

# Mini Project - *WHO'S NEXT*

2조

장 원

정석원

서연주

박소현

객원멤버: 홍성석 선생님

# INDEX

1 기획

2 데이터가공 및 처리

3 웹 구현

4 웹 소개

---



친숙한 주제 선정

TRAVEL



여행 트렌드

다음 여행지는?



매년 증가하는 해외여행객

증가하는 원인과 결과에 대한 의문



많은 여행객들은 어디로 여행을 떠날까





한국관광공사



통계청

**NAVER**

검색

**WIKIPEDIA**  
The Free Encyclopedia

Country code  
Year

The screenshot displays the Studio 3T for MongoDB interface. On the left, the 'Collections' pane shows a database named 'mp01' with a collection named 'year'. The main window is split into two panes. The left pane shows a Python script named 'year\_mongo.py' with the following code:

```
resource > year_mongo.py > ...  
  
17 conn = pymongo.MongoClient('192.168.99.100', 32766)  
18 db = conn.get_database("mp01") # 있으면 가져오고, 없으면 만들어  
19 table = db.get_collection("year") # 있으면 가져오고, 없으면 만들어  
20  
21 # df = pd.read_csv('./resources/korean_chulguk.csv')  
22 # print(df)  
23 # dict1 = df.to_dict()  
24 # print(dict1)  
25  
26 for line in rdr:  
27  
28     dict1 = dict()  
29     for idx, val in enumerate(line):  
30         if idx == 0:  
31             pass  
32             dict1[column[idx]] = val  
33     print(dict1)  
34
```

The right pane shows a table view of the 'year' collection. The table has two columns: 'country' and 'year'. The data is as follows:

country	year
핀란드	15,418
europa	13,544
	15,655
[설정]의...	12,159
여 Wind...	10,602
인증합	7,450

code\_MtoO.py X

Country code

resource > code\_MtoO.py > ...

```

9 conn_m = pymongo.MongoClient('192.168.99.100',32766)
10 db = conn_m.get_database("mp01")
11 coll = db.get_collection("countrycode")
12
13 data1 = coll.find({},{'_id':False}) # id값 빼기
14 # data1 = coll.find()
15
16 for tmp in data1:
17     print(tmp)
18     sql = ""
19     INSERT INTO COUNTRY_COUNTRYNAME(CODE, C_NAME)
20     VALUES(:code, :c_name)
21     ""
22     cursor.execute(sql, tmp)
23     conn_o.commit()
24
25 conn_o.close() # 오라클 연결 끊기
26 conn_m.close() # 몽고DB 연결 끊기

```

- Data Modeler 보고서
- OLAP 보고서
- TimesTen 보고서
- 데이터 덱서너리 보고서
- 분석 뷰 보고서
- 사용자정의 보고서

35	189 세르비아	europe	0	0	0	0	0	0	0	0	0	0	0	0	0	1516	3110	1845	3217
36	190 스웨덴	europe	0	0	0	0	15376	16599	23560	19575	20598	19153	26444	29143	27952	31203	1355	0	0
37	191 리히텐슈타인	europe	0	0	0	0	97	130	134	145	95	49	71	87	83	90	89	95	207
38	192 조지아	europe	0	0	0	0	0	238	432	651	536	662	935	1419	1762	2433	2189	0	0
39	193 산마리노	europe	0	0	0	0	0	0	0	0	0	0	0	50	64	11	0	0	0

year\_MtoO.py X

Year

resource > year\_MtoO.py > ...

```

1 import pymongo
2 import re
3 import cx_Oracle as oci
4 conn_o=oci.connect('admin/1234@192.168.99.100:32764/xe',encoding="euc-kr")
5 cursor=conn_o.cursor()
6
7 conn=pymongo.MongoClient('192.168.99.100',32766)
8 db=conn.get_database("mp01") # 데이터가 들어가져야 몽고 만들어짐 없으면 db생성 있으면 가져오는거
9 coll=db.get_collection("year") # collection 있으면 가져오고 없으면 생성 이까지 테이블 만든거 데이터 들어가기 몽고에 들어감
10
11 data1=coll.find({},{'_id':False})
12 for tmp in data1:
13     dict1=dict()
14     dict1['a']=tmp['continent']
15     dict1['b']=tmp['country']
16
17     dict1['c'] = '0'
18
19     if(tmp['2000'] != '' and tmp['2000'].strip() != '-'):
20         dict1['c'] = tmp['2000'].replace(",","")
21     dict1['d'] = '0'
22     if(tmp['2001'] != '' and tmp['2001'].strip() != '-'):
23         dict1['d'] = tmp['2001'].replace(",","")

