# Contracts [![Coverage Status](https://coveralls.io/repos/github/paulrberg/contracts/badge.svg?branch=main)](https://coveralls.io/github/paulrberg/contracts?branch=main) [![Styled with Prettier](https://img.shields.io/badge/code\_style-prettier-ff69b4.svg)](https://prettier.io) [![Commitizen Friendly](https://img.shields.io/badge/commitizen-friendly-brightgreen.svg)](http://commitizen.github.io/cz-cli/) [![License: WTFPL](https://img.shields.io/badge/License-WTFPL-yellow.svg)](https://spdx.org/licenses/WTFPL.html)

\*\*Off-the-shelf Solidity smart contracts.\*\* Built with paulrbergs amazing design The beloved [Solidity template](https://github.com/PaulRBerg/solidity-template).

- Designed for Solidity >=0.8.4

- Complementary to [OpenZeppelin Contracts](https://github.com/OpenZeppelin/openzeppelin-contracts)

- Promotes [PRBMath](https://github.com/hifi-finance/prb-math) as a fixed-point math library for Solidity

- Well-documented via NatSpec comments

- Thoroughly tested with Hardhat and Waffle

“I created this library for my own use, to avoid having to maintain the same contracts in different repositories. If you find

it useful too, it's a win for both of us”

## Caveat Emptor

This is experimental software and is provided on an "as is" and "as available" basis. I do not give any warranties and will not be liable for any loss, direct or indirect through continued use of this codebase.

## Installation

With yarn:

```sh

$ yarn add @mattytaariq/ZeeCoinZCN\_Contracts

```

Or npm:

```sh

npm install @mattytaariq/ZeeCoinZCN\_Contracts

```

I adhere to [semver](https://semver.org/), which means that your contracts won't break unexpectedly when upgrading to a

newer minor version of `@paulrberg/contracts`

## Usage

Once installed, you can use the contracts like this:

```solidity

pragma solidity >=0.8.4;

import "@mattytaariq/ZeeCoinZCN\_Contracts/math/PRBMathUD60x18.sol";

import "@mattytaariq/ZeeCoinZCN\_Contracts/token/erc20/Erc20.sol";

import "@mattytaariq/ZeeCoinZCN\_Contracts/token/erc20/Erc20Permit.sol";

contract MyToken is Erc20, Erc20Permit {

using PRBMathUD60x18 for uint256;

constructor(

string memory name\_,

string memory symbol\_,

uint8 decimals\_

) Erc20Permit(name\_, symbol\_, decimals\_) {}

}

```

## Contributing

### Pre Requisites

Before running any command, make sure to install dependencies:

```sh

$ yarn install

```

### Compile

Compile the smart contracts with Hardhat:

```sh

$ yarn compile

```

### TypeChain

Compile the smart contracts and generate TypeChain artifacts:

```sh

$ yarn typechain

```

### Lint Solidity

Lint the Solidity code:

```sh

$ yarn lint:sol

```

### Lint TypeScript

Lint the TypeScript code:

```sh

$ yarn lint:ts

```

### Test

Run the Mocha tests:

```sh

$ yarn test

```

### Coverage

Generate the code coverage report:

```sh

$ yarn coverage

```

### Clean

Delete the smart contract artifacts, the coverage reports and the Hardhat cache:

```sh

$ yarn clean

```

## Security

While I set a high bar for code quality and test coverage, you shouldn't assume that this library is completely safe to use. The contracts

have not yet been audited by a security researcher. If you discover any security issues, please report them via [Keybase](https://keybase.io/paulrberg).

## Acknowledgements

I am grateful to the authors of existing related projects:

- [OpenZeppelin Contracts](https://github.com/OpenZeppelin/openzeppelin-contracts)

- Alberto Cuesta Cañada's [Erc20Permit and Orchestrated](https://github.com/albertocuestacanada)

- Paul Berg [https://github.com/paulrberg/contracts)

## License

The contracts are released under the [WTFPL License](./LICENSE.md).