**Guide d'étude du chapitre 2**

Pour les personnes intéressées par les groupes d'étude, nous fournissons quelques ressources que vous pouvez parcourir en groupe. Ce sont juste des suggestions pour vous aider à démarrer et vous n'êtes pas obligé de les suivre.

Veuillez noter que certains de ces exercices peuvent nécessiter plus d'une séance ! Ne vous sentez pas obligé de les terminer en une seule séance. Il peut être utile de commencer à les travailler séparément, puis de vous réunir lorsque vous êtes tous au même point d'arrêt. Vous pouvez partager ce que vous avez vécu, les défis que vous avez rencontrés, et aider ceux qui sont encore en train de travailler sur les exercices.

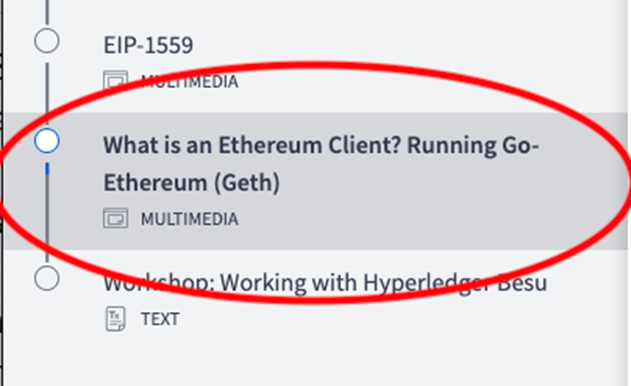
Nous examinerons également ces exemples / tutoriels lors des sessions hebdomadaires. Nous encouragerons les gens à partager leur travail. Vous pouvez également en discuter dans les canaux Discord respectifs. N'oubliez pas : soyez gentil avec les membres de votre groupe !

## **1. Débutant : Configuration du Geth et / ou Hyperledger Besu:**

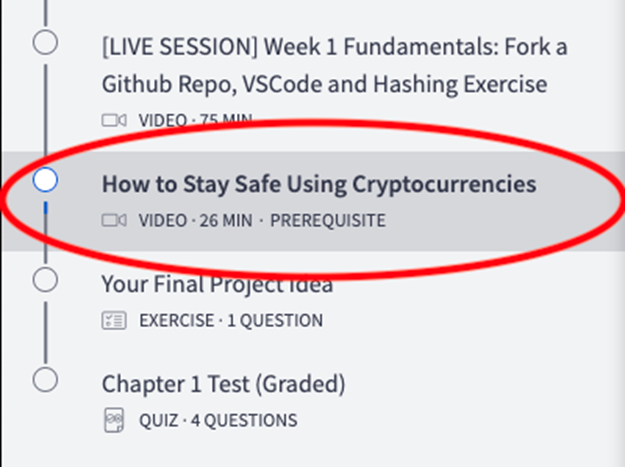
Pour la session de cette semaine, nous abordons d'autres aspects des principes fondamentaux de la cryptographie, mais nous pivotons également pour discuter le protocole Ethereum. Au cours de la présentation du vendredi et de la démonstration de la semaine prochaine, nous allons parler des clients Ethereum.

La création de l'environnement sur votre machine locale pour un client Ethereum a quelques inconvénients, il est donc bon de commencer à l'avance. De plus, cela vous aidera à vous préparer pour la présentation de vendredi où Tom vous guidera dans la mise en place de Hyperledger Besu, le démarrage d'un testnet local et l'intégration de MetaMask.

Pour configurer Geth, veuillez suivre les instructions de cette leçon au chapitre 2, " Qu'est-ce qu'un client Ethereum ? Exécution de Go-Ethereum (geth)" :



Si vous ne pouvez pas accéder à cette leçon, c'est très probablement parce que vous n'avez pas terminé cette leçon du chapitre 1, "Comment rester en sécurité en utilisant les crypto-monnaies", ce que vous devez faire immédiatement :



Vous pouvez faire quelques préparatifs pour la présentation de vendredi en installant [Hyperledger Besu](https://besu.hyperledger.org/en/stable/HowTo/Get-Started/Installation-Options/Install-Binaries/). L'exécution de Besu à partir d'un binaire nécessite Java JDK 11+ pour fonctionner, ce qui peut être un peu délicat. [Vous pouvez l'installer à partir d'ici](https://www.oracle.com/java/technologies/javase-downloads.html).

**2. Technique : Exercice sur** [**les transactions du Bitcoin**](https://github.com/cooganb/bitcoin-whitepaper-exercises/blob/master/transactions/README.md) **et** [**les portefeuilles**](https://github.com/cooganb/bitcoin-whitepaper-exercises/blob/master/wallet/README.md)

Si vous ne l'avez pas encore fait, [clonez ce repo des exercices sur les whitepapers de Bitcoin](https://github.com/cooganb/bitcoin-whitepaper-exercises) et effectuez l'exercice de hachage. Si vous avez besoin d'aide, vous pouvez regarder l'enregistrement du chapitre 1 : [[SESSION EN DIRECT] Semaine 1 - Principes fondamentaux : Forker un repo Github, VSCode et exercice de hachage.](https://courses.consensys.net/courses/take/blockchain-developer-bootcamp-registration-2021/lessons/27760900-live-session-week-1-fundamentals-fork-a-github-repo-vscode-and-hashing-exercise)

**Lors de notre session de mercredi, nous passerons en revue la fonction verifyBlock(...) de l'exercice de hachage, puis nous passerons à la création de signatures digitales à l'aide de la bibliothèque OpenPGP. Veuillez noter que nous utilisons une version non sécurisée d'OpenPGP (3.0.8 vs 5.0.0), ceci est uniquement destiné à des fins éducatives !**

**3. Avancé : commencez à explorer d'autres clients Ethereum**

Outre Hyperledger Besu et Geth, il existe d'autres clients Ethereum. Si vous souhaitez en savoir plus, vous pouvez explorer ces deux clients et leur documentation d'installation :

.

● [Erigon](https://github.com/ledgerwatch/erigon) Précédemment appelé « Turbogeth », Erigon est récemment devenu officiellement son propre client.

● [OpenEthereum AKA Parity](https://docs.nethereum.com/en/latest/ethereum-and-clients/parity/) Ce client a été déprécié en raison du fait que ses constructeurs d'origine ([Parity Technologies](https://www.parity.io/)) se sont davantage intéressés au développement de Polkadot. Cependant, il pourrait être intéressant de fouiller dans le repo ! Écrit en Rust.ntation :

Quelques autres choses intéressantes liées au client Ethereum :

● [Nethereum](https://docs.nethereum.com/en/latest/getting-started/) Une bibliothèque écrite en C# pour "simplifier la gestion des contrats intelligents et l'interaction avec les nœuds Ethereum".

● [Geth Snap Sync](https://blog.ethereum.org/2021/03/03/geth-v1-10-0/) Désormais intégré de manière native à Geth, vous pouvez en savoir plus sur Snap sync [ici](https://github.com/ethereum/devp2p/blob/master/caps/snap.md).

Il y a un canal animé dans le discord appelé #🏃node-runners avec des gens qui sont également intéressés à exécuter leurs propres nœuds Ethereum si vous voulez rejoindre la conversation là-bas ou comparer des notes sur les clients Ethereum !