Solidity – Variables

A Variable is basically a placeholder for the data which can be manipulated at runtime. Variables allow users to retrieve and change the stored information.

Rules For Naming Variables

1. A variable name should not match with reserved keywords.

2. Variable names must start with a letter or an underscore (\_), and may contain letters from “a to z” or “A to Z” or digits from “0 to 9” and characters also.

Example: Geeks123, geeks, \_123geeks are valid variable names

123geeks, $Geeks, 12\_geeks are invalid variable names

3. The name of variables are case sensitive i.e.

Example:

Geeks123 and geeks123 are different variables

Declaration of Variables

In Solidity declaration of variables is a little bit different, to declare a variable the user has to specify the data type first followed by access modifier.

Syntax:

<type> <access modifier> <variable name> ;

Example:

int public int\_var;

Types of Variables

Solidity is a statically typed language i.e. each declared variable always has a default value based on its data type, which means there is no concept of ‘null’ or ‘undefined’. Solidity supports three types of variables:

1. State Variables: Values of these variables are permanently stored in the contract storage. Each function has its own scope, and state variables should always be defined outside of that scope.

Example: In the below example. the contract Solidity\_var\_Test initializes the values of an unsigned integer state variable using a constructor.

**Solidity**

**// Solidity program to**

**// demonstrate state**

**// variables**

**pragma solidity ^0.5.0;**

**// Creating a contract**

**contract Solidity\_var\_Test {**

**// Declaring a state variable**

**uint8 public state\_var;**

**// Defining a constructor**

**constructor() public {**

**state\_var = 16;**

**}**

**}**