

# Aula 13

## Funções – Parte 1 – Introdução



# Algoritmos e Estrutura de Dados I

1º Semestre – CDN



Prof. Dr. Dilermando Piva Jr.









# Conteúdo Programático - Planejamento

Conteúdo Programático		
Semana	Data	Temas/Atividades
1	12/08	Acolhimento e Boas-vindas! Introdução a Disciplina. Formas de Avaliação e Percurso Pedagógico.
2	19/08	Pensamento Computacional. O que é qual sua importância para Ciência de Dados
3	26/08	Primeiro Programa – Variáveis, Tipos de Dados e Saída em Python
4	30/08	<i>/reposição/ Introdução a Computação e representação da informação –História e evolução da computação. A informação e sua representação. Conversão entre bases.</i>
5	02/09	Operadores, Cálculo Simples e Entrada de Dados. Estruturas sequenciais.
6	09/09	Tomando Decisões: Estrutura Condicional (if/else)
7	16/09	Repetição de Ações: Estrutura de Repetição – Introdução aos loops (for e while)
8	23/09	Prática Integrada – Construindo um Jogo Simples (Adivinhação) – Pj1
9	30/09	Estruturas de Dados Parte 1 – listas e sequências (e strings)
10	07/10	Estruturas de Dados Parte 2 – Dicionários e Dados Estruturados
11	14/10	Integração – Projeto de Análise de Dados Simples com Listas e Dicionários – Pj2
12	21/10	<del>Primeira Avaliação Formal (P1). Correção da Avaliação após o intervalo.</del>
13	28/10	Modularização do código – Introdução a Funções
14	04/11	Parâmetros, Retornos e Boas Práticas de Funções
15	11/11	Continuação de funções, Módulos e Pacotes em Python
16	18/11	Introdução a Machine Learning – Pj3
17	25/11	<b>Semana de Apresentação PI de CDN</b>
18	02/12	Introdução a Ciência de Dados – partes 1 e 2 – Pandas, Numpy, Pipeline e exemplos – Pj4
19	09/12	<b>Segunda Avaliação Formal (P2).</b> Correção da Avaliação após o intervalo
20	16/12	<b>Exame / Avaliação Substitutiva.</b> Divulgação do Resultado Final.

# O que é uma FUNÇÃO?



# Como fazer café coado

-  Leve recipiente com água e leve ao fogo;
-  Adicione quatro colheres de café no coador já com o filtro de papel;
-  Quando as primeiras bolhas aparecerem, desligue o fogo;
-  Despeje a água lentamente sobre o café moído em movimentos circulares, emergindo todo o pó;
-  Não é necessário colocar água até o topo do coador. Vá adicionando aos poucos, se for o caso;
-  Espere toda a água baixar no coador.
-  Retire e adicione açúcar, se desejar;
-  Sirva.



# Engenharia de Software

Modularização...



**Temos que  
dividir, para  
conquistar!!**

Imagem: <http://felipevellozo.blogspot.com/2009/03/minhas-imagens-dos-outros-6-na>

# Engenharia de Software

## Modularização...

Técnica altamente recomendável, que consiste em dividir um programa/algoritmo maior ou principal em partes menores ou sub-rotinas (funções) tornando-o mais estruturado, organizado e refinado.

# Engenharia de Software

Modularização em Python...

→ Funções



# Engenharia de Software

## Modularização em Python...

### → Funções (builtin...)

abs()

min()

max()

divmod() → a,b = divmod(5,2) → a=2,b=1

pow() → a = pow(3,2) → a=9

len()

# Engenharia de Software

## Modularização em Python...

### → Funções

Uma função é um recurso (estático) que tem como objetivo a execução de um conjunto de comandos para atingir / realizar um objetivo ESPECÍFICO.

PRINCIPALMENTE QUANDO ESSE PROCESSO É REALIZADO COM FREQUENCIA (REPETIDO VÁRIAS VEZES)

DRY – **D**on't **R**epeat **Y**ourself

Para tanto, essa função pode ou não ter parâmetros. Ela pode ou não retornar algum valor.

# Funções

Execução de um conjunto de ações/comandos para atingir um objetivo ESPECÍFICO

PASSAR MENSAGEM PARA GERENTES  
REUNIÃO SEMANAL COM A DIREÇÃO.

