

### 1º Semestre









## Funções e Procedimentos



- > Também chamados de subalgoritmos
  - São blocos de instruções que realizam tarefas específicas.
  - Ao invés de escrever um código grande, pode-se escrever vários algoritmos menores
    - Modularização
  - Evitam a repetição do código-fonte.
  - São carregados uma vez e podem ser executados quantas vezes for necessário.



## Funções e Procedimentos Cont...

- Reduzem o tamanho do algoritmo.
- > Facilitam a compreensão e visualização do algoritmo.
- Podem ser usadas para economizar espaço e tempo de programação já que podem ser usadas várias vezes num mesmo algoritmo.
- Podem ser:
  - > Funções: que retornam algum valor
  - Procedimentos: que não retorna nada



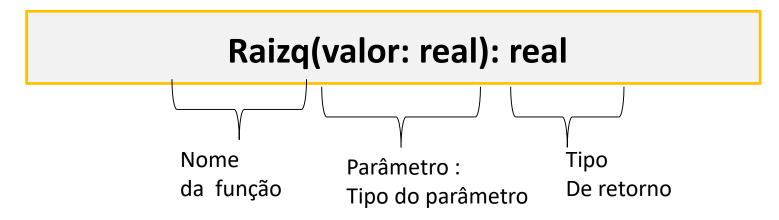




- Uma função é um bloco de instruções que tem por objectivo retornar um valor ou uma informação.
- A chamada de uma função é feita através da citação do seu nome seguido, opcionalmente, dos seus argumentos iniciais entre parêntesis.
- As funções podem ser **predefinidas pela linguagem**(funções que já fazem parte da linguagem e que o programador apenas têm de chamar) ou **criadas pelo programador** (funções que o próprio programador é que têm de declarar e implementar.
- > As **funções criadas pelo programador** são aquelas que queremos a aprender a **declarar**, **implementar** e a **chamar**.



- > A linguagem do Visualg possui diversas funções predefinidas que podem ser usadas na construção de algoritmos
- Exemplo 1: Criar um algoritmo que calcula o valor da raiz quadrada de um número.
  - O Visualg já têm uma função predefinida que nos pode a ajudar a resolver o problema rapidamente.
    - Essa função recebe como **parâmetro** uma variável (do tipo real) e **retorna** um valor (também do tipo real) referente a raiz quadrada do número
- > Assinatura da função para o cálculo da Raiz Quadrada





Como utilizar a função para o cálculo da Raiz Quadrada

```
1 Algoritmo "Função Predef. Raiz Quadrada"
 2 Var
      num, raiz: real
 4 Inicio
      Escreva ("Introduza um número: ")
                                                      Chamada da função passando
 6
      Leia (num)
                                                      a variável num (do tipo real)
                                                      como parâmetro da função
      raiz <- Raizq(num)</pre>
                                                      Raizq
 9
10
      Escreva ("A raiz quadrada de ", num, " é ", raiz)
11 Fimalgoritmo
12
13
```

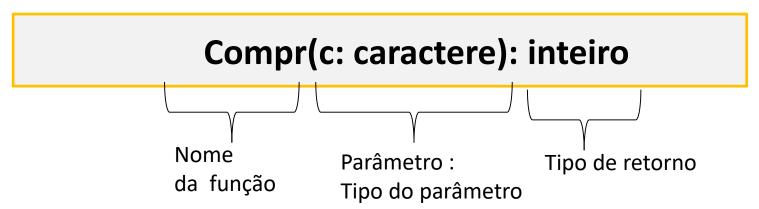


Como utilizar a função para o cálculo da Raiz Quadrada

```
1 Algoritmo "Função Predef. Raiz Quadrada"
 2 Var
      num, raiz: real
                                                     Atribuição à variável raiz
 4 Inicio
                                                    (do tipo real) o retorno da
      Escreva ("Introduza um número: ")
                                                    função Raizq
      Leia (num)
 6
      raiz <- Raizq (num)
 8
 9
      Escreva ("A raiz quadrada de ", num, " é ", raiz)
10
11 Fimalgoritmo
12
13
```



- Exemplo 2: Criar um algoritmo que, dada uma palavra qualquer, informa ao utilizador quantas letras essa palavra possui.
  - O Visualg já têm uma função predefinida que recebe como parâmetro uma variável (tipo caractere) e retorna um valor (inteiro) correspondente à quantidade de caracteres existentes.
- > Assinatura da função para contar caracteres de uma palavra





# Funções predefinidas

Como utilizar a função para contar caracteres de uma palavra

```
2 Algoritmo "Função Predef. Conta Caracteres"
 3 Var
    c: caracter
    qtd: inteiro
                                                   Chamada da função Compr
 6 Inicio
                                                  passando a variável c como
     Escreva ("Introduza uma palavra: ")
                                                  parâmetro
 8
     Leia(c)
 9
10
     qtd <- Compr(c)
11
12
     Escreva ("A palavra têm", qtd, " caracteres")
13 Fimalgoritmo
14
```



- Lista de algumas funções predefinidas do Visualg
  - Funções numéricas, algébricas e trigonométricas
    - Abs( expressão) Retorna o valor absoluto de uma expressão do tipo inteiro ou real. Equivale a | expressão | na álgebra.
    - ArcCos( expressão) Retorna o ângulo (em radianos) cujo co-seno é representado por expressão.
    - ArcSen( expressão) Retorna o ângulo (em radianos) cujo seno é representado por expressão.
    - ArcTan( expressão) Retorna o ângulo (em radianos) cuja tangente é representada por expressão.
    - **Cos( expressão)** Retorna o co-seno do ângulo (em radianos) representado por expressão.



