course/logica-programacao-mergulhe-programacao-javascript

1.

09

**Desafio: hora da prática**

Praticar a lógica de programação, incluindo conceitos como variáveis, condicionais (if), alert, prompt, é essencial para seu desenvolvimento. Esses fundamentos fornecem a base para resolver problemas de forma estruturada, tomar decisões no código, criar interações controladas e interagir eficazmente com as pessoas.

Compreender esses conceitos não apenas facilita o aprendizado de novas linguagens e tecnologias, mas também capacita você a criar soluções inovadoras, depurar eficientemente e manter a qualidade ao longo do ciclo de vida do software. Portanto, investir tempo nesses princípios desde cedo é fundamental para construir uma trajetória bem-sucedida no campo da programação.

Pensando nisso, criamos uma lista de atividades (não obrigatórias) focada em prática para melhorar ainda mais sua experiência de aprendizagem. Bora praticar então?

**Desafios**

1. Mostre um alerta com a mensagem "Boas vindas ao nosso site!".
2. Declare uma variável chamada nome e atribua a ela o valor "Lua".
3. Crie uma variável chamada idade e atribua a ela o valor 25.
4. Defina uma variável numeroDeVendas e atribua a ela o valor 50.
5. Defina uma variável saldoDisponivel e atribua a ela o valor 1000.
6. Exiba um alerta com o texto "Erro! Preencha todos os campos"
7. Declare uma variável chamada mensagemDeErro e atribua a ela o valor "Erro! Preencha todos os campos" Agora exiba um alerta com o valor da variável mensagemDeErro.
8. Use um prompt para perguntar o nome do usuário e armazená-lo na variável nome.
9. Peça ao usuário para digitar sua idade usando um prompt e armazene-a na variável idade.
10. Agora, caso a idade seja maior ou igual que 18, exiba um alerta com a mensagem "Pode tirar a habilitação!".

**Opinião do instrutor**

Para te ajudar a verificar seus códigos, deixo disponibilizado de forma conjunta com a identificação dos 10 números iniciais quando foram propostos, uma possível [resolução dos 10 desafios neste link](https://github.com/alura-cursos/logica-js/tree/desafio_1) para você construir ou validar suas soluções.

**Boa sorte nos estudos!**

10

**Para saber mais: documentação**

A leitura da documentação oficial de uma linguagem de programação é crucial para que uma pessoa desenvolvedora de software possa utilizá-la de forma eficaz.

No caso do JavaScript, essa documentação fornece uma ampla gama de informações, incluindo conceitos sobre sintaxe, as bibliotecas e as funcionalidades disponíveis na linguagem.

Alguns links úteis para a documentação oficial do JavaScript incluem:

* [A documentação da linguagem de programação JavaScript](https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/What_is_JavaScript)
* [Guia de JavaScript: o que é e como aprender a linguagem mais popular do mundo?](https://www.alura.com.br/artigos/javascript)

Ao ler a documentação oficial, temos uma compreensão mais profunda da linguagem e de suas funcionalidades. Isso facilita a escrita de códigos mais limpos, claros e seguros. A documentação também pode ajudar o desenvolvedor ou desenvolvedora a economizar tempo, além de evitar a necessidade de pesquisar em fóruns ou outras fontes de informação não confiáveis.

2.

09

**Desafio: hora da prática**

Programação é prática! Criamos mais uma lista de atividades (não obrigatórias) para você exercitar e reforçar ainda mais seu aprendizado.Bora praticar?

**Desafios**

1. Pergunte ao usuário qual é o dia da semana. Se a resposta for "Sábado" ou "Domingo", mostre "Bom fim de semana!". Caso contrário, mostre "Boa semana!".
2. Verifique se um número digitado pelo usuário é positivo ou negativo. Mostre um alerta informando.
3. Crie um sistema de pontuação para um jogo. Se a pontuação for maior ou igual a 100, mostre "Parabéns, você venceu!". Caso contrário, mostre "Tente novamente para ganhar.".
4. Crie uma mensagem que informa o usuário sobre o saldo da conta, usando uma template string para incluir o valor do saldo.
5. Peça ao usuário para inserir seu nome usando prompt. Em seguida, mostre um alerta de boas-vindas usando esse nome.

**Opinião do instrutor**

Para te ajudar a verificar seus códigos, deixo disponibilizado uma possível [resolução dos desafios neste link](https://github.com/alura-cursos/logica-js/tree/desafio_2) para você construir ou validar suas soluções.

