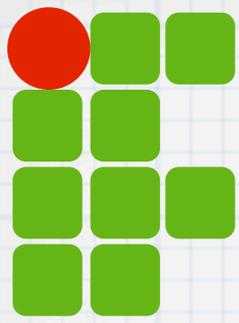


INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Programação de Computadores

Mais arrays

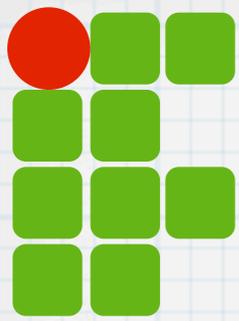
Copyright © 2013 IFRN



O que veremos hoje?

- * Arrays
- * Cópia de arrays
- * Iteração sobre os elementos do array
- * String como arrays
 - * Array de caracteres
 - * Acesso através do conteúdo
 - * Métodos split, join
 - * Método gsub
- * Exercícios

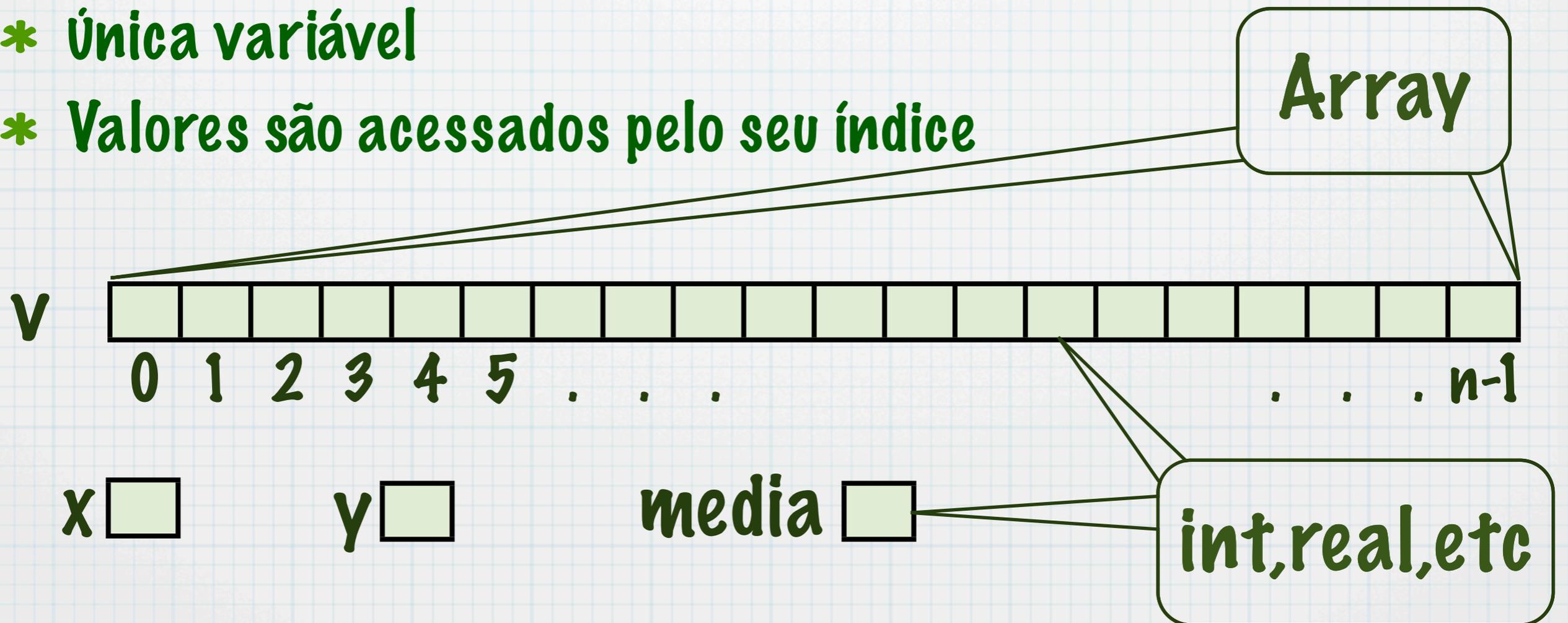


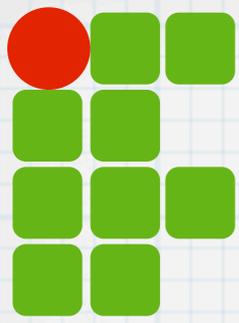


Relembrando

* O que são arrays?

