

NODE.JS



Node.js é um ambiente de **execução de JavaScript no lado do servidor**. Ele permite que os desenvolvedores usem JavaScript para escrever scripts do lado do servidor, produzindo conteúdo dinâmico de páginas da web antes de serem enviadas ao navegador do usuário. Node.js é construído sobre o motor JavaScript V8 do Google Chrome, que o **torna muito rápido na execução de código**. Além disso, é conhecido por sua arquitetura orientada a eventos e não bloqueante, o que facilita a criação de aplicações escaláveis e de alta performance.

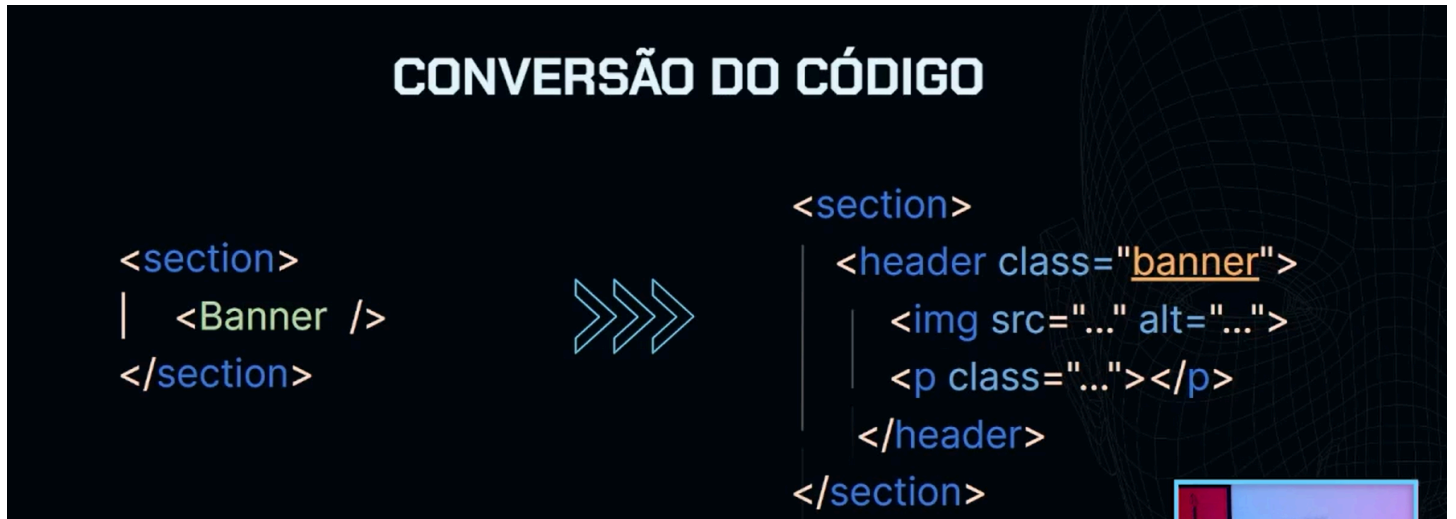
O JavaScript puro é executado no navegador, como exemplo o Edge. Já o Node.js se trata de um novo ambiente de execução, será lá que o código funcionará.

Qual a motivação de sua criação?

- Executar operações locais no computador:
 - Criação de servidores e API's;
 - Leitura e escrita de arquivos;

- Compilação de códigos;
- Automatização de processos

QUAL SUA FUNCIONALIDADE?



Na imagem acima é possível verificar que uma linha de código é interpretada por várias, ou seja, a linha Banner representa a linha " class = "banner" "

Comandos

Power Shell - " node -v " [Esse comando serve para verificar no terminal(Power shell) qual a versão do node.js que está na máquina]

Pwd - Serve para Imprimir o diretório atual. É importante para saber onde estou localizado no máquina

- Diretório = Pasta

```
PS C:\Users\jesus> pwd
Path
----
C:\Users\jesus
```

Mkdir - Tem como funcionalidade a criação de uma pasta

```
PS C:\Users\jesus> mkdir exemplo-node

Diretório: C:\Users\jesus

Mode                LastWriteTime         Length Name
----                -
d-----         12/08/2024    09:41             exemplo-node
```

