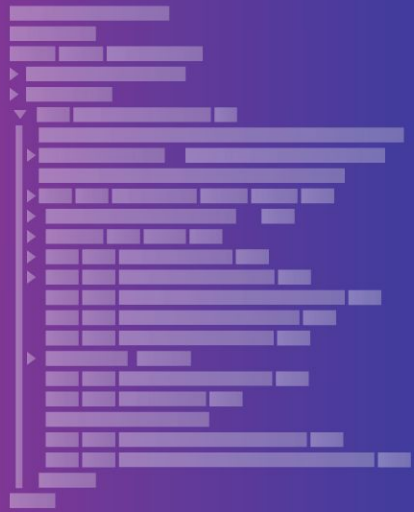
JSON
&
LOCALSTORAGE



DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

AGENDA | M1S03-A5

- **Revisão**
  - setTimeout / setInterval
  - Objetos
- **JSON**
- **Local Storage**

REVISÃO | Timeoit/Interval & Objetos

- **setTimeout / setInterval**
  - `setTimeout(funcao, 1000);`
  - `var si = setInterval(funcao, 2000);`
  - `clearInterval(si);`
- **Objetos**
  - `var meuObj = { idade: 82 };`
  - `meuObj.nome = 'Grace';`
  - `meuObj['idade']; // 82`

OBJETOS

OBJETOS

- **Objetos**: são parecidos com vetores, porém, ao invés de estarmos limitados aos índices numéricos de `0` a `N` (`0`, 1, 2 ... `N`), podemos dar nomes aos índices. Inicializamos um objeto com "`{}`" ao invés de "`[]`".
- **Chaves** (Keys): Chamamos os índices de um objeto de chaves. Quando inserimos um item em um objeto criamos um par chave-valor (key-value).
- **O acesso** se dá através de um "." (ponto) após o nome do vetor, seguido do nome da chave.
  Ex.1: `pessoa.nome;`
  Ex.2: `pessoa['nome'];`

OBJETOS | Exemplos

```
// inicia um vetor com 3 itens
var vetor = [26, 33, 42];
// acessa o segundo item do vetor (33)
vetor[1];    // 33

// inicia um objeto com 3 chaves
var objeto = { a: 26, b: 33, c: 42 };

// acessa a segunda chave do objeto
objeto.b;     // 33
objeto['b'];  // 33
```

Exemplo de vetor e objeto

OBJETOS | Exemplos

```
// inicia um vetor vazio
var vetor = [];

// insere item no vetor
vetor.push(42);

// altera 1º item no vetor
vetor[0] = 33;

// acessa o 1º item do vetor
vetor[0];  // 33
```

Exemplo de uso de vetor (array)

```
// inicia um objeto vazio
var objeto = {};

// insere/altera uma chave no objeto
objeto.num = 33;
objeto['num']  = 42;

// acessa a chave "num" do objeto
objeto.num;     // 42
objeto['num'];  // 42
```

Exemplo de uso de objeto (object)

OBJETOS | Exemplos

```
// inicia um vetor de objetos
var vetor = [
  { a: 26, b: 32, c: 42 },
  { a: 55, b: 99, c: 11 }
]

// acessa índice e chave
vetor[1].c      // 11
vetor[0]['a']   // 26
```

Exemplo de uso de vetor de objetos

```
// inicia um objeto com vetores
var objeto = {
  a: [26, 32, 42],
  b: [55, 99, 11]
}

// acessa chave e índice
objeto.a[2]     // 42
objeto['b'][0] // 55
```

Exemplo de uso de objeto com vetores em suas chaves

JSON

JSON

- **JSON** (**J**ava**S**cript **O**bject **N**otation) é a representação da estrutura de objetos JavaScript em forma de texto (string), baseada em pares de chave-valor e listas.
- Muito utilizado para enviar dados do back-end para o front-end e vice-versa, manipular estruturas de dados com texto.
- Fácil de ler e entender os dados contidos em um texto **JSON**.
- Processamento leve e de fácil interpretação.

JSON

- Embora o **JSON** seja derivado do JavaScript, ele é suportado de forma nativa ou através de bibliotecas na maior parte das principais linguagens de programação.
- O que torna esse formato ideal para troca de dados entre aplicações, mesmo que sejam escritas em diferentes linguagens.
- É o formato favorito para quase todos os serviços web disponíveis publicamente, e também é usado com freqüência para serviços web privados.

