

53 45 52 45 49 20 46 49 45 4c 20
41 4f 53 20 50 52 45 43 45 49 54
4f 53 20 44 41 20 48 4f 4e 52 41
20 45 20 44 41 20 43 49 c3 8a 4e
43 49 41 2c 20 50 52 4f 4d 4f 56
45 4e 44 4f 20 4f 20 55 53 4f 20
45 20 4f 20 44 45 53 45 4e 56 4f
4c 56 49 4d 45 4e 54 4f 20 44 41
20 49 4e 46 4f 52 4d c3 81 54 49
43 41 20 45 4d 20 42 45 4e 45 46
c3 8d 43 49 4f 20 44 4f 20 43 49
44 41 44 c3 83 4f 20 45 20 44 41
20 53 4f 43 49 45 44 41 44 45 2e

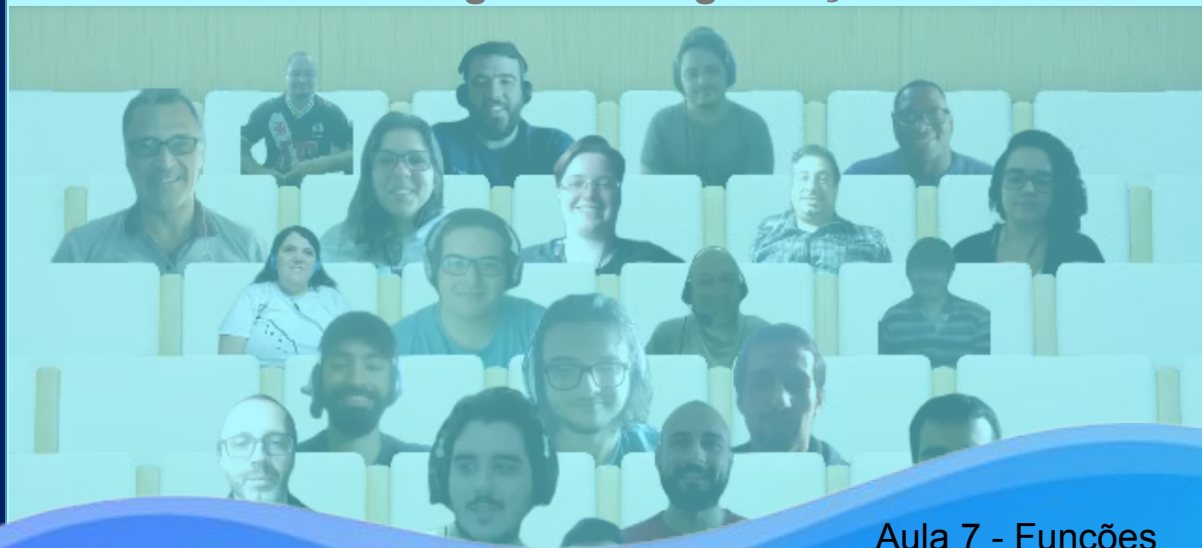
RESIDÊNCIA DE SOFTWARE

**CAPACITAR
TREINAR
EMPREGAR**

TRANSFORMAR



Lógica de Programação



Aula 7 - Funções
Roni Schanuel
25-03-2022

Funções

- Definição : Sequência de instruções executadas somente quando chamadas por um programa em execução
 - Devem executar **uma tarefa** específica
 - Um programa **pode conter diversas funções**, além da função principal **início()** , que é **obrigatória**
 - As funções executam **somente** quando chamadas à partir da função início()
 - Após a execução, o fluxo retorna ao ponto **imediatamente após** o da chamada da função
 - Uma função pode (ou não) **retornar um valor** ao bloco que a chamou
 - Uma função pode (ou não) **necessitar de um ou mais argumentos** ao ser chamada

Vamos olhar uma etapa de cada vez ...



