Package.json

The package.json is the project manifest file. It has all the meta data about the project such as the descriptions, license, location, dependencies, scripts to build, launch and run. Consider this example which has a few dependencies:

```
{
  "name": "my-cool-app",
  "version": "0.1.0",
  "description": "A great new application",
  "main": "server.js",
  "dependencies": {
    "express": "~4.2.0",
    "ws": "~0.4.25"
  },
  "devDependencies": {
    "grunt": "~0.4.0"
  }
}
```

In most cases, it's easy to tell what modules are required and what are the main commands and files to execute just by looking at the package.json file.

Package.json is *required* for npm modules.

Main Properties

Module packaging in Node is done using a `package.json` file. There are many options that can be configured:

- name
- version number
- dependencies
- license
- scripts
- etc

Creating package.json

To create a package.json file, run `npm init` command and answer the questions that appear:

```
$ npm init

This utility will walk you through creating a package.json
file.  It only covers the most common items, and tries to
guess sane defaults.

See `npm help json` for definitive documentation on these
fields and exactly what they do.

Use `npm install <pkg> --save` afterwards to install a package
and save it as a dependency in the package.json file

Press ^C at any time to quit
name: (my-package-name)
```

If you are okay with the default answers to these questions, then you can skip the questions and answer yes to all of them automatically by using `-y` flag, as in `npm init -y`.

Private Modules

The private attribute prevents accidental publishing

```
{
  "name" : "my-private-module",
  "version": "0.0.1",
  ...
  "private": true,
  ...
}
```

When to use `-g` for global installations?

Only use `-g` for command-line tools which you run from the Terminal /Command Prompt. They usually have `bin` in package.json:

```
{
  "name": "stream-adventure",
  "version": "4.0.4",
  "description": "an educational stream adventure",
  "bin": {
    "stream-adventure": "bin/cmd.js"
  },
  "dependencies": {
    ...
```

In other words, anything which you plan to import with `require()` must be local in node_modules NOT in global.