

Estate index  
years in korea

# 프로젝트 제작 동기

클라우드 네이티브에 관하여 공부하던 중 구글의 Colab 서비스에 대해 알게 되었습니다. 앞으로의 미래는 클라우드 환경에 발 맞춰 클라우드 환경에서 코딩 해보고 싶었습니다.

Colab의 환경속에서 테스트 파일로 **california\_housing\_test.csv**가 있었으며, 이에 대한민국에 관한 자료로 테스트 해보고 싶어 이 프로젝트를 시작하게 되었습니다.

# 기본 세팅

```
In [1]: !sudo apt-get install -y fonts-nanum
        !sudo fc-cache -fv
        !rm ~/.cache/matplotlib -rf
```

시각화 할 때 한글 폰트가 없어서 깨져 폰트 설치 코드입니다.

```
In [9]: from google.colab import drive
        #구글드라이브 연동
```

```
drive.mount('/content/drive')
```

내 구글 드라이브 마운트 하기 위한 코드입니다.

# 데이터 스크랩

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	* 통계표명 :	지가지수_연도별																
2	* 수록기간 :	1987년 ~ 2021년																
3	* 조회기간 :	1987년 ~ 2021년																
4	* 출처 :	국토교통부																
5	* 자료다운일	2022.04.14 04:03:09																
6	* 단위 :	지수																
7																		
8																		
9																		
10																		
11		지 역	1987년	1988년	1989년	1990년	1991년	1992년	1993년	1994년	1995년	1996년	1997년	1998년	1999년	2000년	2001년	
12		전국	29.652	37.792	49.872	60.135	67.816	66.963	62.017	61.664	62.004	62.601	62.795	54.255	55.85	56.224	56.975	
13		수도권	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
14		지방	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
15		대도시	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
16		시지역	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
17		군지역	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
18		서울	24.426	31.281	41.755	54.791	60.903	59.207	54.048	53.315	53.411	53.915	54.071	45.283	46.49	46.514	47.399	
19		종로구	31.431	36.087	45.959	59.477	63.084	61.434	57.841	57.57	57.661	58.286	58.73	48.653	50.112	49.852	50.739	
20		중구	32.759	39.823	50.3	61.758	63.037	61.623	57.461	57.467	58.003	58.855	59.445	48.955	50.414	50.157	51.268	
21		용산구	23.838	27.929	35.57	45.531	48.651	47.185	43.725	43.69	45.06	45.735	46.038	38.741	39.741	39.913	40.668	
22		성동구	19.739	25.238	33.505	45.883	52.32	51.784	46.779	46.154	47.139	47.736	47.784	40.265	40.998	40.875	41.512	
23		광진구	-	-	-	-	-	-	-	55.623	56.113	56.603	56.659	47.083	48.012	48.102	48.847	
24		동대문구	27.907	35.854	46.117	57.247	63.437	61.106	56.314	55.703	55.77	56.301	56.696	47.991	49.04	48.696	49.524	
25		중랑구	27.483	36.208	48.929	59.649	65.801	65.576	60.716	60.607	61.289	61.552	61.867	51.836	52.831	52.566	53.3	
26		성북구	26.576	33.777	42.102	54.047	59.666	59.756	57.018	56.995	57.573	58.226	58.834	49.355	50.806	50.792	51.583	
27		강북구	-	-	-	-	-	-	-	61.798	62.144	62.325	62.437	51.865	53.5	53.475	54.395	
28		도봉구	25.519	34.678	46.241	65.176	71.863	71.505	65.407	64.832	65.6	66.358	66.584	56.328	57.926	57.741	58.769	
29		노원구	20.259	27.462	40.318	57.466	65.99	65.019	61.613	61.158	61.342	62.025	62.26	52.194	53.686	53.636	54.818	
30		은평구	30.123	38.026	49.858	59.833	65.359	62.349	57.209	56.35	56.259	56.563	56.75	47.926	49.487	49.661	50.605	
31		서대문구	29.15	35.075	45.549	56.814	62.985	62.142	58.082	57.365	57.05	57.553	57.812	48.415	49.7	49.55	50.517	
32		마포구	24.41	31.123	39.743	50.547	55.117	55.355	51.111	50.769	51.533	51.91	52.128	44.272	45.348	45.259	46.837	
33		양천구	25.349	33.367	45.207	55.81	64.222	62.372	57.046	56.887	57.354	58.126	58.213	49.033	50.001	49.966	50.896	
34		강서구	20.018	26.784	37.207	46.71	52.855	54.082	49.877	50.412	51.918	52.769	53.304	45.634	46.841	46.829	47.422	
35		구로구	26.276	32.918	42.397	55.418	63.282	62.966	59.498	59.462	60.226	60.449	60.807	50.88	51.586	51.284	52.047	

```
df = pd.read_excel('지가지수_연도별.xlsx', sheet_name='Sheet1', header=10)
```

붉은 박스의 데이터가 필요하기에 데이터 시트의 헤더를 변경 해야 합니다.  
Header를 10으로 설정 해줍니다. 0~10으로 계산 하기 때문입니다.

