

Escuela Politecnica Nacional

Nombre: Stiv Quishpe

Link del repositorio

<https://github.com/stiv001/FormattersEnPython.git>

Codigo original sin aplicacion de formatters

```
import numpy as np

import seaborn as sns
import matplotlib.pyplot as plt

import pandas as pd

años = np.arange(2000, 2021)
poblacion_ecuador = np.array(
    [50, 52, 54, 56, 58, 60, 63, 65, 68, 70, 73, 75, 78, 80, 83, 85, 88, 90, 93, 95, 98]
)
poblacion_argentina = np.array(
    [30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70]
)
poblacion_colombia = np.array(
    [20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60]
)

datos = {
    "Año": años,
    "Población Ecuador": poblacion_ecuador,
    "Población Argentina": poblacion_argentina,
```

```

    "Población Colombia": poblacion_colombia,
}
df = pd.DataFrame(datos)

sns.set(style="whitegrid")

plt.figure(figsize=(12, 6))

sns.lineplot(x="Año", y="Población Ecuador", data=df, label="Población Ecuador")
sns.lineplot(x="Año", y="Población Argentina", data=df, label="Población Argentina")

sns.lineplot(x="Año", y="Población Colombia", data=df, label="Población Colombia")

plt.title("Crecimiento de la Población en América Latina (2000-2020)")
plt.xlabel("Año")
plt.ylabel("Población (en miles)")

plt.legend()

plt.show()

```

BLACK

Principales diferencias con el código original:

1. **Longitud de línea:** Black se asegura de que ninguna línea tenga más de 88 caracteres. Por eso, las listas largas se dividen en varias líneas.
2. **Indentación:** Black usa una indentación consistente de 4 espacios.
3. **Espaciado:** Black añade y elimina espacios en blanco según sea necesario para mejorar la legibilidad y cumplir con las convenciones de estilo.
4. **Comillas:** Black mantiene un estilo de comillas consistente, aunque en este caso no hubo cambios en las comillas.

```

import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

años = np.arange(2000, 2021)

```

```
poblacion_ecuador = np.array(  
    [  
        50,  
        52,  
        54,  
        56,  
        58,  
        60,  
        63,  
        65,  
        68,  
        70,  
        73,  
        75,  
        78,  
        80,  
        83,  
        85,  
        88,  
        90,  
        93,  
        95,  
        98,  
    ]  
)  
poblacion_argentina = np.array(  
    [  
        30,  
        32,  
        34,  
        36,  
        38,  
        40,  
        42,  
        44,  
        46,  
        48,  
        50,  
        52,  
        54,  
        56,  
        58,  
    ]  
)
```

```

        60,
        62,
        64,
        66,
        68,
        70,
    ]
)
poblacion_colombia = np.array(
    [
        20,
        22,
        24,
        26,
        28,
        30,
        32,
        34,
        36,
        38,
        40,
        42,
        44,
        46,
        48,
        50,
        52,
        54,
        56,
        58,
        60,
    ]
)

datos = {
    "Año": años,
    "Población Ecuador": poblacion_ecuador,
    "Población Argentina": poblacion_argentina,
    "Población Colombia": poblacion_colombia,
}
df = pd.DataFrame(datos)

```

```

sns.set(style="whitegrid")

plt.figure(figsize=(12, 6))

sns.lineplot(x="Año", y="Población Ecuador", data=df, label="Población Ecuador")
sns.lineplot(x="Año", y="Población Argentina", data=df, label="Población Argentina")
sns.lineplot(x="Año", y="Población Colombia", data=df, label="Población Colombia")

plt.title("Crecimiento de la Población en América Latina (2000-2020)")
plt.xlabel("Año")
plt.ylabel("Población (en miles)")

plt.legend()

plt.show()

```

autopep8

Principales diferencias entre el código original y el formateado con autopep8:

1. **Longitud de línea:** autopep8 intenta mantener las líneas dentro de un límite de 79 caracteres, pero no siempre divide las listas largas en varias líneas como lo hace Black.
2. **Espaciado:** autopep8 ajusta los espacios en blanco alrededor de los operadores y después de las comas para cumplir con PEP 8.
3. **Indentación:** autopep8 asegura que la indentación sea consistente, utilizando 4 espacios por nivel de indentación.
4. **Comillas:** autopep8 no cambia el estilo de las comillas a menos que sea necesario para cumplir con PEP 8.

```

import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

años = np.arange(2000, 2021)
poblacion_ecuador = np.array(
    [50, 52, 54, 56, 58, 60, 63, 65, 68, 70, 73, 75, 78, 80, 83, 85, 88, 90, 93, 95, 98]
)
poblacion_argentina = np.array(
    [30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70]
)

