

[원고] 빅데이터 수집 시스템 개발

14회차 : 시계열 데이터 다루기

내용전문가	백현숙	교수설계(한기대)	홍길동
협력업체	업체명	교수설계(협력업체)	홍길동

NCS 분류 정보	20-01-02-09	정보통신 - 정보기술 - 정보기술개발 - 빅데이터플랫폼구축
능력단위 정보	03	빅데이터 수집시스템 개발
능력단위 요소 정보	2001020903_17v1.1	빅데이터 수집시스템 설계하기

업무	작성자	버전	작성일	특이사항
원고 작성	백현숙	v.1.0	2019/09/11	2회차 원고
한기대 피드백		v.1.1		
원고 보완		v.2.0		
자문 진행		v.2.1		
원고 보완				



- 학습 목차를 작성해 주세요. (NCS 비적용 과정일 경우에는 NCS 및 능력단위 정보를 기입하지 않아도 됩니다.)

NCS 분류 정보	20-01-02-09	정보통신 - 정보기술 - 정보기술개발 - 빅데이터플랫폼구축
능력단위 정보	능력단위코드 기입	빅데이터 수집시스템 개발

[illegible]

과정명	모듈명	회차명
-----	-----	-----

◆ 학습열기

시계열데이터는 일정 시간 간격으로 배치된 데이터들의 수열을 말합니다. 시간에 종속된 형태의 데이터를 말하는데 데이터분석에서 시계열데이터를 기반한 분석은 특별히 까다롭습니다. 이 장에서는 시계열 분석까지는 다루지 않습니다. 분석은 이 과정의 범위를 벗어나는 부분이고, 우리는 여기에서 날짜와 시간 데이터를 다루는 방법에 대해서 학습합니다. 기본적으로 파이썬이 제공하는 날짜와 시간관련 함수들에 대해 살펴보고, 판다스가 제공하는 날짜와 시간관련 api들을 살펴보도록 하겠습니다.

파이썬은 datetime 모듈을 제공하여 오늘 날짜를 얻거나, 현재 날짜와 시간정보를 얻거나 날짜와 날짜사이의 연산을 통해 앞으로의 기한이 얼마나 남았는지를 확인하거나, 채용일로 부터 얼마만큼의 기간 일을 했는지, 쿠팡클래스의 마감일자가 지나면 자동으로 클래스가 마감해야 한다든지, 호텔이나 펜션등의 예약일자가 이미 지나간 날짜는 예약이 되지 않게 해야 한다던지의 일을 할때 유용합니다. 날짜와 시간은 수치 데이터는 아니지만 때로는 수치데이터처럼 연산을 수행하여 위와 같은 일을 처리해야 합니다. 이 장에서 날짜모듈을 이용하여 필요한 정보를 추출하고 연산하는 방법들을 살펴보도록 하겠습니다.

학습자가 학습 화면에 구성된 내용을 보고 있다고 가정하고, 학습 설명으로 함께 들어야 할 음성 내용을 자세하게 기입해 주세요. → 이걸 나레이션해요? 저도 이런걸 나레이션하면 좋겠어요..---

과정명	모듈명	회차명
-----	-----	-----

◆ 학습내용

- 시계열 데이터 다루기

◆ 학습목표

- 날짜데이터와 시간데이터를 문자열 데이터로 전환하거나, 데이터로부터 날짜와 시간등의 데이터를 추출할 수 있다
- 날짜와 시간에 적절한 연산을 수행하여 미래의 시간이나 과거의 시간을 알아낼 수 있다

(기입하지 않습니다.)

간지 부분

◆ 날짜와 시간 API

날짜와 시간데이터 다루기

(기입하지 않습니다.)

