



## Comandos basicos

- \$ npm init | Sirve para crear el archivo package.json el cual contiene toda
- \$ npm init -y | la información acercaada nuestro proyecto
- \$ npm set init.author.email <email> | Asignar nuestros datos como en 'git',
- \$ npm set init.author.name <name> | pero no es tan obligatorio hacer esto.

# Instalación de dependencias

Las dependencias se deben instalar en nuestra carpeta raíz de nuestro proyecto.

\$ npm i <pkg>

\$ npm install <pkg> | npm install <pkg> <sup>(por defecto)</sup> **--save** | npm install <pkg> **-S**

Por defecto se instala como una independencia **requerida** para el proyecto, es decir, que paquete que instalas es necesario para vivir en producción

**\$ npm install <pkg> -D | npm install <pkg> --save-dev**

Este flag nos va a permitir establecer que paquete que vamos a instalar solo es necesario en nuestro entorno local o el entorno de desarrollo.

**\$ npm install <pkg> -g**

Instalar un paquete de forma global. Esto permite que podamos utilizar este paquete en diferentes proyectos. Por lo general, se deben instalar estos paquetes con permisos de administrador.

Si no queremos colocar permisos de administrador a cada rato, podemos seguir esta guía: <https://docs.npmjs.com/resolving-eacces-permissions-errors-when-installing-packages-globally>

`$ npm list -g --depth 0` ( profundidad )

Ver los paquetes que están instalados de forma global.

`$ npm list`

Para listar los paquetes que tiene un proyecto en específico

`$ npm install <pkg> -O`

Podemos instalar de forma opcional un paquete con este comando.

`$ npm install <pkg> --dry-run` ( simula la instalación )

Este flag indica que el paquete no va a ser instalado dentro del proyecto, simplemente es una simulación, nada más nos muestra el output como si se fuese instalado