

Node.js

Lesson 12—Working with Task Managers and Unit Testing



Lesson Overview

In this lesson, you will learn how to work with Grunt and Gulp, along with Mocha and Chai.

Introduction to Task Managers

Many applications require repetitive tasks that are time consuming and boring.



If you need to perform any of the following repeatedly, then you can use a task manager as it automates the following jobs:

1. Minify and concatenate JavaScript and CSS files
2. CSS preprocessing
3. Image optimization
4. Unit testing
5. Watch for changes

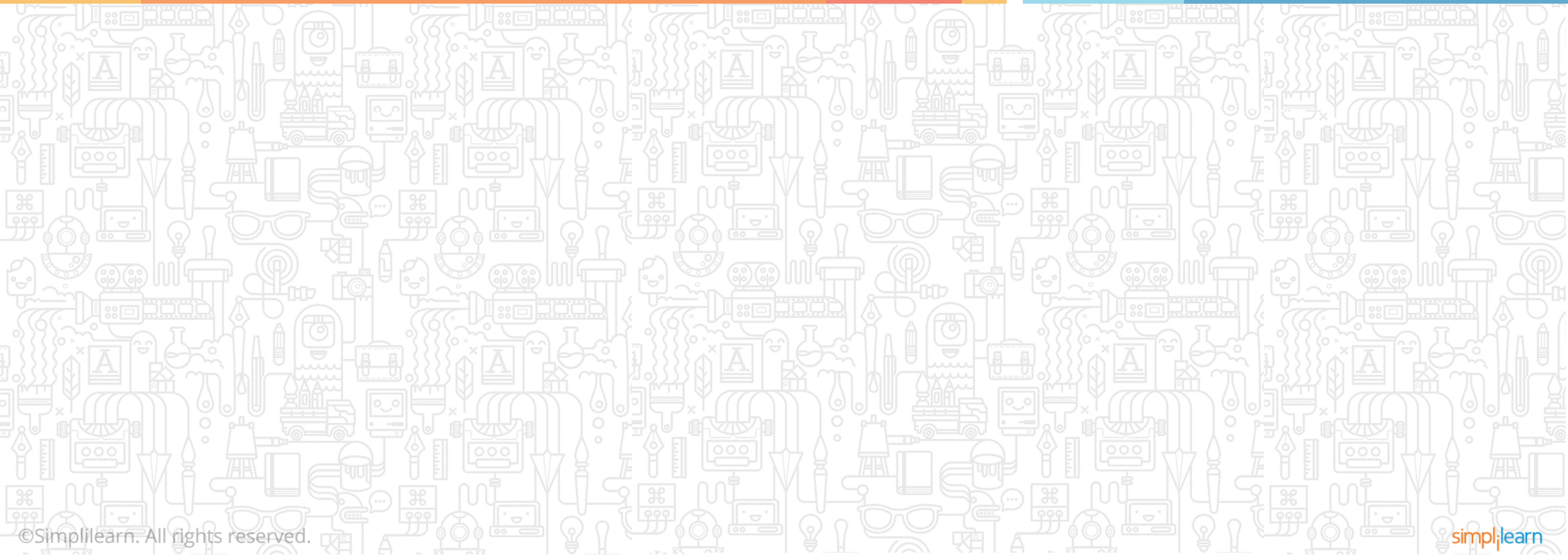
Learning Objectives



- ✓ Working with GRUNT
- ✓ Working with GULP
- ✓ Working with Mocha and Chai

Working with Task Managers and Unit Testing

Topic 1—Work with Grunt



GRUNT

INTRODUCTION

Grunt is a task-oriented command line build tool for JavaScript based application.



Grunt can perform the following tasks:

- Minify and concatenate JavaScript and CSS files
- Automate repetitive tasks
- Separate prod from DEV
- Unit testing
- Watch for changes
- Replace scripts in HTML file
- Work with SASS

GRUNT

WHY GRUNT?

Grunt task runner is preloaded with many benefits.



- Grunt is Open Source
- Very large developer community
- Easy to implement
- Tons of plugins
- Support for custom plugin
- Easy learning curve

GRUNT

COMPARE GRUNT WITH ANT



Grunt is simple and require less configuration



Ant is difficult to maintain and require a lot of complex code

```
1  grunt.initConfig({
2    min: {
3      dist: {
4        src: ['dist/built.js'],
5        dest: 'dist/built.min.js'
6      }
7    }
8  });
```

```
1  <target name="js-compile-all" description="Compile Javascript files with Closure" >
2    <echo>Compiling JS files in ${input.scripts.dir} in closure...</echo>
3    <apply executable="java" dest="${output.scripts.dir}">
4      <arg value="-jar"/>
5      <arg path="${jar.lib.dir}/closure-compiler.jar"/>
6      <arg line="-js"/>
7      <srcfile/>
8      <arg line="--js_output_file"/>
9      <targetfile/>
10     <fileset dir="${output.scripts.dir}" includes="**/*-main.debug.js" />
11     <mapper type="glob" from="*-main.debug.js" to="*-main.min.js"/>
12   </apply>
13   <echo>Finished compiling JS files</echo>
14 </target>
```