cd - Quando quero mudar de pasta

code . - Caso queira abrir aquela pasta do terminal no visual

Ferramentas para manutenção do código

ESLINT - O eslint serve para corrigir os erros apresentados no código JAVASCRIPT

Ctrl + shift + ' = Novo terminal

Ctrl + L = Limpa a tela

NPM: Node Package Manager - Gerenciador de pacotes node

Seta para cima no terminal - Serve para copiar o ultimo código digitado

Comandos de ESLINT

npx eslint "script.js" - Representa o código que quer percorrer e filtrar o código para encontrar possíveis erros de uma pasta só, neste caso a "Script.js". Veja:

```
PS C:\Users\jesus\Documents\alura\NODE-JS\Atividade\3519-nodejs-vidflow-projeto-base> npx eslint
C:\Users\jesus\Documents\alura\NODE-JS\Atividade\3519-nodejs-vidflow-projeto-base\script.js
  3:13  error  'calopsita' is not defined  no-undef

✖ 1 problem (1 error, 0 warnings)

PS C:\Users\jesus\Documents\alura\NODE-JS\Atividade\3519-nodejs-vidflow-projeto-base>
```

Na imagem acima é importante entender que foi encontrado um erro que foi colocado de propósito apenas para teste.

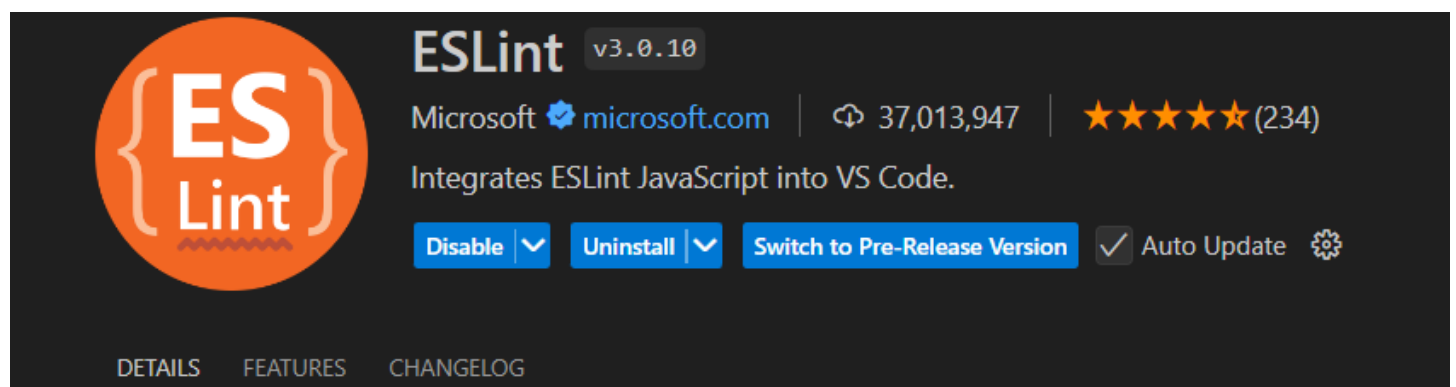
npx eslint . - Já este comando colocado no terminal, significa que ele percorre todos os arquivos necessários.

npm init

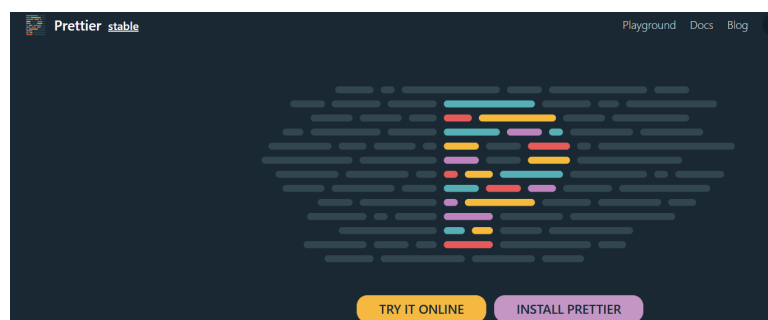
Como você conferiu, ao executar apenas `npm init`, o terminal se torna interativo e um arquivo `package.json` é criado na pasta atual, de acordo com as respostas do terminal.

Outra forma de usar esse comando é executando `npm init --yes` ou `npm init -y` para pular todas as perguntas do terminal e criar o `package.json` imediatamente.

