

```
In [1]: #Використовуючи файл `population_prospects.csv`, створіть карти, котрі відповідають на наступні питання:

#- В яких країнах світу населення у 2100 році буде більшим, ніж було у 2000 році, а в яких меншим?

#- На скільки відсотків збільшиться або зменшиться населення кожної країни у 2100 році у порівнянні з 2000 роком?

#- Яким буде населення країн світу у 2100 році?

#У наборі даних ви знайдете декілька прогнозів щодо зростання / скорочення населення країн світу (змінна `variant`)
#Для фактичних історичних даних (2000 рік) ви можете використовувати значення `Estimates`, для прогнозних
#даних (2100 рік) ви можете використовувати значення `Medium variant`.

#Ви можете використовувати файл `world-countries.json` або будь-який інший файл з геоданими, котрий вам підходить
#для цієї задачі. Ви можете використовувати будь-який доступний вам інструмент візуалізації даних.

#Ви маєте також у невеличкому супровідному тексті обґрунтувати вибір типу карти / типу кольорової шкали, пояснити
#його сильні та слабкі сторони для поставленої задачі.
```

```
In [2]: import pandas as pd
import altair as alt

import geopandas as gpd

alt.data_transformers.disable_max_rows()

df = pd.read_csv("population_prospects.csv")
cor = gpd.read_file('world-countries.json', encoding="utf-8")
```

```
In [3]: # select 2000 eastimates and 2100 medium variant

population_2000 = df[df["variant"] == "Estimates" ]
population_2000 = population_2000[population_2000["year"] == 2000]

population_2100 = df[df["variant"] == "Medium variant" ]
population_2100 = population_2100[population_2100["year"] == 2100]

population_2000 = population_2000.drop(columns =['variant', 'parent_code', 'country_code', 'year'])
population_2100 = population_2100.drop(columns =['variant', 'parent_code', 'country_code', 'year'])

new_df = population_2000.merge(population_2100, on ="country")

new_df = new_df.rename(columns={"population_x": "year_2000"})
new_df = new_df.rename(columns={"population_y": "year_2100"})

print(new_df)
```