Boa sorte nos estudos!

10

**Para saber mais: ponto e vírgula no JavaScript**

No JavaScript, o uso do ponto e vírgula (;) é uma prática recomendada. A linguagem possui um mecanismo chamado "inserção automática de ponto e vírgula" (automatic semicolon insertion - ASI), que tenta adicionar ponto e vírgulas automaticamente em certos pontos do código onde eles são ausentes.

Isso significa que, em alguns casos, o JavaScript tentará "corrigir" a falta de ponto e vírgula inserindo-o automaticamente. No entanto, a interpretação do ASI pode levar a comportamentos inesperados e erros sutis, especialmente quando as regras não são claras.

Portanto, para evitar possíveis problemas e garantir a clareza do código, muitos desenvolvedores preferem adicionar explicitamente ponto e vírgula em seus programas.

Apesar da inserção automática de ponto e vírgula poder ajudar a mitigar erros de sintaxe, é uma boa prática adicionar ponto e vírgula manualmente para evitar ambiguidades e problemas de interpretação. Isso é particularmente importante em situações como quando várias instruções estão em uma mesma linha, ao usar declarações de retorno de valor ou ao minificar o código.

Em projetos colaborativos ou de grande escala, a consistência no estilo de codificação e a clareza do código são cruciais, e o uso explícito de ponto e vírgula contribui para um código mais legível e menos sujeito a erros de interpretação por parte dos programadores e do próprio mecanismo de ASI.

* [Leitura complementar da documentação sobre o JavaScript](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Lexical_grammar#automatic_semicolon_insertion).
* Quer saber o que a comunidade recomenda? [Leia mais neste link do Stack Overflow](https://pt.stackoverflow.com/questions/3341/utilizar-ou-n%C3%A3o-ponto-e-v%C3%ADrgula-no-fim-das-linhas-em-javascript).

3.

07

**Desafio: hora da prática**

Chegamos em mais uma lista de atividades (não obrigatórias) para você praticar e reforçar ainda mais seu aprendizado.Bora?!

**Desafios**

1. Crie um contador que comece em 1 e vá até 10 usando um loop while. Mostre cada número.
2. Crie um contador que começa em 10 e vá até 0 usando um loop while. Mostre cada número.
3. Crie um programa de contagem regressiva. Peça um número e conte deste número até 0, usando um loop while no console do navegador.
4. Crie um programa de contagem progressiva. Peça um número e conte de 0 até esse número, usando um loop while no console do navegador.

**Opinião do instrutor**

Para te ajudar a verificar seus códigos, deixo disponibilizado uma possível [resolução dos desafios neste link](https://github.com/alura-cursos/logica-js/tree/desafio_3) para você construir ou validar suas soluções.

Boa sorte nos estudos!

4.

04

**Math random**

**Guilherme:** Há um aspecto em nosso projeto que ainda nos incomoda: todas as vezes nós sabemos qual é o número secreto. Dessa forma, o jogo não tem graça.

Nosso objetivo é que, toda vez que formos jogar, haja um número novo, um número aleatório. Vamos implementar isso em nosso projeto?

**Mônica:** Vamos explorar as funcionalidades do JavaScript!

**Número aleatório com a função Math.random()**

**Guilherme:** Primeiramente, vamos abrir a [documentação do JavaScript](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript). Para isso, basta procurar por "JavaScript" no *Google* e acessar o primeiro link da *Mozilla*.

Nessa documentação, encontramos todas as informações sobre como o JavaScript funciona. Ela menciona que existe um padrão *ECMAScript*, ou seja, a convenção para saber o que colocar e o que não colocar no JavaScript, as funcionalidades dos navegadores, entre outras coisas.

Vale a pena realizar essa leitura posteriormente!

Neste momento, o que queremos é criar um número aleatório com o JavaScript. Então, vamos procurar por "número aleatório JavaScript Mozilla" no Google e selecionar o [link da documentação do JavaScript](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Math/random).

Realizada a busca no navegador, é exibida a informação de que a função chamada Math.random() retorna um número pseudo aleatório no intervalo entre 0 e 1.

Vamos acessar a documentação para tentar decifrar isso?

**Analisando a documentação**

Por padrão, utilizamos um tipo de número chamado de **inteiro**. Isso significa que é um número sem casas decimais, como 1, 2, 3, -1, -2, -3 e assim por diante.

A função Math.random() retorna um número pseudo aleatório, porque o computador por si só não consegue gerar um número completamente aleatório.

