

MAC2166 - Introdução à Computação

Prof. Dr. Helder Oliveira



Agenda

- **Exercícios de Funções**

Exercício

- Defina uma função `saudacao()` que imprima "Olá, mundo!". No `main()`, chame a função `saudacao()`.

Exercício

- Defina uma função `saudacao()` que imprima "Olá, mundo!". No `main()`, chame a função `saudacao()`.

```
1 def saudacao():  
2     print("Olá, mundo!")  
3  
4 def main():  
5     saudacao()  
6  
7 main()
```

Exercício

- Defina uma função `imprimir_nome(nome)` que imprima "Olá, [nome]". No `main()`, peça o nome ao usuário e chame a função.

Exercício

- Defina uma função `imprimir_nome(nome)` que imprima "Olá, [nome]". No `main()`, peça o nome ao usuário e chame a função.

```
1 def imprimir_nome(nome):  
2     print(f"Olá, {nome}")  
3  
4 def main():  
5     nome = input("Digite seu nome: ")  
6     imprimir_nome(nome)  
7  
8 main()
```

Exercício

- Defina uma função `soma(a, b)` que retorne a soma de dois números. No `main()`, leia dois números do usuário, chame a função e imprima o resultado.

Exercício

- Defina uma função `soma(a, b)` que retorne a soma de dois números. No `main()`, leia dois números do usuário, chame a função e imprima o resultado.

```
1 def soma(a, b):  
2     return a + b  
3  
4 def main():  
5     a = int(input("Digite o primeiro número: "))  
6     b = int(input("Digite o segundo número: "))  
7     resultado = soma(a, b)  
8     print(f"A soma é: {resultado}")  
9  
10 main()
```

Exercício

- Defina uma função `eh_par(numero)` que retorne `True` se o número for par. No `main()`, leia um número e imprima se ele é par ou não.

Exercício

- Defina uma função `eh_par(numero)` que retorne `True` se o número for par. No `main()`, leia um número e imprima se ele é par ou não.

```
1 def eh_par(numero):
2     return numero % 2 == 0
3
4 def main():
5     numero = int(input("Digite um número: "))
6     if eh_par(numero):
7         print("É par")
8     else:
9         print("É ímpar")
10
11 main()
```

Exercício

- Defina uma função `verificar_maioridade(idade)` que imprima se é maior ou menor de idade. No `main()`, leia a idade e chame a função.

Exercício

- Defina uma função `verificar_maioridade(idade)` que imprima se é maior ou menor de idade. No `main()`, leia a idade e chame a função.

```
1 def verificar_maioridade(idade):
2     if idade >= 18:
3         print("Maior de idade")
4     else:
5         print("Menor de idade")
6
7 def main():
8     idade = int(input("Digite a idade: "))
9     verificar_maioridade(idade)
10
11 main()
```

Exercício

- Defina uma função `contar_ate(n)` que imprima de 1 até n. No `main()`, leia o valor de n e chame a função.

Exercício

- Defina uma função `contar_ate(n)` que imprima de 1 até n. No `main()`, leia o valor de n e chame a função.

```
1 def contar_ate(n):
2     i = 1
3     while i <= n:
4         print(i)
5         i += 1
6
7 def main():
8     n = int(input("Digite até onde contar: "))
9     contar_ate(n)
10
11 main()
```

Exercício

- Defina uma função `multiplicacao(a, b)` que retorne o produto. No `main()`, leia dois números e mostre o produto.

Exercício

- Defina uma função `multiplicacao(a, b)` que retorne o produto. No `main()`, leia dois números e mostre o produto.

```
1 def multiplicacao(a, b):  
2     return a * b  
3  
4 def main():  
5     a = int(input("Digite o primeiro número: "))  
6     b = int(input("Digite o segundo número: "))  
7     resultado = multiplicacao(a, b)  
8     print(f"O produto é: {resultado}")  
9  
10 main()
```

Exercício

- Defina uma função `fatorial(n)` que retorne o fatorial de `n`. No `main()`, leia um número e imprima o seu fatorial.

Exercício

- Defina uma função `fatorial(n)` que retorne o fatorial de `n`. No `main()`, leia um número e imprima o seu fatorial.

```
1 def fatorial(n):
2     resultado = 1
3     while n > 1:
4         resultado *= n
5         n -= 1
6     return resultado
7
8 def main():
9     n = int(input("Digite um número para calcular o fatorial: "))
10    print(f"Fatorial de {n} é {fatorial(n)}")
11
12 main()
```

Exercício

- Defina uma função `soma_pares(n)` que some os números pares até n . No `main()`, leia n e mostre o resultado.

