ETHERSCAN CONTRACT SOURCE CODE VERIFICATION

```
6060604052341561000f576
00080fd5b6040516103a838
03806103a88339810160405
280805182019190505b5b5b
336000806101000a8154817
3ffffffffffffffffffffff
ffffffffffffffff02191
6908373ffffffffffffffff
ffffffffffffffffffffffff
f160217905505b80600190
805190602001906100849 29
```

```
pragma solidity ^0.4.0;

contract Demo {
    uint a;

    function Demo(unint _a) {
        a = _a;
    }

    function chng(_a) {
        a = 5 + _a;
    }
}
```

VERIFY AND PUBLISH YOUR SOLIDITY SOURCE CODE

- Demonstrate how to verify and publish your solidity source code on https://etherscan.io. In this demo i will be using the Etherscan Verify Contract Code version 2.0 on the Rinkeby test network. See: https://rinkeby.etherscan.io/verifyContract2

- To verify and publish your solidity source code on Etherscan you need:
  Contract address
  Contract name
  Compiler version used. See: https://youtu.be/YzhhgY6lqKI
  Optimization enabled/disabled
  Solidity contract code
  Runs (Optimizer) value
  Constructor arguments ABI-encoded

SOLC BYTECODE OPTIMIZER

- The solidity compiler solc has the following flags:
  solc —optimize —optimize-runs 200

- The meaning of these flags:
  optimize: Enable bytecode optimizer.
  optimize-runs: Estimated number of contract runs for optimizer tuning
  If not specified defaults to 200.

- Before you deploy your contract enable the bytecode optimiser.
  It will optimise the bytecode thus using less gas and saving you money.

- Remix IDE uses the default optimise-runs value of 200.

CONSTRUCTOR ARGUMENTS ABI-ENCODED

- When you deploy your solidity contract, you can pass constructor argument(s).

```
pragma solidity ^0.4.0;

contract Demo {
    uint a;
    function Demo(uint _a) {
        a = _a;
    }
}
```

- For example, constructor argument value: 10

CONSTRUCTOR ARGUMENTS ABI-ENCODED

- The value 10 needs to be encoded into an **A**pplication **B**inary **I**nterface (ABI) format.
- How this is done is explained in the Ethereum Contract ABI specification: https://github.com/ethereum/wiki/wiki/Ethereum-Contract-ABI
- Explaining the Ethereum Contract ABI specification is a subject for another video.
- You can easily encode constructor arguments into ABI format using a Node.js script: http://www.mobilefish.com/download/ethereum/constructor_arguments_in_abi.js.txt

VERIFY AND PUBLISH YOUR SOLIDITY SOURCE CODE

- In the YouTube video "Compile and deploy solidity code", see:
  https://youtu.be/nnVX6fQUu4o
  i have deployed the same contract several times on the Rinkeby test network using the online Remix IDE, Node.js script and Truffle.

- I will verify and publish two contracts on Etherscan:

  - One contract deployed using the online Remix IDE.

  - One contract deployed using Truffle.

  - The contract can be found at:
    http://www.mobilefish.com/download/ethereum/DemoCombined.sol.txt

CONTRACT DEPLOYED USING REMIX IDE

- Contract address: **0xF943c4db517A12a47e57B8A3fcDfD9FFc1bde036**
  Contract name: **Demo**
  Compiler version used: **v0.4.14**
  Optimization: **enabled**
  Solidity contract code:
  http://www.mobilefish.com/download/ethereum/DemoCombined.sol.txt
  Constructor arguments ABI-encoded [(uint) 10, (string) Hello]:
  0000000000000000000000000000000000000000000000000000000a0000000000000000000000000000000000000000000000000000000000000040000000000000000000000000000000000000000000000000000000000000000548656c6c6f000000000000000000000000000000000000000000000000000000

CONTRACT DEPLOYED USING TRUFFLE

- Contract address: **0x6b78D667309D681D596476b70F2F20998bD3c093**
  Contract name: **Demo**
  Compiler version used: **v0.4.13. (Truffle version 3.4.7)**
  Optimization: **enabled**
  Solidity contract code:
  http://www.mobilefish.com/download/ethereum/DemoCombined.sol.txt
  Constructor arguments ABI-encoded [(uint) 10, (string) Hello]:
  0000000000000000000000000000000000000000000000000000000a0000000000000000000000000000000000000000000000000000000000000040000000000000000000000000000000000000000000000000000000000000000548656c6c6f000000000000000000000000000000000000000000000000000000

CONTRACT DEPLOYED USING TRUFFLE

- Truffle enable bytecode optimizer, see: https://github.com/trufflesuite/truffle-compile/blob/master/index.js