

VARIÁVEIS EM JAVASCRIPT

Variáveis em Programação

A Computação tem, basicamente, duas funções: **fazer cálculos** e **armazenar informações**.

A palavra computador vem de computar, que é contar, e é basicamente isso que um computador faz: cálculos.

Seja em uma pesquisa científica, uma troca de mensagens no Whatsapp ou a exibição de um site da internet, os computadores precisam fazer uma série de **operações binárias** (bits) em seu hardware, para resultar em pixels na sua tela, onde você vê o resultado acontecer.

Porém, só fazer essas operações binárias não serve de muita coisa, se os resultados desses cálculos não forem armazenados.

Imagine que a cada vez que você desliga seu computador ou celular, todas suas informações sejam apagadas: textos, vídeos, fotos, arquivos etc. Isso é inconcebível, computador é usado também para armazenar informações.

O ato de armazenar dados, de uma maneira temporária (até desligar o computador ou fechar um programa) ou permanente (gravar dados em um HD ou servidor), é tão essencial e importante que isso é feito de maneira bem frequente em programação, através de **variáveis**.

Através das variáveis, vamos armazenar quaisquer tipos de informações, para serem acessadas, modificadas, alteradas ou mesmo excluídas futuramente, tudo decidido pelo programador.

Variáveis em JavaScript

Certamente você já se deparou com um jogo feito em JavaScript, e talvez nem tenha percebido.

Você joga uma partida, e marca 100 pontos.

Depois, joga novamente e marca 200, então esse passa a ser seu novo recorde.

Mas como o jogo sabe que 200 é seu novo recorde? Para saber isso, ele teria que saber que sua pontuação anterior era menor que 200.

Ou seja, o número “100” foi armazenado pelo JavaScript em algum lugar, então quando você jogou de novo, sua nova pontuação foi comparada com essa anterior, para saber que seu record foi quebrado.

Você também já deve ter entrado em algum site que pedia seu nome, você digitava e depois ele exibía uma mensagem “Olá, Fulano”.

Ou seja, o Javascript guardou seu nome em algum lugar, para exibir no futuro.

Uma variável foi usada para armazenar o texto “Fulano”, que você escreveu. E depois, o JavaScript foi nessa variável, pegou as informações que tinha nela e exibiu para você o “Olá, Fulano”.

Em alguns sistemas bancários, é permitido fazer 3 saques por dia.

Então, de alguma maneira, o sistema tem que saber quantos saques foram feitos.

Inicialmente, ele armazena o número 0 em uma variável.

Quando você fizer um saque, ele atualiza aquela variável pra armazenar o número 1.

Efetuando outro saque, o sistema muda a variável para o número 2.

Mais uma vez, a variável é incrementada e armazena o número 3 quando você fizer mais um saque naquele dia.

Porém, sempre que você vai fazer um saque, o sistema checka essa variável. Se tentar fazer mais um, o sistema vai ver que número está armazenado na variável, e se for 3, ele não libera mais você para sacar novamente.

Enfim, poderíamos ficar muito mais tempo falando das inúmeras utilidades das variáveis.

Como declarar uma variável em JavaScript

Declarar uma variável é reservar memória para armazenar uma informação.

Essa informação vai ficar associada a essa variável, e para podermos fazer isso temos que usar um comando especial, que diga ao JavaScript “Ei, essa é uma variável! Guarde ela!”.

Esse comando é dado por uma palavra chamada **var**. Assim, para declararmos uma variável, fazemos: **var nome ;**

Ou seja, primeiro escrevemos a palavra “var” e depois escolhemos um nome pra nossa variável e depois usamos o ponto e vírgula.

Por exemplo, se você quiser criar uma variável para armazenar a idade do leitor de seu site, faça:

var idade;

Para armazenar o dia, de uma data:

- **var dia;**
- O mês: **var mes;**
- O ano: **var ano;**

Quando você faz isso, você está reservando um bloco de memória para cada uma dessas informações. Cada bloco de memória tem um endereço específico (tipo **0x2112H**, algo que seu computador entende).

Então, ao invés de usar esse endereço pouco amigável, digite **dia** e pronto, vai estar acessando esse endereço de memória.

Como declarar mais de uma variável em uma linha

Você também pode definir essas três variáveis numa mesma linha, basta separar cada nome por uma vírgula:

var dia, mes, ano;

Ao declarar variáveis dessa maneira, apenas estamos **reservando espaços** em memória, é como se disséssemos “Ei, JavaScript, reserva espaço para essas variáveis, depois vamos usar elas!”