# 더미 데이터 제거

...	지 역	Unnamed: 1	Unnamed: 2	1987년	1988년	1989년	1990년	1991년
0	전국	NaN	NaN	29,652	37,792	49,872	60,135	67,816
1	수도권	NaN	NaN	-	-	-	-	-
2	지방	NaN	NaN	-	-	-	-	-
3	대도시	NaN	NaN	-	-	-	-	-
4	시지역	NaN	NaN	-	-	-	-	-
...	...	...	...	...	...	...	...	...
296	NaN	합천군	NaN	65,239	70,599	78,359	84,672	87,301
297	제주	NaN	NaN	-	-	-	-	-
298	NaN	시지역	NaN	-	-	-	-	-
299	NaN	제주시	NaN	-	-	-	-	-
300	NaN	서귀포시	NaN	-	-	-	-	-

지역명은 301 rows X 3 columns 크기인 배열로 필요 없는

NaN을 제거 및 1차원 배열로 만들어주는 과정이 필요합니다.

데이터들로 그래프를 그리는데 문자열이 들어가면 오류가 생

기기 때문에 "-" 을 0으로 바꿔주는 작업이 필요합니다.

데이터 시트의 헤더의 변수 df의 일부분을 발췌한 사진입니다.

# 코드 설명

```
row = df.iloc[:, [0,1,2]]  
pd_to_np = row.to_numpy()
```

```
place_list = []
```

```
np_dict = {}
```

301 rows X 3 columns 크기의 numpy를 읽어 str 타입이면  
지역명 배열( place\_list [ ] )에 추가하는 코드입니다.

```
for i in range(0, pd_to_np.shape[0]):  
    for j in range(pd_to_np.shape[1]):  
        if type(pd_to_np[i][j]) == str :  
            place_list.append(pd_to_np[i][j])
```

```
estate_indexs = df.values[i][3:]
```

```
estate_index = np.where(estate_indexs == '-',0, estate_indexs)
```

```
np_dict[pd_to_np[i][j]] = estate_index
```

np.where 함수의 기능은 특정 문자를 바꿔주는 함수 입니다.  
또한 np\_dict은 딕셔너리 타입이며 추후에 특정 지역명 검색  
시 사용되는 딕셔너리입니다.

#estate\_indexs의 '-' 값을 0으로 바꿔주는 구문

```
"""
```

pandas를 numpy로 바꿔주며 place\_list에 지역명 추가

place\_list 리스트를 만든 이유는 nan을 지우기 위함이다.

이렇게 코드를 짰 이유는 pandas에서 nan제거 함수가 nan이 있는 함수의 행을 삭제 시켜버린다.

또한 numpy의 nan제거는 isnan() 함수가 있지만, 다중차원이면 안되는것 같다.

```
"""
```

# 딕셔너리 데이터 출력

```
{ '전국': array([29.652, 37.792, 49.872, 60.135, 67.816, 66.963, 62.017, 61.664,
62.004, 62.601, 62.795, 54.255, 55.85, 56.224, 56.975, 62.089,
64.219, 66.693, 70.013, 73.944, 76.818, 76.572, 77.303, 78.113,
79.024, 79.78, 80.685, 82.27, 84.246, 86.52, 89.876, 93.996,
97.675, 101.268, 105.494], dtype=object), '수도권': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 78.258, 78.791, 79.613, 81.133, 82.912, 85.011, 88.261,
92.797, 97.196, 101.431, 106.282], dtype=object), '지방': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 80.309, 81.493, 82.554, 84.257, 86.595, 89.187, 92.728,
96.111, 98.521, 101.001, 104.202], dtype=object), '대도시': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
74.888, 75.48, 76.259, 76.814, 77.77, 79.656, 81.861, 84.334,
88.022, 92.848, 97.226, 101.554, 106.471], dtype=object), '시지역': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 72.53,
73.742, 73.727, 73.872, 74.665, 75.35, 76.292, 77.378, 80.263,
86.338, 93.531, 98.635, 103.556, 101.726, 99.76, 101.604],
dtype=object), '군지역': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 86.072, 86.897, 88.325, 89.976, 92.011, 94.76, 97.409,
99.115, 100.518, 102.648], dtype=object), '서울': array([24.426, 31.281, 41.755, 54.791, 60.903, 59.207, 54.048, 53.315,
53.411, 53.915, 54.071, 45.283, 46.49, 46.514, 47.399, 54.888,
57.755, 60.113, 64.059, 69.933, 74.047, 73.306, 74.329, 74.722,
75.444, 75.729, 76.642, 78.683, 80.798, 83.201, 86.791, 92.097,
```

