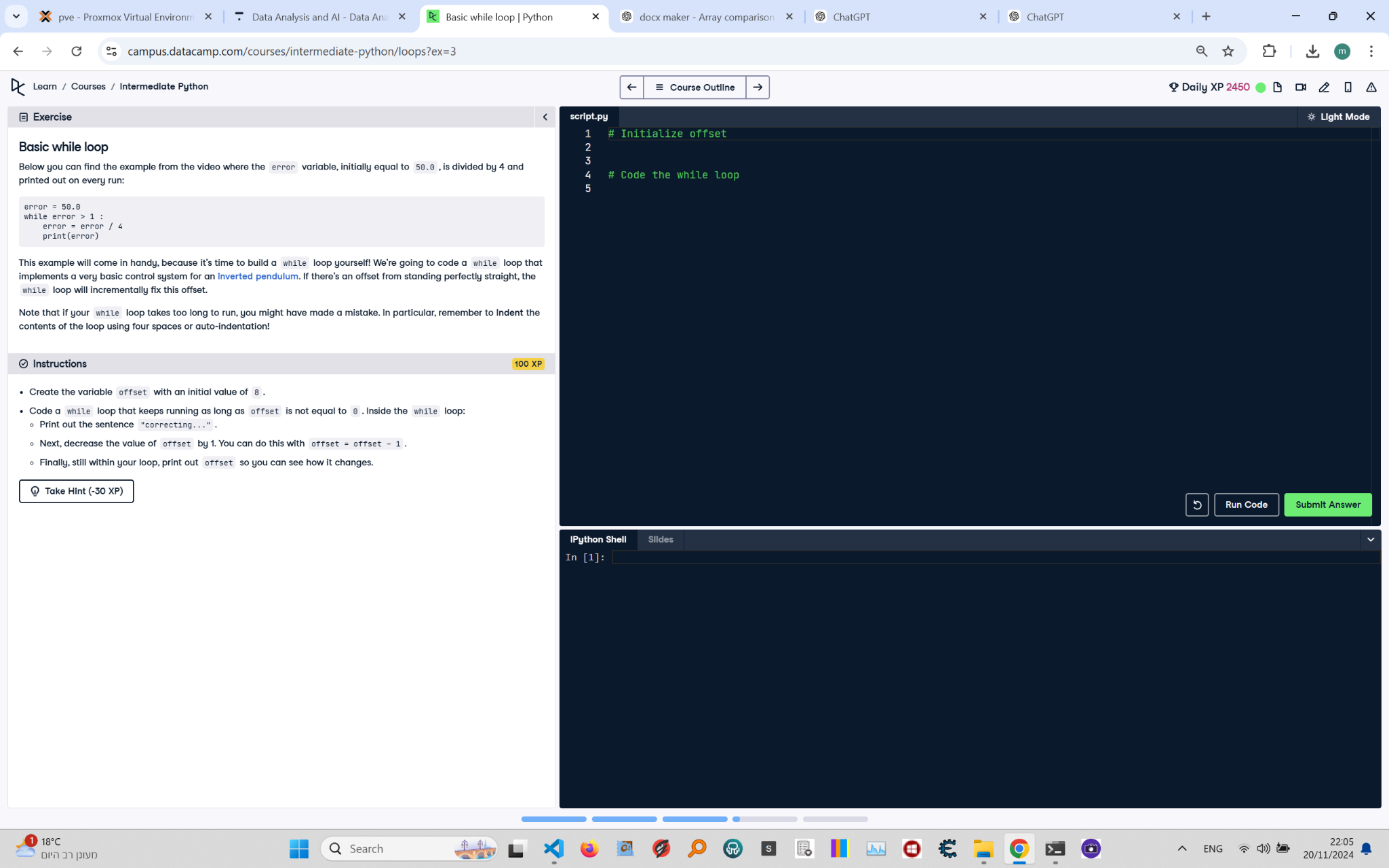
# Basic While Loop in Python



\*\*Question:\*\*

Below you can find an example from the video where the `error` variable, initially equal to 50.0, is divided by 4 and printed out on every run.  
  
```python  
error = 50.0  
while error > 1:  
 error = error / 4  
 print(error)  
```  
  
This example will come in handy, because it's time to build a `while` loop yourself! We're going to code a `while` loop that implements a very basic control system for an inverted pendulum. If there's an offset from standing perfectly straight, the loop will incrementally fix this offset.  
  
\*\*Instructions:\*\*  
1. Create the variable `offset` with an initial value of `8`.  
2. Code a `while` loop that keeps running as long as `offset` is not equal to `0`. Inside the `while` loop:  
 - Print out the sentence `correcting...`.  
 - Next, decrease the value of `offset` by 1 using `offset = offset - 1`.  
 - Finally, still within your loop, print out `offset` so you can see how it changes.

\*\*Answer:\*\*

Here is the Python code that solves the problem:

# Initialize offset  
offset = 8  
  
# Code the while loop  
while offset != 0:  
 print("correcting...")  
 offset = offset - 1  
 print(offset)

\*\*Explanation:\*\*

1. \*\*Initialization\*\*: The variable `offset` is initialized to `8`.  
2. \*\*While loop condition\*\*: The `while` loop runs as long as `offset` is not equal to `0`.  
3. \*\*Inside the loop\*\*:  
 - The message `correcting...` is printed.  
 - The value of `offset` is decreased by `1` using `offset = offset - 1`.  
 - The updated value of `offset` is printed to show how it changes during each iteration.  
4. \*\*Loop termination\*\*: When `offset` reaches `0`, the condition `offset != 0` evaluates to `False`, and the loop terminates.