◆ DateTime 모듈

- DateTime모듈은 파이썬에서 날짜와 시간을 다루기 위해 제공됩니다.
- 이 모듈은 파이썬이 기본으로 지원하는 모듈이며, 날짜 및 시간에 대한 산술연산을 지원합니다.
- DateTime을 이용하여 현재 날짜와 시간을 얻는 방법 은 다음과 같습니다.

```
import datetime ← 모듈을 import 한다
datetime_object = datetime.datetime.now() <- 현재 시간을 얻는다

print(datetime_object) <- 객체를 출력한다

결과 : 2019-10-16 15:50:07.625273
```

◆ DateTime 모듈

- dir 함수를 이용해 클래스 내부의 구조를 확인할 수 있습니다

```
import datetime ← 모듈을 import 한다  
print(dir(datetime)) ← 모듈 내부의 내용을 확인한다
```

```
['MAXYEAR', 'MINYEAR', '__builtins__', '__cached__', '__doc__', '__file__', '__loader__',  
'__name__', '__package__', '__spec__', 'date', 'datetime', 'datetime_CAPI', 'sys', '  
time', 'timedelta', 'timezone', 'tzinfo']
```

◆ DateTime 모듈

- DateTime 모듈은 다음 4개의 클래스로 구성됩니다.
- - date Class - date 클래스는 날짜에 대한 정보를 다룹니다. 연도, 월, 일
 - time Class - time 클래스는 시간에 대한 정보를 다룹니다.
 - datetime Class - datetime 클래스는 날짜와 시간에 대한 모든 정보를 다룹니다.
 - timedelta Class - timedelta 클래스는 두 날짜와 시간 사이의 차이를 다룹니다

◆ Date class

- 현재 날짜를 구하려면 Date클래스의 today함수를 사용합니다.

```
import datetime
```

```
date_object = datetime.date.today()  
print(date_object)
```

결과 : 2019-10-19

◆ Date class

- 생성자를 이용하여 특정 연도와 월, 일을 이용하여 Date객체를 생성하기도 합니다 .

```
import datetime  
d = datetime.date(2019, 10, 13)
```

```
#생성자가 연도, 월, 일 을 입력 받는다, 세개의 매개변수가 있어야한다  
print(d)
```

결과 : 2019-10-13

◆ Date class

- 생성자를 이용하여 특정 연도와 월, 일을 이용하여 Date객체를 생성하기도 합니다 .

```
import datetime  
d = datetime.date(2019, 10, 13)
```

```
#생성자가 연도, 월, 일 을 입력 받는다, 세개의 매개변수가 있어야한다  
print(d)
```

결과 : 2019-10-13

◆ Date 클래스

- from 절을 이용하여 객체를 로딩할 수 도 있습니다. from절로 모듈로부터 date 클래스 객체만 import 합니다.
- 이 경우에는 모듈명을 생략하고 객체를 생성할 수 있습니다

```
from datetime import date  
d = date(2019, 10, 13)  
print(d)
```

◆ timestamp로 부터 날짜 정보 얻기

- timestamp는 유닉스/리눅스 계열에서 사용하는 시간을 나타내는 정수입니다.
- 1970년1월1일 0시0분0초부터 몇초가 지났는 지를 나타내는 정수값입니다
- 이 타임스탬프는 지구자전축의 움직임을 고려하는 윤초(Leap Second)까지 반영하여 아주 엄격하게 정의한 숫자가 아니라 단순히 하루를 86400초로 계산하여 순차적으로 증가시킨 것입니다.

```
from datetime import date
value = 1567345678
timestamp = date.fromtimestamp(1567345678)
print("timestamp =", 1567345678)
print("Date =", timestamp)

결과 :
timestamp = 1567345678
Date = 2019-09-01
```

◆ DateTime클래스

- DateTime 클래스의 now 함수는 현재의 날짜와 시간에 대한 DateTime객체를 반환합니다
- 이 객체는 year, month, day 필드를 통해 연도와 월, 일 정보를 추출합니다.

```
import datetime
today = datetime.datetime.now()
print("년도 :", today.year)
print("월 :", today.month)
print("일 :", today.day)
```

결과 :

년도 : 2019

월 : 10

일 : 19

◆ DateTime클래스를 이용한 시간 알아내기

- 이 객체에는 hour, minute, second 필드를 이용해 시간과 분, 초를 추출할 수 있습니다.

```
import datetime
today = datetime.datetime.now()
print("시간 : ", today.hour)
print("분 : ",today.minute)
print("초 : ",today.second)
```

