

Smart Contract Testing

Chris Smith

@iamchrissmith



<https://www.meridio.co>



Tests

- They're good for your health!
- Help you identify bugs/vulnerabilities in your code
- Help you manage for Solidity upgrades
- Help debug when writing your contracts
- Can provide pseudo-documentation for your contracts

<code>

From Truffle Docs on Testing

Clean-room environment

Truffle provides a clean room environment when running your test files. When running your tests against [Ganache](#) or Truffle Develop, Truffle will use advanced snapshotting features to ensure your test files don't share state with each other. When running against other Ethereum clients like [go-ethereum](#), Truffle will re-deploy all of your migrations at the beginning of every test file to ensure you have a fresh set of contracts to test against.


```
before(async () => {  
    this.capValidator = await CapValidator.deployed();  
    this.simpleToken = await SimpleToken.deployed();  
});
```

```
let capValidator;  
let simpleToken;  
beforeEach(async () => {  
    capValidator = await CapValidator.new({from: owner});  
    simpleToken = await SimpleToken.new(capValidator.address, {from: owner});  
});
```

<code>

Some other testing resources

OpenZeppelin: <https://github.com/OpenZeppelin/openzeppelin-solidity>

Solidity-coverage: <https://github.com/sc-forks/solidity-coverage>

Eth-gas-reporter: <https://github.com/cgewecke/eth-gas-reporter>

Mythril: <https://github.com/ConsenSys/mythril>

Twitter/Github: @iamchrissmith



<https://www.meridio.co>