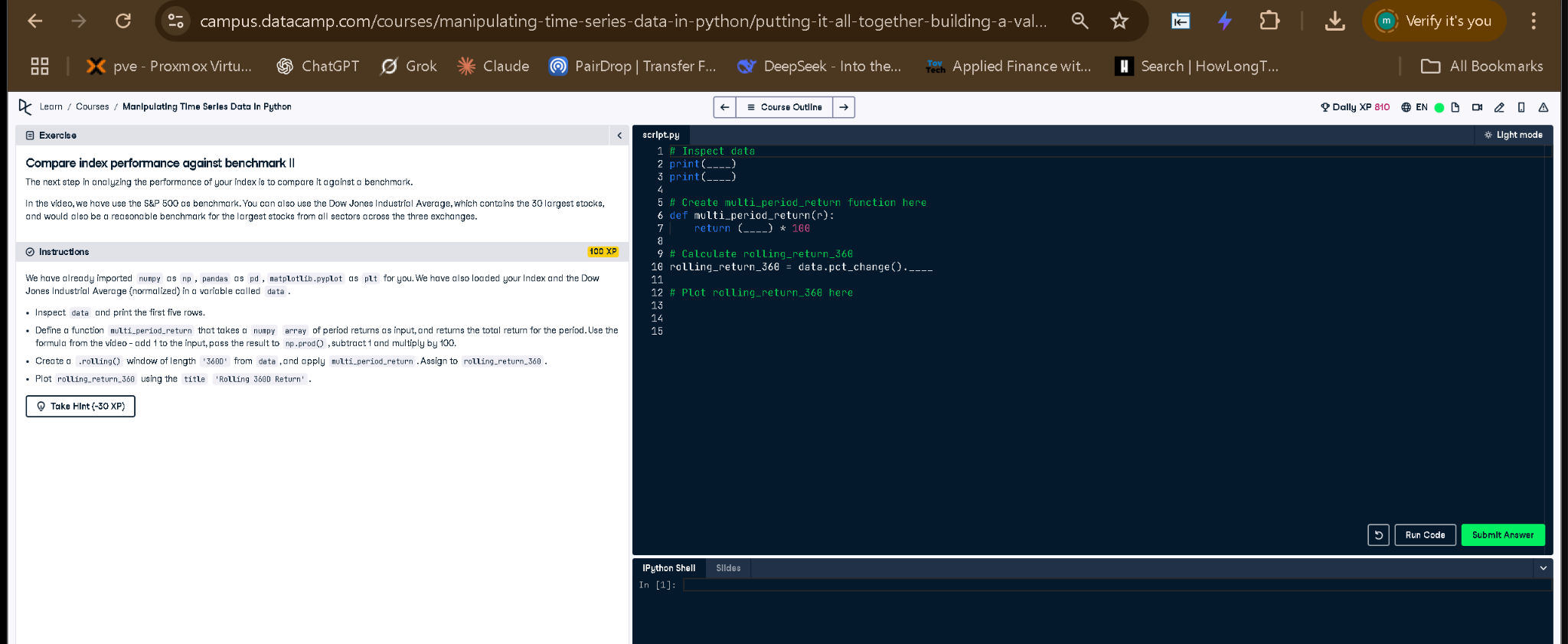
# Compare Index Performance Against Benchmark



## ✅ Full Correct Answer (Time-Based Window):

# Inspect data  
print(data.head())  
print(data.columns)  
  
# Create multi\_period\_return function here  
def multi\_period\_return(r):  
 return (np.prod(r + 1) - 1) \* 100  
  
# Calculate rolling\_return\_360 using a time-based window  
rolling\_return\_360 = data.pct\_change().rolling('360D').apply(multi\_period\_return)  
  
# Plot rolling\_return\_360 here  
rolling\_return\_360.plot(title='Rolling 360D Return')  
plt.show()

## 🧾 Explanation (Simple Words):

This version uses a time-based rolling window ('360D') instead of a fixed row count. This handles gaps in the time series more accurately. We calculate daily percentage changes, then compute rolling returns over 360 days using the custom function, and finally plot the result to see trends over time.