

JavaScript

Módulo Básico





JAVASCRIPT

- **JavaScript**

- A **linguagem JavaScript** foi criada pela **Netscape Communications Corporation** junto com a **Sun Microsystems**
- **1ª versão** foi lançada em **1995** (na época, quem dominava o mercado de **browsers** era o **Netscape Navigator**)
- Em **1996**, a **Netscape** decidiu **entregar** o **JavaScript** para a **ECMA** (**E**uropean **C**omputer **M**anufacturers **A**ssociation)
- A **ECMA** é uma **associação** dedicada à **padronização** de **sistemas de informação**
- Em **1997**, foi lançada a **1ª edição** da **linguagem gerenciada** por essa **associação**

- **JavaScript**

- Por isso, a linguagem **JavaScript** também é chamada de **ECMAScript** e as versões da linguagem estão associadas a esse nome (**ECMAScript6**, **ECMAScript7**, **ECMAScript2018** etc.)
- Importante papel no processo de desenvolvimento de páginas para Web, junto com **HTML** e **CSS**
- **HTML** é utilizado para descrever o conteúdo de uma página web e definir a marcação semântica (**significado**) dos elementos que compõem a página
- **CSS** determina os estilos e a formatação dos elementos (**aparência**)

- *JavaScript*

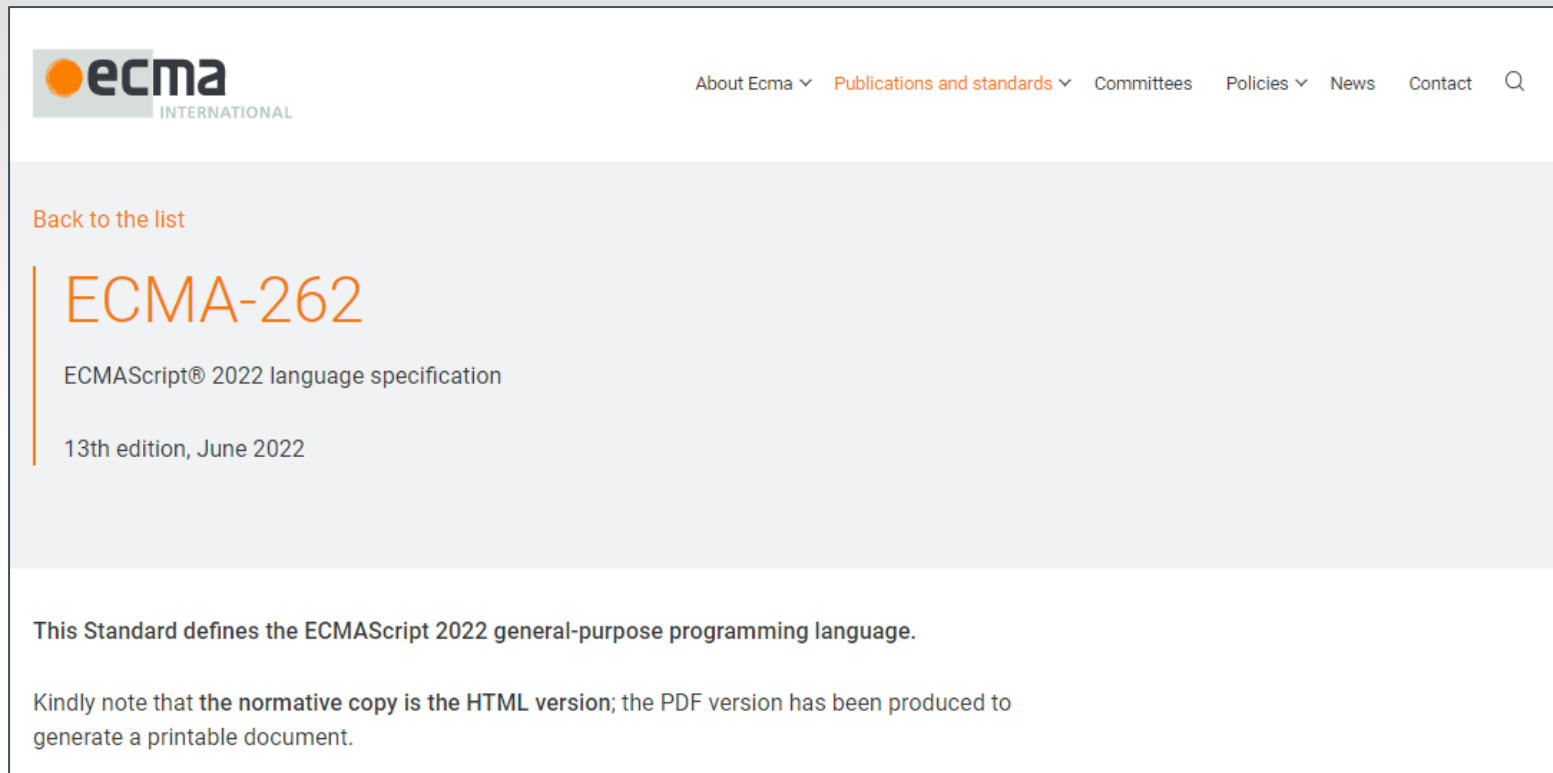


Figura 1 – Últimas novidades das versões ECMAScript
<https://www.ecma-international.org/>

JavaScript

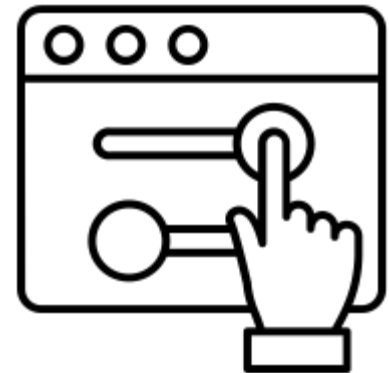
- **JavaScript**
 - *Utilizada para definir o comportamento dos elementos da página*
 - *Os códigos escritos em JavaScript, também chamados de **scripts**, são interpretados diretamente pelos navegadores web*



- ***JavaScript***

- ***As funcionalidades que podem ser inseridas em páginas web a partir da linguagem JavaScript são inúmeras***

