

LOM3258

Introdução à Eletrônica e Computação Física

Introdução

Programação em JavaScript

Programação

Programar é fazer com que os computadores façam o que você quer que eles façam. Um programa de computador é basicamente uma série de instruções que informam ao computador como realizar uma tarefa. Infelizmente, os computadores não falam a mesma língua que nós.

Por exemplo, você não pode simplesmente escrever “mudar a cor do círculo para azul” e esperar que o computador entenda. Uma linguagem de programação atua como um intermediário: é uma linguagem que pode ser entendida tanto por computadores quanto por humanos.

Programação

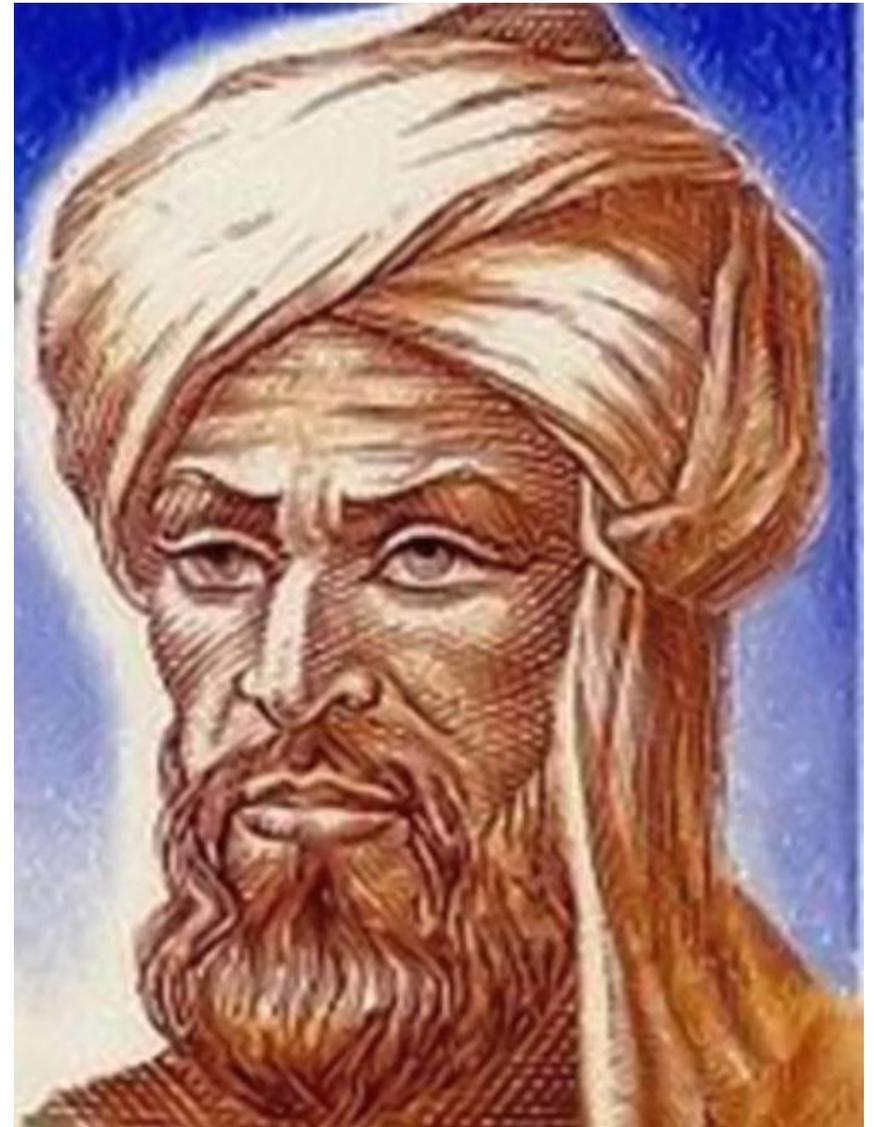
Aprender a programar é um pouco como aprender uma língua estrangeira, exceto que os computadores podem ser viciados em gramática (ainda mais do que meu professor de francês!). Você precisa ter certeza de que colocou tudo no lugar certo e a sintaxe precisa estar correta. Os computadores são poderosos e você pode fazer com que eles façam coisas realmente impressionantes, mas eles também cairão em pedaços se apenas um suporte estiver fora do lugar!

Escrever um programa é basicamente escrever um conjunto de instruções a serem seguidas por um computador. O problema é que eles têm que seguir instruções precisas. Qualquer pequena ambiguidade e o computador fará algo completamente diferente do que você tinha em mente - ou pode até travar.

Algoritmo

A palavra algoritmo é usada com bastante frequência atualmente. Você já deve ter ouvido falar do “algoritmo do Instagram” ou do “algoritmo de pesquisa do Google”.

Mas o que exatamente é um algoritmo? A palavra “algoritmo” é uma latinização do nome de um matemático persa, **Al-Khwarizmi**, que escreveu o primeiro livro algébrico e gostava de explicar seus métodos usando uma abordagem passo a passo.



