

```
class _inheritance(data_analysis):
        pass

test = _inheritance()
test.mean(1,2,3) → 2.0 모니터에 출력
```

data_analysis 클래스의 메서드를 그대로 사용할 수 있음

01 Python 기초

클래스(Class)

클래스 상속 코드 예시

```
class _inheritance(data_analysis):
        pass

test = _inheritance()
test.mean(1,2,3) → 2.0 모니터에 출력
```

data_analysis 클래스의 메서드를 그대로 사용할 수 있음

01 Python 기초

클래스(Class)

클래스 상속 코드 예시

```
class _inheritance(data_analysis):
        def sum(self):
                return self.a + self.b + self.c
```

기존의 클래스에 없던 메소드를 추가할 수 있음

01 Python 기초

클래스(Class)

메서드 오버라이딩

상속한 클래스의 메소드를 덮어 쓰는 것

Ex) data_analysis 클래스의 mean을 다시 정의

01 Python 기초

클래스(Class)

메서드 오버라이딩 코드 예시

```
class _inheritance(data_analysis):
        def mean(self):
                _mean = (self.a+self.b+self.c) // 3
                return _mean

test = _inheritance(1,3,5)
result = _inheritance.mean()
print(result) → 4를 화면에 출력
```

01 Python 기초

클래스(Class)

모듈

다른 파이썬 파일에서 사용할 수 있도록 클래스, 함수, 변수 등을 포함하고 있는 파일

이미 구현된 코드들을 가져와서 사용할 수 있음

작업의 효율성을 높혀줌

01 Python 기초

클래스(Class)

모듈 불러오기

해당 모듈 파일이 있는 디렉토리에서 import 사용

모듈 파일이 환경변수에 등록되어 있다면 어느 디렉토리에서나 호출 가능

01 Python 기초

클래스(Class)

모듈 불러오기

```
import pandas

df = pandas.read_csv("test.csv")
```

pandas 모듈의 read_csv라는 함수를 호출함

01 Python 기초

클래스(Class)

모듈 불러오기

```
import pandas as pd

df = pd.read_csv("test.csv")
```

import ___ as ___와 같이 축약해서 사용 가능

01 Python 기초

클래스(Class)

모듈 불러오기

```
from pandas import read_csv

df = read_csv("test.csv")
```

모듈 내의 함수를 바로 import할 수 있음

01 Python 기초

클래스(Class)

모듈 불러오기

```
from pandas import *
from pandas import read_csv, Dataframe

df = read_csv("test.csv")
```

*을 사용해 모든 함수를 호출할 수 있으며 여러 함수를 지정하는 것도 가능함

함수 외에도 변수, 클래스도 동일한 방법으로 호출이 가능함

01 Python 기초

클래스(Class)

if __name__ == “__main__”: 구문

모듈 파일이 다음과 같을 때

```
# module file1
def sum(a,b):
        return a + b
print(“module file1 입니다”)
```

해당 모듈을 import 하면 “module file1 입니다” 출력

01 Python 기초

클래스(Class)

if __name__ == “__main__”: 구문

위 조건문은 해당 파일이 직접 실행되었을때만 충족

import하거나 인터프리터에서 사용하면 불리언 값이 거짓으로 나옴

01 Python 기초

외장함수

pickle

리스트, 튜플, 딕셔너리와 같은 파이썬 객체 자체를 저장할 수 있음

입력 예시

```
import pickle
my_list = [1,2,3,4,5,]
with open('my_list.txt', 'wb') as f:
        pickle.dump(my_list, f)
```

01 Python 기초

외장함수

pickle

리스트, 튜플, 딕셔너리와 같은 파이썬 객체 자체를 저장할 수 있음

로드 예시

```
import pickle
with open('my_list.txt', 'rb') as f:
        my_list = pickle.load(f)
```

01 Python 기초

Pandas 자료구조

Series
인덱스가 있는 1차원 배열 자료형

```
import pandas as pd
a = pd.Series(['a','b','c','d','e'])
```

```
0 a
1 b
2 c
3 d
4 e
```

01 Python 기초

Pandas 자료구조

Series
인덱스가 있는 1차원 배열 자료형

```
import pandas as pd
a = pd.Series(['a','b','c','d','e'])
```

```
0 a
1 b
2 c
3 d
4 e
```

01 Python 기초

Pandas 자료구조

Series
인덱스가 있는 1차원 배열 자료형

```
import pandas as pd
a = pd.Series(['a','b','c'], index=['no1','no2',no3'])
```

```
no1 a
no2 b
no3 c
```

01 Python 기초

Pandas 자료구조

Series 연산

```
a = pd.Series([1,2,3,4,5])
print(a * 100)
```

```
0 100
1 200
2 300
3 400
4 500
```

01 Python 기초

Pandas 자료구조

Series 인덱싱

인덱스 넘버, 인덱스 벨류를 통해 인덱싱 가능

```
import pandas as pd
a = pd.Series(['a','b','c'], index=['no1','no2',no3'])

print(a[0]) → 'a' 화면에 출력
print(a["no1"]) → 'a' 화면에 출력
```

01 Python 기초

Pandas 자료구조

Series 슬라이싱

인덱스 넘버, 문자열 라벨을 통해 인덱싱 가능

```
import pandas as pd
a = pd.Series(['a','b','c'], index=['no1','no2',no3'])
```

print(a[0:2]) → 'a', 'b' 화면에 출력
print(a["no1":"no3"]) → 'a' , 'b', 'c'화면에 출력

*문자열 라벨 인덱싱은 마지막 인덱스도 포함

01 Python 기초

Pandas 자료구조

Series 논리연산

조건문으로 데이터 취사선택 가능

```
import pandas as pd
a = pd.Series([1,2,3], index=['no1','no2',no3'])

print(a[a < 2]) → 1 화면에 출력
```