```

```

)
poblacion_colombia = np.array(
    [20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60]
)

datos = {
    "Año": años,
    "Población Ecuador": poblacion_ecuador,
    "Población Argentina": poblacion_argentina,
    "Población Colombia": poblacion_colombia,
}
df = pd.DataFrame(datos)

sns.set(style="whitegrid")

plt.figure(figsize=(12, 6))

sns.lineplot(x="Año", y="Población Ecuador", data=df, label="Población Ecuador")
sns.lineplot(x="Año", y="Población Argentina", data=df, label="Población Argentina")
sns.lineplot(x="Año", y="Población Colombia", data=df, label="Población Colombia")

plt.title("Crecimiento de la Población en América Latina (2000-2020)")
plt.xlabel("Año")
plt.ylabel("Población (en miles)")

plt.legend()

plt.show()

```

yapf

Principales diferencias con el código original:

1. **Longitud de línea:** YAPF intenta mantener las líneas dentro de un límite de 80 caracteres, pero se puede configurar para diferentes estilos de longitud de línea.
2. **Espaciado:** YAPF ajusta los espacios en blanco alrededor de los operadores y después de las comas para mejorar la legibilidad.
3. **Indentación:** YAPF asegura que la indentación sea consistente, utilizando 4 espacios por nivel de indentación.

4. **Listas y diccionarios:** YAPF puede ajustar la forma en que se formatean las listas y diccionarios, manteniéndolos en una sola línea si es posible o dividiéndolos en varias líneas para mejorar la legibilidad.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

años = np.arange(2000, 2021)
poblacion_ecuador = np.array([
    50, 52, 54, 56, 58, 60, 63, 65, 68, 70, 73, 75, 78, 80, 83, 85, 88, 90, 93, 95, 98
])
poblacion_argentina = np.array([
    30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70
])
poblacion_colombia = np.array([
    20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60
])

datos = {
    "Año": años,
    "Población Ecuador": poblacion_ecuador,
    "Población Argentina": poblacion_argentina,
    "Población Colombia": poblacion_colombia,
}
df = pd.DataFrame(datos)

sns.set(style="whitegrid")

plt.figure(figsize=(12, 6))

sns.lineplot(x="Año", y="Población Ecuador", data=df, label="Población Ecuador")
sns.lineplot(x="Año", y="Población Argentina", data=df, label="Población Argentina")
sns.lineplot(x="Año", y="Población Colombia", data=df, label="Población Colombia")

plt.title("Crecimiento de la Población en América Latina (2000-2020)")
plt.xlabel("Año")
plt.ylabel("Población (en miles)")

plt.legend()

plt.show()
```

isort

Principales diferencias entre el código original y el formateado con isort:

1. **Orden de las importaciones:** isort organiza las importaciones en un orden lógico y consistente. Generalmente, agrupa primero las importaciones estándar, luego las de terceros y finalmente las locales.
2. **Espaciado entre grupos de importaciones:** isort añade líneas en blanco entre diferentes grupos de importaciones para mejorar la legibilidad.

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
```

```
años = np.arange(2000, 2021)
poblacion_ecuador = np.array(
    [
        50,
        52,
        54,
        56,
        58,
        60,
        63,
        65,
        68,
        70,
        73,
        75,
        78,
        80,
        83,
        85,
        88,
        90,
        93,
        95,
        98,
```



```

    ]
)
poblacion_argentina = np.array(
    [
        30,
        32,
        34,
        36,
        38,
        40,
        42,
        44,
        46,
        48,
        50,
        52,
        54,
        56,
        58,
        60,
        62,
        64,
        66,
        68,
        70,
    ]
)
poblacion_colombia = np.array(
    [
        20,
        22,
        24,
        26,
        28,
        30,
        32,
        34,
        36,
        38,
        40,
        42,
        44,
    ]
)

```

```

        46,
        48,
        50,
        52,
        54,
        56,
        58,
        60,
    ]
)

datos = {
    "Año": años,
    "Población Ecuador": poblacion_ecuador,
    "Población Argentina": poblacion_argentina,
    "Población Colombia": poblacion_colombia,
}
df = pd.DataFrame(datos)

sns.set(style="whitegrid")

plt.figure(figsize=(12, 6))

sns.lineplot(x="Año", y="Población Ecuador", data=df, label="Población Ecuador")
sns.lineplot(x="Año", y="Población Argentina", data=df, label="Población Argentina")
sns.lineplot(x="Año", y="Población Colombia", data=df, label="Población Colombia")

plt.title("Crecimiento de la Población en América Latina (2000-2020)")
plt.xlabel("Año")
plt.ylabel("Población (en miles)")

plt.legend()

plt.show()

```