Funções

Com as funções podemos criar pequenos trechos de programas (blocos) que podem ser reutilizados facilitando o desenvolvimento.

funcao inicio()

É a função raiz do Portugol Studio sendo obrigatório o seu uso para o funcionamento dos algoritmos. Outras funções e procedimentos, podem ser utilizados, mas a funcao inicio() sempre deverá está presente.

Sintaxe:

```
funcao nomeFuncao(){  
}
```

Mais alguns exemplos - Repetição de código

```
programa {  
    funcao inicio(){  
        inteiro i  
        para(i=0; i<20; i++)  
            escreva("**")  
        escreva("\n")  
        escreva("Numeros entre 1 e 5\n")  
        para(i=0; i<20; i++)  
            escreva("**")  
        para(i=1; i<=5; i++)  
            escreva(i, "\n")  
        para(i=0; i<20; i++)  
            escreva("**")  
        escreva("\n")  
    }  
}
```

Note o código repetido. Se
tivermos que consertar,
teremos que fazer o mesmo
ajuste várias vezes



Saída:

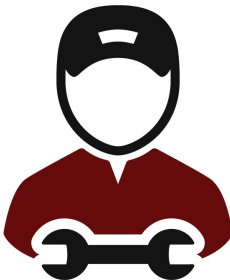
Numeros entre 1 e 5

1
2
3
4
5

Mais alguns exemplos - Repetição de código

```
programa {  
    funcao inicio(){  
        inteiro i  
        escreve_linha()  
        escreva("Numeros entre 1 e 5\n")  
        escreve_linha()  
        para(i=1; i<=5; i++)  
            escreva(i, "\n")  
        escreve_linha()  
    }  
    funcao escreve_linha(){  
        para(i=0; i<20; i++)  
            escreva(" ")  
        escreva("\n")  
    }  
}
```

Observe a diferença ao
encapsularmos esse código
repetido em uma função =D



Saída:

Numeros entre 1 e 5

1
2
3
4
5

Procedimentos

São funções que não possuem retorno realizando apenas a execução do que está dentro do bloco do código.

```
programa
{
    funcao inicio()
    {
        mensagem()
    }

    funcao mensagem(){
        escreva("*****\n")
        escreva("**Residência 2022**\n")
        escreva("*****")
    }
}
```

chamada do procedimento

E se fizermos mais uma chamada ao procedimento?

Procedimentos com argumentos

Passando parâmetros ou argumentos em procedimentos.

```
programa
{

    funcao inicio()
    {
        mensagem(2022)

    }

    funcao mensagem(inteiro ano){
        escreva("*****\n")
        escreva("***Residência "+ano+"**\n")
        escreva("*****")
    }

}
```

chamada passando o argumento

Podemos ter um procedimento chamando outro por exemplo

```
programa
{

    funcao inicio()
    {
        entrada()
    }

    funcao entrada(){
        inteiro ano
        escreva("Digite o ano do curso:")
        leia (ano)
        mensagem(ano)
    }

    funcao mensagem(inteiro ano){
        escreva("*****\n")
        escreva("***Residência "+ano+"**\n")
        escreva("*****")
    }

}
```

Procedimentos com passagem de parâmetros por valor

Passando mais de um parâmetro em procedimentos.

```
programa
{
    funcao inicio()
    {
        inteiro a=10, b= 20
        soma(a,b)
    }

    funcao soma(inteiro x, inteiro y){
        escreva("Soma:", x + y)
    }
}
```


Função

Toda função tem um tipo de retorno. Deve ser utilizado a palavra retorne no bloco da função.