# 코드 설명

```
while True:
    input_name = input("지역을 입력하시오 : ")
    if input_name in np_dict != False:
        break
    else:
        print("지역명을 검색할 수 없습니다.")
```

그래프의 디자인 세팅하는 코드 입니다.

```
plt.figure(figsize=(len(estate_index),len(year))) #표 전체 사이즈
plt.xticks(rotation=45, fontsize=30) #x축 기울기 및 폰트 사이즈
plt.yticks(fontsize=30)
plt.tick_params(axis='x', direction='inout', length=10, pad=15, width=5, color='r') #x축 눈금
plt.grid(True) #표 전체눈금
```

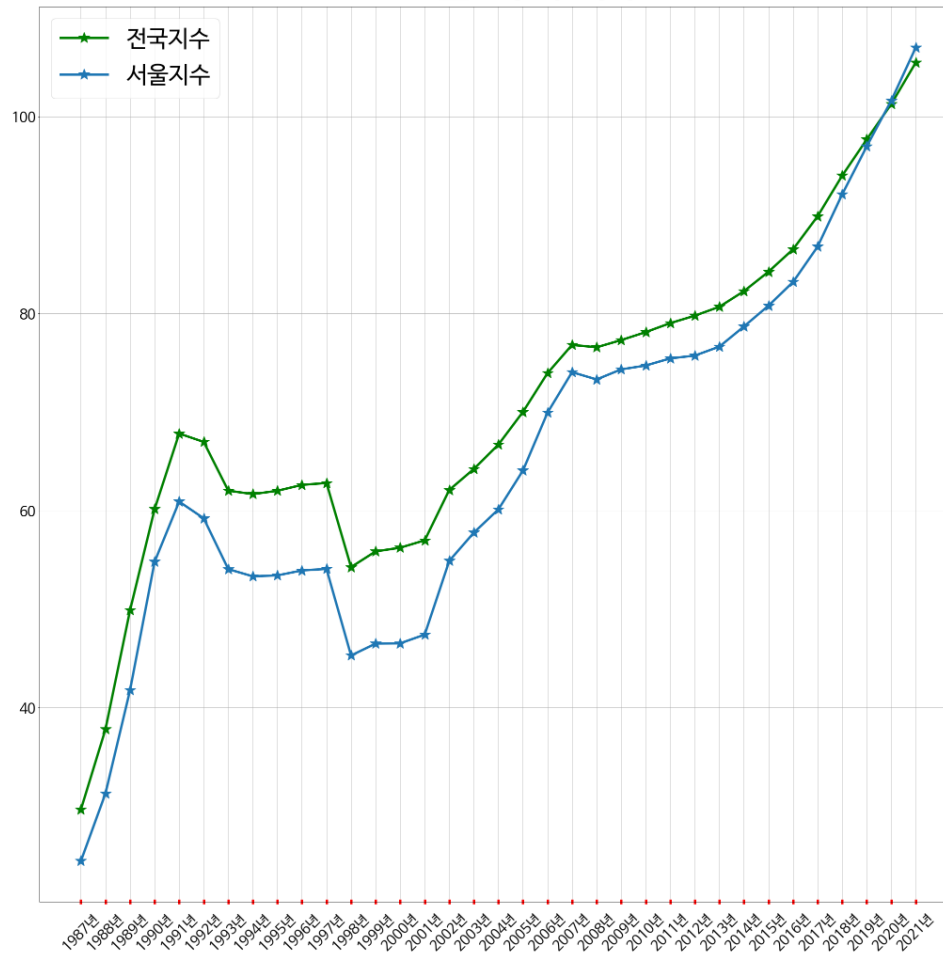
```
plt.plot(year, all_estate_index , marker='*',linewidth=5, markersize=25, color='green', label='{}지수'.format(str(place_list[1])))
plt.plot(year, np_dict[input_name] , marker='*',linewidth=5, markersize=25, label='{}지수'.format(input_name))
plt.legend(loc=2, fontsize=50) #범례
plt.title('연도별 지가지수' , fontsize=100, pad=20)
```

2개의 꺾은선 그래프를 나타내는데 첫 번째 줄 코드는 전국 지가지수이며, 두 번째 줄 코드는 위에서 딕셔너리 변수에서 입력 받은 지역명을 검색해서 그래프로 나타내는 코드입니다.



# 그래프 출력 화면

연도별 지가지수



# 텍스트 추가

```
plt.annotate('역사상 가장 큰 하락', ①  
            xy=(year[11], np_dict[input_name][11]-1), ②  
            xytext=(year[11], np_dict[input_name][11]-6 ), ③  
            arrowprops=dict(facecolor='black'), ④  
            size=20) ⑤
```

- ① 텍스트
- ② 화살표 방향
- ③ 텍스트 위치
- ④ 화살표 특성 (색상)
- ⑤ 폰트 사이즈

# 그래프 출력 화면

연도별 지가지수