- ***Podemos interagir com os usuários de uma página a partir de campos de formulários***
 - ***acessar e modificar o conteúdo e as características de uma página***
 - ***salvar informações no navegador do usuário***
 - ***auxiliar na construção de layouts complexos***
 - ***exibir opções de compras***



JavaScript

- **JavaScript**

- *As funcionalidades que podem ser inseridas em páginas web a partir da linguagem JavaScript são inúmeras*
 - *alterar características de produtos de acordo com as escolhas do cliente*
 - *criar jogos interativos que rodam no browser em qualquer plataforma (Windows, Mac e Linux), e em qualquer dispositivo (computador, tablet ou celular)*



JavaScript

- **JavaScript**

- *Algumas pesquisas de abrangência **mundial** destacam a linguagem de programação **JavaScript**, colocando-a no topo de suas classificações*
- *Essas pesquisas utilizam diversos fatores para medir o **ranking** das **linguagens**, como análise dos arquivos publicados no repositório **GitHub** ou pesquisas realizadas em sites como o **Stack Overflow***



- **JavaScript**
 - **Ranking das linguagens do GitHub**

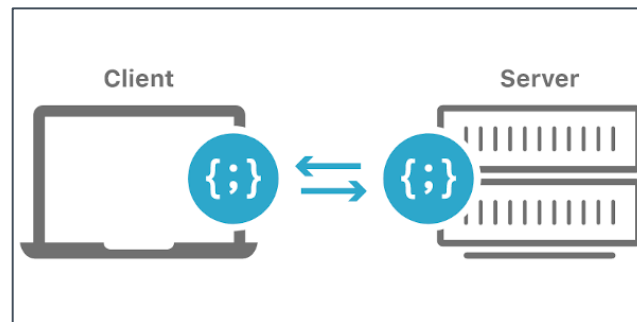


Figura 2 – JavaScript é destaque no ranking das linguagens do GitHub
<https://octoverse.github.com/#geographical-distribution-of-active-users>
Aula 01 | Módulo Básico

JavaScript

- **JavaScript**

- *No processo de programação de sistemas web, há linguagens que rodam no **lado** do **cliente** e que rodam no **lado** do **servidor***
- *JavaScript é utilizada principalmente para rodar scripts no **lado** do **cliente**, embora também seja crescente o número de aplicações desenvolvidas com a linguagem para rodar no **lado** do **servidor***





Mas o que significa
dizer que uma
linguagem roda no
lado do cliente?



