Background Assignment

> 💡 If you haven't yet, please join our discord channel.

This assignment will help students verify that they have the necessary background for ZKU (the next cohort to start on May 2, 2022). It will focus on solidity programming and frontend Javascript. The links within this document should be sufficient for someone with substantial coding experience to pick up the relevant background *within a few weeks of intensive self-studies* ← start this process as soon as now, if you're interested in taking the course but lack actual experience in web3 building.

This is due by **May 2, 2022, 11:59 pm UTC** (first week of class). To secure enrollment, late submission is acceptable till ***May 4, 11:59 pm UTC*** (48 hours after the original deadline).

> 💡 Submission goes here
> Upload all code to GitHub or Gist and add the links to a pdf file.

A. Conceptual Knowledge

Before programming, it is important to know these main concepts. You should be able to explain these concepts to a (smart) five-year-old.

Reference this video for more information.

1. What is a smart contract? How are they deployed? You should be able to describe how a smart contract is deployed and the necessary steps.

2. What is gas? Why is gas optimization such a big focus when building smart contracts?
3. What is a hash? Why do people use hashing to hide information?
4. How would you prove to a colorblind person that two different colored objects are actually of different colors? You could check out Avi Wigderson talk about a similar problem here.

Provide the answers to these questions in your submission.

Solidity Tutorials

Best to first skim the Solidity Docs and then go through the Solidity by Example. We will ask you a few questions about this.

To get more instruction on programming with solidity, the Smart Contract Programmer YouTube channel is a great resource of tutorials for programming with solidity. If you have no experience with web3 programming, watching these videos is a great place to start. Specifically, their Solidity 0.8 tutorials. Another great introductory course is Crypto Zombies.

Other things to know about and learn how to use are

- remix, web ide for writing and deploying smart contracts. Check out this tutorial.
- hardhat or truffle, your choice.
    - hardhat: Great tool for deploying and testing smart contracts. Check out this tutorial. Covers everything from setting up the environment to deploying and testing the contract.
    - truffle/ganache: When used together they provide similar functionality to hardhat. Check out this tutorial.

You should now be able to answer this question:

B. You sure you're solid with Solidity?

1. Program a super simple "Hello World" smart contract: write a `storeNumber` function to store an unsigned integer and then a `retrieveNumber` function to retrieve it. Clearly comment your code. Once completed, deploy the smart contract on remix. Push the .sol file to Github or Gist and include a screenshot of the Remix UI once

deployed in your final submission pdf.

2. On the documentation page, the “Ballot” contract demonstrates a lot of features on Solidity. Read through the script and try to understand what each line of code is doing.
3. Suppose we want to limit the voting period of each Ballot contract to **5 minutes**. To do so, implement the following: Add a state variable `startTime` to record the voting start time. Create a modifier `voteEnded` that will check if the voting period is over. Use that modifier in the `vote` function to forbid voting and revert the transaction after the deadline.
4. Deploy your amended script and test the newly implemented functionality in part 3. Submit (1) your amended version of the contract on Github or Gist and (2) screenshots showing the time of contract deployment as well as the transaction being reverted once past the voting period.

   Note: *All code needs to be submitted either in pdf (only code snippets where ever required) or via Github/Gist links (copy the repo/pull-request links and the commit in the pdf).*

Frontend web3 Tutorials

The most important thing developers need to understand to develop on web3 is how to interact with a smart contract as the backend. This tutorial covers that.

Also, check out how to **build a NFT mint**.

Additional Tutorials

Check out the token faucet demo dapp on Harmony.

Harmony offered this repo ▸, which might be helpful when building both frontend and backend.