- * Um agregado de elementos
- * Capacidade de armazenar uma coleção de valores
- * Única variável
- * Valores são acessados pelo seu índice

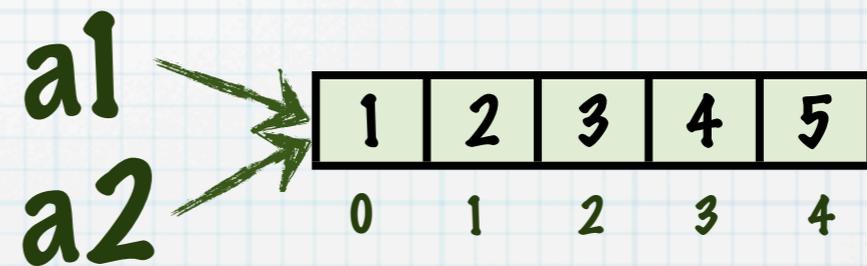




Cópia

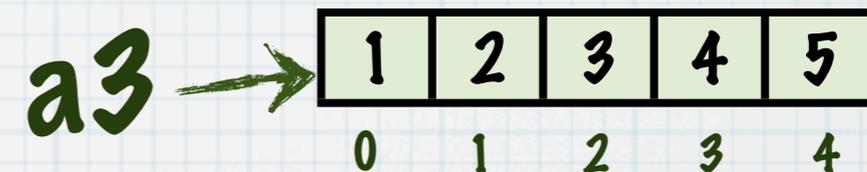
* Atribuição simples não copia

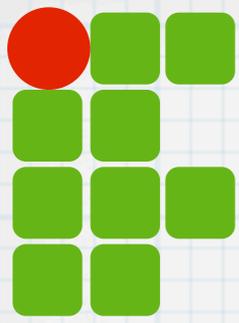
```
a1 = [1, 2, 3, 4, 5]  
a2 = a1
```



* Método dup

```
a1 = [1, 2, 3, 4, 5]  
a3 = a1.dup
```



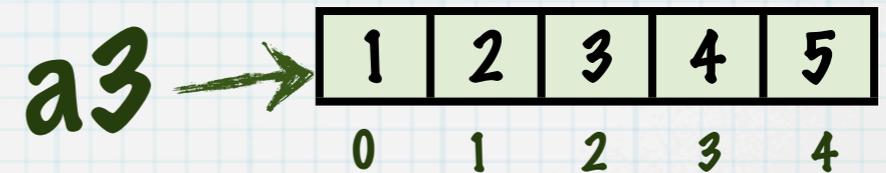
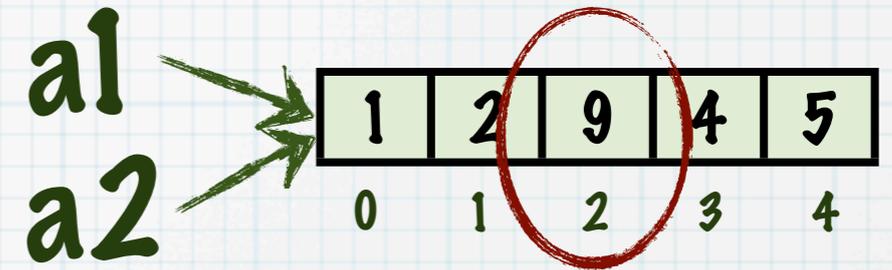


Cópia

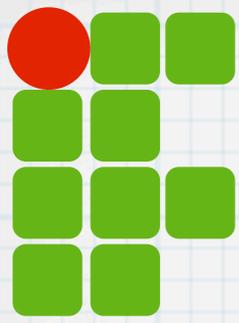
* CUIDADO

- * a1 e a2 são o mesmo
- * Indiferente alterar a1 ou a2

```
a1[2] = 9  
puts a2[2]
```