JavaScript

- **Client Side**

- O próprio **navegador web** (*Chrome, Internet Explorer/Edge, Firefox, Safari, Opera*) deve conter **funcionalidades capazes de interpretar o código JavaScript e executá-lo**

- **Server Side**

- As **linguagens que rodam no lado do servidor** são **executadas em uma programa instalado no servidor web** (*Apache, IIS*) e **retornam para a máquina do cliente apenas o código resultante dessa execução** (*PHP, ASP, Java, C#, Python e JavaScript*)



- **Editores de Código**

- Para **criar programas JavaScript**, podemos utilizar **editores simples** que já estão **instalados no computador**, **editores online disponíveis na internet** ou **instalar programas profissionais de edição de código**
- **Editores simples, como bloco de notas no Windows, não são muito recomendados pela ausência de recursos de auxílio ao programador**
- **Os editores online contêm alguns recursos extras e a vantagem de poder acessar e compartilhar os códigos na internet**

- **Editores de Código**

- **Editores profissionais** contêm diversos recursos que facilitam o desenvolvimento dos programas
- **Editores online** permitem testar diretamente os programas em um navegador web (**w3schools.com** e **js.do**)
- **Editores profissionais** apresentam uma série de recursos que nos auxiliam no desenvolvimento de aplicações (autocompletar os comandos (**IntelliSense**), alertas de erros de sintaxe, formatação (**indentação**) de código, cores diferentes para os comandos e integração com **GitHub**)

- *Saída de dados com `alert()` e `console.log()`*
 - *Exemplo (01):*
 - *Apresentar uma mensagem ao usuário*

```
html > S exemplo_1.html > ...
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8      <script>
9          alert("Bem-vindo ao bkBank Academy")
10         console.log("Meu primeiro programa em JS")
11     </script>
12
13     <title>Exemplo 01 | bkBank Academy</title>
14 </head>
15 <body>
16     <!-- Conteúdo do body -->
17 </body>
18 </html>
```

Figura 3 – Script para apresentar uma mensagem ao usuário
Aula 01 | Módulo Básico

- ***Saída de dados com `alert()` e `console.log()`***
 - ***Exemplo (01):***
 - ***Apresentar uma mensagem ao usuário***

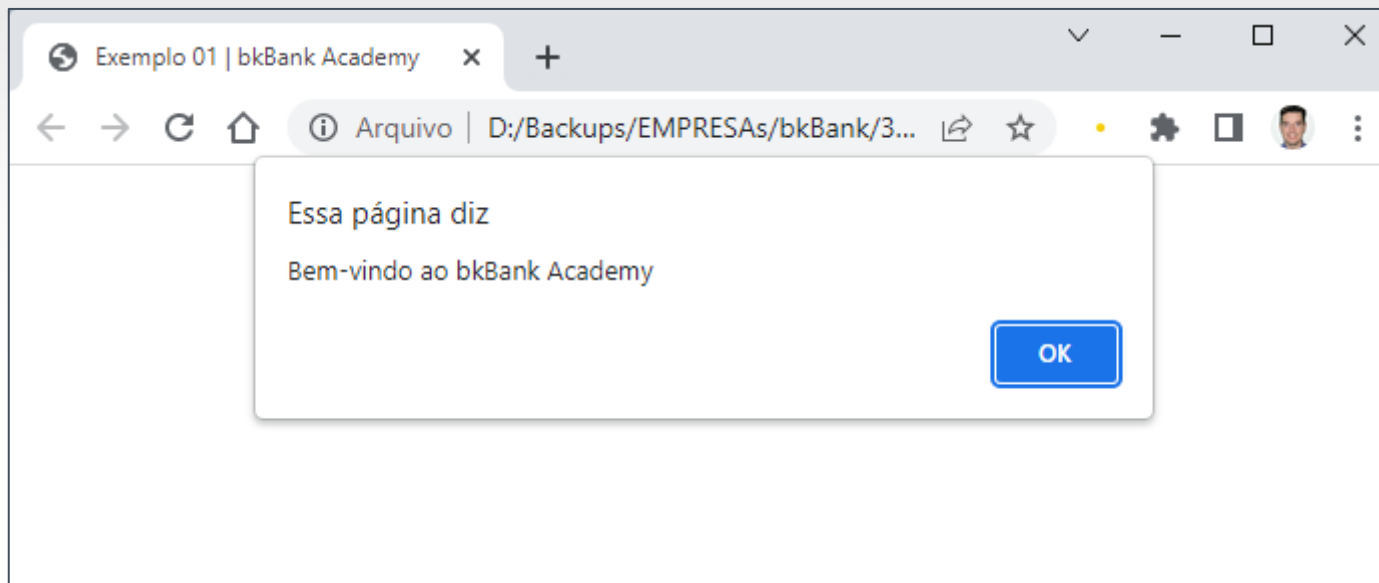


Figura 4 – Exemplo de saída de dados em uma caixa de alerta

JavaScript

- *Saída de dados com `alert()` e `console.log()`*
 - *Exemplo (01):*
 - *Saída gerada pelo comando `console.log()`, utilize **Ferramentas** do **Desenvolvedor** no **browser** e escolha o item **console***

