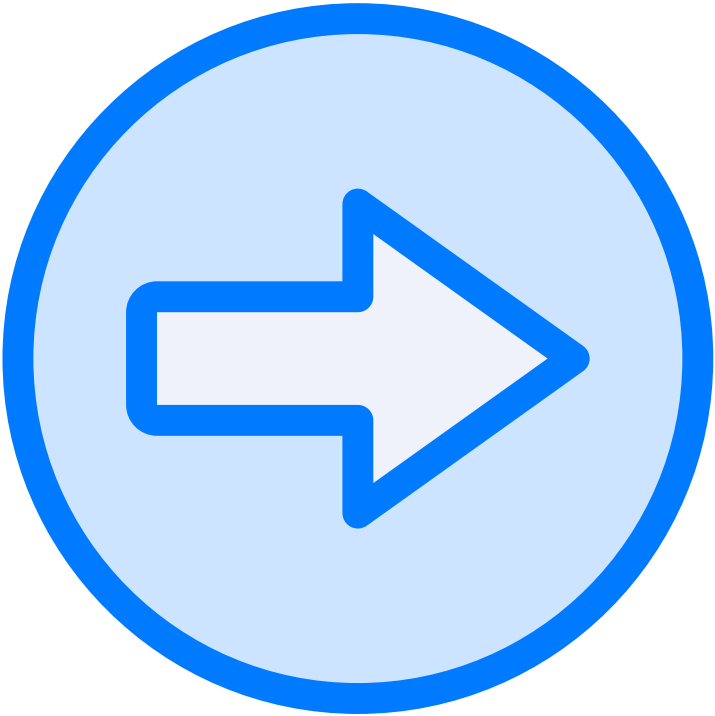
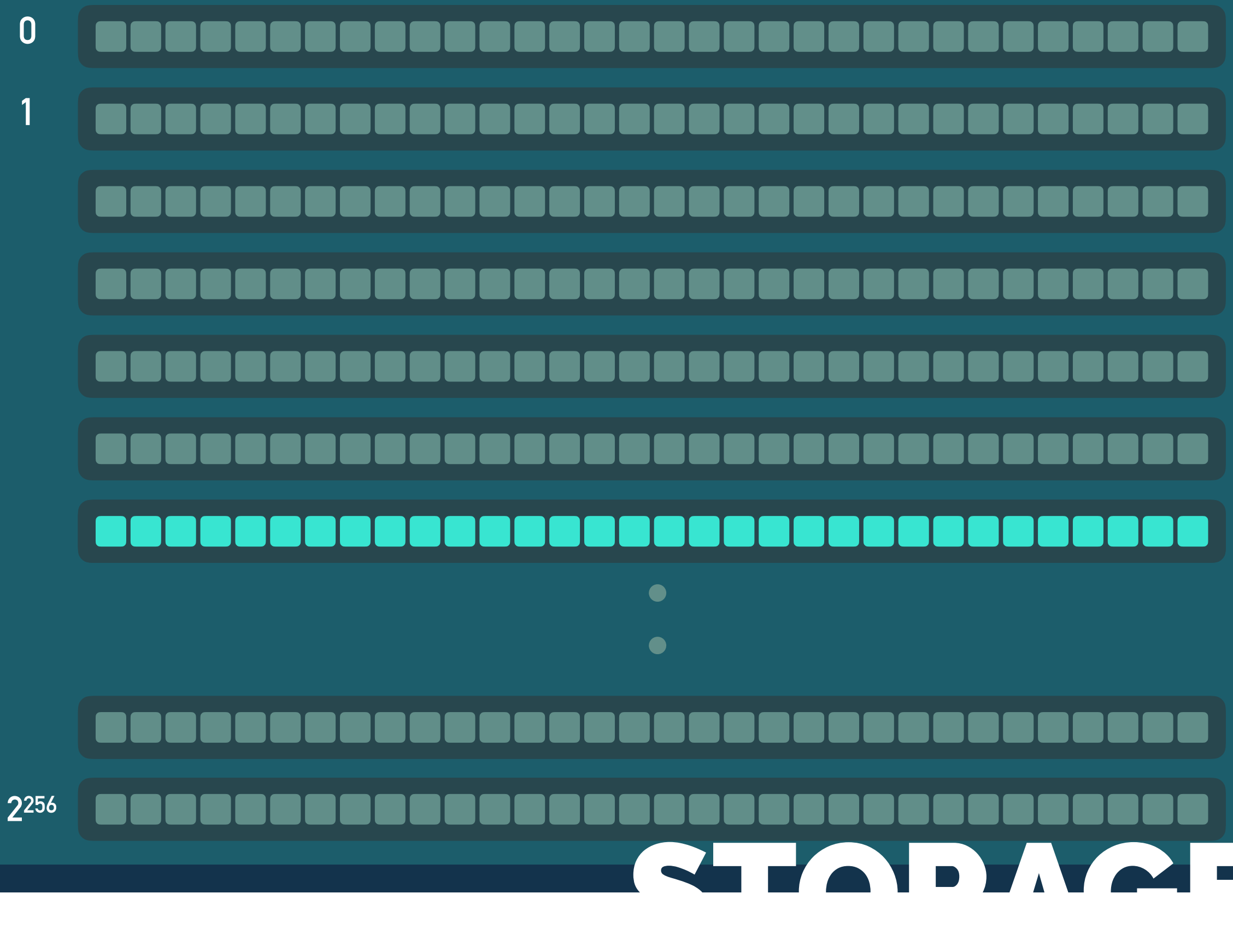
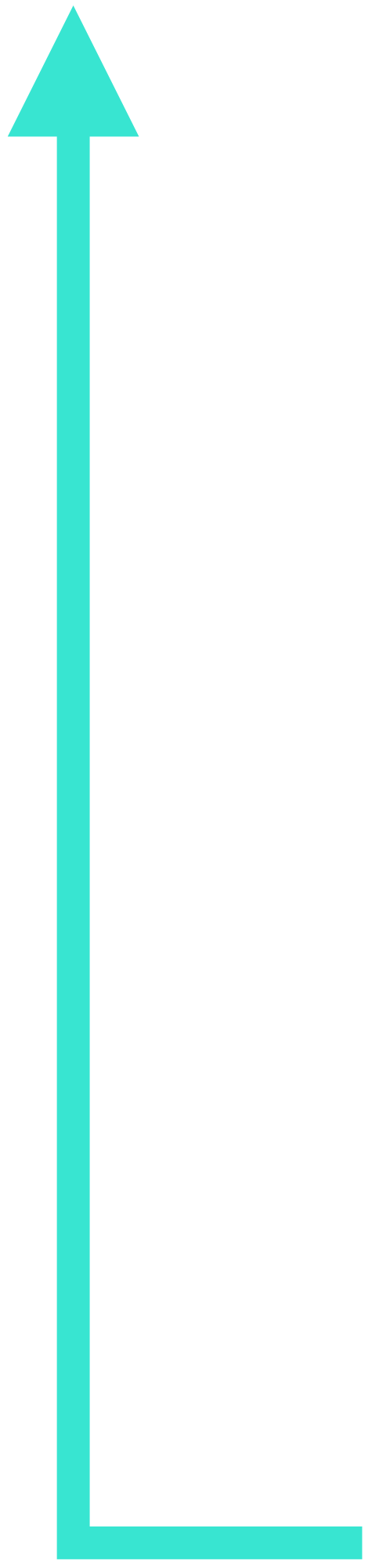
```
bytes32 private constant
            keccak256 “proxy.implementation”

function setAddress(address                    )
    sstore

function getAddress() returns address
                        sload

function    payable public
