

RUBY 2 DIA GETS E NÚMEROS

1. `gets`

- **O que é?:** O `gets` é um método em Ruby usado para ler a entrada do usuário do teclado.
- **Como funciona?:** Quando você chama `gets`, o programa "pausa" e espera que o usuário digite algo e pressione Enter.
- **Exemplo:**

```
ruby
CopyEdit
nome = gets
puts "Olá, #{nome}!"
```
- **Observação:** Ele lê a entrada do usuário e inclui o caractere de nova linha (`\n`) no final da string.

2. `chomp`

- **O que é?:** O `chomp` é um método usado para remover o caractere de nova linha (`\n`) de uma string.
- **Como funciona?:** Normalmente, o `gets` captura a entrada do usuário com o `\n` no final. Usando `chomp`, você pode remover esse caractere extra.
- **Exemplo:**

```
ruby
CopyEdit
nome = gets.chomp
puts "Olá, #{nome}!"
```
- **Observação:** `chomp` pode ser chamado em qualquer string, não só na entrada de usuário, e irá remover o `\n` ou qualquer caractere final especificado.

3. `to_i`

- **O que é?:** O `to_i` é um método que converte uma string em um número inteiro (integer).
- **Como funciona?:** Se a string contiver um valor numérico válido, ele converte essa string em um número inteiro. Se não for possível, retorna 0.
- **Exemplo:**

```
ruby
CopyEdit
numero_str = "10"
numero_int = numero_str.to_i
```

```
puts numero_int # 10
```

- **Observação:** Se a string não representar um número válido, `to_i` retorna 0.

4. `to_s`

- **O que é?:** O `to_s` é um método que converte qualquer tipo de dado em uma string.
- **Como funciona?:** Ele pode ser usado para garantir que qualquer objeto, número ou valor seja convertido para sua representação em string.

- **Exemplo:**

```
ruby
CopyEdit
numero = 10
numero_str = numero.to_s
puts numero_str # "10"
```

- **Observação:** Isso é útil quando você precisa concatenar valores com strings ou gerar uma representação legível de outros tipos de dados.

NUMEROS

1. Tipos de Números em Ruby

- **Inteiros (Integer):** São números inteiros, sem casas decimais.
 - Exemplo: 10, -3, 0
- **Ponto flutuante (Float):** São números com casas decimais.
 - Exemplo: 3.14, -0.5, 0.0

2. Operações Básicas com Números

- **Soma (+):** Soma dois números.

```
ruby
CopyEdit
5 + 3 # Resultado: 8
```

- **Subtração (-):** Subtrai um número de outro.

```
ruby
CopyEdit
10 - 4 # Resultado: 6
```

- **Multiplicação (*):** Multiplica dois números.

```
ruby
CopyEdit
4 * 7 # Resultado: 28
```

- **Divisão (/):** Divide um número por outro.

```
ruby
CopyEdit
10 / 2 # Resultado: 5
```

- **Divisão com ponto flutuante (/):** Quando um dos operandos é um número de ponto flutuante, a divisão será com decimais.

```
ruby
CopyEdit
10 / 3.0 # Resultado: 3.3333333333333335
```

- **Módulo (%):** Retorna o resto da divisão.

```
ruby
CopyEdit
10 % 3 # Resultado: 1
```

- **Exponenciação (**):** Eleva um número a uma potência.

```
ruby
CopyEdit
2 ** 3 # Resultado: 8
```

3. Métodos Úteis para Números

- **abs:** Retorna o valor absoluto de um número.

```
ruby
CopyEdit
(-10).abs # Resultado: 10
```

- **round:** Arredonda um número flutuante para o número inteiro mais próximo.

```
ruby
CopyEdit
3.14159.round # Resultado: 3
```

- **to_i:** Converte um número flutuante ou string para um inteiro.

```
ruby
CopyEdit
3.14.to_i # Resultado: 3
"100".to_i # Resultado: 100
```

- **to_f:** Converte um número inteiro ou string para um número flutuante.

```
ruby
CopyEdit
10.to_f # Resultado: 10.0
"3.14".to_f # Resultado: 3.14
```

4. Conversões entre Tipos de Números

- Para realizar operações com números inteiros e flutuantes ao mesmo tempo, Ruby automaticamente converte inteiros para flutuantes, se necessário. Por exemplo:

```
ruby
CopyEdit
5 / 2 # Resultado: 2 (Divisão inteira)
```

```
5.0 / 2 # Resultado: 2.5 (Divisão com ponto flutuante)
```

- Se você quiser forçar a conversão de um número para outro tipo, use os métodos `to_i` ou `to_f`.

5. Precedência de Operadores

- Em Ruby, os operadores têm precedência, ou seja, alguns operadores são avaliados primeiro que outros. Por exemplo:

```
ruby
CopyEdit
2 + 3 * 4 # Resultado: 14 (Multiplicação é feita primeiro)
```

- Para mudar a ordem das operações, você pode usar parênteses:

```
ruby
CopyEdit
(2 + 3) * 4 # Resultado: 20
```

6. EXEMPLO DE CALCULADORA

```
numero.rb
1 puts "Digite primeiro numero"
2 numero1=gets.chomp.to_i
3
4 puts "Digite o segundo numero"
5 numero2=gets.chomp.to_i
6
7 soma = numero1+numero2
8 mult = numero1*numero2
9 div=numero1/numero2
10 subtr=numero1-numero2
11
12 puts "A soma do #{numero1} e #{numero2} é #{soma}."
13 puts "A multiplicação do #{numero1} e #{numero2} é #{mult}"
14 puts "A divisão do #{numero1} e #{numero2} é #{div}"
15 puts "A subtração do #{numero1} e #{numero2} é #{subtr}"
```