

Programação Script

Entrada/Saída e Expressões aritméticas

Aula 02

Prof. Felipe A. Louza



- 1 Saída de dados: `print()`
- 2 Exemplo: Conversão de Fahrenheit para Celsius
- 3 Entrada de dados: `input()`
- 4 Conversão de Tipos
- 5 Operadores Aritméticos e Expressões
- 6 Precedência de operadores
- 7 Referências

- 1 Saída de dados: `print()`
- 2 Exemplo: Conversão de Fahrenheit para Celsius
- 3 Entrada de dados: `input()`
- 4 Conversão de Tipos
- 5 Operadores Aritméticos e Expressões
- 6 Precedência de operadores
- 7 Referências

Saída de dados: `print()`

Para imprimir um texto na tela, utilizamos o comando `print()`

- O texto pode ser um literal do tipo **string** (entre aspas):

```
1 print("Já vimos isso!")
```

- Saída:

```
1 Já vimos isso!  
2
```

O `print()` pula de linha (a menos que o parâmetro `end` seja alterado)

Saída de dados: `print()`

Lembre-se de combinar corretamente o uso das aspas:

1 ✓

```
1 print("Sem problemas!")
```

2 ✓

```
1 print('Relax!')
```

3 ✗

```
1 print("Ops!');
```

Escrevendo na tela

No meio da **string** podemos incluir **caracteres de formatação** especiais.

- O símbolo especial `\n` é responsável por **pular uma linha** dentro do texto.

```
1 print("Já até usei o print..\n No trabalho")
```

- Saída:

```
1 Já até usei o print..  
2 No trabalho  
3
```

Escrevendo o conteúdo de uma variável na tela

Podemos imprimir, além de texto, o conteúdo de uma variável com o `print()`.

- Separamos o texto e a variável com uma vírgula.

```
1 a = 10
2 print("A variável contém o valor", a)
```

- Saída:

```
1 A_variável_contém_o_valor_10
2
```

Escrevendo múltiplos argumentos

A impressão com **múltiplos argumentos** inclui um **espaço extra** entre cada argumento.

- Separamos cada **argumento** com uma **vírgula**.

```
1 x = 10
2 y = 3.14
3 print("x contém o valor:", x, "\ne y contém o valor:", y)
```

- Saída:

```
1 x contém o valor: 10 ↵
2 e y contém o valor: 3.14 ↵
3
```

Podemos modificar esse **espaço extra** com o parâmetro **sep** do **print()**

Escrevendo o conteúdo de uma variável na tela

Outra opção é **converter** todos os valores em strings (**str**)

- Depois, com o operador **+** concatenamos as strings:

```
1 x = 10
2 y = 3.14
3 print("x contém o valor: "+str(x)+"\ne y contém o valor:"+str(y))
```

- Saída:

```
1 x_contém_o_valor_10↵
2 e_y_contém_o_valor:_3.14↵
3
```

Saída de dados: `print()`

Outros exemplos:

1 ✓

```
1 print("texto!")
```

```
1 print('texto!', 10)
```

```
1 print(1, 2, 3, 4);
```

2 `sep`

```
1 print(1, 2, 3, 4, sep="*");
```

3 `end`

```
1 print(1, 2, 3, 4, sep="*", end="&");
```

Formatando a saída

Podemos **formatar (especificar)** a saída com o método `str.format()`:

- Exemplo:

```
1 x = 10
2 y = 3.14
3 print("x= {} e y = {}".format(x, y))
```

- Saída:

```
1 x=_10_e_y=_3.14↵
2
```

Formatando a saída

Podemos **alterar a ordem** dos parâmetros:

- Exemplos:

```
1 x = 10
2 y = 3.14
3 print("x = {1} e y = {0}".format(y, x))
```

```
1 x = 10
2 y = 3.14
3 print("x = {a} e y = {b}".format(a=x, b=y))
```

Formatando a saída

Podemos especificar o **número de casas decimais** (**N**) que deve ser impresso em um número ponto flutuante **float** usando **.Nf**:

- Exemplo:

```
1 pi = 3.1415
2 r = 7
3 area = pi*r**2
4 print("Área do círculo de raio {0} é igual à {1:.3f}".format(r, area))
```

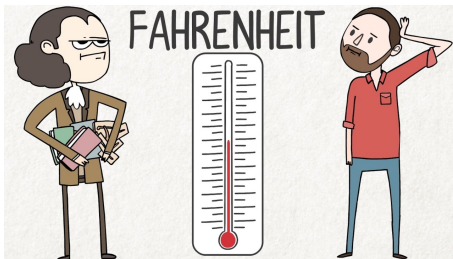
- Saída:

```
1 Área do círculo de raio 7 é igual à 153.934
2
```