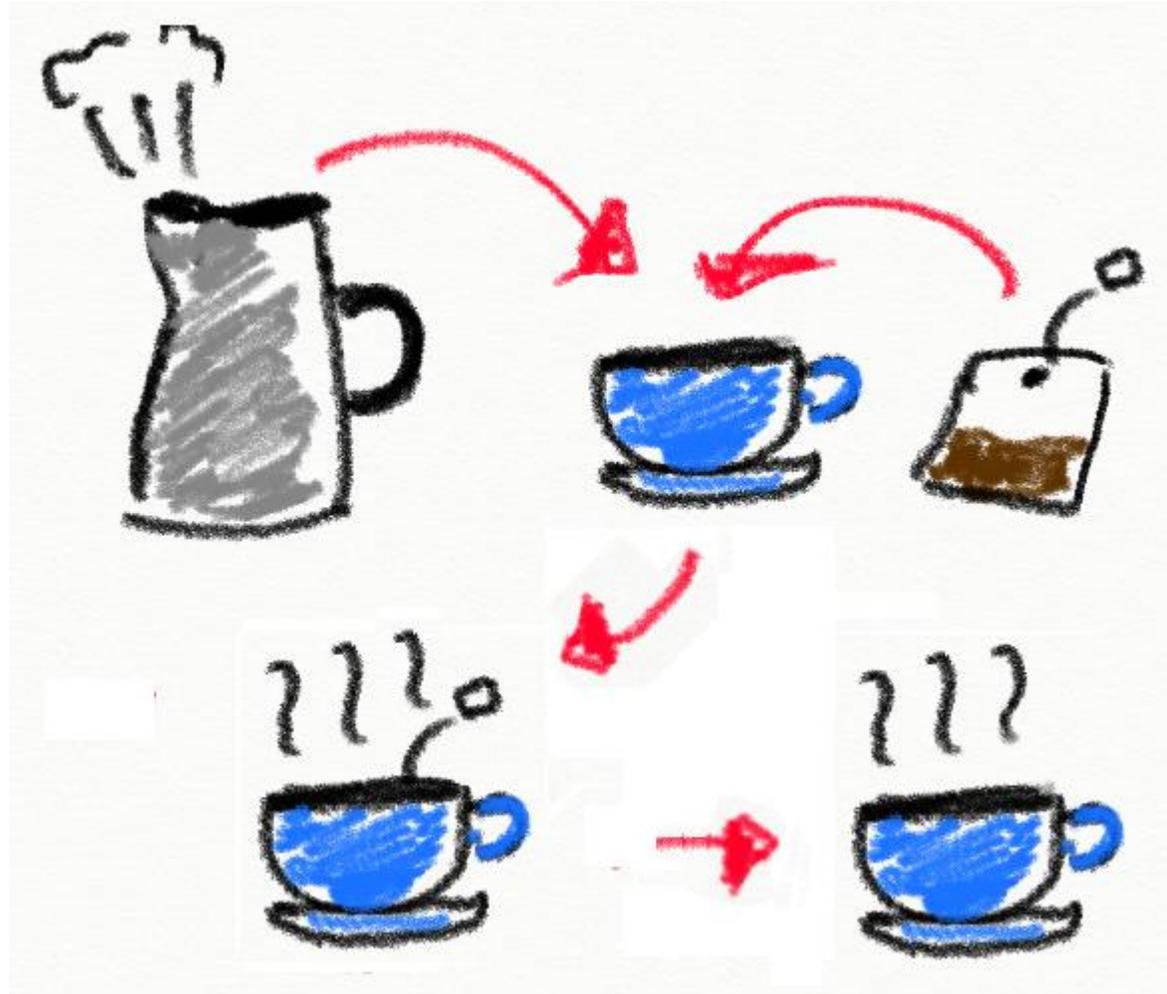
Algoritmo

Isso fez com que a palavra algoritmo fosse usada para descrever qualquer método passo a passo. Deve ficar claro o que fazer em cada etapa e o que cada etapa faz.

Por exemplo, aqui está um algoritmo para fazer uma xícara de chá:

1. Coloque água em uma chaleira no fogo
2. Pegue uma xícara
3. Pegue um saquinho de chá
4. Coloque o saquinho de chá na xícara
5. Quando a chaleira ferver, despeje a água na xícara
6. Deixe fermentar por dois minutos
7. Retire o saquinho de chá da xícara

Algoritmo visual



Pseudocódigo

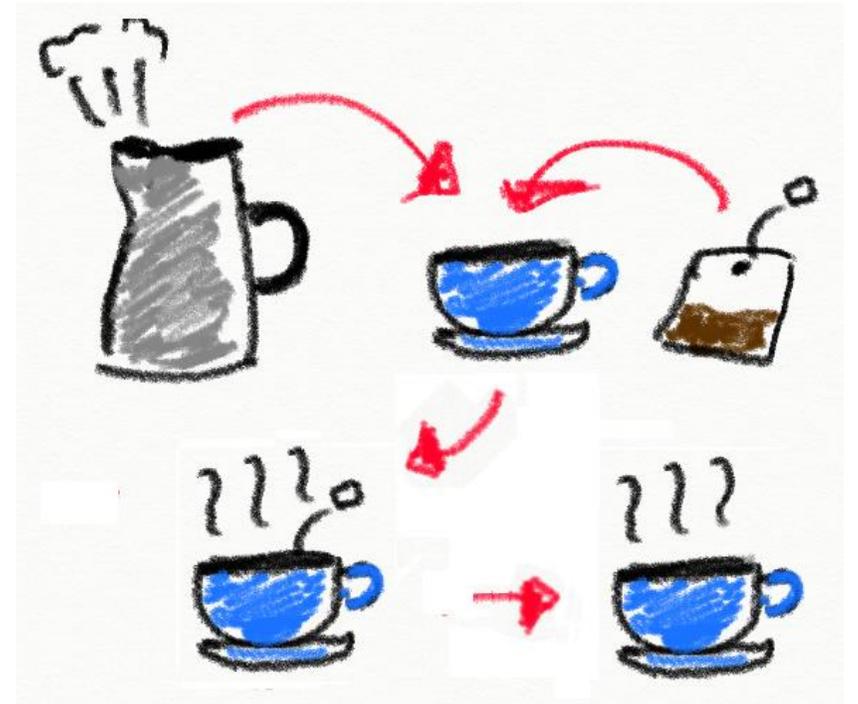
Pseudocódigo é um código “**aparente**” que pode ser escrito para ilustrar o que um programa faz sem ter que se preocupar com os detalhes e a ortografia de uma linguagem de programação real. Ele segue as convenções e a estrutura de um programa e usa comandos precisos, mas sem as especificações de uma linguagem. Por exemplo, a instrução “**exibir uma mensagem na tela**” pode ser escrita em pseudocódigo como **exibir (mensagem)**, enquanto na linguagem de programação JavaScript seria escrita como **print (message)**. Observe que o código JavaScript usa a notação específica de impressão, que é o comando usado para exibir mensagens na tela. Outras linguagens usam comandos diferentes para fazer basicamente a mesma coisa.

