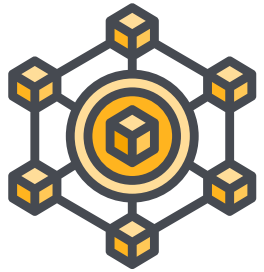
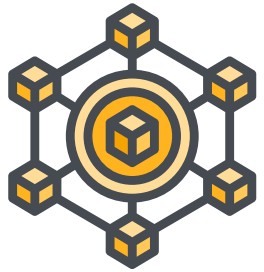
⛽ What is Gas?

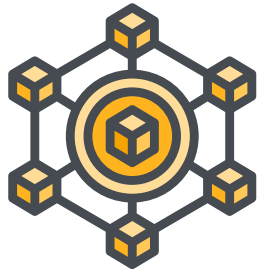
- Gas is the fee paid to execute smart contracts
- Every operation in Solidity costs gas
- More computation = more gas = more ETH



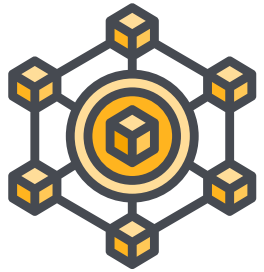
⚡ What is Gas Optimization?



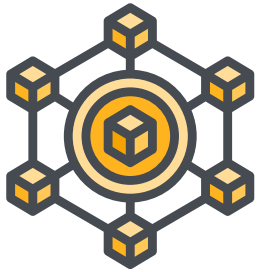
- Writing smart contracts that use less gas
- Same result, lower cost
- Saves money for users and developers



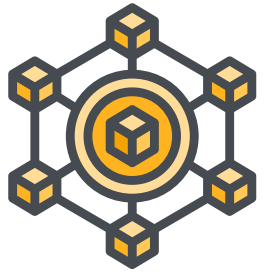
🧠 Storage vs Memory vs Calldata



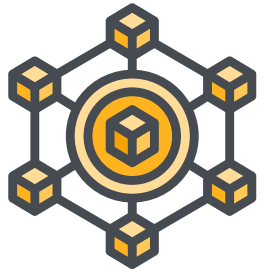
- Storage → Permanent data (❌ very expensive)
- Memory → Temporary data inside functions (✅ cheaper)
- Calldata → Read-only function inputs (✅ cheapest)
- Use memory/calldata instead of storage to save gas



📦 Why Keep State Variables Minimum?

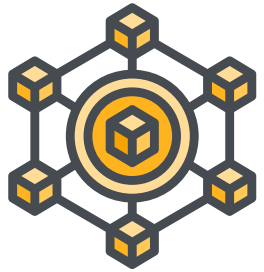


- State variables are stored on blockchain
- More variables = more storage cost
- Unused variables = wasted gas

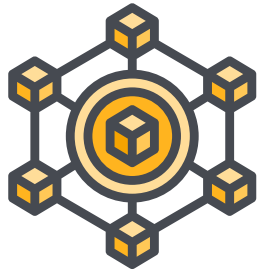


🔁 Loops = Danger 🚨

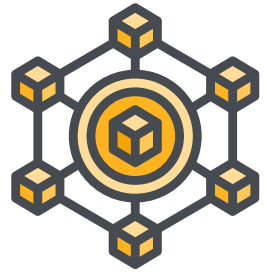
(Why Mapping is Better?)



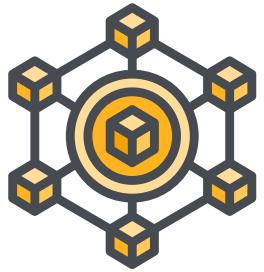
- Loops consume gas for every iteration
- Large loops can fail due to gas limit
- Mappings are faster and cheaper than arrays
- Avoid looping on-chain when possible



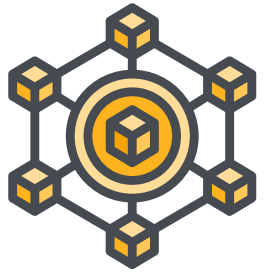
🛑 require vs revert



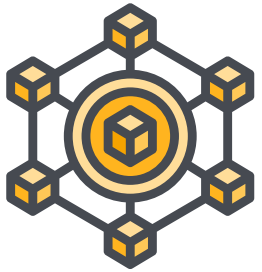
- require() is used for input validation
- revert() is used for custom errors
- require() is simpler and cheaper for basic checks
- Always fail early to save gas



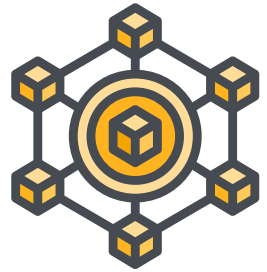
🔢 Short Data Types



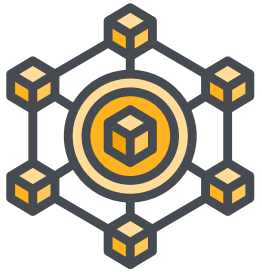
- uint256 is default but not always needed
- Smaller types (uint8, uint16) save space
- Multiple small variables can fit in one storage slot



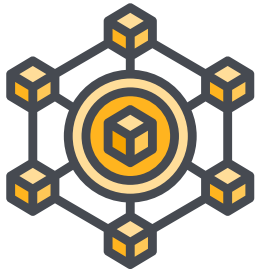
📣 Events vs Storage



- Storage → Expensive, permanent
- Events → Cheap, used for logs
- Use events when data is only for tracking
- Don’t store data if you only need to read it off-chain



🚀 When Does Gas Optimization Matter?



- When contract has many users
- DeFi, NFTs, Games, DAOs
- Small gas savings = big money at scale