JSON | Exemplos

```
// um objeto javascript
var pessoa = {
  id: 19061209,
  nome: 'Grace Hopper',
  nascimento: '1906-12-09',
  condecoracoes: [
    'Legião do Mérito',
    'M. Vitória da 2ª G.Mundial',
    'M. Presid. da Liberdade'
  ]
};
```

Exemplo de um objeto JavaScript com array dentro

```
// uma string JSOM
var pessoaJSON = `{
  "id": 19061209,
  "nome": "Grace Hopper",
  "nascimento": "1906-12-09",
  "condecoracoes": [
    "Legião do Mérito",
    "M. Vitória da 2ª G.Mundial",
    "M. Presid. da Liberdade"
  ]
}`;
```

Representação da estrutura à esquerda em JSON

JSON | Exemplos

```
// um array de objetos javascript
var carros = [
  { ano: 1995, modelo: 'Escort' },
  { ano: 1999, modelo: 'Gol' },
  { ano: 1997, modelo: 'Uno' },
  { ano: 2005, modelo: 'Corsa' },
];
```

Exemplo de um array JavaScript com objetos dentro

```
// uma string JSOM
var carrosJSON = `[
  { "ano": 1995, "modelo": "Escort" },
  { "ano": 1999, "modelo": "Gol" },
  { "ano": 1997, "modelo": "Uno" },
  { "ano": 2005, "modelo": "Corsa" },
]`;
```

Representação da estrutura à esquerda em JSON

JSON | Stringify & Parse

- Para transformarmos objetos JavaScript em texto JSON e texto JSON em objetos JavaScript usaremos as funcionalidades encontrada na entidade **JSON** disponível no ambiente de execução.
- **`JSON.parse(texto)`** é utilizado para converter um texto JSON (string) em objeto. A chamada deste método nos retornará um objeto.
- **`JSON.stringify(objeto)`** é utilizado para converter objetos em texto JSON. A chamada deste método nos retornará um texto JSON.

JSON | Stringify & Parse

```
// transformando um objeto em JSON
var carrosJSON = JSON.stringify(carros);

// carrosJSON terá o seguinte valor
`[
  { "ano": 1995, "modelo": "Escort" },
  { "ano": 1999, "modelo": "Gol" },
  { "ano": 1997, "modelo": "Uno" },
  { "ano": 2005, "modelo": "Corsa" },
]`
```

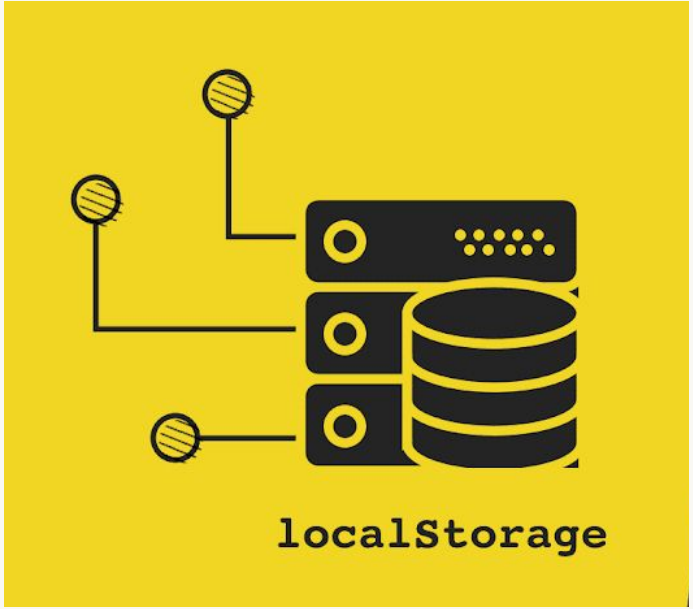
Convertendo um objeto em JSON string

```
// transformando um JSON em objeto
var carros = JSON.parse(carrosJSON);

// carros terá o seguinte valor
[
  { ano: 1995, modelo: 'Escort' },
  { ano: 1999, modelo: 'Gol' },
  { ano: 1997, modelo: 'Uno' },
  { ano: 2005, modelo: 'Corsa' },
]
```