PASSE O EMAIL PARA OS GERENTES



# Funções

Execução de um conjunto de ações/comandos para atingir um objetivo ESPECÍFICO

## **FUNÇÃO SEM PARÂMETRO E SEM RETORNO**

PASSAR MENSAGEM PARA GERENTES  
REUNIÃO SEMANAL COM A DIREÇÃO.



PASSE O EMAIL PARA OS GERENTES

**EXEMPLO: FUNÇÃO reset()**



# Funções

Execução de um conjunto de ações/comandos para atingir um objetivo ESPECÍFICO

PASSAR MENSAGEM PARA O  
CLIENTE  
ESPECIFICADO  
AGRADECENDO A COMPRA  
DOS PRODUTOS  
INFORMADOS.



PASSE O EMAIL PARA O CLIENTE  
<ABC> AGRADECENDO A COMPRA  
DOS PRODUTOS <X e Y>

# Funções

Execução de um conjunto de ações/comandos para atingir um objetivo ESPECÍFICO

## FUNÇÃO COM PARÂMETRO E SEM RETORNO

PASSAR MENSAGEM PARA O  
CLIENTE  
ESPECIFICADO  
AGRADECENDO A COMPRA  
DOS PRODUTOS  
INFORMADOS.



PASSE O EMAIL PARA O CLIENTE  
<ABC> AGRADECENDO A COMPRA  
DOS PRODUTOS <X e Y>

**EXEMPLO: FUNÇÃO `print("texto")`**

# Funções

Execução de um conjunto de ações/comandos para atingir um objetivo ESPECÍFICO

PASSAR MENSAGEM PARA O  
CLIENTE  
ESPECIFICANDO SOLICITANDO  
UM DIA E HORÁRIO PARA  
REUNIÃO

DIA e HORA



PASSE O EMAIL PARA O CLIENTE  
<ABC> SOLICITANDO UM DIA E HORÁRIO  
PARA UMA REUNIÃO

# Funções

Execução de um conjunto de ações/comandos para atingir um objetivo ESPECÍFICO

## FUNÇÃO COM PARÂMETRO E COM RETORNO

PASSAR MENSAGEM PARA O  
CLIENTE  
ESPECIFICANDO SOLICITANDO  
UM DIA E HORÁRIO PARA  
REUNIÃO



PASSE O EMAIL PARA O CLIENTE  
<ABC> SOLICITANDO UM DIA E HORÁRIO  
PARA UMA REUNIÃO

**EXEMPLO:** **FUNÇÃO** `int(3.56)`  
**3** ←



# Funções

Execução de um conjunto de ações/comandos para atingir um objetivo ESPECÍFICO

← → Bloco de comandos

(identação)



# Funções

Execução de um conjunto de ações/comandos para atingir um objetivo ESPECÍFICO

def <identificador>:

← → Bloco de comandos

(identação)



# Funções

Execução de um conjunto de ações/comandos para atingir um objetivo ESPECÍFICO

def <identificador> (lista parâmetros):

← → Bloco de comandos

(identação)

# Funções

Execução de um conjunto de ações/comandos para atingir um objetivo ESPECÍFICO

def <identificador> (lista parâmetros):

← → Bloco de comandos

(identação) return <VALOR>



# Funções

Execução de um conjunto de ações/comandos para atingir um objetivo ESPECÍFICO

Para chamar a função

<identificador> ()

<identificador> (parâmetros)

Variável = <identif.> (parâmet.)

# Função sem Retorno

# Funções sem Retorno

def <identificador> ():

← → Bloco de comandos

(identação)

# Funções sem Retorno

def <identificador> ():

← → Bloco de comandos

(identação)

**GERALMENTE  
UTILIZADA PARA  
A OTIMIZAÇÃO  
DE CÓDIGO  
REPETITIVO**

**Ex:  
MOSTRAR  
CREDENCIAIS**



# Exemplo

```
def mostra_rodape():  
    print("_____")  
    print("| PyPRO - Seja um profissional Python! |")  
    print("_____")  
  
# ...  
  
print("Execução do programa")  
print("Agora vamos chamar a função para mostrar o rodape")  
mostra_rodape()  
print("Execução de mais alguns comandos")  
print("Mostra mais uma vez o rodapé")  
mostra_rodape()  
print("Fim do programa")
```

# Vamos praticar ...

Faça uma função para mostrar uma saudação!

