



INTRODUÇÃO AO JAVASCRIPT



www.javascriptdescomplicado.com.br



JAVASCRIPT DESCOMPLICADO



IMPORTANTE

Este é um material de apoio desenvolvido pela **D&M Educacional** e utilizado em treinamentos e capacitações específicas.

Seu uso é permitido exclusivamente para os alunos que passaram por um treinamento oficial.

É proibido efetuar sua distribuição (cópia para terceiros, sites e redes sociais) sem a autorização expressa e assinada por um diretor da D&M Educacional.

Ao utilizar este material você estará concordando com as restrições de utilização.

Caso não concorde com esta licença, interrompa imediatamente seu uso, destrua qualquer cópia que tiver deste material em todos os seus dispositivos eletrônicos e contate a D&M Educacional para não cancelar sua inscrição no curso.

Havendo interesse em licenciar este material para utilização em outros locais, contate-nos pelo e-mail:

contato@dmeducacional.com.br.



JAVASCRIPT DESCOMPLICADO



Curso de Introdução ao JavaScript

- ☒ Introdução
- ☒ Tipos de dados
- ☒ Variáveis e constantes



Variáveis e Constantes

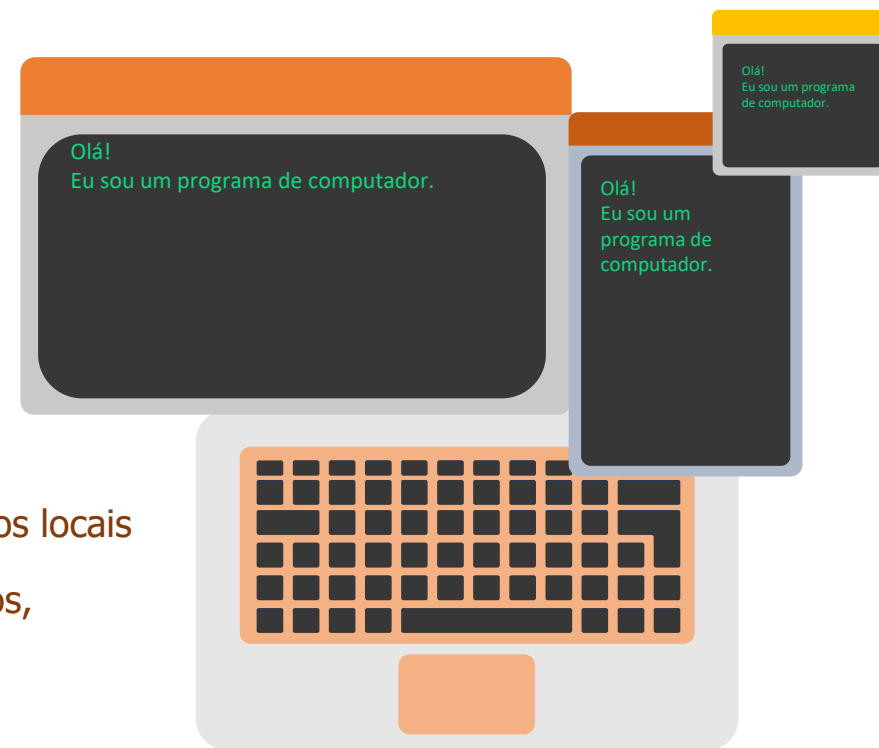
A memória do dispositivo

Diversos dispositivos (celular, tablete, notebook, computador, etc.) utilizam programas.

Um programa é um conjunto de comandos (instruções) que orientam como o computador deve funcionar executar determinada tarefa.

É muito comum que um programa precise trabalhar com dados. Estes dados podem vir de vários locais (teclado, banco de dados, clique do mouse em alguma opção) e serem de vários tipos (números, textos, etc.).

Um software trabalha com estes dados inicialmente armazenando-os na memória do dispositivo.



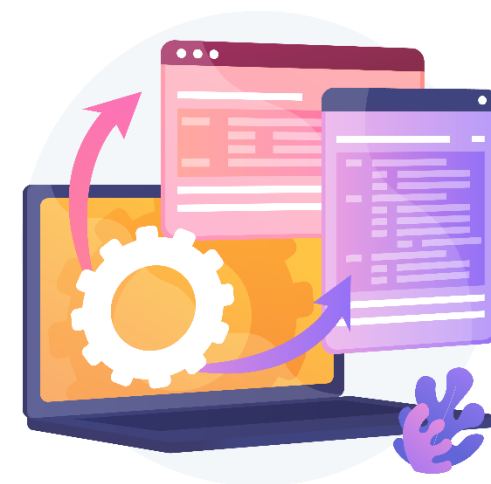
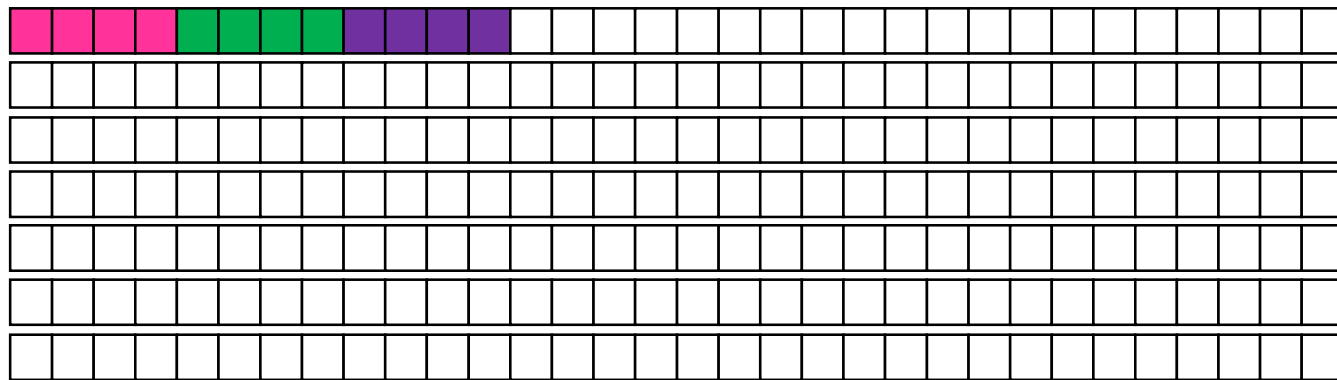
Variáveis e Constantes

A memória do dispositivo

Ao receber estes dados o programa transfere-os para a memória.

Variável é o termo técnico utilizado por uma linguagem de programação para um determinado espaço de memória que contém dados. Cada variável ocupa determinado espaço de memória, conforme o tipo de dado a ser armazenado.

