

```

297 def drawKorea(self, targetData, blockedMap, cmapname):
298     gamma = 0.75
299
300     whitelabelmin = (max(blockedMap[targetData]) -
301                     min(blockedMap[targetData])) * 0.25 + \
302                     min(blockedMap[targetData])
303
304     datalabel = targetData
305
306     vmin = min(blockedMap[targetData])
307     vmax = max(blockedMap[targetData])
308
309     mapdata = blockedMap.pivot_table(index='y', columns='x', values=targetData)
310     masked_mapdata = np.ma.masked_where(np.isnan(mapdata), mapdata)
311
312     plt.figure(figsize=(9, 11))
313     plt.pcolor(masked_mapdata, vmin=vmin, vmax=vmax, cmap=cmapname,
314               edgecolor='#aaaaaa', linewidth=0.5)
315
316     # 지역 이름 표시
317     for idx, row in blockedMap.iterrows():
318         # 괄호치는 구 이름이 겹치는 경우가 많아서 시판위 이름도 같이 표시한다.
319         # (중구, 서구)
320         if len(row['ID'].split()) == 2:
321             dispname = '{}\n{}'.format(row['ID'].split()[0], row['ID'].split()[1])
322         elif row['ID'][:2] == '그성':
323             dispname = '그성'
324         else:
325             dispname = row['ID']
326
327         # 서대문구, 서귀포시 같이 이름이 3자 이상인 경우에 작은 글자로 표시한다.
328         if len(dispname.splitlines())[-1] >= 3:
329             fontsize, linespacing = 10.0, 1.1
330         else:
331             fontsize, linespacing = 11, 1.
332
333         annocolor = 'white' if row[targetData] > whitelabelmin else 'black'
334         plt.annotate(dispname, (row['x'] + 0.5, row['y'] + 0.5), weight='bold',
335                     fontsize=fontsize, ha='center', va='center', color=annocolor,
336                     linespacing=linespacing)
337     self.draw_border_lines(datalabel)

```