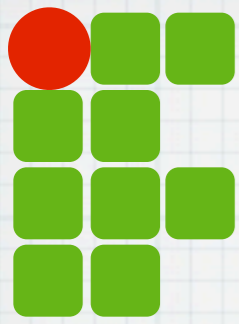


INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Programação de Computadores

Tipos e conversão

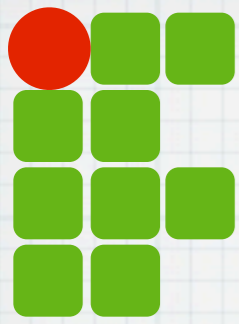
Copyright © 2012 IFRN



O que veremos hoje?

- * Tipo de dados
- * Operações sobre tipos
- * Conversão de tipos
- * Operações sobre números e strings
- * Mais strings
- * Exercícios





Relembrando tipos

* Inteiro

* 1234

* Real

* 1234.56

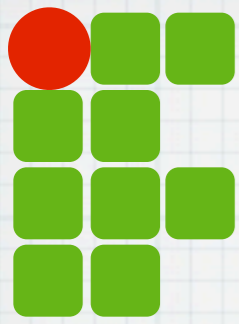
* String (texto)

* "Isto é uma string"

* 'Isto é outra string'



A B C D E F G H I J K L
M N O P Q R S T U V W
X Y Z À Á Ê Ë Ì abcdefghij
klmnopqrstuvwxyz à á â ã ä å ì õ
& 1 2 3 4 5 6 7 8 9 0 (\$ £ . , ! ?)



Operações

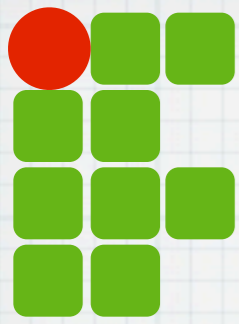
* Método



- * Operação aplicada a um dado de um determinado tipo
- * Ponto após o valor/variável seguido do nome do método (operação)
- * valor.método
- * variável.método



```
jorgiano — ruby — 40x9
irb(main):001:0> x = 10
=> 10
irb(main):002:0> 10.next
=> 11
irb(main):003:0> 10.pred
=> 9
irb(main):004:0> x.next
=> 11
irb(main):005:0> █
```

Conversão de tipos

* Podemos converter um valor de um tipo a outro

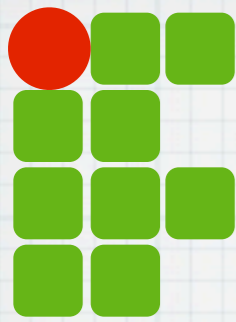
* Métodos

* `to_s`: converte em String

* `to_i`: converte em inteiro

* `to_f`: converte em float

```
jorgiano — ruby — 30x13
irb(main):001:0> x=10
=> 10
irb(main):002:0> y=x.to_s
=> "10"
irb(main):003:0> z=y.to_f
=> 10.0
irb(main):004:0> x
=> 10
irb(main):005:0> y
=> "10"
irb(main):006:0> z
=> 10.0
irb(main):007:0>
```

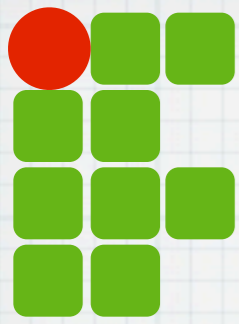
Conversão

* De String para inteiro/float

* Considera início do texto

No irb Não precisaria usar o print

```
jorgiano — ruby — 44x12
irb(main):001:0> nome="IFRN"
=> "IFRN"
irb(main):002:0> ifrn_inteiro=nome.to_i
=> 0
irb(main):003:0> print ifrn_inteiro
0=> nil
irb(main):004:0> nome="20 IFRN"
=> "20 IFRN"
irb(main):005:0> nome.to_i
=> 20
irb(main):006:0> █
```