GRUNT

INSTALLATION

You must install Grunt's command line interface (CLI) globally through Node.js.

```
$ npm install -g grunt-cli
```



This command will install the latest version of Grunt in your current project folder, in package.json

```
$ npm install grunt --save-dev
```

GRUNT

HOW TO USE GRUNT IN A PROJECT

We need to create “gulpfile.js” file which contains all the tasks.

```
module.exports = function(grunt) {  
  // Project configuration.  
  grunt.initConfig({  
    pkg: grunt.file.readJSON('package.json'),  
    uglify: {  
      options: {  
        banner: '/*! <%= pkg.name %> <%=  
grunt.template.today("yyyy-mm-dd") %> */\n'  
      },  
      build: {  
        src: 'src/<%= pkg.name %>.js',  
        dest: 'build/<%= pkg.name %>.min.js'  
      }  
    }  
  });  
  // Load the plugin that provides the "uglify" task.  
  grunt.loadNpmTasks('grunt-contrib-uglify');  
  // Default task(s).  
  grunt.registerTask('default', ['uglify']);  
};
```



GRUNT

SOME OF THE LIBRARIES THAT ARE USING GRUNT

- jQuery
- Yeoman
- Modernizr
- Adobe
- Twitter
- Filament Group
- Bocoup2



Demo for Grunt

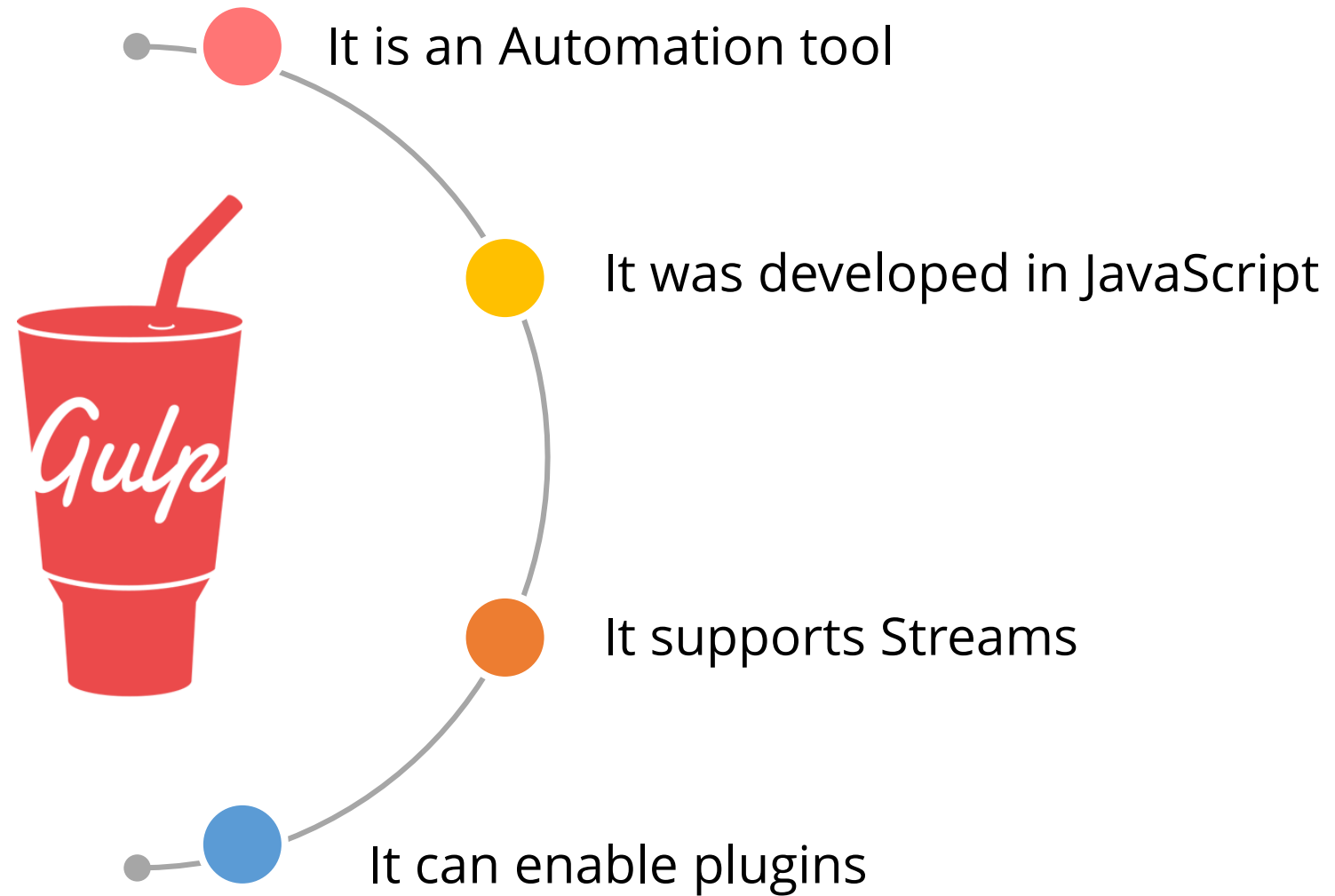


Topic 2—Working with Gulp

Topic 2—Working with Gulp

Working with Gulp

WHAT IS GULP?



