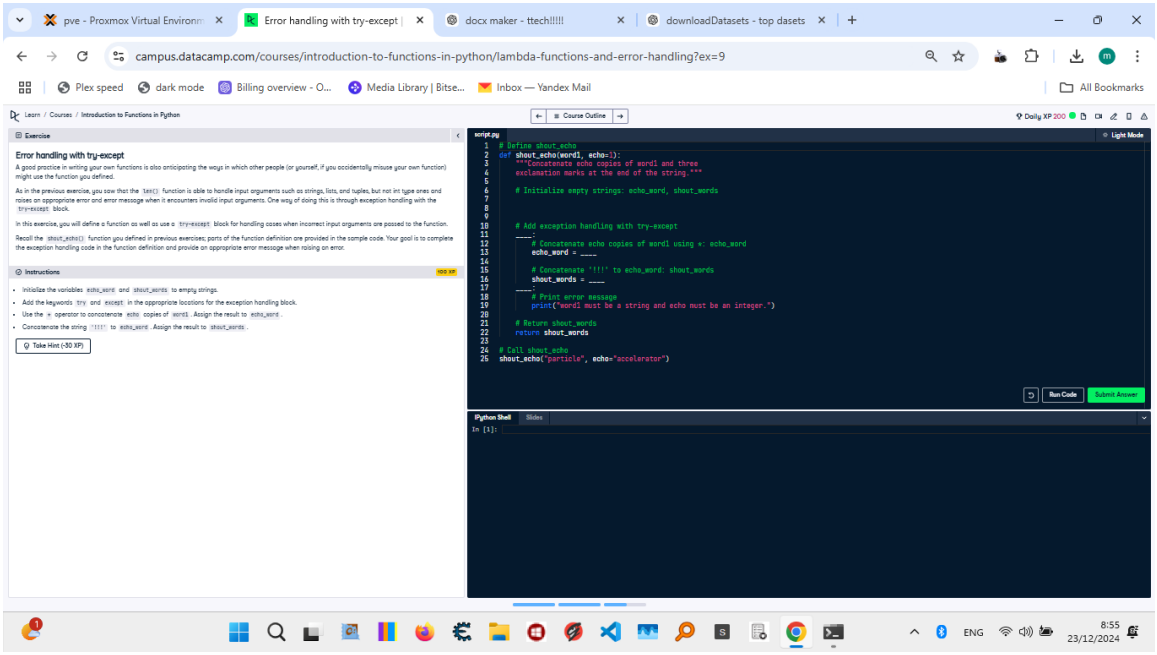
Error Handling with try-except



Question:

A good practice in writing your own functions is also anticipating the ways in which other people (or yourself, if you accidentally misuse your own function) might use the function and provide safeguards.

As in the previous exercises, you saw that the `len()` function is able to handle input arguments such as strings, lists, and tuples, but not int type ones and raises an appropriate error and error message when it encounters invalid input arguments.
One way of doing this is through incorporating safeguards such as a `try-except` block.

In this exercise, you will define a function as well as use a `try-except` block for handling cases when incorrect input arguments are passed to the function.

Instructions:
1. Initialize the variables `echo_word` and `shout_words` to empty strings.
2. Add the keywords `try` and `except` in the appropriate locations for the exception handling block.
3. Use the `*` operator to concatenate `echo` copies of `word1`. Assign the result to `echo_word`.

4. Concatenate the string `"!!!"` to `echo_word`. Assign the result to `shout_words`.

Answer:

```
# Define shout_echo
def shout_echo(word1, echo):
    """Concatenate echo copies of word1 and three exclamation marks at the end of the string."""

    # Initialize empty strings: echo_word, shout_words
    echo_word = ""
    shout_words = ""

    try:
        # Concatenate echo copies of word1 using *: echo_word
        echo_word = word1 * echo

        # Concatenate '!!!' to echo_word: shout_words
        shout_words = echo_word + "!!!"
    except TypeError:
        # Print error message
        print("word1 must be a string and echo must be an integer.")

    # Return shout_words
    return shout_words

# Call shout_echo
shout_echo("particle", echo="accelerator")
```

Explanation:

1. **Initialization:**
   The variables `echo_word` and `shout_words` are initialized to empty strings to ensure they are defined in case of exceptions.

2. **try Block:**
   In the `try` block, the code attempts to concatenate `echo` copies of `word1` using the `*` operator. It then appends `"!!!"` to `echo_word` to create `shout_words`.

3. **except Block:**

If a `TypeError` is raised (for example, if `word1` is not a string or `echo` is not an integer), an appropriate error message is printed: `"word1 must be a string and echo must be an integer."`.

4. **Function Call:**
The function `shout_echo` is called with `"particle"` as `word1` and `"accelerator"` as `echo`, which raises a `TypeError` because `echo` is expected to be an integer but is passed as a string.

5. **Error Handling:**
The exception handling block catches the error and prevents the program from crashing, providing a user-friendly error message instead.