Changes in Solidity 0.5.x

Starting with Solidity 0.5.0 there are some **minor changes** that are not backward compatible with older version of Solidity (0.4.x).

There are just a few changes that are explained bellow.
Any contract compiled by solidity compiler 0.4.x can be transformed in a contract compatible with version 0.5.x with just 1-3 small changes.

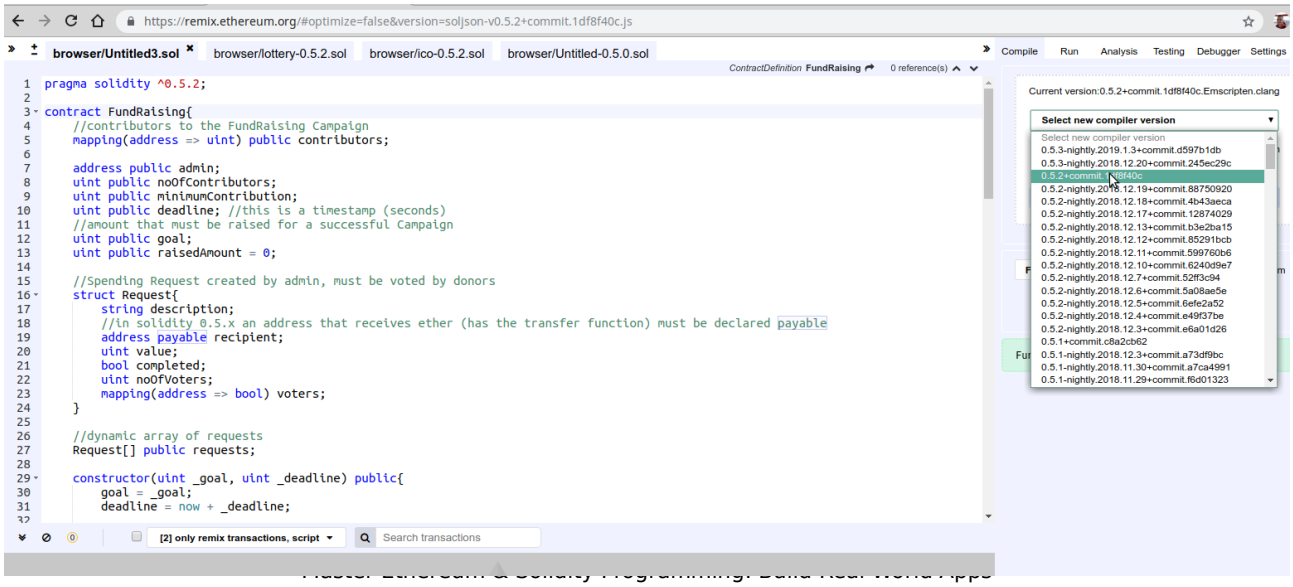
In the resource directory that contains all smart contracts and examples developed in the course, **you'll find 2 versions** (.sol files): one that can be compiled by solidity compiler 0.4.x and another that can be compiled by solidity compiler greater than 0.5.x

Course resource directory:
https://drive.google.com/drive/folders/1x8u4ZBHeSL65Kn81DND0iR4IOmH30Efh

Exemple:
lottery-solidity-0.4.x.sol -> should be compiled by solidity 0.4.29
lottery-solidity-0.5.x.sol -> should be compiled by solidity 0.5.0 or greater

If there is a single version (Exemple: cryptos-fully-compliant-erc20-token.sol) it means that it compiles with any version of solidity compiler without any modification.

If you use the Remix IDE, you can choose the compiler version in COMPILE TAB



Changes in Solidity 0.5.0:

1. The **address data type** was split into **address** and **address payable**, where only address payable provides the **transfer()** function.
If you want to transfer ether to an address, that address must be declared payable:

```
address payable public my_address; //in solidity 0.5.x
address public my_address;      //in solidity 0.4.x
```

2. The fallback function must be declared external, explicitly. It cannot be declared public.

```
  //the fallback payable  function must be external in solidity 0.5.x
  function () external {
  }

  //this was possible in solidity 0.4.x
  function () public{
  }
```

3. The keccak256() function accepts only a single bytes argument. We use the **abi.encodePacked()** function to get the bytes argument from more values

```
//solidity 0.5.x
keccak256(abi.encodePacked(block.difficulty, block.timestamp,
players.length));

//solidity 0.4.x
keccak256(block.difficulty, block.timestamp, players.length);
```

4. The string argument of any setter function must be declared in **memory** and the string return argument of any getter function must also be declared in **memory**

```
//in solidity 0.5.x memory keyword is mandatory for string arguments
function setLocation(string memory _location) public{
        location = _location;
}
    //in solidity 0.5.x memory keyword is mandatory for return string argument
    function getLocation() public view returns(string memory){
        return location;
    }
```

```
//solidity 0.4.x
function setLocation(string _location) public{
        location = _location;
}
//solidity 0.4.x
function getLocation() public view returns(string){
        return location;
}
```

5. Explicit function visibility is mandatory in Solidity 0.5.x.
To migrate from 0.4.x to 0.5.x add public to every function and constructor, and external to every fallback or interface function that does not specify its visibility already.

```
  //in solidity 0.5.x visibilty specifier is mandatory
    function setLocation(string memory _location) public{
        location = _location;
    }
```

```
  //in solidity 0.4.x by default a function is public
    function setLocation(string memory _location) {
        location = _location;
    }
```

For a full list of Solidity 0.5.x changes check this link bellow:
https://solidity.readthedocs.io/en/v0.5.2/050-breaking-changes.html