

Introdução ao Ruby

Tenille Martins

DIO Tech Education



bytemartins



Tenille10

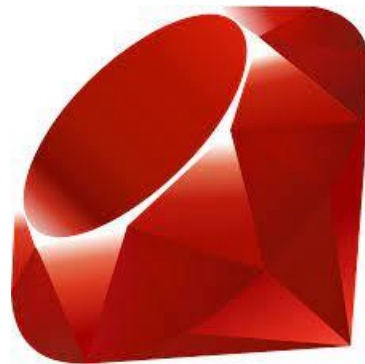


Tenille Martins

Sobre Mim

- Graduanda em Segurança da Informação;
- Professora de Física e Matemática;
- Go Expert;
- Instrutora de Ruby;
- Estudiosa de Python e SI;
- Adoro ver séries, ler, viajar;
- Insta: @bytemartins, Github: Tenille10 e LinkedIn: Tenille Martins

Objetivo Geral



Este curso foi planejado para programadores sem nenhum conhecimento prévio de Ruby; ao final deste primeiro Curso, o DEV conseguirá escrever programas simples em Ruby e será capaz de ler e interpretar um código dentro dessa linguagem.

Pré-requisitos

- Um computador com acesso a internet;
- Muita vontade de aprender;



Percurso

Etapa 1

História do Ruby

Etapa 2

Instalando o Ruby

Etapa 3

Seu primeiro programa

Percurso

Etapa 4

Ruby Irb

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



História do Ruby

// Introdução ao Ruby

Percurso

Etapa 1

História do Ruby

Etapa 2

Instalando o Ruby

Etapa 3

Seu primeiro programa

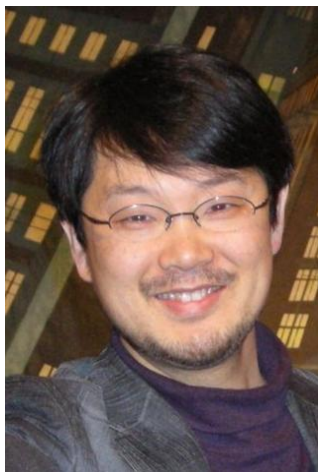
Percurso

Etapa 4

Ruby Irb

História da linguagem

- Criação: 1993;
- Criador: Yukihiro “Matz” Matsumoto ;



Curiosidades da linguagem

- Desenvolvida para ser produtiva e divertida.
- Prioriza que o entendimento de um software deve aproximar-se mais do pensamento humano do que computacional.
- Possui sintaxe bastante natural, de fácil leitura e com código narrado quase que de forma descritiva , permitindo o entendimento de trechos de código logo no início de seus estudos.

Como é a linguagem Ruby?

- Interpretada multiparadigma;
- Tipagem Dinâmica e forte;
- Orientada a Objetos (com herança);
- Gerenciamento de memória automático;
- Open Source;

Por que estudar Ruby?

1. Linguagem popular entre Startups:

A simplicidade e legibilidade dessa linguagem resultam em produtividade e faz da linguagem uma ótima escolha para start ups - ou qualquer empresa que deseje entregar protótipos o mais cedo possível - enquanto o mundo corporativo tende a preferir linguagens como Java e [PHP](#).

Por que estudar Ruby?

2. Ruby on Rails

O Rails é o principal framework de Ruby. Porque? Porque ele é poderosíssimo. Ok, talvez você não saiba direito o que é um framework, então: são coleções de bibliotecas que facilitam muito a vida do desenvolvedor e permitem construir aplicações de modo muito mais rápido.

Por que estudar Ruby?

- 3. Ruby é a 10ª linguagem nos índices de popularidade.
- 4. É fácil e divertida, por isso, fácil de aprender!!!
- 5. Está presente em várias áreas.

Onde Ruby pode ser aplicada?

- Aplicações Web;
- Programas de desktop;
- Automação de testes;
- Área de DevOps;
- Ruby virou uma poderosa linguagem para Backend;