	country	year_2000	year_2100
0	Burundi	6378.871	50904.072
1	Comoros	542.358	2186.509
2	Djibouti	717.577	1331.656
3	Eritrea	2292.413	9061.777
4	Ethiopia	66224.809	294392.903
..
230	Bermuda	65.017	36.933
231	Canada	30588.379	56953.224
232	Greenland	56.177	40.524
233	Saint Pierre and Miquelon	6.298	3.984
234	United States of America	281710.914	433853.891

[235 rows x 3 columns]

```
In [5]: cor['name'] = cor['name'].replace("The Bahamas",'Bahamas')
cor['name'] = cor['name'].replace("Bolivia",'Bolivia (Plurinational State of)')
cor['name'] = cor['name'].replace("Brunei",'Brunei Darussalam')
cor['name'] = cor['name'].replace("Republic of the Congo",'Congo')
cor['name'] = cor['name'].replace("Northern Cyprus",'Cyprus')
cor['name'] = cor['name'].replace("Czech Republic",'Czechia')
cor['name'] = cor['name'].replace("Falkland Islands",'Falkland Islands (Malvinas)')
cor['name'] = cor['name'].replace("Guinea Bissau",'Guinea-Bissau')
cor['name'] = cor['name'].replace("Iran",'Iran (Islamic Republic of)')
cor['name'] = cor['name'].replace("South Korea",'Republic of Korea')
cor['name'] = cor['name'].replace("Moldova",'Republic of Moldova')
cor['name'] = cor['name'].replace("Macedonia",'North Macedonia')
cor['name'] = cor['name'].replace("North Korea",'Republic of Korea')
cor['name'] = cor['name'].replace("Russia",'Russian Federation')
cor['name'] = cor['name'].replace("Republic of Serbia",'Serbia')
cor['name'] = cor['name'].replace("Syria",'Syrian Arab Republic')
cor['name'] = cor['name'].replace("East Timor",'Timor-Leste')
cor['name'] = cor['name'].replace("Taiwan",'China, Taiwan Province of China')
cor['name'] = cor['name'].replace("Venezuela",'Venezuela (Bolivarian Republic of)')
cor['name'] = cor['name'].replace("Vietnam",'Viet Nam')
```

```
In [6]: new_df.rename(columns={"country" : "name"}, inplace=True)

new_df = cor.merge(new_df)

print(new_df)
```

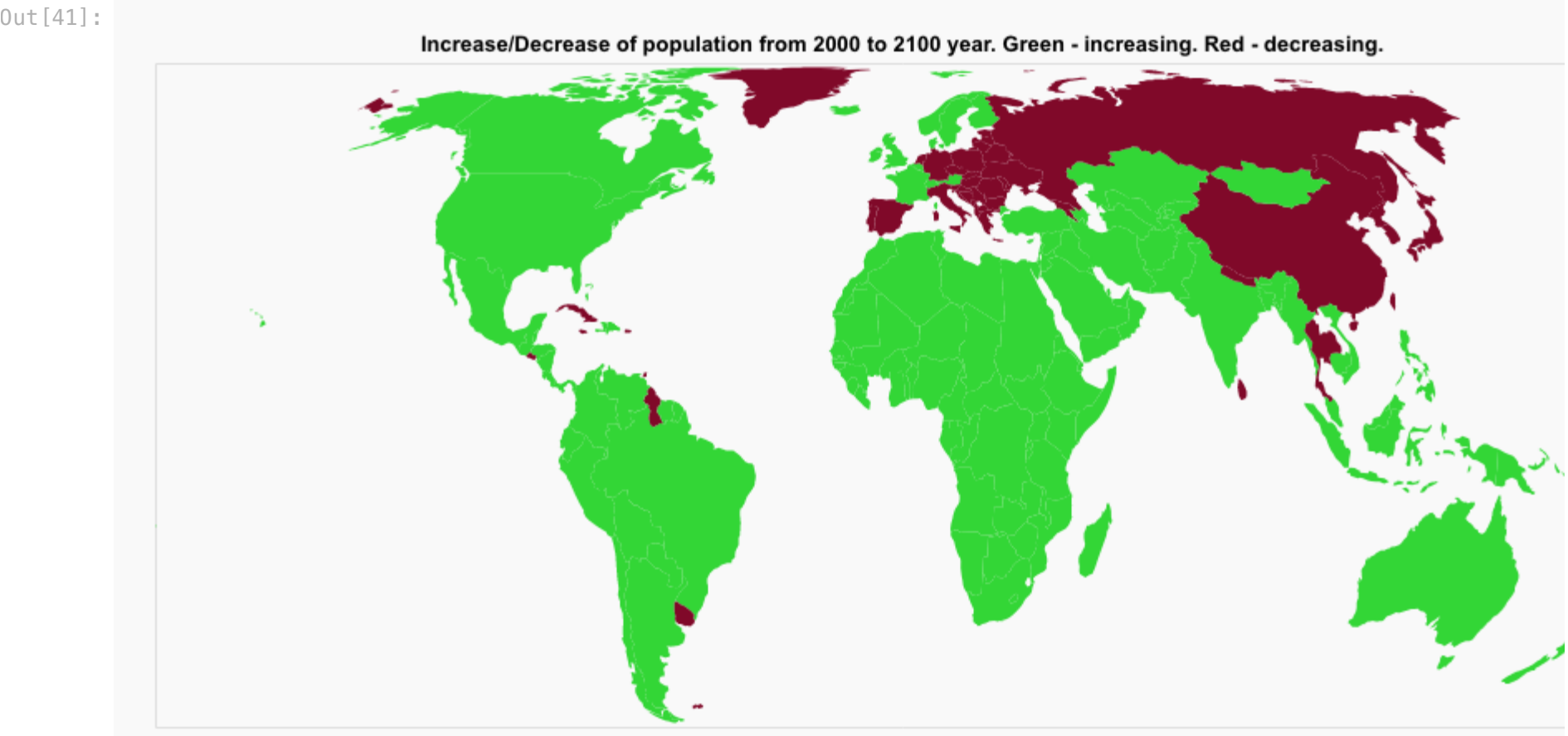
	id	name	\
0	AFG	Afghanistan	
1	AGO	Angola	
2	ALB	Albania	
3	ARE	United Arab Emirates	
4	ARG	Argentina	
..	