O pseudocódigo em língua portuguesa do exemplo do chá poderia ser escrito como:

```
ferver(água,250ml) na chaleira;  
obter(xícara) do armário;  
obter(saquinho_chá) da caixa_chá;  
colocar(saquinho_chá) na xícara;  
quando(chaleira_ferver)  
    { despejar(água, 250ml) na xícara };  
aguardar(2 min);  
remover(saquinho_chá) da xícara;
```

Pseudocódigo em língua inglesa do exemplo do chá

```
boil(water, 250ml) in kettle;  
get(cup) from cupboard;  
get(teabag) from box;  
put(teabag) in cup;  
when(kettle boils)  
    { pour(water, 250ml) in cup };  
wait(2 min);  
remove(teabag) from cup;
```



Pseudocódigo

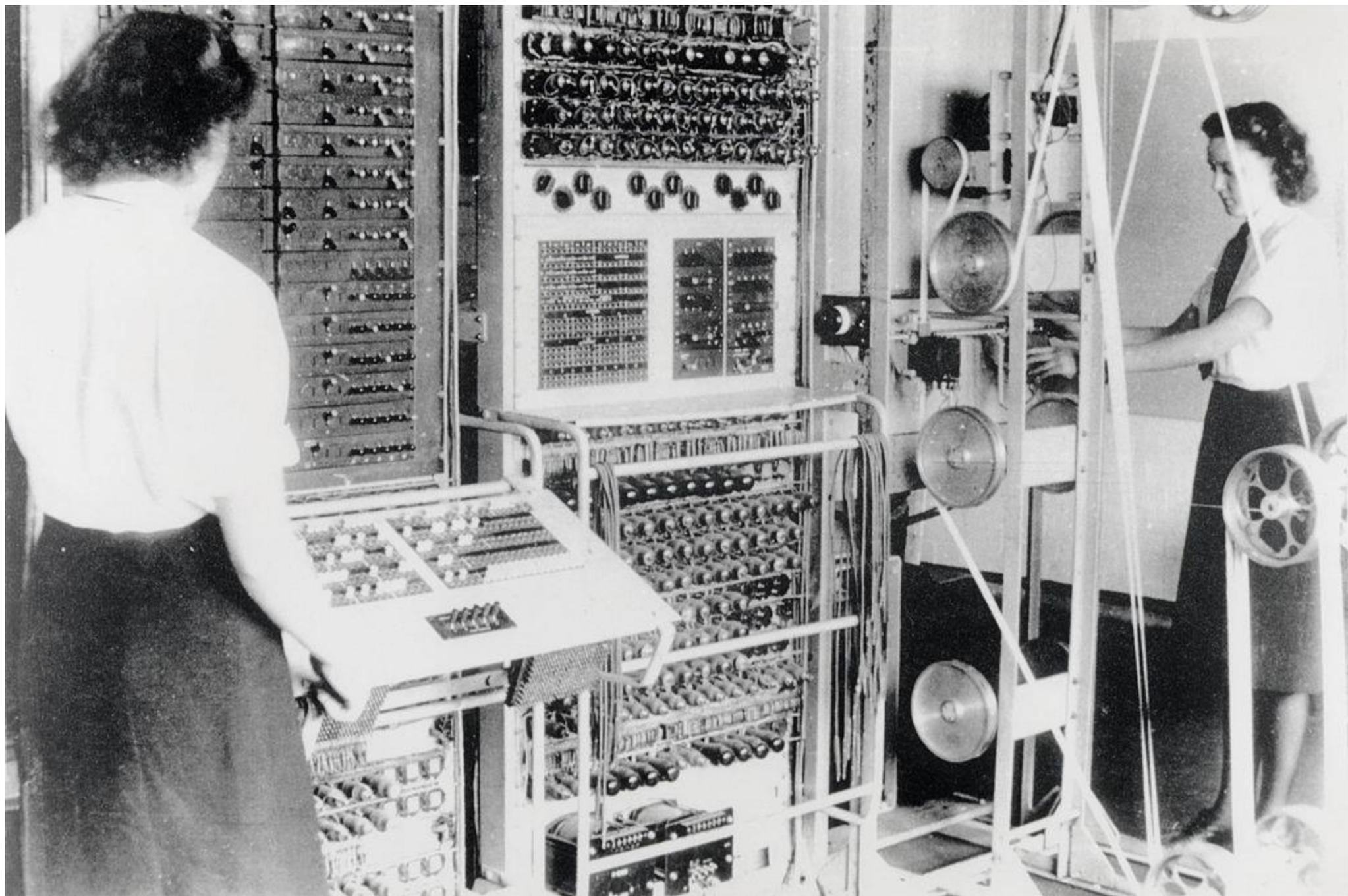
Um programador que sabe como programar em uma linguagem específica deve ser capaz de seguir um exemplo de pseudocódigo e, em seguida, escrevê-lo em sua linguagem de escolha.

O pseudocódigo é útil para:

- planejar um programa antes de começar a codificar e também
- facilitar o compartilhamento de ideias entre programadores que usam diferentes linguagens de programação.

Uma breve história da programação

Os primeiros computadores foram programados com cartões perfurados para representar o sistema numérico binário, o sistema numérico que os computadores usam. O sistema binário é composto inteiramente de 1 e 0 que se traduzem vagamente em "ligado" e "desligado". Nos cartões, um furo representava 1 e a ausência de furo 0. Depois disso, as pessoas começaram a desenvolver linguagens que poderiam ser usadas para “falar” com o computador.



Uma breve história da programação

Os primeiros programas de computador foram escritos em código de máquina e linguagem Assembly. Essas são linguagens de programação de baixo nível que estão intimamente associadas ao hardware de um computador. Isso significa que podem ser linguagens difíceis de programar porque envolvem a escrita de código abstrato fortemente vinculado à arquitetura de um computador. Na verdade, é raro encontrar alguém que programe em código de máquina ou Assembly hoje em dia, mas aqueles que o fazem trabalharão em estreita colaboração com o hardware de um computador - por exemplo, escrevendo drivers de dispositivo ou trabalhando em sistemas embarcados.

Uma breve história da programação

Linguagens de programação de alto nível usam abstrações que tornam o código mais fácil para humanos ler e escrever. Os programas são escritos em uma linguagem como Swift, C # ou Java, que é então compilado em código de máquina e executado. Os programas escritos usando essas linguagens são muito rápidos, tornando as linguagens de alto nível adequadas para escrever jogos e software de negócios profissional onde a velocidade é importante.