결과 :
시간 : 11
분 : 10
초 : 35

- ◆ DateTime클래스를 이용한 요일 알아내기
 - DateTime클래스는 weekday라는 함수를 제공합니다
 - weekday함수는 요일을 숫자로 반환합니다.
 - 월요일:0, 화요일:1, 수요일:2, 목요일:3, 금요일:4, 토요일:5 일요일:6

```
import datetime
today = datetime.datetime.now()
print("년도 :", today.year)
print("월 :", today.month)
print("일 :", today.day)
```

결과 :

년도 : 2019

월 : 10

일 : 19

◆ DateTime클래스를 이용한 시간 알아내기

- DateTime객체에는 weekday()함수를 제공하는데 이 함수는 요일에 대한 정보를 반환합니다
- 월요일:0, 화요일:1, 수요일:2, 목요일:3, 금요일:4, 토요일:5 일요일:6

```
weekday = ["월요일", "화요일", "수요일", "목요일", "금요일", "토요일", "일요일"]  
today = datetime.datetime.now()  
print("요일 : ",today.weekday())  
day = today.weekday()  
print("{0}-{1}-{2} 는 {3} 입니다".format(today.year, today.month, today.day, weekday[day]))
```

◆ datetime 모듈 예제
파일명 : exam14_1.py

```
import datetime

print("현재 날짜와 시간 알아보기")
today = datetime.datetime.now()
print(today)

#내부구조 확인-클래스의 내부구조를 확인할 수 있다
print()
print(dir(datetime))
print()

#현재 날짜
print()
date_object = datetime.date.today()
print(date_object)
```

#연도, 월, 일 을 입력하여 date객체 생성하기

```
d = datetime.date(2019, 10, 13)
print(d)
```

#from 절을 이용하여 모듈을 import 한다

#이럴경우에는 모듈명을 사용하지 않고 클래스명만으로 객체 생성이 가능하다

```
from datetime import date
a = date(2019, 10, 13)
print(a)
```

#timestamp로 부터 날짜 추출하기

```
value = 1567345678
timestamp = date.fromtimestamp(1567345678)
print("timestamp =", 1567345678)
print("Date =", timestamp)
```

#각 필드를 이용해 날짜와 시간 추출하기

```
today = datetime.datetime.now()
print(type(today))
print("년도 : ", today.year)
print("월 : ", today.month)
print("일 : ", today.day)
```

```
today = datetime.datetime.now()
print("시간 : ", today.hour)
print("분 : ", today.minute)
print("초 : ", today.second)
```

```
#요일 월요일:0, 화요일:1, 수요일:2, 목요일:3, 금요일:4, 토요일:5 일요일:6
weekday = ["월요일", "화요일", "수요일", "목요일", "금요일", "토요일", "일요일"]
today = datetime.datetime.now()
print("요일 : ", today.weekday())
day = today.weekday()
print("{0}-{1}-{2} 는 {3} 입니다".format(today.year, today.month, today.day, weekday[day]))
```

◆ datetime.timedelta

- timedelta 는 두 날짜와 시간 사이의 차이를 저장하는 클래스입니다
- date 객체와 date객체간에 연산의 결과는 timedelta 객체로 반환됩니다

```
from datetime import datetime, date
d1 = date(2019,10, 13)
d2 = date(2019,12, 31)

d3 = d2 - d1

print("올해 마지막 일자까지 남은 기간은 ", d3, "입니다")
```

과정명	모듈명	회차명
-----	-----	-----

◆ timedelta 의 인자값

- 1주 : `datetime.timedelta(weeks=1)`
- 1일 : `datetime.timedelta(days=1)`
- 1시간 : `datetime.timedelta(hours=1)`
- 1분 : `datetime.timedelta(minutes=1)`
- 1초 : `datetime.timedelta(seconds=1)`
- 1밀리초 : `datetime.timedelta(milliseconds=1)`
- 1마이크로초 : `datetime.timedelta(microseconds=1)`

