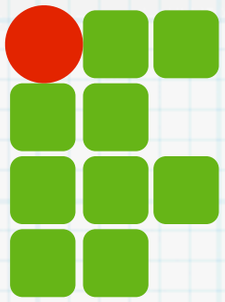


INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Programação de Computadores

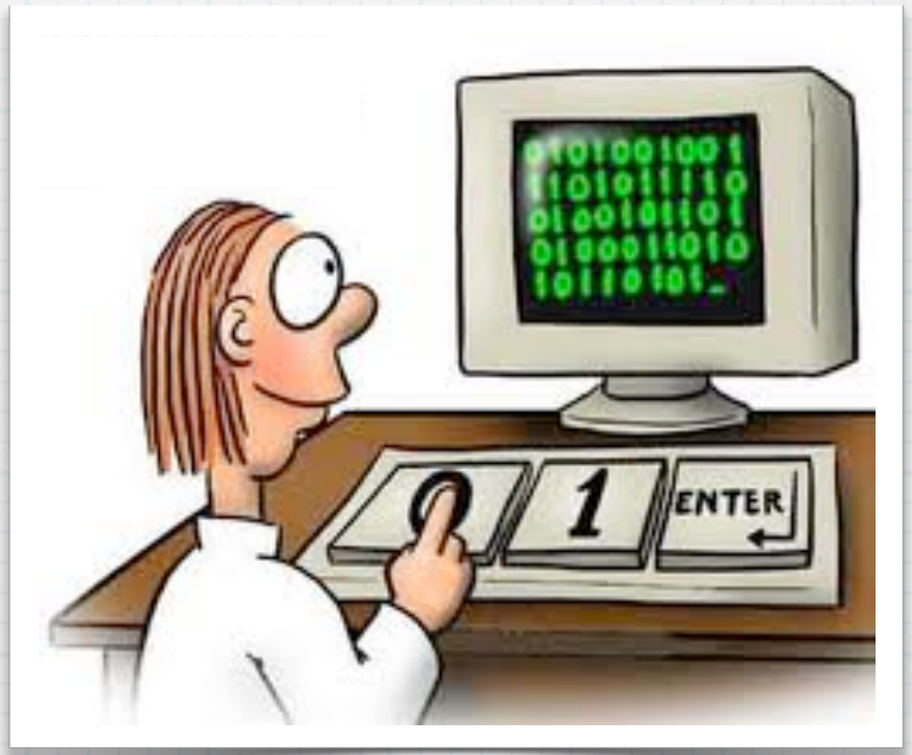
Dividir para conquistar
Funções/métodos

O que veremos hoje?

* Introdução
* Funções/métodos
  * Chamada a funções
* Definição
  * Parâmetros
  * Retorno
* Exercícios

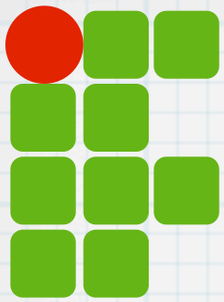
Introdução

* Comum mesmo trecho de código usado em vários lugares diferentes
  * Raiz quadrada, etc
* Alteração pode ser necessária
* Programador deve procurar o mesmo trecho em todo o programa
* Programas longos aumentam a complexidade
* Necessidade de gerenciar a complexidade



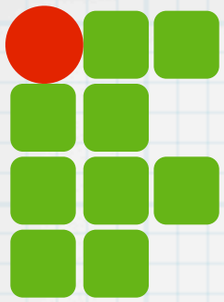
Função/método

* Uma função é um bloco de código, que possui um nome, e pode ser executado a partir de diferentes pontos do programa.
* Resultado do processamento é substituído no lugar chamado
* Parâmetros são passados para a função
  * Exemplo de função: raiz quadrada
  * Parâmetro: calcular a raiz de qual número???



Por que?

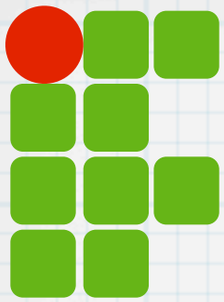
* Reduzir código
  * Mesma função usada em vários pontos diferentes
* Organizar (modularizar) programa
  * Blocos de códigos pequenos e bem definidos
* Melhorar legibilidade
  * Funções possuem nomes
* Gerenciar complexidade
  * Trabalhar com trechos pequenos facilita compreensão
  * Facilita a identificação e correção de erros



Exemplo de função

* Raiz quadrada
  * Math.sqrt (numero)

```
x = gets.to_f
y = Math.sqrt(x)
z = y * Math.sqrt(123)
w = Math.sqrt(y*2)+10
```