- * Sempre use o dup para criar um novo array que é cópia de outro



Iteração

* Operação for sobre arrays

Processa o
bloco com
cada elemento
do array
atribuído a x

```
for x in array do
```

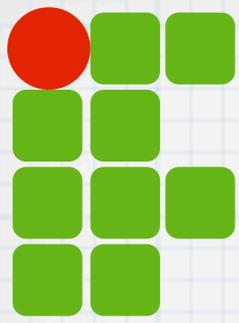
```
  ...
```

```
  ...
```

```
  usa x
```

```
  ...
```

```
end
```



Iteração

* Exemplo

```
a = [12, 3, 13, 34, 65]
```

```
soma = 0
```

```
for num in a do  
  soma = soma+num  
end
```

Mesmo que

```
soma = 0
```

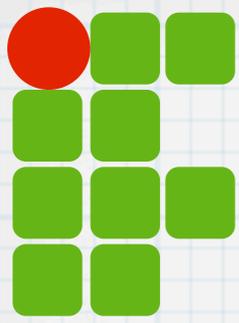
```
soma = soma+a[0]
```

```
soma = soma+a[1]
```

```
soma = soma+a[2]
```

```
soma = soma+a[3]
```

```
soma = soma+a[4]
```



Iteração

* Exemplo

```
a = [12, 3, 13, 34, 65]
```

```
soma = 0
```

```
for num in a do
```

```
  soma = soma + num
```

```
end
```

Mesmo que

```
soma = 0
```

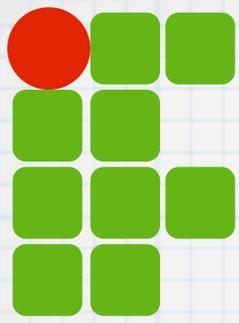
```
soma = soma + a[0]
```

```
soma = soma + a[1]
```

```
soma = soma + a[2]
```

```
soma = soma + a[3]
```

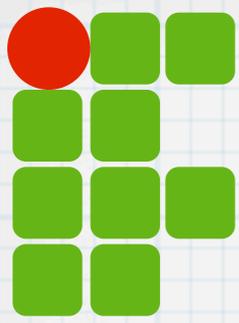
```
soma = soma + a[4]
```



Exemplo

* Somar os quadrados dos elementos de um array

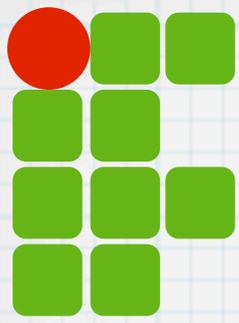
```
soma = 0
for num in a do
  soma = soma+(num*num)
end
puts soma
```



Exemplo

* **Mostrar os números pares**

```
for num in a do
  if (num%2 == 0) then
    puts num
  end
end
```



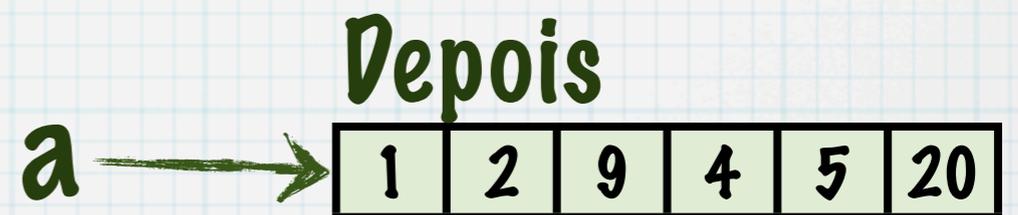
Arrays

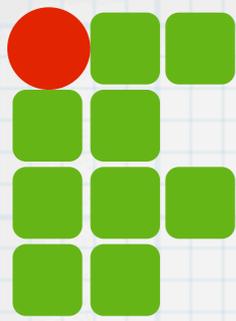
* Adiciona elemento no fim de um array

* Operador <<

```
a = [12, 3, 13, 34, 65]
```

```
a << 20
```



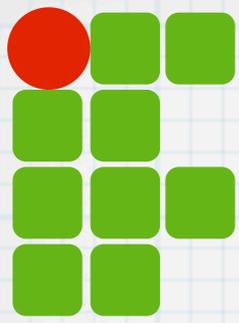


Exemplo

- * Criar dois array, um com as notas maiores que a média e outro com as menores que a média

```
notas = 30.times.map do gets.to_f end
maiores = []
menores = []
for nota in notas do
  if (nota < 6.0) then
    menores << nota
  else
    maiores << nota
  end
end

puts maiores
puts menores
```