Working with Gulp

USES OF GULP

Concatenate multiple CSS files

Preprocessing of SASS, LESS, CoffeeScript, and so on

Automate Testing

Trigger tasks when the code changes

Custom plugins

Working with Gulp

INSTALLING GULP

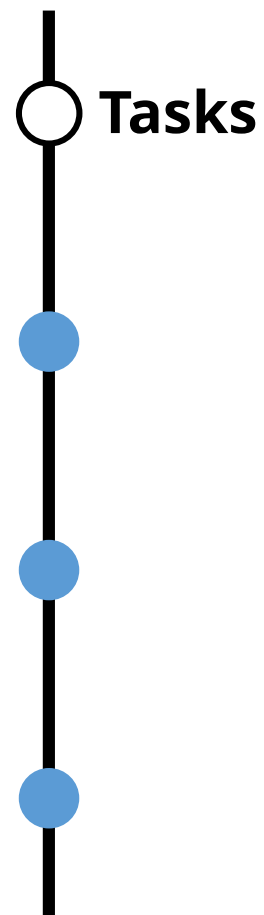
```
----- install gulp  
> npm install -g gulp  
> npm install gulp --save-dev
```

```
----- configure tasks  
> nano gulpfile.js
```

```
----- run the file  
> gulp
```

Working with Gulp

FEATURES OF GULP

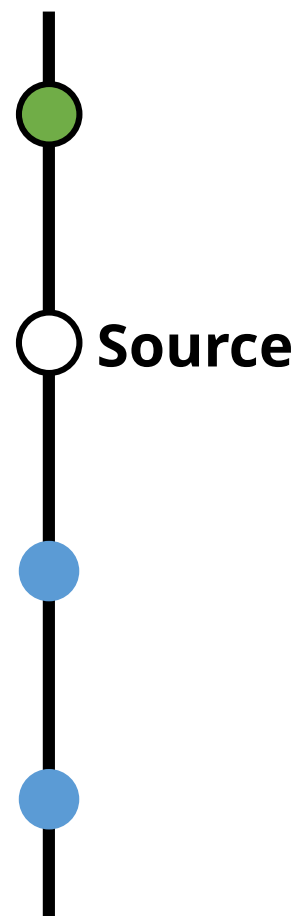


```
gulp.task('js', function () {  
  ...  
});  
  
gulp.task('images', function () {  
  ...  
});  
  
gulp.task('server', function () {  
  ...  
  pipe(livereload()); // automatic browser reload  
});  
  
gulp.task('default', function () {  
  ...  
});
```

You can run multiple tasks together in Gulp.

Working with Gulp

FEATURES OF GULP



```
webapp/
├── build
│   ├── images
│   ├── css
│   └── js
│       └── vendor
├── node_modules
│   ├── gulp
│   ├── gulp-concat
│   ├── gulp-html-replace
│   ├── gulp-notify
│   ├── gulp-uglify
│   └── gulp-uglifycss
├── src
│   ├── images
│   ├── css
│   └── js
│       └── vendor
├── package.json
└── gulpfile.js
```

