# Tema 06 –Objetos e Array de Objetos

## Introdução

Olá, futuras mentes desenvolvedoras, sejam bem-vindos do curso de Frontend do Instituto da Oportunidade Social. Nessa aula, você aprenderá sobre objetos e array de objetos.

## Objetos

Vimos que arrays são um conjunto de valores que não precisam ser do mesmo tipo e que objetos são uma coleção de dados e/ou funcionalidades relacionadas. Objetos, geralmente, são construídos com pares **chave: valor**, onde a chave é um rótulo escolhido pelo programador e o valor é o conteúdo armazenado nesse rótulo. Por exemplo:

|  |  |  |
| --- | --- | --- |
| **Sintaxe geral** | **Exemplo 01:** | **Exemplo 02:** |
| const ou let nome\_do\_objeto = {  chave01: valor01,  chave02: valor02,  chave03: valor03,  ...  chaveXX: valorXX,  } | const carro = {  tipo:"Fiat",  modelo:"500",  cor:"branco"  }; | let pessoa = {  firstName: "Irmão do",  lastName: "Jorel",  idade: 10,  corDosOlhos: "preto"  }; |

Você deve observar que o nome da chave pode ser qualquer um desde que não contenha espaços ou caracteres especiais. Pode ser em qualquer língua: português, inglês, alemão, sueco, etc. Mas, lembre-se que, ser usar o nome em português não colocar acentuação gráfica. Isso é uma regra obrigatória.

Outra observação é a convenção de camelCase, que comumente usada por programadores para o nome da chave com duas palavras ou mais palavras, exemplo: firstName, corDosOlhos, nomePesssoa, etc. Isso é uma convenção não uma regra obrigatória.

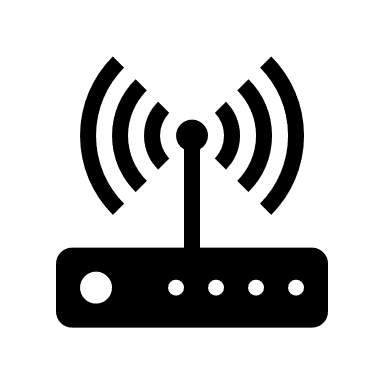
Vamos para de papo e começar a praticar.

### Vamos praticar

Siga os passos para criar o projeto:

1. Abra o VS Code e escolha um diretório de trabalho para o seu projeto.
2. Crie um diretório para seu projeto com o nome representativo, por exemplo, 06\_JS\_Array\_Objetos.
3. Crie um arquivo dentro do diretório do projeto com o nome index.html.
4. Insira o seguinte código no seu arquivo index.html.

|  |
| --- |
| <!DOCTYPE html>  <html lang="pt-br">      <head>          <meta charset="UTF-8" />  <link rel="shortcut icon" href="#" />          <title>Objetos e Array de Objetos</title>      </head>      <body>          <script src="./js/main.js"></script>      </body>  </html> |

1. Esse código mostra a marcação <script> sem nenhum código JavaScript entre a abertura e o fechamento da tag, apenas o atributo **src** com o valor main.js. Isso significa que o código JavaScript está em um arquivo externo. Portanto, temos que criar esse novo arquivo **main.js** dentro do diretório do projeto.
2. Vamos deixar o arquivo main.js vazio e à medida que vamos aprendendo coisas novas iremos completar a implementação do código JavaScript.
3. Abra o arquivo index.html, clique no botão **Go Live** da extensão **Live Server** e abra as Ferramentas de desenvolvimento do navegador web (Atalho **F12**).

Agora, vamos implementar todo o nosso código desse tema no arquivo main.js e ver as mensagens impressas no console.

1. Vamos como utilizar objetos em JS. No arquivo, **main.js** digite o seguinte código.

|  |  |
| --- | --- |
| 1 | // Objetos  let pessoa = {      firstName: 'Irmão do',      lastName: 'Jorel',      idade: 10,      corDosOlhos: 'preto',      hobbies: ['música', 'filmes', 'esportes'],      endereco: {          rua: 'Rua do bobos',          numero: 0,          cidade: 'São Paulo',          estado: 'SP',      },  };  console.log(pessoa);  console.log(pessoa.firstName);  console.log(pessoa.lastName, pessoa.idade);  console.log(pessoa.hobbies[1]);  console.log(pessoa.endereco.cidade); |
| 2 |
| 3 |
| 4 |
| 5 |
| 6 |
| 7 |
| 8 |
| 9 |
| 10 |
| 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |
| 19 |
| 20 |

1. Antes de explicar o que foi programado vamos ver o resultado mostrado no console.

Graphical user interface, text, application

Description automatically generated

O comando na linha 16 imprime todo o objeto, podemos expandir e todos os pares chave: valor do objeto.