```

Windows 정품 인증  
[설정]으로 이동하여 Windows를 정품 인증합니다.

```
resource > year_MtoO.py > ...
40 dict1['j'] = tmp['2007'].strip() != '' and tmp['2007'].replace(",","")
41 dict1['k'] = '0'
42 dict1['l'] = '0'
43 if(tmp['2008'] != '' and tmp['2008'].strip() != '-'):
44     dict1['k'] = tmp['2008'].replace(",","")
45 dict1['l'] = '0'
46 if(tmp['2009'] != '' and tmp['2009'].strip() != '-'):
47     dict1['l'] = tmp['2009'].replace(",","")
48 dict1['m'] = '0'
49 if(tmp['2010'] != '' and tmp['2010'].strip() != '-'):
50     dict1['m'] = tmp['2010'].replace(",","")
51 dict1['n'] = '0'
52 if(tmp['2011'] != '' and tmp['2011'].strip() != '-'):
53     dict1['n'] = tmp['2011'].replace(",","")
54 dict1['o'] = '0'
55 if(tmp['2012'] != '' and tmp['2012'].strip() != '-'):
56     dict1['o'] = tmp['2012'].replace(",","")
57 dict1['p'] = '0'
58 if(tmp['2013'] != '' and tmp['2013'].strip() != '-'):
59     dict1['p'] = tmp['2013'].replace(",","")
60 dict1['q'] = '0'
61 if(tmp['2014'] != '' and tmp['2014'].strip() != '-'):
62     dict1['q'] = tmp['2014'].replace(",","")
63 dict1['r'] = '0'
64 if(tmp['2015'] != '' and tmp['2015'].strip() != '-'):
65     dict1['r'] = tmp['2015'].replace(",","")
66 dict1['s'] = '0'
67 if(tmp['2016'] != '' and tmp['2016'].strip() != '-'):
68     dict1['s'] = tmp['2016'].replace(",","")
69 dict1['t'] = '0'
70 if(tmp['2017'] != '' and tmp['2017'].strip() != '-'):
71     dict1['t'] = tmp['2017'].replace(",","")
72 dict1['u'] = '0'
73 if(tmp['2018'] != '' and tmp['2018'].strip() != '-'):
74     dict1['u'] = tmp['2018'].replace(",","")
75 dict1['v'] = '0'
76 if(tmp['2019'] != '' and tmp['2019'].strip() != '-'):
77     dict1['v'] = tmp['2019'].replace(",","")
78
79
80 print("aaaa:",dict1)
81
82
83 sql= """
84 INSERT INTO COUNTRY_TRAVELBYCOUNTRY(CONTINENT, COUNTRY,Y2000,Y2001,Y2002,Y2003,Y2004,Y2005,Y2006,Y2007,Y2008,Y2009,Y2010,Y2011,Y2012,Y2013,Y2014,Y2015,Y2016,Y2017,Y2018,Y2019)
85 VALUES(:a, :b, :c, :d, :e, :f, :g, :h, :i, :j, :k, :l, :m, :n, :o, :p, :q, :r, :s, :t, :u, :v)
86 """
87
```

데이터 처리 시 주의사항!!

