```solidity
1  pragma solidity ^0.4.8;
2
3  contract Hello {
4
5      // A string variable
6      string public greeting;
7
8      // Events that gets logged on the blockchain
9      event GreetingChanged(string _greeting);
10
11     // The function with the same name as the class is a constructor
12     function Hello(string _greeting) {
13         greeting = _greeting;
14     }
15
16     // Change the greeting message
17     function setGreeting(string _greeting) {
18         greeting = _greeting;
19
20         // Log an event that the greeting message has been updated
21         GreetingChanged(_greeting);
22     }
23
24     // Get the greeting message
25     function greet() constant returns (string _greeting) {
26         _greeting = greeting;
27     }
28 }
29
```

Solidity version: 0.4.8+commit.60cc1668.Emscripten.clang

Change to: 0.4.10-nightly.2017.3.3+commit.6bfd894f ▾

☐ Text Wrap   ☐ Enable Optimization   ☑ Auto Compile   ⚙ Compile

🟩 Attach   🟥 Transact   🟥 Transact (Payable)   🟦 Call

▾ Hello   1403 bytes

At Address	Create	string _greeting

Bytecode   606060405234610000576040516105760565b380380610576b833981016040528

Interface   [{"constant":false,"inputs":[{"name":"_greeting","type":"string"}],"name":"set

Web3 deploy
```
var _greeting = /* var of type string here */ ;
var helloContract = web3.eth.contract([{"constant":fal
var hello = helloContract.new(
   _greeting,
   {
     from: web3.eth.accounts[0],
     data: '0x606060405234610000576040516105760565b380380610576b38038061
     gas: '4700000'
   }, function (e, contract){
   console.log(e, contract);
   if (typeof contract.address !== 'undefined') {
        console.log('Contract mined! address: ' + con
   }
})
```

Metadata location   bzzr://a63d0b3449ebe3923dda93af66f138c1aef28f4a1d3a51f6c4f1c6326(

Toggle Details

browser/escrow.sol

FunctionDefinition deposit →   0 reference(s) ∧ ∨   🗄 Execution cost: 20702 gas

▾ browser
  escrow.sol
▸ config

```solidity
1   pragma solidity ^0.4.11;
2
3 ▾ contract Escrow {
4       uint balance;
5       address public manager;
6       address public worker;
7       address private escrow;
8       uint private start;
9       bool managerOk;
10      bool workerOk;
11
12 ▾   function Escrow(address manager_address, address worker_address) public {
13          // this is the constructor function that runs ONCE upon initialization
14          manager = manager_address;
15          worker = worker_address;
16          escrow = msg.sender;
17          start = now; //now is an alias for block.timestamp, not really "now"
18      }
19
20 ▾   function accept() public {
21          if (msg.sender == manager){
22              managerOk = true;
23          } else if (msg.sender == worker){
24              workerOk = true;
25          }
26          if (managerOk && workerOk){
27              payBalance();
28
```

⚠ (warning icon at line 12)

≫   ⊘   ⓪   ☐   [2] only remix transactions, script ▾   🔍 Search transactions

from	0x14723a09acff6d2a60dcdf7aa4aff308fddc160c 📋
to	Escrow.deposit() 0x0fdf4894a3b7c5a101686829063be52ad45bcfb7 📋
gas	3000000 gas 📋
transaction cost	21742 gas 📋
execution cost	470 gas 📋
hash	0xa3f20f58a2908d8ba133ce81e2e1f4d25ac7793ce311b713348ee06d6edb443b 📋
input	0xd0e...30db0 📋
decoded input	{} 📋
decoded output	{}
logs	[] 📋 📋
value	40 wei 📋

Compile   Run   Settings   Analysis   Debugger   Support

Environment   JavaScript VM   ⚡ VM (-) ⇕ ℹ

Account   0x147...c160c (99.9999999999993302 ⇕ 📋 ⊂

Gas limit   3000000

Value   0   ▦   wei ⇕

Escrow ⇕

Deploy ∧

manager_addr [address]
ess:
worker_addres
s:
[address]

🧰   [transact]

Load contract from Address   [At Address]

Transactions recorded: ③   ∨

Deployed Contracts   🗑

▾   Escrow at 0x0fd...bcfb7 (memory)   📋 ✕

[accept]
[cancel]
[deposit]
[kill]
[manager]

0: address:
0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c

[worker]

0: address:

browser/ballot.sol | **browser/Hello.sol** ✕

```solidity
1  pragma solidity ^0.4.0;
2  contract myContract {
3      uint myVariable;
4
5      function myContract() {
6          myVariable = 5;
7      }
8
9      function setMyVariable(uint param) {
10          myVariable = param;
11      }
12
13      function getMyVariable() constant returns (uint) {
14          return myVariable;
15      }
16  }
```

Environment | JavaScript VM   ⚡ VM (-) ↕ ⓘ

Account | 0x147...c160c (99.9999999999974142 ↕ 🗎

Gas limit | 3000000

Value | 100 | finney ↕

myContract ↕

[ ]   **Create**

Load contract from Address | At Address

⌄ ⊘ | [2] only remix transactions, script ⌄ | 🔍 Search transactions | ☐ Listen on network

0 pending transactions   ⊟ 💾 ▶

call to myContract.getMyVariable

[vm] from:0x147...c160c, to:0x 0x, value:0 wei, data:ef551...51770, 0 logs, hash:0x7f3...aacf3   Details   Debug

call to myContract.getMyVariable errored: VM error: invalid opcode.
invalid opcode
     The execution might have thrown.
     Debug the transaction to get more information.

creation of myContract pending...

[vm] from:0x147...c160c, to:myContract.(constructor), value:0 wei, data:0x606...80029, 0 logs, hash:0x8e9...83dbf   Details   Debug

✕

⌄   **myContract at 0x...0x (memory)**   🗎

getMyVariable

setMyVariable | 90

>

browser/MyContract.sol

Compile | Run | Analysis | Testing | Debugger | Settin

ContractDefinition **MyContract** → 0 reference(s) ∧ ∨

▸ browser

▸ config

```solidity
1  pragma solidity 0.5.2;
2
3  contract MyContract {
4      Person[] public people;
5
6      uint256 peopleCount;
7
8      struct Person {
9          string _firstName;
10         string _lastName;
11     }
12
13     function addPerson(string memory _firstName, string memory _lastName) public {
14         people.push(Person(_firstName, _lastName));
15         peopleCount += 1;
16     }
17  }
```

Current
version:0.5.2+commit.1df8f40c.Emscripten.clang

Select new compiler version ⬍

☑ Auto compile   ☐ Enable Optimization
☐ Hide warnings

⟳ Start to compile (Ctrl-S)

MyContract ⬍   ⬆ Swarm

Details   📄 ABI   📄 Bytecode

☐ [2] only remix transactions, script ▾   🔍 Search transactions

Static Analysis raised 2 warning(s) that requires your attention. Click here to show the warning(s). ✕

MyContract ✕

- Welcome to Remix v0.7.5 -

You can use this terminal for:

- Checking transactions details and start debugging.
- Running JavaScript scripts. The following libraries are accessible: