What is Grunt?

In 2012, JavaScript developer Ben Alman launched the command task runner Grunt, which he has been managing and developing with a small team ever since. The program's code, which is available under the **MIT license**, can be downloaded from the official website and is also available to use for free at GitHub. As with other comparable tools, Grunt is based on the JavaScript runtime environment, Node.js, and, thanks to **grunt-cli**, offers its own command line interface, which can be installed via the **Node Package Manager** like the task runner itself.

To integrate Grunt into a web project, two files must be added to the root directory of that project. The file *package.json* typically contains **all relevant metadata** about the automatable tasks, such as the name, version, license, or related dependencies. Located in the valid JavaScript (*gruntfile.js*) or CoffeeScript (*gruntfile.coffee*), the

Grunt file contains the code used to integrate and configure individual **task modules**. The modules, also known as **plugins**, are installed via the npm package managements mentioned above.

The official plugin folder counts **well over6000 Grunt tasks** for various purposes, such as *grunt-contrib-clean* (cleans files and folders), *grunt-contrib-cssmin* (CSS minimization), and *grunt-contrib-connect* (starts local web servers). Many of these plugins are in principle no different to interfaces to standalone tools. For example, *grunt-contrib-uglify*, allows users to control the JavaScript minimizer UglifyJS from the Grunt file.

In addition to the option to modify or extend existing tasks, experienced developers can also **create their own automation modules** to adjust the task runner to suit their specific requirements.

What is Gulp?

Gulp is a free task runner, initially launched in July 2013 by American software company Fractal Innovations in collaboration with the GitHub community. As with Grunt, the program is available under an open source **MIT license.** It is based on the JavaScript platform **Node.js** and, like its competitor, uses the **npm** package manager. Regarding the structure, Grunt and Gulp are relatively similar to one another; Gulp is also a command line tool, so it has a suitable user interface with **gulp-cli.** The *package.json* configuration file and the gulpfile (*gulpfile.js*), which lists possible tasks, are also usually used. If both are added to the web directory, the task runner can be used for **workflow optimization.**

The gulpfile, which contains JavaScript only, combines individual file operations so that they can be executed via the **node.js module stream**. Most of the individual stream processes run in memory before the result is written back to the respective files at the end. This is one of the reasons why the task runner is recognized for

its **excellent performance.** Since the tasks are programmed from the beginning, but not configured, **proficiency innode.js and JavaScript is a basic requirement** to use Gulp. For professionals, this means a lot of freedom, but it also increases the margin of error.

Gulp also has a number of premade tasks in plugin form. In the official directory on the gulp.js website, users can find over 3000 extensions, such as *gulp-jekyll* (a compilation of Jekyll projects), *gulp-php-minify* (optimization of PHP codes), and *gulp-cssmyicons* (for converting SVG icons into CSS).

Gulp vs. Grunt: similarities and differences at a glance

At a first glance, Grunt and Gulp don't seem to differ much from one another in their basic functions; both automation tools are available under the MIT license, so the **source code is open** and freely available. Both applications **can be controlled from the command line** and have their own interface installed along with it. The task runners also use the same package manager, **npm.** Thanks to their large plugin directories, Grunt and Gulp can both easily **automate a huge number of tasks.** If there are no extensions for the desired process, they can be programmed with both tools, although due to the

complex structure, both task runners require **knowledge of JavaScript and node.js.**

However, while Gulp is primarily based on the node.js module stream, Grunt mainly uses the fs (file system) module, which highlights one of the most important differences between the two tools: **Grunt is strictly file-oriented** and creates temporary local files during the execution of the tasks. **Gulp**, on the other hand, **handles the processes via the memory** and writes them in the target file immediately, giving the program a speed advantage.

A second distinguishing feature is the respective concept of the two solutions. Grunt's programming and structure gives users some direction; **completed tasks located there are already defined** and must then simply be configured. In comparison, Gulp allows far more space for independent programming by **providing only the individual modules.** While on the one hand, this makes it easier to understand the backgrounds and contexts, it also demands much more from the user. The bigger a project is, the more **Gulp's** strengths come into play, which is why the new task runner is now **the first choice for many people.** Thanks to the lower requirements, however, **Grunt** is still a valuable tool for **smaller, manageable projects.**

//EZEBUIRO UCHECHUKWU VINCENT