Empresas que usam Ruby



Site Oficial da linguagem

Ruby: www.ruby-lang.org

Ruby on rails: www.rubyonrails.org

Ruby Installer: www.rubyinstaller.org

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



Instalando o Ruby

// Introdução ao Ruby

Percurso

Etapa 1

~~História do Ruby~~

Etapa 2

Instalando o Ruby

Etapa 3

Seu primeiro programa

Percurso

Etapa 4

Ruby Irb

Site Oficial da linguagem

Ruby: www.ruby-lang.org

Ruby on rails: www.rubyonrails.org

Ruby Installer: www.rubyinstaller.org

VS Code: <https://code.visualstudio.com/download>

Codeanywhere: <https://codeanywhere.com/signin>

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



Seu primeiro programa

// Introdução ao Ruby

Percurso

~~Etapa 1~~

~~História do Ruby~~

~~Etapa 2~~

~~Instalando o Ruby~~

Etapa 3

Seu primeiro programa

Percurso

Etapas

Etapas

Etapas 1

Etapas 2

Etapas 4

Etapas 5

Ruby Irb

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



Ruby Irb

// Introdução ao Ruby

Percurso

~~Etapa 1~~

~~História do Ruby~~

~~Etapa 2~~

~~Instalando o Ruby~~

~~Etapa 3~~

~~Seu primeiro programa~~

Percorso

Etapa 4

Ruby Irb

Site Oficial da linguagem

Ruby IRB: significa *Interactive Ruby*

Links Úteis

- **Repositório no GitHub:**
<https://github.com/Tenille10/>
- **Documentação Oficial: https:**
www.ruby-lang.org
- **Referências:** Ruby: Aprenda a programar na linguagem mais divertida.

Para saber mais

- **Site na Web:** www.ruby-lang.org
- **Artigo:** <https://pt.wikipedia.org/wiki/Ruby>
- **Livro:** Ruby: Aprenda a programar na linguagem mais divertida.

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



Conceitos de Ruby

Tenille Martins

DIO Tech Education



bytemartins

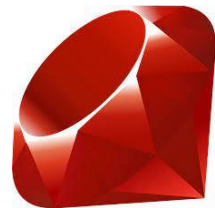


Tenille10



Tenille Martins

Objetivo Geral



Este curso foi planejado para programadores sem nenhum conhecimento prévio de Ruby; ao final deste primeiro Curso, o DEV conseguirá escrever programas simples em Ruby e será capaz de ler e interpretar um código dentro dessa linguagem.

Pré-requisitos

- Um computador com acesso a internet;
- Muita vontade de aprender;

Como as pessoas pensam que é programação



Como realmente é



Percurso

Etapa 1

Tipos de dados

Etapa 2

Operadores matemáticos

Etapa 3

Strings

Percurso

Etapa 4

Entradas e Saídas

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



Tipos de dados

// Conceitos de Ruby

Percorso

Etapa 1

Tipos de dados

Etapa 2

Operadores matemáticos

Etapa 3

Strings

Percurso

Etapa 4

Entradas e Saídas

Tipos de Dados

- Classificação dos dados que iremos utilizar em nosso programa.
- Tipagem dinâmica: Quem vai classificar é o próprio Ruby

Quais são?

- Integer: Inteiro.
- Float: Números não inteiros (números decimais, números que contém pontos flutuantes.).
- Boolean: Verdadeiro ou falso (tomada de decisões).
- String: Conjunto de caracteres (escrever textos).

Quais são?

- Array: Permite armazenar uma lista ordenada, utilize colchetes [].
- Symbol: Assemelha-se a String, mas ele é imutável, isto é, é um objeto que ocupa sempre o mesmo espaço na memória, use : (maisonome) para representá-lo.
- Hash: Coleção de dados organizados por chave única e seus valores, use {} para representá-lo .

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



Operadores Matemáticos

// Conceitos em Ruby

Percurso

Etapa 1

~~Tipos de dados~~

Etapa 2

Operadores matemáticos

Etapa 3

Strings

Percurso

Etapa 4

Entradas e Saídas

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



Strings

// Conceitos em Ruby

Percurso

Etapa 1

~~Tipos de dados~~

Etapa 2

~~Operadores matemáticos~~

Etapa 3

Strings

Percurso

Etapa 4

Entradas e Saídas

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



Entradas e Saídas

// Conceitos em Ruby

Percurso

Etapa 1

~~Tipos de dados~~

Etapa 2

~~Operadores matemáticos~~

Etapa 3

~~Strings~~

Percurso

Etapa 4

Entradas e Saídas

Links Úteis

- **Repositório no GitHub:**
<https://github.com/Tenille10/>
- **Documentação Oficial: https:**
www.ruby-lang.org
- **Referências:** Ruby: Aprenda a programar na linguagem mais divertida.

Para saber mais

- **Site na Web:** www.ruby-lang.org
- **Livro:** Ruby: Aprenda a programar na linguagem mais divertida.

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



Estruturas de Controle

Tenille Martins

DIO Tech Education



bytemartins



Tenille10



Tenille Martins

Objetivo Geral



Este curso foi planejado para programadores sem nenhum conhecimento prévio de Ruby; ao final deste primeiro Curso, o DEV conseguirá escrever programas simples em Ruby e será capaz de ler e interpretar um código dentro dessa linguagem.