164	VUT	Vanuatu	
165	YEM	Yemen	
166	ZAF	South Africa	
167	ZMB	Zambia	
168	ZWE	Zimbabwe	

	geometry	year_2000	year_2100
0	POLYGON ((61.21082 35.65007, 62.23065 35.27066...	20779.957	74937.961
1	MULTIPOLYGON (((16.32653 -5.87747, 16.57318 -6...	16395.477	188283.132
2	POLYGON ((20.59025 41.85540, 20.46317 41.51509...	3129.246	1088.338
3	POLYGON ((51.57952 24.24550, 51.75744 24.29407...	3134.067	12909.869
4	MULTIPOLYGON (((-65.50000 -55.20000, -66.45000...	36870.796	56802.493
..
164	MULTIPOLYGON (((167.84488 -16.46633, 167.51518...	184.964	968.068
165	POLYGON ((53.10857 16.65105, 52.38521 16.38241...	17409.071	53171.320
166	POLYGON ((31.52100 -29.25739, 31.32556 -29.401...	44967.713	79190.832
167	POLYGON ((32.75937 -9.23060, 33.23139 -9.67672...	10415.942	81546.198
168	POLYGON ((31.19141 -22.25151, 30.65986 -22.151...	11881.482	30965.421

[169 rows x 5 columns]

```
In [41]: #- В яких країнах світу населення у 2100 році буде більшим, ніж було у 2000 році, а в яких меншим?
alt.Chart(new_df).project(type = 'equalEarth').mark_geoshape().encode(
    tooltip = alt.Tooltip('name:N'),
    color = alt.condition(
        alt.datum.year_2100 > alt.datum.year_2000,
        alt.value('#33d636'),
        alt.value('#800929'), bind='legend'
    )
).properties(width = 900, height = 400, background = '#F9F9F9', padding = 25,