Quando um programa precisa, ele acessa a variável da memória e manipula (cálculos) ou exibe seu dado (valor) na tela.



Variáveis e Constantes

Dando nomes às variáveis de memória

Em JavaScript existem diversas formas para se criar uma variável, porém, algumas regras na criação do **nome** de uma variável devem ser observadas:

- Uma variável deve ter seu nome iniciado por uma letra, pelo símbolo de sublinhado "_" ou de dólar "\$".
- Após a 1ª letra, é possível utilizar números em seu nome
- Não é permitido inserir espaços em nome de variáveis
- Não pode utilizar uma palavra reservada pela linguagem

Exemplos de nomes válidos:

`nome, hoje, acesso, dataViagem, SalarioLiquido, _ultimoAcesso, $valor2021`

Exemplos de nomes inválidos:

`20nomes` (começa com um número)

`nome da empresa` (tem espaço)

`var` (é palavra reservada do JavaScript)



Variáveis JavaScript são sensíveis à caixa. Nome e nome são diferentes.



Variáveis e Constantes

Dando nomes às variáveis de memória

Um nome de variável pode ser criado de várias formas:

- Totalmente em maiúsculo (**NOME**)
- Totalmente em minúsculo (**idade**)
- Apenas a 1ª letra diferenciada (**E**mpresa)
- Começando com sublinhado (**_**salario)
- Começando com sinal de dólar (**\$**taxa)
- Nome composto por 2 partes, onde a primeira é em minúsculo e a segunda começa com a letra maiúscula (**salarioBruto**)

Independente do padrão que você decida utilizar, busque criar nomes da forma mais simples possível, de tal maneira que um simples olhar em seu nome já permita a compreensão do tipo de dado e seus valores possíveis.



Variáveis e Constantes

Caracteres acentuados em nome de variáveis

Ainda que não seja proibido, ao criar um identificador (exemplo: nome de uma variável), uma boa prática é não utilizar acentos, pois problemas podem ocorrer em algumas circunstâncias.

Os arquivos armazenam internamente as suas letras em formato numérico (em bytes) e usam tabela auxiliar para mapear o número armazenado com o caractere correspondente (letra).

No início, existia apenas a tabela ASC (e posteriormente ASCII), porém, outras tabelas de caracteres foram criadas.

Conforme o padrão em que a página web é aberta, um caractere acentuado (exemplo: à ã ç) pode ser trocado por outro caractere.

Se os dados forem gravados num banco, eles podem sofrer variações (conforme a codificação de caracteres do banco de dados).

Desta forma, podem ocorrer problemas quanto ao entendimento do nome correto de identificadores.



Variáveis e Constantes

Variáveis e tipos de dados

Os tipos de dados armazenados em variáveis mais comuns são:

Inteiro – compreende o conjunto de número inteiros, podendo ser positivo, negativo ou até o número zero, como por exemplo:

```
saldo = -500
```

```
$Idade = 52
```

```
_temperatura = 0
```

```
capitalEmpresa = 2850900
```

Lembrei da aula de tipos de dados.

Não precisamos declarar os tipos das variáveis, porque o **JavaScript** faz isto automaticamente.

A variável `_temperatura` provavelmente será criada como do tipo `byte` sem sinal que compreende valores na faixa de `0` a `255`.



Variáveis e Constantes

Variáveis e tipos de dados

Real – Também conhecida como *float* (inglês) ou **ponto flutuante**, este tipo de dado compreende, além dos números inteiros (um real pode ser um inteiro), também o conjunto de números fracionários e das conhecidas dízimas periódicas (número sem fim após a casa decimal). Em função das limitações do computador, uma dízima periódica é aproximada (armazenada conforme um determinado número de casas).

Exemplos:

```
tamanhoMolecula = 0.00000000000012
```

```
$temperatura = -8.50
```

```
salarioAtual= 1043.20
```

```
__gramas= 0.280
```

```
contribuicaoSalario = 2850 * (8.5/100)
```



Variáveis e Constantes

Variáveis e tipos de dados

String – Este tipo de dado consegue armazenar o equivalente a texto, como uma letra, número ou símbolo, e deve estar delimitado por aspas que podem ser simples ou duplas. Exemplos:

```
notaFinal = 'B'
```

```
$nome = "Ana Clara silva"
```

```
posicao = '5'
```

```
_prefixo= "@"
```

```
site = "www.javascriptdescomplicado.com.br"
```

Números armazenados no formato de string são usados para exibição na tela e na maioria dos casos não podem ser usados em cálculos (existem casos em que a linguagem efetua uma conversão do tipo caractere para tipo numérico). Algumas funções e métodos permitem efetuar esta conversão:

Função/Método	Descrição	Exemplo
parseInt()	Converte para inteiro	parseInt("15.20") → 15
parseFloat()	Converte para float	parseFloat("0.000012") → 0.000012
Number()	Analisa e converte a string	Number("15") → 15 / Number("10.02") → 10.02



Variáveis e Constantes

Variáveis e tipos de dados

LÓGICO – Este tipo de dado pode ser do tipo `true` (verdadeiro) ou `false` (falso).

Nem todas as linguagens de programação usam este tipo de dado.

```
solteiro = false
```

```
_limiteAtingido = true
```

```
$falha = false
```

```
_LIMITE = true
```

```
$limiteAtingido = ($saldo < 0)
```

Condição que retorna `true` ou `false` conforme o valor de `$saldo`

Se `$saldo` for menor que 0 → `true`

Se `$saldo` for maior ou igual a 0 → `false`



Variáveis e Constantes

Variáveis e tipos de dados

Em algumas linguagens (C, JavaScript, etc.), uma variável é considerada falsa quando seu valor for igual a zero (0) e verdadeira quando for diferente de zero.

```
cadastroLiberado = 0 // número - representa false
```

```
$ativo = 1 // número - representa true
```

```
if($ativo) {  
    // faça isto
```

```
}
```

```
$saldo = -5200 // número - representa true
```

O **JavaScript** entende que uma expressão que retorne o valor 0 é compreendida como **false** e qualquer valor diferente de 0 é compreendido como **true**.



Expressão = algo que retorna um valor / dado (exemplo: um cálculo)



JAVASCRIPT

DESCOMPLICADO



O professor pediu desculpas, pois vai fazer uma inserção / explicação sobre os **Locais** onde o **JavaScript** pode ser inserido.

Depois ele continua a explicação de variáveis e constantes.

Curso de Introdução ao JavaScript

- ☒ Introdução
- ☒ Tipos de dados
- ☒ Variáveis e constantes
- ☒ O código JavaScript



O CÓDIGO JAVASCRIPT

Locais de criação

O código de programação da linguagem **JavaScript** pode ser criado em diversos locais.

