

# Glossário Completo de Ruby para Iniciantes

Este glossário foi desenvolvido para pessoas que desejam **aprender Ruby do zero**, mesmo que **nunca tenham programado antes**. Ele traz conceitos básicos, exemplos comentados e explicações claras sobre a linguagem.

---

## ✦ O que é Ruby?

Ruby é uma linguagem de programação **simples, elegante e poderosa**, muito conhecida por ser usada no framework **Ruby on Rails** para desenvolvimento web. Tem uma sintaxe parecida com o inglês e é bastante amigável para quem está começando.

Exemplo básico:

```
puts "Olá, mundo!"
```

---

## ✦ Variáveis

Ruby **não exige declaração de tipo**:

```
nome = "Lucas"  
idade = 25  
altura = 1.75
```

---

## ✦ Tipos de Dados

Tipo	Exemplo
Integer	10, -5, 100
Float	3.14, -0.5
String	"Texto"
Boolean	true, false
Array	[1, 2, 3]
Hash	{nome: "Ana"}
Symbol	:nome
Nil	nil (nulo)

---

## ✦ Operadores

Tipo	Exemplo
Aritméticos	+, -, *, /, %, **
Comparação	==, !=, >, <, >=, <=
Lógicos	&&, ^

---

## ✦ Entrada de Dados

```
puts "Digite seu nome:"
nome = gets.chomp
puts "Olá, #{nome}!"
```

---

## ✦ Condicionais

```
idade = 18

if idade >= 18
  puts "Maior de idade"
else
  puts "Menor de idade"
end
```

---

## ✦ Estruturas de Repetição

### For:

```
for i in 1..5
  puts i
end
```

### While:

```
contador = 0
while contador < 5
  puts contador
  contador += 1
end
```

### Times:

```
5.times do
  puts "Repetindo..."
end
```

---

## ✦ Arrays

```
frutas = ["Maçã", "Banana", "Uva"]
```

```
puts frutas[0]
```

Adicionar elemento:

```
frutas << "Pera"
```

---

## ✦ Hashes (Dicionários)

```
peessoa = {nome: "Ana", idade: 30}  
puts pessoa[:nome]
```

---

## ✦ Métodos (Funções)

```
def saudacao(nome)  
  "Olá, #{nome}!"  
end  
  
puts saudacao("Carlos")
```

---

## ✦ Funções Lambda (Procs e Lambdas)

```
soma = -> (a, b) { a + b }  
puts soma.call(2, 3)
```

---

## ✦ Manipulação de Strings

```
texto = "Ruby é incrível"  
puts texto.upcase  
puts texto.downcase  
puts texto.gsub("incrível", "fácil")
```

---

## ✦ Manipulação de Arquivos

```
File.open("arquivo.txt", "w") do |arquivo|  
  arquivo.puts "Olá, arquivo!"  
end
```

---

## ✦ Tratamento de Erros

```
begin  
  numero = Integer(gets)  
rescue ArgumentError  
  puts "Digite um número válido!"  
end
```

---

## ✦ Classes e Objetos (Programação Orientada a Objetos)

```
class Pessoa
  def initialize(nome)
    @nome = nome
  end

  def falar
    puts "Olá, meu nome é #{@nome}"
  end
end

p1 = Pessoa.new("Lucas")
p1.falar
```

---

## ✦ Iteradores e Blocos

```
[1, 2, 3].each do |num|
  puts num * 2
end
```

---

## ✦ Gems (Bibliotecas externas)

Instalando:

```
gem install nome_da_gem
```

Usando:

```
require 'nome_da_gem'
```

---

## ✦ Programação Funcional

Map:

```
quadrados = [1, 2, 3].map { |n| n**2 }
puts quadrados
```

---

## ✦ Boas Práticas

- ☐ Usar nomes claros
- ☐ Separar responsabilidades em métodos pequenos

- ☐ Utilizar classes para organizar melhor o código
  - ☐ Testar o código frequentemente
- 

## ✦ Aplicações Comuns em Ruby

- **Desenvolvimento Web (Ruby on Rails)**
- **Automação de tarefas**
- **Manipulação de arquivos e sistemas**
- **APIs e Web Services**