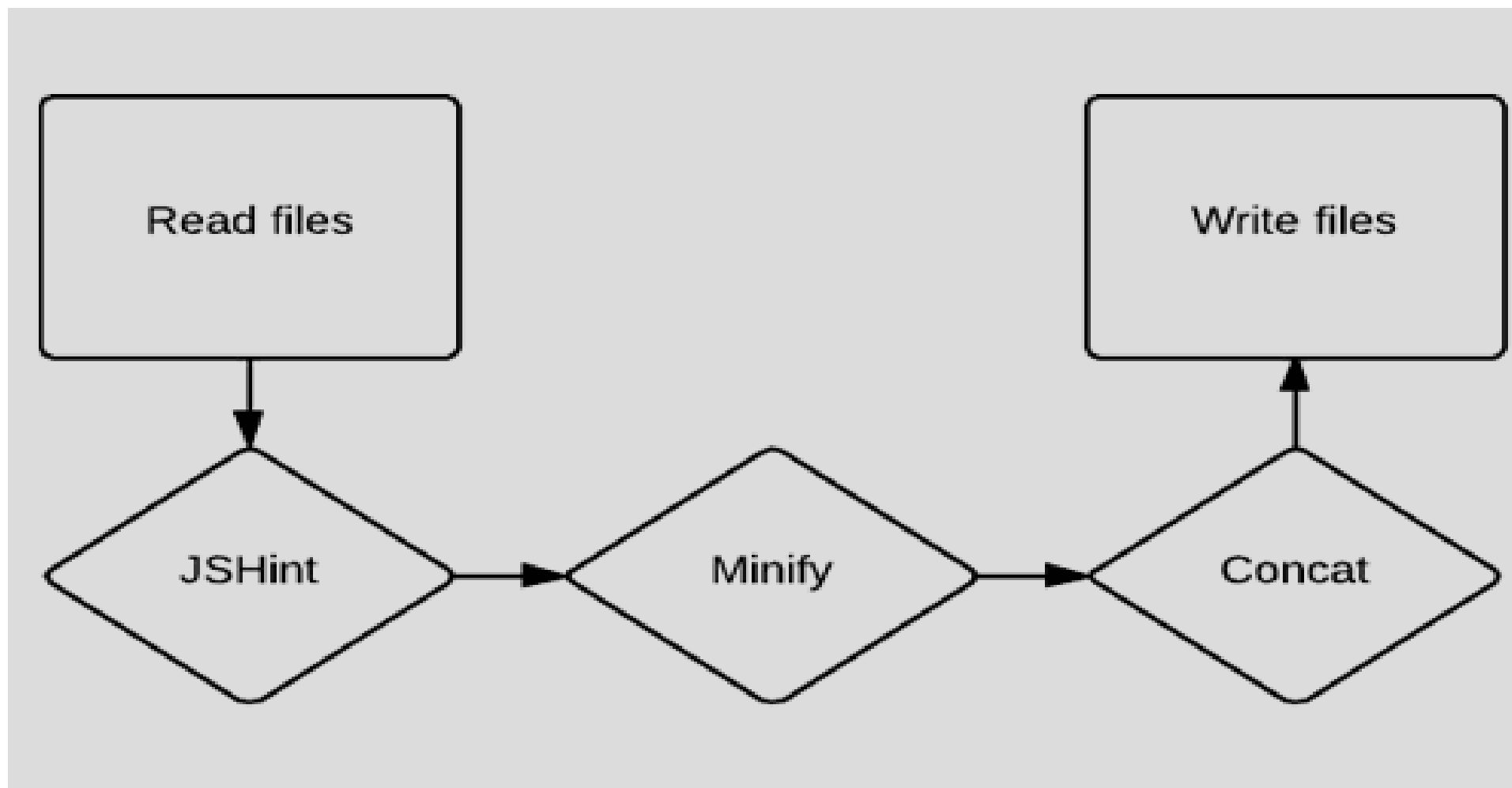
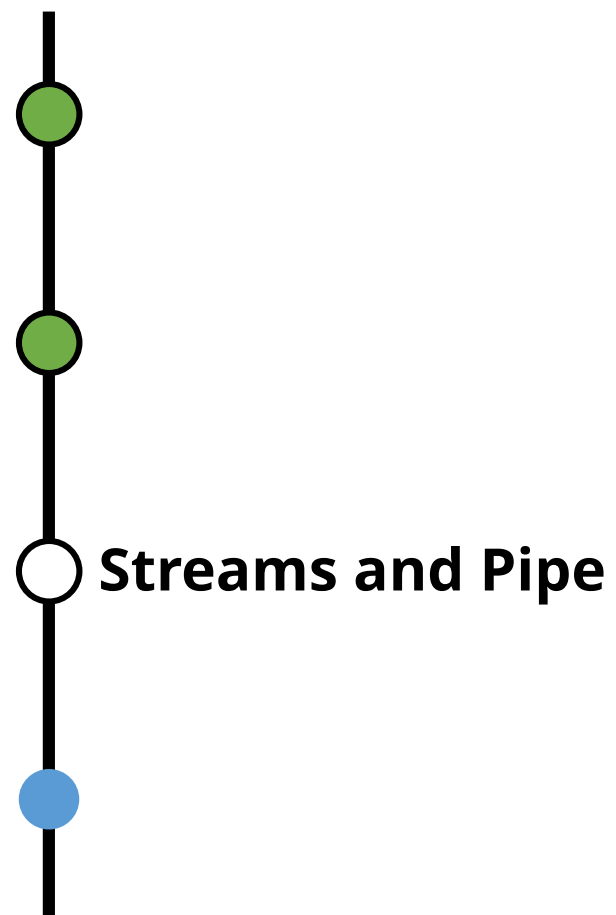
```
// it will process all js files and save to  
build/js
```

```
gulp.task('js', function () {  
    return gulp.src('src/js/*.js')  
        .pipe(gulp.dest('build/js'));  
});
```

It will process all JavaScript files and save to 'build/js'.