Alguns exemplos são:

- Quase que em qualquer lugar dentro do código da página web (usualmente dentro da seção de corpo / body da página).
- Dentro de uma função JavaScript
- Como parte do código de criação de um objeto HTML
- Num arquivo externo



O CÓDIGO JAVASCRIPT

Locais de criação



```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="utf-8">
    <title>Página web</title>
  </head>
  <body>
    <h1>Minha página Web</h1>
    <p>Meu nome é Steve e sou um dos alunos deste curso.</p>
    <p>Estou achando bem interessante a forma como estou aprendendo.</p>
    <p>Hoje é
    <script>
      hoje = new Date()
      document.write( hoje.getDate()      + "/"      +
                      (hoje.getMonth()+1) + "/"      +
                      hoje.getFullYear())
    </script>
    </p>
  </body>
</html>
```



pag02.html

👉 A marca **<script>** informa ao navegador o início de um código de linguagem JavaScript.

👉 O comando **new** com a chamada da função **Date()** cria um objeto do tipo data. Este objeto tem dentro de si as informações do dia de hoje.

👉 Neste exemplo, utiliza-se o objeto **document** que representa todo o código HTML. Este objeto tem o método **write**, que consegue escrever algo na tela do navegador.

👉 O objeto **hoje** está sendo acessado e suas informações (dia atual, mês e ano) estão sendo exibidas na tela.

👉 O método **getMonth()** do objeto **hoje** retorna um mês de 0 a 11.

O CÓDIGO JAVASCRIPT

Locais de criação



Que interessante.

Minha página Web

Meu nome é Steve e sou um dos alunos deste curso.

Estou achando bem interessante a forma como estou aprendendo.

Hoje é **9/2/2021**

 O objeto `Date()` será melhor detalhado posteriormente.

O CÓDIGO JAVASCRIPT

Locais de criação



```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="utf-8">
    <title>Página web</title>
    <script>
      function BoasVindas() {
        alert("Seja bem vindo (a) à minha página web.")
      }
    </script>
  </head>
  <body onclick="BoasVindas()">
    <h1>Minha página Web</h1>
    <p>Meu nome é Steve e sou um dos alunos deste curso.</p>
    <p>Estou achando bem interessante a forma como estou aprendendo.</p>
  </body>
</html>
```



pag03.html

👉 Este exemplo mostra uma técnica que é a criação de uma **função**, código de programação JavaScript que executa uma tarefa e que recebe um nome.

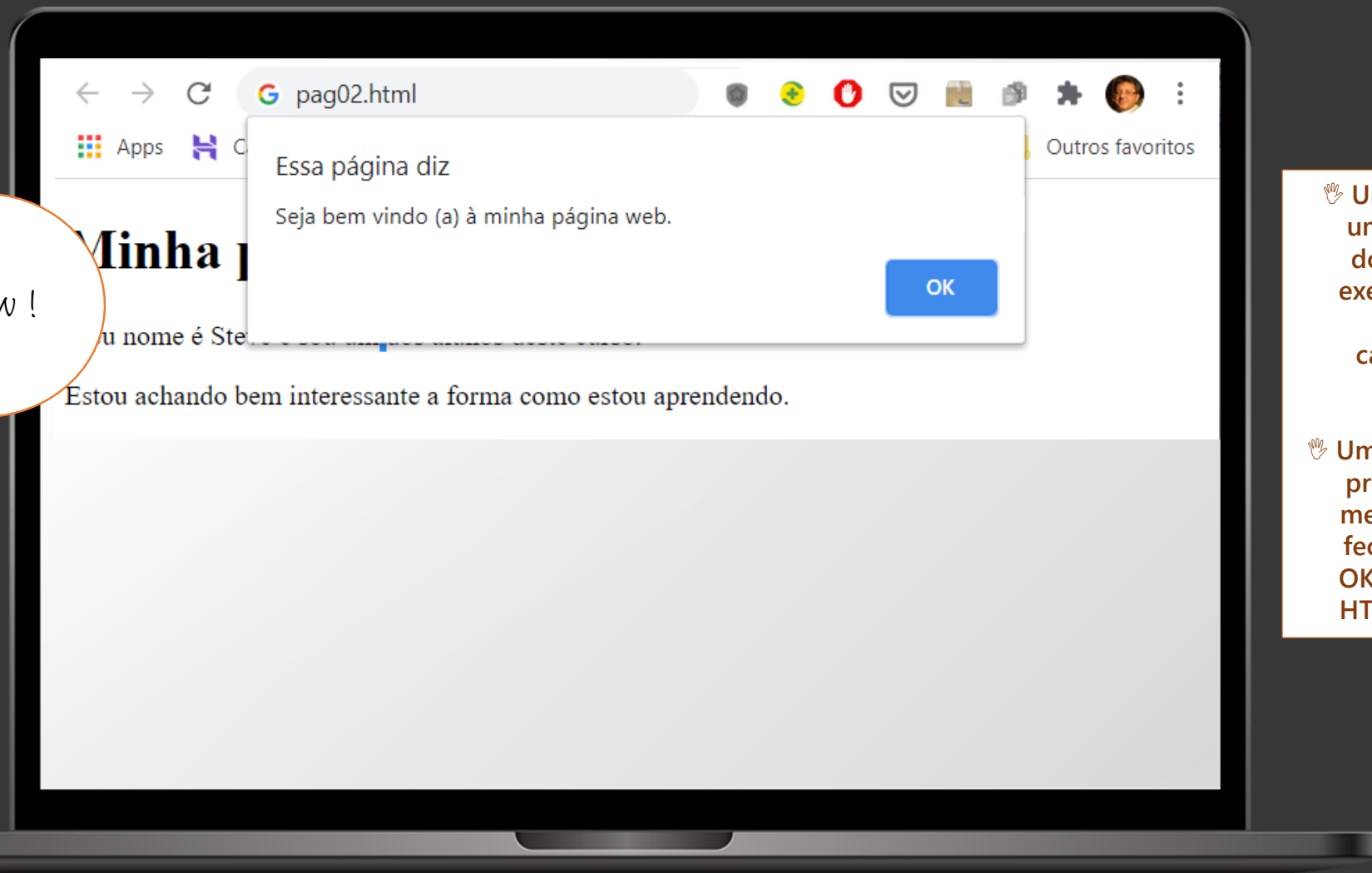
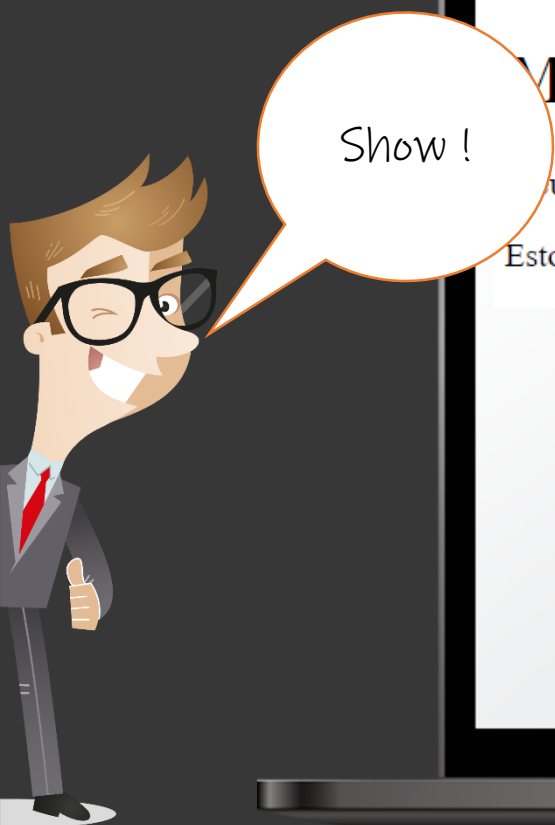
👉 Esta função só será executada se for chamada de forma explícita, dentro da página web.