Exercício

- Defina uma função `soma_pares(n)` que some os números pares até `n`. No `main()`, leia `n` e mostre o resultado.

```
1 def soma_pares(n):
2     soma = 0
3     i = 0
4     while i <= n:
5         if i % 2 == 0:
6             soma += i
7             i += 1
8     return soma
9
10 def main():
11     n = int(input("Digite um número: "))
12     print(f"Soma dos pares até {n}: {soma_pares(n)}")
13
14 main()
```

Exercício

- Defina uma função `senha_valida(senha)` que verifique se a senha tem mais de 8 caracteres. No `main()`, peça uma senha e diga se é válida.

Exercício

- Defina uma função `senha_valida(senha)` que verifique se a senha tem mais de 8 caracteres. No `main()`, peça uma senha e diga se é válida.

```
1 def senha_valida(senha):  
2     return len(senha) > 8  
3  
4 def main():  
5     senha = input("Digite sua senha: ")  
6     if senha_valida(senha):  
7         print("Senha válida")  
8     else:  
9         print("Senha inválida")  
10  
11 main()
```

Exercício

- Defina uma função `adivinhar_numero(segredo)` que permita o usuário tentar adivinhar um número até acertar. No `main()`, defina um número secreto.

Exercício

- Defina uma função `adivinhar_numero(secreto)` que permita o usuário tentar adivinhar um número até acertar. No `main()`, defina um número secreto.

```
1 def adivinhar_numero(secreto):
2     chute = int(input("Adivinhe o número: "))
3     while chute != secreto:
4         print("Errado!")
5         chute = int(input("Tente novamente: "))
6     print("Acertou!")
7
8 def main():
9     adivinhar_numero(7)
10
11 main()
```

Exercício

- Defina uma função `menu_opcoes()` que exiba um menu de opções e só termine quando o usuário escolher sair. **No `main()`**, chame `menu_opcoes()`.

```
1 - Opção 1
2 - Opção 2
3 - Sair
Escolha uma opção:
```

Exercício

- Defina uma função `menu_opcoes()` que exiba um menu de opções e só termine quando o usuário escolher sair. **No `main()`**, chame `menu_opcoes()`.

```
1 def menu_opcoes():
2     opcao = 0
3     while opcao != 3:
4         print("\n1 - Opção 1")
5         print("2 - Opção 2")
6         print("3 - Sair")
7         opcao = int(input("Escolha uma opção: "))
8         if opcao == 1:
9             print("Você escolheu a opção 1")
10        elif opcao == 2:
11            print("Você escolheu a opção 2")
12        elif opcao == 3:
13            print("Saindo...")
14        else:
15            print("Opção inválida!")
16
17 def main():
18     menu_opcoes()
19
20 main()
```

Exercício

- Defina uma função `tabuada(numero)` que imprima a tabuada de 1 a 10 do número fornecido. No `main()`, leia um número e chame a função.

Exercício

- Defina uma função `tabuada(numero)` que imprima a tabuada de 1 a 10 do número fornecido. No `main()`, leia um número e chame a função.

```
1 def tabuada(numero):
2     i = 1
3     while i <= 10:
4         print(f"{numero} x {i} = {numero * i}")
5         i += 1
6
7 def main():
8     numero = int(input("Digite um número para ver a tabuada: "))
9     tabuada(numero)
10
11 main()
```

Exercício

- Defina uma função `retangulo(largura, altura)` que desenhe um retângulo de asteriscos usando dois `while` aninhados. No `main()`, leia a largura e altura e chame a função.

Exercício

- Defina uma função `retangulo(largura, altura)` que desenhe um retângulo de asteriscos usando dois `while` aninhados. No `main()`, leia a largura e altura e chame a função.

```
1 def retangulo(largura, altura):
2     linha = 0
3     while linha < altura:
4         coluna = 0
5         while coluna < largura:
6             print("*", end="")
7             coluna += 1
8         print()
9         linha += 1
10
```

```
10
11 def main():
12     largura = int(input("Digite a largura: "))
13     altura = int(input("Digite a altura: "))
14     retangulo(largura, altura)
15
16 main()
```

Exercício

- Defina uma função `classificar_nota(nota)` que: Se $\text{nota} \geq 9$, imprima "Excelente", Se $\text{nota} \geq 7$, imprima "Bom", Se $\text{nota} \geq 5$, imprima "Regular", Senão, imprima "Reprovado". Use if-elif-else. No `main()`, leia uma nota e chame a função.

