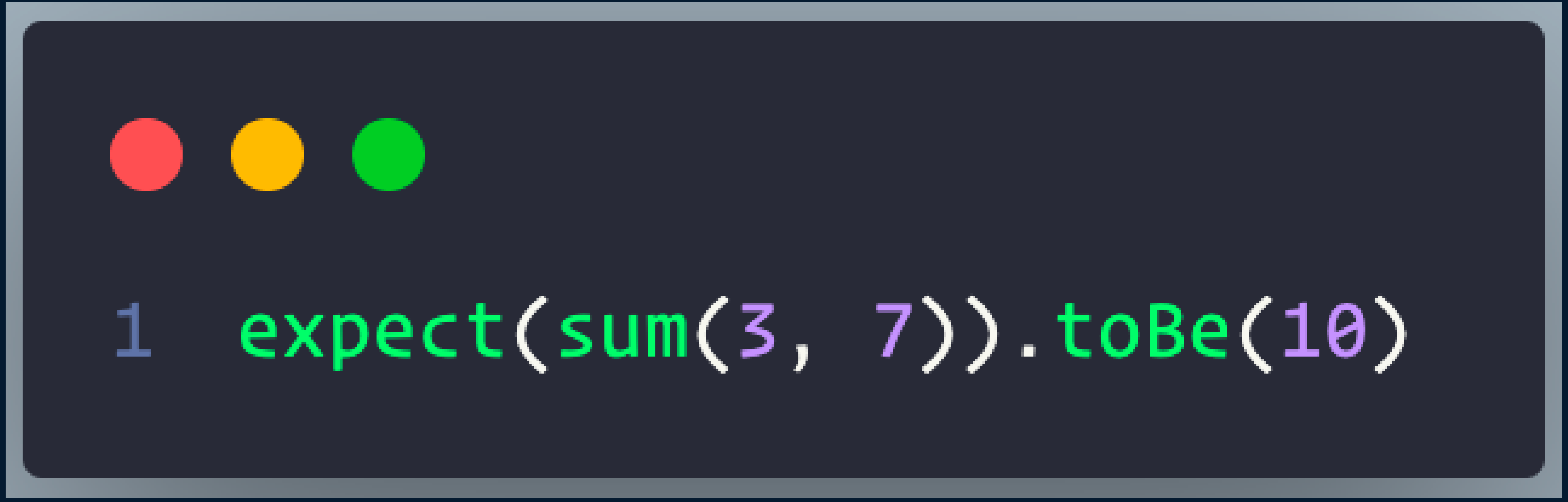
Desmitificando Jest

Jesus Hernandez
Frontend Developer

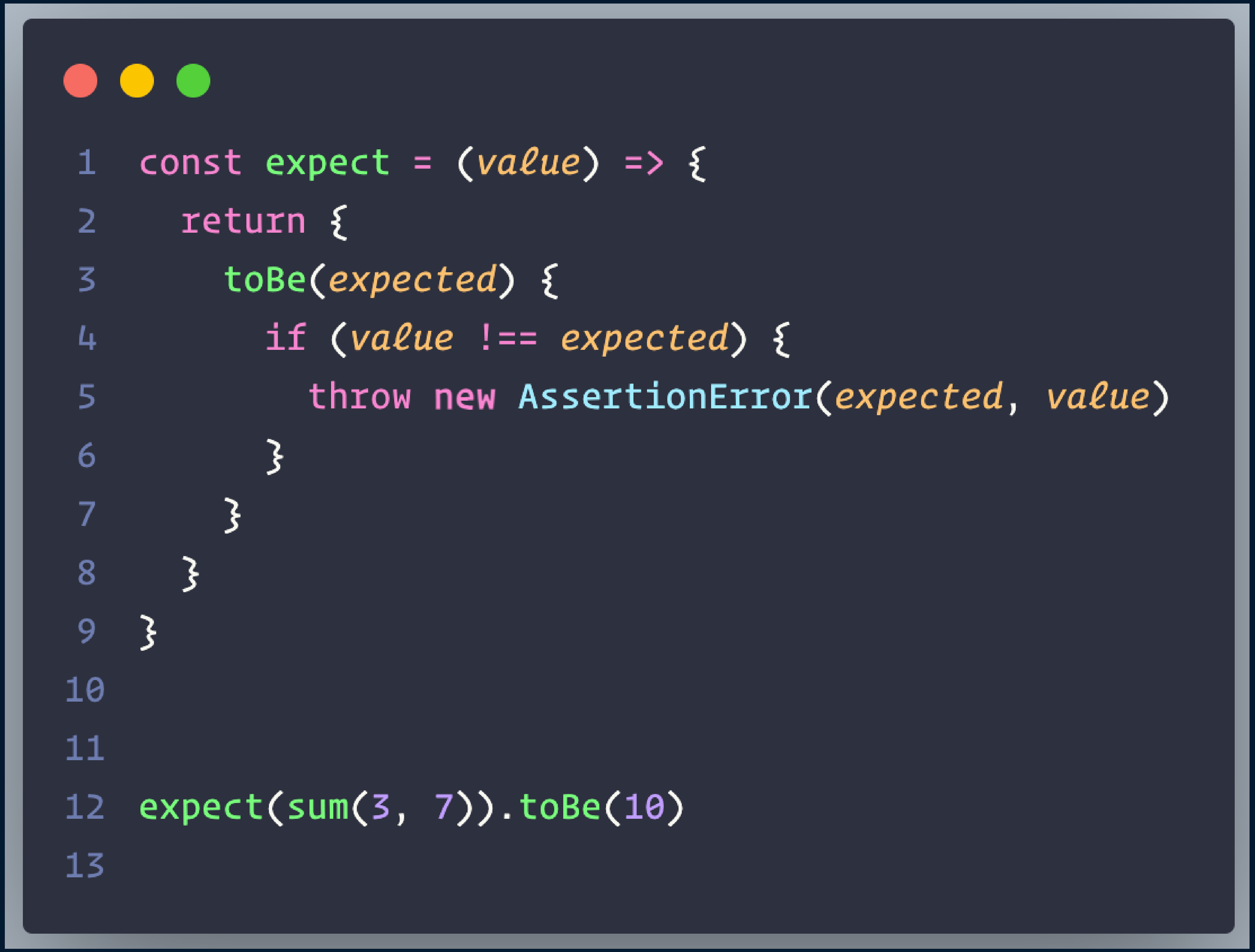
Jesuhrz

Expect (Jest)



