

MAC2166 - Introdução à Computação

Prof. Dr. Helder Oliveira



Agenda

- Leitura via teclado (= input())
- Classes str e int
- Função de conversão int()
- Operadores relacionais

Recebendo Dados do Usuário

- Leitura via teclado
- A função `input` é responsável por receber dados do usuário.
- O usuário deve escrever algo e pressionar a tecla `<enter>`.
- Normalmente, armazenamos o valor lido em uma variável.
- A função `input` obtém os dados fornecidos pelo console no formato de **string (str)**.
- Devemos fazer uma conversão dos dados se quisermos trabalhar com números.

Exemplos de Entrada de Dados

- Sintaxe da função input:

```
x = input("Mensagem opcional")
```

- Armazenando os valores lidos nas variáveis a e b:

```
a = input("Digite um valor para a variável a: ")  
b = input("Digite um valor para a variável b: ")  
print(int(a) + float(b))
```

Exercício

- Escreva um programa que leia dois números inteiros e imprima a soma dos dois números.

Exercício

- Escreva um programa que leia dois números inteiros e imprima a subtração dos dois números.

```
a = input("Digite um valor para a variável a: ")  
b = input("Digite um valor para a variável b: ")  
print(int(a) - int(b))
```

Função de Conversão

- Alguns tipos de dados permitem que o seu valor seja convertido para outro tipo (cast).
- Para isso, podemos usar as seguintes funções:
 - `int()` converte o valor para o tipo `int` (número inteiro).
 - `float()` converte o valor para o tipo `float` (número real).
 - `str()` converte o valor para o tipo `str` (string).
 - `bool()` converte o valor para o tipo `bool` (booleano).

Exemplos de Conversões de Tipos

- Convertendo uma string para um número inteiro:

```
1 a = "45"  
2 b = int(a)  
3 print(a)  
4 # '45'  
5 type(a)  
6 # <class 'str'>  
7 print(b)  
8 # 45  
9 type(b)  
10 # <class 'int'>
```

Exemplos de Conversões de Tipos

- Convertendo uma string para um número real:

```
1 a = "4.5"  
2 b = float(a)  
3 print(a)  
4 #'4.5'  
5 type(a)  
6 # <class 'str'>  
7 print(b)  
8 # 4.5  
9 type(b)  
10 # <class 'float'>
```

Exemplos de Conversões de Tipos

- Nem toda string pode ser convertida para um valor numérico:

```
1 a = "MAC2166"  
2 int(a)  
3 # ValueError: invalid literal for int() with base 10  
4 float(a)  
5 # ValueError: could not convert string to float: 'MAC2166'
```

Exemplos de Conversões de Tipos

- Convertendo valores numéricos:

```
1 a = 3.3
2 b = int(a)
3 print(b)
4 # 3
5 c = float(b)
6 print(c)
7 # 3.0
```

Exemplos de Conversões de Tipos

- Valores numéricos ou booleanos podem ser convertidos para strings:

```
1 str(102)
2 # '102'
3 str(3.1416)
4 # '3.1416'
5 str(True)
6 # 'True'
7 str(False)
8 # 'False'
```

Exemplos de Conversões de Tipos

- Convertendo um número inteiro numa string:

```
1 a = 45
2 b = str(a)
3 a * 3
4 # 135
5 b * 3
6 # '454545'
```

Exemplos de Conversões de Tipos

- Qualquer número, com exceção do 0 (zero), quando convertido para booleano resulta em True:

```
1 bool (-1.2)
2 # True
3 bool(0)
4 # False
5 bool (0.5)
6 # True
7 bool(2)
8 # True
```

Exemplos de Conversões de Tipos

- Qualquer string, com exceção da string vazia (""), quando convertida para booleano resulta em True:

```
1 bool("MAC2166")
2 # True
3 bool("")
4 # False
5 bool("Programação de Computadores")
6 # True
7 bool("X")
8 # True
```

Exemplos de Conversões de Tipos

- Valores booleanos podem ser convertidos para números:

```
1 int(True)
2 # 1
3 int(False)
4 # 0
5 float(True)
6 # 1.0
7 float(False)
8 # 0.0
```

Exemplos de Conversões de Tipos

- Valores booleanos podem ser usados em operações aritméticas, sem necessidade de conversão explícita:

```
1 False + False
2 #0
3 True + False
4 #1
5 True + True
6 #2
7 5.2 + (3 < 7)
8 #6.2
```

Exercício

- Peça para o usuário digitar seu nome e imprima uma mensagem personalizada.

Exercício

- Peça para o usuário digitar seu nome e imprima uma mensagem personalizada.

```
# Entrada do usuário
```

```
nome = input("Digite seu nome: ")
```

```
# Exibição da mensagem personalizada
```

```
print("Olá,", nome, "! Seja bem-vindo ao mundo da programação!")
```

Exercício

- Peça ao usuário para digitar seu nome e idade. Em seguida, imprima uma mensagem personalizada.