Uma breve história da programação

Linguagens de script são outro tipo de linguagem de alto nível, mas são interpretadas, em vez de compiladas, o que significa que são traduzidas em código de máquina quando o programa é executado, e não de antemão. Isso significa que eles costumam ser mais lentos do que as linguagens compiladas, embora os intérpretes estejam se tornando mais sofisticados, tornando algumas linguagens interpretadas quase tão rápidas quanto as linguagens compiladas. Algumas linguagens de script comuns que você pode ouvir falar são Python, Ruby e, é claro, JavaScript.

JavaScript

A linguagem que aprenderemos neste livro é JavaScript, muitas vezes referida como a linguagem da Web. Praticamente todos os navegadores da web podem executar JavaScript, o que o torna uma das linguagens de programação mais populares do mundo.

JavaScript é uma ótima linguagem para usar ao aprender a programar. Tem uma barreira baixa de entrada: tudo o que você precisa para programar em JavaScript é um navegador da web. É fácil começar e o básico é fácil de aprender. É também uma linguagem flexível e expressiva que pode criar uma variedade de programas poderosos.

JavaScript

JavaScript é uma linguagem de script interpretada e compilada em tempo de execução. Isso significa que é necessário um mecanismo para interpretar e executar um programa. Isso geralmente é feito por um navegador da web, mas existem mecanismos JavaScript que podem executar programas sem um navegador. JavaScript também é uma linguagem dinâmica, o que significa que os elementos de um programa podem mudar durante a execução, ao contrário de uma linguagem compilada como C++.

A História do JavaScript

A World Wide Web começou como um monte de páginas vinculadas por **hiperlinks**. Os usuários logo queriam mais interação com essas páginas, então a **Netscape** (um dos primeiros fornecedores de navegadores) pediu a um de seus funcionários, Brendan Eich, para desenvolver uma nova linguagem para seu navegador **Navigator**. Ele criou um protótipo de linguagem de script chamada **Mocha** em apenas dez dias. A nova linguagem foi renomeada para **LiveScript**, mas foi rapidamente rebatizada como **JavaScript** para que pudesse se beneficiar da publicidade que a linguagem Java da Sun Microsystems estava atraindo na época.

A História do JavaScript

O nome “**JavaScript**” costuma causar alguma confusão infeliz, com o JavaScript muitas vezes considerado “Java lite”, embora as duas linguagens sejam completamente independentes.

JavaScript fez sua estreia em 1995 e marcou o início de uma nova era empolgante de ser capaz de programar um navegador da web para fazer coisas. (Infelizmente, no início, o uso mais comum era criar anúncios pop-up e logotipos giratórios!)

A História do JavaScript

Em 2005, sites como o Google Maps e o Gmail começaram a aparecer e demonstraram que o JavaScript era capaz de criar aplicativos ricos para a Internet que pareciam e se comportavam como aplicativos nativos de desktop. Esse progresso continuou em ritmo acelerado e quase todos os sites hoje em dia usarão JavaScript de alguma forma. JavaScript também encontrou seu caminho em uma variedade de outras coisas, como aplicativos de smartphones, vestíveis (*wearables*) e bancos de dados.

Google Maps



Manchester

Cloudy · 1°C
3:19 PM



Directions



Save



Nearby



Send to your
phone



Share



Versões de JavaScript

Em 1996, foi tomada a decisão de padronizar o **JavaScript** com a ajuda da European Computer Manufacturers Association, que hospedaria o padrão. Essa linguagem padronizada foi denominada **ECMA Script** para evitar infringir a marca “Java”. Isso causou ainda mais confusão com base no nome, mas eventualmente o ECMA Script foi usado para se referir à especificação e o JavaScript foi usado para se referir à própria linguagem. Este ainda é o caso, embora haja um debate constante sobre a mudança do nome. Uma sugestão é renomear oficialmente o **JavaScript** apenas como “**JS**” - que geralmente é chamado de qualquer maneira.

Versões de JavaScript

Quando o **JavaScript** foi padronizado em 1997, a especificação era conhecida como **ECMAScript** versão 1. Em 2015, decidiu-se publicar uma nova especificação a cada ano, com a versão nomeada a partir do ano em que foi publicada. Como resultado dessa mudança, a versão 6 do **ECMAScript** foi renomeada para **ES2015** quando foi publicada e, desde então, tem havido uma nova versão do JavaScript a cada ano.



Compatibilidade com versões anteriores

Um conceito importante no desenvolvimento da linguagem JavaScript é o da compatibilidade com versões anteriores. Isso significa que todo código antigo deve funcionar da mesma maneira quando interpretado por um mecanismo executando uma nova especificação.

É um pouco como dizer que um PlayStation 5 ainda deve ser capaz de rodar qualquer jogo criado para todos os PlayStations anteriores. Isso evita que o JavaScript “invada a Web”, fazendo mudanças drásticas que significariam que o código antigo e legado usado em alguns sites pode não funcionar como esperado em navegadores modernos.

Linguagens de programação mais populares 2021

Front-end web development	Back-end web development	Mobile development
 JavaScript	 JavaScript	 Swift
 Elm	 Scala	 Java
 TypeScript	 Python	 Objective C
	 Go	 JavaScript
	 Ruby	
Game development	Desktop applications	Systems programming
 Unity	 Scala	 Go
 TypeScript	 Go	 Rust
	 Python	

Linguagens de programação mais populares 2021

Language	Development					
	Front-end web	Back-end web	Mobile	Game	Desktop applications	Systems programming
 JavaScript	●	●	●			
 Elm	●					
 TypeScript	●			●		
 Scala		●			●	
 Python		●			●	
 Go		●			●	●
 Ruby		●				
 Swift			●			
 Java			●			
 Objective C			●			
 Unity				●		
 Rust						●

Linguagens de programação mais populares

- Java
- C
- C++
- C#
- Python
- Go
- Swift
- Matlab
- JavaScript
- C#
- Python
- Go
- Swift

Olá Mundo! Seu primeiro programa JavaScript

É uma tradição, ao aprender linguagens de programação, começar com um programa Hello, World! Este é um programa simples que produz a frase “Hello, World!” para anunciar sua chegada ao mundo da programação.