    title="Increase/Decrease of population from 2000 to 2100 year.\nGreen - increasing. Red - decreasing")

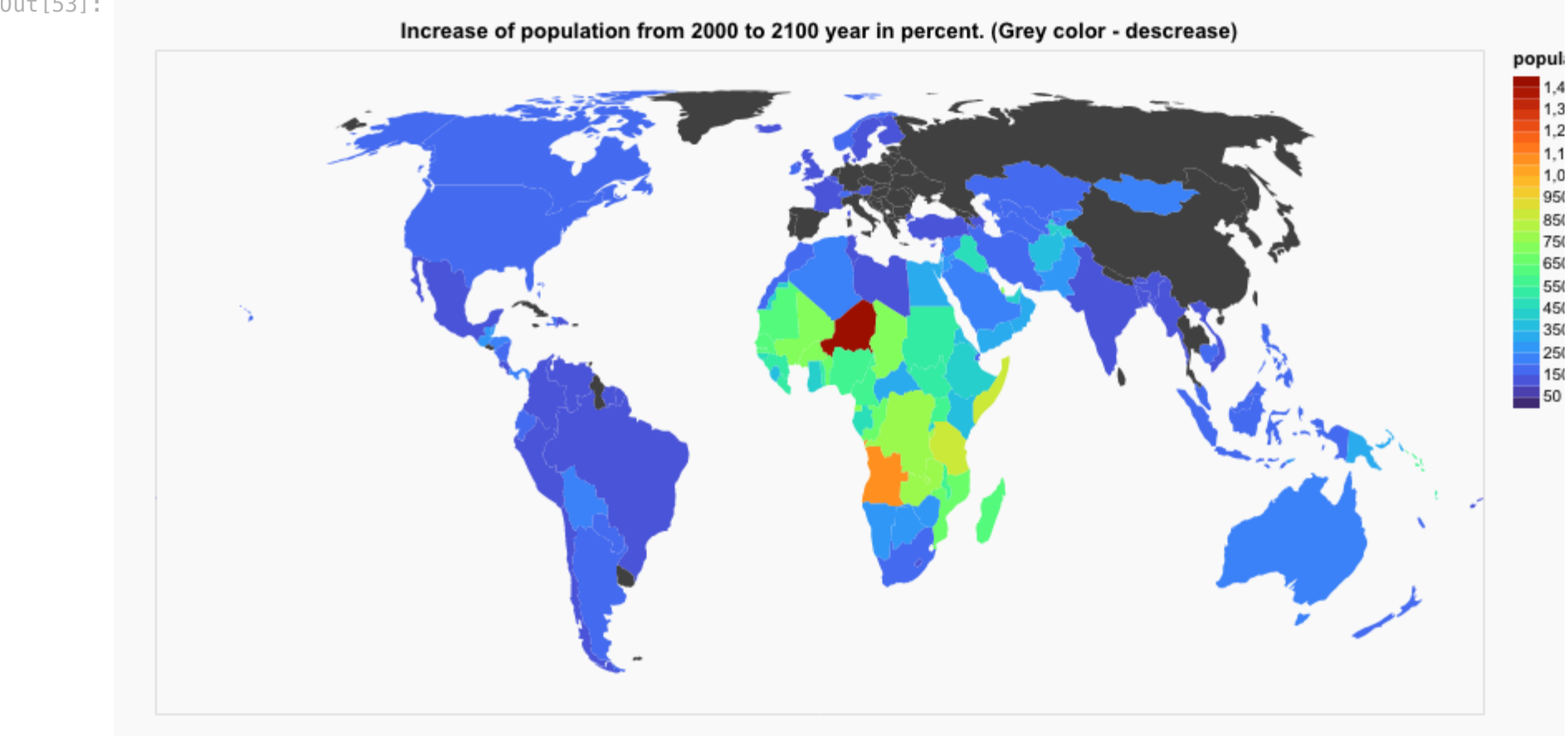
# Мапа з кольорами дозволяє показати дані по свім країнам,
# темночервоний інтуїтивно сприймається як занепад, а зелений як ріск та нове життя.
# Цей спосіб не дуже ок як для ПДФ, який ви читаєте адже tooltip не спрацює. Країни розрізнити буде важко.
# Але чи треба ?
# Якщо треба, то краще це зробити в ікселі адже підписувати малу мапу - погана ідея.
```



```
In [53]: #- На скільки відсотків збільшиться або зменшиться населення кожної країни у 2100 році у порівнянні з 2000 роком?

alt.Chart(new_df).project(type = 'equalEarth').transform_calculate(
    p="datum.year_2100 / datum.year_2000 * 100"
).mark_geoshape().encode(
    tooltip = alt.Tooltip(['name:N']),
    color = alt.condition(
        alt.datum.p < 100 ,
        alt.value('#404040'),
        alt.Color("p:Q", title = 'population increase %', scale=alt.Scale(scheme='turbo'), bin = alt.Bin(step = 5))
    ).properties(width = 800, height = 400, background = '#F9F9F9', padding = 25,
        title="Increase of population from 2000 to 2100 year in percent. (Grey color - decrease)")

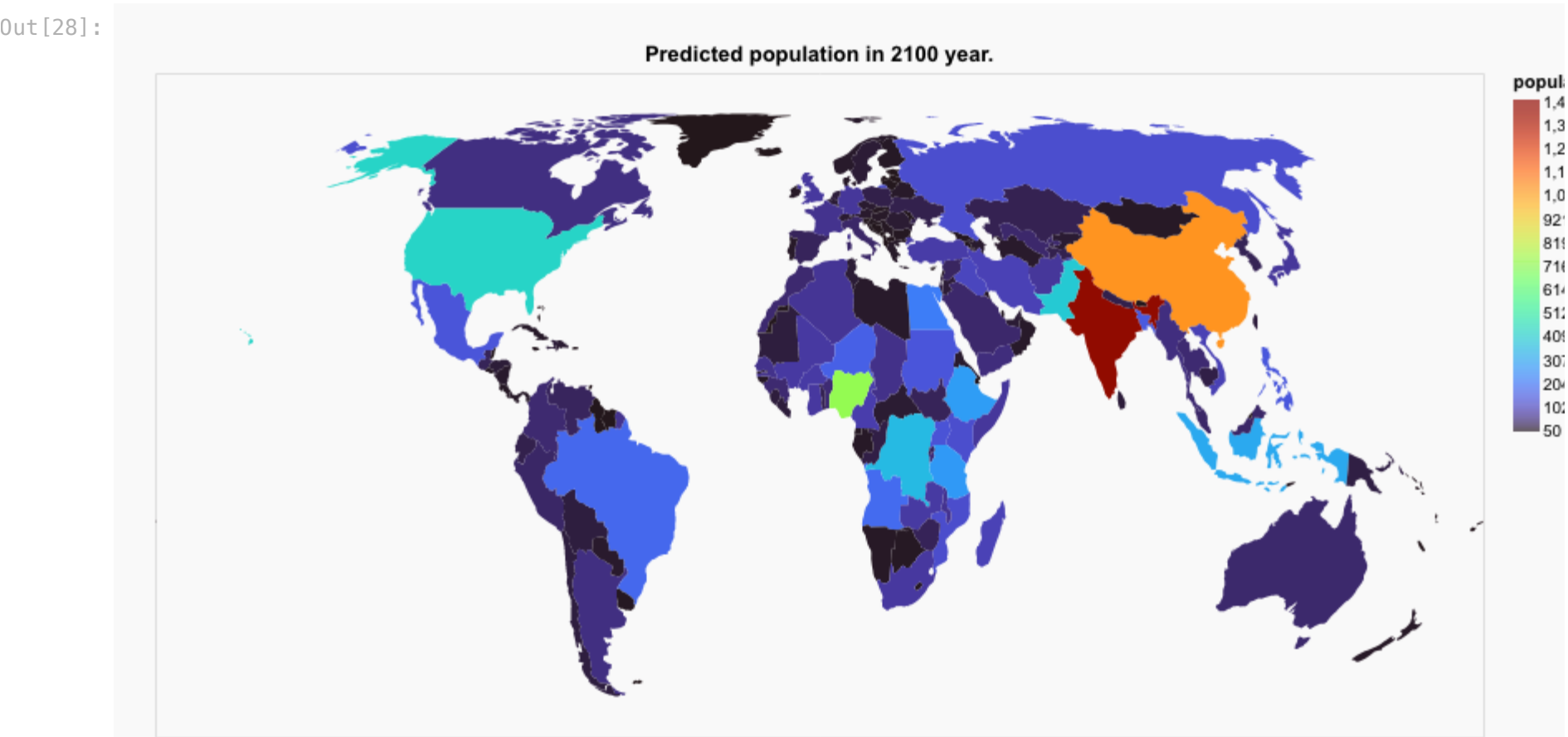