Ele tentará obter de algum lugar, de alguma forma, alguma coisa no próprio computador, seja o relógio, o delta, ou o tempo, multiplicará por algo e nos fornecerá um número.

Existem diversas funcionalidades. Existe algo chamado *seed* (semente), por exemplo, um número que colocamos no computador e que gerará possíveis números aleatórios.

Não é nada disso que queremos agora. Por isso, se diz que a função retorna um número pseudo aleatório e não um número aleatório.

**Mônica:**

Inclusive, existem outros conteúdos aqui na Alura. Há um artigo que estará no **Para saber mais** que explica minuciosamente como funciona a geração de números aleatórios.

**Guilherme:** Nosso desafio é ler a documentação e tentar entender como funciona a função Math.random(). No site, é mencionado que o número pseudo aleatório que será gerado está no intervalo de 0 e 1 ([0, 1[), incluindo o próprio 0, mas sem incluir o próprio 1.

Em outras palavras, isso significa que será gerado um número maior do que 0, ou seja, algo como 0.9354, e menor do que 1. O valor passado como segundo argumento para a função sempre será excluído.

Abaixo na documentação, temos alguns [exemplos](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Math/random#exemplos) de como gerar essa função.

Vamos testar isso no navegador?

**Mônica:** Vamos lá!

**Guilherme:** Vamos clicar com o botão direito na tela do navegador para abrir a aba "Inspecionar", como já aprendemos anteriormente. Em seguida, vamos clicar na aba "Console".

Para limpar o console, usaremos o atalho "Ctrl + L".

Vamos começar a digitar a função Math.random(). Ao fazer isso, note que uma sugestão para completar a função é apresentada. Duas coisas importantes nessa função: o "M" de “*Math*” precisa ser em letra maiúscula e o "*random*" em letra minúscula.

Sempre que quisermos utilizar essa geração de números pseudo aleatórios, vamos abrir e fechar parênteses ao final da função.

**Math**.**random**()

Dessa forma, ao teclar "Enter", será gerado um número aleatório, que no caso do instrutor foi 0.581 e assim por diante. Usando novamente a função, é gerado outro número: 0.929.

Existe um problema: nosso jogo não solicita que se digite um número menor do que zero.

**Mônica:** Imagine quão difícil seria acertar o número 0.2595555, nós precisaríamos de várias tentativas para chegarmos próximo do número.

**Guilherme:** Não conseguiríamos acertar esse número com tantas casas decimais. Então, qual é o problema? Podemos utilizar a função Math.random? Sim, mas não desta forma.

E se nós pegássemos esse número gerado, que é sempre menor do que 1, e mudássemos a posição da vírgula uma casa para a direita? Ou seja, se nós multiplicássemos esse número por 10, nós teríamos um número maior do que zero.

Então, vamos estar essa multiplicação por 10 no console.

**Math**.**random**() \* 10

**Mônica:** Ao pressionar "Enter", teremos números maiores que zero, um avanço, mas ainda há muitas casas decimais nos números, e nós precisamos apenas do número inteiro.

**Guilherme:** Já sabemos que vamos utilizar Math.random() \* 10, mas precisamos apenas da parte inteira do número. Há um comando em JavaScript que faz exatamente isso: ele pega somente a parte inteira do número, descartando os decimais após a vírgula. Esse comando é chamado de parseInt.

Mas temos um desafio. Precisamos pegar todo o conteúdo Math.random() \* 10 e utilizá-lo dentro do comando parseInt(). Então, vamos colocar esse trecho entre os parênteses do comando e pressionar "Enter".

parseInt(**Math**.**random**() \* 10)

**Mônica:**

Caso queira saber mais sobre parseInt(), recomendamos que você consulte a documentação. Saber ler a documentação é um recurso muito importante para quem trabalhará com desenvolvimento de software.

Às vezes, o exemplo que precisamos está tão longe do que usamos que parece que a documentação não nos ajuda, porém é importante conseguir pesquisar, testar e não ter medo de executar o código.

Isso não é exclusivo do JavaScript. Qualquer linguagem que você decidir utilizar nessa jornada também terá uma documentação bem completa disponível na internet. É apenas uma questão de aprender a pesquisar nela corretamente!

**Guilherme:** Toda vez que pedirmos para gerar um número com parseInt(Math.random() \* 10), o computador gerará um número onde o valor pelo qual estamos multiplicando será exclusivo. Ou seja, não será gerado o número 10.