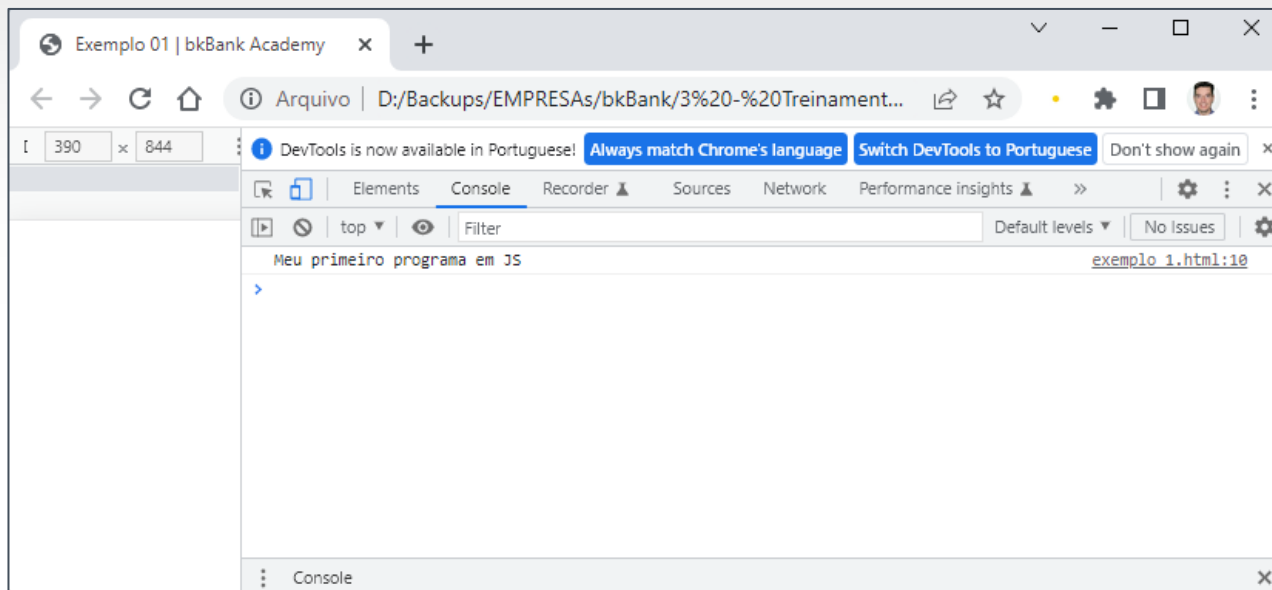


Figura 5 – Saída de dados gerada por um comando `console.log` (Google Chrome F12)

- ***Variáveis e Constantes***

- *Uma das principais funções de um programa é **interagir** com os **usuários***
- *Uma das formas de realizar essa **interação** é pela solicitação de informações e, a partir dessas informações, **implementar ações** e **apresentar respostas***
- *As **variáveis** são **espaços alocados na memória do computador** que **permitem guardar informações e trabalhar com elas***
- *Os **valores armazenados em uma variável** podem ser **alterados durante a execução do programa***

- ***Variáveis e Constantes***

- ***Exemplos de Variáveis:***

- A **descrição**, a **quantidade** e o **preço** de um **produto** ou, então, o **nome**, o **salário** e a **altura** de uma **pessoa**

- As **variáveis declaradas** devem **possuir um nome**, seguindo **algumas regras** de nomenclatura

- **Nomes de variáveis não podem:**

- **Conter espaços**
 - **Começar por número**
 - **Conter caracteres especiais, como +, -, *, /, (,), {, }, !, @, #**
 - **Utilizar nomes de palavras reservadas da linguagem, como **function**, **var**, **new**, **for** ou **return****

- ***Variáveis e Constantes***

- *Variáveis escritas com **letras maiúsculas** são **diferentes** de variáveis escritas com **letras minúsculas***
- *O uso do caractere "_" é válido, porém **dê preferência** para declarar variáveis com o nome em **letras minúsculas** e o uso de uma **letra maiúscula** para destacar palavras compostas (**camelcase**)*
- ***Identificadores Válidos:***
 - *cidade, nota1, primeiroCliente*
 - *novoSalario, precoFinal, dataVenda*