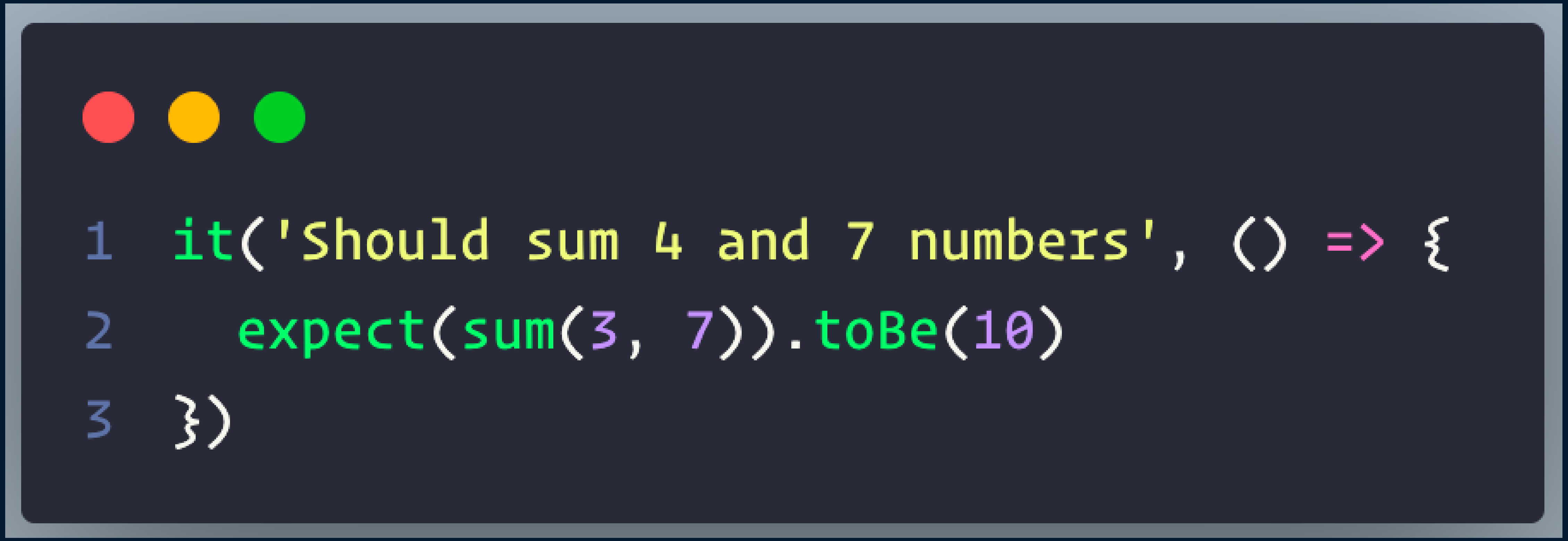
```
expect(sum(3, 7)).toBe(10)
```

Expect (From scratch)



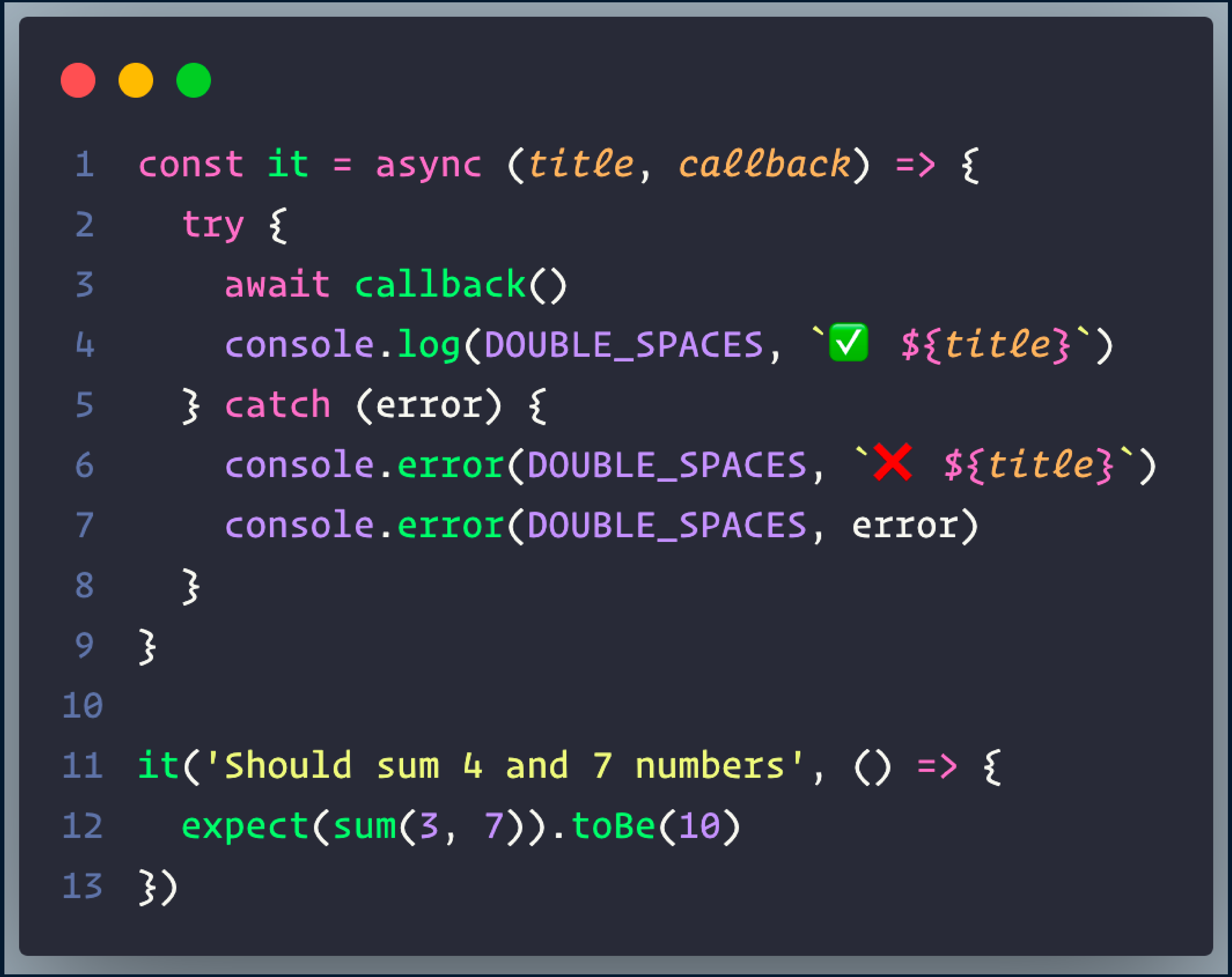
```javascript
const expect = (value) => {
  return {
    toBe(expected) {
      if (value !== expected) {
        throw new AssertionError(expected, value)
      }
    }
  }
}


expect(sum(3, 7)).toBe(10)

```

It (Jest)



```
it('Should sum 4 and 7 numbers', () => {
  expect(sum(3, 7)).toBe(10)
})
```

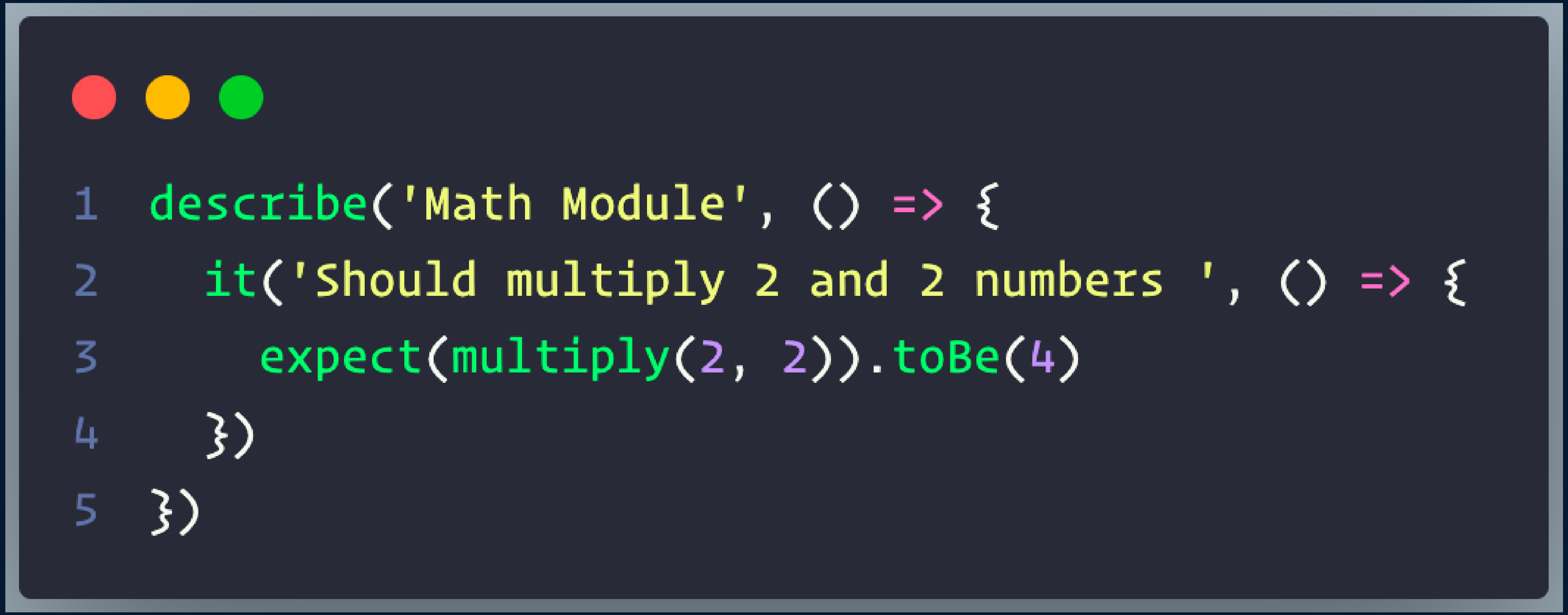
It (From scratch)



```javascript
const it = async (title, callback) => {
  try {
    await callback()
    console.log(DOUBLE_SPACES, `✅ ${title}`)
  } catch (error) {
    console.error(DOUBLE_SPACES, `❌ ${title}`)
    console.error(DOUBLE_SPACES, error)
  }
}

it('Should sum 4 and 7 numbers', () => {
  expect(sum(3, 7)).toBe(10)
})
```