Pré-requisitos

- Um computador com acesso a internet;
- Muita vontade de aprender;

Como as pessoas pensam que é programação



Como realmente é



Percurso

Etapa 1

If, Else, Elsif

Etapa 2

Unless

Etapa 3

Case

Percurso

Etapa 4

For

Etapa 5

While

Etapa 6

Times

Percurso

Etapa 7

Do/While

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



If, Else, Elsif

// Estruturas de Controle

Percorso

Etapa 1

If, Else, Elsif

Etapa 2

Unless

Etapa 3

Case

Percurso

Etapa 4

For

Etapa 5

While

Etapa 6

Times

Percurso

Etapa 7

Do/While

Estruturas de Controle

- É uma parte do código que escrevemos para uma “tomada de decisão”.
- Dois tipos: **Condicional** (condição de execução) e Iteração (repetição de um conjunto de instruções)

Iteração

- If (se): Determina se uma instrução é verdadeira(true). A partir desse resultado determina se as instruções dentro do bloco serão executadas.
- Else (senão): Caso o If seja falso(false), ele informa o que deverá ser feito.
- Elsif: Quando for necessário verificar mais uma condição If

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



Unless

// Estruturas de Controle

Percurso

Etapa 1

~~If, Else, Elsif~~

Etapa 2

Unless

Etapa 3

Case

Percurso

Etapa 4 For

Etapa 5 While

Etapa 6 Times

Percurso

Etapa 7

Do/While

Tipos de Dados

- Unless (a não ser que): Executado quando a condição for falsa.

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



Case

// Estruturas de Controle

Percurso

~~Etapa 1~~

~~If, Else, Elsif~~

~~Etapa 2~~

~~Unless~~

Etapa 3

Case

Percurso

Etapa 4 For

Etapa 5 While

Etapa 6 Times

Percurso

Etapa 7

Do/While

Tipos de Dados

- Case (caso): Parecido com o If, mas quando há diversas condições. (diversos casos).

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



For

// Estruturas de Controle

Percurso

~~Etapa 1~~

~~If, Else, Elsif~~

~~Etapa 2~~

~~Unless~~

~~Etapa 3~~

~~Case~~

Percorso

Etapa 4

For

Etapa 5

While

Etapa 6

Times

Percurso

Etapa 7

Do/While

Estruturas de Controle

- É uma parte do código que escrevemos para uma “tomada de decisão”.
- Dois tipos: Condicional (condição de execução) e **Iteração** (repetição de um conjunto de instruções)

Iteração

- For (“durante”): Usado para percorrer uma coleção de elementos.

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



While

// Estruturas de Controle

Percurso

~~Etapa 1~~

~~If, Else, Elsif~~

~~Etapa 2~~

~~Unless~~

~~Etapa 3~~

~~Case~~

Percurso

Etapa 4

For

Etapa 5

While

Etapa 6

Times

Percurso

Etapa 7

Do/While

Iteração

- While (enquanto): Repete um bloco **enquanto** sua condição for verdadeira.

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



Times

// Estruturas de Controle

Percurso

~~Etapa 1~~

~~If, Else, Elsif~~

~~Etapa 2~~

~~Unless~~

~~Etapa 3~~

~~Case~~

Percurso

~~Etapa 4~~

~~For~~

~~Etapa 5~~

~~While~~

Etapa 6

Times

Percurso

Etapa 7

Do/While

Iteração

- Times (“vezes”): Executa uma repetição por determinado número de **vezes**.

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



Do while

// Estruturas de Controle

Percurso

~~Etapa 1~~

~~If, Else, Elsif~~

~~Etapa 2~~

~~Unless~~

~~Etapa 3~~

~~Case~~

Percurso

~~Etapa 4~~

~~For~~

~~Etapa 5~~

~~While~~

~~Etapa 6~~

~~Times~~

Percurso

Etapas

- 1. Introdução
- 2. Variáveis
- 3. Operadores
- 4. Estruturas de Controle
- 5. Funções
- 6. Vetores
- 7. Do/While**
- 8. Ponteiros
- 9. Arquivos
- 10. Bibliotecas

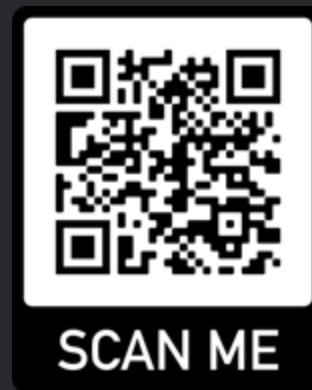
Do/While

Iteração

- Do/While (fazer enquanto): Cria uma laço de repetição que só irá parar quando uma instrução BREAK for verdadeira.

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



Links Úteis

- **Repositório no GitHub:**
<https://github.com/Tenille10/>
- **Documentação Oficial: https:**
www.ruby-lang.org
- **Referências:** Ruby: Aprenda a programar na linguagem mais divertida.

Para saber mais

- **Site na Web:** www.ruby-lang.org
- **Livro:** Ruby: Aprenda a programar na linguagem mais divertida.

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)

