



62

Saída de Dados

- Python fornece várias funções integradas que estão prontamente disponíveis para nós no prompt do Python.
- Algumas das funções `input()` e `print()` são amplamente utilizadas para operações de entrada e saída padrão, respectivamente.
- Veremos a seção de saída primeiro.

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

63

Saída de Dados

- Usamos a função `print()` para enviar dados para o dispositivo de saída padrão (tela).
- Também podemos enviar dados para um arquivo, mas isso será abordado mais tarde.
- Dois exemplos de seu uso é dado ao lado.

```
print("Olá Mundo!!!")  
  
saldo = 20  
print("Seu saldo é ", saldo)
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

64

Saída de Dados

- A função `print()` possui a seguinte sintaxe:

```
print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False)
```

- `objects`: são os valores a serem impressos na tela
- `sep`: é o tipo de separador, por padrão são apresentados espaços em branco, mas podemos colocar outros tipos de caracteres
- `end`: após imprimir todos os valores, o `end` é impresso. Geralmente o padrão é uma nova linha.

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

65

Saída de Dados

- Teste estes exemplos

```
print(1, 2, 3, 4)
print(1, 2, 3, 4, sep='*')
print(1, 2, 3, 4, sep='#', end='&')
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

66

Saída de Dados - Formatando a saída

- Às vezes, gostaríamos de formatar nossa saída para torná-la atraente. Isso pode ser feito usando o método `str.format()`.
- Este método é visível para qualquer objeto string.

```
x = 5; y = 10
print('O valor de x é {} e o valor de y é {}'.format(x,y))
```

- As chaves `{}` são usadas como marcadores de posição. Podemos especificar a ordem em que são impressos usando números (índice de tupla). Veja:

```
print('A linguagem {0} é mais simples que {1}'.format('Python','JavaScript'))
print('A linguagem {1} é mais simples que {0}'.format('Python','JavaScript'))
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

67

Saída de Dados - Formatando a saída

- Podemos também utilizar argumentos como palavra-chave ao formatar uma string

```
print('Olá {nome}, {saudacao}'.format(saudacao = 'bem vinda', nome = 'Ana'))
```

- As chaves {} são usadas como marcadores de posição. Podemos especificar a ordem em que são impressos usando números (índice de tupla). Veja:

```
print('A linguagem {0} é mais simples que {1}'.format('Python','JavaScript'))  
print('A linguagem {1} é mais simples que {0}'.format('Python','JavaScript'))
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

68

Entrada de dados

- Para permitir flexibilidade, podemos querer obter a entrada do usuário. Em Python, temos a `input()` função para permitir isso. A sintaxe para `input()` é:

```
input([prompt])
```

- Exemplo:

```
num = input('Informe um número: ')
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

69

Entrada de dados

- Aqui, podemos ver que o valor inserido 10 é uma string, não um número. Para converter isso em um número, podemos usar as funções `int()` ou `float()`..

```
num = int(input('Informe um número: '))
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

70

Saída de dados

```
print(2 + 2)
print(4 * 5)
print("2" + "3")
print("A" + "B")
print("A" + "5")
print("Dias do mês de janeiro" + 31)
print("Dias do mês de janeiro", 31)
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

71

Saída de dados

```
a = 4
b = 5
total = a + b
print("O total é: ",total)
print("O total é:",total)
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida
reprodução e uso sem autorização

72

Entrada de dados

```
nome = input("Informe o seu nome")
print("Seja bem vindo", nome, "como posso te ajudar hoje?")
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida
reprodução e uso sem autorização

73

Entrada de dados

```
a = input("Digite um número")
b = input("Digite um outro número")
total = a + b
print("O total é", total)
```

- O que você achou da saída???

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

74

Entrada de dados

- O comando input recebe como entrada uma cadeia de caracteres
- Você se lembra que não podemos realizar operações com caracteres?
 - Mesmo tendo a aparência de número, ele não é um tipo número, mas tipo string
 - Solução?
- Vamos converter
 - int, float, str

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

75

Entrada de dados