- **Variáveis e Constantes**

- Podemos declarar uma **variável** utilizando os comandos **var**, **let** ou **const**
- Nas últimas versões do JavaScript, passou-se a **recomendar** o uso de **const** ou **let**
- Uma **variável criada** a partir de um desses comandos possui um **escopo local** (**de bloco**), o que pode **evitar desperdício de memória** (ao **final** do **bloco**, a **variável deixa** de **existir**)
- **const** e **let** impedem que um **programa** seja **executado** com **alguns problemas no código** (**declarar duas variáveis com o mesmo nome no script**)

- ***Variáveis e Constantes***

- *A atribuição de valor para uma variável é feita com o sinal de "="*
- *É possível declarar uma variável e atribuir-lhe diretamente um valor com o uso da palavra reservada **const** (ou **let**) e do sinal de atribuição*

```
const idade = 18
```



- **Variáveis e Constantes**

- *Em algumas linguagens, o uso de **constantes** é restrito apenas à declaração de **variáveis** de **controle** ou de **configuração** do **sistema**, como o **número máximo** de **alunos** a serem manipulados pelo programa ou os **dados** de **conexão** com um **BD***
- *No JavaScript **moderno**, o uso de **const** passou a ter um novo significado e tem se tornado o padrão da linguagem*
- *Declarar uma **variável** com **const** indica que essa **variável** deve possuir uma **única atribuição** de valor e **não** será alterada no decorrer do programa, caso contrário, declare-a com **let***

- *Entrada de Dados com `prompt()`*

- *Exemplo (02):*

- *Comando (**método**) `prompt()` exibe uma caixa com texto e um espaço para digitação*

```
html > exemplo_2.html > html
1  <!DOCTYPE html>
2  <html Lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8      <script>
9          const nome = prompt("Qual é o seu nome?")
10         alert("Olá " + nome)
11     </script>
12
13     <title>Exemplo 02 | bkBank Academy</title>
14 </head>
15 <body>
16     <!-- Conteúdo do body -->
17 </body>
18 </html>
```

Figura 6 – Script do exemplo 02

- ***Entrada de Dados com `prompt()`***
 - ***Exemplo (02):***
 - ***Comando (`método`) `prompt()` exibe uma caixa com com texto e um espaço para digitação***

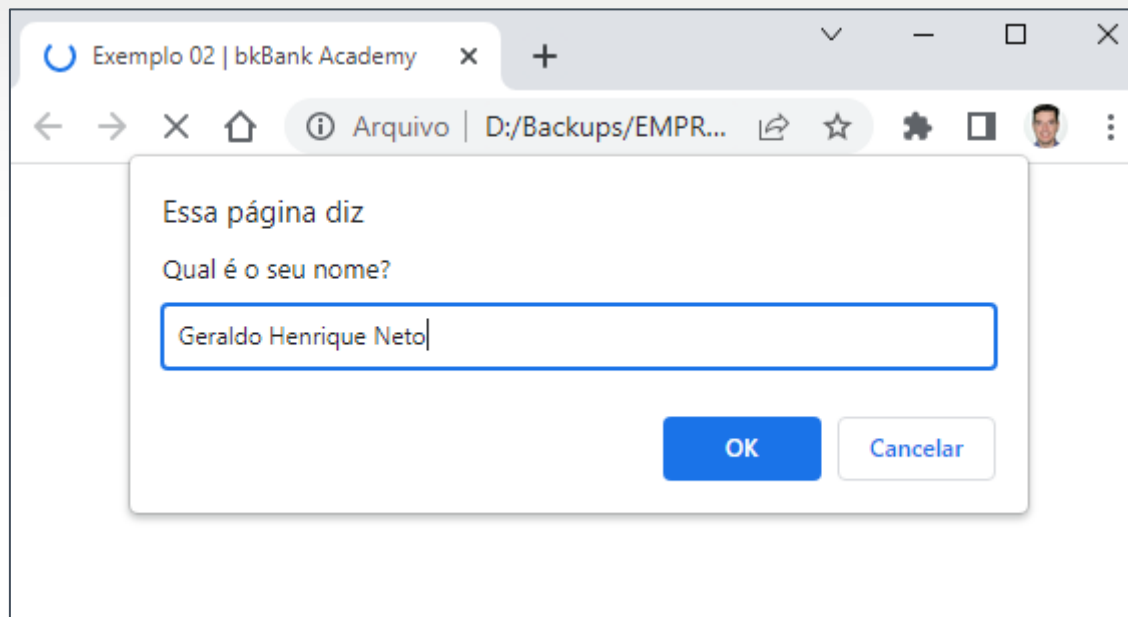


Figura 7 – Execução do método `prompt`

- ***Entrada de Dados com `prompt()`***
 - ***Exemplo (02):***
 - ***Comando (*método*) `prompt()` exibe uma caixa com texto e um espaço para digitação***