Como escolher o nome de uma variável em JS

No início da jornada em programação o estudante fica tentado a usar a mesma estratégia que usa na Matemática, isto é, batizar as variáveis de “a”, “b”, “x” ou “y”, pois é bem mais simples e fácil.

Mas a medida que for programando, seus scripts passarão a ter dezenas, centenas ou mesmo milhares de linhas de código, e certamente você não vai mais se lembrar do nome de todas as variáveis, caso tenha escolhido essas letras simples como nome.

Quer armazenar a idade de alguém? Coloque a variável com nome “**idade**”. Quer armazenar uma mensagem de erro? Use algo como “**msg_error**” como nome.

Vai armazenar o valor dos juros produzidos por uma aplicação financeira? Use “**juros**”, pois assim você vai facilmente se lembrar, e seu código vai ficar mais legível e fácil de ser entendido por outros desenvolvedores web.

Regras para declarar variáveis em JavaScript

É você quem escolhe os nomes de suas variáveis. Porém procure escolher sempre **nomes que façam sentido**, que tenham a ver com a informação que vai ser armazenada na variável.

Isso não quer dizer que possamos escolher todo e qualquer nome, existem algumas regras.

O nome de suas variáveis devem **começar** com uma **letra**, com o sinal de underline _ ou com o cifrão \$.

Ou seja, **não começar com número**, nem **hífen** ou **algum outro caractere**.

Embora não possa iniciar o nome de caractere com número, **você pode usar números no nome de sua variável**:

var IP1, IP_2, \$IP3 ;

Devemos lembrar que as **variáveis**, em JavaScript, são **case sensitive**, ou seja, letra minúscula difere de letra maiúscula.

Assim, a variável: **var JS**; é diferente da variável: **var js**;

Por fim, a última regra é **evitar palavras chave**, palavras que são **reservadas** pelo JS, que são usadas para se programar em JavaScript. Por exemplo, a palavra **var** não pode ser usada como nome de uma variável, pois ela é usada para declararmos uma variável.

PALAVRAS RESERVADAS EM JAVASCRIPT

- break
- case
- catch
- continue
- debugger
- default
- delete
- do
- else
- false
- finally
- for
- function
- if
- in
- instanceof
- new
- null
- return
- switch
- this
- throw
- true
- try
- typeof
- var
- void
- while
- with

PALAVRAS RESERVADAS PELO NAVEGADOR

- alert
- blur
- closed
- document
- focus
- frames
- history
- innerHeight
- innerWidth
- lenght
- location
- navigator
- open
- outerHeight
- outerWidth
- parent
- screen
- screenX
- screenY
- statusBar
- window

PALAVRAS RESERVADAS PARA PROPRÓSITOS FUTUROS

- abstract
- boolean
- byte
- char
- const
- double
- enum
- export
- extends
- final
- float
- goto
- implements
- import
- int
- interface
- let
- long
- native
- package
- private
- protected
- public
- short
- static
- super
- synchronized
- throws
- transient
- volatile
- yeld

Tipos de Dados em JavaScript

No JavaScript, vamos trabalhar com os seguintes tipos de dados:

- Números
- Strings
- Booleanos
- Null
- Objetos
- Funções

Tipo de dado: **Número**

Esse é o mais óbvio tipo de dado.

O dia de seu nascimento é um número, seu futuro salário é um número, o ano de fabricação de um carro, a potência do motor desse carro, etc.

Basicamente, temos dois tipos de números:

- **Inteiro:** 1, 2, 3, 4, 5, 2112, -10 etc
- **Decimais:**

Em língua portuguesa, representamos números decimais, utilizando como separador da parte inteira a vírgula.

Preço: **R\$ 1,99**

Mas em programação, usamos **ponto** no lugar de **vírgula**: 1,99 para gente é **1.99** em computação.

Tipo de dado: String

Strings são, nada mais nada menos, que textos.

Uma palavra, uma frase, um caractere como: ! @ # \$ % "+, etc.

Uma string é sempre representada dentro de aspas. Exemplos de strings:

- "Olá mundo"
- "Campo Largo"
- "a"
- "b"
- "c"

Cuidado com variáveis que são números porém, são tratadas como strings:

- Número: **2112**
- : **"2112"**

Lembre que para poder operar matematicamente, a variável precisa ser número, jamais uma string.