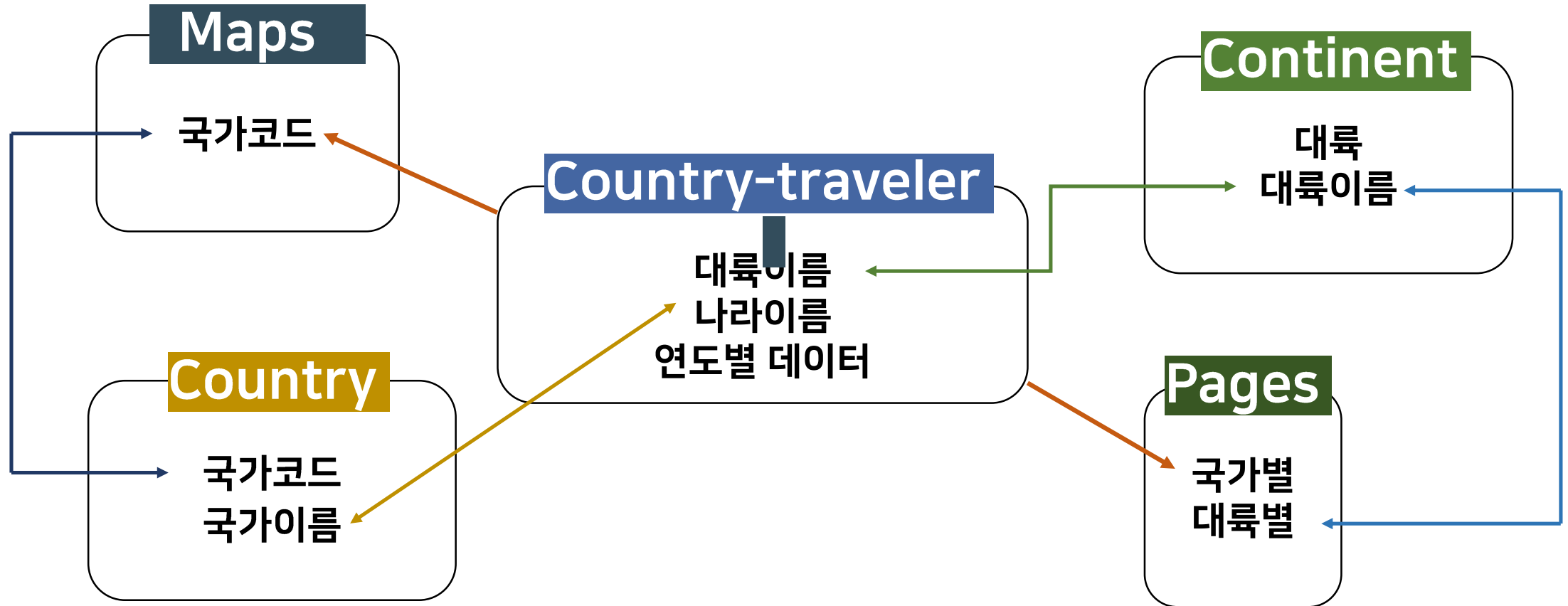
- 콤마 제거
- 공백 제거
- '-' 짝대기도 제거

Windows 정품 인증  
[설정]으로 이동하여 Windows를 정품 인증

# Don't reinvent the wheel







```
urlpatterns = [  
    re_path(r'(?P<code>[A-Z]{2})/', views.country, name='country'),      # 나라별 페이지  
    re_path(r'(?P<cont>[a-z]{4,10})/', views.continent, name="continent"), # 대륙별 페이지  
    re_path(r'^top/(?P<year>[0-9]{0,4})', views.top, name="top"),        # top 페이지  
]
```

```
# 국가명을 국가코드에서 가져온다.  
countryName = CountryName.objects.get(code=code)  
yearData = [ i for i in range(2000,2020) ]      # 2000~2019  
  
# 빈 리스트에 일치하는 국가명의 DB를 가져온다.  
countryData = list()  
obj = TravelByCountry.objects.get(country=countryName)  
  
# 가져온 data를 딕셔너리로 만든다.  
for key, value in obj.__dict__.items():  
    # 해당연도 데이터만 가져와서 변수에 담는다.  
    print(key, value)  
    if 'y' == key[0] :  
        countryData.append(value)
```

만약 데이터가 1억개라면 ?

