

In Solidity, a struct is a flexible data structure format that allows several data types to be combined into a single variable or a special type. A struct's name designates the subsets of variables it contains after the data types have been grouped into it.

Think of structs as containers that can hold various kinds of objects so that when you move the container, all the contents move with it. As a result, a struct replies in accordance with the data types contained in it when a Solidity developer declares or calls the name of the struct.

1. You can develop more complex data types with various characteristics using Solidity's structures. By building a struct, you can declare whatever type you want.
2. They are helpful for collecting related data into one category.
3. Structures may be imported into one contract from another after being declared outside of it. It often serves as a record representation. The struct keyword, which generates a new data type, is used to define a structure.

## Syntax

```
struct <structure_name>
{
  <data type> variable_1;
  <data type> variable_2;
}
```



## Declaration

1. Declare a Struct Inside a Contract
2. Declare a Struct outside a Contract with Import Access

**YouTube Link:**



[https://www.youtube.com/embed/1hYj\\_rrpMas](https://www.youtube.com/embed/1hYj_rrpMas)

## Solidity Source File