Vamos manter essa tradição e escrever um Olá, mundo! programa em JavaScript. Será uma única declaração que registra a frase “Hello, World!” para o console.

As 9 melhores linguagens de programação para aprender em 2021

Se você é novo no campo de desenvolvimento de software, a parte mais difícil de aprender programação é decidir por onde começar. Existem centenas de linguagens de programação amplamente utilizadas, cada uma com suas próprias complexidades e idiossincrasias.

A boa notícia é que, ao começar sua jornada como desenvolvedor de software, você começará a descobrir qual linguagem de programação será mais adequada para você, seus interesses e seus objetivos de carreira.

<https://www.fullstackacademy.com/blog/nine-best-programming-languages-to-learn>

As 9 melhores linguagens de programação para aprender em 2021

1. JavaScript
2. Swift
3. Scala
4. Go
5. Python
6. Elm
7. Ruby
8. C#
9. Rust

<https://www.fullstackacademy.com/blog/nine-best-programming-languages-to-learn>

1. JavaScript

É impossível ser um desenvolvedor de software hoje em dia sem usar JavaScript de alguma forma. De acordo com a Pesquisa de Desenvolvedor 2020 do Stack Overflow, JavaScript é a linguagem mais popular entre os desenvolvedores pelo oitavo ano consecutivo. Quase 70% dos respondentes da pesquisa relataram que haviam usado JavaScript no ano passado.

Junto com HTML e CSS, JavaScript é essencial para o desenvolvimento front-end da web. A maioria dos sites mais populares da web, do Facebook e Twitter ao Gmail e YouTube, depende do JavaScript para criar páginas da web interativas e exibir conteúdo dinamicamente para os usuários.

Embora o JavaScript seja principalmente uma linguagem de front-end executada no navegador, ele também pode ser usado no lado do servidor por meio do Node.js para construir aplicativos de rede escalonáveis. Node.js é compatível com Linux, SunOS, Mac OS X e Windows.

Como o JavaScript tem uma sintaxe indulgente e flexível e funciona em todos os principais navegadores, é uma das linguagens de programação mais amigáveis para iniciantes.

2. Swift

Se você está interessado em produtos Apple e desenvolvimento de aplicativos móveis, Swift é um bom lugar para começar. Anunciado pela primeira vez pela Apple em 2014, o Swift é uma linguagem de programação relativamente nova usada para desenvolver aplicativos iOS e macOS.

O Swift foi otimizado para desempenho e construído desde o início para corresponder às realidades do desenvolvimento iOS moderno. O iOS não só funciona em todos os iPhone e iPad, mas também é a base para outros sistemas operacionais, como watchOS (para Apple Watches) e tvOS (para Apple TVs). Além disso, a Apple não vai a lugar nenhum como líder da indústria de tecnologia e os aplicativos iOS continuam a ser os mais lucrativos no mercado de aplicativos móveis.

3. Scala

Se você está familiarizado com Java, uma linguagem de programação clássica por si só, vale a pena conferir seu primo moderno, Scala. Scala combina os melhores recursos do Java (como sua estrutura orientada a objetos e seu ambiente de tempo de execução JVM ultrarrápido) com um toque moderno.

Como uma linguagem de programação funcional, Scala permite que os engenheiros elevem a qualidade de seu código para se assemelhar à matemática pura. Scala permite a programação simultânea, permitindo que procedimentos complexos sejam executados em paralelo. Além disso, é uma linguagem fortemente “tipada”. Os engenheiros podem criar e personalizar seus próprios tipos de dados, permitindo-lhes ter a paz de espírito sabendo que faixas inteiras de bugs são impossíveis em tempo de execução.

4. Go (GoLang)

Uma das principais linguagens favorecidas pelo Google, Go é a pequena linguagem que poderia. Como uma linguagem de baixo nível, Go é ideal para engenheiros que desejam entrar no campo da programação de sistemas. Ele abrange muito da mesma funcionalidade de C e C ++ sem a sintaxe difícil e curva de aprendizado íngreme. É a linguagem perfeita para construir servidores da web, pipelines de dados e até pacotes de aprendizado de máquina.

Como uma linguagem compilada, Go é executado "perto do metal", permitindo um tempo de execução extremamente rápido. É uma linguagem de código aberto, e desenvolvedores ambiciosos podem ver suas contribuições pessoais adotadas e apreciadas por programadores em todo o mundo.

5. Python

Python é talvez **a linguagem de programação mais amigável de todas** nesta lista. Costuma-se dizer que a sintaxe do Python é clara, intuitiva e quase semelhante ao inglês, o que, como o Java, o torna uma escolha popular para iniciantes.

Também como o Java, o Python tem uma variedade de aplicativos que o tornam uma opção versátil e poderosa na escolha da melhor linguagem de programação para seu caso de uso. Se você estiver interessado em desenvolvimento web de back-end, por exemplo, a estrutura Django de código aberto, escrita em Python, é popular, fácil de aprender e rica em recursos. Django tem sido usado no desenvolvimento de alguns sites populares como Mozilla, Instagram e Spotify.

Python também possui pacotes como NumPy e SciPy que são comumente usados nas áreas de computação científica, matemática e engenharia. Outras bibliotecas Python, como TensorFlow, PyTorch, scikit-learn e OpenCV, são usadas para criar programas em ciência de dados, aprendizado de máquina, processamento de imagens e visão computacional. A ciência e os aplicativos de dados do Python o tornam uma ótima escolha para os que têm inclinações acadêmicas.

6. Elm

Uma das línguas mais novas em nossa lista, o que começou como uma tese de estudante de Harvard agora cresceu e se tornou um ponto de paixão para desenvolvedores front-end em todo o mundo.

Elm compila em JavaScript, tornando-o ideal para construir interfaces de usuário de execução rápida com zero erros em tempo de execução. Elm é uma linguagem de programação funcional, permitindo que os desenvolvedores criem interfaces do lado do cliente sem as armadilhas declarativas de HTML e CSS.