```
a = int(input("Digite um número"))
b = int(input("Digite um outro número"))
total = a + b
print("O total é", total)
```

- Agora convertemos o valor assim que o recebemos no input

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

76

Operadores aritméticos

- Operadores são símbolos especiais em Python que realizam cálculos aritméticos, de comparação ou lógicos. O valor em que o operador opera é chamado de operando.
 - Os **operadores aritméticos** em Python desempenham o mesmo papel dos operadores na matemática
 - Os **operadores de comparação** são usados para comparar valores. Eles retornam True ou False de acordo com a condição
 - Os **operadores lógicos** são os operadores and, or, not.

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

77

Operadores aritméticos

Operador	Significado	Exemplo
+	Some dois operandos	$x + y + 2$
-	Subtraia o operando direito do esquerdo	$x - y - 2$
*	Multiplique dois operandos	$x * y$
/	Divida o operando esquerdo pelo direito (sempre resulta em flutuante)	x / y
%	Módulo - resto da divisão do operando esquerdo pelo direito	$x \% y$ (resto de x / y)
//	Divisão de piso - divisão que resulta em número inteiro ajustado à esquerda na linha do número	$x // y$
**	Expoente - operando esquerdo elevado à potência direita	$x ** y$ (x elevado à potência y)

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

78

Operadores aritméticos

```
x = 15
y = 4

print('x + y =',x+y)
print('x - y =',x-y)
print('x * y =',x*y)
print('x / y =',x/y)
print('x // y =',x//y)
print('x ** y =',x**y)
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

79

Operadores aritméticos

```
import math

a = int(input("Informe o valor de a"))
b = int(input("Informe o valor de b"))
c = int(input("Informe o valor de c"))

delta = (b**2) - (4*a*c)
raisQ = math.sqrt(delta)

x1 = (-b + raisQ)/(2*a)
x2 = (-b - raisQ)/(2*a)

print("x1:" , x1)
print("x2:" , x2)
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

80

Operadores de comparação

Operador	Significado	Exemplo
>	Maior que - verdadeiro se o operando esquerdo for maior que o direito	$x > y$
<	Menor que - verdadeiro se o operando esquerdo for menor que o direito	$x < y$
==	Igual a - Verdadeiro se ambos os operandos forem iguais	$x == y$
!=	Diferente de - Verdadeiro se os operandos não forem iguais	$x != y$
>=	Maior que ou igual a - verdadeiro se o operando esquerdo for maior ou igual ao direito	$x >= y$
<=	Menor ou igual a - Verdadeiro se o operando esquerdo for menor ou igual ao direito	$x <= y$

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

81

Operadores de comparação

```
x = 15
y = 4

print('x > y é', x>y)

print('x < y é', x<y)

print('x == y é', x==y)

print('x != y é', x!=y)

print('x >= y é', x>=y)

print('x <= y é', x<=y)
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

82

Operadores de comparação

▪ O que comparar?

- <valor> <op_comparação> <valor>
- <variável> <op_comparação> <valor>
- <variável> <op_comparação> <variável>
- <variável> <op_comparação> <expressão>
- <expressão> <op_comparação> <valor>
- <expressão> <op_comparação> <variável>
- <expressão> <op_comparação> <expressão>

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

83

Operadores lógicos

Operador	Significado	Exemplo
e	Verdadeiro se ambos os operandos forem verdadeiros	x e y
ou	Verdadeiro se algum dos operandos for verdadeiro	x ou y
não	Verdadeiro se o operando for falso (complementa o operando)	não x