내 레이 션	
--------------	--

◆ datetime 모듈 예제

파일명 : exam14_2.py

```
from datetime import datetime, date, timedelta
```

```
d1 = date(2019,10, 13)
```

```
d2 = date(2019,12, 31)
```

```
#날짜연산
```

```
d3 = d2 - d1
```

```
print("올해 남은날", d3)
```

```
print("type of d3 =", type(d3))
```

```
t1 = datetime(year = 2019, month = 10, day = 19, hour = 7, minute = 9, second = 33)
```

```
t2 = datetime(year = 2019, month = 10, day = 19, hour = 5, minute = 1, second = 13)
```

```
t3 = t1 - t2
```

```
print("t3 =", t3)
```

```
print("type of t3 =", type(t3))
```

```
올해 남은날 79 days, 0:00:00
type of d3 = <class 'datetime.timedelta'>
t3 = 2:08:20
type of t3 = <class 'datetime.timedelta'>
```

#timedelta 사이의 연산

```
t1 = timedelta(weeks = 2, days = 5, hours = 1, seconds = 33)
t2 = timedelta(days = 4, hours = 11, minutes = 4, seconds = 54)
t3 = t1 - t2
print("t3 =", t3)
```

```
t1 = timedelta(seconds = 33) #음수로 나타낼 수 도 있다
```

```
t2 = timedelta(seconds = 54)
```

```
t3 = t1 - t2
```

```
print( " t3 = " , t3)
```

```
print( " t3 = " , abs(t3))
```

```
#기간을 초로 나타내기
```

```
t = timedelta(days = 1, hours = 2, seconds = 33, microseconds = 233)
```

```
print("total seconds =", t.total_seconds())
```

#지금 부터 3시간 뒤

```
current = datetime.today()
```

```
after = current + timedelta(hours=3)
```

```
print( current )
```

```
print( after )
```

```
t3 = 14 days, 13:55:39
t3 = -1 day, 23:59:39
t3 = 0:00:21
total seconds = 93633.000233
2019-10-20 01:10:24.044908
2019-10-20 04:10:24.044908
```


◆ datetime의 format

- datetime 을 원하는 서식으로 변경하여 문자열로 출력할 수 있습니다.
- strftime() 은 datetime 오브젝트를 서식문자열을 이용해 문자열로 전환할 수 있습니다.

```
import datetime
```

```
#서식 날짜데이터 바꾸기
```

```
today = datetime.datetime.now()
```

```
print(today)
```

```
print(today.strftime('%Y-%m-%d'))
```

```
print(today.strftime('%H:%M:%S'))
```

```
print(today.strftime('%Y-%m-%d %H:%M:%S'))
```

```
2019-10-20 18:21:39.962898
```

```
2019-10-20
```

```
18:21:39
```

```
2019-10-20 18:21:39
```

◆ datetime의 format

서식	의미
%Y	year[0000~9999]
%m	month[01, 02, 03, ~ ,12]
%d	day[01,02,03,~3,1]
%H	hour[00,02, ~ ,23]
%M	minute[00,02, ~ , 59]
%S	second[00~59]
%B	month[Janury, Febury~ December]

◆ timezone

- pytz 모듈을 이용하여 특정 지역의 시간을 확인할 수 있습니다.
- 특별히 지정을 하지 않으면 날짜는 OS에 지정된 timezone을 사용합니다.
- pytz 모듈의 timezone함수에(지역이름)을 전달하면 그 지역의 시간 개체를 반환합니다
-

```
from datetime import datetime
import pytz
format = "%Y-%m-%d, %H:%M:%S"
local = datetime.now()
print("현재지역 :", local.strftime(format))

tz_NY = pytz.timezone('America/New_York')
datetime_NY = datetime.now(tz_NY)
print("뉴욕시간 :", datetime_NY.strftime(format))
```

