**Creating custom Errors**

**Introduction:**

In the previous lesson, we learned how to make our contracts more gas efficient. In this lesson, we will further enhance their efficiency.

**Require:**

One way to improve gas efficiency is by optimizing our `require` statements. Currently, the `require` statement forces us to store the string ‘sender is not an owner’. Each character in this string is stored individually, making the logic to manage it complex and expensive.

**Custom Errors:**

Introduced in Solidity 0.8.4, custom errors can be used in `revert` statements. These errors should be declared at the top of the code and used in `if` statements. The cheaper error code is then called in place of the previous error message string, reducing gas costs.

We can start by creating a custom error:

Error NotOwner();

Then, we can replace the `require` function with an `if` statement, using the `revert` function with the newly created error:

If (msg.sender != i\_owner) {

Revert NotOwner();

}

By implementing custom errors, we reduce gas costs and simplify error handling in our smart contracts.

**Conclusion:**

In this lesson, we have learned how to further optimize gas efficiency in Solidity contracts by using custom errors instead of traditional require statements with strings.