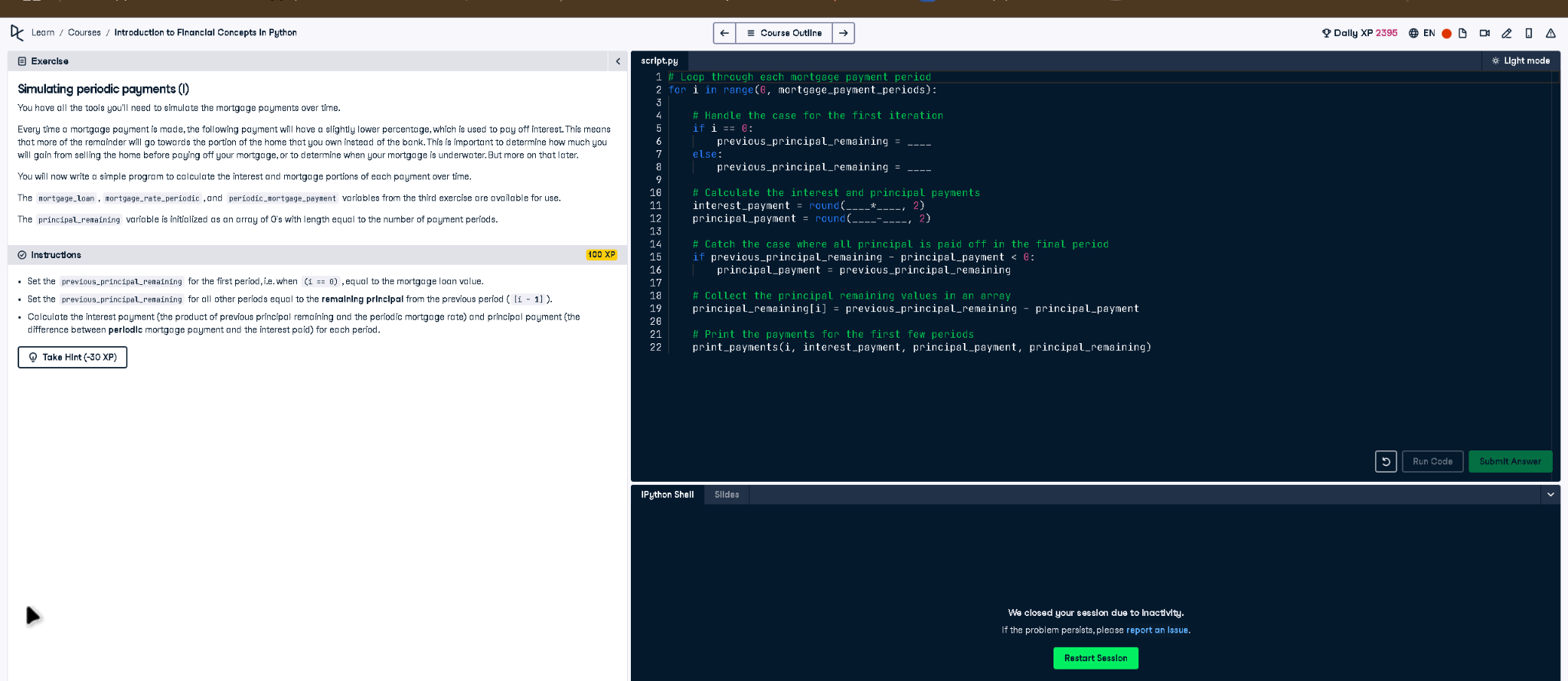
Simulating Periodic Payments Completion - Full Answer



# Full Code Answer:

# Loop through each mortgage payment period  
for i in range(0, mortgage\_payment\_periods):  
  
 # Handle the case for the first iteration  
 if i == 0:  
 previous\_principal\_remaining = mortgage\_loan  
 else:  
 previous\_principal\_remaining = principal\_remaining[i - 1]  
  
 # Calculate the interest and principal payments  
 interest\_payment = round(previous\_principal\_remaining \* mortgage\_rate\_periodic, 2)  
 principal\_payment = round(periodic\_mortgage\_payment - interest\_payment, 2)  
  
 # Catch the case where all principal is paid off in the final period  
 if previous\_principal\_remaining - principal\_payment < 0:  
 principal\_payment = previous\_principal\_remaining  
  
 # Collect the principal remaining values in an array  
 principal\_remaining[i] = previous\_principal\_remaining - principal\_payment  
  
 # Print the payments for the first few periods  
 print\_payments(i, interest\_payment, principal\_payment, principal\_remaining)

# Question:

How do you fill the missing values to simulate interest and principal payments for periodic mortgage payments?

# 20-word Explanation (Question):

To simulate periodic mortgage payments, track remaining principal, calculate interest from it, and determine the principal portion each period.

# Answer:

Set first iteration's principal to mortgage\_loan, subsequent to previous principal, then compute interest and principal payments accordingly.

# 20-word Explanation (Answer):

We update previous\_principal\_remaining each iteration, calculate interest as principal \* rate, then subtract to find principal payment each period.