◆ 서시과 타임존 예제

#파일명 :exam14_3.py

```
import datetime
```

#서식 날짜데이터 바꾸기

```
today = datetime.datetime.now()
```

```
print(today)
```

```
print(today.strftime('%Y-%m-%d'))
```

```
print(today.strftime('%H:%M:%S'))
```

```
print(today.strftime('%Y-%m-%d %H:%M:%S'))
```

```
print(today.strftime('%Y-%m-%d %B'))
```

```
sDate = '2019-10-10 14:22:34'
```

#str형태의 날짜를 date 타입으로 전환하기

```
date1 = datetime.datetime.strptime(sDate, '%Y-%m-%d %H:%M:%S')
```

```
print(type(date1))
```

```
print(date1)
```

#타임존

```
from datetime import datetime
import pytz
format = "%Y-%m-%d, %H:%M:%S"
local = datetime.now()
print("현재지역 :", local.strftime(format))
tz_NY = pytz.timezone('America/New_York')
datetime_NY = datetime.now(tz_NY)
print("뉴욕시간 :", datetime_NY.strftime(format))
tz_London = pytz.timezone('Europe/London')
datetime_London = datetime.now(tz_London)
print("런던시간 :", datetime_London.strftime(format))
```

#타임존

```
from datetime import datetime
import pytz
format = "%Y-%m-%d, %H:%M:%S"
local = datetime.now()
print("현재지역 :", local.strftime(format))
tz_NY = pytz.timezone('America/New_York')
datetime_NY = datetime.now(tz_NY)
print("뉴욕시간 :", datetime_NY.strftime(format))
tz_London = pytz.timezone('Europe/London')
datetime_London = datetime.now(tz_London)
print("런던시간 :", datetime_London.strftime(format))
```

◆ 달의 마지막 날 구하기

- calendar 모듈의 monthrange 함수를 사용하면 해당 달의 시작일과 마지막일을 구할 수 있습니다
- 해당연도와 월을 두면 tuple형태로 시작일과 마지막일을 반환합니다.

```
import datetime
```

```
import calendar
```

```
start_day, last_day = calendar.monthrange(2019, 10)
```

```
print("시작일 : ", start_day)
```

```
print("마지막일 : ", last_day)
```

◆ 달의 마지막 날 구하기 예제

파일명 : exam14_4.py

#날짜 연산하기

import datetime

import calendar

start_day, last_day = calendar.monthrange(2019, 10)

print(start_day)

print(last_day)

#2019년 한해의 말일들을 출력합니다

for i in range(1,13):

start_day, last_day = calendar.monthrange(2019, i)

print(i, "월의 마지막날은 ", last_day, " 입니다")

간지 부분

◆ 판다스

1. 시계열데이터 다루기

(기입하지 않습니다.)

◆ 시계열데이터

- Pandas 라이브러리에서도 시계열 데이터 타입을 제공합니다.
- pandas에서는 인덱스를 DateTimeIndex로 만들어야 합니다.
- DatetimeIndex는 특정한 순간에 기록된 타임스탬프(timestamp) 형식의 시계열 자료를 다루기 위한 인덱스이다.
- DatetimeIndex의 라벨값이 반드시 일정한 간격일 필요는 없습니다
- to_datetime 함수 : 문자열 형태의 날짜와 시간을 datetimeindex로 변환한다
- date_range 함수 : 특정범위의 날짜와 시간 데이터를 자동 생성한다

◆ 시계열데이터

-

```
import pandas as pd
import numpy as np
```

#날짜 문자열은 연도, 월, 일이 구분이 되면 된다.

```
date_str = ["2019-10-01", "2019-10-2", "2019/10/3", "20191004"]
```

```
idx = pd.to_datetime(date_str)
```

```
idx = pd.to_datetime(date_str)
```

```
print(idx)
```

```
<class 'pandas.core.indexes.datetimes.DatetimeIndex'>
DatetimeIndex(['2019-10-01', '2019-10-02', '2019-10-03', '2019-10-04'], dtype='datetime64[ns]', freq=None)
2019-10-01    1.764052
2019-10-02    0.400157
2019-10-03    0.978738
2019-10-04    2.240893
dtype: float64
```

