

1. while

O **while** repete um bloco de código enquanto a condição for **verdadeira**.

Exemplo:

```
contador = 1

while contador <= 5
  puts "Contagem: #{contador}"
  contador += 1
end
```

- Aqui, o código vai imprimir os números de **1** a **5**.
 - A condição `contador <= 5` é verificada antes de cada iteração.
-

2. until

O **until** é semelhante ao **while**, mas a condição é verificada até que seja **falsa**.

Exemplo:

```
contador = 1

until contador > 5
  puts "Contagem: #{contador}"
  contador += 1
end
```

- A execução continua enquanto a condição for **falsa** (nesse caso, enquanto `contador <= 5`).
-

3. for

O **for** é um tipo de loop tradicional, que percorre uma sequência de valores.

Exemplo:

```
for i in 1..5
  puts "Contagem: #{i}"
end
```

- Aqui, o loop percorre o intervalo de 1 a 5 (`1..5`) e imprime o valor de `i` a cada iteração.
-

4. each

O **each** é muito utilizado em **arrays** e **hashes**. Ele itera sobre cada elemento da coleção.

Exemplo com Array:

```
frutas = ["maçã", "banana", "laranja"]

frutas.each do |fruta|
  puts "Eu gosto de #{fruta}!"
end
```

- O bloco de código (do ... end) será executado para cada item da lista `frutas`.

Exemplo com Hash:

```
usuario = { nome: "Weslley", idade: 25, cidade: "São Paulo" }

usuario.each do |chave, valor|
  puts "#{chave}: #{valor}"
end
```

- Aqui, o método **each** percorre o Hash e executa o bloco para cada par chave-valor.
-

5. times

O **times** é uma forma mais compacta de repetir algo um número específico de vezes.

Exemplo:

```
5.times do
  puts "Repetindo a mensagem!"
end
```

- Este código vai imprimir a mensagem **5 vezes**.
-

Conclusão:

Essas estruturas são fundamentais para controlar a repetição de blocos de código em Ruby. Cada uma tem suas particularidades e pode ser escolhida conforme a situação que você está tratando.