Text

Description automatically generated

Observe que a linha mostrada em main.js 16:9, o 16 é a linha da instrução que executou e resultou na impressão do log console.

Na linha 17, acessamos apenas o campo/chave firstName do objeto e o resultado é:



Observe que a linha mostrada em main.js 17:9, o 17 é a linha da instrução que executou e resultou na impressão do log console.

Na linha 18, acessamos apenas o campo/chave lastName do objeto e a idade do objeto e o resultado é:



Observe que a linha mostrada em main.js 18:9, o 18 é a linha da instrução que executou e resultou na impressão do log console.

Na linha 19, acessamos apenas o segundo elemento do campo/chave hobbies do objeto e o resultado é:



Observe que a linha mostrada em main.js 18:9, o 18 é a linha da instrução que executou e resultou na impressão do log console.

Na linha 20, acessamos apenas o campo/chave cidade do objeto **endereco** que está dentro do nosso objeto pessoa e o resultado é:



Observe que a linha mostrada em main.js 20:9, o 20 é a linha da instrução que executou e resultou na impressão do log console.

1. Podemos abordar, agora, um assunto bem legal, que é **atribuição via desestruturação (destructuring assignment)**. O JavaScript permite extrair os dados de um array ou um objeto e armazenar em variáveis simples através da **atribuição via desestruturação (destructuring assignment)**. Mas como isso funciona? Vamos ver na prática.
2. Continuando a implementação do projeto, insira o seguinte código no arquivo main.js:

|  |  |
| --- | --- |
| 22 | // Atribuição via desestruturação  console.clear();  const {      firstName,      lastName,      endereco: { cidade },  } = pessoa;  console.log(`${firstName} ${lastName} é de ${cidade}`); |
| 23 |
| 24 |
| 25 |
| 26 |
| 27 |
| 28 |
| 29 |
| 30 |

1. O resultado é mostrado no console do navegador:

Graphical user interface, text, application, email

Description automatically generated

Observe que foi possível usar apenas o nome da variável, que é mais curto, no comando da linha 30 para acessar o conteúdo para ser impresso. Isso só foi possível por causa da **atribuição via desestruturação, que foi feita nas linhas 25 até 29.**

**Importante: Para a atribuição via desestruturação ocorrer corretamente em objetos, você deve sempre usar os nomes das variáveis iguais as chaves que você deseja extrair. Caso contrário o conteúdo será undefined. Sabia bem esse assunto, isso é muito utilizado em JS.**

1. Já em arrays a **atribuição via desestruturação (destructuring assignment) funciona um pouco diferente**. Basicamente, é só você igualar um vetor a outro.
2. Continuando a implementação do projeto, insira o seguinte código no arquivo main.js:

|  |  |
| --- | --- |
| 22 | // Atribuição via desestruturação com arrays  console.clear();  const array = [1, 2, 3, 4, 5];  let [valor01, valor02, ...resto] = array;  console.log(valor01);  console.log(valor02);  console.log(resto); |
| 23 |
| 24 |
| 25 |
| 26 |
| 27 |
| 28 |

1. O resultado é mostrado no console do navegador:

Graphical user interface, text, application, email

Description automatically generated

Observe que com arrays só foi necessário fazer uma igualdade entre eles na linha 25:

|  |  |  |
| --- | --- | --- |
| **Esse cara é um array, observe os colchetes** | **Igualdade** | **Esse também é um array** |
| [valor01, valor02, ...resto] | = | array |

Você deve observar também o operador **rest** ( ... ), que permite trabalhar com múltiplos parâmetros de arrays e funções. O **rest** é utilizado para extrair múltiplos valores do array ou para a função receber uma quantidade indefinida de valores.

No exemplo, o primeiro elemento do **array** é armazenado na variável **valor01**, o segundo elemento do **array** é armazenado na variável **valor02** e os demais (a partir do terceiro elemento) foram armazenados na variável resto, que é um array também como mostra no console:



Você pode testar os seguintes comandos:

|  |
| --- |
| let [valorA, valorB, valorC ...restante] = array;  console.log(valorA);  console.log(valorB);  console.log(valorC);  console.log(restante); |

E verá que a variável **restante** irá armazenar o conteúdo a partir do quarto elemento até do final do array.

