**Project Setup**

**Introduction:**

Let’s begin by coding `FundMe`, a crowdfunding contract allowing users to send funds, which the owner can later withdraw. Before we start, let’s clean up our Remix IDE workspace

**Setting up the project:**

Start from scratch by opening your Remix IDE (<https://remix.ethereum.org/>) and deleting all existing contracts. Next, create a new contract named `FundMe`.

**Important:** Before you start coding, try to write down in plain English what you want your code to achieve. This helps clarify your goals and structure your approach.

We want `FundMe` to perform the following tasks:

1. **Allow users to send funds into the contract:** users should be able to deposit funds into the ‘FundMe’ contract
2. **Enable withdrawal of funds by the contract owner:** the account that owns `FundMe` should have the ability to withdraw all deposited funds
3. **Set a minimum funding value in USD:** there should be a minimum amount that can be deposited into the contract

Let’s outline the core structure of the contract:

// SPDX-License-Identifier: MIT

Pragma solidity ^0.8.18;

Contract FundMe {}

**Fund and withdraw functions:**

The FundMe contract will have two primary functions that serve as the main interaction points:

1. Fund: allows users to deposit funds into the contract
2. withdraw: grants the contract owner the ability to withdraw the funds that have been previously deposited

First, let’s code the `fund` function and leave the `withdraw` function commented out for the moment.

Contract FundMe {

// send funds into our contract

Function fund() public {}

// owner can withdraw funds

/\*function withdraw() public {}\*/

}

**Conclusion:**

In this lesson, we created a new `FundMe` contract and broadly defined the logic that will be performed.