

Blockchain and Smart Contracts

Dr Joshua Ellul

joshua.ellul@um.edu.mt

Department of Computer Science

University of Malta

- Download and Install NodeJS
 - <https://nodejs.org/en/download/>

Compiling

- Many different ways:
 - the online compiler:
 - <https://remix.ethereum.org>
 - Truffle (we'll see soon)
 - Using solc and web3
 - ... more

HelloWorld

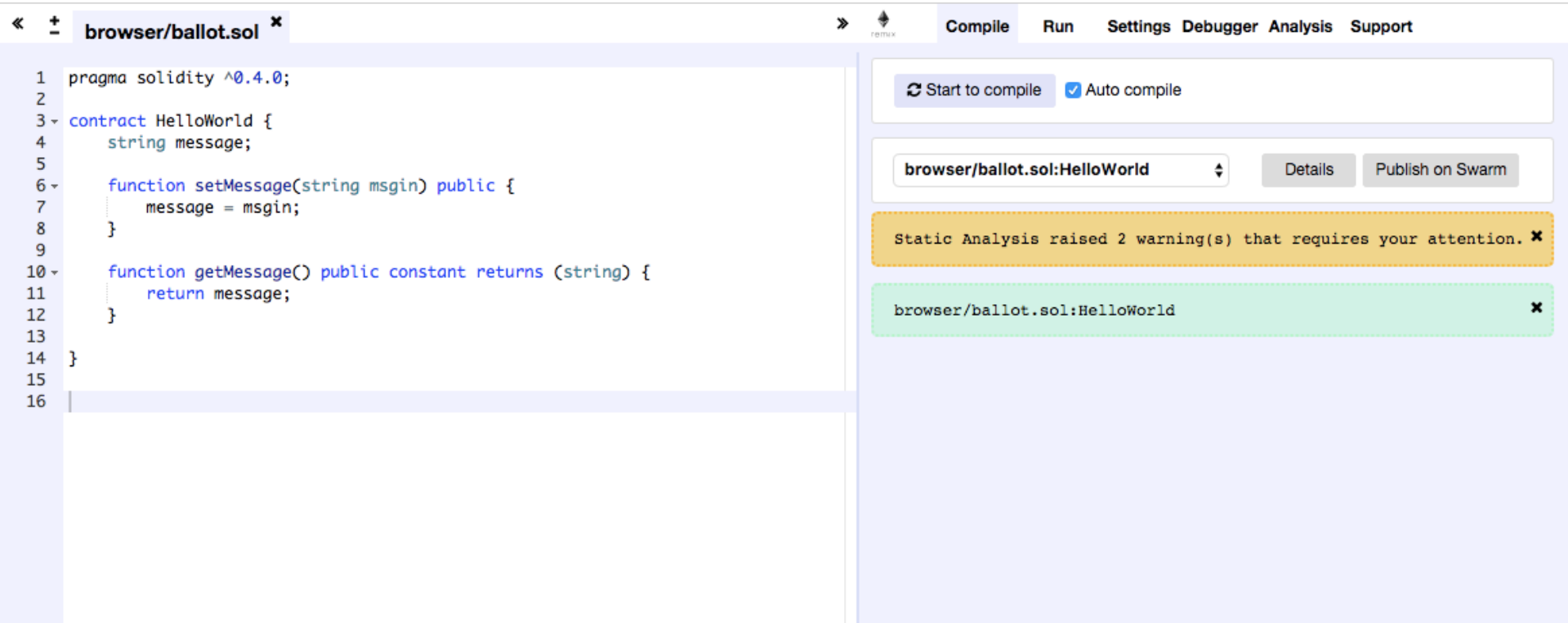
```
pragma solidity ^0.4.0;

contract HelloWorld {
    string message;

    function setMessage(string msgin) public {
        message = msgin;
    }

    function getMessage() public constant returns (string) {
        return message;
    }
}
```

remix.ethereum.org



The image shows the Remix IDE interface. On the left, a code editor displays a Solidity contract named `HelloWorld`. The contract includes a `setMessage` function and a `getMessage` function. The right panel shows the `Compile` tab, which includes a `Start to compile` button, an `Auto compile` checkbox, a dropdown menu showing the selected contract, and buttons for `Details` and `Publish on Swarm`. A warning message from Static Analysis is displayed below the dropdown.

```
1 pragma solidity ^0.4.0;
2
3 contract HelloWorld {
4     string message;
5
6     function setMessage(string msgin) public {
7         message = msgin;
8     }
9
10    function getMessage() public constant returns (string) {
11        return message;
12    }
13 }
14
15
16
```

Compile Run Settings Debugger Analysis Support

Start to compile ☒ Auto compile

browser/ballot.sol:HelloWorld Details Publish on Swarm

Static Analysis raised 2 warning(s) that requires your attention. ✕

browser/ballot.sol:HelloWorld ✕

Setting up a Test Environment

- Easiest way to test ethereum and solidity:
 - EthereumJS
 - Runs on NodeJS
 - Install:
 - `npm install -g ethereumjs-testrpc`
 - (sudo?)
 - Run testrpc:
 - `$ testrpc`