Tipo de dado: Boolean

Eles podem assumir dois valores apenas:

- 1 – **True**
- 2 – **False**

Ou seja, **verdadeiro** e **falso**. Eles são chamados de valores lógicos e são muito utilizados nos testes e laços de repetição.

Convencionalmente, o número 0 é sinônimo de falso e tudo que for diferente de 0, isto é, - 1, 1, -2, 2, etc, é considerado **verdadeiro**.

Tipo de dado: Null

É a representação do vazio, do nada.

String nula: ""

Uma variável que não recebeu nenhum valor ou tipo de dado, é **inicializada como nula**, Null.

O vazio é um importante tipo de dado também.

Tipo de dado: Objetos

São as **'coisas'**.

Uma **imagem** é um objeto, um **botão** é um

objeto, um **parágrafo** específico pode ser um objeto, o **título da página**, um **formulário** etc.

É uma representação geral de alguma 'coisa'.

Tipo de dado: Função

Por fim, temos as funções, que **são blocos de código**, específicos, que fazem uma tarefa específica e podem ser **'chamados'** para serem executados.

Na aula 02 criamos duas funções de exemplo, a **mostraData()** cuja função é exibir a data e a **mostraMensagem()** que quando invocada, exibe uma frase (uma string).

Atribuindo e acessando variáveis

Em JavaScript, para atribuirmos um valor à uma variável usamos o símbolo de **igual (=)**.

Observe os exemplos a seguir:

```
1 var idade;
2 idade = 18;
3
4 var idade = 18;
5
```

O diagrama mostra o código acima com setas vermelhas e rótulos explicando cada linha:

- Uma seta aponta para a linha 1 (`var idade;`) com o rótulo "declaração da variável".
- Uma seta aponta para a linha 2 (`idade = 18;`) com o rótulo "atribuição da variável".
- Uma seta aponta para a linha 4 (`var idade = 18;`) com o rótulo "declaração e atribuição da variável".

Na primeira situação, primeiro se declarou a variável e em seguida, atribuiu-se o valor 18.

Na segunda situação, a declaração e a atribuição ocorreu na mesma linha de código.

Em ambos os casos temos um espaço de **memória reservado**, que pode ser acessado pela **var idade**, e no qual, está gravado o valor **18**.

```
var nome = "Olá mundo!";
```

Neste caso criamos uma variável, cujo nome é **nome** e que é do tipo string.

Agora vamos ver como acessar as variáveis que acabamos de declarar, para tanto vamos utilizar um **método** que permite imprimir na

janela do **navegador**, os valores armazenados pelas variáveis idade e nome.

Crie uma pasta em sua área de trabalho com o nome:

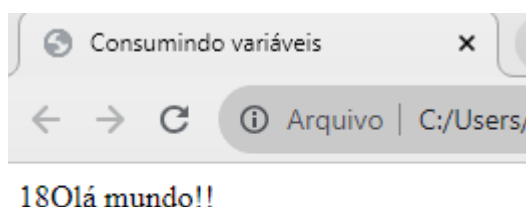
JAVASCRIPT_AULA04_SHIPDUPLA



Em seguidas, abra a pasta usando o VS Code e crie um arquivo **index.html**, inserindo o script abaixo:

```
index.html x
index.html > ...
1  <!DOCTYPE html>
2  <html lang="pt-br">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Consumindo variáveis</title>
8      <script type="text/javascript">
9          var idade = 18;
10         document.write(idade);
11         var nome = "Olá mundo!!";
12         document.write(nome);
13     </script>
14
15 </head>
16
17 <body>
18
19 </body>
20
21 </html>
```

Ao executarmos nosso script no navegador veremos:



Como você pode observar, os dois scripts foram executados e realizaram o que lhes foi solicitado, em especial o **consumo** das variáveis **idade** e **nome**.

TAREFA

Dentro da pasta da aula 04, crie um arquivo **tarefa.html** e nele, crie um site que, usando o método **document.write()**, exiba no navegador a seguinte mensagem: “Essa é a tarefa de SHIPDUPLA e somando as datas de nascimento dos membros o resultado é: _____”.

Obrigatória a criação das seguintes variáveis:

- ⇒ nome
- ⇒ data1
- ⇒ data2