**NODE**

**NPM – Node Package Manager**

**Criar npm**

npm init -y

**Installar pacotes:**

npm install {nome\_do\_pacote}

npm install express

**Atualizar pacotes:**

npm update

**Alterar a dependencies de um package (dev, prod):**

npm install express --save-dev // muda para devDependencies

npm install express --save-prod // muda para dependencies

**Instalar versão específica de package:**

npm install express@2.1.0 -E // -E para remover o ‘^’

**Desinstalar um pacakge:**

npm uninstall express

**Listar packages:**

npm ls // mostra os packages e suas dependencias

npm ls -depth=0 // mostra somente os packages instalados

npm outdated // mostra os packages desatualizados

**MANIPULANDO ARQUIVOS – FS**

**Leitura:**

Importando o módulo:

const *fs* = require('fs').*promises*; // importa o módulo fs como promises

Criamos uma função para ler os arquivos e pastas de um caminho:

async function readdir(*rootDir*) {

*rootDir* = *rootDir* || path.resolve(\_\_dirname);

    const files = await *fs*.readdir(*rootDir*);

    walk(files, *rootDir*);

}

**\*IMPORTANTE:** fs.readdir retorna os arquivos e pastas do caminho informado

Na função walk, criamos uma lógica com fs.stat (que retorna status do arquivo lido) para podermos criar uma leitura recursiva dos itens.

async function walk(*files*, *rootDir*) {

    for (let file of *files*) {

// cria um caminho completo do arquivo

        const fileFullPath = path.resolve(*rootDir*, file);

        const stats = await *fs*.stat(fileFullPath);

        if (/\.git/g.test(fileFullPath)) continue; // re

        if (/node\_modules/g.test(fileFullPath)) continue; // re

// se o arquivo for uma pasta, ele recomeça a leitura

        if (stats.isDirectory()) {

            readdir(fileFullPath);

            continue;

        }

        if (!/\.html$/g.test(fileFullPath)) continue; // re

        console.log(fileFullPath);

    }

}