7. Ruby

Ruby é outra linguagem de script comumente usada para desenvolvimento web. Em particular, é usado como base para a popular estrutura de aplicativo da web Ruby on Rails.

Iniciantes geralmente gravitam em torno de Ruby porque ele tem a reputação de ter uma das comunidades de usuários mais amigáveis e úteis. A comunidade Ruby até tem um ditado não oficial: “Matz é legal e nós somos legais”, encorajando os membros a modelar seu comportamento gentil e atencioso no inventor chefe de Ruby, Yukihiro Matsumoto.

Além da comunidade ativa e de sua sintaxe direta, Ruby também é uma boa linguagem para aprender, graças à sua associação com grandes empresas de tecnologia. Twitter, Airbnb, Bloomberg, Shopify e inúmeras outras startups criaram seus sites usando Ruby on Rails em algum momento.

8. C#

Como C++, C # (pronuncia-se C sharp) é uma linguagem orientada a objetos de propósito geral construída sobre os fundamentos de C. Foi originalmente projetada pela Microsoft como parte de sua estrutura .NET para construir aplicativos Windows.

C # usa uma sintaxe semelhante a outras linguagens derivadas de C, como C++, por isso é fácil aprender se você vem de outra linguagem da família C. C # não é apenas a escolha certa para o desenvolvimento de aplicativos da Microsoft, mas também é a linguagem que os desenvolvedores móveis usam para criar aplicativos de plataforma cruzada na plataforma Xamarin.

Além disso, qualquer pessoa interessada no desenvolvimento de Realidade Virtual (VR) deve considerar aprender C #. C # é a linguagem recomendada para construir videogames 3D e 2D usando o popular mecanismo de jogo Unity, que produz um terço dos principais jogos do mercado.

9. Rust

Rust é um pouco arrogante entre os outros idiomas desta lista, mas isso não significa que não seja um idioma valioso para aprender. A Pesquisa de Desenvolvedor de 2020 do Stack Overflow descobriu que Rust era a linguagem de programação mais amada entre os desenvolvedores pelo quinto ano consecutivo, com 86,1 por cento dos desenvolvedores do Rust dizendo que querem continuar trabalhando com ela.

Desenvolvido pela Mozilla Corporation, o Rust, assim como C e C++, destina-se principalmente à programação de sistemas de baixo nível. O que Rust adiciona à mistura, no entanto, é uma ênfase na velocidade e segurança. Rust enfatiza a escrita de um "código seguro", evitando que os programas acessem partes da memória que não deveriam, o que pode causar um comportamento inesperado e travamentos do sistema. As vantagens do Rust significam que outras grandes empresas de tecnologia, como Dropbox e Coursera, já estão começando a usá-lo internamente. Embora possa ser um pouco mais difícil de dominar do que outras linguagens para iniciantes, as habilidades de programação do Rust provavelmente terão uma boa recompensa, já que a popularidade da linguagem só continuará a crescer no futuro próximo.

É hora de codificar (programar).

Ao decidir qual linguagem de programação aprender, é importante não se envolver em tendências extravagantes e concursos de popularidade. As melhores linguagens de programação para aprender em 2021 são provavelmente as mesmas que foram melhores para aprender em 2017 e 2018, e isso continuará a ser verdade nos próximos anos também.

Embora o campo da programação de computadores mude rapidamente, as linguagens que discutimos acima têm um grande poder de permanência. Ao aprender um ou mais desses idiomas, você estará em uma excelente posição não apenas neste ano, mas nos anos que virão.

Ao iniciar sua jornada na codificação, somente você pode responder à pergunta sobre a melhor linguagem de programação a aprender. Faça sua seleção com base em seus interesses e no tipo de desenvolvimento de software em que deseja se envolver.

Síntese das melhores linguagens de programação por aplicação

- Desenvolvimento web frontend: JavaScript
- Desenvolvimento web backend: JavaScript, Java, Python, PHP, Ruby
- Desenvolvimento móvel: Swift, Java, C #
- Desenvolvimento de jogos: C ++, C #
- Aplicativos de desktop: Java, C ++, Python
- Programação de sistemas: C, Rust

As linguagens de programação mais populares em 2021 e além



Ranking linguagens de programação mais populares IEEE Spectrum

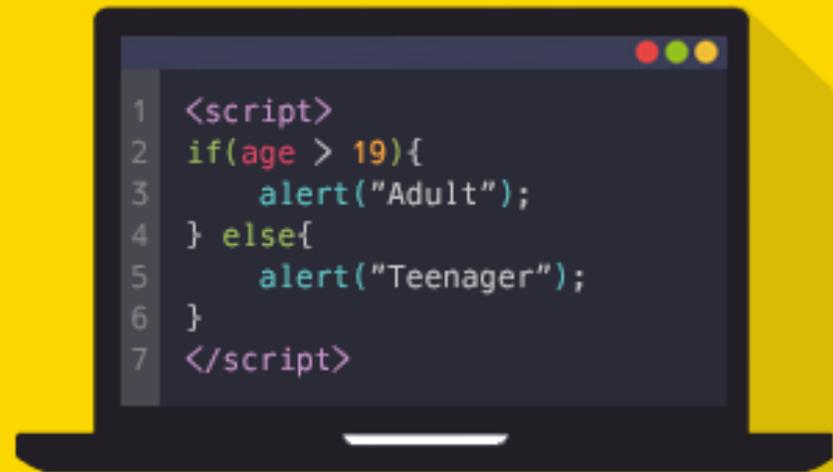
Language Rank	Types	Spectrum Ranking
1. Python	 	100.0
2. C	  	99.7
3. Java	  	99.5
4. C++	  	97.1
5. C#	  	87.7
6. R		87.7
7. JavaScript	 	85.6
8. PHP		81.2
9. Go	 	75.1
10. Swift	 	73.7

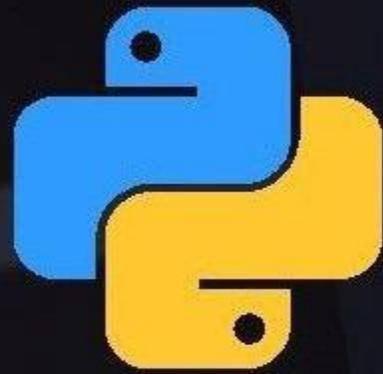
As linguagens de programação mais populares para Web





