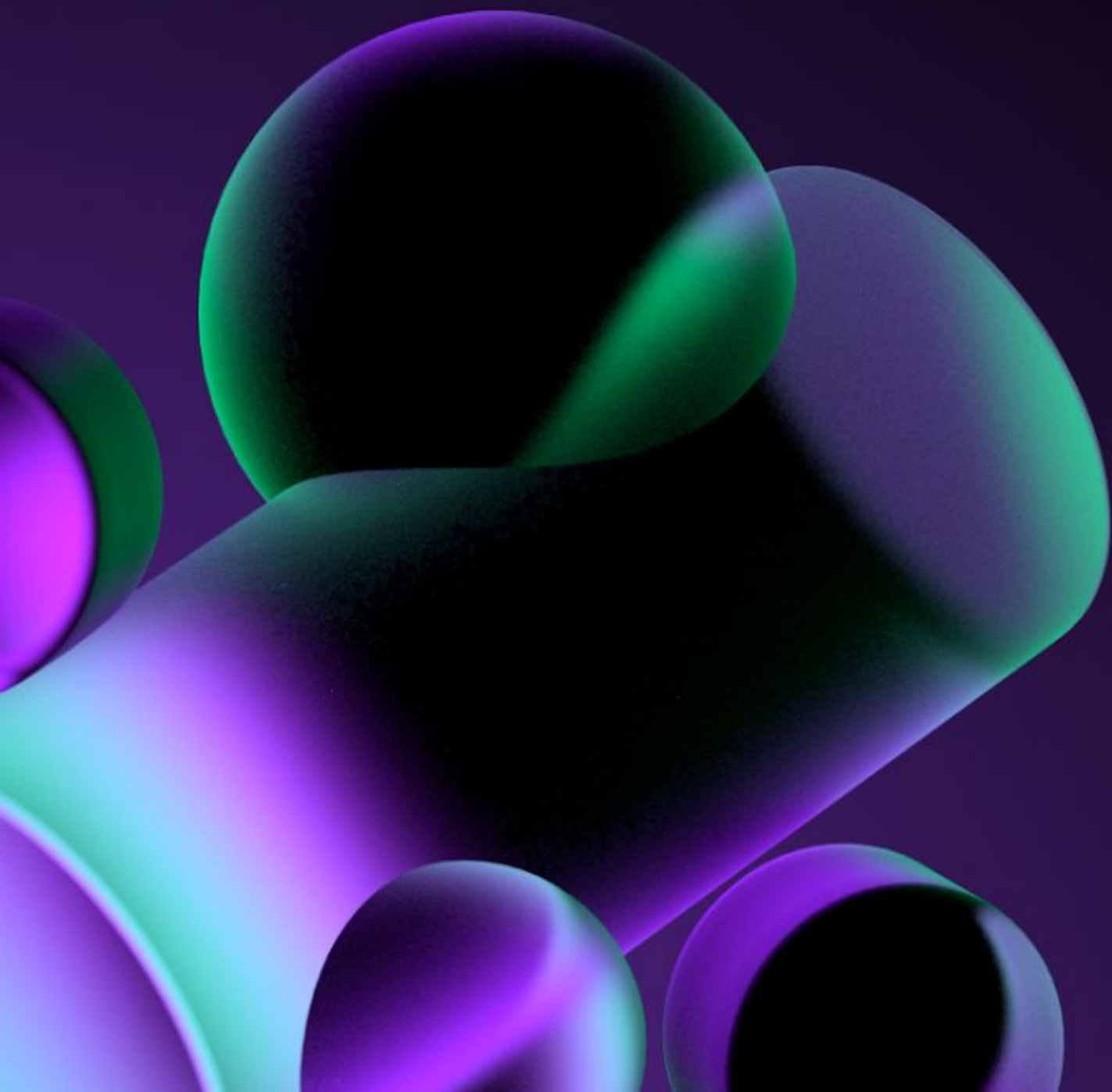


# Summer School of Solana

## LECTURE 6

Introduction to Anchor framework

# About this lecture



# About this lecture

- **Introduction to Anchor framework**
  - Turnstile program (in Anchor)
- **Task assignment**





# Anchor framework



Armani Ferrante 🦎🚢  
@armaniferrante

For every new technology, the pace of innovation exceeds the pace of education. If there's tons of examples to copy and paste, you're late. If you're confused because there's no docs, good. You've discovered a secret that has yet to be revealed to the rest of the world.

