Objetivo:

I. JSON – JavaScript Object Notation;

II. Estruturação e desestruturação.

Observação: Para fazer os exercícios e exemplos recomenda-se o uso do VS Code ou da interface de programação online https://replit.com/.

I. JSON – JavaScript Object Notation

JSON (em português, Notação de Objeto JavaScript) é uma estrutura utilizada para armazenar dados de forma estruturada.

A estrutura do objeto é formada por um par de chaves. Como exemplo, a variável carro possui como conteúdo um objeto:

```
carro = {}
```

Dentro do objeto JavaScript os conteúdos são colocados como propriedades. A propriedade é formada por um nome seguido por dois pontos e valor da propriedade. Como exemplo, o objeto colocado na variável pessoa possui as propriedades nome, idade, endereco e contas.

O valor da propriedade nome está à direita dos dois pontos, ou seja, o valor da propriedade nome é Ana.

As propriedades obrigatoriamente são separadas por vírgulas.

```
const pessoa = {
  nome: "Ana",
  idade: 22,
  endereco: {
    logradouro: "R. Afonso Pena",
    numero: 123,
    bairro: "Vila Jardim",
    cidade: "SJC",
    uf: "SP"
  },
  contas: [84,43,105]
}
```

O nome da propriedade é chamado de chave, desta forma, tem-se que a propriedade tem a notação `key:value`.

Para acessar o conteúdo da propriedade nome precisamos colocar o nome da variável que contém o objeto, ou seja, `pessoa.nome`. O ponto (.) significa que o termo à direita está dentro do objeto à esquerda, ou seja, `nome` está dentro de `pessoa`.

```
//acessa a propriedade nome do objeto pessoa
```

```
console.log("Nome", pessoa.nome)
//a propriedade endereco possui um objeto como valor
//acessa a propriedade logradouro do objeto que está na propriedade endereco
console.log("Logradouro", pessoa.endereco.logradouro)
```

O conteúdo de uma propriedade poder ser de qualquer tipo de dado:

- O conteúdo da propriedade `nome` é do tipo string;
- O conteúdo da propriedade `idade` é do tipo int;
- O conteúdo da propriedade `endereco` é outro objeto JavaScript;
- O conteúdo da propriedade `contas` é um array.

Para acessar o conteúdo de uma propriedade aninhada dentro de outro objeto, assim como a propriedade `logradouro`, temos de colocar o nome da variável seguida pelo nome da propriedade que a contém (`endereco`):

```
pessoa.endereco.logradouro
```

Para acessar um elemento de um array de uma propriedade precisamos colocar o nome da variável seguida pelo nome da propriedade e o índice de posição no array. O código a seguir retorna a 2ª posição do array, isto é, retorna o valor 43:

```
pessoa.contas[1]
```

II. Estruturação e desestruturação

A desestruturação (destructuring) é uma característica do JS que permite extrair valores de objetos ou arrays e atribuí-los a variáveis de forma mais concisa.

No exemplo a seguir a atribuição por desestruturação (destructuring assignment) é usada para copiar os valores das propriedades `bairro` e `cidade` do objeto JSON para as variáveis `bairro` e `cidade`. A desestruturação ocorre entre chaves **{ }** no lado esquerdo da atribuição.

```
const endereco = {
    logradouro: 'Rua um',
    nro: 123,
    bairro: 'Vila Jardim',
    cep: 12345678,
    cidade: 'Jacareí',
    uf: 'SP'
};

// atribuição por desestruturação: as variáveis recebem os valores das propriedades
const {bairro,cidade} = endereco;
console.log(bairro); // o resultado é Vila Jardim
console.log(cidade); // o resultado é Jacareí
```

Observação: a atribuição por desestruturação só funciona se as variáveis tiverem exatamente os nomes das propriedades do objeto JSON.