String - métodos

* size

- * Retorna inteiro com quantidade de caracteres

* upcase

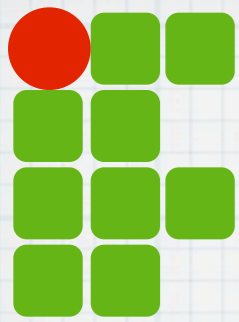
- * Retorna uma nova string com os caracteres convertidos em maiúsculos

* downcase

- * Retorna uma nova string com os caracteres convertidos em minúsculos

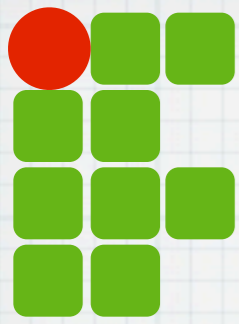
* capitalize

- * Retorna uma nova string com a inicial em maiúscula



String - métodos

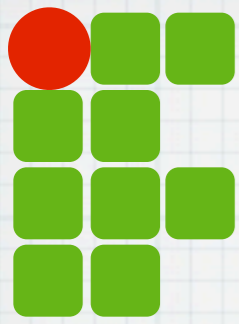
```
ruby — ruby — 51x17
irb(main):024:0* presidente="Dilma Roussef"
=> "Dilma Roussef"
irb(main):025:0> minusculo = presidente.downcase
=> "dilma roussef"
irb(main):026:0> maiusculo=presidente.upcase
=> "DILMA ROUSSEF"
irb(main):027:0> presidente
=> "Dilma Roussef"
irb(main):028:0> minusculo
=> "dilma roussef"
irb(main):029:0> maiusculo
=> "DILMA ROUSSEF"
irb(main):030:0> normal=minusculo.capitalize
=> "Dilma roussef"
irb(main):031:0> presidente.size
=> 13
irb(main):032:0>
```

Soma de strings

- * Cria uma nova string cujo conteúdo é a concatenação das strings

```
ruby — ruby — 37x9
irb(main):042:0* a = "IFRN"
=> "IFRN"
irb(main):043:0> b = "TADS"
=> "TADS"
irb(main):044:0> c = a+b
=> "IFRNTADS"
irb(main):045:0> c = a+" "+b
=> "IFRN TADS"
irb(main):046:0>
```

Operações

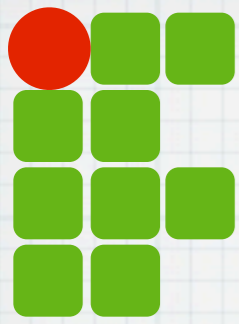
* Qual o resultado?

- * `x = "12" + "23"`
- * `x = "12 + 23"`
- * `x = 12 + 23`

* Não há soma/subtração entre números e strings

```
ruby — ruby — 45x6
irb(main):049:0> "a"+10
TypeError: can't convert Fixnum into String
    from (irb):49:in `+'
    from (irb):49
    from /usr/bin/irb2:12:in `<main>'
irb(main):050:0> █
```

- * Deve-se converter o número em string (`to_s`) ou a string em número (`to_i`)



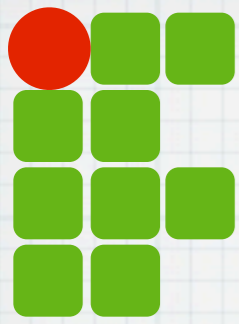
Operações

- * Multiplicação de string por número
- * Cria uma nova string com n vezes caracteres

```
x="Texto"*5
```