Soma de arrays

* Um terceiro array é criado

$a1 = [1, 2, 3, 4]$

$a2 = [5, 4, 3, 2, 1]$

$a3 = a1 + a2$

$a1 \rightarrow$

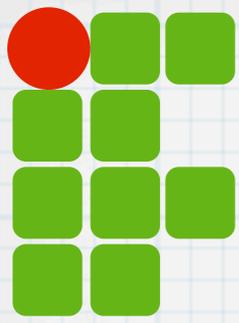
1	2	3	4
---	---	---	---

$a2 \rightarrow$

5	4	3	2	1
---	---	---	---	---

$a3 \rightarrow$

1	2	3	4	5	4	3	2	1
---	---	---	---	---	---	---	---	---



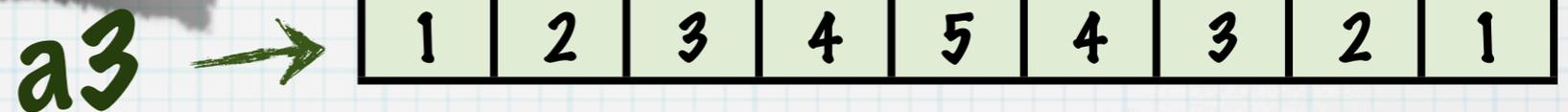
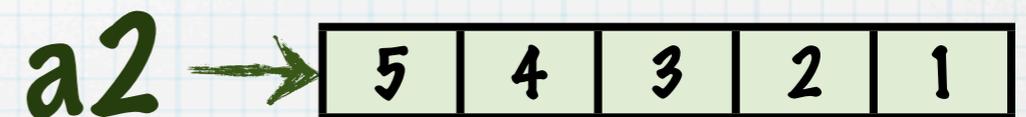
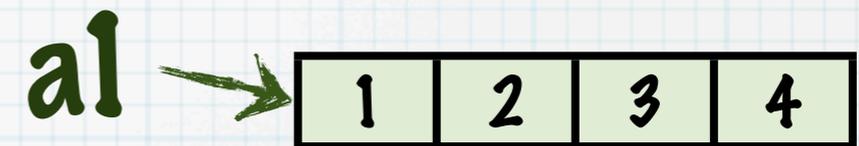
Soma de arrays

* Um terceiro array é criado

$$a1 = [1, 2, 3, 4]$$

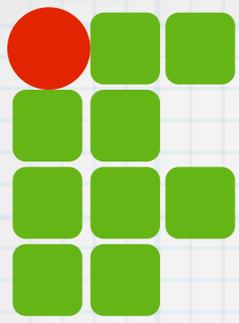
$$a2 = [5, 4, 3, 2, 1]$$

$$a3 = a1 + a2$$



$$a4 = a3 + [9]$$

$$a5 = a1 + [5, 6, 7, 8]$$



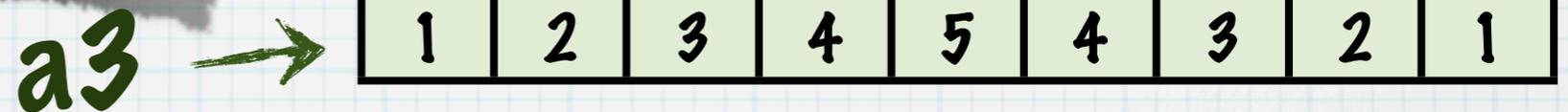
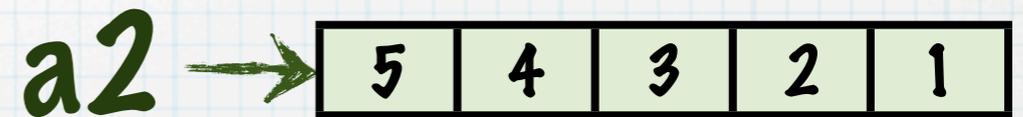
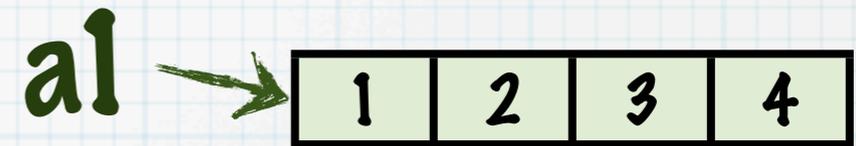
Soma de arrays