De modo oposto, existe a estruturação de objetos JSON. Na estruturação, variáveis são usadas para compor objetos JSON, os nomes das variáveis serão as propriedades do objeto e os valores das variáveis serão os valores das propriedades. No exemplo a seguir, as variáveis nome, idade e peso serão as propriedades do objeto JSON. A estruturação ocorre entre chaves `{}` no lado direito da atribuição.

```
const nome = "Ana";
const idade = 22;
const peso = 61.5;
// na estruturação as variáveis são colocadas nas chaves do lado direito da atribuição
const pessoa = {nome,idade,peso};
console.log(pessoa); // o resultado é { nome: 'Ana', idade: 22, peso: 61.5 }
```

A desestruturação de arrays é feita usando colchetes `[]` no lado esquerdo da atribuição. A desestruturação permite extrair elementos de um array e atribuí-los às variáveis individuais. No exemplo a seguir os 3 primeiros elementos do array serão copiados, respectivamente, para as variáveis `nome`, `carro` e `fruta`. Observe que o 4º elemento do array não foi copiado.

```
const textos = ["Ana","Uno","Laranja","Couve"];
// desestruturação para atribuir os elementos do array às variáveis
const [nome, carro, fruta] = textos;
console.log(nome); // o resultado é Ana
console.log(carro); // o resultado é Uno
console.log(fruta); // o resultado é Laranja
```

A estruturação de arrays é usada para criar arrays combinando elementos de outras variáveis. Ela é realizada usando colchetes `[]` no lado direito da atribuição. No exemplo a seguir os valores das variáveis foram usados para compor os valores do array a ser criado:

```
const base = 2;
const altura = 3;
const profundidade = 4;
// estruturação para criar um array combinando os valores das variáveis base, altura e profundidade
const medidas = [base,altura,profundidade];
console.log(medidas); // o resultado é [ 2, 3, 4 ]
```

A desestruturação de um objeto JSON pode ser feita na atribuição do parâmetro de uma função. No exemplo a seguir, a desestruturação ocorre colocando as chaves `{}` na atribuição do parâmetro da função exibir:

```
// retira do objeto JSON apenas a propriedade carro
function exibir({carro}){
    console.log(carro);
}

const obj = {
```

```
    pessoa: {
        nome: "Ana",
        idade: 22
    },
    carro: {
        marca: "Fiat",
        modelo: "Uno"
    }
};

// chama a função passado o JSON que está na variável obj
exibir(obj);
```

Podemos desestruturar um objeto JSON aninhado. No exemplo a seguir as chaves verdes desestrutura o objeto para obter a propriedade `veiculo` e as chaves amarelas desestrutura o objeto `veiculo` para obter a propriedade tipo:

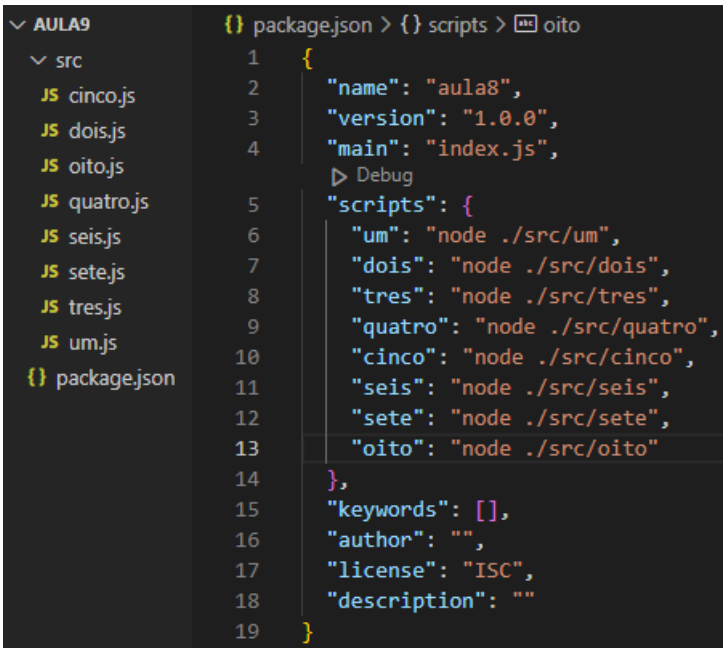
```
const pessoa = {
    nome: "Ana",
    veiculo: {
        tipo: {
            marca: "GM",
            modelo: "Corsa"
        },
        ano: 2010
    }
};

const {veiculo:{tipo}} = pessoa;
console.log(tipo);
```

