Enum

Solidity supports enumerables and they are useful to model choice and keep track of state.

Enums can be declared outside of a contract.

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

contract Enum {
    // Enum representing shipping status
    enum Status {
        Pending,
        Shipped,
        Accepted,
        Rejected,
        Canceled
    }

    // Default value is the first element listed in
    // definition of the type, in this case "Pending"
    Status public status;

    // Returns uint
    // Pending  - 0
    // Shipped  - 1
    // Accepted - 2
    // Rejected - 3
    // Canceled - 4
    function get() public view returns (Status) {
        return status;
    }

    // Update status by passing uint into input
    function set(Status _status) public {
        status = _status;
```

```
    }

    // You can update to a specific enum like this
    function cancel() public {
        status = Status.Canceled;
    }

    // delete resets the enum to its first value, 0
    function reset() public {
        delete status;
    }
}
```

Declaring and importing Enum

File that the enum is declared in

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

enum Status {
    Pending,
    Shipped,
    Accepted,
    Rejected,
    Canceled
}
```

File that imports the enum above

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

import "./EnumDeclaration.sol";

contract Enum {
    Status public status;
}
```