JavaScript





python

```
12 ACCEPTABLE_RETURN_TYPES = (str, int, float, bool)
```

```
13
```

```
14
```

```
15
```

```
16
```

```
17
```

```
18
```

```
19
```

```
20
```

```
21
```

```
22
```

```
23
```

```
24
```

```
25
```

```
26
```

```
class Base(ContainerAware, metaclass=abc.ABCMeta):
```

```
    """The base class for all controllers.
```

```
    Attributes:
```

```
    __action__ (string): The last action that was
```

```
    """
```

```
    def execute(self, **kwargs):
```

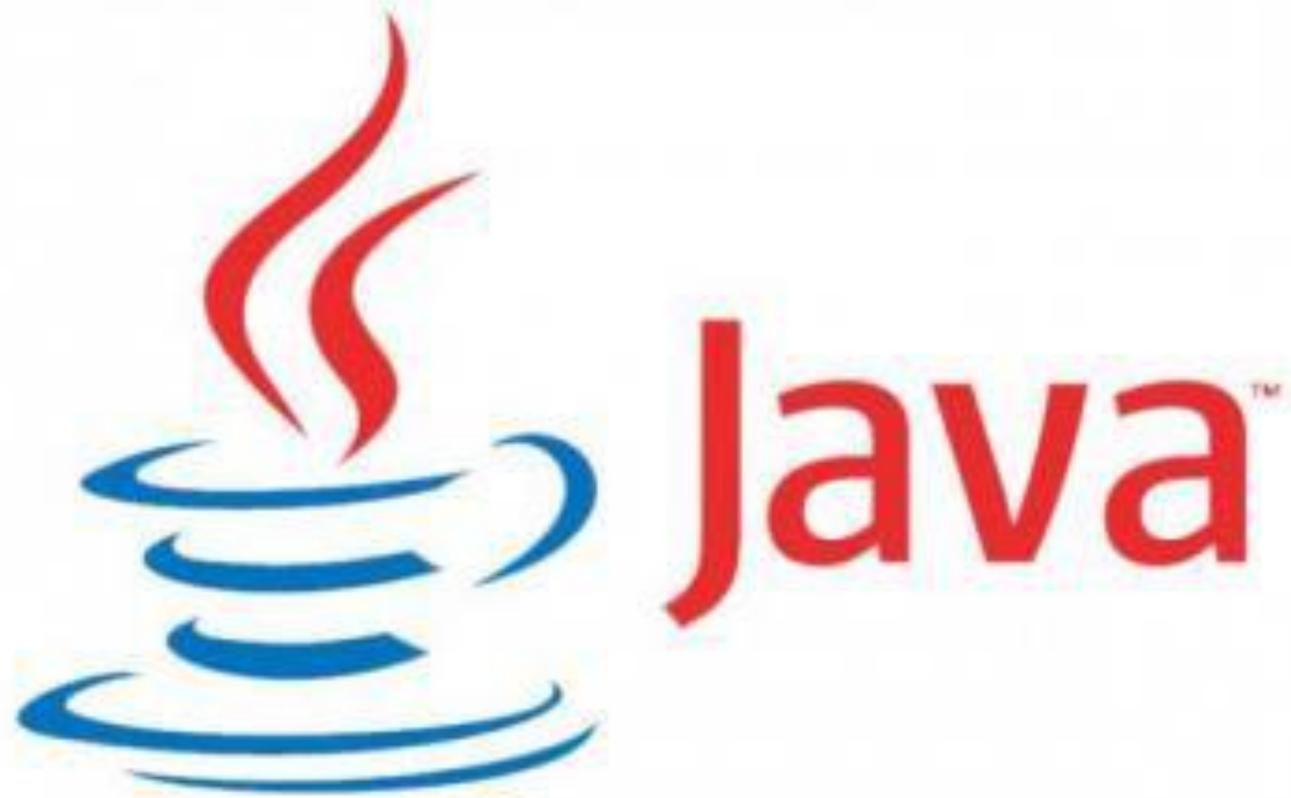
```
        method = self.get_execute_method(**kwargs)
```

```
        self.__action__ = method
```

```
        return method(**kwargs) or {}
```

```
    method
```

```
    kwargs):
```