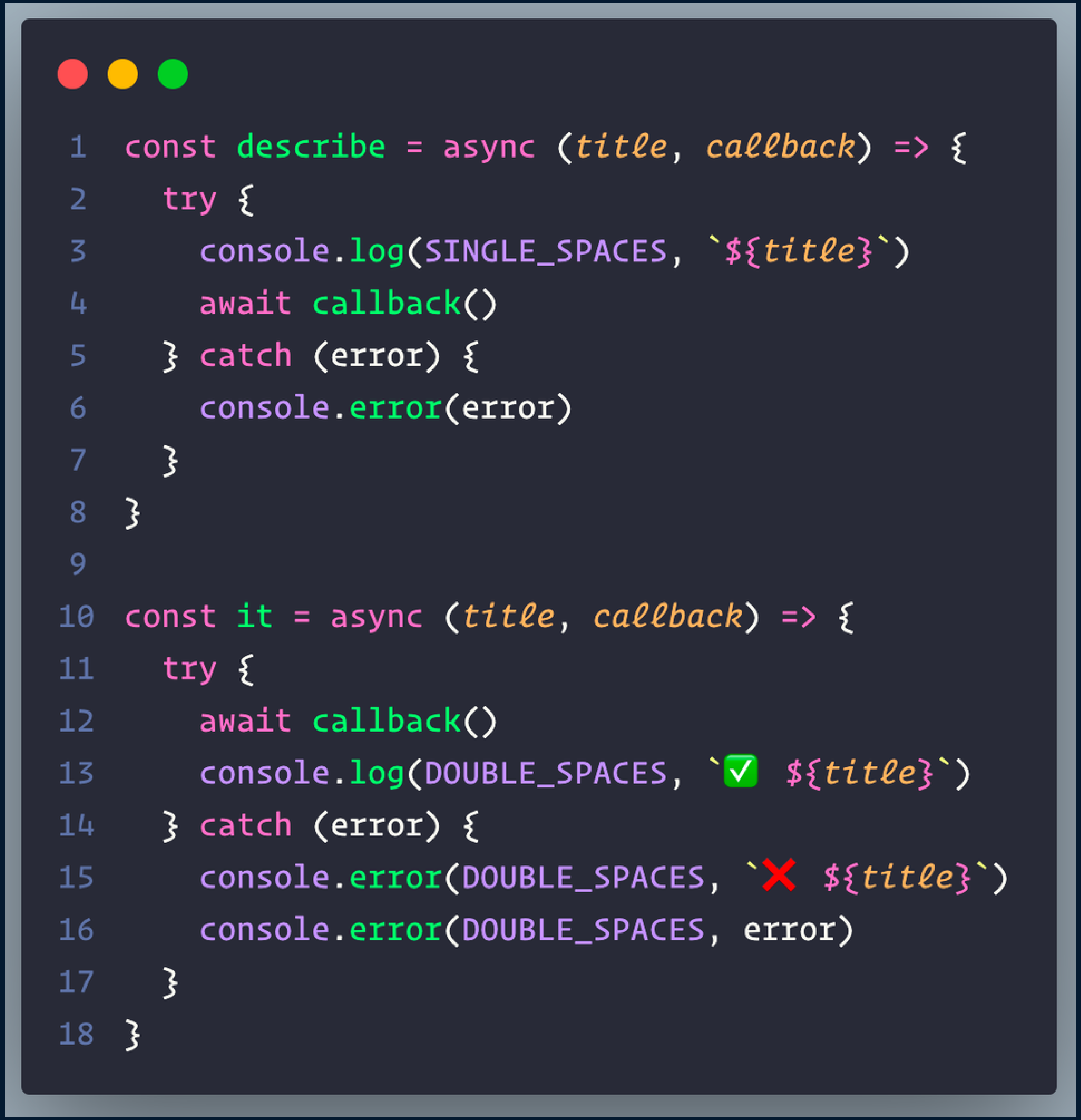
Jesuhrz

Describe (Jest)

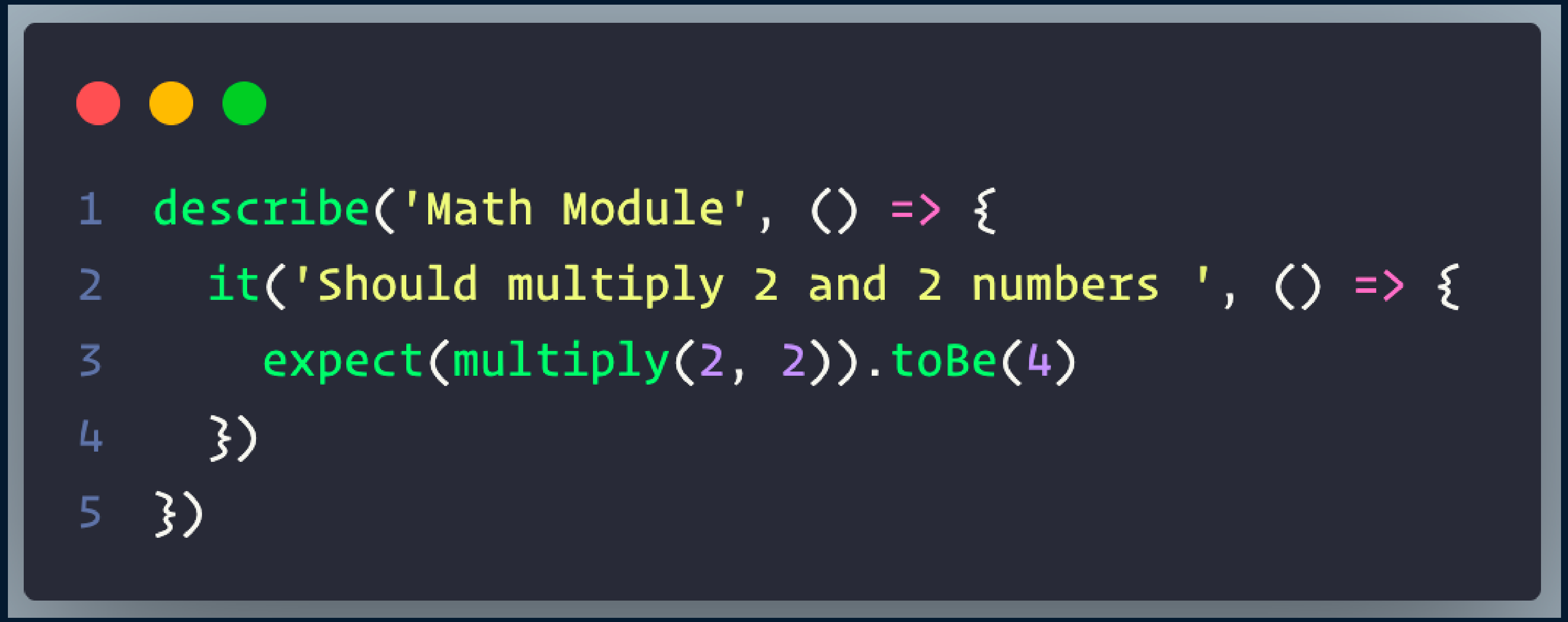


```
describe('Math Module', () => {
  it('Should multiply 2 and 2 numbers ', () => {
    expect(multiply(2, 2)).toBe(4)
  })
})
```