👉 Foi interceptado o evento **onclick** da marca **<body>**. Este evento é executado se (e quando) o (a) usuário (a) clicar na página.

👉 Evite interceptar eventos que o usuário não está esperando ou tenha conhecimento, ou, coloque uma indicação na página caso queira prover algum tipo de interatividade.

O CÓDIGO JAVASCRIPT

Locais de criação



✎ Um dos problemas em inserir um código **JavaScript** no meio do código é o fato de que ele executa este código, na medida em que está fazendo o carregamento (e exibindo os elementos) da página.

✎ Uma janela **Modal** interrompe o processamento para exibir sua mensagem. O usuário precisará fechar a janela (clicar no botão OK) para que o resto do código HTML seja lido pelo navegador.

O CÓDIGO JAVASCRIPT

Locais de criação



```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="utf-8">
    <title>Página web</title>
    <script>
      function BoasVindas() {
        alert("Seja bem vindo (a) à minha página web.")
      }
    </script>
  </head>
  <body>
    <h1>Minha página Web</h1>
    <p>Meu nome é Steve e sou um dos alunos deste curso.</p>
    <p>Estou achando bem interessante a forma como estou aprendendo.</p>
    <p onmouseover="BoasVindas()"><b>Passe o mouse aqui</b></p>
  </body>
</html>
```

pag04.html

Este exemplo é uma variação do anterior e demonstra que é possível interceptar um evento de um objeto (passar o mouse sobre o parágrafo), desviando o fluxo do programa para uma função existente.



Esta função só será executada se o (a) usuário (a) passar o mouse sobre o parágrafo.

O CÓDIGO JAVASCRIPT

Locais de criação



É possível separar o código JavaScript num arquivo externo (**funções.js**), e fazer uma referência deste arquivo em qualquer página web desejada. Deixar o código em arquivos externos, separados da página web, possibilita reutilização de código, redução de digitação e de erros.

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="utf-8">
    <title>Página web</title>
    <script src="funcoes.js"> </script>
  </head>
  <body>
    <h1>Minha página Web</h1>
    <p>Meu nome é Steve e sou um dos alunos deste curso.</p>
    <p>Estou achando bem interessante a forma como estou aprendendo.</p>
    <p onmouseover="BoasVindas()"><b>Passe o mouse aqui</b></p>
  </body>
</html>
```

pag05.html

```
function BoasVindas() {
  alert("Seja bem vindo (a) à minha página web.")
}
```

funcoes.js

⚠ Cuidado para não colocar a marca `<script>` no arquivo externo.

O CÓDIGO JAVASCRIPT

Locais de criação

Em geral, quando se inicia o aprendizado de **JavaScript**, o local mais fácil é colocar o código dentro da página.

Os códigos rapidamente se tornam funções, na medida em que o aprendizado evolui.

Em alguns casos específicos pode-se colocar código dentro dos objetos, porém, é preciso prestar atenção, pois isto pode limitar o funcionamento do objeto da página.

Com o tempo, e em função das facilidades obtidas, os desenvolvedores utilizam arquivos externos com códigos em **JavaScript** como uma forma de separar o que é código do que é exibição (**HTML**).

Como é que Eu vou saber
se devo colocar o
JavaScript na página,
numa função, dentro de
um objeto ou num
arquivo externo?



JAVASCRIPT

DESCOMPLICADO



Curso de Introdução ao JavaScript

- ☒ Introdução
- ☒ Tipos de dados
- ☒ O código JavaScript
- ☒ Variáveis e constantes

Voltamos à nossa
programação anterior!!

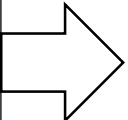
Variáveis e Constantes

Criação de variáveis



```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Criação de Variáveis</title>
06   </head>
07   <body>
08     <script>
09       var nome
10       nome = "Ana Cláudia"
11
12       document.write("Seja bem vinda, " + nome )
13     </script>
14   </body>
15 </html>
```

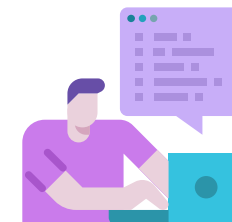
pag06.html



Seja bem vinda, Ana Cláudia

O código em destaque introduz a forma mais tradicional de criação de variáveis na linguagem **JavaScript** que é por meio do uso do comando **var**, seguido do nome da variável a ser criada (linha 09), e posteriormente (linha 10) a atribuição de seu valor (a string "**Ana Cláudia**").

- 👉 Declarar uma variável é o termo técnico usado em sua criação.
- 👉 Atribuição de valor significa colocar um dado/valor numa variável



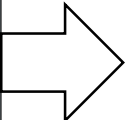
Variáveis e Constantes

Criação de variáveis



