**Ethereum advanced:**

**GAS:**

* Solidity is turing complete language which allows you to have loops, taket his code and execute it many times. The ethereum network has a limited amount of CPU power and storage because of the PCs that make up the network obviusly dosnt have unlimited storage and CPU power.
* Therefore you have limited resources and an unlimited coding language.
* In order to prevent spam on the ethereum network you have to pay for each transaction your smart contract executes. When you invoke a smart contract the gas fees for each instruction is already hard coded into the protocol. Ann isntruction could for example be add multiply and so on, add costs 3 gas and multiply costs 5 gas. These prices are hard coded into the protocol. Therefore you cant have the price in ether because ether price will change and therefore would have affected the price of the instructions.
* Gas needs to be converted into ether because you will be paying in ether when you are executing smart contracts. 1 gas will vary in cost depending on the price of ether so that something dosnt become to expensive.

**Token standars and NFTs:**

* Most tokens are smart contracts built upon the ethereum network, and represent their own cryptocurrency. Ethereum should be considered a coin while the rest of the currencies should be called tokens.
* ERC20 standards are used for these tokens. Which means that all exchanges and all wallets will be able to support all the ERC20 tokens, because they already support ERC20.
* You have different type of tokens:

Fungible tokens: All tokens are the same, we make no difference between different tokens and have the same value. ERC20 is the standard of fungible tokens.

Non Fungible tokens:

NFTs are mostly used in games, in games you have for example different items that are all unique to themself. ERC721 or 1155 are the standards for NFTs.

* We can create new tokens, new currencies and so on.

**Web 3.0 and token economy:**

* In web 2.0 with google and facebook, the value was not captured in the https invention, but it was in the applications built on top of it like google and facebook.
* In blockchain it is a bit different. Ethereum form example there are many tokens relying on ethereum and ethereum is on the top, most of the value is captured in ethereum and not the applications built on top like it was in web 2.0
* The smart contract is a smaller piece of code running on the ethereum blockchain, while the dap is more the entire experience. The smart contract might be connected to a website and a server connected to the website aswell and the entire thing would be called a dap.
* Web 3.0 where you have decentralized websites and daps bbeing developped.
* On ethereum you have the ether main token, and when any of the other tokens want to execute smart contracts they need ether to pay. Therefore the value of ethereum grows as the other tokens built on ethereum is. Ijndirectly investing in all of the other things being built on otp of ethereum.
* Steemit social network have their own token steem and the whole idea is that when you contribute to the network by posting and you tell the network what is good by liking posts. Usually facebook for example has captured all this value, but steemit makes it so the poster gains the value and this would never be possible without cryptocurrency because it is base don inflation.
* With web 3.0 that cna have their own cryptocurrency and economijc model we have a whole other dynamic on the internet because form example ssmall startups will be able to compete with larger companies. Because usually you dont want to be an early adopter, but with cryptocurrency this is much more attractive because you can earn value in their token, and if the comapny grows then you gain more money aswell because of the tokens you are holding. Incetiveses people to be an early adopter because you earn more money as the token grows.

Homework:

ERC20:

1. The benefits of setting a token standard is that it creates less friction in the system. This is because all the different comands in solidity have the same notation in every single token built upon ERC20. This results in wallets and exchanges being able to communicate with all of the different ERC20 tokens as soon as they are added because they have the same way of writing the codes.
2. A few examples of functions in the ERC20 token stanards are:

totalSupply() gives the maximum amount of tokens that exist

balanceOf(account adress) gives a public adresses balance of ERC20 tokens

Web 3.0 and tokens:

* + Benefits of web 3.0 is that it is a lot more attractive to be an early adopter of for example a small startup company. This is because the value of their token at an early stage is very low which means that you can get a lot of the tokens for a smaller price. When the company grows the value of token grows and you earn alot more money aswell.
  + A token is a digital asset/cryptocurrency that can either be fungible or non-fungible. And it exists on a blockchain network. For example Ethereum is a coin and dogecoin is a token built upon the ethereum network.
  + You create a token on ethereum by coding smart contracts and then connecting the smart contracs to a token adress and then deploying these smart contracts onto the ethereum blockchain. The code follows either the ERC20 (fungible token) standard or ERC721 and ERC1155 (NFT)