## Array de objetos

Do mesmo jeito temos arrays com valores diversos, podemos ter arrays de objetos. Vamos ver isso na prática.

### Vamos praticar

Siga os passos para continuar a implementação no arquivo **main.js**:

1. Continuando a implementação do projeto, insira o seguinte código no arquivo main.js:

|  |  |
| --- | --- |
| 40 | // Array de objetos  console.clear();  const tarefas = [      {          id: 1,          texto: 'Levar o lixo para fora',          isCompleted: true,      },      {          id: 2,          texto: 'Encontrar com o chefe',          isCompleted: true,      },      {          id: 3,          texto: 'Consulta no dentista',          isCompleted: false,      },  ];  console.log(tarefas);  console.log(tarefas[1].texto);  console.log(tarefas[2].id, tarefas[2].texto, tarefas[2].isCompleted); |
| 41 |
| 42 |
| 43 |
| 44 |
| 45 |
| 46 |
| 47 |
| 48 |
| 49 |
| 50 |
| 51 |
| 52 |
| 53 |
| 54 |
| 55 |
| 56 |
| 57 |
| 58 |
| 59 |
| 60 |
| 61 |
| 62 |

1. O resultado é mostrado no console do navegador:

Graphical user interface, text, application, email

Description automatically generated

Você pode ver que a array tarefas possui diversos objetos dentro dele (linhas 42 até 58). E para acessar a cada objeto você deve referenciar o índice do mesmo. Por exemplo, na linha 60, acessamos todo o array e o resultado é:

Graphical user interface, text, application

Description automatically generated

Na linha 61, acessamos o campo/chave texto do segundo objeto do array e o resultado é:



Na linha 62, acessamos os campos/chaves do terceiro objeto do array e o resultado é:



## JSON

JSON (JavaScript Object Notation) é um formato compacto, de padrão aberto e independente usado para a troca de dados entre sistemas de forma simples e rápida. Ele utiliza texto legível a humanos, no formato atributo-valor (natureza auto-descritiva). Isto é, um modelo de transmissão de informações no formato texto, muito usado em serviços na internet.

### Vamos praticar

Siga os passos para continuar a implementação no arquivo **main.js**:

1. Continuando a implementação do projeto, insira o seguinte código no arquivo main.js:

|  |  |
| --- | --- |
| 64 | // Transformar um array em JSON  console.log(JSON.stringify(tarefas)); |
| 65 |

1. O resultado é mostrado no console do navegador:

Graphical user interface, text, application, email

Description automatically generated

Observe que resultado do comando na linha 65 foi a impressão de uma string na notação JSON, utilizando o método **stringify()**.

|  |
| --- |
| [{"id":1,"texto":"Levar o lixo para fora","isCompleted":true},{"id":2,"texto":"Encontrar com o chefe","isCompleted":true},{"id":3,"texto":"Consulta no dentista","isCompleted":false}] |

Da mesma forma que você consegue transformar um objeto em JSON, é possível transformar um JSON em um objeto.

1. Continuando a implementação do projeto, insira o seguinte código no arquivo main.js:

|  |  |
| --- | --- |
| 67 | // JSON  const tarefa\_JSON =      '[{"id":1,"texto":"Levar o lixo para fora","isCompleted":true},{"id":2,"texto":"Encontrar com o chefe","isCompleted":true},{"id":3,"texto":"Consulta no dentista","isCompleted":false}]';  const objeto = JSON.parse(tarefa\_JSON);  console.clear();  console.log(objeto); |
| 68 |
| 69 |
| 70 |
| 71 |
| 72 |
| 73 |

1. O resultado é mostrado no console do navegador:

Graphical user interface, text, application

Description automatically generated

Observe na linha 70 como retornamos o mesmo array de objetos original, convertendo o JSON em objeto pelo método **parse()**.

## Conclusão

Existem muitas utilidades sobre arrays de objeto e o operador rest. Confira esses links e teste os códigos mostrados neles:

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Operators/Destructuring_assignment>

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Operators/Spread_syntax>

<https://www.devmedia.com.br/javascript-operadores-rest-e-spread/41200>

Para saber mais sobre JSON, vale a pena dar uma conferida nos links:

<https://pt.wikipedia.org/wiki/JSON>

<https://www.freecodecamp.org/news/javascript-array-of-objects-tutorial-how-to-create-update-and-loop-through-objects-using-js-array-methods/>

Esse segundo link está em inglês, mas você pode usar a extensão do Google Translate no Chrome para traduzir e ler o artigo.