```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Criação de Variáveis</title>
06   </head>
07   <body>
08     <script>
09       var nome
10       nome = "Ana Cláudia"
11       var _idade = 22 , $cargo = "programadora" , Salario= 2500.00
12
13       document.write(nome + " é " + $cargo + " e ganha R$ " + Salario )
14     </script>
15   </body>
16 </html>
```

pag07.html

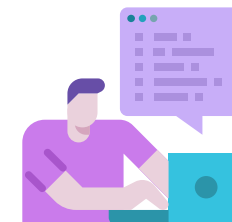


Ana Cláudia é programadora e
ganha R\$ 2500

Variações do comando **var** podem ser verificadas na linha 11, onde diversas variáveis são criadas e na mesma linha têm seus valores atribuídos.

O objeto **document** (linha 13) representa o HTML contido na página e a chamada de seu método **write** permite escrever algo na página.

Note que para exibir variáveis e textos foi utilizado o sinal de "+" que atua como concatenador de strings.



Variáveis e Constantes

Criação de variáveis



É possível criar variáveis no JavaScript sem utilizar o comando `var`:

```
...  
08  
09     <script>  
10         nome = "Ana Cláudia"  
11         _idade = 22 , $cargo = "programadora" , Salario= 2500.00  
12         document.write(nome + " é " + $cargo + " e ganha R$ " + Salario )  
13  
14         nome = "José Antunes"  
15         _idade = 35 , $cargo = "Gerente de TI" , Salario= 5350.20  
16         document.write("<br>" + nome + " é " + $cargo + " e ganha R$ " + Salario)  
17     </script>  
18 </body>  
    </html>
```

pag08.html

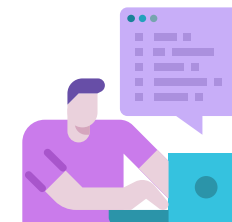
Ana Cláudia é programadora e
ganha R\$ 2500
José Antunes é Gerente de TI e
ganha R\$ 5350.2

Ao não utilizar o comando `var`, as variáveis são criadas com **escopo global**.

Posteriormente será explicada a questão do escopo de variáveis e analisadas as implicações de se utilizar ou não o comando `var`.

Por ser uma linguagem **fracamente tipada**, as variáveis têm seu tipo definido no momento da atribuição de seus valores.

As variáveis podem ter seus valores mudados, posteriormente, no programa.



Ao enviar o texto `
` para o navegador (linha 15), ele o trata como um comando de quebra de linha HTML

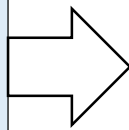
Variáveis e Constantes

Criação de constantes



```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Criação de Variáveis</title>
06   </head>
07   <body>
08     <script>
09       const nomeEmpresa = "Extreme Programming Corporation", vagas=50
10
11       document.write(nomeEmpresa + " está c/" + vagas + " vagas de TI.")
12     </script>
13   </body>
14 </html>
```

pag09.html



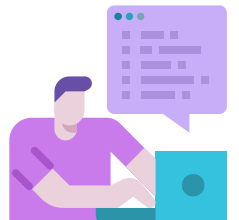
Extreme Programming
Corporation está c/ 50 vagas
de TI.

Uma constante é um espaço em memória que armazena um valor que não muda.

Em geral, ela é utilizada numa linguagem de programação para ter dados fixos registrados, que podem ser inseridos em mensagens ou usados na criação de variáveis.

É preciso tomar cuidado para não alterar o valor de uma constante, caso contrário, ocorrerá um erro no funcionamento do programa.

👉 É necessário atribuir (inicializar) a constante no momento de sua declaração (criação)



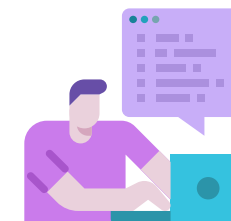
Operadores

Operadores Aritméticos



São aqueles que realizam cálculos aritméticos básicos.

Operador	Função
+	Operador de adição
-	Operador de subtração (ou negação)
*	Operador de multiplicação
/	Operador de divisão
%	Operador de módulo (resto de divisão inteira)
++	Operador de incremento
--	Operador de decremento
**	Operador de Exponenciação



Operadores

Operadores de soma + e subtração -

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Operador Aritmético - Soma e Subtração</title>
06   </head>
07   <body>
08     <script>
09       var salario= 1720.52
10       var abono= 780
11       var desconto= 220
12       var total= (salario + abono - desconto)
13       alert ("Total do Salário do mês: " + total)
14     </script>
15   </body>
16 </html>
```

salario.html



Não use vírgula para separar a unidade da casa decimal (sempre ponto).



Operadores

Operador de divisão /

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Operador Aritmético - Divisão</title>
06   </head>
07   <body>
08     <script>
09       var divida= 12890.15
10       var numeroParcelas= 12
11       var valorParcela=(divida/ numeroParcelas)
12       document.write("Valor de cada parcela é : " + valorParcela)
13     </script>
14   </body>
15 </html>
```

divida.html



Operadores

Operadores de multiplicação *

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Operador Aritmético - Multiplicação</title>
06   </head>
07   <body>
08     <script>
09       var valorDependente = 230.25
10       var dependentes = 3
11       var total =(valorDependente * dependentes)
12       document.write("Desconto de dependentes é : " + total )
13     </script>
14   </body>
15 </html>
16
```

multiplica.html



Operadores

Operadores de resto de divisão inteira %

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Operador Aritmético - Resto de Divisão inteira</title>
06   </head>
07   <body>
08     <script>
09       var divida = 18790
10       var valorParcela= 1250
11       var numParcelas= divida / valorParcela
12       var saldo = divida % valorParcela
13
14       document.write( "Você deverá pagar " + numParcelas +
15                       " parcelas de " + valorParcela +
16                       " e haverá uma sobra de " + saldo )
17     </script>
18   </body>
19 </html>
```

resto.html



Operadores

Operadores Aritméticos – Incremento ++ e Decremento --

O operador de incremento (++) foi criado para agilizar o processamento de operações que necessitam que uma variável seja incrementada em 1.

Da mesma forma foi criado o operador de decremento (--) que subtrai em 1 a variável aplicada.

Um incremento ou decremento pode ocorrer em dois momentos:-

- Antes de uma operação
- Após uma operação

Quando ocorre antes da operação matemática, o operador é conhecido como pré-incremento (ou pré-decremento).

Quando ocorre posteriormente, ele é conhecido como pós-incremento ou pós-decremento.

Exemplos

soma++

++ItensFaltantes

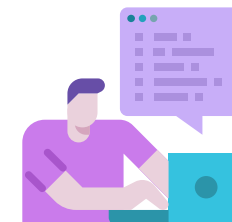
Exemplos

subtracao--

--contador



Estes operadores só alteram o valor das variáveis que estão sendo aplicadas



Operadores

Operadores Aritméticos – Incremento ++ e Decremento --

Algo muito comum é a necessidade de uma operação alterar o valor de uma variável após efetuar um cálculo. Exemplo:

```
...  
    <script>  
        var total = 0, contador = 1  
  
        total = contador  
        contador = contador + 1  
    </script>  
...
```

Parte desta operação pode ser simplificada:

```
total = contador++
```

Neste exemplo, primeiro será feita a operação matemática (inserir na variável **total** o valor da variável **contador**, para somente em seguida incrementar a variável **contador**).

Qual será o valor de total e contador, após a execução deste código?



Operadores

Operadores Aritméticos – Incremento ++ e Decremento --

Algo muito comum é a necessidade de uma operação alterar o valor de uma variável após efetuar um cálculo e os operadores de incremento e decremento também podem ajudar. Exemplo:-

```
...  
    <script>  
        var a= 1, b= 3  
  
        soma = (++ a + b ++)  
  
    </script>  
...
```

Neste exemplo, as variáveis **a** e **b** são inicializadas respectivamente com os valores 1 e 2. A variável **soma** recebe a soma das 2 variáveis.

Ocorre que a variável **a** está sendo **pré-incrementada** e desta forma esta operação de incremento é executada primeiro, fazendo com que o valor de **a** mude para 2, antes do cálculo que define o valor da variável **soma**.

A variável **b**, embora também esteja sendo incrementada, está sendo **pós-incrementada**, o que significa que ela só será incrementada após execução da operação matemática. Assim, a variável **soma** receberá o valor de **a**, (que vale 2) somada à **b**, (que vale 3, pois ainda não foi incrementada). Após a operação da soma ser executada, **b** é incrementada para 4.



Operadores

Operador de Exponenciação

O JavaScript facilita o cálculo de exponenciação por meio de um operador próprio. Este tipo de cálculo é realizado pela multiplicação de um número, por ele mesmo, um certo número de vezes.

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Operador Aritmético - Resto de Expoente</title>
06   </head>
07   <body>
08     <script>
09       var base = 4, expoente = 3
10       var potencia = base ** expoente
11
12       document.write(base + " elevado a " + expoente + "=" + potencia )
13     </script>
14   </body>
15 </html>
```

expoente.html

Neste exemplo, o valor 4 (base) é elevado à potência 3 (expoente), o que equivale à conta:

$4 \times 4 \times 4 \rightarrow 16 \times 4 \rightarrow 64$



Prática de JavaScript

Criação de variáveis e aritmética básica

Será feita uma prática da aprendizagem para colocar em prática um pouco do que foi estudado.

Serão criados alguns formulários básicos e algumas ações executadas serão um pouco mais avançadas, as quais serão explicadas com maior detalhe posteriormente no curso.

Será feita a explicação básica necessária para o adequado entendimento da prática proposta.



Prática de JavaScript

[01] Cálculo do preço da compra 1/3

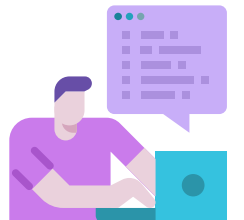


O programa a seguir tem variáveis criadas referente o valor de um produto e a quantidade que o usuário deseja comprar.

O objetivo desta prática é alterar a página web, de forma a **efetuar o cálculo da compra e exibir a informação do valor da compra na tela.**

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Valor da Compra 1</title>
06   </head>
07   <body>
08     <script>
09       var preco, quantidade
10       preco = 25.14
11       quantidade = 7
12     </script>
13   </body>
14 </html>
```

pag10.html



Prática de JavaScript

[01] Cálculo do preço da compra 2/3 - RESOLUÇÃO

A resolução é relativamente simples e pode ser feita conforme os passos a seguir.

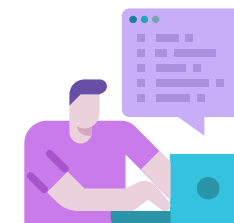
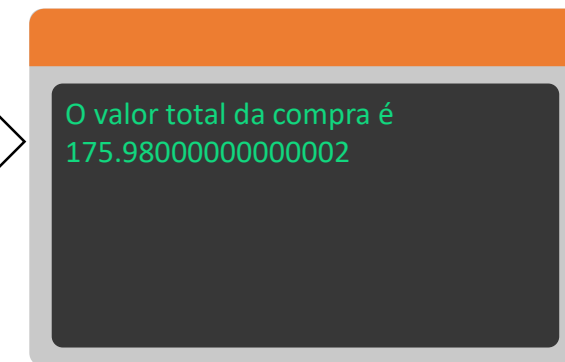
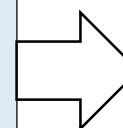
Linha 12: Criar a variável com o valor total da compra

Linha 12: Atribuir na variável total o cálculo

Linha 13: Exibir a variável na tela, com um texto explicativo

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Valor da Compra 1</title>
06   </head>
07   <body>
08     <script>
09       var preco, quantidade
10       preco = 25.14
11       quantidade = 7
12       var total = preco * quantidade
13       document.write("O valor total da compra é " + total )
14     </script>
15   </body>
16 </html>
```

pag10.html



Prática de JavaScript

[01] Cálculo do preço da compra 3/3 – RESOLUÇÃO

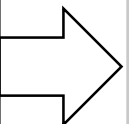


Verificou-se que o valor resultante da multiplicação foi gerado com um grande número de casas decimais e um valor ínfimo que não bate com a conta. Este é um bug matemático muito conhecido (JavaScript) e tem a ver com a forma pela qual o processador lida com a precisão de números com casas decimais.

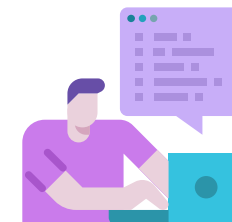
Para contornar este problema, será utilizado o método `toFixed()`, existente num objeto do tipo número (`Number`), que permite exibir um determinado número conforme as casas decimais desejadas. Neste caso, serão exibidas apenas as 2 primeiras casas decimais do número.