* Um terceiro array é criado

$$a1 = [1, 2, 3, 4]$$

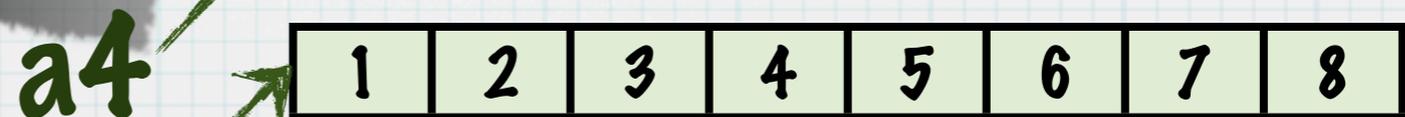
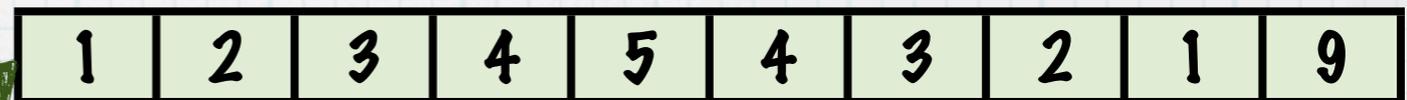
$$a2 = [5, 4, 3, 2, 1]$$

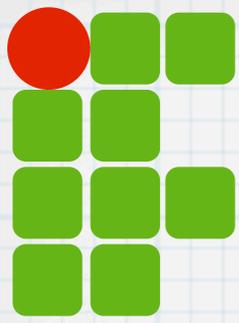
$$a3 = a1 + a2$$



$$a4 = a3 + [9]$$

$$a5 = a1 + [5, 6, 7, 8]$$



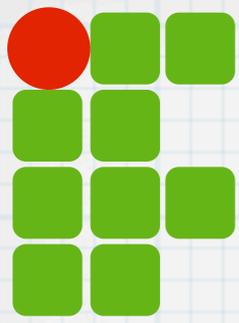


Strings

* Acesso a partes pelo conteúdo

```
msg = "Eu gosto de programar em Ruby!"  
msg["gosto de"] = "adoro"
```

Eu gosto de programar em Ruby!

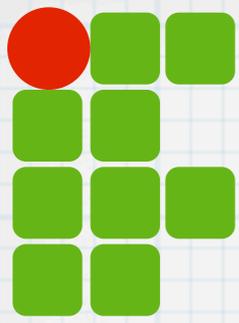


Strings

* Acesso a partes pelo conteúdo

```
msg = "Eu gosto de programar em Ruby!"  
msg["gosto de"] = "adoro"
```

Eu **gosto de** programar em Ruby!



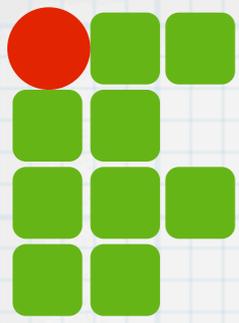
Strings

* Acesso a partes pelo conteúdo

```
msg = "Eu gosto de programar em Ruby!"  
msg["gosto de"] = "adoro"
```

Eu **gosto de** programar em Ruby!

Eu **adoro** programar em Ruby!