    delegatecall
```



Proxy

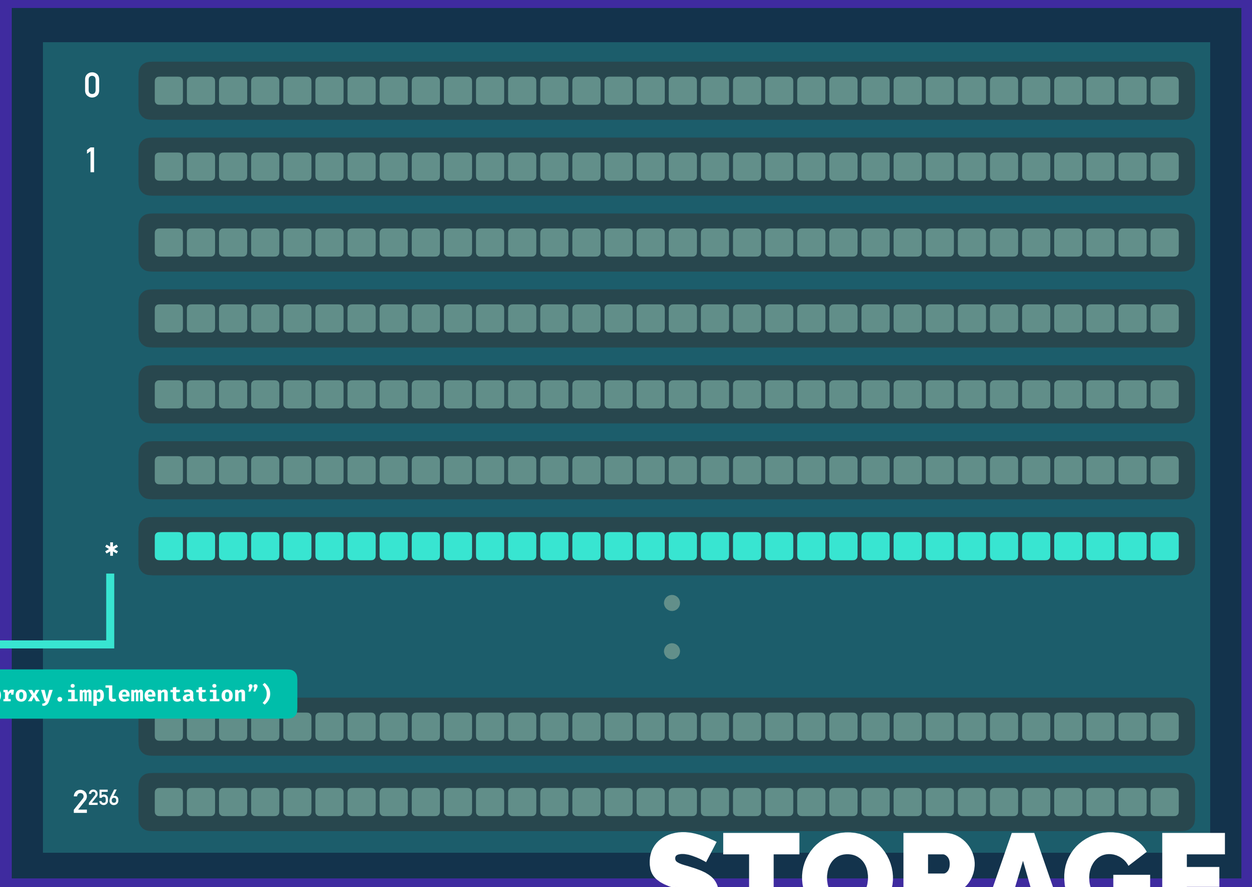

0
1
2^{256}
STORAGE

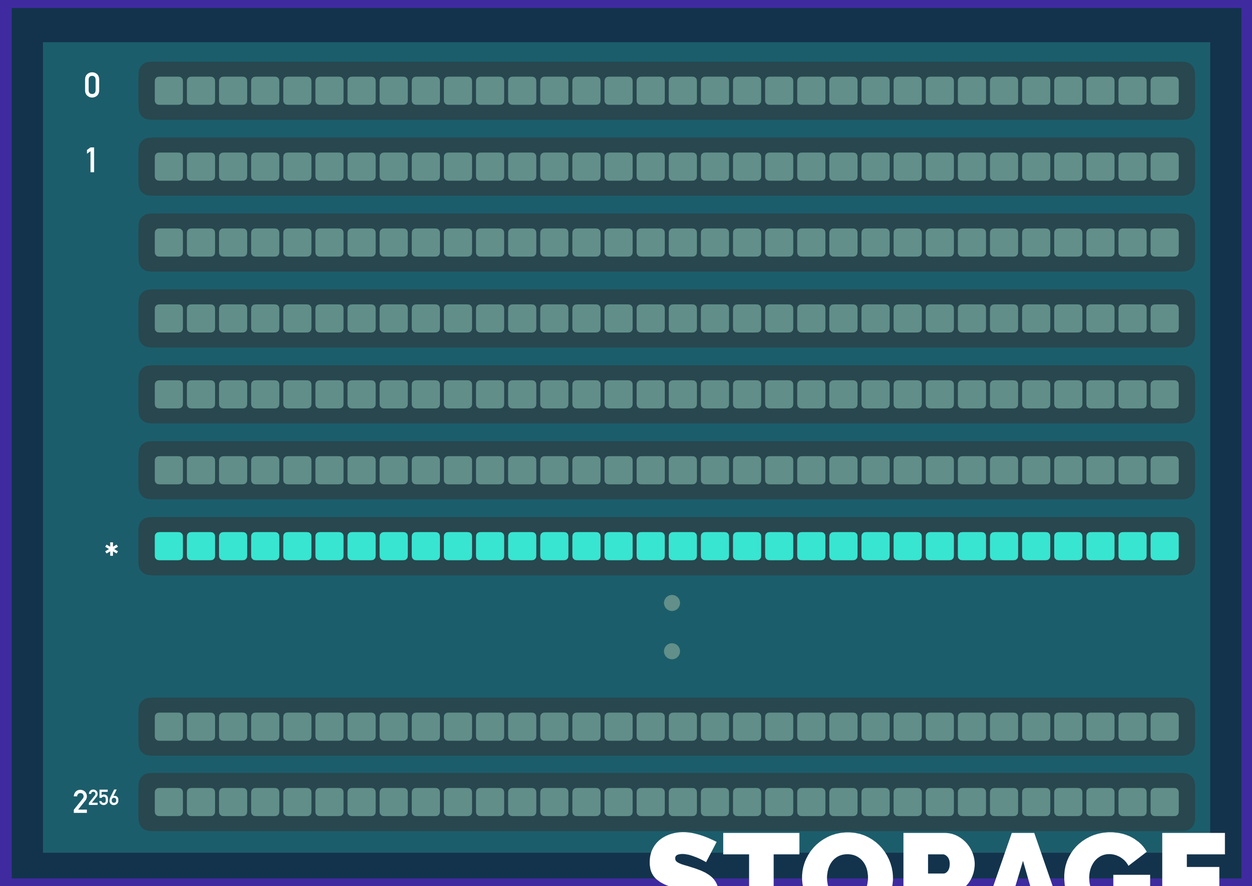
* keccak256(“proxy.implementation”)



```
contract Proxy {

bytes32 private constant implementationPos =
            keccak256(“proxy.implementation”);


function setAddress(address _implementation) {
  assembly {
    sstore(implementationPos, _implementation)
  }
}

function getAddress() returns(address _implementation) {
  assembly {
    _implementation := sload(implementationPos)
  }
}

function () payable public {
  … delegatecall() …
}

}
```

Proxy



```
contract Proxy {

bytes32 private constant implementationPos =
            keccak256(“proxy.implementation”);


function setAddress(address _implementation) {
  assembly {
    sstore(implementationPos, _implementation)
  }
}

function getAddress() returns(address _implementation) {
  assembly {
    _implementation := sload(implementationPos)
  }
}

function () payable public {
  … delegatecall() …
}

}
```

Proxy