

N_ALG PROG_A2 - Texto de apoio

Site: [EAD Mackenzie](#)

Tema: ALGORITMOS E PROGRAMAÇÃO I {TURMA 01D} 2023/1

Livro: N_ALG PROG_A2 - Texto de apoio

Impresso por: CAIO FRESSATTI PINHEIRO .

Data: quinta, 2 fev 2023, 16:51

Índice

1. ESTRUTURA SEQUENCIAL
2. OPERAÇÕES MATEMÁTICAS
3. EXPRESSÕES E TIPOS DE DADOS
4. FUNÇÕES MATEMÁTICAS
5. PROGRAMAS UTILIZANDO ESTRUTURA SEQUENCIAL
6. REFERÊNCIAS

1. ESTRUTURA SEQUENCIAL

Saída de Dados

Os programas que escrevermos poderão obter e apresentar dados aos usuários.

Já vimos, em exemplos anteriores, a função `print` (saída de dados), que é usada para exibir informações na tela.

```
>>> print("Meu primeiro programa em Python!")
Meu primeiro programa em Python!
>>>
```

Ela permite que seja exibida uma string, o conteúdo de uma variável, uma expressão matemática ou booleana.

```
>>> a=10
>>> a
10
>>> print(a)
10
>>> print("a = ", a)
a = 10
>>> print("a = ", a, " a+ 2 = ", a+2)
a = 10 a+ 2 = 12
>>>
~~~
```

Exemplos

```
a=1
b=2
#exemplo 1
print ('O número que está em a é %d e em b é %d' %(a,b))

#exemplo 2
print ('O número que está em a é %.2f e em b é %.2f' %(a,b))

#exemplo 3
print ('O número que está em a é', a,'e em b é', b)

#exemplo 4
print('O número que está em a é {} e em b é {}'.format(a,b))

#exemplo 5
print(f'O número que está em a é {a} e em b é {b}')

a=1.0
b=2.0
#exemplo 6
print(f'O número que está em a é {a} e em b é {b}')
```

Entrada de Dados

Se quisermos pedir ao usuário que forneça um valor, via teclado, usaremos a função `input`.

```
>>> input()
```

Nesse caso, um cursor será exibido, aguardando a digitação do dado (qualquer dado poderá ser digitado!).

Para que o usuário saiba o que deve ser informado, podemos exibir uma mensagem da seguinte forma:

```
>>> input("Nome: ")
Nome:
```

Aqui, o cursor será exibido logo após o texto Nome: aguardando a digitação (agora o usuário tem uma ideia do que deve ser digitado!).

Para guardar o dado digitado que será usado posteriormente, faremos a entrada de dados ser atribuída a uma variável.

```
>>>
>>> nome= input('Digite no nome da faculdde:')
Digite no nome da faculdde:Faculdade de Computação e Informática
>>> print(nome)
Faculdade de Computação e Informática
>>>
```

Observe: Os dados digitados no teclado são sempre do tipo string. Para entrada de valor numérico, o dado deverá ser convertido para o tipo apropriado.

```
>>> idade = int(input("Digite sua idade: "))
Digite sua idade: 20
>>> print(idade)
20
>>> idade
20
```

Caso queira converter o dado de entrada em um número de ponto flutuante, utilize **float**.

```
>>> altura = float(input("Digite sua altura: "))
Digite sua altura: 1.62
>>> print(altura)
1.62
>>> altura
1.62
```

Para saber qual é o tipo de dado armazenado em uma variável, use a função **type**.

```
>>> type(idade)
<class 'int'>
>>> type(altura)
<class 'float'>
>>> type(nome)
<class 'str'>
```

2. OPERAÇÕES MATEMÁTICAS

Agora que conhecemos os tipos numéricos e string em Python, vejamos como fazer operações com esses tipos.

Um **operador** é um símbolo que representa a operação que pode ser realizada em um ou mais operandos. Operadores que atuam sobre um operando são chamados **operadores unários**. Operadores que atuam sobre dois operandos são chamados de **operadores binários**.

Operadores Aritméticos

Operadores	Significado	Exemplo	Resultado
-x	negação	-10	-10
x + y	soma	2 + 4	6
x - y	subtração	2 - 4	-2
x * y	multiplicação	2 * 4	8
x / y	divisão	25 / 10	2.5
x // y	divisão truncada	25 // 10	2
		25 // 10.0	2.0
x % y	Módulo ou resto de divisão	25 % 10	5
x ** y	exponenciação	2 ** 4	16

O operador (-) pode ser um operador unário (negação) ou um operador binário (subtração). Todos os outros operadores são binários.

A exponenciação pode ser utilizada tanto para números inteiros quanto para números em ponto flutuante, tanto na base quanto no expoente.

Exemplos:

