ERC20

- Manual Implementation
- OpenZeppelin implementation

## Quickstart

```
git clone https://github.com/PatrickAlphaC/foundry-erc20-cu
cd foundry-erc20-cu
forge install
forge build
```

## Installing OpenZeppelin Contracts Package

```
forge install OpenZeppelin/openzeppelin-contracts --no-commit
```

## Start a local node

```
make anvil
```

## Deploy

This will default to your local node. You need to have it running in another terminal in order for it to deploy.

```
make deploy
```

## Testing

```
forge test
```

or

```
forge test --fork-url $SEPOLIA_RPC_URL
```

## Deployment to a testnet or mainnet

1. Setup environment variables

You'll want to set your `SEPOLIA_RPC_URL` and `PRIVATE_KEY` as environment variables. You can add them to a `.env` file, similar to what you see in `.env.example`.

- `PRIVATE_KEY`: The private key of your account (like from metamask). **NOTE:** FOR DEVELOPMENT, PLEASE USE A KEY THAT DOESN'T HAVE ANY REAL FUNDS ASSOCIATED WITH IT.
  - You can learn how to export it here.
- `SEPOLIA_RPC_URL`: This is url of the sepolia testnet node you're working with. You can get setup with one for free from Alchemy

Optionally, add your `ETHERSCAN_API_KEY` if you want to verify your contract on Etherscan.

1. Get testnet ETH

Head over to faucets.chain.link and get some testnet ETH. You should see the ETH show up in your metamask.

2. Deploy

```
make deploy ARGS="--network sepolia"
```

## Scripts

After deploy to a testnet or local net, you can run the scripts.

Using cast deployed locally example:

```
cast send <ERC20_CONTRACT_ADDRESS> "transfer()"  --value 0.1ether --
private-key <PRIVATE_KEY> --rpc-url $SEPOLIA_RPC_URL
```

or, to create a ChainlinkVRF Subscription:

```
make createSubscription ARGS="--network sepolia"
```