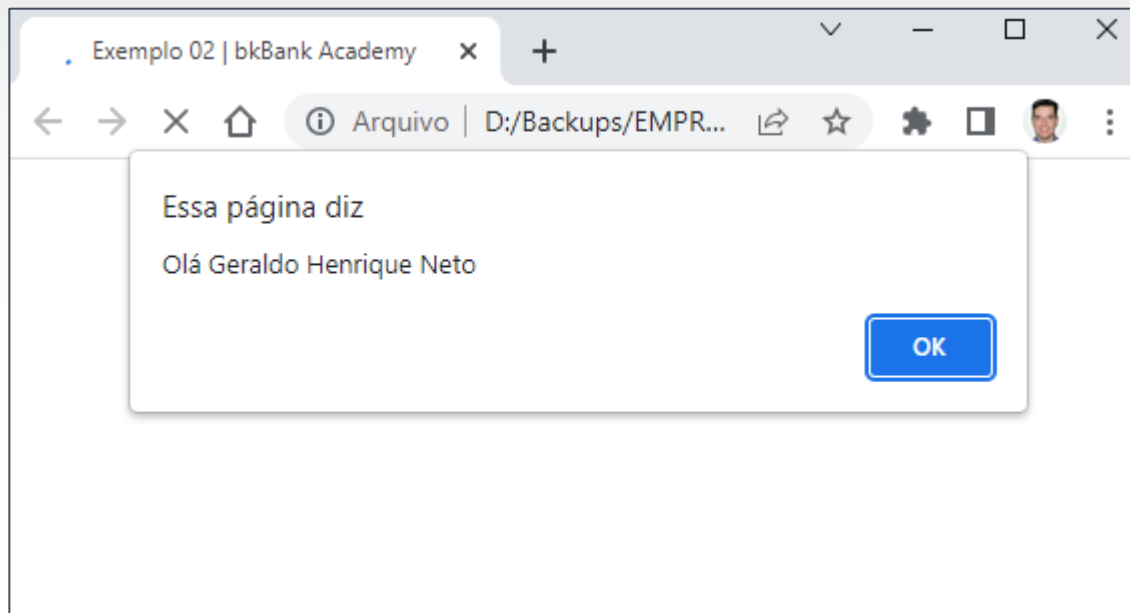


Figura 8 – Execução do alert com um texto concatenando com uma variável

- *Entrada de Dados com prompt()*
 - *Outra forma de exibir mensagens que contenham um **texto fixo** e o **conteúdo** de uma **variável** é com o uso do chamado **Template Strings***
 - *Deve-se delimitar a mensagem entre ``` (crases) e inserir o nome das **variáveis** usando a sintaxe: `${nomeVariavel}`*
- ```
alert(`Olá ${nome}`);
```
- *O uso de **ponto e vírgula** `;` no **final** dos **comandos** em é **opcional***

- **Comentários**

- *Um comentário é uma observação inserida pelo programador no código, com o objetivo de explicar algum detalhe do programa*
- *No cenário das grandes empresas, que desenvolvem projetos em equipes, a inserção de comentários nos códigos de um programa assume um papel ainda **mais relevante***
- *Os comentários **não** afetam a execução do programa*
- *Eles são ignorados pela linguagem, mas essenciais para o programador*

- **Comentários**

- *Explicar os detalhes de alguma fórmula utilizada em um sistema, como o cálculo do décimo terceiro salário em um sistema de folha de pagamento*
- *Salientar a função de algum comando **não** utilizado com frequência*
- *Alguma particularidade da linguagem ou sistema em desenvolvimento*
- *Também são importantes quando estamos programando um sistema maior, que será **concluído** em outro momento ou após alguma outra tarefa*

- **Comentários**

- ***Gastar alguns segundos digitando um breve comentário no sistema poderá economizar um bom tempo quando você precisar dar manutenção nesse sistema no futuro***
- ***Não poupe nas linhas de comentário nos sistemas que você desenvolverá***



- **Comentários**

- *Em JavaScript, os comentários podem ser inseridos para **uma** ou **várias linhas***