# В мене нажалі не вийшло зробити зменшення населення так як альаір не сприймав мої відемні значення.
# Через те я відемні значення зобразила як темносирій - теж колір смерті + від відрізняється від turbo бібліотеки
# Темні насичені Кольори досить добре індекують зміни.
# Можна було розділити на дві карти - карти країн, де популяція зроста та карта де популяція зростає.
# Але в країнах, де є зменшення , воно зменшується не дуже сильно, не на 99%, тому окрему мапу я робити не стала
```



```
In [28]: ##- Яким буде населення країн світу у 2100 році?

alt.Chart(new_df).project(type = 'equalEarth').mark_geoshape().encode(
    tooltip = alt.Tooltip(['name:N', 'year_2100:Q']),
    color = alt.condition(
        alt.Color("year_2100:Q", title = 'population', scale=alt.Scale(scheme='turbo'), bin = alt.Bin(step = 5))
    ).properties(width = 800, height = 400, background = '#F9F9F9', padding = 25,
        title="Predicted population in 2100 year.")

# Кольорова схема turbo досить добре передає розподіл по майбутній популяції.
# В інших схемах були проблеми, так як було тільки два кольора, то аутлаери займали 90% розподілу, а інші
# країни були зовсім непомітні (світло-білі). На темному контраст видно краще.
# Також додала padding щоб трохи нормалізувати розподіл кольорів.
# Можна зробити Tooltip з населенням може допомогти орієнтуватись.
# Альтернатива також таблицка в ікселі.
```



```
In [ ]:
```