◆ to_datetime 예제
파일명 : exam14_5.py

```
import pandas as pd
import numpy as np

#날짜 문자열은 연도-월-일이 구분이 되면 된다.
date_str = ["2019-10-01", "2019-10-2", "2019/10/3", "20191004"]
idx = pd.to_datetime(date_str)
print(type(idx))
print(idx)

np.random.seed(0)
s = pd.Series(np.random.randn(4), index=idx)
print(s)

#인덱스를 이용하여 출력할 수 있다
print( s['2019-10-01'])
print( s['2019-10-02'])
```

```
<class 'pandas.core.indexes.datetimes.DatetimeIndex'>
DatetimeIndex(['2019-10-01', '2019-10-02', '2019-10-03', '2019-10-04'], dtype='datetime64[ns]', freq=None)
2019-10-01    1.764052
2019-10-02    0.400157
2019-10-03    0.978738
2019-10-04    2.240893
dtype: float64
1.764052345967664
0.4001572083672233
0.9787379841057392
2.240893199201458
```

◆ date_range 함수

- date_range 함수는 특정 기간의 날짜데이터를 반복하여 생성하도록 도와주는 함수이다

`date_range(start=시작일, 종료일)`

- 시작일부터 종료일까지 반복하여 날짜데이터 생성

`date_range(시작일, periods=숫자)`

- 시작일부터 periods 에 지정된 숫자만큼 데이터 생성

`date_range(시작일, 종료일 또는 periods, freq)`

- 시작일부터 종료일 또는 지정된 숫자만큼을 freq에서 지정한 인수에 따라, 특정 요일에 해당하는등에 지정된 일자에 대한 데이터를 생성한다

◆ freq 인수

인수	의미	인수	의미
S	초	M	각 달(month)의 마지막 날
T	분	MS	각 달의 첫날
H	시간	BM	주말이 아닌 평일 중에서 각 달의 마지막 날
D	일(day)	BMS	주말이 아닌 평일 중에서 각 달의 첫날
B	주말이 아닌 평일	WOM-2THU	각 달의 두번째 목요일
W	주(일요일)	Q-JAN	각 분기의 첫달의 마지막 날
W-MON	주(월요일)	Q-DEC	각 분기의 마지막 달의 마지막 날

◆ resample 연산

- resample 연산을 쓰면 시간 간격을 재조정하는 리샘플링(resampling)이 가능하다.

이 때 시간 구간이 작아지면 데이터 양이 증가한다고 해서 업-샘플링(up-sampling)이라 하고
시간 구간이 커지면 데이터 양이 감소한다고 해서 다운-샘플링(down-sampling)이라 부른다.

```
import pandas as pd
import numpy as np
```

```
ts = pd.Series(np.random.randn(100), index=
    pd.date_range( "2019-1-1", periods=100, freq="D"))
print(ts.tail(20))
```

```
print(ts.resample('W').mean())
```

◆ date_range 함수 예제

파일명 : exam14_6.py

```
import pandas as pd
import numpy as np
```

```
#2019년 9월 1일부터 2019년 9월 30일까지 생성
d = pd.date_range(start="2019-9-1", end="2019-9-30")
print(d)
```

```
d = pd.date_range(start="2019-1-1", periods=60)
print(d)
```

```
#2019년 1월중 일요일만
d = pd.date_range(start="2019-10-1", end="2019-10-30", freq='W')
print(d)
```

```
#2019년 각 달의 마지막날만 데이터 생성
np.random.seed(0)
ts = pd.Series(np.random.randn(12),
               index=pd.date_range(start="2019-01-01", periods=12, freq="M"))
print(ts)
```



```
#resample 연산
ts = pd.Series(np.random.randn(100),
               index=pd.date_range("2019-1-1", periods=100, freq="D"))
print(ts.tail(20)) #마지막 20개만 출력

print(ts.resample('W').mean()) #D(일일) -> W(주)형식으로 전환
```

과정명	모듈명	회차명
-----	-----	-----

◆ 적용하기

datetime 모듈을 사용하여 다음 문제들을 해결하세요

- 1. 어제, 오늘, 내일 날짜를 출력하세요
- 2. 이달의 마지막 날은 몇일입니까
- 3. 오늘은 무슨 요일인가요?
- 4. 이달의 마지막 날까지 몇일 남았습니까?

