Smart Contract

Once we have the environment set up, we actually need to create a basic contract. The hello world for DApps is a greeter contract that does both reading and writing to the chain.

```
//SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import "hardhat/console.sol";

contract Greeter {
  string greeting;

  constructor(string memory _greeting) {
    console.log("Deploying a Greeter with greeting:", _greeting);
    greeting = _greeting;
  }

  function greet() public view returns (string memory) {
    return greeting;
  }

  function setGreeting(string memory _greeting) public {
    console.log("Changing greeting from '%s' to '%s'", greeting, _greeting);
    greeting = _greeting;
  }
}
```

Our contract is in solidity. This cannot be accessed directly from our Javascript frameworks. For that, we need to "compile" it to Javscript (ABI). Think of this as a wrapper that connects to the contract and can help you call functions of the contract without having to worry about the details (marshalling etc.) yourself.

The actual command is pretty simple:

```
npx hardhat compile
```

A new file will be created in:

```
src/artifacts/contracts/Greeter.sol/Greeter.json
```

We will import this file when writing our react script.

Starting a Dummy Network and Deploying the Contract

```
npx hardhat node
```

These are test accounts created for the purpose of testing the local network. We'll use these for a while. Make sure you don't use them on actual Mainnet or even the public testnets.

Take note of the private key and address of the first test account.

Let's go ahead and import these to our MetaMask Wallet.

First, connect to the localhost network. To do this, click on the `Network` dropdown (top mid) of MetaMask and select `localhost`. Rest of the configurations are fine.

Then, `Account -> Import Account`. Then paste your private key (for first test account) in there. You should have 10,000 ETH in there. They're not too useful though so don't get excited. Let's call this `hh-test0x` account.

Let's create a new account outside of Hardhat and send it some Ether.

Go to Metamask and `Account -> Create Account`. Give it a useful name. I'm using `recly-test0x`. Keep this account safe for now. We'll be using it later on. Save the private key by going to `...` menu (the dots menu), then `Account Details`. Click on `Export Private Key`. Enter the password for MetaMask and save your private key somewhere. All of these are test keys and should defnitely not be used on mainnet.

Switch to `hh-test0x` account and send some ether to `recly-test0x`. (When you do this again the next time you start this tutorial from scratch, you will get a nonce error. For that, simply go to `Accounts -> Settings -> Advanced -> Reset Account`.)

Note the output in console as well as ether changes in your MetaMask accounts.