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Valor da Compra 1</title>
06   </head>
07   <body>
08     <script>
09       var preco, quantidade
10       preco = 25.14
11       quantidade = 7
12       var total = preco * quantidade
13       document.write("O valor total da compra é " + total.toFixed(2) )
14     </script>
15   </body>
16 </html>
```

pag12.html



O valor total da compra é 175.98



Prática de JavaScript

Considerações



Apesar de ser possível criar variáveis e efetuar cálculos com base nas mesmas, usando a linguagem **JavaScript**, o ideal é permitir que o usuário tenha a liberdade de informar os valores desejados.

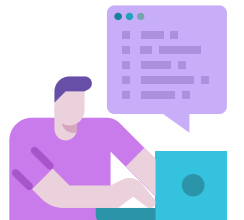
Uma das formas de interação com o usuário que podem ser feitas na página web por meio do uso do **JavaScript** é por meio da função `prompt()`.

Esta função permite exibir um texto ao usuário e solicitar o preenchimento de algum tipo de dado.

Ela faz parte do objeto **BOM** (**Browser Object Model**), uma estrutura de memória que armazena objetos e métodos do navegador.

Este objeto traz outros 2 métodos interessantes que são o `alert()`, que exibe um alerta na tela, e aguarda o usuário clicar no botão para fechar a janela e o método `confirm()` que exibe um texto na tela e aguarda o (a) usuário (a) clicar numa das opções possíveis.

A explicação, a seguir, demonstra o funcionamento do método `prompt()`.



Prática de JavaScript

[02] Cálculo do preço da compra – 1/2



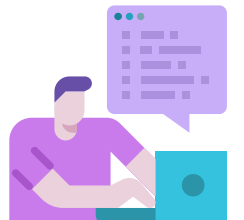
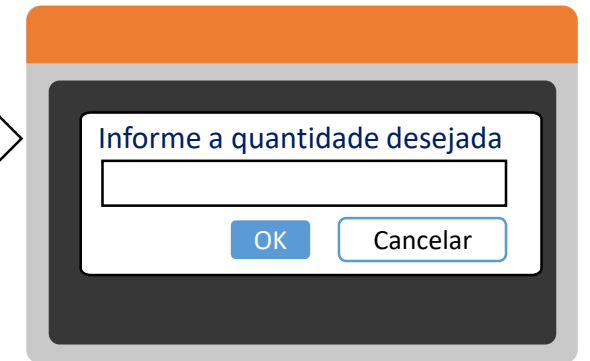
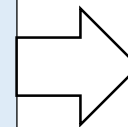
As variáveis são preparadas para uso na linha 09. O preço do produto está fixado em 25.14.

O método `prompt()` exibe uma janela de diálogo com o texto "Informe a quantidade desejada" (linha 10), uma caixa de texto e 2 botões (OK e Cancelar). Em seguida, aguarda-se o (a) usuário (a) digitar algo.

Após digitar e teclar <ENTER> (ou cancelar), o valor total é calculado (linha 12) e exibido na tela (linha 13).

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Valor da Compra 2</title>
06   </head>
07   <body>
08     <script>
09       var preco = 25.14, quantidade
10       quantidade = prompt("Informe a quantidade desejada")
11
12       var total = preco * quantidade
13       document.write("O valor total da compra é " + total.toFixed(2) )
14     </script>
15   </body>
16 </html>
```

pag13.html



Prática de JavaScript

[02] Cálculo do preço da compra – 2/2

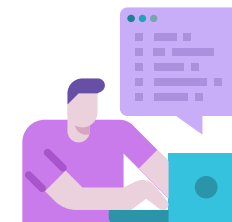
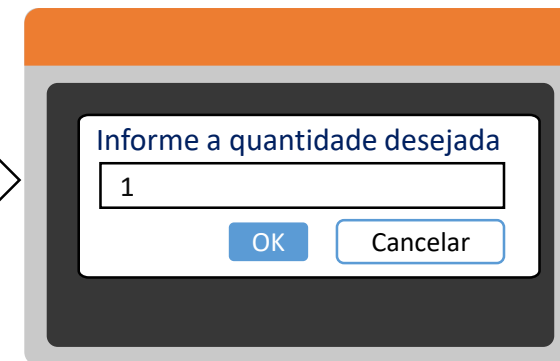
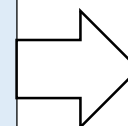


O método `prompt()` aceita um 2º argumento que é um valor padrão a ser exibido dentro da caixa (linha 10).

Será adicionado ao código algumas funcionalidades. A 1ª delas é verificar se o (a) usuário (a) digitou um valor numérico válido. Para isto, será utilizada a função `isNaN()`, que verifica se o argumento passado não é do tipo numérico. Na confirmação de que o objeto não é numérico, será emitido um aviso para o (a) usuário (a).

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Valor da Compra 2</title>
06   </head>
07   <body>
08     <script>
09       var preco = 25.14, quantidade
10       quantidade = prompt("Informe a quantidade desejada", "1")
11       if(isNaN(quantidade)) {
12         alert("A quantidade informada não é válida.")
13       }
14       else {
15         var total = preco * quantidade
16         document.write("O valor total da compra é " + total.toFixed(2) )
17       }
18     </script>
19   </body>
20 </html>
```

pag13.html



Prática de JavaScript

Considerações



Ainda que haja a possibilidade de interação e coleta de dados junto ao (à) usuário (a), numa página web isto traz algumas dificuldades, como por exemplo, alguns textos de informação não mais ficam visíveis na interface, e aquilo que foi digitado pode gerar dúvidas aos usuários.

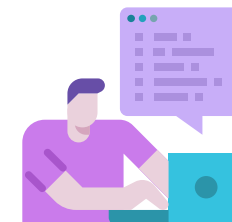


Realmente, hoje estou tão distraído que não sei se digitei **5** ou **7**. E pra complicar, a janelinha sumiu da tela..

Este tipo de funcionalidade deve ser considerada de forma pontual (em alguns casos) e com parcimônia.

Considerando que é importante ao usuário saber o que ele digitou, bem como qual é a base pela qual a página traz um determinado resultado, outras formas de interação para este tipo de problema devem ser consideradas.

Uma das mais utilizadas é o uso de formulários web, que pode atender de melhor forma o problema apresentado.



Prática de JavaScript

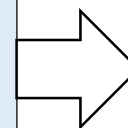
[03] Cálculo do preço da compra – 1/6



Será criada uma página com um formulário para exibição e coleta de dados junto ao (à) usuário (a):

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Valor da Compra 3</title>
06   </head>
07   <body>
08     Preço: <b>25,14</b> <br>
09     Quantidade (até 99):
10     <input type="number" maxlength="2" value="1">
11     <hr>
12     Total: <input type="number" readonly>
13     <br>
14     <input type="button" value="Calcular">
15   </body>
16 </html>
```

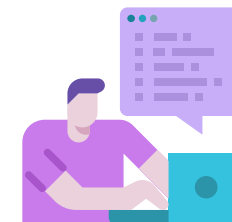
pag14.html



Preço: 25,14
Quantidade (até 99): <input type="text" value="1"/>
Total: <input type="text"/>
<input type="button" value="Calcular"/>

Esta página:

- Exibiu o valor do produto de forma textual e fixa (linha 08)
- Criou uma caixa de número que traz o valor padrão 1 e limita o valor máximo (linha 10).
- Criou uma caixa de número não editável (*read only*) para mostrar o valor total da conta (linha 12).
- Criou o botão Calcular que, porém, que ainda não faz nada.
- Será criado o código JavaScript que efetuará o cálculo, ao se clicar no botão.



