---

title: Compiling smart contracts description: An explanation of why you need to compile smart contracts and what compilation actually does. lang: en incomplete: true

---

You need to compile your contract so that your web app and the Ethereum virtual machine (EVM) can understand it.

Prerequisites {#prerequisites}

You might find it helpful to have read our intro tosmart contracts and the Ethereum virtual machine before reading about compilation.

The EVM {#the-evm}

For the EVM to be able to run your contract it needs to be in**bytecode**. Compilation turns this:

```solidity pragma solidity 0.4.24;

contract Greeter {

function greet() public constant returns (string) {
    return "Hello";
}

} ```

**into this**

```
PUSH1 0x80 PUSH1 0x40 MSTORE PUSH1 0x4 CALLDATASIZE LT PUSH2 0x41 JUMPI PUSH1 0x0 CALLDATALOAD PUSH29
0x100000000000000000000000000000000000000000000000000000000 SWAP1 DIV PUSH4 0xFFFFFFFF AND DUP1 PUSH4 0xCFAE3217
EQ PUSH2 0x46 JUMPI JUMPDEST PUSH1 0x0 DUP1 REVERT JUMPDEST CALLVALUE DUP1 ISZERO PUSH2 0x52 JUMPI PUSH1 0x0
DUP1 REVERT JUMPDEST POP PUSH2 0x5B PUSH2 0xD6 JUMP JUMPDEST PUSH1 0x40 MLOAD DUP1 DUP1 PUSH1 0x20 ADD DUP3 DUP2
SUB DUP3 MSTORE DUP4 DUP2 DUP2 MLOAD DUP2 MSTORE PUSH1 0x20 ADD SWAP2 POP DUP1 MLOAD SWAP1 PUSH1 0x20 ADD SWAP1
DUP1 DUP4 DUP4 PUSH1 0x0 JUMPDEST DUP4 DUP2 LT ISZERO PUSH2 0x9B JUMPI DUP1 DUP3 ADD MLOAD DUP2 DUP5 ADD MSTORE
PUSH1 0x20 DUP2 ADD SWAP1 POP PUSH2 0x80 JUMP JUMPDEST POP POP POP POP SWAP1 POP SWAP1 DUP2 ADD SWAP1 PUSH1 0x1F
AND DUP1 ISZERO PUSH2 0xC8 JUMPI DUP1 DUP3 SUB DUP1 MLOAD PUSH1 0x1 DUP4 PUSH1 0x20 SUB PUSH2 0x100 EXP SUB NOT
AND DUP2 MSTORE PUSH1 0x20 ADD SWAP2 POP JUMPDEST POP SWAP3 POP POP POP PUSH1 0x40 MLOAD DUP1 SWAP2 SUB SWAP1
RETURN JUMPDEST PUSH1 0x60 PUSH1 0x40 DUP1 MLOAD SWAP1 DUP2 ADD PUSH1 0x40 MSTORE DUP1 PUSH1 0x5 DUP2 MSTORE
PUSH1 0x20 ADD PUSH32 0x48656C6C6F0000000000000000000000000000000000000000000000000000 DUP2 MSTORE POP SWAP1
POP SWAP1 JUMP STOP LOG1 PUSH6 0x627A7A723058 KECCAK256 SLT 0xec 0xe 0xf5 0xf8 SLT 0xc7 0x2d STATICCALL ADDRESS
SHR 0xdb COINBASE 0xb1 BALANCE 0xe8 0xf8 DUP14 0xda 0xad DUP13 LOG1 0x4c 0xb4 0x26 0xc2 DELEGATECALL PUSH7
0x8994D3E002900
```

## Web applications {#web-applications}

The compiler will also produce the**Application Binary Interface (ABI)** which you need in order for your application to understand the contract and call the contract's functions.

The ABI is a JSON file that describes the deployed contract and its smart contract functions. This helps bridge the gap between web2 and web3

A JavaScript client library will read the**ABI** in order for you to call on your smart contract in your web app's interface.

Below is the ABI for the ERC-20 token contract. An ERC-20 is a token you can trade on Ethereum.

```
json [{ "constant": true, "inputs": [], "name": "name", "outputs": [{ "name": "", "type": "string" }],
"payable": false, "stateMutability": "view", "type": "function" }, { "constant": false, "inputs": [{ "name":
"_spender", "type": "address" }, { "name": "_value", "type": "uint256" }], "name": "approve", "outputs": [{
"name": "", "type": "bool" }], "payable": false, "stateMutability": "nonpayable", "type": "function" }, {
"constant": true, "inputs": [], "name": "totalSupply", "outputs": [{ "name": "", "type": "uint256" }],
"payable": false, "stateMutability": "view", "type": "function" }, { "constant": false, "inputs": [{ "name":
"_from", "type": "address" }, { "name": "_to", "type": "address" }, { "name": "_value", "type": "uint256" }],
"name": "transferFrom", "outputs": [{ "name": "", "type": "bool" }], "payable": false, "stateMutability":
"nonpayable", "type": "function" }, { "constant": true, "inputs": [], "name": "decimals", "outputs": [{ "name":
"", "type": "uint8" }], "payable": false, "stateMutability": "view", "type": "function" }, { "constant": true,
"inputs": [{ "name": "_owner", "type": "address" }], "name": "balanceOf", "outputs": [{ "name": "balance",
"type": "uint256" }], "payable": false, "stateMutability": "view", "type": "function" }, { "constant": true,
"inputs": [], "name": "symbol", "outputs": [{ "name": "", "type": "string" }], "payable": false,
```
```

"stateMutability": "view", "type": "function" }, { "constant": false, "inputs": [{ "name": "_to", "type": "address" }, { "name": "_value", "type": "uint256" }], "name": "transfer", "outputs": [{ "name": "", "type": "bool" }], "payable": false, "stateMutability": "nonpayable", "type": "function" }, { "constant": true, "inputs": [{ "name": "_owner", "type": "address" }, { "name": "_spender", "type": "address" }], "name": "allowance", "outputs": [{ "name": "", "type": "uint256" }], "payable": false, "stateMutability": "view", "type": "function" }, { "payable": true, "stateMutability": "payable", "type": "fallback" }, { "anonymous": false, "inputs": [{ "indexed": true, "name": "owner", "type": "address" }, { "indexed": true, "name": "spender", "type": "address" }, { "indexed": false, "name": "value", "type": "uint256" }], "name": "Approval", "type": "event" }, { "anonymous": false, "inputs": [{ "indexed": true, "name": "from", "type": "address" }, { "indexed": true, "name": "to", "type": "address" }, { "indexed": false, "name": "value", "type": "uint256" }], "name": "Transfer", "type": "event" }]

Further reading {#further-reading}

- ABI spec – *Solidity*

Related topics {#related-topics}

- JavaScript client libraries
- Ethereum virtual machine