- 1 Saída de dados: `print()`
- 2 Exemplo: Conversão de Fahrenheit para Celsius
- 3 Entrada de dados: `input()`
- 4 Conversão de Tipos
- 5 Operadores Aritméticos e Expressões
- 6 Precedência de operadores
- 7 Referências

Conversão de Fahrenheit para Celsius

Vamos criar um programa que converte uma temperatura dada em
Fahrenheit para **Celsius**



Conversão de Fahrenheit para Celsius

Relembrando:

$$^{\circ}\text{C} = (^{\circ}\text{F} - 32) \times 5 / 9$$

- Então, dada uma temperatura $F = 78$, podemos escrever o seguinte programa em Python:

```
1 F = 78
2 C = (F-32)*5/9
3 print("A temperatura em Celsius é", C)
```


Conversão de Fahrenheit para Celsius

- Ao executar `converte.py`:

```
1 $ python3 converte.py
```

- Saída:

```
1 A temperatura em Celsius é 23.888888889
```

Conversão de Fahrenheit para Celsius

Vamos supor que em **outro dia** você precise converter a temperatura $F = 93$.

- Você vai ter que abrir o editor de textos e **mudar o programa**.

```
1 F = 93
2 C = (F-32)*5/9
3 print("A temperatura em Celsius é", C)
```

Ruim!

Não é nada prático!

Entrada de dados

- Ao invés de ter um valor fixo, vamos fazer um programa que **recebe da entrada** do usuário o valor a ser calculado

- 1 Saída de dados: `print()`
- 2 Exemplo: Conversão de Fahrenheit para Celsius
- 3 Entrada de dados: `input()`
- 4 Conversão de Tipos
- 5 Operadores Aritméticos e Expressões
- 6 Precedência de operadores
- 7 Referências

Entrada de dados

A função `input()`:

- Realiza a leitura de dados a partir do teclado.
- Aguarda que o usuário digite um valor e atribui o valor digitado a uma variável.
- Todos os dados lidos são do tipo string (`str`).

```
1 print("Digite um número:")
2 a = input()
3 print("O número digitado é:"+a)
```

Entrada de dados

Mensagem no comando `input()`:

- Podemos incluir um texto a ser impresso diretamente no comando `input()`

```
1 a = input("Digite um numero:")  
2 print("O número digitado é:"+a)
```

Entrada de dados

Lendo números inteiros:

- Precisamos converter a string lida do teclado usando a função `int()`.

```
1 a = int(input("Digite um numero:"))  
2 a = a*10  
3 print("O número digitado vezes 10 é: ", a)
```

Podemos fazer o mesmo para **números ponto flutuante**.

- Usamos a função `float()`:

```
1 a = float(input("Digite um numero:"))  
2 a = a*10  
3 print("O número digitado vezes 10 é {:.2f}: ".format(a))
```

Entrada de dados

Nos dois exemplos anteriores é **esperado** que o usuário **digite um número**.

- Se o usuário digitar **um texto não numérico** o programa encerrará com um **erro de execução**.

```
1 Digite um numero:
2 um
3 Traceback (most recent call last):
4   File "../PS/teste.py", line 2, in <module>
5     a = int(input())
6 ValueError: invalid literal for int() with base 10: 'um'
```


- 1 Saída de dados: `print()`
- 2 Exemplo: Conversão de Fahrenheit para Celsius
- 3 Entrada de dados: `input()`
- 4 Conversão de Tipos
- 5 Operadores Aritméticos e Expressões
- 6 Precedência de operadores
- 7 Referências

Conversão de Tipos

Já vimos que as funções `int()`, `float()` e `str()` servem para converter dados de um tipo em outro:

- A conversão só ocorre se o dado estiver bem formado. Por exemplo `int("aaa")` resulta em um erro.
- Ao convertermos um número `float` para `int` ocorre um truncamento, ou seja, toda parte fracionária é desconsiderada.

Conversão de Tipos

Exemplo:

```
1 >>> a = "ola"
2 >>> int(a)
3 Traceback (most recent call last):
4   File "<stdin>", line 1, in <module>
5   ValueError: invalid literal for int() with base 10: 'ola'
6 >>> int(2.99)
7 2
8 >>> int(-2.99)
9 -2
10 >>> float("3.1415")
11 3.1415
12 >>>
```

- 1 Saída de dados: `print()`
- 2 Exemplo: Conversão de Fahrenheit para Celsius
- 3 Entrada de dados: `input()`
- 4 Conversão de Tipos
- 5 Operadores Aritméticos e Expressões**
- 6 Precedência de operadores
- 7 Referências

Operadores aritméticos

Operadores aritméticos em Python:

Operação	Operador	Exemplo	Exemplo em Python
Adição	+	$a + 2$	<code>a+2</code>
Subtração	-	$b - c$	<code>b-c</code>
Multiplicação	*	$a \times 5$	<code>a*5</code>
Divisão	/	$d \div e$	<code>d/e</code>
Divisão inteira	//	$a \div b$	<code>a//b</code>
Potenciação	**	a^b	<code>a**b</code>
Módulo	%	$5 \bmod 2$	<code>5%2</code>

- Módulo é o resto da divisão inteira.