Working with Gulp

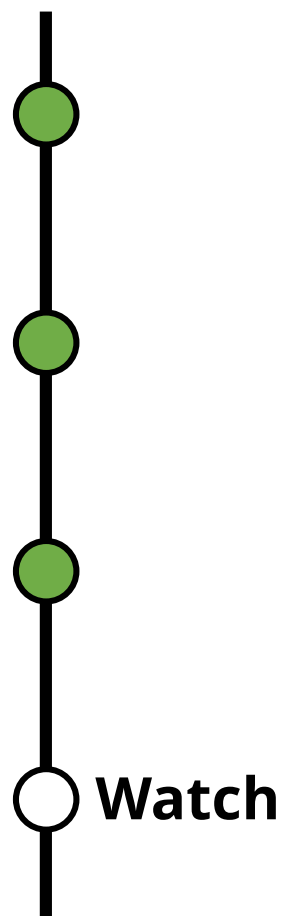
FEATURES OF GULP



Streams and Pipes can be used in Gulp.

Working with Gulp

FEATURES OF GULP



```
gulp.task('setWatchTask', function () {  
  gulp.watch('js/*.js', ['build']);  
});  
  
gulp.task('build', function () {  
  ...  
});
```

Watch method is used to monitor your source files.

Working with Gulp

PLUGINS IN GULP

You can install all related plugins listed below:

----- install all related plugins

```
"devDependencies": { "gulp-concat": "^2.2.1",  
                    "gulp-jshint": "^1.3.0",  
                    "gulp-rename": "^1.1.0",  
                    "gulp-sass": "^0.6.1",  
                    "gulp-uglify": "^0.3.1" }
```

Working with Gulp

PLUGINS IN GULP - CONCAT

Concat plugin simply combines all source files into a stream, similar to the cat method in Unix.

```
var concat = require('gulp-concat');

gulp.task('js', function () {

    return gulp.src('js/*.js')
        .pipe(concat('app.js'));
        .pipe(gulp.dest('build'));

});
```

Working with Gulp

PLUGINS IN GULP - UGLIFY

Uglify is a very common minification extension for gulp. This is used to concatenate operation.

```
var concat = require('gulp-concat');

gulp.task('js', function () {

    return gulp.src('js/*.js')
        .pipe(concat('app.js'));
        .pipe(gulp.dest('build'));

});
```

Working with Gulp

PLUGINS IN GULP

Some of the additional plugins are:

```
imagemin
```

```
gulp-coffee/sass/less
```

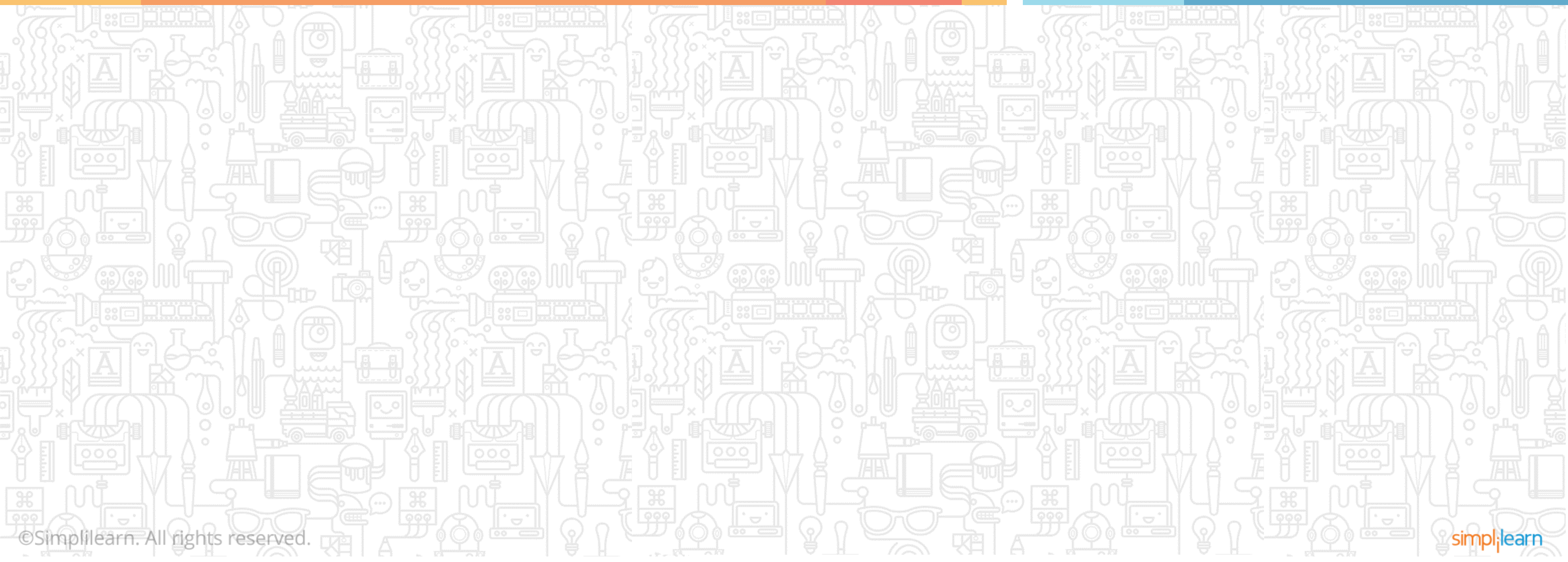
```
livereload
```

```
gulp-util
```

```
compress/decompress
```

