

## 23 - Usando o npm

- O npm é disponibilizado junto com o Node.js, ou seja, ao baixar e instalar o Node.js você também terá o npm. Caso ainda não possua o Node.js e o npm, instale-os.
  - No Linux você pode instalar o Node e o npm utilizando o gerenciador de pacotes da sua distribuição (apt para Debian/Ubuntu, yum para Fedora, pacman para Arch, etc)
  - No Linux também é possível usar o asdf, um gerenciador de versões, para instalar o Node e o npm. Passo a passo da instalação: <https://asdf-vm.com/guide/getting-started.html>
  - No Windows a maneira mais fácil de instalar o npm é através do site <https://nodejs.org/pt-br/>, baixando a versão LTS (versão mais estável, menos sujeita a bugs e falhas de segurança)
  - **Atenção!** Se você utiliza o WSL deve instalar o Node e o npm dentro da sua distribuição Linux do WSL, seguindo o método de instalação para Linux

1. Uma vez que tenha o npm instalado, vamos começar a utilizá-lo abrindo a linha de comando e testando a instalação. Execute o comando abaixo e ele deverá retornar a versão do npm:

```
npm --version
```

2. Agora já podemos acessar uma pasta e criar um projeto gerenciado pelo npm com o seguinte comando:

Obs.: Veja que ele criou o arquivo "package.json", um arquivo com informações do nosso projeto.

```
npm init
```

3. Para instalar um novo módulo no nosso projeto usamos o comando:

Obs.: Veja que isso criará a pasta "node\_modules", que é onde todos os nossos módulos instalados ficam localizados.

```
npm install --save lodash
```

4. Já podemos criar um arquivo "index.js", importar o módulo lodash e utilizá-lo:

Obs.: Repare que estamos trabalhando com o Node.js e não com o browser, então vamos usar o CommonJS.

```
const _ = require("lodash") console.log(_.isArray([]))  
console.log(_.kebabCase("Isaac Pontes"))
```

5. Outra funcionalidade importante é a instalação de dependências de desenvolvimento, o que pode ser feito com a flag:

Obs.: Uma dependência de desenvolvimento é um módulo que o nosso projeto só utiliza enquanto estamos codificando ele, para ele se usado de verdade ela não será necessária.

```
npm install --save-dev eslint
```

6. Um último recurso interessante para mencionar nessa nossa introdução são as dependências globais. Uma dependência global é um módulo que não fica vinculado a um projeto específico, mas é instalado no seu computador para você usar onde e quando quiser. Para instalar um módulo globalmente usamos o comando com a flag:

```
npm install --global sass
```

7. E por fim, também podemos remover módulos manualmente excluindo-os do arquivo "package.json" ou usando o comando:

```
npm uninstall eslint
```