Solidity Hack 101 - Level 1

State Variables:- Are those variables which are define gloablly

```
contract function{
    unit value=1;
    function value(){
        return value;
    }
}
```

Local Variables:- are those variables which are store on function

For Example:-

```
function getvalue() public {
    uint value=1;
    return value;
}
```

Data-Types:

uint:- Usigned Integers, only Positive Numbers

uint256:- Usigned Integers which stored 256 bytes in memory, 256 is default number

uint8:- same as 256 but In this we stored 8 byte in memory

string:- Same as C++

byte32:- It will store your string value as format of byte32 basicallt array and other functionality -- Byte arrays, declared with the keyword "bytes", is a fixed-size array used to store a predefined number of bytes up to 32, usually declared along with the keyword (bytes1, bytes2).

address:- Address variable contain your username because ethereum or blockchain network have address. (20-Byte)

```
struct Mystruct {
    uint256 value,
    string mystring,
    :- Similar as C++, This will help us to build own data types

}
```

Use:-

`Mystruct public struct = Mystruct(1."Hello World");`

```
function getvalue() public pure returns(uint){
    uint value=1;
    return value;
}
```

push(_value) Basically This push function use to add value of end of the array

Arrary:-

```
uint count; //  1,2,3,4 not negative value

//Arrays
uint[] public uniarry = [1,2,3];

string[] public stringArrary = ["Apple","banana","Carrot"];
```

`string[] public value;` uint256[][] public arry2d =1,2,3],[3,4,5,6

Theory Of Data Types:-

Solidity has eight value types: signed/unsigned integers, booleans, fixed point numbers, addresses, byte arrays, literals, enums, and contract and function types. Here’s a quick cheatsheet of what they stand for:

Signed/Unsigned integers - Integer data types store whole numbers, with signed integers storing both positive and negative values and unsigned integers storing non-negative values.

Booleans - Boolean data type is declared with the bool keyword, and can hold only two possible constant values, true or false.

Fixed-point numbers - Fixed point numbers represent decimal numbers in Solidity, although they aren’t fully supported by the Ethereum virtual machine yet.

Addresses - The address type is used to store Ethereum wallet or smart contract addresses, typically around 20 bytes. An address type can be suffixed with the keyword “payable”, which restricts it to store only wallet
addresses and use the transfer and send crypto functions.

Byte arrays - Byte arrays, declared with the keyword “bytes”, is a fixed-size array used to store a predefined number of bytes up to 32, usually declared along with the keyword (bytes1, bytes2).

Literals - Literals are immutable values such as addresses, rationals and integers, strings, unicode and hexadecimals, which can be stored in a variable.

Enums - Enums, short for Enumerable, are a user-defined data type, that restrict the value of the variable to a particular set of constants defined within the program.

Contract & Function Types - Similar to other object oriented languages, contract and function types are used to represent classes and their functions respectively. Contracts contain functions that can modify the contract’s state variables.

What is a signed integer (int)?

A signed integer is a value data type declared with the int keyword to store positive or negative integer values, ranging from -2 to the 255th power and 2 to the 255th power - 1.

Solidity Integer Sizes

The keyword int serves as an abbreviation for int256, an integer data value that can store up to a 256-bit integers or data units. This value data type can also be specified into smaller data values in steps of 8, such as int8, int16, int32, int64, int128 and int256. Developers can use the smaller data sizes such as int8, int16, int32, or int64 if they want to restrict the size of the variable and optimize performance.

What is an unsigned integer (uint)?

An unsigned integer is a value data type declared with the uint keyword which stores a integer value equal to or greater than zero, ranging from 0 to 2to the 255th power - 1.

Uint Data Sizes

Similar to a signed integer, the keyword uint servers an abbreviation for uint256, an unsigned integer data value that can store up to a 256-bit integers or data units.

An unsigned integer value data type can also be specified into smaller data values in steps of 8 from a minimum range of uint8 to maximum range of uint256. The denominations are uint8, uint16, uint32, uint64, uint128 and uint256.

The key difference between signed and unsigned integers is that signed integers can store both positive and negative values, while unsigned integers can store only non-negative values. Thus, although both signed and unsigned integers are both value data types, unsigned integers are restricted in the range of values it can store.

In smart contracts, this difference places an emphasis to the developer to prioritize the value that is stored in the data type to avoid an error.