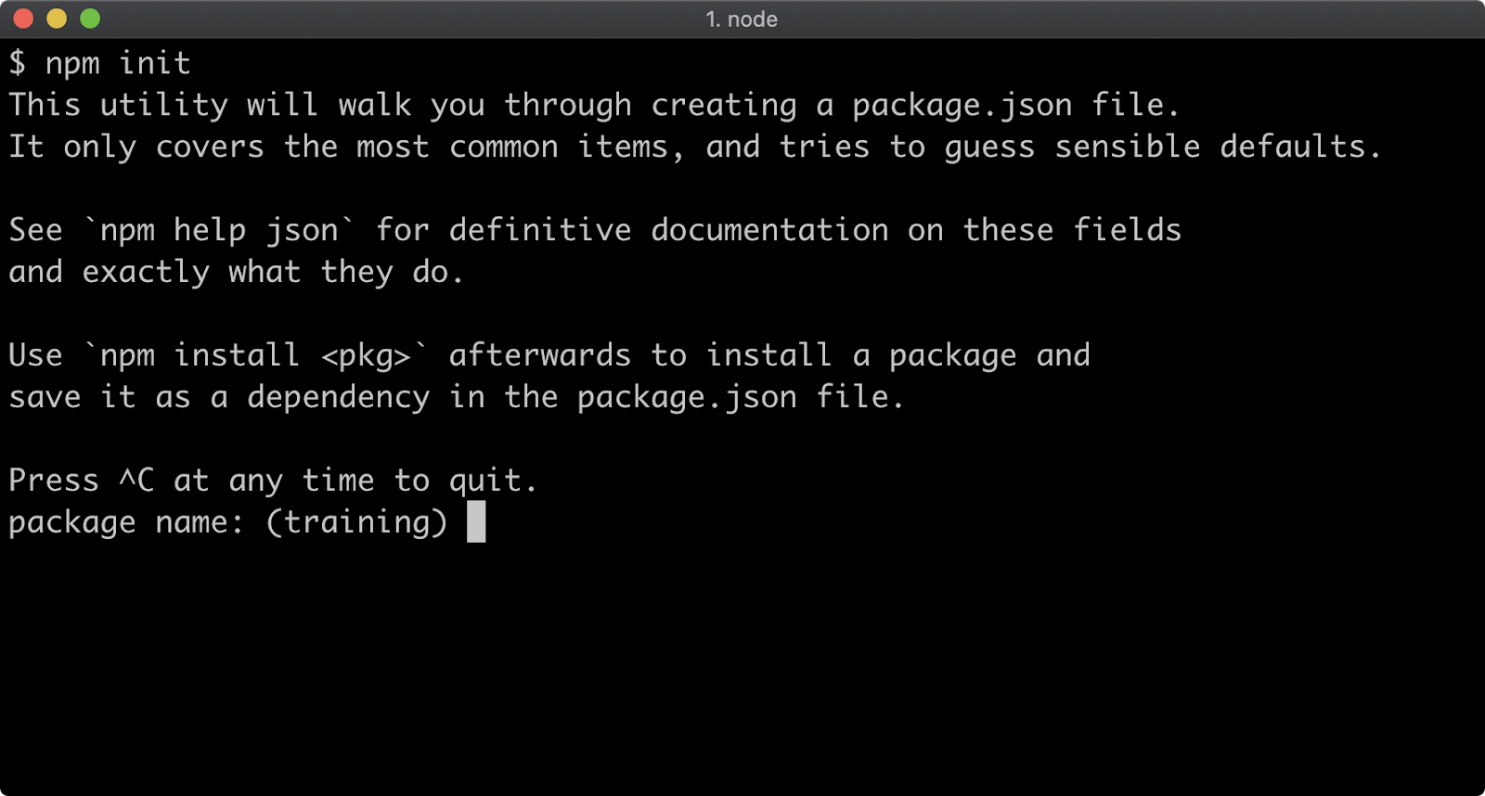
**Initializing a Package**

A package is a folder with a **package.json** file in it (and then some code). A Node.js application or service is also a package, so this could equally be titled "Initializing an App" or "Initializing a Service" or generically, "Initializing a Node.js Project".