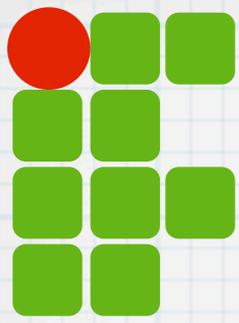
Strings

* Acesso a partes pelo conteúdo

```
msg = "Eu gosto de programar em Ruby!"  
msg["gosto de"] = "adoro"
```

Eu **gosto de** programar em Ruby!

Eu **adoro** programar em Ruby!

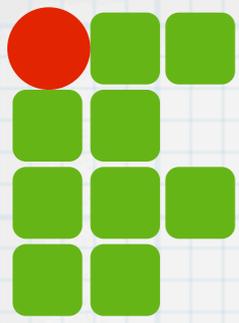


Métodos

* split

- * Divide a string em partes e coloca em um array
- * O espaço é o separador de strings

```
texto = "teste de como dividir uma string"  
palavras = texto.split
```



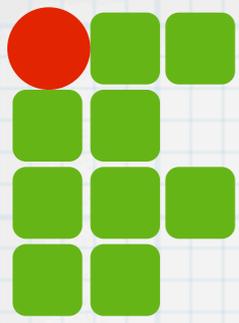
Métodos

* split

- * Divide a string em partes e coloca em um array
- * O espaço é o separador de strings

```
texto = "teste de como dividir uma string"  
palavras = texto.split
```

```
palavras = ["teste", "de", "como", "dividir", "uma", "string"]
```

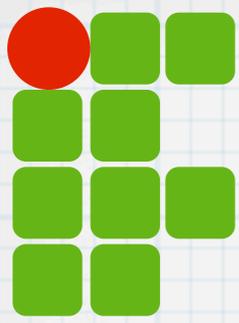


Métodos

- * Split aceita um parâmetro

- * 0 separador

```
texto = "teste de como dividir uma string"  
msg = texto.split("e")
```



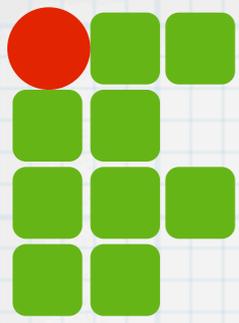
Métodos

* Split aceita um parâmetro

* 0 separador

```
texto = "teste de como dividir uma string"  
msg = texto.split("e")
```

```
msg = ["t", "st", " d", " como dividir uma string"]
```

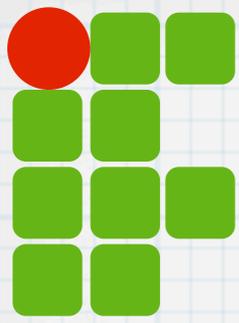


Métodos

* Split

* Separar a string em um array de caracteres

```
nome="Alfredo"  
a_nome=nome.split("")
```



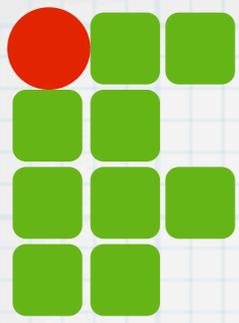
Métodos

* Split

* Separar a string em um array de caracteres

```
nome="Alfredo"  
a_nome=nome.split("")
```

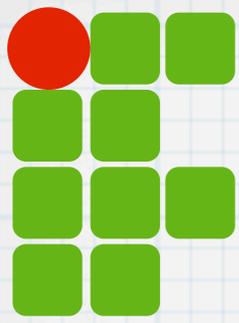
```
a_nome = ["A", "l", "f", "r", "e", "d", "o"]
```



Exemplo

* Contar palavras

```
txt = gets  
palavras = txt.split  
puts "O texto tem #{palavras.size} palavras."
```

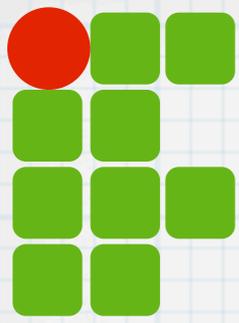


Exemplo

* Contar palavras

```
txt = gets
palavras = txt.split
puts "0 texto tem #{palavras.size} palavras."
```

