

# Smart Contracts (Reference Types)

- Reference types natively available to Solidity:
- Fixed-size byte arrays (Technically a value type)
  - bytes (alias for bytes1), bytes2, bytes3,...bytes32

# Smart Contracts (Reference Types)

- Arrays
- Statically sized arrays
- Dynamically sized arrays

# Smart Contracts (Reference Types)

- Array Members
- length:
  - Used to hold the number of elements in the array
  - Dynamic arrays can be resized using this member
- push:
  - Elements can be appended to an array using this member

# Smart Contracts (Reference Types)

- Structs
- Solidity provides a way to create new types
- A collection of variables organized into a container

# Smart Contracts (Reference Types)

- Mappings
- Can be seen as hash tables which are virtually initialized such that every possible key exists, and is mapped to a value whose byte-representation is all zeroes

# Smart Contracts (Reference Types)

- Mappings do not have a concept of length
- Mappings can be used for state variables only, except as storage reference types in internal functions

# Smart Contracts (Reference Types)

- Mappings can be marked as public, and a getter function will be created to query the keys
- The value of a mapping can be a mapping. This is called a nested mapping, and the getter will have one parameter for each key