

Unit 18.1 Highlights

(Source: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/const>)
(Source: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/let>)

let - Declares variables which are limited in scope to the block, statement, or expression on which it is used.

const - Declaration creates a constant whose scope can be either global or local to the block in which it is declared. The value of a constant cannot change through reassignment, and it can't be redeclared.

BDD - software development methodology in which an application is specified and designed by describing how its behavior should appear to an outside observer.

TDD - software development process that relies on the repetition of a very short development cycle: requirements are turned into very specific test cases, then the software is improved to pass the new tests

mocha.js - Node.js testing framework (<https://www.chaijs.com/>)

chaijs - BDD / TDD assertion library for node (<https://mochajs.org>) to verify things are correct

describe() - groups tests in Mocha and takes two arguments: the name of the test group, and a callback function.

it() - used for an individual / unit test cases and takes two arguments: a string explaining what the test should do, and a callback function which contains the test

expect() - The BDD style where the semantic style of the test is written BEFORE the code is written and uses the past tense

Ex. 1

```
//===== test.js=====//

const expect = require('chai').expect; //load the assertion library of choice
const jump = require('./jump'); //load the export code

//group the tests
describe('might as well jump', function() {
  it('jump as high as you can', function() {
    expect(jump(8)).to.equal(8);
  });

  it('should return undefined if no value is passed into jump method', function() {
    expect(jump()).to.equal(null);
  });

  it('should return null if a value other than a number is sent as a parameter', function() {
    expect(jump([1,2,3,4,5])).to.equal(null);
  });
});

//===== jump.js=====//

module.exports = function(input){
  if (!input || typeof input !== 'number'){
    return null;
  }else{
    return input
  }
}
```