## Intergrate

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

euskate 1/24 board app

Latest commit b08ac50 20 hours ago

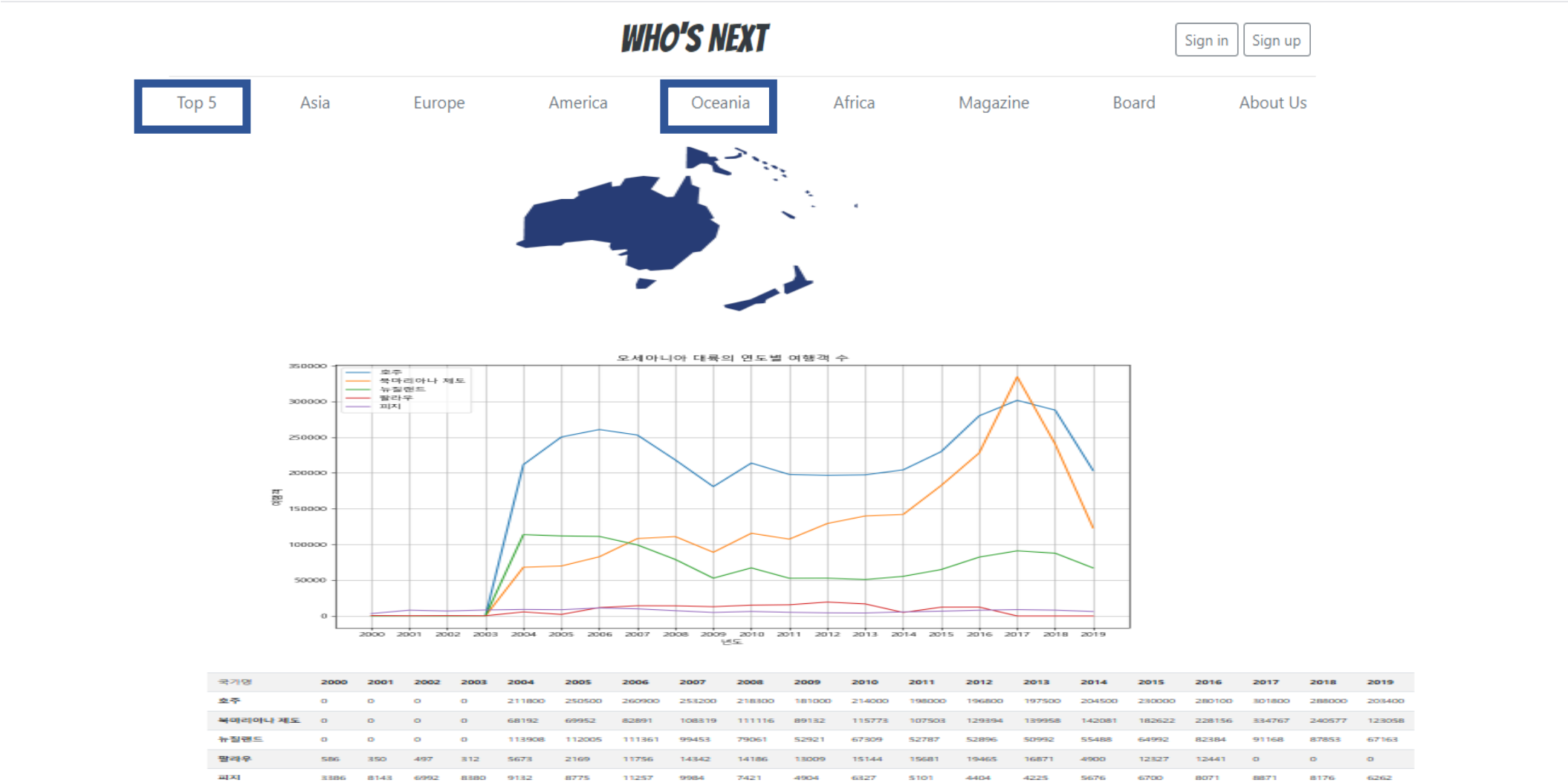
📁 .vscode/.ropeproject	1/24 member 기능 등 수정	22 hours ago
📁 board	1/24 board app 추가	20 hours ago
📁 country	1/24 board app 추가	20 hours ago
📁 magazine	1/24 member 기능 등 수정	22 hours ago
📁 member	1/24 member 기능 등 수정	22 hours ago
📁 mp01	1/24 board app 추가	20 hours ago
📁 resource	1/24 member 기능 등 수정	22 hours ago
📁 static	1/24 board app 추가	20 hours ago
📁 templates	1/24 board app 추가	20 hours ago
📄 chromedriver.exe	1/22 working commit 1	3 days ago
📄 logs.md	1/24 board app 추가	20 hours ago
📄 manage.py	first commit	8 days ago
📄 migration.md	modify	2 days ago
📄 조별과제 생각해봄.md	회의	2 days ago