No nosso projeto, precisamos de um número entre 1 e 10. Como parseInt(Math.random() \* 10) gera um número entre 0 e 9, nós podemos simplesmente adicionar 1 a esse resultado por meio de concatenação. Porém, essa explicação ficará para a próxima aula.

parseInt(**Math**.**random**() \* 10 + 1)

Dessa forma, será gerado um número entre 1 e 10 através de um comando JavaScript.

**Conclusão**

Nosso próximo desafio é adicionar esse comando ao nosso código, testá-lo e torcer para acertarmos o número aleatório. Conseguir um número entre 1 e 10 é fácil. Então, que tal aumentarmos o nível de dificuldade para entre 1 e 100? **Faremos isso na sequência!**

08

**Desafio: hora da prática**

E vamos exercitar nosso conhecimento com mais uma lista de atividades (não obrigatório) para você praticar e reforçar ainda mais seu aprendizado!

**Desafios finais**

1. Crie um programa que utilize o console.log para exibir uma mensagem de boas-vindas.
2. Crie uma variável chamada "nome" e atribua a ela o seu nome. Em seguida, utilize o console.log para exibir a mensagem "Olá, [seu nome]!" no console do navegador.
3. Crie uma variável chamada "nome" e atribua a ela o seu nome. Em seguida, utilize o alert para exibir a mensagem "Olá, [seu nome]!" .
4. Utilize o prompt e faça a seguinte pergunta: Qual a linguagem de programação que você mais gosta?. Em seguida, armazene a resposta em uma variável e mostre no console do navegador.
5. Crie uma variável chamada "valor1" e outra chamada "valor2", atribuindo a elas valores numéricos de sua escolha. Em seguida, realize a soma desses dois valores e armazene o resultado em uma terceira variável chamada "resultado". Utilize o console.log para mostrar a mensagem "A soma de [valor1] e [valor2] é igual a [resultado]." no console.
6. Crie uma variável chamada "valor1" e outra chamada "valor2", atribuindo a elas valores numéricos de sua escolha. Em seguida, realize a subtração desses dois valores e armazene o resultado em uma terceira variável chamada "resultado". Utilize o console.log para mostrar a mensagem "A diferença entre [valor1] e [valor2] é igual a [resultado]." no console.
7. Peça ao usuário para inserir sua idade com prompt. Com base na idade inserida, utilize um if para verificar se a pessoa é maior ou menor de idade, exibindo uma mensagem apropriada no console.
8. Crie uma variável "numero" e peça um valor com prompt verifique se é positivo, negativo ou zero. Use if-else para imprimir a respectiva mensagem.
9. Use um loop while para imprimir os números de 1 a 10 no console.
10. Crie uma variável "nota" e atribua um valor numérico a ela. Use if-else para determinar se a nota é maior ou igual a 7 e exiba "Aprovado" ou "Reprovado" no console.
11. Use o Math.random para gerar qualquer número aleatório e exiba esse número no console.
12. Use o Math.random para gerar um número inteiro entre 1 e 10 e exiba esse número no console.
13. Use o Math.random para gerar um número inteiro entre 1 e 1000 e exiba esse número no console.

**Opinião do instrutor**

Para te ajudar a verificar seus códigos, deixo disponibilizado uma possível [resolução dos desafios neste link](https://github.com/alura-cursos/logica-js/tree/desafios) para você construir ou validar suas soluções.

Boa sorte nos estudos!

09

**Para saber mais: preciso decorar cada linha de código ou comando?**

Compreender cada linha de código ou comando em detalhes é certamente uma aspiração louvável, mas não é necessário decorar. O desenvolvimento de software moderno é uma tarefa complexa, e as linguagens de programação oferecem uma ampla gama de recursos e bibliotecas.

Em vez de memorizar cada linha, é mais valioso entender os conceitos fundamentais por trás das estruturas de programação e saber como usar a documentação efetivamente.

A documentação de uma linguagem de programação é uma ferramenta essencial para todos os desenvolvedores. Ela não apenas fornece uma referência rápida para a sintaxe e os comandos, mas também explica os conceitos subjacentes, oferece exemplos práticos e ajuda a compreender os diferentes recursos disponíveis.

Através da documentação você pode aprender a utilizar bibliotecas, explorar casos de uso avançados e entender as melhores práticas recomendadas pela comunidade. Isso economiza tempo, evita erros e permite que você se mantenha atualizado com as últimas atualizações da linguagem.