Describe (From scratch)

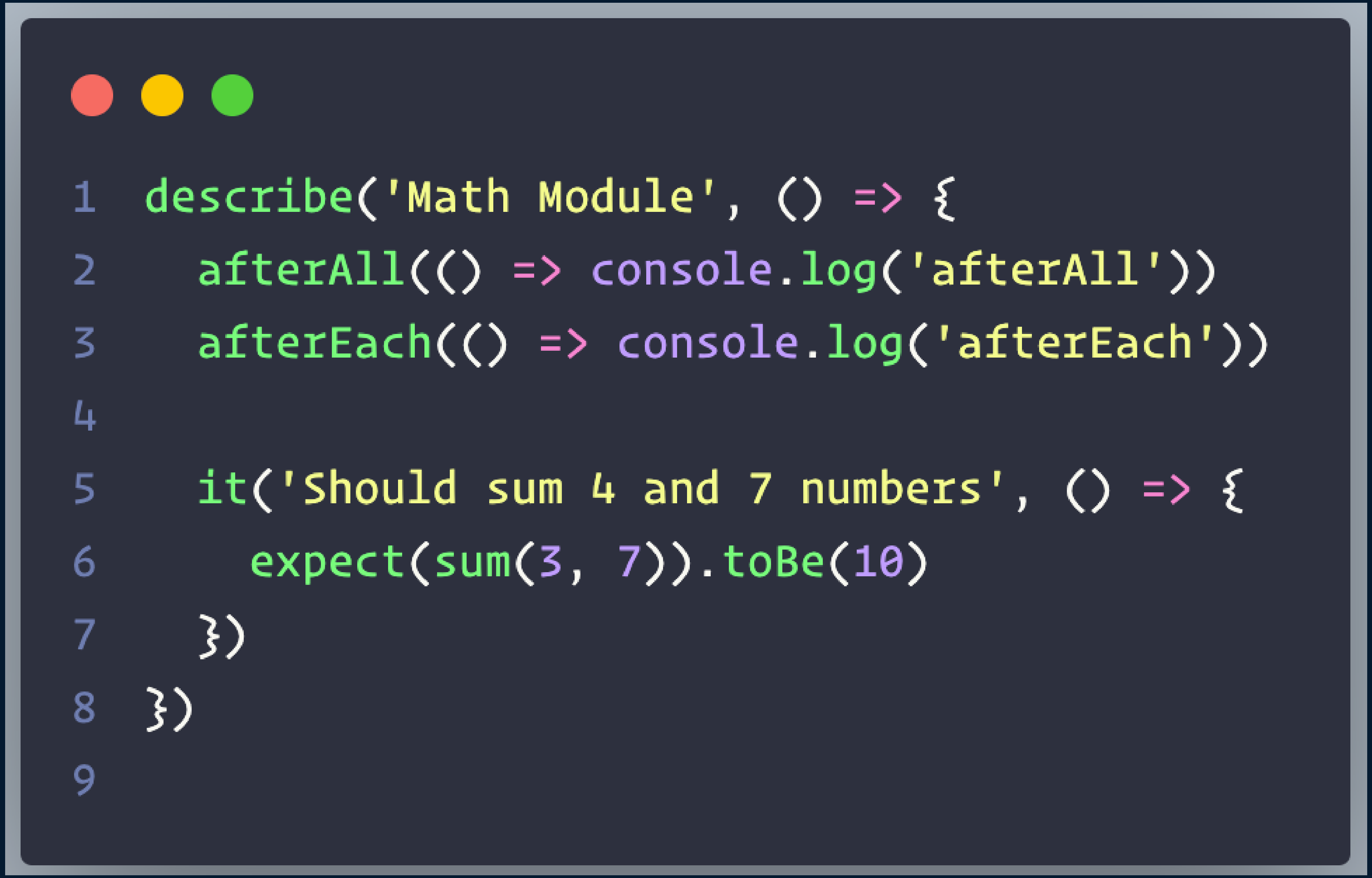


```
const describe = async (title, callback) => {
  try {
    console.log(SINGLE_SPACES, `${title}`)
    await callback()
  } catch (error) {
    console.error(error)
  }
}

const it = async (title, callback) => {
  try {
    await callback()
    console.log(DOUBLE_SPACES, `✅ ${title}`)
  } catch (error) {
    console.error(DOUBLE_SPACES, `❌ ${title}`)
    console.error(DOUBLE_SPACES, error)
  }
}
```



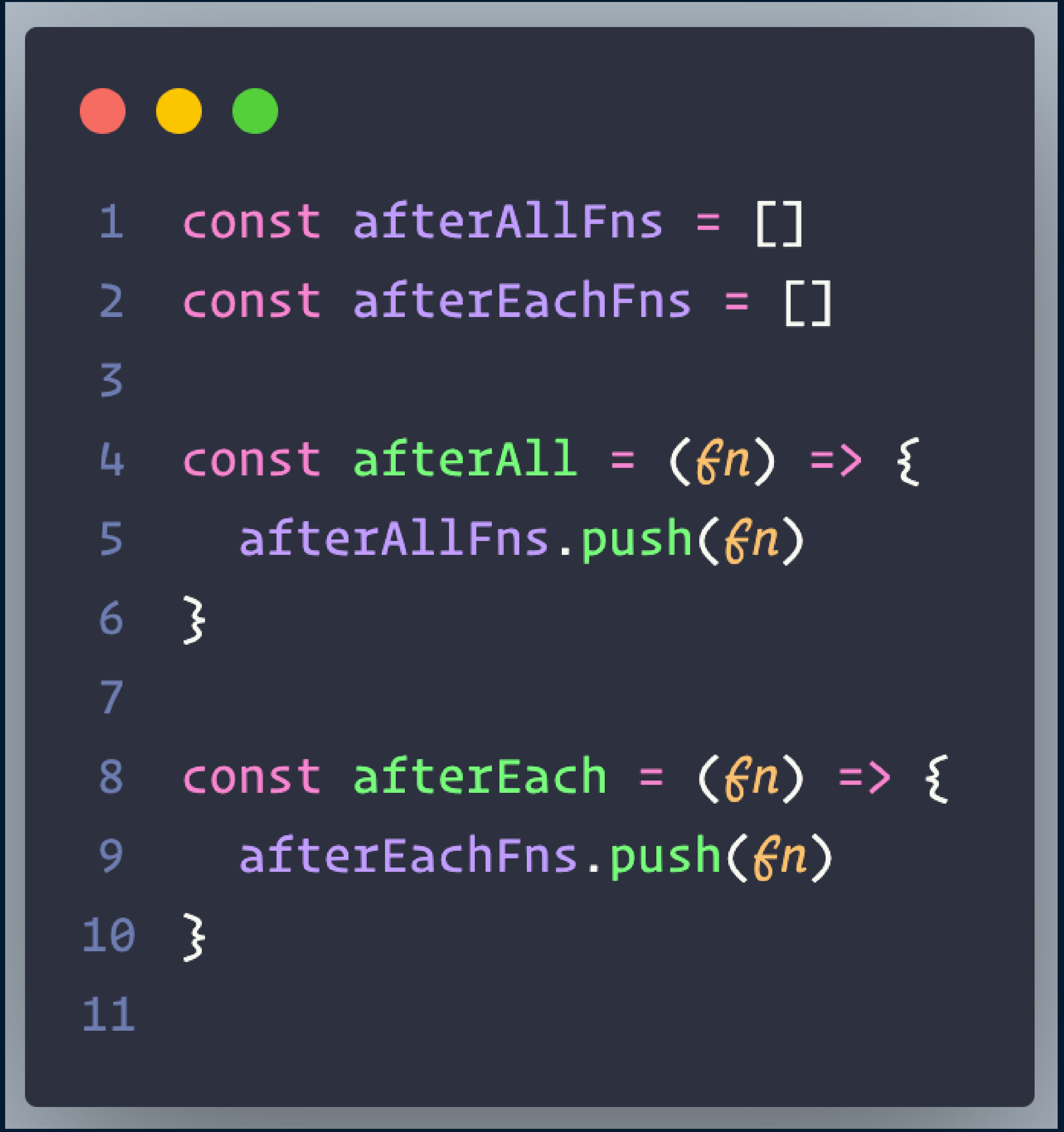
```
describe('Math Module', () => {
  it('Should multiply 2 and 2 numbers ', () => {
    expect(multiply(2, 2)).toBe(4)
  })
})
```

