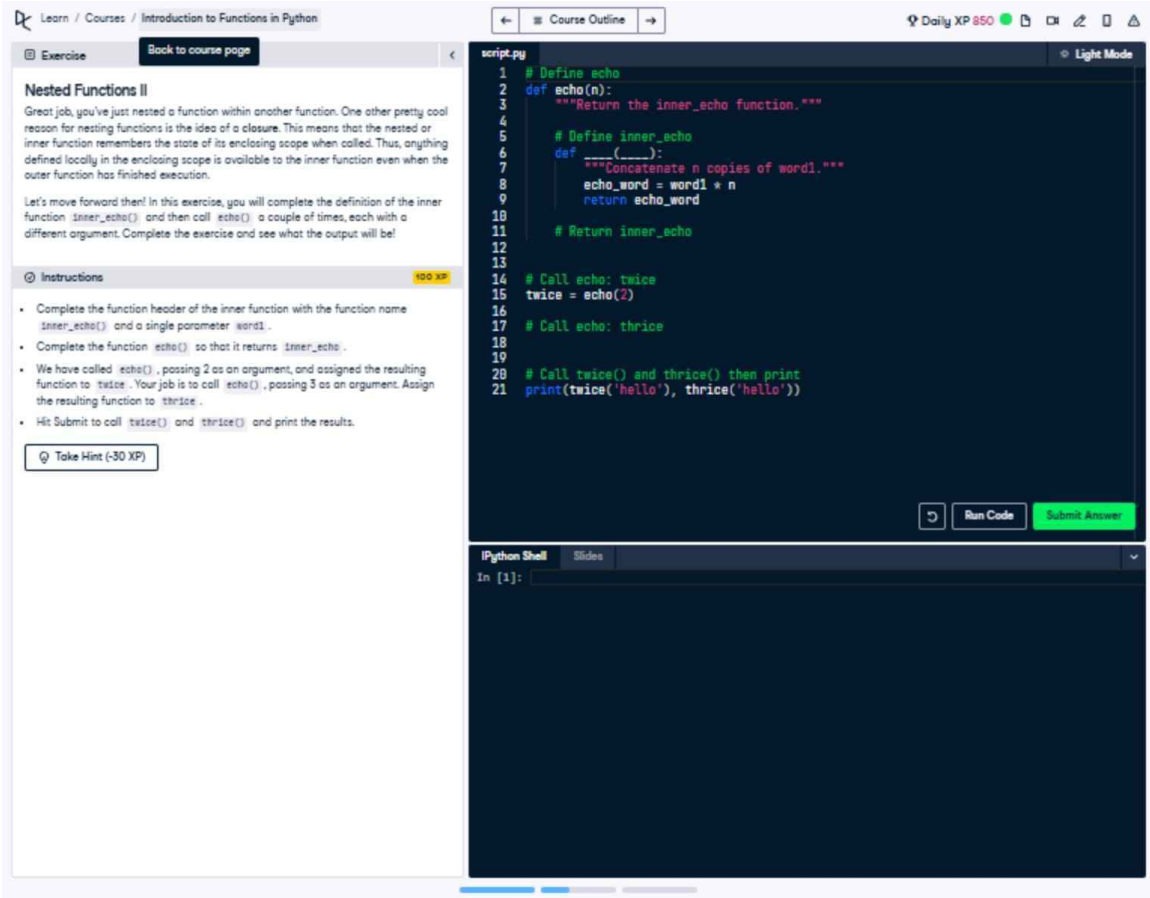
Nested Functions II



Question:

Great job, you've just nested a function within another function. One other pretty cool reason for nesting functions is the idea of a closure.
This means that the inner function remembers the state of its enclosing scope when called.
Thus, anything defined locally in the enclosing scope is available to the inner function even when the outer function has finished execution.

Instructions:
1. Complete the function header of the inner function with the function name `inner_echo()` and a single parameter `word1`.
2. Complete the function `echo()` so that it returns `inner_echo`.
3. We have called `echo()`, passing 2 as an argument, and assigned the resulting function to `twice`. Your job is to call `echo()`, passing 3 as an argument. Assign the resulting function to `thrice`.

4. Hit Submit to call `twice()` and `thrice()` then print the results.

Answer:

```
# Define echo
def echo(n):
    """Return the inner_echo function."""

    # Define inner_echo
    def inner_echo(word1):
        """Concatenate n copies of word1."""
        echo_word = word1 * n
        return echo_word

    return inner_echo

# Call echo: twice
twice = echo(2)

# Call echo: thrice
thrice = echo(3)

# Call twice() and thrice() then print
print(twice('hello'), thrice('hello'))
```

Explanation:

1. def echo(n): - This defines the outer function `echo`, which takes a single parameter `n`. It will return a nested function `inner_echo`.
2. def inner_echo(word1): - Defines a nested function `inner_echo` that takes a single parameter `word1`. This function concatenates `word1` repeated `n` times.
3. echo_word = word1 * n - Creates a string by repeating `word1` exactly `n` times.
4. return inner_echo - The outer function `echo` returns the inner function `inner_echo`.
5. twice = echo(2) - Calls `echo` with `n=2` and assigns the resulting function `inner_echo` to the variable `twice`.
6. thrice = echo(3) - Calls `echo` with `n=3` and assigns the resulting function `inner_echo` to the variable `thrice`.
7. print(twice('hello'), thrice('hello')) - Calls the functions `twice` and `thrice` with the argument `'hello'`, printing the results `'hellohello'` and

`'hellohellohello'` respectively.