```solidity
// SPDX-License-Identifier: MIT
// OpenZeppelin Contracts (last updated v4.7.0) (utils/StorageSlot.sol)

pragma solidity ^0.8.0;

/**
 * @dev Library for reading and writing primitive types to specific storage slots.
 *
 * Storage slots are often used to avoid storage conflict when dealing with upgradeable contracts.
 * This library helps with reading and writing to such slots without the need for inline assembly.
 *
 * The functions in this library return Slot structs that contain a `value` member that can be used to read or write.
 *
 * Example usage to set ERC1967 implementation slot:
 * ```
 * contract ERC1967 {
 *     bytes32 internal constant _IMPLEMENTATION_SLOT = 0x360894a13ba1a3210667c828492db98dca3e2076cc3735a920a3ca505d382bbc;
 *
 *     function _getImplementation() internal view returns (address) {
 *         return StorageSlot.getAddressSlot(_IMPLEMENTATION_SLOT).value;
 *     }
 *
 *     function _setImplementation(address newImplementation) internal {
 *         require(Address.isContract(newImplementation), "ERC1967: new implementation is not a contract");
 *         StorageSlot.getAddressSlot(_IMPLEMENTATION_SLOT).value = newImplementation;
 *     }
 * }
 * ```
 *
 * _Available since v4.1 for `address`, `bool`, `bytes32`, and `uint256`._
 */
library StorageSlot {
    struct AddressSlot {
        address value;
    }

    struct BooleanSlot {
        bool value;
    }

    struct Bytes32Slot {
        bytes32 value;
    }

    struct Uint256Slot {
        uint256 value;
    }

    /**
     * @dev Returns an `AddressSlot` with member `value` located at `slot`.
     */
    function getAddressSlot(bytes32 slot) internal pure returns (AddressSlot storage r) {
        /// @solidity memory-safe-assembly
        assembly {
            r.slot := slot
        }
    }

    /**
     * @dev Returns an `BooleanSlot` with member `value` located at `slot`.
     */
    function getBooleanSlot(bytes32 slot) internal pure returns (BooleanSlot storage r) {
        /// @solidity memory-safe-assembly
        assembly {
            r.slot := slot
        }
```

```
67        }
68
69        /**
70         * @dev Returns an `Bytes32Slot` with member `value` located at `slot`.
71         */
72        function getBytes32Slot(bytes32 slot) internal pure returns (Bytes32Slot storage
          r) {
73            /// @solidity memory-safe-assembly
74            assembly {
75                r.slot := slot
76            }
77        }
78
79        /**
80         * @dev Returns an `Uint256Slot` with member `value` located at `slot`.
81         */
82        function getUint256Slot(bytes32 slot) internal pure returns (Uint256Slot storage
          r) {
83            /// @solidity memory-safe-assembly
84            assembly {
85                r.slot := slot
86            }
87        }
88    }
89
```