Hooks (Jest)

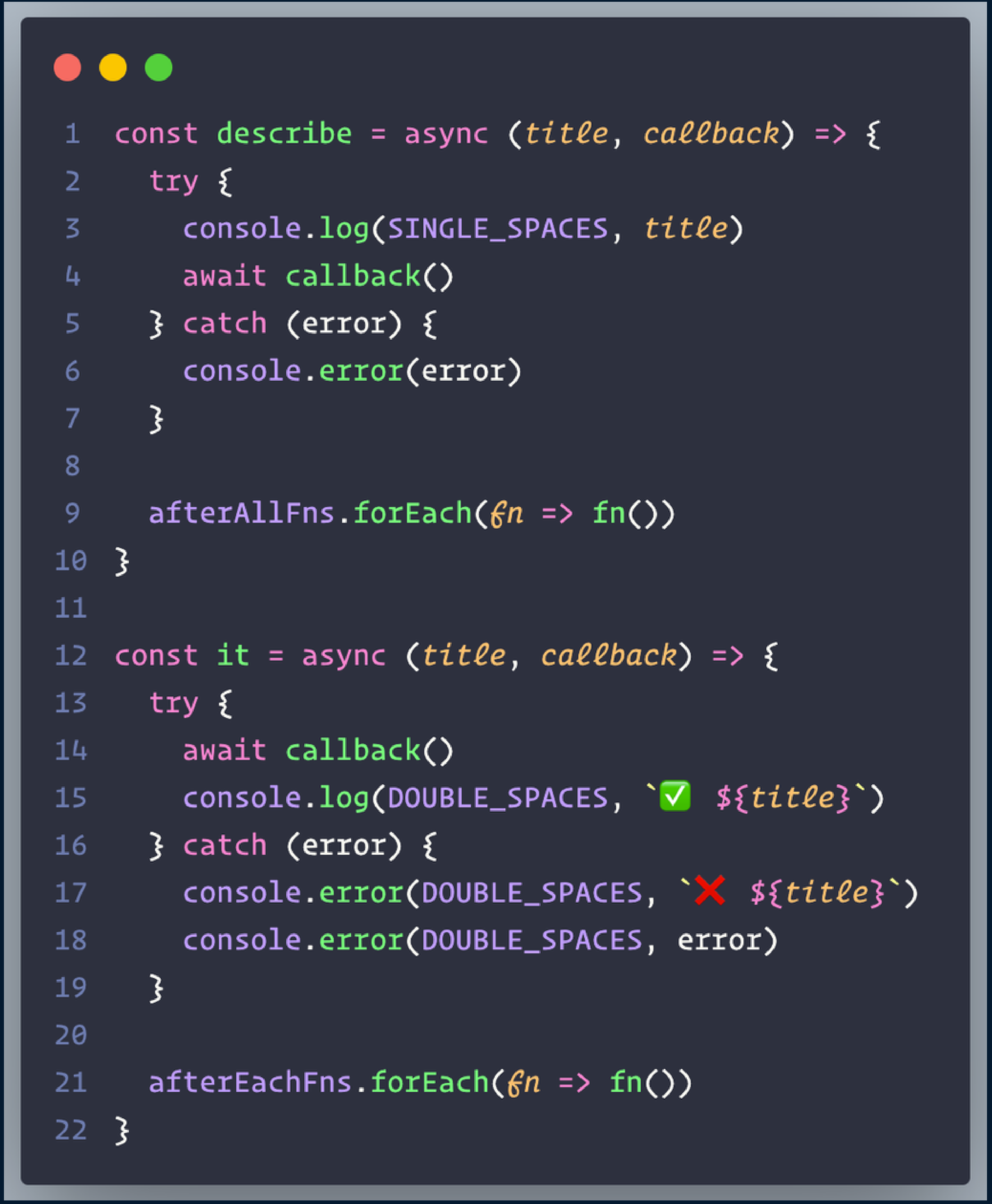


```
describe('Math Module', () => {
  afterAll(() => console.log('afterAll'))
  afterEach(() => console.log('afterEach'))

  it('Should sum 4 and 7 numbers', () => {
    expect(sum(3, 7)).toBe(10)
  })
})
```

Hooks (From scratch)