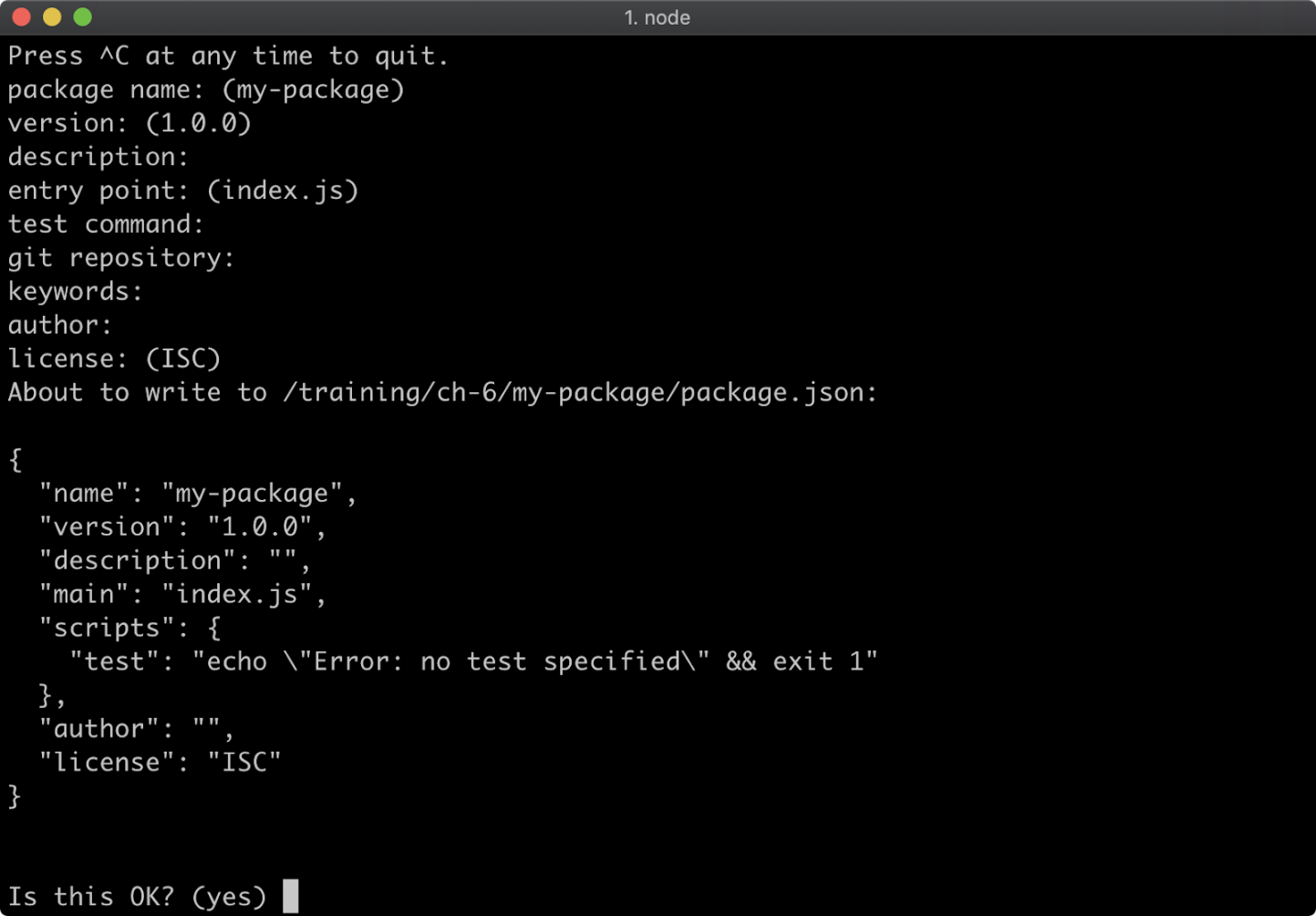
The **npm init** command can be used to quickly create a **package.json** in whatever directory it's called in.