Uso de função

* Substitui o trecho inteiro de código onde se usaria o valor de retorno
* Faz parte da expressão
  * `x1 = (-b + Math.sqrt(delta))/2*a`

```
if (delta > 0) then
  raizesreais = 2
  raizdelta = Math.sqrt(delta)
  r1 =  (-b + raizdelta) / (2 * a)
  r2 =  (-b - raizdelta) / (2 * a)
elsif (delta == 0) then
  raizesreais = 1
  r1 = (0-b) / (2.0 * a)
else
  raizesreais = 0
end
```

Uso de função

* Substitui o trecho inteiro de código onde se usaria o valor de retorno
* Faz parte da expressão
  * `x1 = (-b + Math.sqrt(delta))/2*a`

```
if (delta > 0) then
  raizesreais = 2
  raizdelta = Math.sqrt(delta)
  r1 =  (-b + raizdelta) / (2 * a)
  r2 =  (-b - raizdelta) / (2 * a)
elsif (delta == 0) then
  raizesreais = 1
  r1 = (0-b) / (2.0 * a)
else
  raizesreais = 0
end
```

Este trecho de código é responsável apenas pelo cálculo das raízes de uma equação.

Definição de função

```
def nome(lista de parametros)
  inst01
  inst02
  ...
  return valor
end
```

Definição de função

Definição de uma nova função/método

```
def nome(lista de parametros)
  inst01
  inst02
  ...
  return valor
end
```

Definição de função

Definição de uma nova função/método

Nome da função

```
def nome(lista de parametros)
  inst01
  inst02
  ...
  return valor
end
```

Definição de função

Definição de uma nova função/método

Nome da função

```
def nome(lista de parametros)
  inst01
  inst02
  ...
  return valor
end
```

lista de parâmetros, separados por vírgula

Definição de função

Definição de uma nova função/método

Nome da função

```
def nome(lista de parametros)
    inst01
    inst02
    ...
    return valor
end
```

trecho de código que implementa a função

lista de parâmetros, separados por vírgula

Definição de função

Definição de uma nova função/método

Nome da função

```
def nome(lista de parametros)
    inst01
    inst02
    ...
    return valor
end
```

trecho de código que implementa a função

valor de retorno da função

lista de parâmetros, separados por vírgula

Exemplo

Maior de dois números

```
def maior (a,b)
  x = a
  if (b>a) then
    x = b
  end
  return x
end
```

Menor de dois números

```
def menor (a,b)
  x = a
  if (b<a) then
    x = b
  end
  return x
end
```

Uso

```
x = gets.to_i
y = gets.to_i
z = maior(x,y)
w = menor(x,y)
...
```

Exemplo

* Fatorial de um número

```
def fatorial(x)
   y = 1
   for i in 1..x do
      y = y * i
   end
   return y
end
```

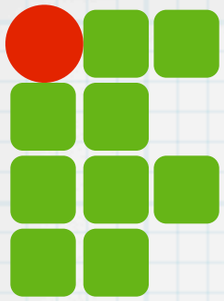
Exemplo

* Fatorial de um número

```
def fatorial(x)
  y = 1
  for i in 1..x do
    y = y * i
  end
  return y
end
```

Programa completo

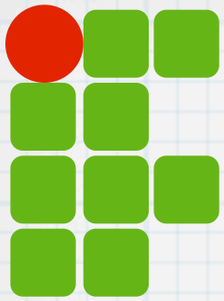
```
def fatorial(x)
  y = 1
  for i in 1..x do
    y = y * i
  end
  return y
end

numero = gets.to_i
f = fatorial(numero)
puts f
```



Exemplo

* índice do menor elemento de um array

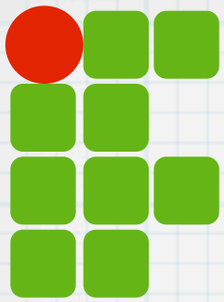
```
def indice_menor(a)
  imenor = -1
  if (a.class == Array) then
    imenor = 0
    for i in 1..(a.size-1) do
      if (a[i]<a[imenor])
        imenor = i
      end
    end
  end
  return imenor
end
```



Exemplo

* Função que retorna um array com a soma das linhas de uma matriz

```
def soma_linhas(m)
  l=[]
    for i in 0..(m.size-1) do
      l[i]=0
      for j in 0..(m[i].size-1) do
        l[i] = l[i]+m[i][j]
      end
    end
  return l
end

m = 3.times.map do
      3.times.map do
         gets.to_i
      end
    end
s = soma_linhas(m)
```



Conclusão

* Funções/métodos ajudam a melhor organizar o programa
* Devem ser relativamente pequenas
  * Melhor dividir uma função extensa em várias funções pequenas
* Precisam ser definidas antes de serem usadas.
  * A chamada a uma função só pode ser realizada depois do bloco que a define (`def ... end`)
  * Pode ser em arquivos separados (veremos mais adiante)

Dúvidas?