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

84

Operadores lógicos

```
x = True
y = False

print('x e y is',x and y)
print('x e y is',x and x)
print('x e y is',y and y)

print('x ou y is',x or y)
print('x ou y is',x or x)
print('x ou y is',y or y)

print('not x é',not x)
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

85

Operadores de atribuição

Operador	Exemplo	Equivalente a
=	x = 5	x = 5
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
%=	x %= 5	x = x % 5
//=	x //= 5	x = x // 5
**=	x **= 5	x = x ** 5

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

86

Operadores de atribuição

```
x = True
y = False

print('x e y is', x and y)
print('x e y is', x and x)
print('x e y is', y and y)

print('x ou y is', x or y)
print('x ou y is', x or x)
print('x ou y is', y or y)

print('not x é', not x)
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

87

Operador de identidade

- `is` e `is not` são os operadores de identidade em Python. Eles são usados para verificar se dois valores (ou variáveis) estão localizados na mesma região da memória. Duas variáveis iguais não significa que sejam idênticas.

Operador	Significado	Exemplo
<code>is</code>	Verdadeiro se os operandos forem idênticos (referir-se ao mesmo objeto)	<code>x é True</code>
<code>is not</code>	Verdadeiro se os operandos não forem idênticos (não se referem ao mesmo objeto)	<code>x não é True</code>

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

88

Operadores de identidade

```
x1 = 5
y1 = 5
x2 = 'Olá Mundo'
y2 = 'Olá Mundo'
x3 = [1,2,3]
y3 = [1,2,3]

print(x1 is not y1)

print(x2 is y2)

print(x3 is y3)
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

89

Operador de associação

- `in` e `not in` são os operadores de associação em Python. Eles são usados para testar se um valor ou variável é encontrado em uma sequência (string, lista, tupla, conjunto e dicionário).

Operador	Significado	Exemplo
<code>in</code>	Verdadeiro se o valor / variável for encontrado na sequência	5 em x
<code>not in</code>	Verdadeiro se o valor / variável não for encontrado na sequência	5 não em x

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

90

Operadores de associação

```
x = 'Olá Mundo'
y = {1:'a',2:'b'}

print('M' in x)

print('Mundo' not in x)

print(1 in y)

print('a' in y)
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

91

Operações sobre strings

- Podemos realizar diversas operações em strings, a seguir serão apresentados alguns exemplos.

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

92

Operações sobre strings

```
frase= "bom dia meus amigos"

#capturando valores dentro da string
print(frase[4])

#verificar o tamanho da string
print(len(frase))

#verificar se uma palavra está na frase ==> meus
print("meus" in frase)

#verificar se uma palavra NÃO está na frase ==> tarde
print("dia" not in frase)

if("boa tarde" not in frase):
    print("Estamos de manhã")
else: #se estiver na frase
    print("Estamos de tarde")
```

93

Operações sobre strings

```
#fatias de corte
texto = "Semana Que Vem é Feriado"
fatia = texto[2:9] #2,3,4,5,6 e 7
print(fatia)

#fatia a partir do início
fatia = texto[:9]
print(fatia)

#fatia até o fim
fatia = texto[3:]
print(fatia)
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

94

Operações sobre strings

```
#tornar os caracteres maiúsculos
print(texto.upper())

fatia = texto[3:6].upper()
print(fatia)

#tornar os caracteres minúsculos
print(texto)
print(texto.lower())

#remover espaço em branco antes ou depois da string
print(texto.strip())
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

95

Operações sobre strings

```
#substituir string
frase = "Olá, Mundo"
print(frase.replace("Mundo", "Alunos"))

#dividir string
frase = "Banana, arroz, feijão, uva"

result = frase.split(",")
print(result[0])
print(result[1])
print(result[2])
print(result[3])
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

96

Por hoje é só!

Orientações

- Os códigos de exemplo desta aula
- Pratique os exercícios desta aula
- Somente com a prática constante você será fluente em Python!

Qualquer dúvida:

- www.arieldias.com
- contato@arieldias.com

Bons estudos! Até a próxima!

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

97