Beginner-Friendly Smart Contracts — Full Pack (Testnet-Ready)

This document consolidates beginner-friendly smart contract examples for learning Solidity. Contracts are grouped into two categories: (1) General / Non-Payment and (2) Crypto & Payments. All examples are simplified and intended for testnets only.

Quick Setup

1) Install MetaMask and connect to a testnet (Sepolia, Holesky, or Polygon Amoy). 2) Get test ETH from a faucet. 3) Open Remix (https://remix.ethereum.org), create a new .sol file, paste code, compile with Solidity ^0.8.20, and deploy using Injected Provider - MetaMask. 4) Interact with deployed contracts via Remix. Track transactions on the testnet explorer.

General (Non-Payment) Examples

1) Hello World

Use-case: Minimal contract storing a string.
Teaches: State variables, getters/setters.
Test: Deploy with "Hello", call setMessage("Hola").

2) To-Do List

Use-case: Each user adds tasks and marks them complete.
Teaches: Structs, arrays, mappings.
Test: Add tasks, call tasks() to view, toggle() to mark done.

3) Simple Voting

Use-case: Create a proposal, users vote yes/no.
Teaches: Mappings, preventing double votes, counters.
Test: Deploy with proposal, multiple accounts vote, read tallies.

4) Minimal ERC-20 Token

Use-case: Deploy your own token and transfer between users.
Teaches: Balances, transfer, events.
Test: Deploy with initial supply, transfer tokens, read balances.

5) Minimal NFT (ERC-721-like)

Use-case: Mint unique IDs representing NFTs.
Teaches: Token IDs, ownership, transfer logic.

Test: Mint token, transfer to another account, verify owner.

Crypto & Payments Examples

1) Payment Splitter (ERC-20)

Split received tokens among team wallets. Test: send tokens to contract, call pay(totalAmount).

2) Pull Subscriptions (ERC-20)

Recurring allowances approved by user; merchant charges each period.

3) Escrow with Arbiter + Timeout (ERC-20)

Buyer funds escrow; release to seller or refund after deadline.

4) Tip Jar (ETH) with Messages

Fans send ETH with optional note; owner withdraws later.

5) Merchant Paylink (ERC-20)

Simple 'pay this amount with memo' function.

6) Fee-Taking Forwarder (ETH)

Forward ETH to seller while taking small fee to treasury.

7) Token Streaming (ERC-20)

Continuous payments recipients withdraw over time.

8) Request-to-Pay (ERC-20)

Sender requests payment; recipient accepts or rejects.

Deployment & Testing Notes

- For ERC-20 contracts, remember to approve(contract, amount) before transferFrom calls. - Events provide an off-chain audit trail (for UIs, receipts, or analytics). - These examples omit advanced security and production hardening (e.g., reentrancy guards). - Use test stablecoins or your own ERC-20 to simulate real tokens.

Next Steps

- Wrap Merchant Paylink or Request-to-Pay in a small frontend with QR codes. - Extend ERC-20/NFTs with metadata and access control. - Index contract events with TheGraph or scripts for histories. - Practice modifying contracts (e.g., add events, restrict access) to deepen understanding.