(기입하지 않습니다.)

◆ 적용하기- 풀이

```
from datetime import datetime, timedelta
import calendar
```

```
today = datetime.now()
yesterday = today - timedelta(days=1)
tomorrow = today + timedelta(days=1)
```

#1. 오늘 어제 내일

```
print("오늘 : ", today )
print("어제 : ", yesterday)
print("내일 : ", tomorrow)
```

#2번.이달의 마지막날

```
start_day, last_day = calendar.monthrange(today.year, today.month)
print("이달의 마지막 날은 ", last_day, "입니다")
```

(기입하지 않습니다.)

◆ 적용하기 -> 풀이

#3번. 오늘은 무슨 요일인가요?

```
day=["월요일", "화요일", "수요일", "목요일", "금요일", "토요일", "일요일"]  
print("오늘은 ", day[today.weekday()], "입니다")
```

#4번 이달의 마지막날까지 며칠남았습니까?

```
last_date = datetime(today.year, today.month, last_day)  
period = last_date - today  
print( period, "남았습니다.")
```

(기입하지 않습니다.)

◆ 문제풀기

문제 1) 오늘 날짜와 시간에 대한 DateTime 객체를 얻는 함수는?

- ① today()
- ② weekday()
- ③ now()
- ④ thistime()

정답) 3

해설) 오늘 날짜와 시간은 now 함수를 사용합니다. 오늘 날짜만은 today()함수를 사용합니다

난이도) 5(1:아주어려움, 2:어려움 3: 보통 4: 쉬움 5: 아주 쉬움)

관련페이지번호) 6

(기입하지 않습니다.)

번호	문제	정답	난이도	해설	관련학습보기
2	<p>오늘이 무슨 요일인지 숫자로 요일을 반환하는 함수는?</p> <p>① now</p> <p>② today</p> <p>③ weekday</p> <p>④ day</p>	3	4	<p>weekday 함수는 해당 요일을 수치자료로 반환합니다.</p> <p>월요일을 0으로 보고 차례대로 숫자를 부여합니다</p>	16
3	<p>지금으로부터 6시간 뒤의 시간을 알아내려고 합니다. 맞는 코드를 작성하세요</p>	해설참조	1	<p>current = datetime.today()</p> <p>after = current + timedelta(hours=6)</p>	24
4	<p>2019년 2월달의 마지막일을 하려고 합니다. 올바른 코드를 작성하세요</p>	해설참조	4	<p>start_day, last_day = calendar.monthrange(2019, 2)</p>	32

번호	문제	정답	난이도	해설	관련학습 보기
5	<p>다음 중 타임스탬프에 대한 설명을 잘못된 것은 ?</p> <p>① timestamp는 유닉스/리눅스 계열에서 사용하는 시간을 나타내는 정수입니다.</p> <p>② 1970년1월1일 0시0분0초부터 몇초가 지났는 지를 나타내는 정수 값입니다</p> <p>③ 이 타임스탬프는 지구자전축의 움직임을 고려하는 윤초(Leap Second)까지 반영하여 아주 엄격하게 정의한 숫자입니다</p> <p>④ 단순히 하루를 86400초로 계산하여 순차적으로 증가시킨 것입니다</p>	3	4	타임스탬프는 단순히 하루를 86400초로 계산하여 순차적으로 증가시킨것입니다	13
6	<p>strftime함수의 서식문자와 내용을 잘못 연결한것은?</p> <p>① %Y - 연도</p> <p>② %m - 월</p> <p>③ %M - 월</p> <p>④ %d - 일</p>	3	3	월은 %m을 사용합니다. 대문자는 분을 나타냅니다	22
7	<p>strftime함수를 사용하여 오늘 날짜를 문자열의 형태로 다음 처럼 출력하려면 어떤 서식을 써야 하는지 바른것을 고르시오</p> <p>결과 : 2019-10-20 October</p> <p>① %Y-%M-%D</p> <p>② %Y-%M-%D %B</p> <p>③ %Y-%m-%d %B</p> <p>④ %Y-%m-%d %h</p>	3	2	연도 월 일 월이 영어로 나왔으므로 %Y-%m-%d %B를 사용한다	26

