**import seaborn as sns**

This imports the **Seaborn** library and gives it a short name sns.  
Seaborn is built on top of Matplotlib, but gives **beautiful and easy-to-use charts**.

**import pandas as pd**

This imports the **Pandas** library and gives it the alias pd.  
We use **Pandas DataFrame** to organize data into a neat table format — rows and columns.

**import matplotlib.pyplot as py**

This imports the plotting part of **Matplotlib** and gives it the name py.  
We still use Matplotlib to show the chart and manage figure size, titles, etc.

Seaborn is built **on top of Matplotlib**.  
That means Seaborn handles the **styling and high-level plotting** (like making a heatmap or boxplot easily), but underneath, it still uses **Matplotlib to actually draw the charts**.

So, Seaborn = “beautiful wrapper” around Matplotlib.  
That’s why we often still import matplotlib.pyplot — for fine control, customizations, or saving plots

**hours = list(range(1, 11))**

This creates a list of numbers from 1 to 10 — representing **hours worked**.

e.g. : [1, 2, 3, ..., 10]

**earnings = [200 \* h for h in hours]**

This uses a list comprehension to calculate the **money earned** for each hour.

It assumes ₹200 per hour.

Example:

* 1 hour ₹200
* 2 hours ₹400
* ...
* 10 hours ₹2000

**df = pd.DataFrame({...})**

Creates a **DataFrame** (like an Excel table) using the hours and earnings lists.  
This makes it easy for Seaborn to read and plot the data.

Looks like:

| **Hours Worked** | **Money Earned (₹)** |
| --- | --- |
| 1 | 200 |
| 2 | 400 |
| ... | ... |
| 10 | 2000 |

**sns.set(style='whitegrid')**

This sets the **style** of the plot.  
whitegrid adds **grid lines** on a white background — makes it clean and easy to read.

**py.figure(figsize=(8, 5))**

Sets the **size of the figure** (width=8 inches, height=5 inches).  
This gives your chart some space so labels and points don’t feel cramped.

**sns.lineplot(...)**

This is the **actual plot** using Seaborn!  
It draws a **line graph** showing how money increases as hours increase.

**Inside:**

* data=df: Use the DataFrame df
* x='Hours Worked': Use column Hours Worked for x-axis
* y='Money Earned (₹)': Use column Money Earned (₹) for y-axis
* marker='o': Adds circular dots to each data point
* color='teal': Teal is the color of the line
* linewidth=2: Line thickness is 2 units

**Very Imp note:**

**Seaborn and Matplotlib are built to work together because:**

***Seaborn is built on top of Matplotlib*.**

**That means:**

**When you call “sns.lineplot(...)”, under the hood it's actually creating a Matplotlib figure and axes.**

**Any Matplotlib commands you run afterward (like py.title() or py.xlabel()) will apply to the current active Matplotlib plot, which is the one Seaborn created.**

**py.title(...)**

Sets the **title** of the chart:

'Earnings Based on Hours Worked'

**py.xlabel(...)**

Sets the **label for the x-axis**: 'Hours Worked'

**py.ylabel(...)**

Sets the **label for the y-axis**: 'Money Earned (₹)'

**py.tight\_layout()**

This **automatically adjusts** spacing between elements so that **nothing overlaps or gets cut off**.

**py.show()**

This **displays the chart** on the screen.  
Always the last step in a plot!

Why doesn't sns (Seaborn) have a .show() function?

Ans:-

Because Seaborn **doesn't manage the figure window** — it just **adds plots and styles** to a figure that's actually managed by **Matplotlib**.

In simple terms:  
**Matplotlib builds and shows the house.**  
**Seaborn decorates and furnishes it beautifully.**

When you run:

sns.lineplot(...)

*Internally, Seaborn is calling Matplotlib functions like*:

matplotlib.pyplot.plot(...)

But Seaborn **does not**:

* Create the window to show the plot
* Handle rendering the figure on screen

That job is still done by:

import matplotlib.pyplot as py

py.show()

**Why sns (seaborn) does not have show()?**

Because Seaborn is **only a high-level interface**, it **doesn't try to replace Matplotlib** — just **enhance it**.

If Seaborn tried to include show(), it would:

* Duplicate what Matplotlib already does well
* Confuse users who are already used to py.show()

Summary:-

Seaborn makes charts prettier and easier, but it relies on Matplotlib to actually **display** them. That’s why we still need to use plt.show() from Matplotlib.