

# LESSON SC 5 – msg.sender

University of West Attica Department of Electrical and Electronics Engineering Ioannis Christidis Christoforos Kachris

Support by Ethereum Foundation ESP

# What will we accomplish!

In this lesson we will learn about two concepts:

- 1. constructor
- 2. msg.sender

We will continue on our ICR.sol.



#### <u>Owner of Cars = Owner of the Smart</u> Contract



In a previous lesson, we talked about how the owner of the cars is able register their cars inside the SC.

We will build the <u>ICR SC to only have one owner</u>. (Later in the course we will expand our logic for multiple owners)

To interact with SC, the owner uses their account that has an address.

We could save their address in the SC so that we can use it in the ICR functionality.

For example, the registerCar function should only be used by the owner.

For now, let's assume that <u>the owner is the deployer of the SC</u>.

So we will create a public address *state variable* called owner.

address public owner;



# Access the Deployer the Smart Contract

If you remember from a previous lesson, we mentioned that <u>when a transaction is made on the</u> <u>blockchain, there is a from address in the transaction details</u>.

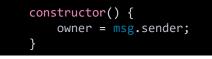
#### This from address is the address that made the transaction.

Knowing that, <u>we can access this address inside our SC with the unique global variable</u>, <u>msg.sender</u> that solidity has available.

And since we know that <u>the owner will be the address that deploys the SC</u>, we can <u>get their</u> <u>address in the constructor</u>.

The constructor, as we saw before, is <u>a special function that runs only once during the</u> <u>deployment of the SC.</u> If a SC does not have a constructor, the default behavior is equivalent to an empty constructor.

So, in the constructor we will make the owner address equal to the msg.sender.





### Outro

Nice, we learned about the msg.sender and the constructor.

Next, we will go over some *access control concepts*.