Exercício

- Peça ao usuário para digitar seu nome e idade. Em seguida, imprima uma mensagem personalizada.

```
# Solicita a entrada do usuário
nome = input("Digite seu nome: ")
idade = input("Digite sua idade: ")
# Exibe a mensagem personalizada
print("Olá,", nome + " Você tem", idade, "anos.")
```

Exercício

- Peça ao usuário para digitar um número e exiba seu tipo antes e depois da conversão para inteiro.

Exercício

- Peça ao usuário para digitar um número e exiba seu tipo antes e depois da conversão para inteiro.

```
# Entrada do usuário
numero = input("Digite um número: ")
# Exibe o tipo original
print("O tipo da variável antes da conversão é:", type(numero))
# Converte para inteiro
numero = int(numero)
# Exibe o tipo após a conversão
print("O tipo da variável após a conversão é:", type(numero))
```

Exercício

- Peça ao usuário dois números inteiros, converta-os e exiba a soma.

Exercício

- Peça ao usuário dois números inteiros, converta-os e exiba a soma.

```
# Entrada dos números como string
num1 = input("Digite o primeiro número inteiro: ")
num2 = input("Digite o segundo número inteiro: ")

# Conversão para int
num1 = int(num1)
num2 = int(num2)

# Soma dos números
soma = num1 + num2

# Exibe o resultado
print("A soma de", num1, "e", num2, "é:", soma)
```

Tipo bool

- Já vimos que o tipo bool é utilizado para representar os valores booleanos verdadeiro (True) e falso (False).

```
1 a = True
2 b = False
```

- O uso mais comum dessas variáveis é na verificação de expressões relacionais e lógicas.

Expressões Relacionais

- Expressões relacionais são aquelas que realizam uma comparação entre duas expressões e retornam:
 - True, se o resultado for verdadeiro.
 - False, se o resultado for falso.
- Os operadores relacionais são:
 - == igualdade.
 - != diferente.
 - > maior que.
 - < menor que.
 - >= maior ou igual que.
 - <= menor ou igual que.
- Nos próximos exemplos, considere que foram feitas as seguintes atribuições:

```
1 a = 20
2 b = 21
```

Expressões Relacionais

- $\langle \text{expressão} \rangle == \langle \text{expressão} \rangle$: retorna verdadeiro quando as expressões forem iguais.

```
1 a == (10 * 2) # a = 20
2 # True
3 b == (10 * 2) # b = 21
4 # False
```

- $\langle \text{expressão} \rangle != \langle \text{expressão} \rangle$: retorna verdadeiro quando as expressões forem diferentes.

```
1 a != (10 * 2) # a = 20
2 # False
3 b != (10 * 2) # b = 21
4 # True
```

Expressões Relacionais

- $\langle \text{expressão} \rangle > \langle \text{expressão} \rangle$: retorna verdadeiro quando a expressão da esquerda tiver valor maior que a expressão da direita.

```
1 # a = 20 e b = 21
2 a > b
3 # False
```

- $\langle \text{expressão} \rangle < \langle \text{expressão} \rangle$: retorna verdadeiro quando a expressão da esquerda tiver valor menor que a expressão da direita.

```
1 # a = 20 e b = 21
2 a < b
3 # True
```

Expressões Relacionais

- $\langle \text{expressão} \rangle \geq \langle \text{expressão} \rangle$: retorna verdadeiro quando a expressão da esquerda tiver valor maior ou igual que a expressão da direita.

```
1 # a = 20 e b = 21
2 a >= b
3 # False
```

- $\langle \text{expressão} \rangle \leq \langle \text{expressão} \rangle$: retorna verdadeiro quando a expressão da esquerda tiver valor menor ou igual que a expressão da direita.

```
1 # a = 20 e b = 21
2 a <= b
3 # True
```

Expressões Relacionais com Strings

- Ordem considerada para os caracteres do alfabeto:
 - ABC...XYZabc...xyz

```
1 "a" > "b"  
2 # False  
3 "a" == "a"  
4 # True  
5 "a" == "A"  
6 # False  
7 "Z" < "a"  
8 # True  
9 "z" < "a"  
10 # False  
11
```

Operadores Relacionais

Operador	Descrição	Exemplo	Avalia para
==	igual a	5 == 5	True
!=	diferente de	8 != 5	True
>	Maior que	3 > 10	False
<	Menor que	5 < 8	True
>=	Maior ou igual a	5 >= 10	False
<=	Menor ou igual a	5 <= 5	True

Operador **in**

- O que faz?
 - O operador **in** verifica se um elemento existe em uma sequência (como lista, string ou tupla).
- Exemplo com string:

```
palavra = "programacao"  
print('a' in palavra)  
# Saída: True  
print('z' in palavra)  
# Saída: False
```

- Verifica se a letra 'a' está na palavra "programacao". Retorna True se presente e False se ausente.

