Arrays in Solidity

Arrays in Solidity are used to store multiple values of the same type in a single variable. They can be **fixed-size** or **dynamic-size**.

---

1. Fixed-Size Arrays

- A **fixed-size array** has a set length that cannot be changed once defined.
- You can store a specific number of elements, but you cannot add or remove elements.
- For example, if you create an array of 5 elements, it will always have 5 slots.

2. Dynamic Arrays

- A **dynamic-size array** allows you to add or remove elements during the execution of the contract.
- The size of the array can grow or shrink based on what is done in the contract.

3. Array Length

- Every array has a **length**, which tells you how many elements are stored in it.
- For dynamic arrays, the length can change as elements are added or removed.

4. Accessing Elements

- You access elements in an array by their **index** (position).
- The first element in an array is at index 0, the second is at index 1, and so on.

5. Storage and Memory

- Arrays can be stored either in **storage** (persistent data on the blockchain) or in **memory** (temporary data used only during the execution of a function).

Structs in Solidity

Structs in Solidity allow you to define custom data types that can group multiple variables of different types together. They are particularly useful when you need to represent more complex data structures within your contract. Structs provide Solidity Developers with a way to build **custom data types**.

Mappings

Mappings are an important data type in Solidity. With a mapping we can take values of one data type and map them to values of another data type.