Exercícios

Veja o vídeo se tiver dúvidas nos exercícios: https://youtu.be/V_6Bi9hIGuI

Para fazer os exercícios crie um projeto de nome aula9 no VS Code assim como é mostrado a seguir. Cada programa deverá estar num arquivo separado da pasta src. Crie uma propriedade para cada exercício na propriedade scripts do arquivo package.json. Para rodar o arquivo use: npm run propriedade, onde propriedade será um, dois, ... e oito.



Exercício 1: Complete o código para imprimir no console os valores das propriedades inicio e fim do objeto que está na variável contrato.

```
const contrato = {
    inicio: "01/02/1995",
    fim: "30/07/1997",
    tipo: "locação",
    valor: "123.45"
};
```

Exemplo de saída:

```
Início: 01/02/1995
Fim: 30/07/1997
```

Exercício 2: Complete o código para imprimir no console as notas p1, p2 e projeto do objeto que está na variável disciplina.

```
const disciplina = {
    nome: "Algoritmos",
    carga: 80,
    pesos: {
        p1: 0.25,
        p2: 0.35,
        projeto: 0.4
    },
    notas: {
        p1: 8.2,
        p2: 7.5,
        projeto: 9
    }
};
```

Exemplo de saída:

```
P1: 8.2
P2: 7.5
Projeto: 9
```

Exercício 3: Alterar o programa do Exercício 2 para imprimir no console a nota final. A nota final é calculada:

pesoP1 * notaP1 + pesoP2 * notaP2 + pesoProjeto * notaProjeto.

Exemplo de saída:

```
Nota final: 8.275
```

Exercício 4: Refazer o Exercício 3 usando o objeto que está na variável disciplina a seguir. Observe que as notas e pesos estão num array.

Exemplo de saída:

```
Nota final: 8.275
```

```
const disciplina = {
    nome: "Algoritmos",
    carga: 80,
    pesos: [0.25,0.35,0.4],
    notas: [8.2,7.5,9]
};
```

Exercício 5: Refazer o Exercício 3 usando o objeto que está na variável disciplina a seguir.

Exemplo de saída:

```
Nota final: 8.275
```

```
const disciplina = {
    nome: "Algoritmos",
    carga: 80,
    notas: [
        {
            peso: 0.25,
            nota: 8.2
        },
        {
            peso: 0.35,
            nota: 7.5
        },
        {
            peso: 0.4,
            nota: 9
        }
    ]
};
```

Exercício 6: Desestruturar as propriedades `inicio` e `fim` do objeto que está na variável `contrato` e imprimir no console.

Exemplo de saída:

```
Início: 01/02/1995
Fim: 30/07/1997
```

```
const contrato = {
    inicio: "01/02/1995",
    fim: "30/07/1997",
    tipo: "locação",
    valor: "123.45"
};
```

Exercício 7: Desestruturar as propriedades `p1` e `p2` do objeto que está na propriedade `pesos` da variável `disciplina` e imprimir no console.

```
const disciplina = {
    nome: "Algoritmos",
    carga: 80,
    pesos: {
        p1: 0.25,
        p2: 0.35,
        projeto: 0.4
    },
    notas: {
        p1: 8.2,
        p2: 7.5,
        projeto: 9
    }
};
```

Exemplo de saída:

```
P1: 0.25
P2: 0.35
```

Exercício 8: Desestruturar para as variáveis `p1` e `p2` os dois primeiros elementos do array que está na propriedade `pesos` do objeto que está na `disciplina` e imprimir no console.

```
const disciplina = {
    nome: "Algoritmos",
    carga: 80,
    pesos: [0.25,0.35,0.4],
    notas: [8.2,7.5,9]
};
```

Exemplo de saída:

```
Peso P1: 0.25
Peso P2: 0.35
```