Por exemplo: Olá Pessoal!!

# Vamos praticar ...

Faça uma função para mostrar uma saudação!  
Por exemplo: Olá Pessoal!!

```
def saudacao():  
    print("Olá Pessoal!!")
```

# Atividade com IA

- Vamos saber mais sobre:
  - Funções...



- Faça individualmente, e depois compartilhe com o seu colega esses conceitos.
  - *“O que acontece no programa quando eu chamo uma função que acabei de definir? Pode descrever o passo a passo?”* – Foco> **fluxo de execução**
  - *“Como eu poderia explicar o conceito de função para alguém que nunca programou, usando uma analogia simples?”* – capacidade de abstração
  - *“Quais são erros comuns cometidos ao usar funções em Python?”*
  - *“Por que não devemos repetir o mesmo código várias vezes? Como funções nos ajudam nesse caso?”* – Foco> **reutilização de código** e manutenção

# Função com Retorno

# Funções com Retorno

def <identificador> ():

← → Bloco de comandos

(identação) return <lista de retornos>

# Exemplo

```
def um_megabits():  
    valor = 1024 * 1024  
    valor2 = pow(2, 20)  
    return valor  
  
# ...  
  
x = um_megabits()  
print(f"O total de bits é: {x}")
```



# Função com Parâmetro e Retorno

# Funções com Parâmetro e Retorno

def <identificador> (parâmetros):

← → Bloco de comandos

(identação) return <lista de retornos>

# Exemplo 1

```
def area_circulo(raio):  
    PI = 3.141592  
    area = PI * pow(raio, 2)  
    return area
```

```
# ...
```

```
r = float(input("Digite o raio: "))  
print(f"A área do círculo de raio {r} é igual a {area_circulo(r)}")
```

# Exemplo 2

#Funções que chamam outras funções

```
def area_circulo(raio):  
    PI = 3.141592  
    area = PI * pow(raio, 2)  
    return area  
  
def area_cilindro(altura, raio):  
    area = area_circulo(raio)*altura  
    return area  
  
# ...  
  
r = float(input("Digite o raio: "))  
h = float(input("Digite a altura: "))  
print(f"A área do cilindro de raio {r} e altura {h} é igual a {area_cilindro(r,h)}")
```

# Vamos praticar ...

Faça uma função para converter temperaturas em Celsius passadas como parâmetro para temperaturas em Fahrenheit.

# Atividade com IA

- Vamos consultar a IA para nos ajudar:
  - Conversão de celsius para fahrenheit



- Faça individualmente, e depois compartilhe com o seu colega esses conceitos.
  - “Qual a fórmula de conversão de celsius para fahrenheit?”
  - “E o inverso? De Fahrenheit para Celsius?”
  - “Como posso implementar de forma linear essas formulas utilizando em Python?”

# Vamos praticar ...

Faça uma função para converter temperaturas em Celsius passadas como parâmetro para temperaturas em Fahrenheit.

```
def converte_para_fahrenheit(celsius):  
    fahrenheit = celsius * 9/5 + 32  
    print(f"{celsius}°C = {fahrenheit}°F")
```



# Atividade com IA

- Vamos saber mais sobre:
  - Funções... Mais conceitos...



- Faça individualmente, e depois compartilhe com o seu colega esses conceitos.
  - *“Qual a diferença entre passar um parâmetro por valor e por referência em uma função? Como o Python lida com isso?” – foco> variáveis imutáveis vs mutáveis nos parâmetros.*
  - *“Posso usar variáveis definidas fora da função dentro dela? Como o Python busca variáveis no escopo global e local?”*
  - *“Como escolher bons nomes para funções e parâmetros?”*
  - *“O que acontece se eu definir duas funções com o mesmo nome em um programa Python?”*
  - *“Como as funções facilitam a resolução de problemas maiores na programação?”*

# VAMOS PARA A PRÁTICA ?!!!



# Desafio 1

## Verificador de Números Pares

Objetivo: Consolidar a criação de uma função com parâmetro e retorno simples.

### Enunciado:

Crie uma função chamada `eh_par(numero)` que receba um número inteiro como parâmetro e retorne `True` se ele for par e `False` caso contrário.

