Data Locations - Storage, Memory and Calldata

Variables are declared as either `storage`, `memory` or `calldata` to explicitly specify the location of the data.

- `storage` - variable is a state variable (store on blockchain)
- `memory` - variable is in memory and it exists while a function is being called
- `calldata` - special data location that contains function arguments, only available for `external` functions

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.3;

contract DataLocations {
    uint[] public arr;
    mapping(uint => address) map;
    struct MyStruct {
        uint foo;
    }
    mapping(uint => MyStruct) myStructs;

    function f() public {
        // call _f with state variables
        _f(arr, map, myStructs[1]);

        // get a struct from a mapping
        MyStruct storage myStruct = myStructs[1];
        // create a struct in memory
        MyStruct memory myMemStruct = MyStruct(0);
    }

    function _f(
        uint[] storage _arr,
        mapping(uint => address) storage _map,
        MyStruct storage _myStruct
    ) internal {
        // do something with storage variables
    }

    // You can return memory variables
    function g(uint[] memory _arr) public returns (uint[] memory) {
        // do something with memory array
    }

    function h(uint[] calldata _arr) external {
        // do something with calldata array
    }
}
```

Try on Remix