*// para comentários de uma linha*

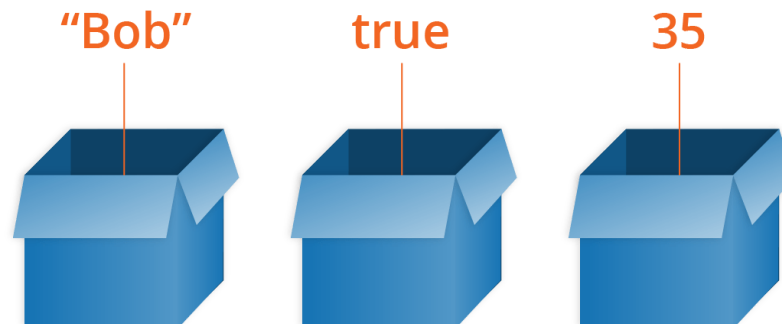
*/\**

*para comentários de duas  
ou mais linhas*

*\*/*



- ***Tipos de Dados e Conversões de Tipos***
  - *As variáveis manipuladas em um programa são de um determinado tipo*
  - *Principais Tipos: **strings** (texto), números e booleanos (**true** ou **false**)*
  - *Conhecer o tipo de uma variável permite identificar quais operações são possíveis para essa variável*



- *Tipos de Dados e Conversões de Tipos*
  - *Exemplo (03):*
    - *Operações envolvendo strings e números*

```
html > exemplo_3.html > html
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8 <script>
9 const a = "20"
10 const b = a * 2 // b = 40
11 const c = a / 2 // c = 10
12 const d = a - 2 // d = 18
13 const e = a + 2 // e = 202 ???
14 alert("e: " + e) // exibe o valor de uma variável
15 </script>
16
17 <title>Exemplo 03 | bkBank Academy</title>
18 </head>
19 <body>
20 <!-- Conteúdo do body -->
21 </body>
22 </html>
```

Figura 9 – Script do exemplo 03  
Aula 01 | Módulo Básico

- *Tipos de Dados e Conversões de Tipos*

- Para *solucionar* esse **problema**, é necessário *converter* o **texto** em **número**
- Essa *conversão* pode ser *realizada* pelos *métodos* **Number()**, **parseInt()** e **parseFloat()**
- *Modifique* a *linha* que *declara* a *variável* "**e**" para *conter* o *método* **Number()**

```
const e = Number(a) + 2 // e = 2
```

- *Tipos de Dados e Conversões de Tipos*
  - **Não** é obrigatório definir o tipo da variável na sua declaração
  - Ela assume um tipo no momento da atribuição de valor à variável
  - A atribuição de um conteúdo entre aspas (simples ou dupla, mas sempre aos pares) cria uma variável do tipo *String*
  - Para variáveis numéricas, **não** devem ser utilizadas as aspas
  - As variáveis booleanas podem conter os valores *true* ou *false*

- ***Tipos de Dados e Conversões de Tipos***
  - As **entradas** de **dados** realizadas com o método **prompt()** criam variáveis do tipo **String**, exceto se houver uma função de conversão de dados como **Number()**
  - Exibir uma variável que não recebeu uma atribuição de valor gera uma saída **"undefined"**

- ***Tipos de Dados e Conversões de Tipos***
  - ***Exemplo (04):***
    - ***Declara variáveis e exibe seus tipos a partir do comando typeof***

```
<script>
 const fruta = "Banana"
 const preco = 3.50
 const levar = true

 let novoValor

 console.log(typeof fruta) // String
 console.log(typeof preco) // number
 console.log(typeof levar) // boolean
 console.log(typeof novoValor) // undefined
</script>
```

Figura 10 – Script do exemplo 04

- *Tipos de Dados e Conversões de Tipos*
  - *É possível verificar se o número é inteiro ou possui decimais: `Number.isInteger()`*

```
console.log(Number.isInteger(12)) // true
console.log(Number.isInteger(3.50)) // false
```

- **Exemplos**

**A. Elaborar um programa que leia um número. Calcule e informe o dobro desse número.**

```
html > exemplo_5.html > html > body
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8 <script>
9 const num = prompt("Número: ") // lê um dado de entrada
10 const dobro = num * 2 // calcula o dobro
11 alert("Dobro é: " + dobro) // exibe a resposta
12 </script>
13
14 <title>Exemplo 05 | bkBank Academy</title>
15 </head>
16 <body>
17 <!-- Conteúdo do body -->
18 </body>
19 </html>
```

Figura 11 – Script do exemplo (A)



