Declaração de Variáveis

- Var: Uma variável declarada com a palavra-chave **var** na linguagem javascript, permite que você declare e/ou atribua a ***mesma variável*** quantas vezes quiser.
- Let: Não permite ser declarada novamente no mesmo escopo de código.
- Const: A **declaração const** cria uma variável cujo o valor é fixo, ou seja, uma constante somente leitura. Isso não significa que o valor é imutável, apenas que a variável constante não pode ser alterada ou retribuída.

Definição de Escopo - Delimitação, espaço sem restrição, até onde minha variável é visível, acessível.

JS - Objetos

Na vida real, um carro é um objeto.
Um carro tem **propriedades** como peso e cor, e **métodos** como ligar e desligar:

Objeto	Propriedades	Métodos
	car.name = Fiat car.weight = 850 kg car.model = 500 car.color = branco	car.start () car.drive () car.brake () car.stop ()

JS - Objetos

Objetos são variáveis também. Mas os objetos podem conter muitos valores.

Este código atribui **muitos valores** (Fiat, 500, branco) para uma **variável** carro

```
var car = {type:"Fiat", model:500, color:"white"};
```

Os métodos são **ações** que podem ser executadas em objetos.

Métodos são armazenados em propriedades como **definições de** função.

JS - Objetos

A sintaxe para acessar a propriedade de um objeto é:

```
objectName.property          // person.age
```

ou

```
objectName["property"]       // person["age"]
```

ou

```
objectName[expression]       // x = "age"; person[x]
```

JS - Objetos

JavaScript para ... em loop

O JavaScript para ... na declaração percorre as propriedades de um objeto.

Sintaxe

```
for (variable in object) {
    code to be executed
}
```

JS - Objetos

Adicionando Novas propriedades

Você pode adicionar novas propriedades a um objeto existente, basta dar-lhe um valor.

Suponha que o objeto pessoa já existe - você pode, então, dar-lhe novas propriedades:

Exemplo

```
person.nationality = "English";
```

JS - Objetos

Apagar Propriedades

A **exclusão** de palavras-chave elimina uma propriedade de um objeto:

Exemplo

```
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
delete person.age;   // or delete person["age"];
```

JS - Objetos

Acessando Métodos de objeto

Você cria um método de objeto com a seguinte sintaxe:

```
methodName : function() { code lines }
```

Você acessa um método de objeto com a seguinte sintaxe:

```
objectName.methodName()
```

Exemplo 3 – Objetos

```
<!DOCTYPE html>
<html>
    <head>
        <title>Uma pagina Web</title>
        <meta name="keywords" content="HTML, CSS, JS, Exemplo">
        <meta name="description" content="Uma página Web">
        <meta name="author" content="Prof Natacsha">
    </head>
    <body>
        <p id="demo"></p>

        <script>
        function person(firstName,lastName,age,eyeColor) {
            this.firstName = firstName;
            this.lastName = lastName;
            this.age = age;
            this.eyeColor = eyeColor;
            this.changeName = function (name) {
                this.lastName = name;
                console.log(myMother.lastName);
                document.write (myMother.lastName);
                window.alert (myMother.lastName);
            }
        }
        var myMother = new person("Sally","Rally",48,"green");
        myMother.changeName("Doe");
        document.getElementById("demo").innerHTML =
        "My mother's last name is " + myMother.lastName;

        </script>
    </body>
</html>
```