Em vez de se preocupar em memorizar cada detalhe, concentre-se em desenvolver habilidades de resolução de problemas, compreender os princípios de design de software e aprender a pesquisar eficientemente na documentação. A capacidade de ler e interpretar a documentação é uma habilidade valiosa, pois permite que você aprenda novas linguagens e tecnologias de maneira eficaz, adaptando-se rapidamente às mudanças do cenário de desenvolvimento. Portanto, em sua jornada como pessoa desenvolvedora, lembre-se de que a habilidade de compreender e usar a documentação é tão importante quanto saber escrever código.

Nesta aula, [vimos como usar a documentação para gerar um número aleatório através da documentação do Mozilla](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Math/random). Porém, existe também o [W3Schools (W3S)](https://www.w3schools.com/js/default.asp) que é um recurso online amplamente conhecido e utilizado para aprender diversas tecnologias de desenvolvimento web, incluindo JavaScript (JS).

O site oferece tutoriais interativos, exemplos de código, referências de sintaxe e conceitos fundamentais relacionados ao JavaScript e outras linguagens web.

Ao explorar o W3Schools, você pode adquirir uma compreensão sólida dos princípios do JS, desde o básico até tópicos mais avançados, como manipulação do DOM, interações do usuário e requisições assíncronas.

O W3Schools é uma ferramenta valiosa para aprimorar suas habilidades em JavaScript, oferecendo um ambiente prático para experimentar o código, compreender os conceitos-chave e aplicar o conhecimento adquirido em seus próprios projetos de desenvolvimento web.

5.

05

**Referências**

**1.**[**Lógica de Programação Crie seus primeiros programas usando Javascript e HTML**](https://www.casadocodigo.com.br/products/livro-programacao?_pos=1&_sid=4661f8240&_ss=r)

Este livro apresenta uma abordagem totalmente prática. Uma didática pensada no iniciante, com a qual os conceitos são apresentados com motivações práticas, através do surgimento da necessidade para depois mostrar a solução.

**2.**[**Lógica de programação com Portugol**](https://www.casadocodigo.com.br/products/livro-portugol?_pos=2&_sid=4661f8240&_ss=r)

Neste livro, Joice Mendes e Rafael Muniz apresentam todos os conceitos necessários para a criação da lógica de programação e dos algoritmos. Você vai aprimorar sua percepção lógica e aprender a aplicá-la na programação, cobrindo tópicos desde a sintaxe do Portugol, variáveis, comandos, estruturas condicionais, operadores relacionais e lógicos, estruturas de repetição, até vetores, matrizes e funções. O material é recheado com 85 exemplos de código, 55 exercícios de fixação com gabarito e um projeto prático ao longo do aprendizado. Todos os capítulos contam com um vídeo complementar disponibilizado na internet.

**3.**[**Livro: "Estruturas de Dados e Algoritmos com JavaScript"**](https://www.google.com.br/books/edition/Estruturas_de_dados_e_algoritmos_com_Jav/0nWKDwAAQBAJ?hl=pt-BR&gbpv=1&dq=estrutura+de+dados+javascript&printsec=frontcover)

Este livro aborda de forma detalhada as estruturas de dados e algoritmos mais comuns, fornecendo exemplos práticos em JavaScript.

**4.**[**Site: MDN Web Docs**](https://developer.mozilla.org/pt-BR/)

A documentação oficial da Mozilla Developer Network (MDN) é uma excelente fonte de informações sobre JavaScript. Lá você encontrará explicações detalhadas sobre a sintaxe, recursos da linguagem e exemplos de código.

**5. [Eloquent JavaScript 3rd edition (2018)](https://eloquentjavascript.net/" \t "_blank)**

Este é um livro sobre JavaScript, programação e as maravilhas do mundo digital. Um guia essencial para toda a pessoa desenvolvedora web. Em inglês.

**6.**[**Algoritmos - Teoria e Prática, Thomas H. Cormen**](https://books.google.com.br/books/about/Algoritmos_Teoria_e_Pr%C3%A1tica.html?id=6iA4LgEACAAJ&source=kp_book_description&redir_esc=y)

Este livro apresenta um texto abrangente sobre o moderno estudo de algoritmos para computadores. É uma obra clássica, cuja primeira edição tornou-se amplamente adotada nas melhores universidades em todo o mundo, bem como padrão de referência para profissionais da área.