Sintaxe:

funcao tipoDeRetorno nomeDaFuncao()

Exemplo sem parâmetro:

função cadeia texto()

Exemplo com parâmetro:

função real calculaMedia(real a, real b)

Exemplo:

O algoritmo abaixo faz a leitura de duas notas, essas notas são passadas como parâmetro por valor para a função **calculaMedia** e após o cálculo e retornada a média.

```
programa
{
    funcao inicio()
    {
        real n1, n2
        escreva("Digite a nota1:")
        leia(n1)

        escreva("Digite a nota2:")
        leia(n2)

        escreva(calculaMedia(n1,n2))
    }

    funcao real calculaMedia(real a, real b){
        retorne (a + b)/2
    }
}
```

Exemplo de função com passagem de parâmetro por valor

Quando chamamos a função **incrementa** estamos passando o valor da variável **a** para a variável **i** da função **incrementa**. Quando chamamos a função o valor original da variável **a** não é alterado.

Qual valor será impresso
no exemplo abaixo?

```
programa
{
    funcao inicio()
    {
        inteiro a = 100

        incrementa(a)
        escreva("O valor é:", incrementa(a), "\n")
        escreva("O valor é:", a)
    }

    funcao inteiro incrementa(inteiro i){
        i = i + 10
        retorne i
    }
}
```

Memória	
a	100
i	110 (100+10)

Exemplo de função com passagem de parâmetro por referência

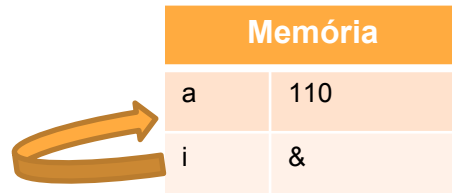
Na passagem de parâmetros por referência a variável aponta para a variável de origem. As alterações na função afetam a variável de origem. Para usar a passagem de parâmetros por referência devemos utilizar o **&** antes do nome da variável do parâmetro.

Qual valor será impresso
no exemplo abaixo?

```
programa
{
    funcao inicio()
    {
        inteiro a = 100

        incrementa(a)
        escreva("O valor é:", incrementa(a), "\n")
        escreva("O valor é:", a)
    }

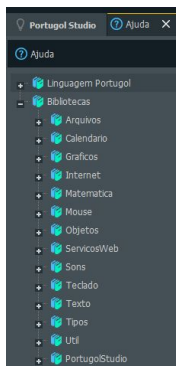
    funcao inteiro incrementa(inteiro &i){
        i = i + 10
        retorne i
    }
}
```



A variável **i** aponta para o endereço de **a**

Biblioteca de funções

São funções disponibilizadas por uma linguagem para facilitar o trabalho na criação de algoritmos. No menu ajuda do Portugol temos todas as funções disponíveis.



Para usar uma função precisamos importar para o nosso algoritmo. Vamos importar a biblioteca **Util** e utilizar a função **sorteia, aguarde e caixa_alta**.

```
programa
{
    inclui biblioteca Util
    inclui biblioteca Texto
    funcao inicio()
    {
        escreva (Util.sorteia(1,100))
        Util.aguarde(3000)
        escreva(Texto.caixa_alta("\nCurso de Programação"))
    }
}
```

O que será impresso na linha em destaque abaixo?

```
escreva(Texto.numero_caracteres(Texto.caixa_alta("\nCurso de Programação")))
```

Biblioteca de funções

Exemplo:

```
programa
{
    inclui biblioteca Matematica --> mat

    funcao inicio()
    {
        real base, expoente, resultado
        escreva("Digite a base: ")
        leia(base)

        escreva("Digite o expoente:")
        leia(expoente)

        resultado = mat.potencia(base, expoente)
        escreva( base, " elevado a ", expoente, " = ", resultado)

        escreva("\nMaior número:", mat.maior_numero(base, expoente))
    }
}
```

potencia é um exemplo de uma função existente na biblioteca de funções do Portugol

Biblioteca de funções

Exemplo:

```
programa
{
    inclui biblioteca Util --> u

    funcao inicio()
    {
        inteiro vetor[10]

        para(inteiro i=0; i < 10; i++){
            vetor[i] = u.sorteia(100,1000)
        }

        para(inteiro i=0; i < 10; i++){
            escreva(i + "-" +vetor[i], "\n")
        }

    }
}
```

Utilizando a função **sorteia** da biblioteca **Util**