For this example a new folder called **my-package** is used, every command in this section is executed with the **my-package** folder as the current working directory.

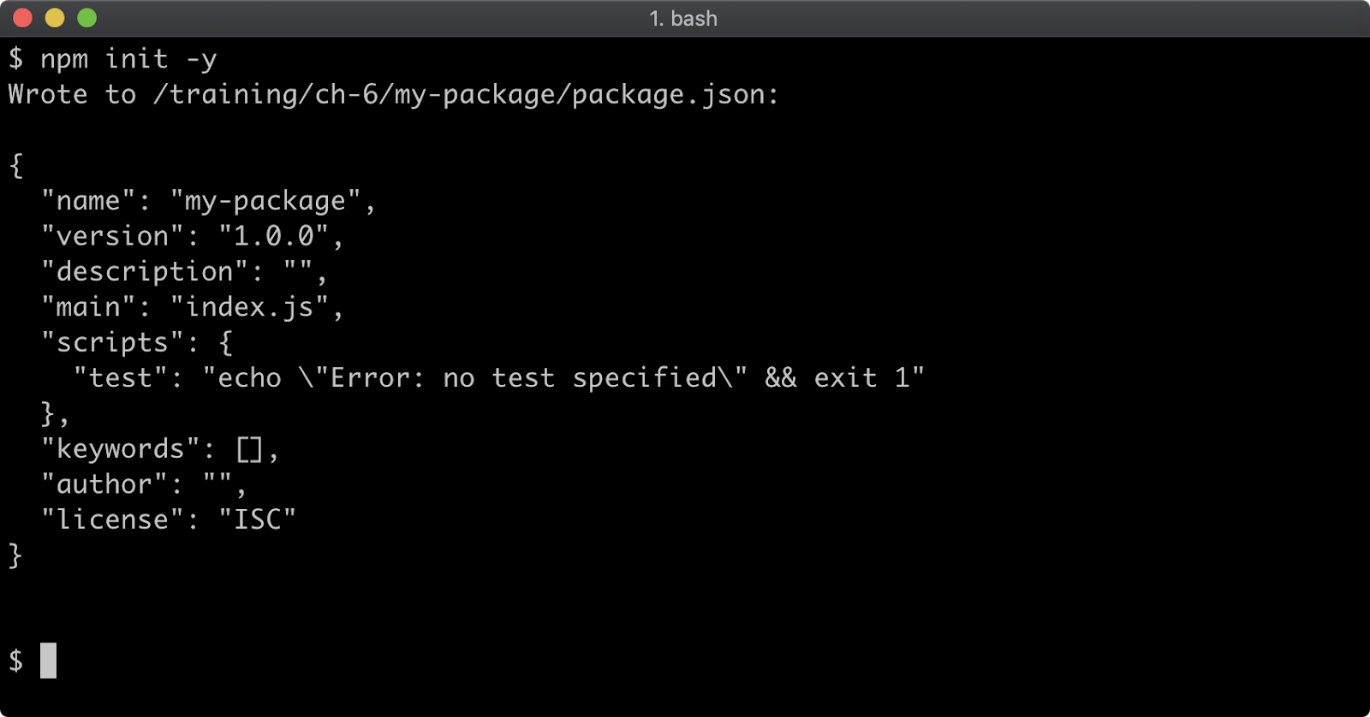
Running **npm init** will start a CLI wizard that will ask some questions:



For our purposes we can hit return for every one of the questions.



A shorter way to accept the default value for every question is to use the **-y** flag:



The default fields in a generated **package.json** are:

* **name** – the name of the package
* **version** – the current version number of the package
* **description** – a package description, this is used for meta analysis in package registries
* **main** – the entry-point file to load when the package is loaded
* **scripts** – namespaced shell scripts, these will be discussed later in this section
* **keywords** – array of keywords, improves discoverability of a published package
* **author** – the package author
* **license** – the package license.

The **npm init** command can be run again in a folder with an existing **package.json** and any answers supplied will update the **package.json**. This can be useful when the package has also been initialized as a git project and has had a remote repo added. When run in a git repository, the **npm init -y** command will read the repositories remote URI from git and add it to **package.json**.