

A woman with long, dark hair and glasses is shown from the chest up, looking down at a laptop screen. The background is dark and slightly blurred.

IN

# INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 12 - PROJETO

# O QUE IREMOS APRENDER

**01**

JSON

**02**

PROJETO

IN

# JSON

JavaScript Object Notation (JSON) é um formato padrão baseado em texto para representar dados estruturados com base na sintaxe de objeto JavaScript.

É comumente usado para transmitir dados em aplicações web (por exemplo, enviar alguns dados do servidor para o cliente, para que possam ser exibidos em uma página web, ou vice-versa).



# JSON

Embora se assemelhe muito à sintaxe literal do objeto JavaScript, ele pode ser usado independentemente do JavaScript, e muitos ambientes de programação apresentam a capacidade de ler (analisar) e gerar JSON.

```
[  
  {  
    "nome": "Érick",  
    "idade": "20",  
    "sexo": "Masculino"  
  },  
  {  
    "nome": "Pedro",  
    "idade": "24",  
    "sexo": "Masculino"  
  },  
  {  
    "nome": "Raama",  
    "idade": "18",  
    "sexo": "Feminino"  
  }]
```

# DADOS JSON

Os dados JSON consistem em pares chave/valor semelhantes às propriedades do objeto JavaScript. A chave e os valores são escritos entre aspas duplas separadas por dois pontos : O objeto JSON é escrito entre chaves { } e eles podem conter vários pares chave/valor . Também podemos ter um array de objetos e é escrita entre colchetes [ ]

```
<script>
    {"chave": "valor"}
</script>
```

//Array de objetos

```
let pessoas= [
    { "name": "John", "age": 22 },
    { "name": "Peter", "age": 20 },
    { "name": "Mark", "age": 23 }
]
```

# CONVERTENDO OBJETOS EM JSON

Você pode acessar dados JSON usando a notação de ponto “.”. Você também pode usar a sintaxe de colchetes [] para acessar dados JSON.

```
const dados = {  
    "name": "Raama",  
    "age": 18.,  
    "hobby": {  
        "reading": true,  
        "gaming": false  
    },  
    "class": ["JavaScript", "HTML", "CSS"]  
  
    console.log(dados.name)  
    console.log(dados.hobby)
```

```
const dados = {  
    "name": "Raama",  
    "age": 18.,  
    "hobby": {  
        "reading": true,  
        "gaming": false  
    },  
    "class": ["JavaScript", "HTML", "CSS"]  
  
    console.log(dados["name"])  
    console.log(dados["hobby"])
```

# CONVERTENDO OBJETOS EM JSON

Mencionamos anteriormente que o JSON basicamente se parece com um objeto JavaScript dentro de uma string. Você também pode converter objetos JavaScript para o formato JSON usando a função integrada de JavaScript `JSON.stringify()`.

```
const lago = {
  agua: true,
  peixe: "lambari",
  tamanho: "2 km",
  count: 1
};
console.log(JSON.stringify(lago));
```

```
{"agua":true,"peixe":"lambari","tamanho":"2 km","count":1}
```



# CONVERTENDO JSON EM OBJETOS

Você pode converter dados JSON em um objeto JavaScript usando a função integrada `JSON.parse()`

```
const lago = {
  agua: true,
  peixe: "lambari",
  tamanho: "2 km",
  count: 1
};

console.log(JSON.parse(JSON.stringify(lago)));
```

```
▼ {agua: true, peixe: 'lambari', tamanho: '2 km', count: 1} ⓘ
  agua: true
  count: 1
  peixe: "lambari"
  tamanho: "2 km"
▶ [[Prototype]]: Object
```

# JSON

Em resumo, os objetos JSON são uma parte essencial do nosso toolkit de desenvolvimento de software. Eles nos permitem representar dados de maneira estruturada, são amplamente compatíveis e são amplamente utilizados em uma variedade de cenários de desenvolvimento. À medida que avançamos em nosso projeto, devemos ter em mente como os objetos JSON podem ser usados para facilitar a comunicação, o armazenamento e o processamento de dados.

# ATIVIDADE PRÁTICA

## **Atividade 01**

Crie um objeto JSON que represente informações sobre um livro. O objeto deve ter propriedades como título, autor, ano de publicação e uma lista de palavras-chave.

## **Atividade 02**

Dada a seguinte lista de objetos JSON que representam estudantes, adicione um novo estudante, remova o estudante com um determinado ID e atualize as notas de todos os estudantes em uma determinada disciplina.

# PROJETO

## Projeto de catálogos de livros online

Desenvolva um sistema de catálogo de livros online que permita aos usuários visualizar, adicionar, buscar e classificar livros. As informações dos livros devem ser armazenadas em um arquivo JSON. Cada livro deve conter detalhes como título, autor, gênero, ano de publicação e avaliações dos usuários.

# PROJETO

## Funcionalidades:

- Listar Livros:
  - Crie uma função para exibir todos os livros disponíveis no catálogo.
- Adicionar Livro:
  - Permita aos usuários adicionar novos livros ao catálogo. Solicite informações como título, autor, gênero, ano de publicação e permita a adição de avaliações.

# PROJETO

## Buscar Livro:

- Implemente uma funcionalidade de busca que permita aos usuários encontrar livros com base no título, autor ou gênero.

## • Classificar Livros:

- Adicione a capacidade de classificar os livros com base em diferentes critérios, como título, autor ou avaliação.

# PROJETO

Avaliar Livro:

- Permita aos usuários atribuir avaliações aos livros existentes.

Salvar e Carregar do Arquivo JSON:

- Implemente funções para salvar as alterações no arquivo JSON e carregar os dados existentes ao iniciar o programa.

# SE LIGA NO CONTEÚDO DA PRÓXIMA AULA!

AULA 13 DE JAVASCRIPT.  
TRY CATCH



INFINITY SCHOOL  
VISUAL ART CREATIVE CENTER

# TRY CATCH

try e catch são usados para lidar com exceções ou erros que podem ocorrer durante a execução do seu código. Em JavaScript, utilizamos o try catch porque eles permitem lidar com erros e exceções de maneira controlada, evitando que o código pare de funcionar abruptamente.



# TRY CATCH

Exemplos da Try Catch em JavaScript:

```
<script>
  try {
    const valor = 10 / 0; // Isso vai lançar uma exceção de 'Divisão por zero'
    console.log("Valor:", valor); // Esta linha não será executada devido à exceção
  } catch (erro) {
    console.error("Ocorreu um erro:", erro.message); // Saída: Ocorreu um erro: Divisão por zero
  } finally {
    // Saída: Este código é executado independentemente de exceções.
    console.log("Este código é executado independentemente de exceções.");
  }

</script>
```

A woman with long, dark hair and glasses is shown from the chest up, looking down at a laptop screen. The background is dark and slightly blurred.

IN

# INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 12 - PROJETO