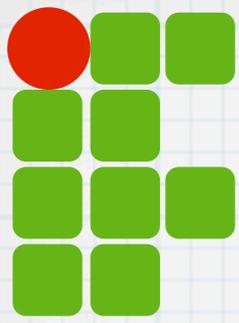
```
tmp — bash — 47x5
Jorgiano:tmp jorgiano$ ruby conta_palavras.rb
Isto e um teste de contagem de palavras
0 texto tem 8 palavras.
Jorgiano:tmp jorgiano$
```



Exemplo

* Contar frases

```
txt = gets  
frases = txt.split(".")  
puts "O texto tem #{frases.size} frases."
```

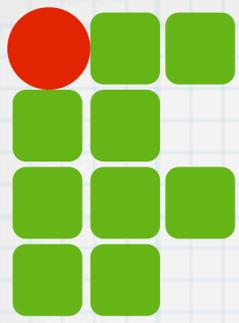


Exemplo

* Contar frases

```
txt = gets
frases = txt.split(".")
puts "O texto tem #{frases.size} frases."
```

```
tmp — bash — 50x5
Jorgiano:tmp jorgiano$ ruby conta_frases.rb
Oi. Isto e apenas um teste. Voce entendeu?
O texto tem 3 frases.
Jorgiano:tmp jorgiano$
```

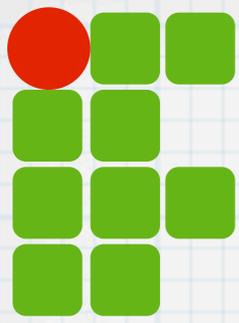


Métodos

* join

* Cria uma string a partir de um array

```
a1= ["Um", "monte", "de", "palavras"]  
txt=a1.join  
puts txt
```



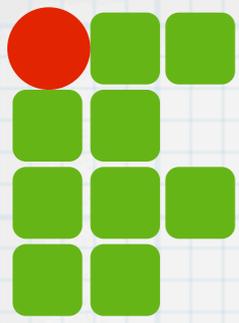
Métodos

* join

* Cria uma string a partir de um array

```
a1= ["Um", "monte", "de", "palavras"]  
txt=a1.join  
puts txt
```

```
tmp — bash — 50x5  
  
Jorgiano:tmp jorgiano$ ruby junta.rb  
Ummontedepalavras  
Jorgiano:tmp jorgiano$
```

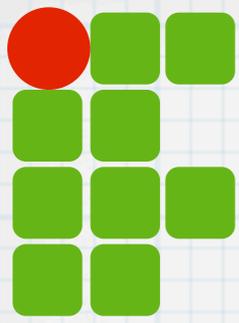


Métodos

* join

* Cria uma string a partir de um array

```
a1= ["Um", "monte", "de", "palavras"]  
txt=a1.join (" ")  
puts txt
```



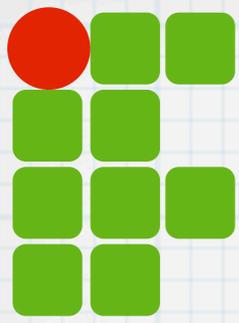
Métodos

* join

* Cria uma string a partir de um array

```
a1= ["Um", "monte", "de", "palavras"]  
txt=a1.join (" ")  
puts txt
```

```
tmp — bash — 50x5  
Jorgiano:tmp jorgiano$ ruby junta.rb  
Um monte de palavras  
Jorgiano:tmp jorgiano$ _
```

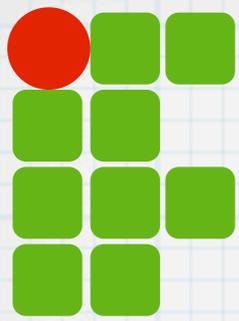


Métodos

* gsub

* Substitui um padrão na string

```
txt = "Teste de substituição de texto"  
txt1 = txt.gsub("te", "**")  
puts txt1
```