Exercício

- Defina uma função `classificar_nota(nota)` que: Se $\text{nota} \geq 9$, imprima "Excelente", Se $\text{nota} \geq 7$, imprima "Bom", Se $\text{nota} \geq 5$, imprima "Regular", Senão, imprima "Reprovado". Use if-elif-else. No `main()`, leia uma nota e chame a função.

```
1 def classificar_nota(nota):
2     if nota >= 9:
3         print("Excelente")
4     elif nota >= 7:
5         print("Bom")
6     elif nota >= 5:
7         print("Regular")
8     else:
9         print("Reprovado")
10
```

```
10
11 def main():
12     nota = float(input("Digite a nota do aluno: "))
13     classificar_nota(nota)
14
15 main()
```

Exercício

- Defina uma função `desenhar_triangulo(n)` que desenhe um triângulo de asteriscos: 1 asterisco na primeira linha, 2 na segunda, até n asteriscos. Use laços aninhados. No `main()`, peça um número e desenhe o triângulo.

Exercício

- Defina uma função `desenhar_triangulo(n)` que desenhe um triângulo de asteriscos: 1 asterisco na primeira linha, 2 na segunda, até n asteriscos. Use laços aninhados. No `main()`, peça um número e desenhe o triângulo.

```
1 def desenhar_triangulo(n):
2     linha = 1
3     while linha <= n:
4         coluna = 1
5         while coluna <= linha:
6             print("*", end="")
7             coluna += 1
8         print()
9         linha += 1
10
```

```
10
11 def main():
12     n = int(input("Digite a altura do triângulo: "))
13     desenhar_triangulo(n)
14
15 main()
```

Exercício

- Defina uma função `menu_compras()` que tenha as opções: "Comprar livro", "Comprar roupa", "Comprar comida" e "Sair". Cada opção imprime uma mensagem diferente usando if-elif-else. O menu deve repetir até o usuário escolher sair.

Exercício

- Defina uma função `menu_compras()` que tenha as opções: "Comprar livro", "Comprar roupa", "Comprar comida" e "Sair". Cada opção imprime uma mensagem diferente usando if-elif-else. O menu deve repetir até o usuário escolher sair.

```
1 def menu_compras():
2     opcao = 0
3     while opcao != 4:
4         print("\n1 - Comprar livro")
5         print("2 - Comprar roupa")
6         print("3 - Comprar comida")
7         print("4 - Sair")
8         opcao = int(input("Escolha uma opção: "))
9         if opcao == 1:
10            print("Livro comprado!")
11        elif opcao == 2:
12            print("Roupa comprada!")
13        elif opcao == 3:
14            print("Comida comprada!")
15        elif opcao == 4:
16            print("Saindo...")
17        else:
18            print("Opção inválida!")
19
20 def main():
21     menu_compras()
22
23 main()
```

Exercício

- Crie uma função `labirinto_simples()` que simula a travessia de um labirinto:
- Existem 3 corredores (corredor 1, corredor 2, corredor 3). Em cada corredor, o jogador precisa escolher entre duas portas (porta 1 ou porta 2). Se escolher porta 1, ele avança. Se escolher porta 2, volta ao começo do corredor.
- Use dois `while` aninhados: um para os corredores e um para as escolhas de portas. No `main()`, apenas chame a função `labirinto_simples()`.

Exercício

```
1 def labirinto_simples():
2     corredor = 1
3     while corredor <= 3:
4         print(f"Você está no corredor {corredor}")
5         porta_certa = False
6         while not porta_certa:
7             escolha = int(input("Escolha a porta (1 ou 2): "))
8             if escolha == 1:
9                 print("Porta correta! Avançando...\n")
10                porta_certa = True
11            else:
12                print("Porta errada! Volte e tente novamente!\n")
13                corredor += 1
14        print("Parabéns! Você atravessou o labirinto.")
15
16 def main():
17     labirinto_simples()
18
19 main()
```

Dúvidas