**7. [JavaScript: O Guia Definitivo](https://www.amazon.com.br/JavaScript-Guia-Definitivo-David-Flanagan/dp/856583719X/ref=sr_1_1?keywords=javascript&qid=1701835643&sr=8-1&ufe=app_do%3Aamzn1.fos.6121c6c4-c969-43ae-92f7-cc248fc6181d" \t "_blank)**

Referência completa para programadores, JavaScript: O guia definitivo fornece uma ampla descrição da linguagem JavaScript básica e das APIs JavaScript do lado do cliente definidas pelos navegadores Web. Recomendado para programadores experientes que desejam aprender a linguagem de programação da Web e para programadores JavaScript que desejam ampliar seus conhecimentos e dominar a linguagem, este é o guia do programador e manual de referência de JavaScript completo e definitivo.

**8.**[**HTML5 e CSS3 Domine a web do futuro**](https://www.casadocodigo.com.br/products/livro-html-css?_pos=2&_sid=ee24eb627&_ss=r)

Neste livro você irá aprender a criar páginas elegantes de forma simples! HTML e CSS, quando bem utilizados, podem ser o sucesso de um projeto e, com os novos recursos, muito do que antes era trabalhoso agora não é mais. Aprenda as melhores técnicas para escrever seu site por meio de exemplos práticos de funcionalidades úteis do cotidiano. Construa menus, aplique efeitos, estilize elementos visuais, melhore a semântica da sua página e muito mais!

**9.**[**Guia Front-End: O caminho das pedras para ser um dev Front-End**](https://www.casadocodigo.com.br/products/livro-guia-frontend?_pos=5&_sid=ee24eb627&_ss=r)

Neste livro, Diego Eis nos guia sobre o mundo de desenvolvimento web por meio de uma análise franca e objetiva de diversas tecnologias adotadas, necessidades do mercado e postura profissional. Você não vai aprender diretamente sobre essas tecnologias aqui, mas certamente vai desenvolver um senso mais apurado e uma nova forma de olhar para elas, o que é fundamental nesse mundo de aprendizado não linear.

**10.**[**Como utilizar operadores de comparação em Javascript**](https://www.alura.com.br/artigos/operadores-matematicos-em-javascript?_gl=1*1dp3zv3*_ga*NzU2NTAyMDMyLjE2ODcxOTg5NTE.*_ga_1EPWSW3PCS*MTcwMTgzNTcwNC4zMTIuMS4xNzAxODM3NzE1LjAuMC4w*_fplc*RWNyYmRncFY3MFF2ajR3U3dGQ1hWUEFsUDVSS1VnRklHYlpIbXMwNzlaaXk4RGdGYVVwSVR1RlhROHQydXJqSU5jMW12U3MwYjdOamJZb2NuNGdZRFBqJTJGOVZycDFmYUt5UyUyRloyMkJ2aWFKUTc5YVppNyUyQjJDV0lNb0NQV3J3JTNEJTNE)

Neste artigo de Rafa Ballerini você aprenderá as diferenças entre operadores de comparação em JavaScript e como utilizá-los

**11.**[**Documentação MDN: O que é JavaScript?**](https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/What_is_JavaScript)

Neste primeiro artigo há uma análise profunda da linguagem, respondendo questões como "O que é JavaScript?", e "O que ele faz?", para você se sentir confortável com a proposta da linguagem.

**12. [JavaScript Tutorial - Documentação W3Schools](https://www.w3schools.com/js/default.asp" \t "_blank)**

Este tutorial ensina JavaScript do básico ao avançado. Em Inglês

**13.**[**Guia de JavaScript: o que é e como aprender a linguagem mais popular do mundo?**](https://www.alura.com.br/artigos/javascript?_gl=1*7i282i*_ga*NzU2NTAyMDMyLjE2ODcxOTg5NTE.*_ga_1EPWSW3PCS*MTcwMTgzNTcwNC4zMTIuMS4xNzAxODM3ODE5LjAuMC4w*_fplc*RWNyYmRncFY3MFF2ajR3U3dGQ1hWUEFsUDVSS1VnRklHYlpIbXMwNzlaaXk4RGdGYVVwSVR1RlhROHQydXJqSU5jMW12U3MwYjdOamJZb2NuNGdZRFBqJTJGOVZycDFmYUt5UyUyRloyMkJ2aWFKUTc5YVppNyUyQjJDV0lNb0NQV3J3JTNEJTNE)

Neste artigo, você vai conhecer o que é JavaScript, para que serve e como utilizá-lo.