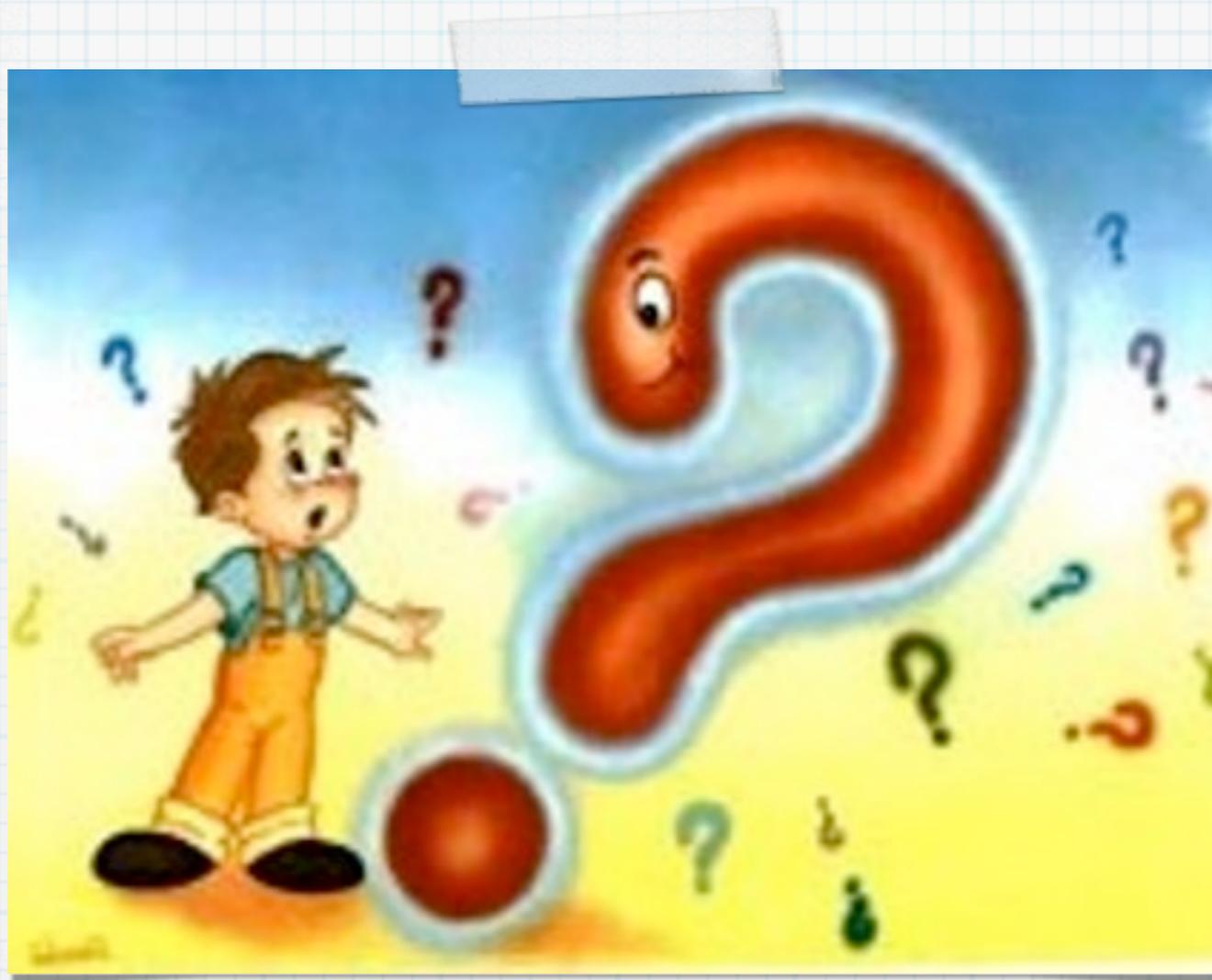
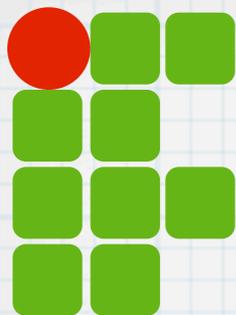
Métodos

* gsub

* Substitui um padrão na string

```
txt = "Teste de substituição de texto"  
txt1 = txt.gsub("te", "**")  
puts txt1
```

```
tmp — bash — 50x5  
Jorgiano:tmp jorgiano$ ruby substitui.rb  
Tes** de substituição de **xto  
Jorgiano:tmp jorgiano$
```



Dúvidas?