Exemplo

- O que será impresso pelo código a seguir?

```
print((3 * 4) / 2 == (2 * 3))  
# True  
print((4 / 3) <= 1.33)  
# False
```

Exercício

- Crie um programa que receba dois números e imprima True se o primeiro for maior que o segundo e False caso contrário.

Exercício

- Crie um programa que receba dois números e imprima True se o primeiro for maior que o segundo e False caso contrário.

```
a = int(input("Digite o primeiro número: "))  
b = int(input("Digite o segundo número: "))  
print(a > b)
```

Exercício

- Escreva um programa que receba dois valores e retorne True se eles forem iguais e False caso contrário.

Exercício

- Escreva um programa que receba dois valores e retorne True se eles forem iguais e False caso contrário.

```
x = input("Digite o primeiro valor: ")  
y = input("Digite o segundo valor: ")  
print(x == y)
```

Exercício

- Faça um programa que receba dois números e imprima **True** se o primeiro for múltiplo do segundo, e **False** caso contrário.

Exercício

- Faça um programa que receba dois números e imprima **True** se o primeiro for múltiplo do segundo, e **False** caso contrário.

```
a = int(input("Digite um número: "))  
b = int(input("Digite outro número: "))  
# Verifica se o primeiro número é múltiplo do segundo  
print(bool(a % b == 0))
```

Exercício

- Escreva um programa que receba três números e imprima True se o primeiro número estiver entre o segundo e o terceiro.

Exercício

- Escreva um programa que receba três números e imprima True se o primeiro número estiver entre o segundo e o terceiro.

```
n = int(input("Digite um número: "))  
limite_inferior = int(input("Digite o limite inferior: "))  
limite_superior = int(input("Digite o limite superior: "))  
print(limite_inferior <= n <= limite_superior)
```

Exercício

- Faça um programa que receba um número e retorne True se ele for par e False caso contrário.

Exercício

- Faça um programa que receba um número e retorne True se ele for par e False caso contrário.

```
# Recebe o número do usuário
numero = float(input("Digite um número: "))

# Verifica se o número é par
resultado = (numero - 2 * int(numero / 2)) == 0

# Exibe o resultado
print(resultado)
```

Exercício

- Peça a idade de uma pessoa e verifique se ela tem 18 anos ou mais.

Exercício

- Peça a idade de uma pessoa e verifique se ela tem 18 anos ou mais.

```
idade = int(input("Digite sua idade: "))  
print(idade >= 18)
```

Exercício

- Peça ao usuário uma palavra e uma letra e verifique se a letra está dentro da palavra.

Exercício

- Peça ao usuário uma palavra e uma letra e verifique se a letra está dentro da palavra.

```
palavra = input("Digite uma palavra: ")  
letra = input("Digite uma letra: ")  
print(letra in palavra)
```

Exercício

- Peça a idade e verifique se a pessoa pode dirigir e votar ao mesmo tempo. Para isso, a idade deve ser **maior ou igual a 18**.

Exercício

- Peça a idade e verifique se a pessoa pode dirigir e votar ao mesmo tempo. Para isso, a idade deve ser **maior ou igual a 18**.

```
idade = int(input("Digite sua idade: "))  
print(idade >= 18 and idade <= 100)  
# Considerando um limite superior razoável
```

Exercício

- Peça um número ao usuário e verifique se ele está entre 10 e 50 (inclusive).

Exercício

- Peça um número ao usuário e verifique se ele está entre 10 e 50 (inclusive).

```
num = int(input("Digite um número: "))  
print(10 <= num <= 50)
```

Exercício

- Solicite um número e retorne True se ele for positivo e par ao mesmo tempo.

Exercício

- Solicite um número e retorne True se ele for positivo e par ao mesmo tempo.

```
# Recebe o número do usuário
numero = float(input("Digite um número: "))

# Verifica se o número é positivo e par ao mesmo tempo
resultado = (numero > 0) * ((numero - 2 * int(numero / 2)) == 0)

print(bool(resultado))
```

Exercício

- Peça ao usuário uma senha e verifique se ela **não** é igual a "12345" (uma senha fraca).

Exercício

- Peça ao usuário uma senha e verifique se ela **não** é igual a "12345" (uma senha fraca).

```
senha = input("Digite sua senha: ")  
print(senha != "12345")
```

Exercício

- Peça um número e verifique se ele é múltiplo de 3 e múltiplo de 5 ao mesmo tempo.

Exercício

- Peça um número e verifique se ele é múltiplo de 3 e múltiplo de 5 ao mesmo tempo.

```
# Recebe o número do usuário
numero = float(input("Digite um número: "))

# Verifica se o número é múltiplo de 3 e 5 ao mesmo tempo
resultado = (numero / 3 == int(numero / 3)) * (numero / 5 == int(numero / 5))

# Exibe o resultado
print(bool(resultado))
```

Dúvidas



Referencias

- Slides baseados em:
 - Zanoni Dias, MC102- Algoritmos e Programação de Computadores, 2020.