```
>>> 2**4
16
>>> 2.5**1.2
3.002811084953578
```

Python possui dois operadores para divisão, a divisão dita normal e a divisão truncada, em que o tipo resultante depende dos operandos. Caso os operandos sejam inteiros, o resultado da divisão truncada será um inteiro. Caso ao menos um dos operandos seja float, o resultado será um float.

Exemplo:

```
a= 5
b = 2
c = a+b
print(f'Soma = {c}')
c = a - b
print(f'Subtração = {c}')
c = a * b
print(f'Multiplicação = {c}')
c = a/b
print(f'Divisão = {c}')
c = a//b
print(f'Quociente da Divisão = {c}')
c = a%b
print(f'Resto da Divisão = {c}')
```

Saída:

```
= RESTART: C:/Regiane/Mackenzie  
os - função-matemática/exemplo_  
Soma = 7  
Subtração = 3  
Multiplicação = 10  
Divisão = 2.5  
Quociente da Divisão = 2  
Resto da Divisão = 1  
>>> |
```

3. EXPRESSÕES E TIPOS DE DADOS

Uma **expressão** é uma combinação de símbolos que avalia um valor e consiste de uma combinação de operadores e operandos. Uma expressão pode conter também uma variável.

Exemplo: $4 + (3 * k)$

As expressões que avaliam valores do tipo numérico são chamadas de **expressões aritméticas**.

Para avaliarmos uma expressão, é importante sabermos a **precedência dos operadores**. O natural seria considerar a precedência da matemática, porém cada linguagem de programação tem sua própria regra. Pode ser ou não igual à da matemática.

A **associatividade do operador** é a ordem em que os operadores são aplicados quando têm o **mesmo nível de prioridade**.

Operador	Associatividade
**	Direita para esquerda
- (negação)	Esquerda para direita
* / // e %	Esquerda para direita
+ e - (subtração)	Esquerda para direita

4. FUNÇÕES MATEMÁTICAS

Anteriormente, vimos as expressões matemáticas compostas por constantes, variáveis e operadores aritméticos unários e binários.

Agora, vamos inserir funções e constantes matemáticas definidas em um módulo chamado **math**.

Para usar as funções do módulo, é necessário que ele seja importado da biblioteca padrão do Python: **import math**.

Nesse módulo, encontraremos um conjunto variado de funções como as trigonométricas (por exemplo: sin, cos, tan etc.) e logarítmicas (por exemplo: log) e as constantes matemáticas (por exemplo: pi).

Veja a seguir a tabela com algumas funções:

Sintaxe	Descrição
<code>math.acos(x)</code>	retorna o arco cosseno de x
<code>math.asin(x)</code>	retorna o arco seno de x
<code>math.atan(x)</code>	retorna o arco tangente de x
<code>math.ceil(x)</code>	retorna o menor inteiro maior ou igual a x como int
<code>math.cos(x)</code>	retorna o cosseno de x
<code>math.degrees(r)</code>	converte radianos para graus
Sintaxe	Descrição
<code>math.e</code>	constante e (aproximadamente 2.7182818284590451)
<code>math.exp(x)</code>	retorna o exponencial de x (e^x)
<code>math.fabs(x)</code>	retorna o valor absoluto de x
<code>math.factorial(x)</code>	retorna $x!$
<code>math.floor(x)</code>	retorna o maior inteiro menor ou igual a x como um int
<code>math.log(x,b)</code>	retorna $\log_b x$ (se b for omitido, retorna log x na base e)
<code>math.log10(x)</code>	retorna $\log_{10} x$
<code>math.modf(x)</code>	retorna a parte fracionária e a parte inteira como dois floats
<code>math.pi</code>	constante π (aproximadamente 3.1415926535897931)
<code>math.pow(x,y)</code>	retorna x^y
<code>math.radians(g)</code>	converte graus para radianos
<code>math.sin(x)</code>	retorna o seno de x
<code>math.sqrt(x)</code>	retorna a raiz quadrada de x
<code>math.tan(x)</code>	retorna a tangente de x
<code>math.trunc(x)</code>	retorna a parte inteira de x como um int; igual a <code>int(x)</code>

*Essa tabela de funções não está completa, portanto, não esgota o assunto, sendo possível na documentação da linguagem conhecer as outras funções matemáticas disponíveis em Python.

Exemplo:

```
import math
x = 3.954
print(f'Valor original de x é {x}')
y = math.ceil(x) # arredonda para cima
print(f'Valor arredondado de {x} para cima é {y}')
z = math.floor(x) # arredonda para baixo
print(f'valor arredondado de {x} para baixo é {z}')
res1 = math.trunc(x)
print(f'Valor truncado de {x} é {res1}')
print(f'Parte inteira e parte fracionária de {x} é igual {math.modf(x)}')

a = -20
res = math.fabs(a)
print(f'Valor absoluto de {a} é igual a {res}')
```

Saída:


```
= RESTART: C:\Regiane\Mackenzie\Moodle_1sem_2020\AlgoritmosI\1-Introdução\Exe  
os - função-matemática\exemploArredonda.py  
Valor original de x é 3.954  
Valor arredondado de 3.954 para cima é 4  
valor arredondado de 3.954 para baixo é 3  
Valor truncado de 3.954 é 3  
Parte inteira e parte fracionária de 3.954 é igual (0.9540000000000002, 3.0)  
Valor absoluto de -20 é igual a 20.0  
>>> |
```

Outro Exemplo:

```
import math  
num = int(input("Número:"))  
res = math.factorial(num)  
print('Fatorial = ', res)|
```

Saída:

```
= RESTART: C:\Regi  
os - função-matemá  
Número:4  
Fatorial = 24  
>>>
```

5. PROGRAMAS UTILIZANDO ESTRUTURA SEQUENCIAL

Nos programas que construímos anteriormente, sempre utilizamos a mesma sequência de instruções: receber dados iniciais, realizar operações matemáticas e apresentar resultados finais.

Essa estrutura de controle, caracterizada pela execução das instruções na ordem em que foram escritas é denominada **Estrutura Sequencial**.

Exemplos

1. Escreva um programa que receba o raio de uma esfera, calcule e apresente o seu volume

$$v = \frac{4}{3}\pi r^3$$

```
import math
print('Cálculo do volume da esfera')
raio = float(input('Raio:'))
vol = 4/3* math.pi* math.pow(raio,3)
print(f'Volume = {vol}')
```

Saída:

```
= RESTART: C:\Regiane\Mackenzie\2sem202
o-matemática\volume.py
Raio:2
Volume da esfera: 33.510321638291124
>>> |
```

2. Desenvolva um programa para converter a temperatura de Celsius para Fahrenheit. A fórmula de conversão é: $c = 5 \cdot (f - 32) / 9$

```
# c = 5*(f-32)/9
f = int(input('Temperatura em Fahrenheit: '))
c = 5*(f-32)/9
print('Temperatura em celsius: ', int(c), '°')
print('Temperatura em celsius: ', c)
```

Saída:

```
= RESTART: C:\Regiane\Mackenzie\2sem2020\Algo:
o-matemática\conversao.py
Temperatura em Fahrenheit: 62
Temperatura em celsius: 16 °
Temperatura em celsius: 16.666666666666668
>>>
```

3. Três amigos, Carlos André e Felipe, decidiram rachar igualmente a conta em um bar. Faça um programa para ler o valor total da conta e imprimir quanto cada um deve pagar, mas faça com que Carlos e André não paguem centavos. Por exemplo: uma conta de R\$101,53 resulta em R\$33,00 para Carlos, R\$33,00 para André e R\$35,53 para Felipe.

```
valor= float(input('Valor da conta:'))
divisao = valor/3
cada = int(divisao)
sobra = valor - 2 * cada
print('Carlos irá pagar R$ %.2f ' %cada)
print('André irá pagar R$ %.2f ' %cada)
print('Felipe irá pagar R$ %.2f ' %sobra)
```

Saída:

```
>>>
= RESTART: C:/Regiane/Mackenzie/
o-matemática/conta_amigos.py
Valor da conta:101.53
Carlos irá pagar R$ 33.00
André irá pagar R$ 33.00
Felipe irá pagar R$ 35.53
>>>
```

$$f = (c * 9/5) + 32$$

6. REFERÊNCIAS

DIERBACH, C. *Introduction to Computer Science Using Python: A Computational Problem Solving Focus*. New York: Wiley, 2012.

MENEZES, N. N. C. *Introdução à Programação com Python: algoritmos e lógica de programação para iniciantes*. São Paulo: Novatec, 2014.