Depois, peça para o usuário digitar um número e mostre a mensagem apropriada.

Exemplo de uso:

```
n = int(input("Digite um número: "))
if eh_par(n):
    print("O número é par!")
else:
    print("O número é ímpar!")
```

# Desafio 2

## Calculadora de IMC

Objetivo: Criar função com múltiplos parâmetros e retorno de valor.

### Enunciado:

Implemente a função `calcular_imc(peso, altura)` que recebe o peso (kg) e a altura (m), e retorna o valor do IMC. Depois, crie uma segunda função `classificar_imc(imc)` que classifica o resultado como:

- Abaixo do peso ( $< 18.5$ )
- Peso normal (18.5 a 24.9)
- Sobrepeso (25 a 29.9)
- Obesidade ( $\geq 30$ )

Exemplo de uso:

```
p = float(input("Peso (kg): "))
a = float(input("Altura (m): "))
imc = calcular_imc(p, a)
print(f"IMC: {imc:.2f} - {classificar_imc(imc)}")
```

# Desafio 3

## Simulador de Crédito

Objetivo: Funções com lógica condicional e múltiplos retornos.

### Enunciado:

Crie a função `verificar_emprestimo(renda, valor_solicitado)` que avalia se um empréstimo pode ser aprovado com base na seguinte lógica:

- Se o valor solicitado for até 30% da renda, retorna "Aprovado"
- Se for entre 30% e 50%, retorna "Aprovado com restrições"
- Acima de 50%, retorna "Negado"

A função deve retornar a string da decisão.

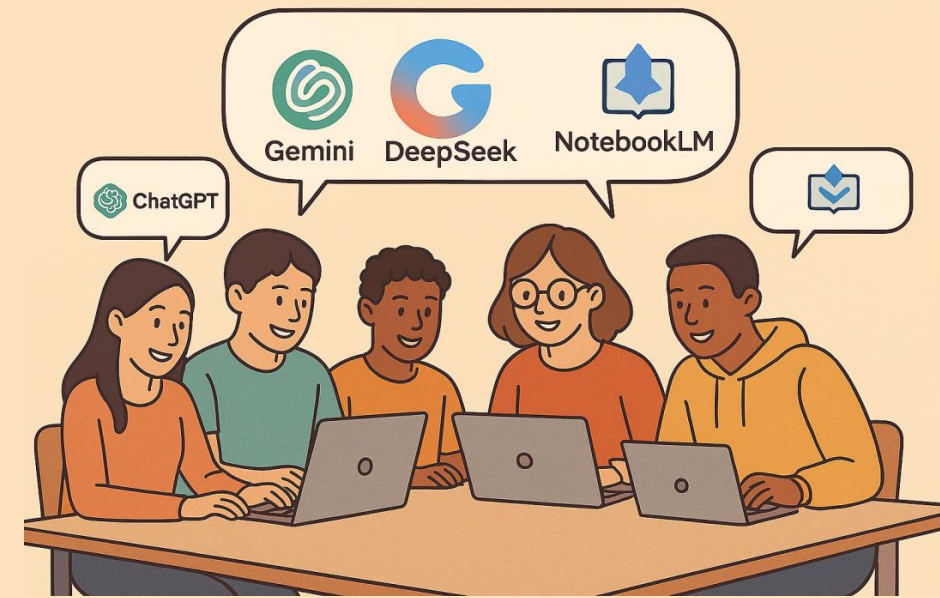
Exemplo de uso:

```
resultado = verificar_emprestimo(3000, 1200)
print(f"Situação: {resultado}")
```



# Atividade com IA

- Utilize a IA para corrigir os desafios acima
  - Conte com o apoio da IA para saber se o que você fez está correto.



- Faça individualmente...
  - *O seguinte desafio/exercício “ <<digite aqui o exercício>> “, eu resolvi da seguinte forma  
<< cole aqui seu código>>  
Pode verificar se minha resposta está correta?  
Sugira modificações para melhorar a legibilidade e coerência do código. Explique as alterações, caso existam.*

# Próxima Aula



- Ler os conteúdos da semana 14 e ver os vídeos indicados.
- No curso online, ver a videoaula correspondente!

***Boa semana e bons estudos!!***