O valor de x é:

```
"TextoTextoTextoTextoTexto"
```

O método `class`

* Informa qual o tipo (classe) do valor/variável

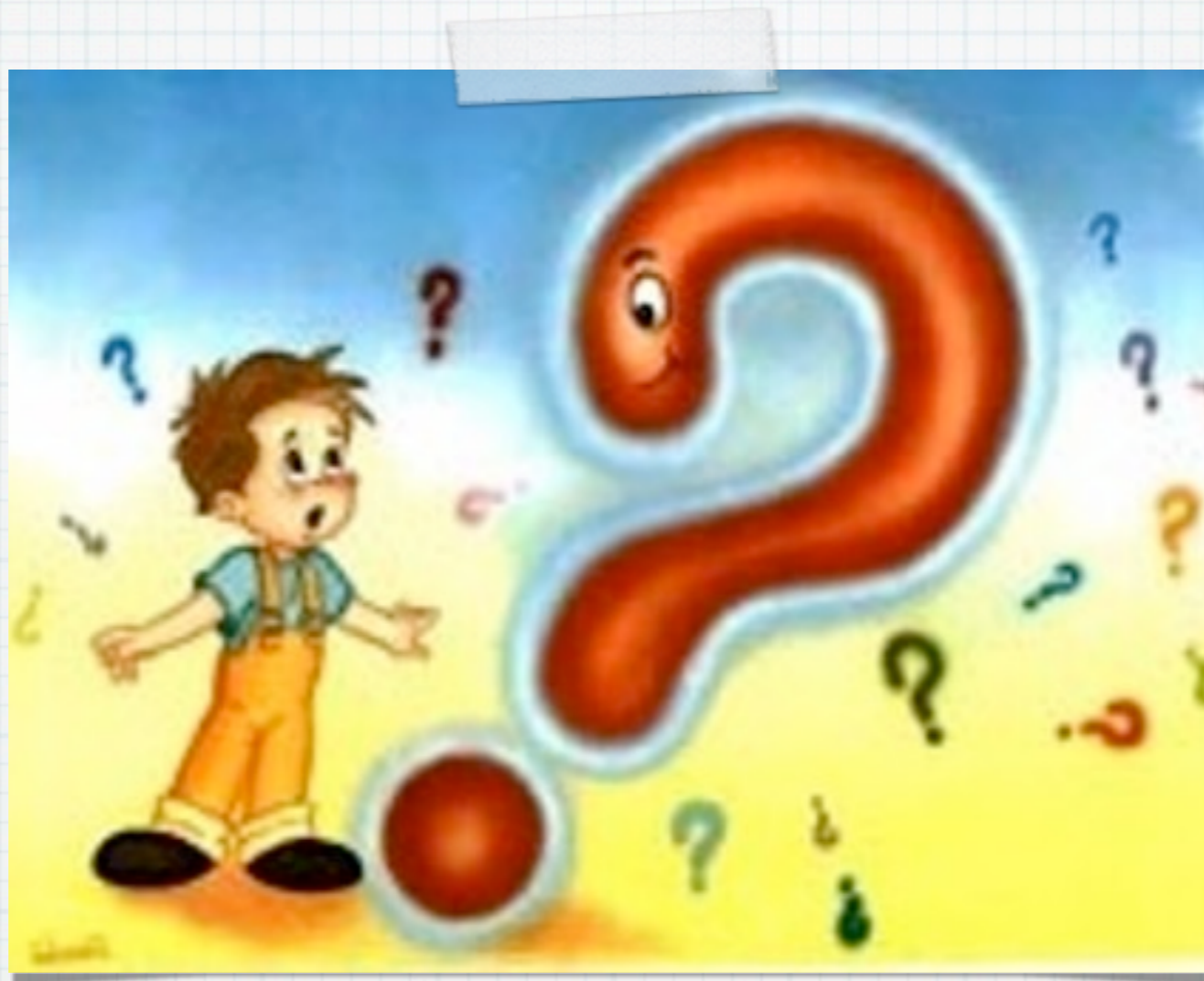
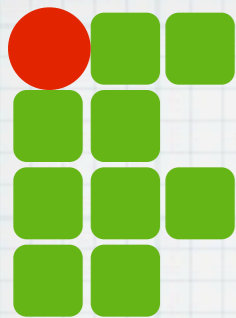
*

```
jorgiano — ruby — 40x11
irb(main):015:0* x = 1234
=> 1234
irb(main):016:0> x.class
=> Fixnum
irb(main):017:0> nome = "Programacao"
=> "Programacao"
irb(main):018:0> nome.class
=> String
irb(main):019:0> 10.3.class
=> Float
irb(main):020:0> █
```

Inteiro

Texto

Real



Dúvidas?