

# Topics in Solidity

Hari

Dappcamp. May 25, 2022

# Designing contracts for gas efficiency

Why?

- ▶ Execution in EVM costs gas.
- ▶ You need to architect the code to minimize gas.
- ▶ This does **not** mean using inline assembly.

## Gas consuming operations

- ▶ Storage: around 22000 gas for writing a fresh data in storage. Reading from storage is 2100 gas.
- ▶ External calls. Calls (cold) has 2600 gas base. Additional 9000 gas for sending ETH in a call, as well as memory expansion costs.

## Example of a bad design

```
https://gist.github.com/hrkrshnn/  
1025ec2b7672b1a1e2e40b2ab9508a75
```

## Design tip

- ▶ Try to minimize storage writes, storage reads and external calls.

## What can we improve here?

```
contract C {  
    address owner = msg.sender;  
    function receive() external payable {  
    }  
    function withdraw() external {  
        require(msg.sender == owner);  
        payable(msg.sender).transfer(address(this).balance);  
    }  
}
```

# Immutable

```
contract C {  
    address immutable owner = msg.sender;  
    function receive() external payable {  
    }  
    function withdraw() external {  
        require(msg.sender == owner);  
        payable(msg.sender).transfer(address(this).balance);  
    }  
}
```

Use constants if values are known at compile time. Use immutables if values are known at deploy time.

## Uniswap example

`https://github.com/Uniswap/v2-core/blob/master/  
contracts/UniswapV2Factory.sol`

`https://github.com/Uniswap/v3-core/blob/main/  
contracts/UniswapV3Factory.sol`



## Contract deployment

- ▶ Runtime code: the code that gets executed when you call a deployed contract.
- ▶ Deploy time code: the code that is part of the constructor and returns the runtime code.
  - ▶ For example, the constructor can initialize the relevant state variables (**sstore**). Then the final code gets returned.

```
contract One {
  uint _one;
  constructor() {
    _one = 1;
  }
  function one() external view returns (uint) {
    return _one;
  }
}
```

## Exercise

1. Try deploying a simple contract in Remix that uses a state variable.
2. Change the state variable to an immutable.
3. Use the remix debugger to debug the deployment transaction in both cases.

## Slides

<https://hrkrshnn.com/t/dappcamp.pdf>