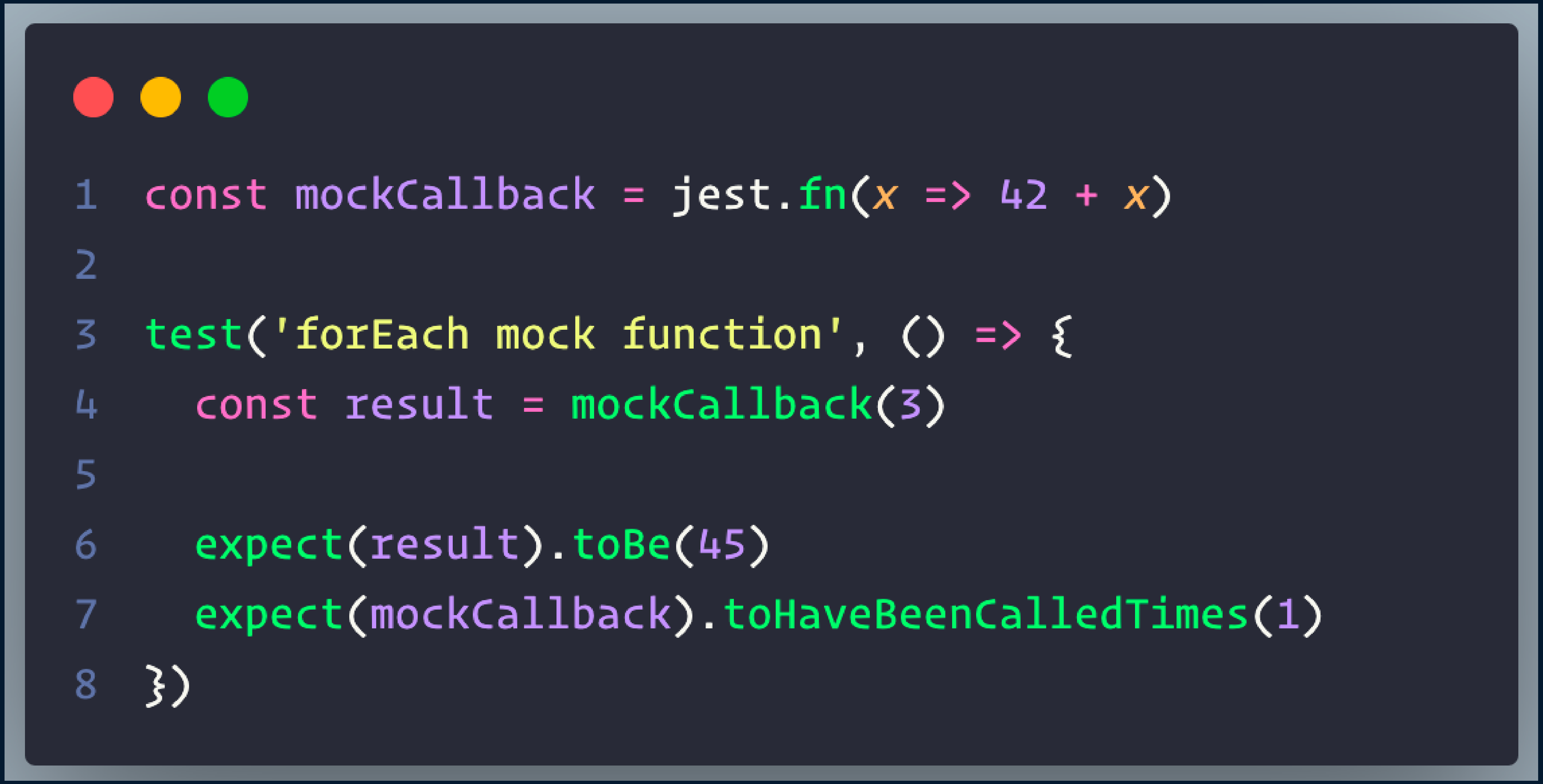


```javascript
const afterAllFns = []
const afterEachFns = []

const afterAll = (fn) => {
  afterAllFns.push(fn)
}

const afterEach = (fn) => {
  afterEachFns.push(fn)
}

```



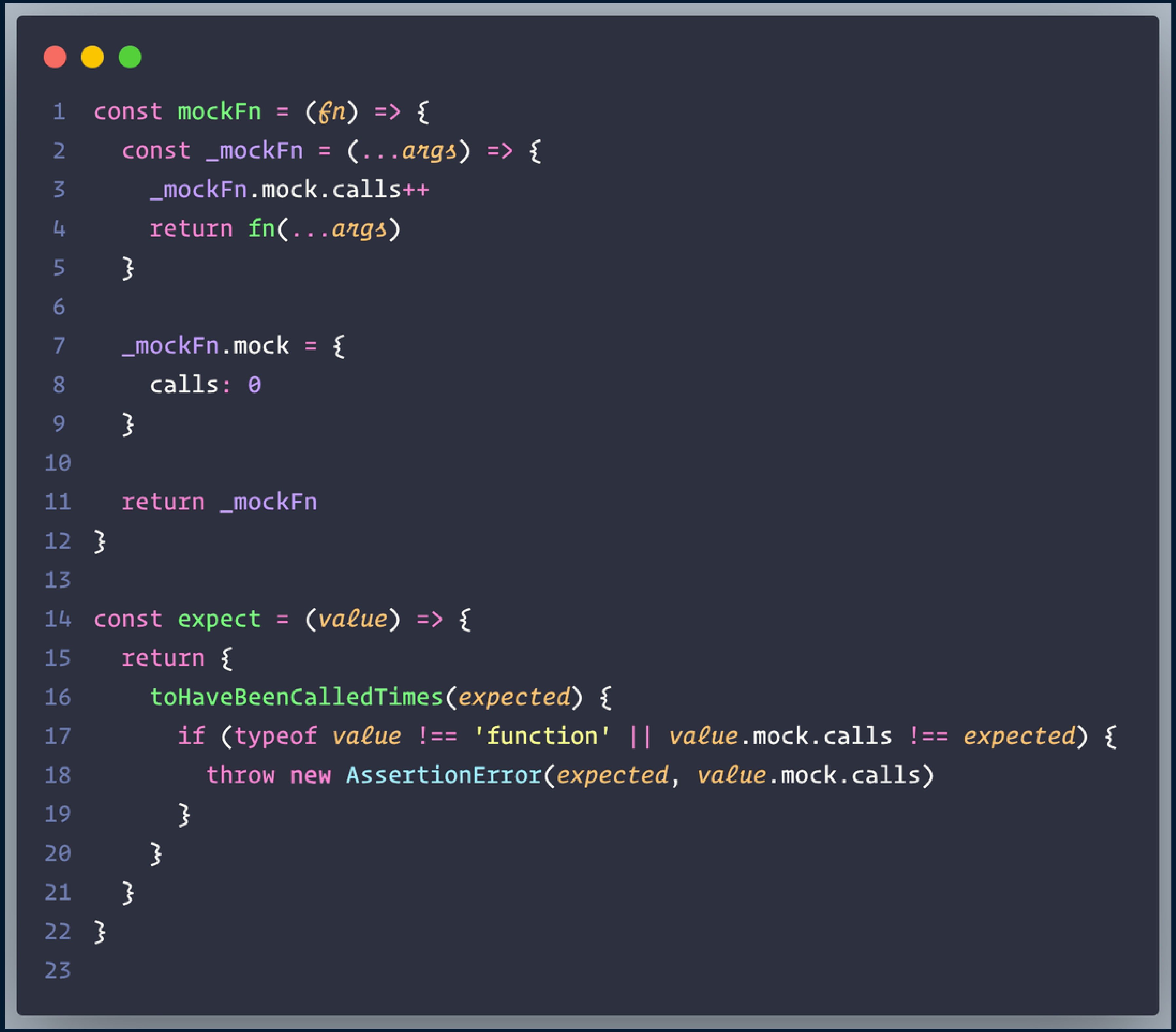
```javascript
const describe = async (title, callback) => {
  try {
    console.log(SINGLE_SPACES, title)
    await callback()
  } catch (error) {
    console.error(error)
  }

  afterAllFns.forEach(fn => fn())
}

const it = async (title, callback) => {
  try {
    await callback()
    console.log(DOUBLE_SPACES, `✅ ${title}`)
  } catch (error) {
    console.error(DOUBLE_SPACES, `❌ ${title}`)
    console.error(DOUBLE_SPACES, error)
  }

  afterEachFns.forEach(fn => fn())
}
```

Mock Functions (Jest)