9:31 AM · Sep 5, 2021 · Twitter Web App

# Anchor

- Developed by [Armani Ferrante](#)
  - Open-source
  - Writing Vanilla Solana is pain
  - One stop shop for Solana development



# Anchor

- Framework for Solana's Sealevel runtime
  - Build secure Solana programs
  - Rust eDSL for writing Solana programs
  - IDL specification
  - Generating clients from IDL
  - CLI and workspace management
- Familiar to Truffle, Web3.js

[Anchor Book](#)





## Anchor basics

```
// use this import to gain access to common anchor features
use anchor_lang::prelude::*;

// declare an id for your program
declare_id!("Fg6PaFpoGXkYsidMpWTK6W2BeZ7FEfcYkg476zPFsLnS");

// write your business logic here
#[program]
mod hello_anchor {
    use super::*;
    pub fn initialize(_ctx: Context<Initialize>) -> Result<()> {
        Ok(())
    }
}

// validate incoming accounts here
#[derive(Accounts)]
pub struct Initialize {}
```





```
use anchor_lang::prelude::*;

declare_id!("Fg6PaFpoGXkYsidMpWTK6W2BeZ7FEfcYkg476zPFsLnS");

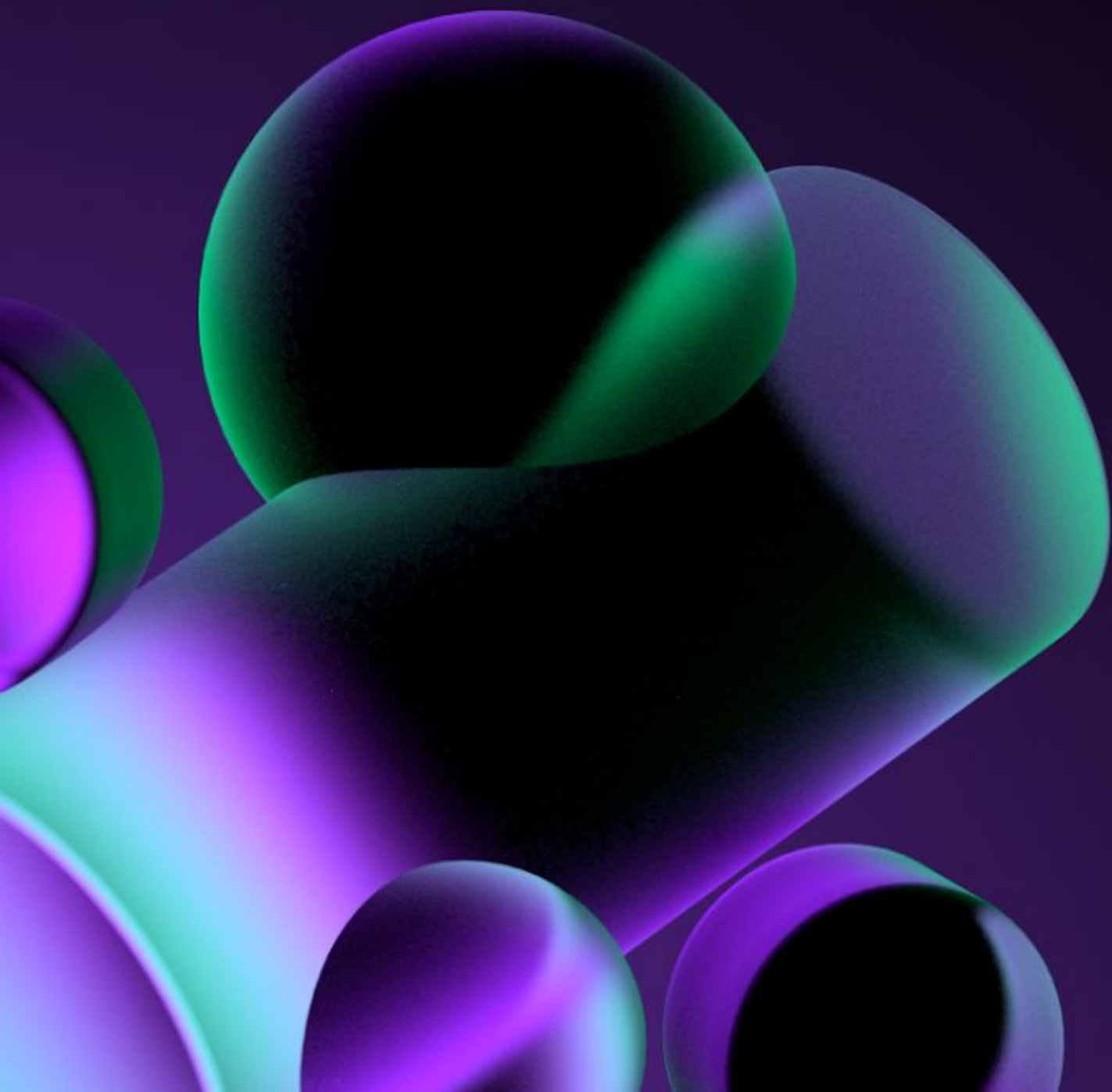
#[program]
mod hello_anchor {
    use super::*;

    pub fn set_data(ctx: Context<SetData>, data: u64) -> Result<()> {
        ctx.accounts.my_account.data = data;
        Ok(())
    }
}

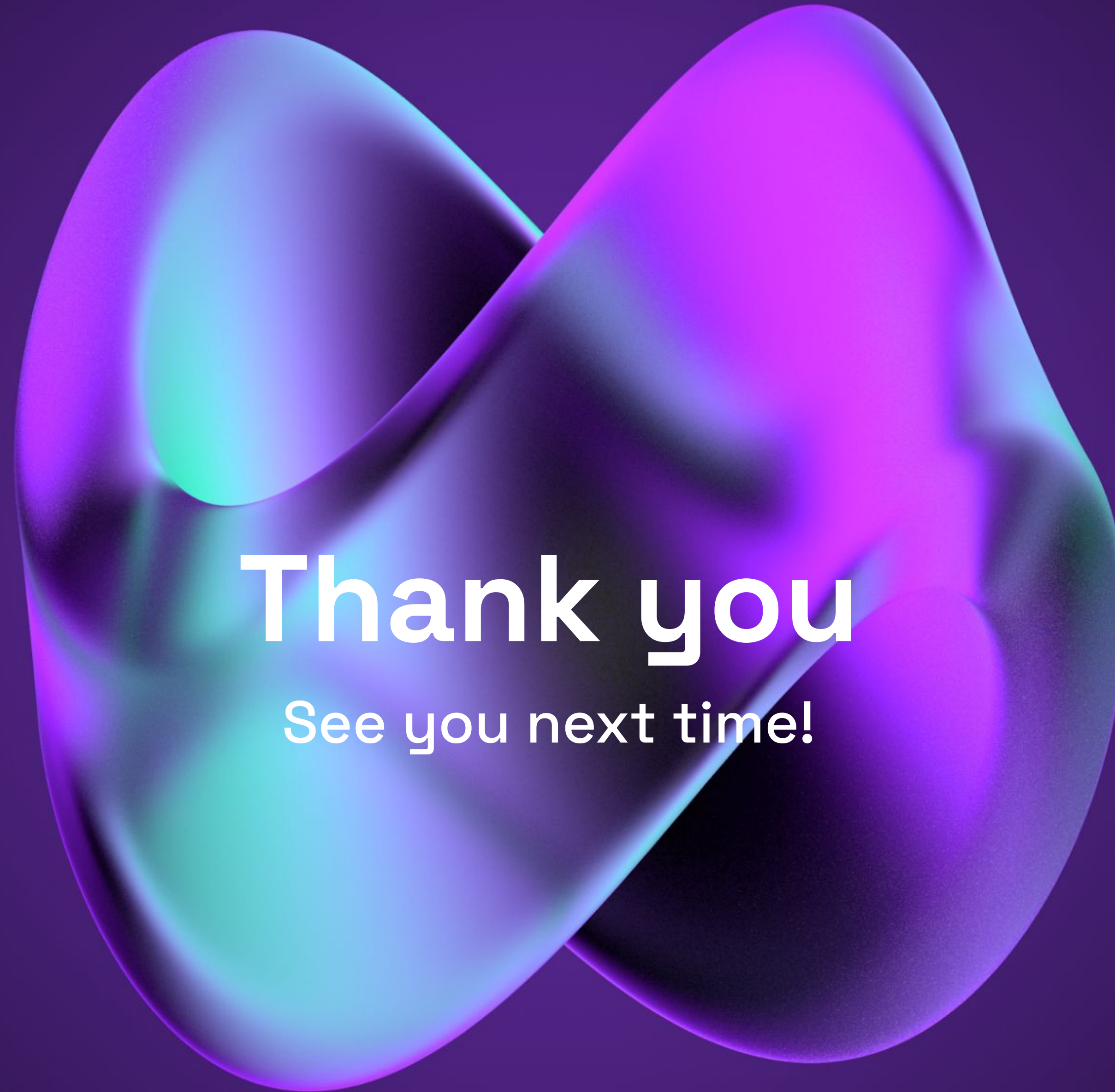
#[account]
#[derive(Default)]
pub struct MyAccount {
    data: u64
}

#[derive(Accounts)]
pub struct SetData<'info> {
    #[account(mut)]
    pub my_account: Account<'info, MyAccount>
}
```

**Hands on example**







# Thank you

See you next time!