Branch 이용

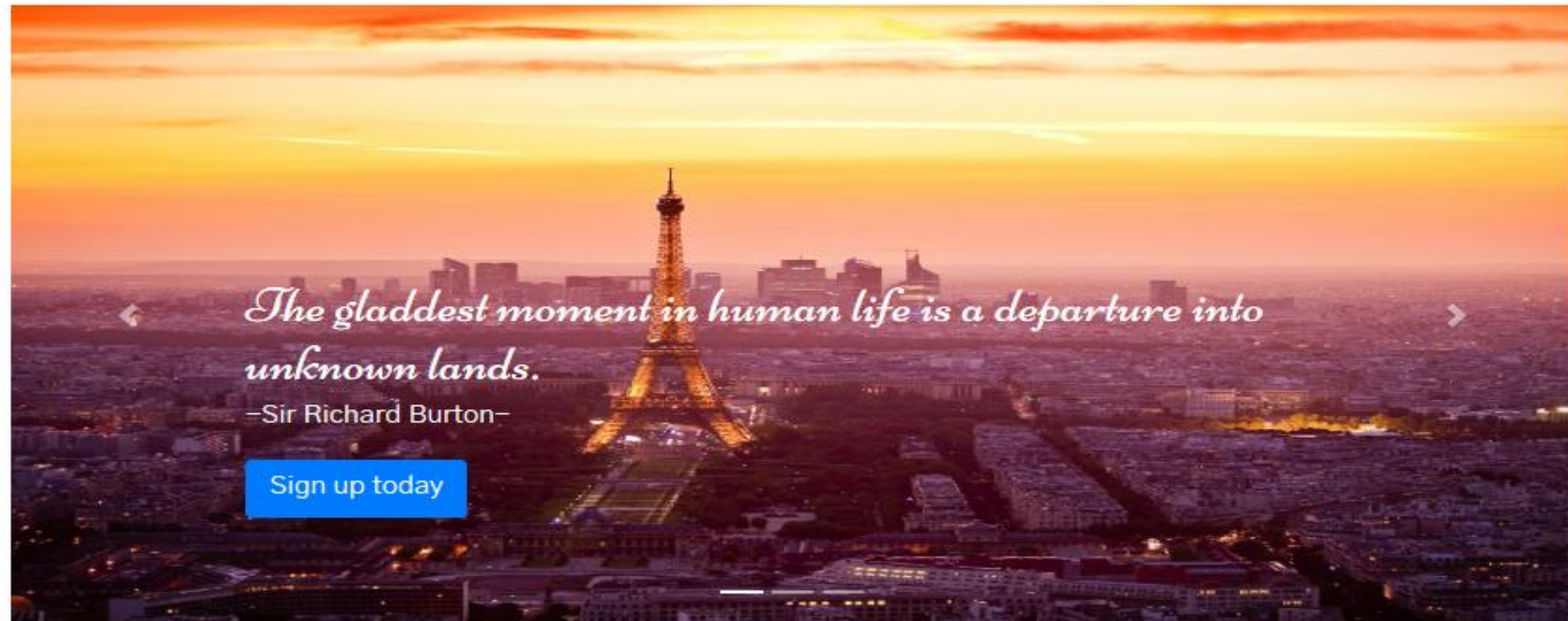
github

[Sign up](#)[About Us](#)

Window  
[설정]으로



**WHO'S NEXT**

[Sign in](#)[Sign up](#)[Top 5](#)[Asia](#)[Europe](#)[America](#)[Oceania](#)[Africa](#)[Magazine](#)[Board](#)[About Us](#)

**WHO'S NEXT**

[SIGN IN](#)[SIGN UP](#)[Top 5](#)[Asia](#)[Europe](#)[America](#)[Oceania](#)[Africa](#)[Magazine](#)[Board](#)[About Us](#)

## 회원가입

아이디

aa

비밀번호

....

비밀번호 확인

....

이름

\*\*\*\*\*

이메일

0000



이메일 주소에 '@'를 포함해 주세요. '0000'에 '@'가 없습니다.

질문 있나요?

---

Q&A





감사합니다