이 번	문제	정답	난이 도	해설	관련학습보기
8	<p>판다스 라이브러리의 to_datetime 을 이용해 DateTime 인덱스를 생성하려고 합니다. 문자열로 지정한 날짜 입력 형식이 잘못 된것은?</p> <p>① “2018, 9, 1” ② “2019-9-1” ③ “2019/90/04” ④ “20191004”</p>	1	4	날짜는 2019-10-01 이나 2019/10/02 또는 201091005 형태로 주어야 합니다.	35
9	<p>일정 기간동안 datetimeindex를 생성하려고 합니다. 2019.9.1일부터 30일간 연속된 날짜데이터를 생성하는 코드를 작성하시오</p> <p>import pandas as pd</p>	해설참조	1	pd.date_range(start="2019-9-1", periods=30)	37
10	<p>1년동안 각달의 마지막 날만 pandas 라이브러리를 이용해 작성하려고 합니다. 올바른 코드를 작성하십시오</p>	해설참조	1	pd.date_range(start="2019-01-01", periods=12, freq="M"))	38

과정명	모듈명	회차명
-----	-----	-----

◆ 핵심요약

▶ DateTime객체

- DateTime 모듈에는 Date, Time, DateTime ,TimeDelta 클래스가 있습니다.
- 현재 날짜와 시간은 DateTime클래스의 now함수를 이용해 얻을 수 있습니다
- TimeDelta 클래스는 두 날짜와의 차이나 두 시간과의 차이를 얻고자 할때 사용합니다. 날짜와 날짜는 서로 산술연산이 가능하고 음수도 가질 수 있습니다
- 각 달의 마지막날은 Calendar 객체의 monthrange 함수를 이용해 얻습니다.

▶ 시계열데이터

- 판다스 라이브러리는 to_datetime 함수를 이용해 날짜와 시간타입을 인덱스로 만들 수 있습니다.
- 날짜 인덱스를 이용해 데이터에 대한 접근이 가능합니다.
- 날짜 인덱스는 date_range 함수를 이용해 특정기간동안의 날짜데이터, 특정시점부터 원하는 일수만큼(periods), 원하는 특정일에 대한 연속된 데이터를 생성할 수 있습니다.
- resampling 을 통해 이미 작성된 샘플링의 개수를 조절 할 수 있습니다

(기입하지 않습니다.)

과정명	모듈명	회차명
-----	-----	-----

◆ 학습맺음

- 이번 회차에서는 파이썬에서 날짜와 시간을 다루는 부분을 학습해보았습니다 수고많으셨습니다

◆ 참고자료

- <https://docs.anaconda.com/anaconda/>
- <https://code.visualstudio.com/docs/getstarted/themes>
- https://pandas.pydata.org/pandas-docs/stable/getting_started/tutorials.html
- <https://www.programiz.com/python-programming/datetime>
- <https://allenjeon.tistory.com/235>
- <https://www.programiz.com/python-programming/datetime>
- <https://datascienceschool.net/view-notebook/8959673a97214e8fafdb159f254185e9/>
- <https://pandas.pydata.org/pandas-docs/version/0.24/reference/api/pandas.DatetimeIndex.html>

(기입하지 않습니다.)



회차 메타데이터

- 주요학습내용 : 해당 회차 또는 레슨의 주요학습내용을 자세히 기입해 주세요.
- 검색 키워드 : 학습자가 검색창에 어떤 검색어를 입력하면 본 회차 또는 본 레슨이 검색될 수 있을지 검색 키워드를 5개 기입해 주세요.

제목	주요학습내용	검색 키워드1	검색 키워드2	검색 키워드3	검색 키워드4	검색 키워드5
14. 시계열데이터 다루기	DateTime 모듈, Pandas library	DateTime	timedelta	timezone	to_datetime	DatetimeIndex