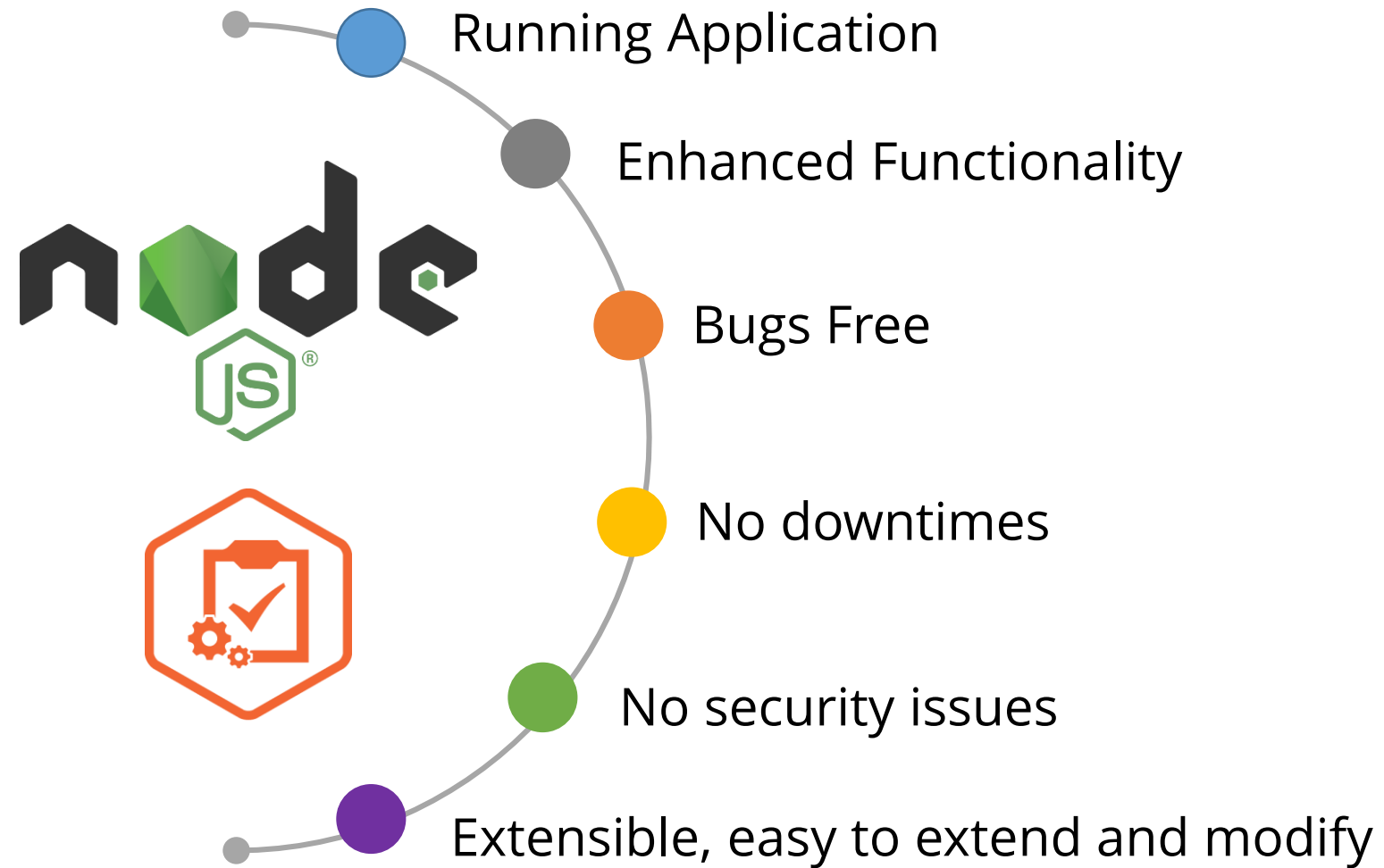
Working with Task Managers and Unit Testing

