Making the Zombie Factory

Summary

All Solidity code is encapsulated in contracts. A contract is the fundamental building block of Ethereum applications.

All Solidity source code usually start with a version pragma used to declare the version of the Solidity compiler the code should use.

Variables declared directly in the contract are called state variables and they are stored permanently in contract storage, being written to the blockchain.

Reference types must have an explicit data location specified. One of the possible data locations is `memory` .

There is an `event` type which is a way for the contract to communicate that something happened on the blockchain to an app that can listen for certain events and act on them.

We can use "Web3.js", an Ethereum javascript library to create a frontend that interacts with a contract.

What other data locations are there?

There are three data locations:

- `memory`
- `storage`
- `calldata`

Memory variables are temporary variables stored in memory. **Storage** variables are persistent, they are written on the blockchain and are the most expensive to modify. **Calldata** variables are temporary like memory variables, but their value cannot be changed.