Expressões e aritméticas

Podemos atribuir **expressões aritméticas** com valores em **variáveis** ou resultados de outras expressões:

- A sintaxe é a seguinte:

variável = **expressão**

- Exemplo:

```
1 >>> a = 1
2 >>> b = a
3 >>> c = a + b
```

Expressões Aritméticas

Alguns exemplos:

- *expressão* + *expressão*

```
1 >>> 56+9
2 65
```

- *expressão* - *expressão*

```
1 >>> 56-9
2 47
```

- *expressão* * *expressão*

```
1 >>> 56*9
2 504
```

Expressões Aritméticas

O resultado da **divisão** é sempre um número ponto flutuante.

- *expressão* / *expressão*

```
1 >>> 6/4  
2 1.5
```

Python

6 / 4	1.5
6.0 / 4.0	1.5
6.0 / 4	1.5
6 / 4.0	1.5

Expressões Aritméticas

O resultado da **divisão inteira** é sempre um número inteiro.

- *expressão* // *expressão*

```
1 >>> 6//4
2 1
```

Python

6 // 4	1
6.0 // 4.0	1
6.0 // 4	1
6 // 4.0	1

Se um dos operandos for **ponto flutuante** faz uma divisão truncada.

Expressões Aritméticas

A **potenciação** calcula o valor da expressão **à esquerda** **elevado** ao valor da expressão **à direita**.

- *expressão* ** *expressão*

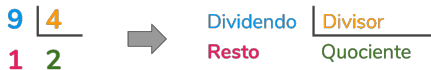
```
1 >>> 2**4
2 16
3 >>> 2.2**4
4 23.425600000000006
```

Expressões Aritméticas

O resto da divisão (inteira) de duas expressões:

- *expressão* % *expressão*

```
1 >>> 9%4
2 1
```



$$\text{Dividendo} = \text{Divisor} * \text{Quociente} + \text{Resto}$$

Mais sobre o operador % em Python

- O tipo retornado depende dos operandos.

	Python
6 % 4	2
6.0 % 4.0	2.0
6.0 % 4	2.0
6 % 4.0	2.0

- 1 Saída de dados: `print()`
- 2 Exemplo: Conversão de Fahrenheit para Celsius
- 3 Entrada de dados: `input()`
- 4 Conversão de Tipos
- 5 Operadores Aritméticos e Expressões
- 6 Precedência de operadores
- 7 Referências

Precedência de operadores

Deve-se **tomar cuidado** ao agrupar expressões com **mais de um operador**.

- Exemplo:

```
1 a = 3
2 b = 6
3 c = a + b * 2
4 print(c)
```

- O código calcula $a + (b \times 2)$.

Precedência

A **precedência** de operadores é a **ordem** na qual os operadores serão avaliados primeiro.

- Em **Python**, os operadores são avaliados na seguinte ordem:
 - 1 ******
 - 2 *****, **/**, **//**, na ordem em que aparecerem na expressão.
 - 3 **%**
 - 4 **+** e **-**, na ordem em que aparecerem na expressão.
- Podemos alterar a ordem de execução utilizando parênteses **()**

Em geral, a precedência segue as regras da matemática.

Precedência

Passo 1. $y = 2 * 5 * 5 + 3 * 5 + 7;$ (Multiplicação mais à esquerda)

$2 * 5$ é 10

Passo 2. $y = 10 * 5 + 3 * 5 + 7;$ (Multiplicação mais à esquerda)

$10 * 5$ é 50

Passo 3. $y = 50 + 3 * 5 + 7;$ (Multiplicação antes da adição)

$3 * 5$ é 15

Passo 4. $y = 50 + 15 + 7;$ (Adição mais à esquerda)

$50 + 15$ é 65

Passo 5. $y = 65 + 7;$ (Última adição)

$65 + 7$ é 72

Passo 6. $y = 72$ (Última operação — coloca **72** em **y**)

Fim

Dúvidas?

- 1 Saída de dados: `print()`
- 2 Exemplo: Conversão de Fahrenheit para Celsius
- 3 Entrada de dados: `input()`
- 4 Conversão de Tipos
- 5 Operadores Aritméticos e Expressões
- 6 Precedência de operadores
- 7 Referências

- ① Materiais adaptados dos slides do Prof. Eduardo C. Xavier, da Universidade Estadual de Campinas.