

Solidity Constructs

READING TASK | 7

## Last Updated: January 11, 2022

# Solidity Constructs

# Solidity Constructs

* Introduction to functions
  + <https://docs.soliditylang.org/en/v0.8.11/control-structures.html#function-calls>
* Ether, Time unit conversions & Global variables
  + The global variables and objects available in the EVM contract
  + Throwing exception using the throw keyword
    - <https://docs.soliditylang.org/en/v0.8.11/units-and-global-variables.html>
* Hacker proofing + View & Pure Function Modifier
  + Recent hacking attacks & why hackers were successful
    - <https://consensys.github.io/smart-contract-best-practices/known_attacks/>
  + Best practices to avoid contract vulnerabilities
  + view and pure modifiers
  + Constants, Fallback functions & Payable
  + Variable and Function visibility
    - private & public
    - internal & external
      * <https://docs.soliditylang.org/en/v0.5.0/contracts.html#function-modifiers>
    - <https://www.c-sharpcorner.com/article/address-and-function-in-solidity>
    - <https://docs.soliditylang.org/en/v0.5.0/contracts.html#function-modifiers>
    - <https://docs.soliditylang.org/en/v0.4.21/control-structures.html>
* Error Handling in Contracts
  + revert
  + require
  + Assert
    - <https://docs.soliditylang.org/en/v0.8.11/control-structures.html#error-handling-assert-require-revert-and-exceptions>
* Mappings, Enumerations, and Structs
  + Mapping
    - <https://docs.soliditylang.org/en/v0.5.0/types.html#mappings>
  + Enum
    - Enums: The word Enum in Solidity stands for Enumerable. They are user-defined types that contain human-readable names for a set of constants, called members. The data representation for an enum is the same as the one in the C language.
    - <https://jeancvllr.medium.com/solidity-tutorial-all-about-enums-684adcc0b38e>
  + Struct
    - <https://docs.soliditylang.org/en/v0.5.0/types.html#structs>
* Object Orientation
  + Overloading
  + Inheritance, Abstract contracts, Interfaces
  + Polymorphism
    - <https://docs.soliditylang.org/en/v0.5.0/contracts.html?highlight=interface#interfaces>
    - <https://docs.soliditylang.org/en/v0.5.0/contracts.html#inheritance>
    - <https://docs.soliditylang.org/en/v0.5.0/contracts.html?highlight=fallback#function-overloading>
    - <https://docs.soliditylang.org/en/v0.5.0/contracts.html#abstract-contracts>
    - <https://docs.soliditylang.org/en/v0.5.0/contracts.html#libraries>
* Function modifiers
  + Modifiers are code that can be run before and/or after a function call. Modifiers can be used to:
    - Restrict access
    - Validate inputs
    - Guard against reentrancy hack
    - <https://medium.com/coinmonks/solidity-tutorial-all-about-modifiers-a86cf81c14cb>
    - Code Example: <https://solidity-by-example.org/function-modifier/>
* Events
  + Solidity defines events with the event keyword. After events are called, their arguments are placed in blockchain known as logs. logs are related to the contract address. They stay in the blockchain as long as blocks are accessible. <https://www.bitdegree.org/learn/solidity-events>
* Moreover, You can check the below link and practices the programs for all above concepts:
  + <https://learning-dcs-bbn.netlify.app/part%2005-module%2001-lesson%2002_new%20introduction%20to%20solidity%20and%20creating%20a%20token/01.%20let's%20write%20smart%20contracts>