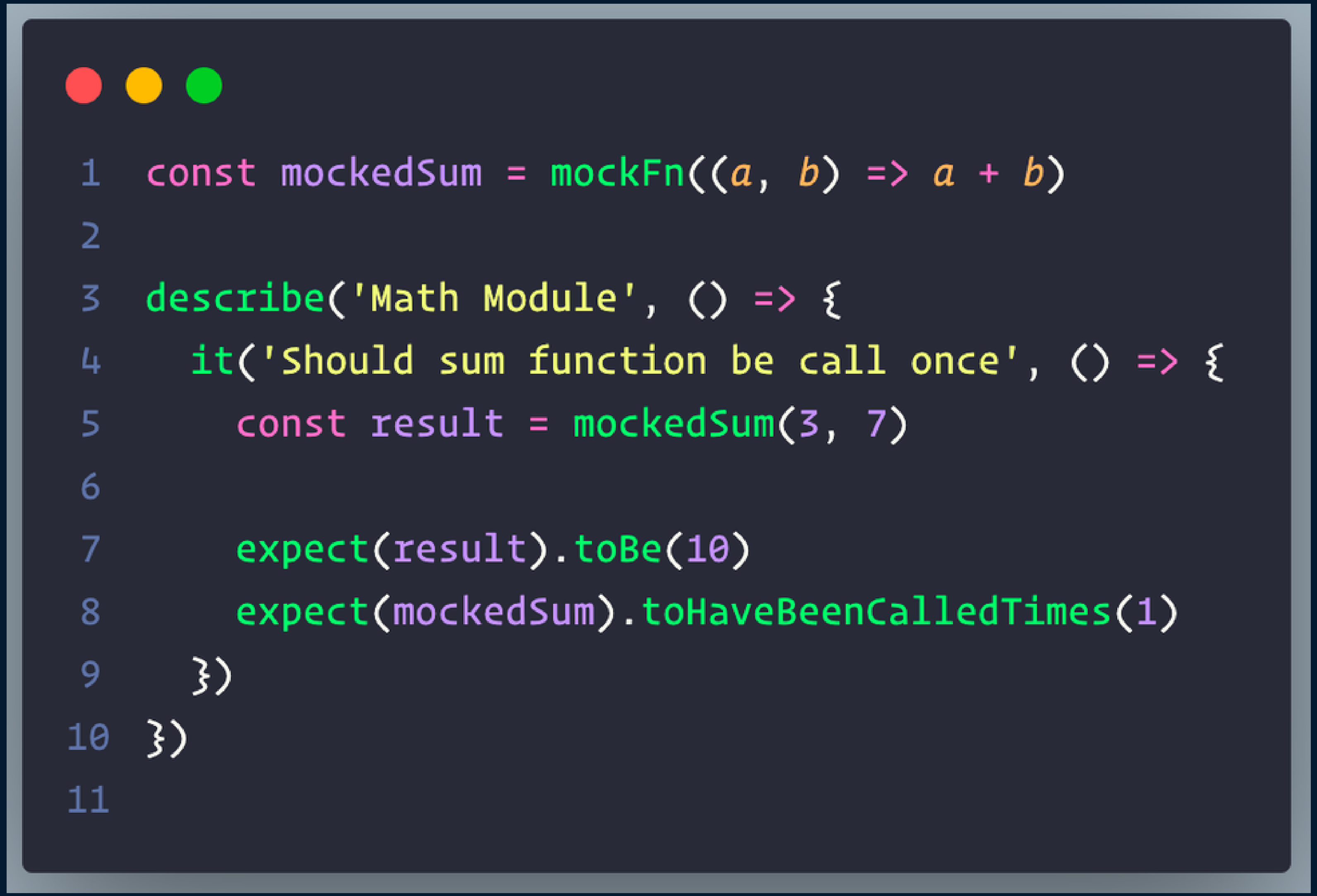


```javascript
const mockCallback = jest.fn(x => 42 + x)

test('forEach mock function', () => {
  const result = mockCallback(3)

  expect(result).toBe(45)
  expect(mockCallback).toHaveBeenCalledTimes(1)
})
```

Mock Functions (From scratch)



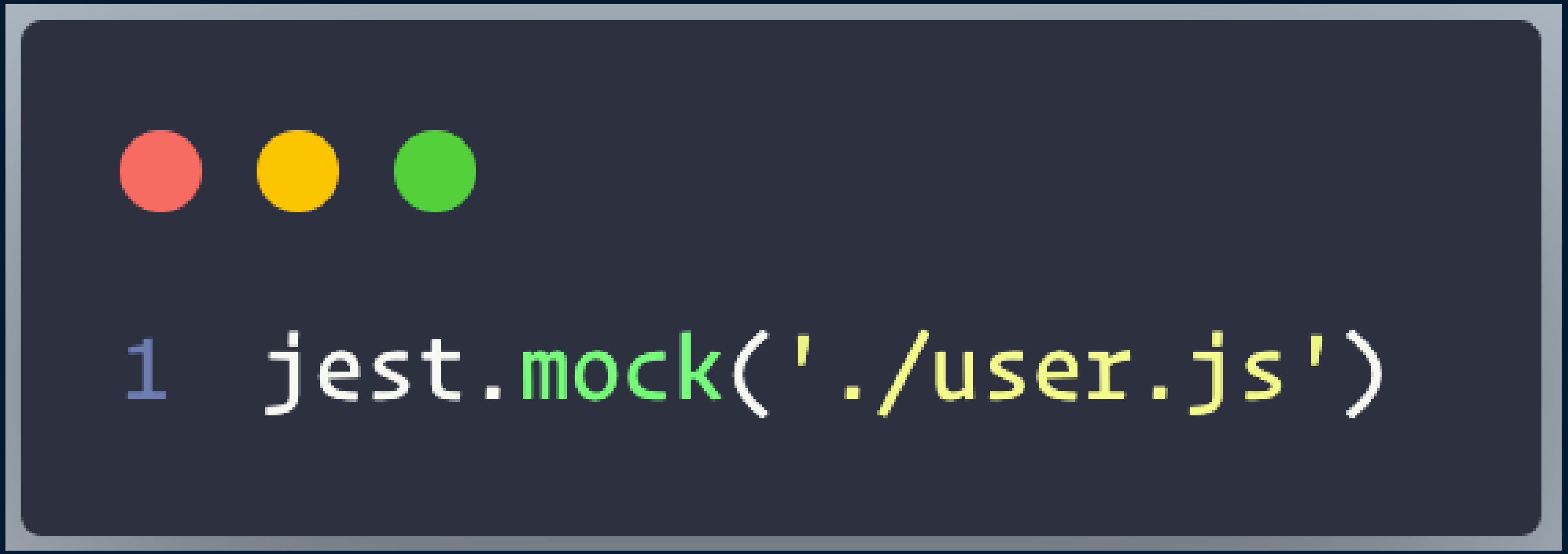
```javascript
const mockFn = (fn) => {
  const _mockFn = (...args) => {
    _mockFn.mock.calls++
    return fn(...args)
  }

  _mockFn.mock = {
    calls: 0
  }

  return _mockFn
}

const expect = (value) => {
  return {
    toHaveBeenCalledTimes(expected) {
      if (typeof value !== 'function' || value.mock.calls !== expected) {
        throw new AssertionError(expected, value.mock.calls)
      }
    }
  }
}
```



```javascript
const mockedSum = mockFn((a, b) => a + b)

describe('Math Module', () => {
  it('Should sum function be call once', () => {
    const result = mockedSum(3, 7)

    expect(result).toBe(10)
    expect(mockedSum).toHaveBeenCalledTimes(1)
  })
})
```

Mock Module (Jest)