01 Python 기초

Pandas 자료구조

Series 논리연산

조건문으로 데이터 취사선택 가능

```
import pandas as pd
a = pd.Series([1,2,3], index=['no1','no2',no3'])

print(a[a < 2]) → 1 화면에 출력
```

01 Python 기초

Pandas 자료구조

Dataframe
엑셀과 유사한 형태의 자료구조

여러 Series가 합쳐져 있는 형태임

```
temp = {“수업”:[“국어”, “영어”,”수학”],
        “숙제”:[True, False, False],
        “학점”:[2,3,3]}

df = pd.DataFrame(temp)
```

01 Python 기초

Pandas 자료구조

Dataframe
딕셔너리 자료형으로 만들 수 있음
리스트를 통해 인덱스를 만들 수 있음

```
temp = {“수업”:[“국어”, “영어”,”수학”],
        “숙제”:[True, False, False],
        “학점”:[2,3,3]}

df = pd.DataFrame₩
        (temp, index=[‘1교시’,’2교시‘,’3교시‘])
```

01 Python 기초

Pandas 자료구조

Dataframe 인덱싱

df

	수업	숙제	학점
1교시	국어	True	2
2교시	영어	False	3
3교시	수학	False	3

print(df['숙제'])

출력값:
1교시 국어
2교시 영어
3교시 수학

→Series 자료형

01 Python 기초

Pandas 자료구조

Dataframe column 인덱싱

df

	수업	숙제	학점
1교시	국어	True	2
2교시	영어	False	3
3교시	수학	False	3

```
print(df[['숙제']])
```

출력값:
1교시 국어
2교시 영어
3교시 수학

→Dataframe 자료형

01 Python 기초

Pandas 자료구조

Dataframe column 인덱싱

df

	수업	숙제	학점
1교시	국어	True	2
2교시	영어	False	3
3교시	수학	False	3

```
print(df[['숙제']])
```

출력값:
1교시 국어
2교시 영어
3교시 수학

→Dataframe 자료형

01 Python 기초

Pandas 자료구조

Dataframe column 슬라이싱

df

	수업	숙제	학점
1교시	국어	True	2
2교시	영어	False	3
3교시	수학	False	3

```
print(df[['수업','숙제']])
```

출력값:

	수업	숙제
1교시	국어	True
2교시	영어	False
3교시	수학	False

01 Python 기초

Pandas 자료구조

Dataframe column 슬라이싱

df

	수업	숙제	학점
1교시	국어	True	2
2교시	영어	False	3
3교시	수학	False	3

```
print(df[['수업','숙제']])
```

출력값:

	수업	숙제
1교시	국어	True
2교시	영어	False
3교시	수학	False

01 Python 기초

Pandas 자료구조

Dataframe column 추가

df['학점*100'] = df['학점'] * 100

	수업	숙제	학점	학점*100
1교시	국어	True	2	200
2교시	영어	False	3	300
3교시	수학	False	3	300

01 Python 기초

Pandas 자료구조

Dataframe column 삭제

del df['학점*100']

	수업	숙제	학점
1교시	국어	True	2
2교시	영어	False	3
3교시	수학	False	3

01 Python 기초

Pandas 자료구조

Dataframe column 조건문 슬라이싱

df[df['학점']<3]

	수업	숙제	학점
1교시	국어	True	2

01 Python 기초

Pandas 자료구조

Dataframe row 인덱싱, 슬라이싱
항상 슬라이싱의 형태로 콜론(:)을 사용해야함

df[:'1교시']

	수업	숙제	학점
1교시	국어	True	2

01 Python 기초

Pandas 자료구조

Dataframe row 인덱싱, 슬라이싱
항상 슬라이싱의 형태로 콜론(:)을 사용해야함

df[:'1교시']

	수업	숙제	학점
1교시	국어	True	2

01 Python 기초

Pandas 자료구조

Dataframe 연산

df['합'] = df['column1'] + df['column2']

'합'이라는 column이 생성되며 column1과 column2를 더한 값을 지님

사칙연산 모두 가능

01 Python 기초

Pandas

pandas 파일 출력

```
df = pd.read_csv("example.csv")
```

*파일에 한글이 포함되어 있는 경우

```
df = pd.read_csv("example.csv", engine='python')
```

01 Python 기초

Pandas

pandas 파일 입력

df.to_csv(“example.csv”)

01 Python 기초

Pandas

pandas 주요 메소드

mean(), var(), std(), min(), max() 등

df = pd.read_csv("네이버.csv", engine='python')

print(df['종가'].mean()) → 660597.9845402767

01 Python 기초

Pandas

pandas 주요 메소드

apply(): 컬럼에 특정 함수 적용

```
df = pd.read_csv("네이버.csv", engine='python')

df[['종가', '시가']].apply₩
                    (lambda x: (x/2), axis=1)
```

01 Python 기초

Pandas

pandas 주요 메소드

sort_values([column])

df = pd.read_csv("네이버.csv", engine='python')

df.sort_values(['종가'], inplace=True)

*inplace 값이 True이면 따로 할당해줄 필요 없음

01 Python 기초

연습문제

1. 종가를 적절히 수정해주세요
2. 네이버 주식의 당일 상승폭 칼럼을 만들어주세요
3. 3월1일부터 6월1일까지의 상승폭의 표준편차를 구해주세요
4. 거래량이 1년 평균 거래량의 1.5배 이상일때의 평균 종가를 구해주세요