Topic 3—Node.js Unit Testing



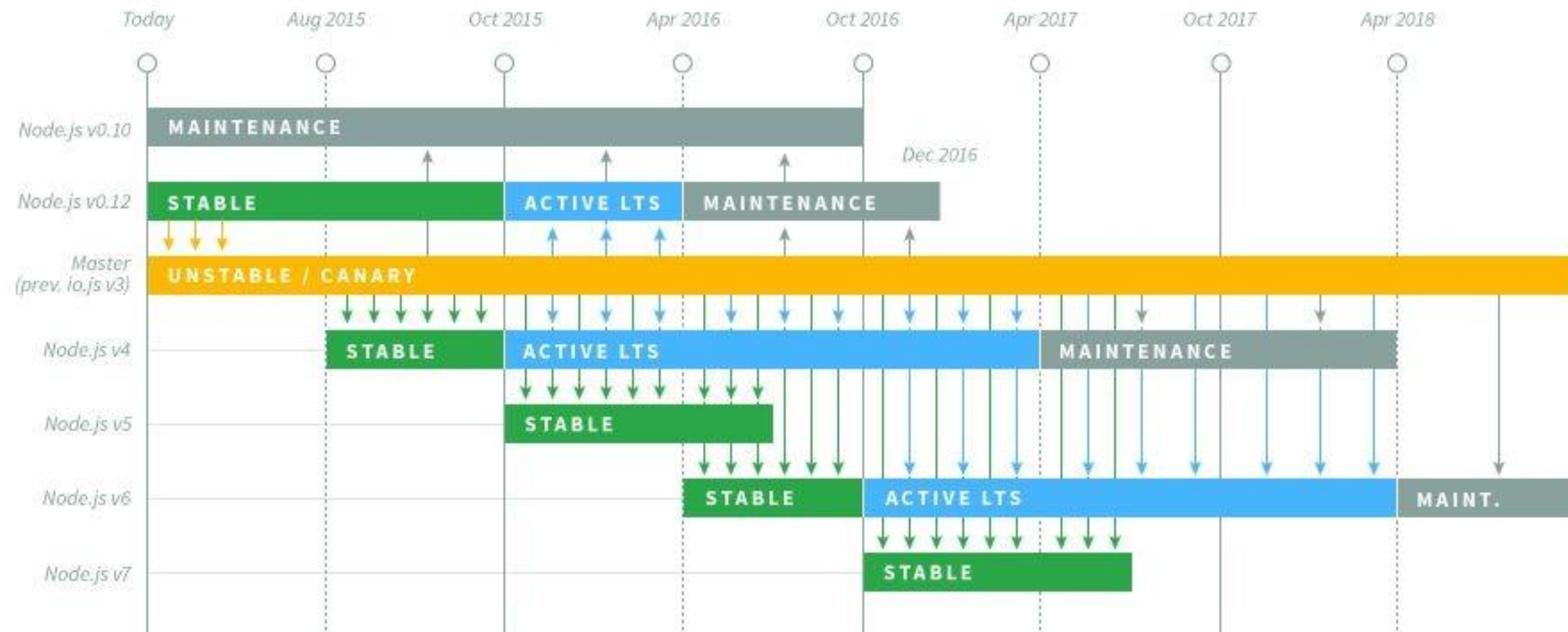
Node.js Unit Testing

WHY NODE.JS UNIT TESTING



Node.js Unit Testing

Node.js Long Term Support Release Schedule



COPYRIGHT © 2015 NODESOURCE, LICENSED UNDER CC-BY 4.0

Node.js Unit Testing

MANUAL TESTING

```
// cal_multiply.js  
  
var mul = require('./mul');  
  
var tm = mul(61, 71);  
  
console.log(tm);
```

Node.js Unit Testing

INSTALLING MOCHA AND CHAI

Installation steps:



```
npm install -g mocha
```



```
npm install chai --save-dev
```

Node.js Unit Testing

CHAI



```
var expect = require('chai').expect; var mul =  
require('./multiply');
```

```
describe('mul', () => {  
  it('should return 56 for 8 * 7', () => {  
    expect(mul(8,7)).to.equal(56);  
  });  
});
```

Node.js Unit Testing

CHAI STYLES

There are two styles in Chai: Assertive style and BDD style.

Assertion style:

```
require('chai').assert => assert.equal(expected, actual);
```