Prática de JavaScript

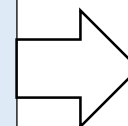
[03] Cálculo do preço da compra – 2/6



O código JavaScript será gravado no arquivo `calculo.js`, o qual será utilizado pela página web.

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Valor da Compra 3</title>
06     <script src="calculo.js"></script>
07   </head>
08   <body>
09     Preço: <b>25,14</b> <br>
10     Quantidade (até 99):
11     <input type="number" id="qt" maxlength="2" value="1">
12     <hr>
13     Total: <input type="number" id="tot" readonly>
14     <br>
15     <input type="button" value="Calcular" onclick="Calcular()" >
16   </body>
17 </html>
```

pag14.html

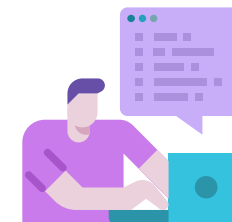


Preço: 25,14
Quantidade (até 99): <input type="text" value="1"/>
Total: <input type="text"/>
<input type="button" value="Calcular"/>

As caixas de número receberão uma identificação (atributo `id`) para possibilitar ao JavaScript o acesso e alteração (linhas 11 e 13). O botão de cálculo será alterado para chamar a função JavaScript `Calcular()`, que será criada dentro do arquivo `calculo.js`.



Um dos erros mais comuns no início do aprendizado é deixar a marca `<script>` aberta. Lembre-se sempre de fechá-la.



Prática de JavaScript

[03] Cálculo do preço da compra – 3/6



Código JavaScript – versão 1.0 - arquivo **calculo.js**

É necessário acessar a caixa de número onde o (a) usuário (a) digitou a quantidade a ser adquirida.

Esta caixa foi identificada no formulário como **qt**.

```
<input type="number" id="qt" maxlength="2" value="1">
```

No **JavaScript**, o método **getElementById** do objeto **document** consegue acessar um objeto pelo seu **id**.

É possível criar um objeto, no **JavaScript**, vinculado ao objeto **qt** criado no HTML. Desta forma, qualquer alteração que se faça neste objeto (no **JavaScript**) reflete automaticamente na tela.

```
var obj = document.getElementById("qt")
```

Ocorre que não é isso o que se precisa. A única coisa que se precisa deste objeto é recuperar o que foi digitado dentro dele.

Desta forma, não existe a necessidade de se criar um objeto no **JavaScript**.

O que se deseja é pegar o valor digitado dentro desta caixa de número.

Este valor encontra-se na propriedade **value**.

Será criada uma variável (**JS**) para armazenar o **value** do objeto **qt**.

```
var qtd = document.getElementById("qt").value
```



Prática de JavaScript

[03] Cálculo do preço da compra – 4/6



Código JavaScript – versão 1.0 - arquivo **calculo.js**

Tudo aquilo que vem de um formulário web, vem na forma de String (texto).

É necessário converter o valor digitado, armazenado na forma de String para um número.

Considerando que espera-se a digitação de um número inteiro para o campo qt, será usada a função `parseInt()` que efetua este tipo de conversão:

```
var qtd = parseInt(document.getElementById("qtd").value)
```

O próximo passo consiste em validar a quantidade obtida, para verificar se o (a) usuário (a) digitou um número inteiro. Para isto, será utilizada a estrutura condicional **if..else**.

```
if( (qtd < 1) || ( qtd > 99) )  
{  
    alert("A quantidade informada é inválida.")  
}
```

O trecho de código acima contém a parte do **if()** que verifica se a quantidade for menor que 1 ou maior que 99, e, em caso afirmativo, emite um alerta na tela.



Se a caixa de número ficou em branco ou foi digitado um valor negativo, a rotina acima será acionada.



Prática de JavaScript

[03] Cálculo do preço da compra – 5/6



Código JavaScript – versão 1.0 - arquivo **calculo.js**

Verifica-se agora a estrutura condicional completa, com o else.

```
if( (qtd <1) || ( qtd > 99) )
{
    alert("A quantidade informada é inválida.")
}
else
{
    var obTot = document.getElementById("tot")
    var total = 25.14 * qtd
    obTot.value = total
}
```

O else (senão), é executado se o **if()** for falso, ou seja, se **qtd** for maior que zero e menor que 100.

Neste caso, ocorrerá o seguinte:

- Será criado um objeto (**obTot**) que aponta para a caixa de número do formulário que armazena o total da conta
- Será criada a variável **total** calculando o valor unitário do produto x a quantidade digitada
- O valor da variável **total** será inserido no atributo **value** do objeto **obTot**, desta forma, automaticamente atualizando o objeto **tot**, na tela do navegador.



Prática de JavaScript

[03] Cálculo do preço da compra – 6/6



Código JavaScript – versão 1.0 - arquivo **calcula.js**

É criada a função Calcular() com todo o código JS feito.

A página já está pronta pra funcionar.

```
01 // arquivo calcula.js
02 function Calcular() {
03     var qtd = parseInt(document.getElementById("qt").value)
04     if( (qtd < 1) || ( qtd > 99) )
05     {
06         alert("A quantidade informada é inválida.")
07     }
08     else
09     {
10         var obTot = document.getElementById("tot")
11         var total = 25.14 * qtd
12         obTot.value = total
13     }
14 }
```

calcula.js

👉 Sempre grave/salve o arquivo antes de rodar.



Prática de JavaScript

[04] Eventos



Algo muito comum em nossas vidas são alguns eventos, como festas de aniversários, reuniões de comitês e até aquele churrasco regular com alguns membros da família.

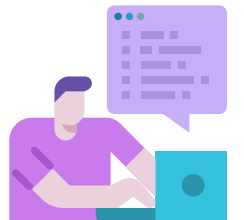
Nosso objetivo, nesta prática é criar uma página com um formulário que aceite valores das despesas elencadas, e que calcule uma estimativa de custo de um determinado evento.

Vamos supor a seguinte necessidade para cálculo do custo estimado de um determinado evento:

- Custo com pessoal envolvido
- Custo com as comidas
- Custo com bebidas
- Custo com a locação
- Outros custos

ESTIMATIVA DE CUSTOS DE EVENTO	
Pessoal	<input type="text"/>
Comidas	<input type="text"/>
Bebidas	<input type="text"/>
Locação	<input type="text"/>
Outros	<input type="text"/>
Custo Total	<input type="text"/>

CALCULAR



Até a próxima aula !



Prof. Carlos Majer

contato@carlosmajer.com.br

(11) 9.9695-4871



www.javascriptdescomplicado.com.br