- **Exemplos**

***B. Elaborar um programa que leia dois números. Calcule e informe a soma desses números.***



```
html >  exemplo_6.html >  html
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8 <script>
9 const num1 = Number(prompt("1º Número: ")) // lê os números
10 const num2 = Number(prompt("2º Número: "))
11 const soma = num1 + num2 // calcula a soma
12 alert(`Soma é: ${soma}`) // exibe o resultado
13 </script>
14
15 <title>Exemplo 06 | bkBank Academy</title>
16 </head>
17 <body>
18 <!-- Conteúdo do body -->
19 </body>
20 </html>
```

Figura 12 – Script do exemplo (B)

- **Exemplos**

**C. Elaborar um programa que leia o valor de um jantar. Calcule e informe o valor da taxa de garçom (10%) e o valor total a ser pago.**

```
html > S exemplo_7.html > html
1 <!DOCTYPE html>
2 <html Lang="pt-br">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8 <script>
9 const jantar = Number(prompt("Valor do Jantar R$ ")) // lê o valor do jantar
10 const garcom = jantar * 0.10 // calcula variáveis de saída
11 const total = jantar + garcom
12 alert(`Taxa Garçom R$ ${garcom.toFixed(2)}\nTotal R$ ${total.toFixed(2)}`)
13 </script>
14
15 <title>Exemplo 07 | bkBank Academy</title>
16 </head>
17 <body>
18 <!-- Conteúdo do body -->
19 </body>
20 </html>
```

Figura 13 – Script do exemplo (C)

- **Exemplos**

***D. Elaborar um programa que leia a duração de uma viagem em dias e horas. Calcule e informe a duração total da viagem em número de horas.***

```
html > S exemplo_8.html > html > head
1 <!DOCTYPE html>
2 <html Lang="pt-br">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8 <script>
9 const dias = Number(prompt("Nº Dias: ")) // lê os dados de entrada
10 const horas = Number(prompt("Nº Horas: "))
11 const total = (dias * 24) + horas // calcula a duração
12 alert(`Total de Horas: ${total}`) // exibe o total
13 </script>
14
15 <title>Exemplo 08 | bkBank Academy</title>
16 </head>
17 <body>
18 <!-- Conteúdo do body -->
19 </body>
20 </html>
```

Figura 14 – Script do exemplo (D)  
Aula 01 | Módulo Básico

- **Observações**

- Para se tornar um **programador**, deve ser **persistente** e **cuidadoso** com os **detalhes da codificação**
- Sua **organização lógica do programa** pode estar correta, mas, se um **pequeno erro de grafia (sintaxe)** existir no código, ele **não é executado**
- **Lógica de programação** é algo que se **aprende com treinamento**
- Ao **criar um algoritmo**, vamos **“ensinar” o computador a realizar uma tarefa**

- **Observações**

- Com a **compreensão** e a **prática** dos **exercícios**, passamos a **assimilar** a forma como as **estruturas** de um **programa** devem ser **organizadas** (a **lógica** é utilizada para **montar corretamente** essas **estruturas**)
- **Uma boa regra a seguir para resolver os primeiros exercícios de programação é que todo o programa tem três etapas: entrada, processamento e saída**

- **Observações**

- *JavaScript é uma linguagem de destaque no cenário atual. É muito valorizada por empresas de desenvolvimento web pela capacidade de adicionar inúmeros recursos nas páginas (Ex.: interação com os usuários e à criação de layouts profissionais)*
- *Existem diversas opções de editores de código. Alguns estão disponíveis online e são úteis para realizarmos pequenos testes. Para criar programas maiores, utilize um editor profissional que contém diversos recursos visando auxiliar o trabalho do programador*

- **Observações**

- A linguagem JavaScript dispõe dos comandos (métodos) **prompt()** e **alert()** para realizar pequenas interações com os usuários. Eles permitem praticar as etapas de **entrada** e **saída** de dados de um algoritmo. É possível também utilizar o método **console.log()** para **exibir** uma **saída** de dados que é **apresentada** no **console**, nas **Ferramentas** do **Desenvolvedor**, dos **browsers web**
- **Variável** é um **conceito fundamental** para a **criação** de **programas**. São as **variáveis** que **permitem armazenar** os **dados de entrada**, **armazenar** um **cálculo** ou **outro processamento** e, posteriormente, **exibir** uma **saída** de **dados personalizada**

# Referências

Duckett, J.; Javascript e JQuery - Desenvolvimento de interfaces web interativas. Alta Books, 2018.

Flanagan, D.; JavaScript: The Definitive Guide, 7th Edition. O'Reilly Media, Inc. 2020.

Scott A. D., MacDonald M., Powers S.; JavaScript Cookbook, 3rd Edition. O'Reilly Media, Inc. 2021.

