



# StarkNet.js 101

---

Sean, in collaboration with

✦ March 2022



# How did I get involved with StarkNet?

- Looking for a long term ETH scaling solution
- Wanted to build a web application
- No library for web applications to interface with StarkNet

# What is StarkNet.js?

- Allow JavaScript programs, like Web and Mobile Apps to interface with the StarkNet network
- Follows ethers.js architecture in Ethereum
- Primary APIs are Providers, Account, Signer, and Contract

# Starknet.js V3 API - Overview

- Provider
- Account
- Signer
- Contract

# Provider

## API

The Provider API allows you to interface with the StarkNet network, without signing transactions or messages.

Typically, these are *read* calls on the blockchain.

# Account

## API

An Account extends [Provider](#) and inherits all of its methods.

It also introduces new methods that allow Accounts to create and verify signatures with a custom [Signer](#).

This API is the primary way to interact with an account contract on StarkNet.

# Signer

API

The Signer API allows you to sign transactions and messages, and also allows you to get the public key.

# Contract

## API

Contract API can do data transformations in JavaScript based on an ABI. They can also call and invoke to StarkNet through a provided Signer.

Contracts allow you to transform Cairo values, like Uint256 to BigNumber. It could also allow users to pass their own transformers, similar to JSON.parse.



# Account x ERC20

---

## Example Code & Demo

- <https://github.com/0xs34n/starknet.js-workshop>

# Starknet.js V3 Release

- Starknet.js V3 will be released tomorrow with a new API
  - Try it today with `npm install starknet@next`
- New docs to be released at [starknet.js.com](https://starknet.js.com)! Examples today will be included
- Special thanks to Janek of Argent, Miljan of Shard Labs, and Dhruv of Jedi Swap and other Open Source Contributors to Starknet.js

# Thank You

◆ Questions?