Writing your First Contract and Deploying using Truffle

- Install truffle:
 - `npm install -g truffle`
 - (sudo?)
- Create project:
 - `mkdir test_contract`
`cd test_contract`
`truffle init`

Writing your First Contract and Deploying using Truffle (cont)

- Create the contract file, in the **contracts directory** HelloWorld.sol:

```
pragma solidity ^0.4.0;

contract HelloWorld {
    string message;

    function setMessage(string msgin) public {
        message = msgin;
    }

    function getMessage() public constant returns (string) {
        return message;
    }
}
```


Writing your First Contract and Deploying using Truffle (cont)

- Create migrations/2_deploy_contracts.js and modify to look like:

```
var HelloWorld = artifacts.require("./HelloWorld.sol");
module.exports = function(deployer) {
  deployer.deploy(HelloWorld);
};
```

Writing your First Contract and Deploying using Truffle (cont)

- Configure truffle.js to connect to localhost:

```
module.exports = {  
  networks: {  
    development: {  
      host: "localhost",  
      port: 8545,  
      network_id: "*" // Match any network id  
    }  
  }  
};
```

Writing your First Contract and Deploying using Truffle (cont)

- Ensure testrpc is running
- Compile, HelloWorld and then deploy:
truffle compile
truffle migrate

```
Joshuas-MacBook-Air:test_contract joshuaellul$ truffle migrate
Using network 'development'.

Running migration: 1_initial_migration.js
  Deploying Migrations...
  ... 0x0511876fbd65e88fa51ee69f5de53942a189eb03aebaf3b8927d4a2a9ccdf321
  Migrations: 0xeb807074bf066ff95b2f8a61b88c9bb2732fd0ad
  Saving successful migration to network...
  ... 0xa98056f37aa1c0263ef93c091455da8fb21668866e04c1858e08cdf420070c5c
[Saving artifacts...
Running migration: 2_deploy_contracts.js
  Deploying HelloWorld...
  ... 0x3a3420a3c402ce535850438e953006fe43fb0a52400cf4e0ef31527b093b87c9
  HelloWorld: 0xe1e494d6a86fb02c4b976e08455d60013701a281
  Saving successful migration to network...
  ... 0xe3e6e5a38cfc6fc74dd6508384eb6368787061f08acca1c47047eeaf0a74f594
  Saving artifacts...
```

Writing your First Contract and Deploying using Truffle (cont)

- testrpc output (contract created?)

```
Transaction: 0x0511876fbd65e88fa51ee69f5de53942a189eb03aebaf3b8927d4a2a9ccdf321
Contract created: 0xeb807074bf066ff95b2f8a61b88c9bb2732fd0ad
Gas usage: 269543
Block Number: 1
```

Writing your First Contract and Deploying using Truffle (cont)

- Test the contract:
truffle console

- Call getMessage:

```
[truffle(development)> HelloWorld.deployed().then(instance => instance.getMessage.call())
```

- Set the message:

```
[truffle(development)> HelloWorld.deployed().then(instance => instance.setMessage.sendTransaction('hello world'))  
'0x6080086ecf31c4e53de1c82ddce56386ba2e8c8eefc3dcc9becf8586e7cf4e33'
```

- Check again:

```
[truffle(development)> HelloWorld.deployed().then(instance => instance.getMessage.call())  
'hello world'  
_
```

Anatomy of a Smart Contract

```
pragma solidity ^0.4.0;

contract Overview {
    uint256 someField;

    //constructor
    function Overview() public {
        someField = 0;
    }

    function publicStateChange(uint256 inputState) public {
        someField = inputState;
    }

    function publicConstant() public constant returns (uint256) {
        return someField;
    }

    function acceptsEther() public payable {
        someField += msg.value;
    }
}
```

Lab

- Go through the 'Solidity in Depth' tutorial starting from:
<http://solidity.readthedocs.io/en/develop/solidity-in-depth.html>
- All the way to the Cheat Sheet:
<http://solidity.readthedocs.io/en/develop/miscellaneous.html#cheatsheet>

Assignment

- Use case: A company's sole managing director wants to allow for the shareholders to make (binary) decisions, which he will propose to the share holders. For the sake of keeping decisions secret, until the director deems fit, the director would like to be the only one who is aware of how shareholders have voted until the director decides to let the shareholders know the outcome for a particular decision.
- Create an Ethereum smart contract that allows for the following functionality:

Assignment

- 1. The director will be the one to upload the contract. He should thereafter be recognised as the director because he was the one to upload the contract.
- 2. The director would like the ability to upload any number of questions (which require a true or false response to). The director will upload each question, one at a time.

Assignment

- 3. The director would like the ability to add and remove shareholders from being able to vote and being able to see results for approved decisions at any point.
- 4. Each shareholder may only vote for each decision once.
- 5. The director should be able to close the voting process for a specific question. The majority result should then be able to be seen by all shareholders.