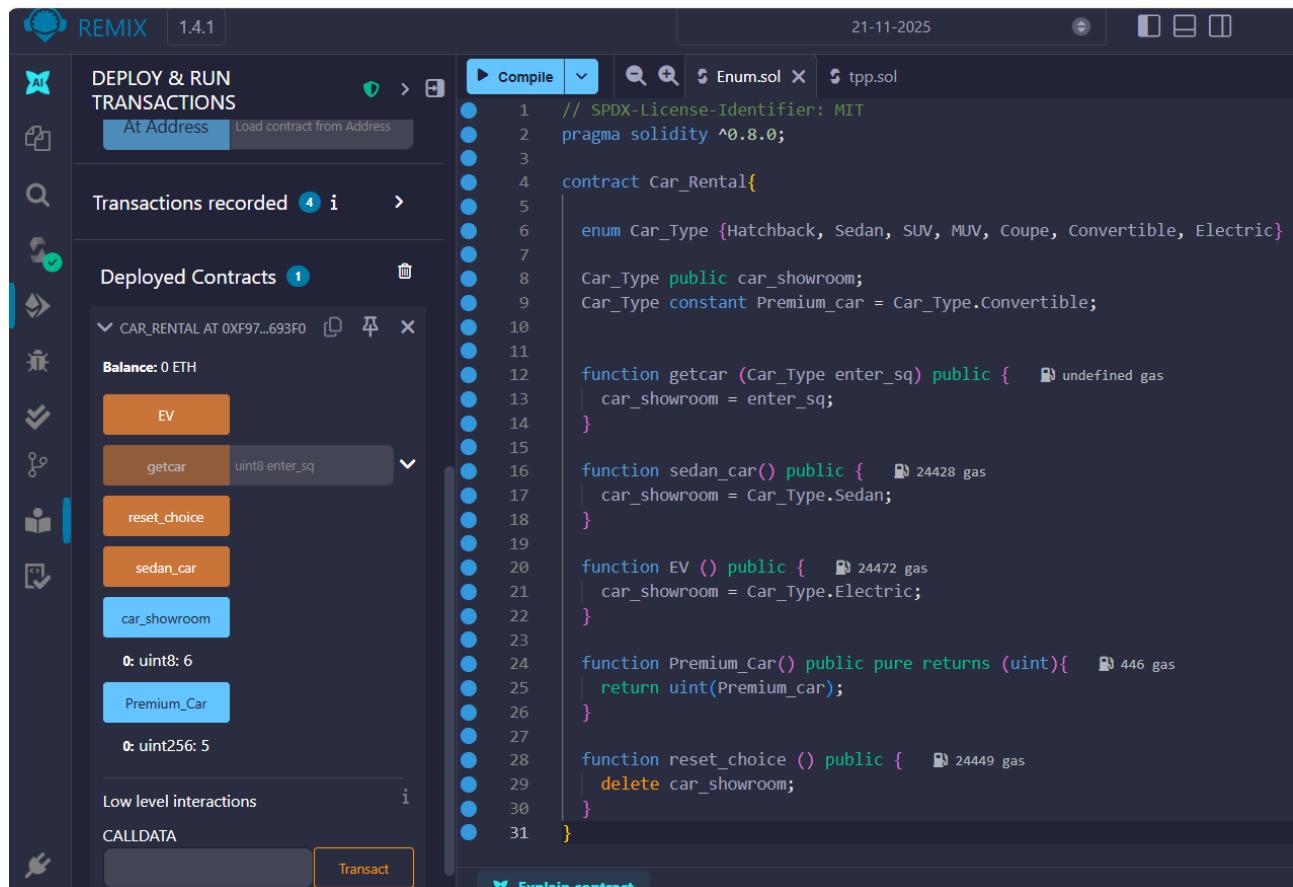


## Car Rental Smart Contract For Select Car Using Enum (Solidity)

This document describes a Solidity smart contract that demonstrates the use of **enums** to manage different car types in a car rental or showroom system. It is designed for beginners learning blockchain development.



The screenshot shows the REMIX 1.4.1 interface. On the left, there's a sidebar with various icons for interacting with the contract. The main area is divided into sections: "DEPLOY & RUN TRANSACTIONS" and "Deployed Contracts". Under "Deployed Contracts", it shows a deployed contract named "CAR\_RENTAL AT 0xF97...693F0" with a balance of 0 ETH. Below this, there are several buttons for interacting with the contract: "EV", "getcar", "reset\_choice", "sedan\_car", "car\_showroom", and "Premium\_Car". To the right of these buttons, there are two tabs: "Enum.sol" and "tpp.sol". The "Enum.sol" tab contains the Solidity code for the "Car\_Rental" contract, which defines an enum for car types and includes functions for selecting different car types and returning the premium car type.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract Car_Rental{
    enum Car_Type {Hatchback, Sedan, SUV, MUV, Coupe, Convertible, Electric}
    Car_Type public car_showroom;
    Car_Type constant Premium_car = Car_Type.Convertible;

    function getcar (Car_Type enter_sq) public {
        car_showroom = enter_sq;
    }

    function sedan_car() public {
        car_showroom = Car_Type.Sedan;
    }

    function EV () public {
        car_showroom = Car_Type.Electric;
    }

    function Premium_Car() public pure returns (uint){
        return uint(Premium_car);
    }

    function reset_choice () public {
        delete car_showroom;
    }
}
```

### Features

- Enum-based car categories
- Car selection using functions
- Constant premium car type
- Reset functionality using delete
- Solidity version ^0.8.0

### Available Car Types

Hatchback, Sedan, SUV, MUV, Coupe, Convertible, Electric

### Key Solidity Concepts Used

enum, state variables, constant variables, pure functions, delete keyword