

# Fundamentos de Programação

Prof. Márcio Miguel Gomes



JESUÍTAS BRASIL



Somos infinitas possibilidades

# Funções

- Também conhecidas como métodos ou procedimentos
- Organização do código fonte em blocos padronizados
- Reaproveitamento de códigos executados de forma idêntica em diversas partes do programa

# Funções

- Facilita a inclusão de novas funcionalidades, manutenção e a correção de falhas
- Passagem de parâmetros é opcional
- Retorno de resultados é opcional

# Estrutura de uma Função

```
def nome_da_funcao(lista de parametros):  
    # Uma ou mais instruções  
    # Utilizando expressões e operadores  
    # Utilizando estruturas de controle  
    # Tanto de seleção quanto repetição  
    return valor
```

# Função básica

```
def minha_funcao():  
    print('\nEstou usando funções!\n')
```

Declaração e implementação da função. Ela só será executada quando for chamada

```
# Uma ou mais instruções antes da chamada da função  
minha_funcao()
```

```
# Uma ou mais instruções após a chamada da função
```

Chamada da função. Ela é executada nesse momento, podendo ser chamada diversas vezes

# Função com parâmetros

```
def minha_funcao(par1, par2):  
    print('\nParâmetros recebidos:')  
    print('par1 =', par1)  
    print('par2 =', par2)  
    print()
```

Parâmetros são dados enviados pelo chamador para dentro da função, e podem ser usados livremente dentro dela

```
if __name__ == '__main__':  
    # Uma ou mais instruções antes da chamada da função  
    minha_funcao(1, 2.2)  
    minha_funcao(par2='B', par1='A')  
    # Uma ou mais instruções após a chamada da função
```

No Python, é muito comum colocar o código principal em um “bloco main”

Por padrão, os parâmetros no Python são passados por ordem de declaração, mas também podem ser passados por nome

# Função com retorno

```
def minha_funcao(par1, par2):  
    soma = par1 + par2  
    return soma
```

A função soma os valores dos dois parâmetros e retorna o resultado para o chamador

```
if __name__ == '__main__':  
    v1 = float(input('\nInforme o primeiro número: '))  
    v2 = float(input('Informe o segundo número: '))  
    retorno = minha_funcao(v1, v2)  
    print('\nRetorno da função:', retorno, '\n')
```

Na chamada da função, são passados os dois valores e o retorno é armazenado em uma variável

# Programas com Menus

- Exibição de uma lista de opções numeradas
- Uma variável para armazenar a escolha do usuário
- Laço para permanecer no programa enquanto a escolha for diferente da opção “Sair”
- Estrutura de seleção para identificar a escolha informada
- Um procedimento por escolha



# Menu

```
import os

# Menu com as opções de escolha
def menu():
    os.system('cls' if os.name == 'nt' else 'clear') # Limpa a tela - Win/Linux
    print('\n...: Desenvolvendo programas com menus :...\n')
    print('1 - Item 1')
    print('2 - Item 2')
    print('3 - Item 3')
    print('9 - Sair\n')
    item = input('Escolha uma opção: ')
    return item
```

# Opções do Menu

```
# Uma função para cada item do menu
def opcao_1():
    print('\nOpção escolhida: 1\n')

def opcao_2():
    print('\nOpção escolhida: 2\n')

def opcao_3():
    print('\nOpção escolhida: 3\n')
```

# Processamento do Menu

```
# Processamento do menu e chamada das funções
escolha = '0'

while(escolha != '9'):
    escolha = menu()
    if escolha == '1':
        opcao_1()
    elif escolha == '2':
        opcao_2()
    elif escolha == '3':
        opcao_3()
    elif escolha == '9':
        print('\nSaindo...\n')
    else:
        print('\nOpção desconhecida!\n')

    input('Pressione ENTER para continuar')
```

# Atividade

1. Calculadora básica: Solicite ao usuário um número, um operador e em seguida outro número, por exemplo:  $1 + 1$   $2.3 * 2$   $5 - 2.7$   $9.3 / 2.4$ 
  - Interprete a expressão e dê o resultado correto;
  - Cada operação matemática deve ser uma função que retorna o resultado da operação para o chamador;
  - A impressão do resultado deve ser feita a partir do chamador;

# Atividade

2. Declare uma função chamada “ehPositivo” que recebe um número como parâmetro. Deverá retornar *True* caso o parâmetro seja positivo ou zero e retornar *False* caso o número seja negativo. Crie um programa que chama a função e imprime o resultado a partir do chamador;
3. Faça um procedimento chamado “raizes”, que recebe 3 parâmetros e calcula as raízes conforme a fórmula de Baskara. O procedimento deverá utilizar a função “ehPositivo” para verificar se o delta da fórmula de Baskara é positivo, e imprimir na tela as raízes calculadas ou informar que não existem raízes. Para testar o procedimento, faça a leitura dos 3 parâmetros no “main” e em seguida chame o procedimento passando os 3 parâmetros;

# Atividade

4. Faça um programa que solicite ao usuário informar uma hora, minuto e segundo no formato “hh:mm:ss”. Crie uma função chamada “horaParaFloat” que recebe esses 3 parâmetros separadamente. Essa função deverá retornar um número float representando as horas, minutos e segundos como um número fracionário. Ex: “01:15:30” = 1,2583 ou “13:20:15” = 13,3375. Imprima o número fracionário a partir do “main”;
5. Altere o programa anterior e adicione um procedimento chamado “floatParaHora”, que recebe como parâmetro um número fracionário e imprime na tela as horas, minutos e segundos correspondentes, no formato “hh:mm:ss”;

# Atividade

6. Desenvolva um programa que pergunte ao usuário quanto ele ganha por hora (opção 1 no menu) e o número de horas trabalhadas no mês (opção 2 no menu). Calcule e mostre o detalhamento do seu contracheque no referido mês, conforme o seguinte menu:
- Salário bruto (opção 3);
  - Quanto pagou ao INSS (opção 4) (11%);
  - Quanto pagou ao sindicato (opção 5) (2%);
  - O salário líquido (opção 6);
  - Sair (opção 7);
- Caso não tenha sido informado o valor recebido por hora ou a quantidade de horas trabalhadas, deve gerar uma mensagem avisando que não há dados suficientes;