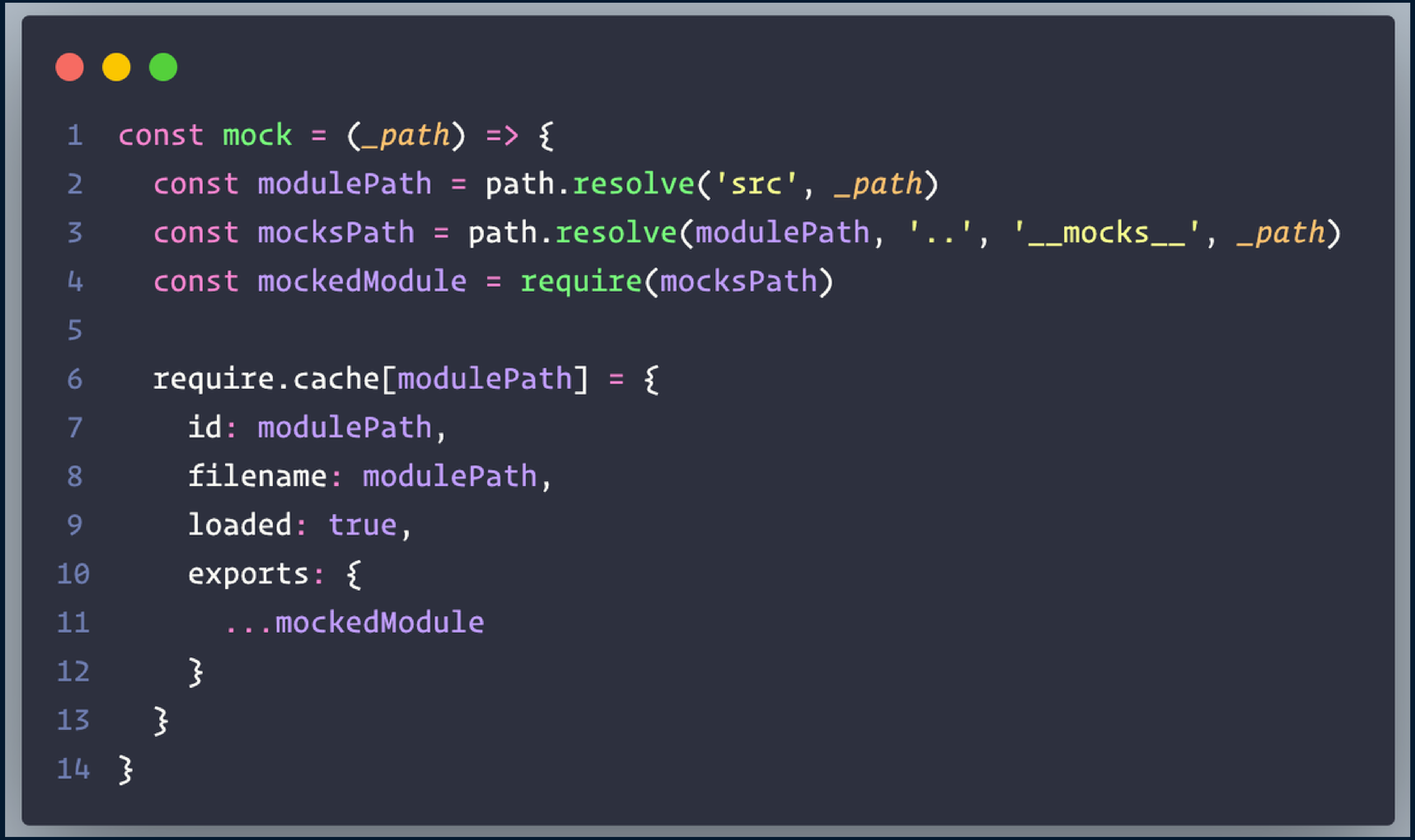
"Manual mocks are defined by writing a module in a __mocks__/ subdirectory immediately adjacent to the module. For example, to mock a module called user in the models directory, create a file called user.js and put it in the models/__mocks__ directory."



```
jest.mock('./user.js')
```

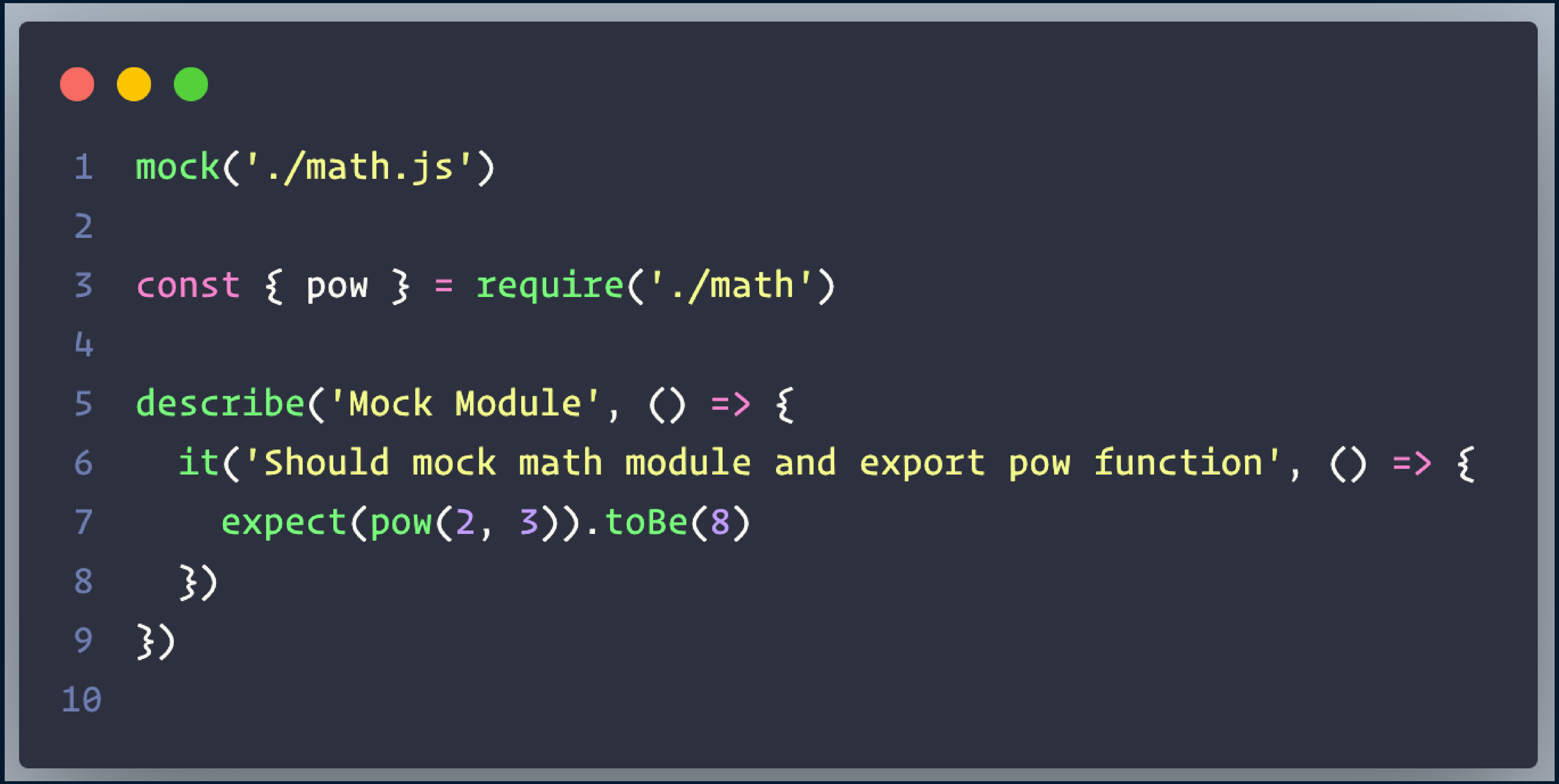
https://jestjs.io/docs/manual-mocks#mocking-node-modules

Mock Module (From scratch)



```
const mock = (_path) => {
  const modulePath = path.resolve('src', _path)
  const mocksPath = path.resolve(modulePath, '..', '__mocks__', _path)
  const mockedModule = require(mocksPath)

  require.cache[modulePath] = {
    id: modulePath,
    filename: modulePath,
    loaded: true,
    exports: {
      ...mockedModule
    }
  }
}
```

Mock Module (From scratch)



```javascript
mock('./math.js')

const { pow } = require('./math')

describe('Mock Module', () => {
  it('Should mock math module and export pow function', () => {
    expect(pow(2, 3)).toBe(8)
  })
})

```

Jesuhrz

Gracias

https://github.com/jesuhrz/desmitificando-jest