EXTENSÃO ESLINT



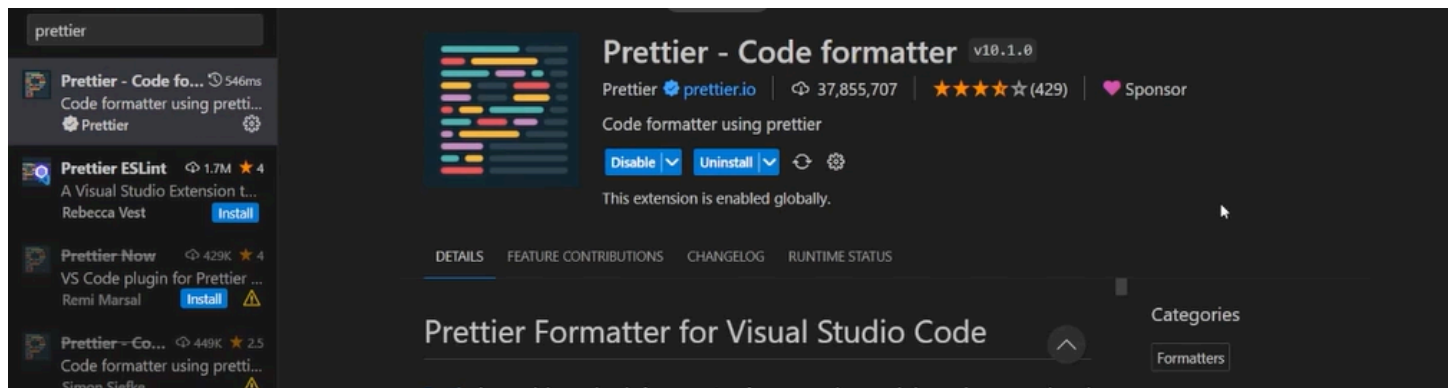
FORMATAÇÃO E PADRONIZAÇÃO DE CÓDIGO



Instalar como dependência desse projeto - `npm install --save-dev --save-exact prettier`

Saber a Versão - `npx prettier -v`

Extensão - Prettier



Versionamento

0.14.4

4 - Versão de correção/Atualização (Resolver bugs, problemas e tudo mais)

14 - Versão menor (Talvez traga novos recursos)

1. Versão Maior (Muda a base, como por exemplo a sintaxe)

O versionamento semântico em Node.js, e em geral, é uma convenção para atribuir versões a projetos de software, expressa no formato ``MAJOR.MINOR.PATCH``:

1. MAIOR: Incrementado quando há mudanças incompatíveis na API.

2. MENOR: Incrementado quando novas funcionalidades são adicionadas de forma compatível com versões anteriores.

3. PATCH: Incrementado para correções de bugs que mantêm a compatibilidade.

Por exemplo, a versão ``2.1.4`` indica a segunda versão principal, com uma atualização menor e quatro correções de bugs desde a última versão.

QUERO DIMINUIR A SINTAXE QUE EU TESTO PARA ENTRAR NO JOGO

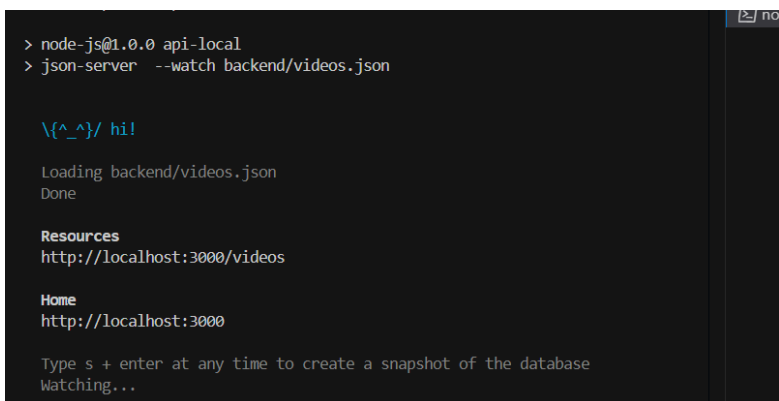
A

Em script, na pasta package.JSON, você coloca o seguinte código:

```
"scripts": {  
  "test": "echo \"Error: no test specified\" && exit 1",  
  "api-local": "json-server --watch backend/videos.json"
```

api-local representa uma nova variável, e atribuímos essa variável o link de envio para abrir no projeto

no terminal, ao invés de usarmos: **npx json-server --watch backend/videos.json**, Agora colocamos assim no terminal: **NPM RUN API-LOCAL**



```
> node-js@1.0.0 api-local  
> json-server --watch backend/videos.json  
  
\\(^_^)/ hi!  
  
Loading backend/videos.json  
Done  
  
Resources  
http://localhost:3000/videos  
  
Home  
http://localhost:3000  
  
Type s + enter at any time to create a snapshot of the database  
Watching...
```

npm install axios@1.6.0 -E