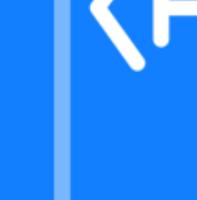


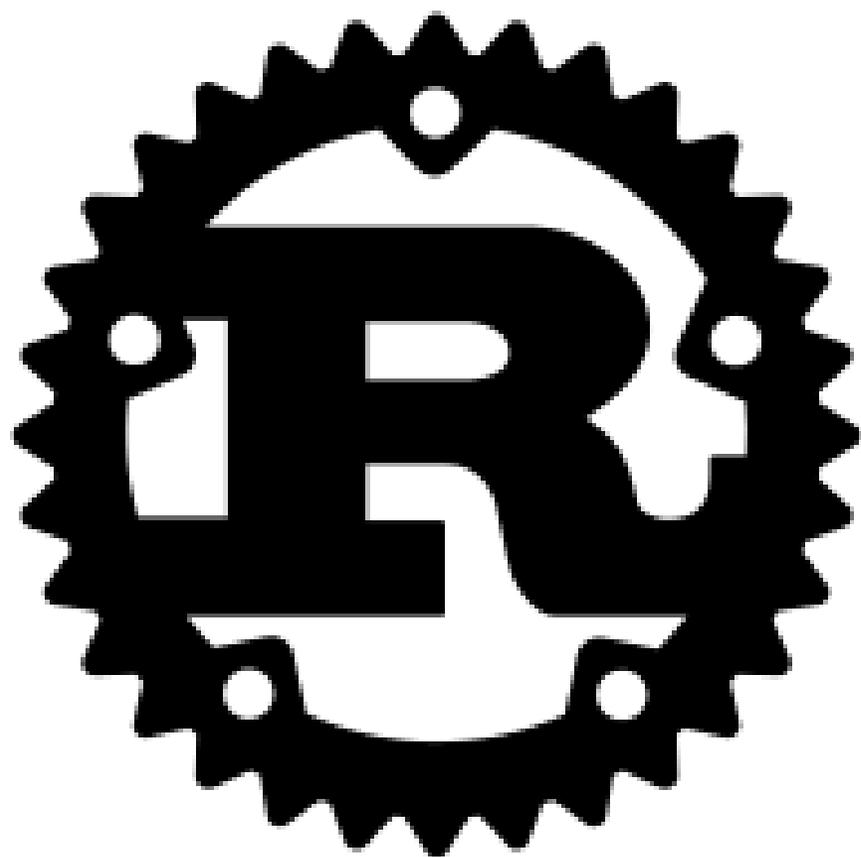




Kotlin

<HTML>





ust

TypeScript

CSS





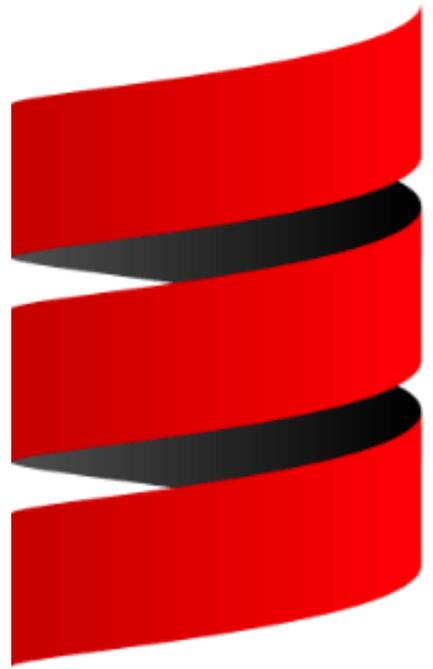
Swift

```
14 public partial class Form1 : Form
15 {
16     private SpeechSynthesizer _SS = new SpeechSynthesizer();
17
18     public Form1()
19     {
20         InitializeComponent();
21
22         foreach (var voice in _SS.GetVoices())
23         {
24             var voiceName = voice.VoiceName;
25
26             listBox1.Items.Add(voiceName);
27         }
28         _SS.Volume = 100;
29         _SS.Rate = 5;
30     }
31 }
32 }
33 }
34 }
```





Perl



Scala

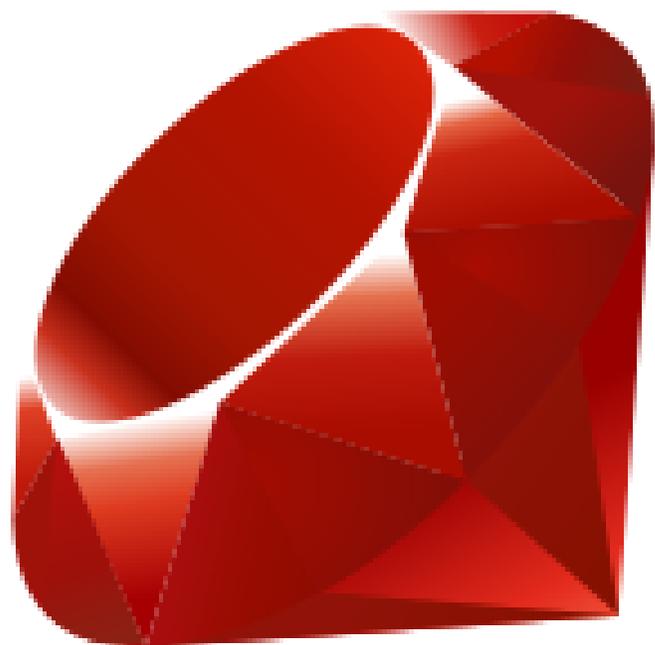
SQL



R Programming



Golang



Ruby

Mentalidade de um programador

Antes de terminarmos o capítulo, precisamos dar algumas palavras sobre o estilo de programação. Obviamente, o objetivo de escrever um programa de computador é fazer com que ele faça o que deve fazer, mas o estilo do programa também é importante. Seu código deve ser claro e conciso, além de ser consistente na maneira como segue as convenções - desde os nomes que você escolhe para as variáveis até a quantidade de espaço em branco que você deixa entre os blocos de código. Seu código também deve ser fácil de ler e incluir comentários que explicam o que o código faz.

Mentalidade de um programador

Você deve sempre procurar melhorar seu código para torná-lo mais eficiente e fácil de seguir. Esse processo é chamado de refatoração e deve ser feito regularmente para que seu código permaneça atualizado e não se torne obsoleto.

Mentalidade de um programador

Um princípio importante na programação é a regra "Não se repita", ou DRY (***Don't Repeat Yourself***). Isso significa que você deve sempre evitar a repetição de muito código. Seguir este princípio ajudará a tornar seu código mais flexível e fácil de manter e atualizar. (É sempre mais fácil mudar um pedaço de código do que centenas!) Se isso soa um pouco como se os programadores fossem preguiçosos, então você está absolutamente certo! Um bom programador sempre estará procurando escrever um código que faça o trabalho da maneira mais eficiente e elegante possível. Menos código significa menos chance de bugs e é mais fácil de manter.

Mentalidade de um programador

Um bom programa não aparece apenas instantaneamente. Ele começa como um plano e cresce organicamente até seu estado final. O desenvolvimento de um programa envolve muita solução de problemas e, às vezes, você pode dar errado e ficar preso. Isso é normal e você deve estar preparado para fazer alterações e se adaptar. Um bom plano e um código bem comentado que segue o princípio DRY impedirá que seu código se torne uma bagunça complexa que é impossível de seguir e compreender.

Resumo

- Um algoritmo é um conjunto de instruções passo a passo que completam uma tarefa específica.
- Pseudocódigo é usado para escrever programas sem usar uma linguagem de programação específica.
- JavaScript foi criado em 1995 e é considerado a linguagem da web.
- O ambiente principal do JavaScript é o navegador, mas também pode ser executado em outros ambientes.
- Cada versão do JavaScript deve ser compatível com versões anteriores.
- O código deve ser fácil de ler e refatorado regularmente para mantê-lo funcionando sem problemas.
- Lembre-se do princípio DRY ao codificar. Não se repita!