**Stage 4**

-Vamos deixar o site carregar as informações e depois acessar os elementos no DOM com o JS. Precisamos que o site esteja carregado antes de acessar o DOM. Por isso JS vai ficar no final do body.

-Use nomes de funções e variáveis mais explicativas. No mercado você trabalha com outras pessoas, elas precisam entender o código.

-**GitHub:**

\*O projeto do node.js está aberto no GitHub.

\*Você pode tornar-se contribuidor de ferramentas importantes e ganhar experiência.

\*Deixe toda a linguagem do seu perfil e seus repositórios em **inglês**.

-**API:**

**\***Lúdico:em umrestaurante, o garçom vem até a mesa, anota as informações do seu pedido. Depois caminha até a cozinha e leva o pedido até uma pessoa que lê os pedidos para o cozinheiro. Ele prepara a comida e deixa no local de saída para o garçom entregar na mesa.

\*Técnica: compra online de roupa, usuário coloca o item no carrinho e faz o checkout/pagamento com cartão de outra empresa. Os dados são inseridos no mesmo site.

Mesa = Site, Garçom = API, bloco de notas = documento JSON.

O site acessa a API, manda o JSON com as informações, a cozinha é o sistema do Mercado Pago. Processa os dados de pagamento e manda a resposta.

O site da loja não tem acesso ao sistema do Mercado Pago, a API vem do próprio Mercado Pago e permite a comunicação entre os dois.

\*O GitHub possui uma API e nós podemos acessar e pedir dados.

+Nós vamos acessar as informações públicas do seu perfil.

\***Consumir API**: Quando o sistema usuário da API faz a transação de dados. A proprietária da API: fornece dados.

**Discover** -> Curso de API com node.

\*<https://api.github.com>

+Ao entrar em uma rota (links) de dentro da API, você vê um resultado parecido com object JS.

+JSON -> resposta da API, pode ser em JSON ou URL.

+APIs podem ter conteúdo privado ou público.

+user-url

“A student focused on web development: JavaScript, HTML, CSS, Git.”

+<span></span> – forma de texto sem ser um parágrafo, na mesma linha.

-**JSON** (melhor amigo):

\*É uma forma de documento simples e leve para transacionar informações. Muito usado em APIs.

\*Sintaxe: escrita parecida com o object. Texto

Descrição gerada automaticamente

\*fetch: comando, vai até a URL da API e traz as informações do JSON.

\*promise: recurso do JavaScript para pegar respostas de algo.

.then() -> entra aqui se o fetch funcionar.

\*Transforme em json.

-**Arrow functions**:

\*Forma contraída da function (igual utilizamos para acessar o DOM).

\*Função anônima, sem nome.

1.Parâmetro/variável => {

code

}

2.(param1, param2) => {

code

}

3.() => {

code

}

-**GitHub Pages:**

\*Só funciona para páginas com HTML/CSS e JavaScript.

\*Repositório -> Settings -> Pages.

\*None -> Main -> /root or /docs -> Save

\*Cada repositório tem seu próprio GitHub Pages e deve ser habilitado.

-Senha: #SemLimites

-A tecnologia e a mente humana não têm limites. A tecnologia já avançou muito, imagina daqui pra frente com as ideias e cooperação das pessoas!