Convertendo uma JSON string em objeto

LOCALSTORAGE



LOCALSTORAGE

- Recurso de **armazenamento de dados** em formato de texto (strings) presente no navegador e acessível via JavaScript.
- Pode armazenar **até 5MB** por site, sem data de expiração.
- Acessaremos este recurso através do objeto **`localStorage`** ou **`window.localStorage`**.
- Como permite apenas armazenamento de texto, podemos fazer bom uso das funcionalidades do recurso **JSON**.
- Caso execute **`getItem`** que não exista, receberemos o valor **`null`**.
- Cada site (domínio) tem seu próprio escopo localStorage.

LOCALSTORAGE

- `localStorage.setItem('chave', 'valor');`
  Armazena a string 'valor' em 'chave'
- `var dado = localStorage.getItem('chave');`
  Recupera o valor armazenado em 'chave'
- `localStorage.removeItem('chave');`
  Apaga o valor armazenado em 'chave'
- `localStorage.clear();`
  Apaga todos valores armazenados no localStorage

LOCALSTORAGE

```
// armazena item em localStorage
localStorage.setItem('foo', '13');

// recupera item de localStorage
var bar = localStorage.getItem('foo');

console.log(bar);  // '13'

// remove item de localStorage
localStorage.removeItem('foo');

// remove todos itens de localStorage
localStorage.clear();
```

Exemplo de uso de vetor do localStorage

LOCALSTORAGE | JSON

```
// inicia objeto
var obj = { a: 'bah', b: 32 };

// converte para string JSON
var objJSON = JSON.stringify(obj);
// '{"a":"bah","b":32}'

// armazena item em localStorage
localStorage.setItem('foo', objJSON);
```

Exemplo de armazenamento de objeto com JSON

```
// recupera item de localStorage
var bar = localStorage.getItem('foo');

// converte string JSON para objeto
var barObj = JSON.parse(bar);
// { a: 'bah', b: 32 }

barObj.a;
// 'bah'
```

Exemplo de recuperação de JSON e conversão para objeto

LOCALSTORAGE | JSON

```
// inicia objeto
var obj = { a: 'bah', b: 32 };

// converte para string JSON
// armazena item em localStorage
localStorage.setItem(
  'foo',
  JSON.stringify(obj)
);
```

Exemplo de armazenamento de objeto com JSON

```
// recupera item de localStorage
// converte string JSON para objeto
var barObj = JSON.parse(
  localStorage.getItem('foo')
);

barObj.a;  // 'bah'
```

Exemplo de recuperação de JSON e conversão para objeto

MATERIAL COMPLEMENTAR

JSON em Javascript | https://youtu.be/ziOjvR56qOw

O que é JSON | https://youtu.be/K1f7G0JMkLU

localStorage para salvar preferências com JavaScript | https://youtu.be/6deCUoDuMQc

MATERIAL COMPLEMENTAR

Objeto - JavaScript | developer.mozilla.org/docs/Web/JavaScript/Reference/Global_Objects/Object

JavaScript Objects | https://developer.mozilla.org/en-US/docs/Web/API/setInterval

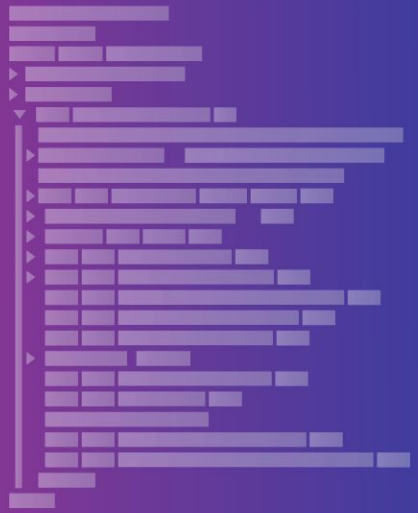
Trabalhando com JSON | https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Objects/JSON

Window.localStorage API | https://developer.mozilla.org/pt-BR/docs/Web/API/Window/localStorage

Window localStorage | https://www.w3schools.com/jsref/prop_win_localstorage.asp

DEVinHouse

Parcerias para desenvolver a sua carreira



OBRIGADO!

SENAI

<LAB365>