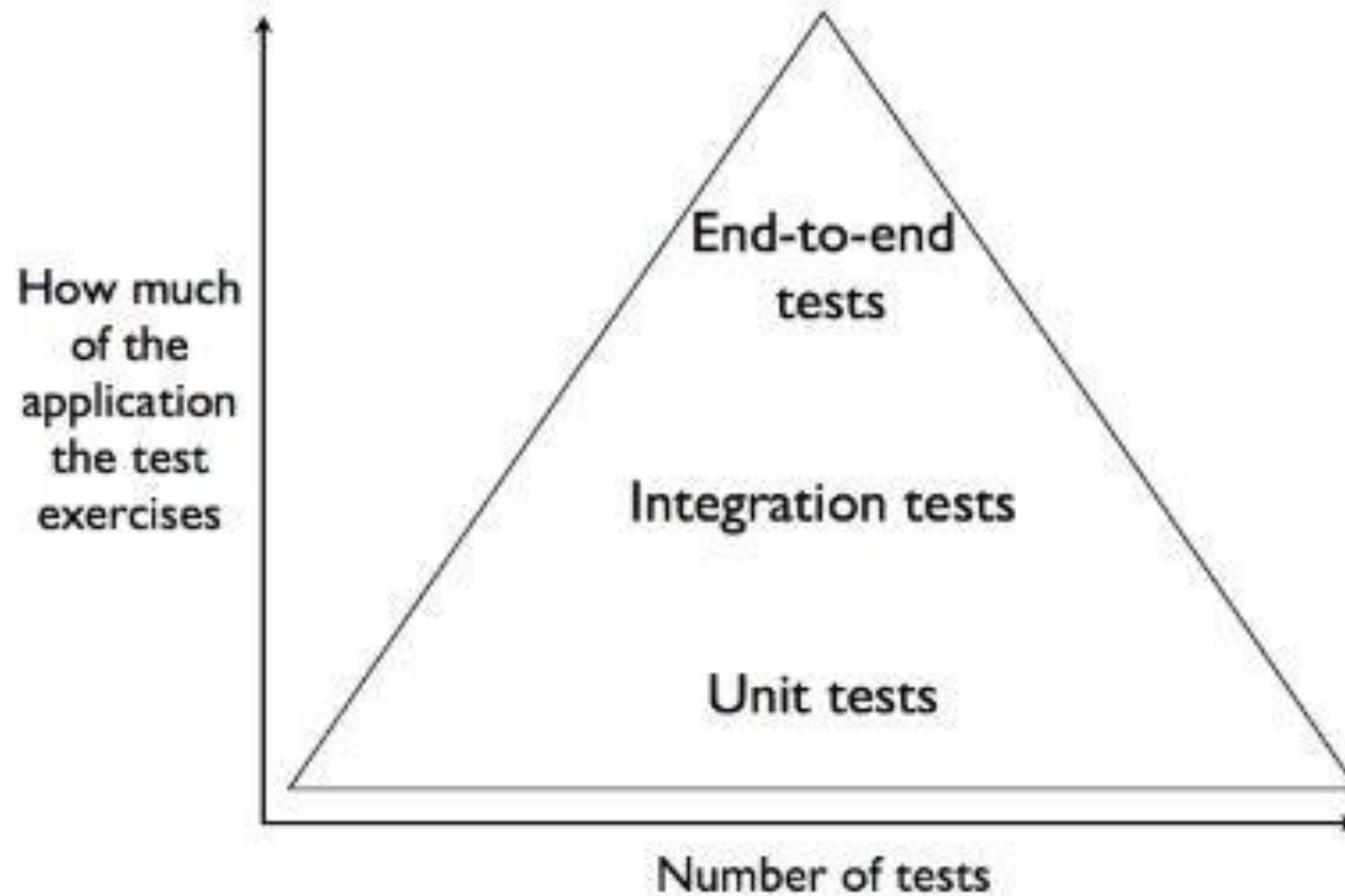
- Similar to built in assert, but more assertions
E.g.: `assert.typeOf()`, `assert.lengthOf()`

BDD style:

- `require('chai').expect => expect(actual).to.equal(expected);`
- `require('chai').should() => actual.should.equal(expected);` // don't do!
- E.g.: `expect(x).to.be.null; expect(obj).to.be.instanceOf(ctor);`
`expect(foo).to.not.equal(bar);`

Node.js Unit Testing

TESTING GRAPH



Node.js Unit Testing

Node.js Unit Testing is:

Extremely Fast: Executes in milliseconds!

Small in size: Tests only a single piece of functionality at a time

Simple to Implement: Easy and convenient to implement

Isolated: Usually have no dependencies between each other

Readable: Can be accessible to others as well

Clean Code: Same code quality as the software they are testing



**QUIZ
1**

What job can a task manager perform?

- a. Minifying and concatenating JavaScript and CSS files
- b. CSS Pre-processing
- c. Image optimisation and Unit testing
- d. All of the Above



**QUIZ
1**

What job can a task manager perform?

- a. Minifying and concatenating JavaScript and CSS files
- b. CSS Pre-processing
- c. Image optimisation and Unit testing
- d. All of the above



The correct answer is **d. All of the above**

Minifying and concatenating JavaScript and CSS files, CSS Pre-processing, Image optimisation, and Unit testing are the jobs that a task manager can perform.

QUIZ
2

Grunt supports custom plugins.

- a. True
- b. False



**QUIZ
2**

Grunt supports custom plugins.

- a. True
- b. False



The correct answer is **a. True.**

Grunt supports custom plugins.

QUIZ
3

___ is the filename used for configuring tasks.

- a. gulpfile.js
- b. gulp_file.js
- c. gulp_file.js
- d. All of the Above



QUIZ
3

___ is the filename used for configuring tasks.

- a. gulpfile.js
- b. gulp_file.js
- c. gulp_file.js
- d. All of the Above



The correct answer is **a. gulpfile.js.**

gulpfile.js is the filename used for configuring tasks.

QUIZ
4

Which Module in gulp is used for Minifying the js file?

- a. gulp-minify
- b. gulp-uglify
- c. gulp-zip
- d. All of the Above



**QUIZ
4**

Which Module in gulp is used for Minifying the js file?

- a. gulp-minify
- b. gulp-uglify
- c. gulp-zip
- d. All of the Above



The correct answer is **b. gulp-uglify.**

gulp-uglify module in gulp is used for Minifying the js file.

Key Takeaways



- ✓ Grunt is a task-oriented command line build tool for JavaScript based Application.
- ✓ Gulp is an automated tool that was developed in JavaScript.
- ✓ Node.js testing can be done using Mocha and Chai.



Thank You