# Funções Funções predefinidas Cont...

- Lista de algumas funções predefinidas do Visualg
  - Funções para manipular caracteres
    - Asc (s : caracter) : Retorna um inteiro com o código ASCII do primeiro caracter da expressão.
    - Carac (c : inteiro) : Retorna o caracter cujo código ASCII corresponde à expressão.
    - **Caracpnum (c : caracter)** : Retorna o inteiro ou real representado pela expressão. Corresponde a StrToInt() ou StrToFloat() do Delphi, Val() do Basic ou Clipper, etc.
    - Compr (c : caracter) : Retorna um inteiro contendo o comprimento (quantidade de caracteres) da expressão.
    - Copia (c : caracter ; p, n : inteiro) : Retorna um valor do tipo caracter contendo uma cópia parcial da expressão, a partir do caracter p, contendo n caracteres. Os caracteres são numerados da esquerda para a direita, começando de 1. Corresponde a Copy() do Delphi, Mid\$() do Basic ou Substr() do Clipper.



### Criação de Funções

- A criação de uma função deve ser realizada dentro da secção das variáveis
- As funções retornam sempre apenas um valor para o algoritmo que o chamou ou invocou
- As funções possuem um tipo de retorno associado
- Uma função pode ter 0, 1 ou mais parâmetros
- Sintaxe



### Criação de Funções Sintaxe da função



### Criação de Funções

Com as funções / procedimentos surgem duas novas classificações para as variáveis

#### Variáveis Locais

- Declaradas dentro dos subalgoritmos(funções / procedimentos)
- Podem ser utilizadas APENAS dentro das funções / procedimentos.
- O algoritmo que chamou a função / procedimento não tem acesso a estas variáveis.

### Variáveis Globais

- São variáveis declaradas na secção var do algoritmo .
  - Qualquer função / procedimento pode alterar o valor ou utiliza-la durante o seu processamento.





**Exemplo 1:** Criar uma função para calcular o dobro de um número passado como parâmetro

```
1 Algoritmo "Dobro"
 2
 3 Var
    // Seção de Declarações das variáveis
 5
     num, res: real
 7 funcao Dobro (valor: real): real
 8 Var
   dobroValor: real
10
11 Inicio
12
     dobroValor <- valor * 2
13
14
     retorne dobroValor
15
16 fimfuncao
17
18 Inicio
19
   // Seção de Comandos, procedimento, funções, operadore
     Escreva ("Digite um numero: ")
20
21
     Leia (num)
22
23
     res <- Dobro (num)
     Escreva ("O dobro de ", num, " = ", res)
24
25
26 Fimalgoritmo
```





Exemplo 1: Criar uma função para calcular o dobro de um número passado como parâmetro

```
1 Algoritmo "Dobro"
 3 Var
      // Secão de Declarações das variáveis
                                               Variáveis Globais
     num, res: real
 7 funcao Dobro (valor: real): real
 8 Var
     dobroValor: real
10
11 Inicio
12
                                                      Declaração da função
13
     dobroValor <- valor * 2
     retorne dobroValor
15
16 fimfuncao
18 Inicio
19
                                            funções, operadore
     // Seção de Comandos, procedimento,
20
     Escreva ("Digite um numero: ")
21
     Leia (num)
                                                          Comandos
22
23
     res <- Dobro (num)
     Escreva ("O dobro de ", num, " = ", res)
24
25
26 Fimalgoritmo
```



### Criação de Funções **Notas**

- Os algoritmos podem possuir várias funções.
- Uma função pode não ter parâmetros.
- As funções podem possuir mais de um parâmetro:
  - Parâmetros de um mesmo tipo são separados por vírgula
  - Parâmetros de tipos diferentes são separados por ponto e vírgula
- Funções podem ser chamadas várias vezes durante a execução de um algoritmo













## Funções **Exercícios**

- Criar uma função que receba três valores reais como parâmetro e retorne a média desses valores.
- Crie uma função que devolve a soma de dois números.
- Escreva uma função que recebe as 4 notas de um aluno por parâmetro e uma letra. Se a letra for A, a função calcula a média aritmética das notas do aluno, se for P, a sua média ponderada (pesos: 2, 3, 4 e 6).
- Crie uma função que leia um número não determinado de valores positivos e calcule a soma desses números. A função deve parar de ler números quando um número negativo for digitado e retornar a soma dos números lidos.
- Crie uma função que recebe por parâmetro o raio de uma esfera e calcula o seu volume (v = 4/3 \* pi \* raio3).
- Crie uma função que receba por parâmetro um número inteiro e verifique se o número é positivo ou negativo. Retorne um valor lógico (verdadeiro ou falso).