

PEDRO MACIEL XAVIER

COMPUTAÇÃO I

Python 3

ESTUDO DIRIGIDO

UFRJ

2021

Prefácio

A intenção deste livro é reunir exercícios interessantes para um curso de introdução à computação na forma de pequenos projetos. Conta com enunciados elaborados e desenvolvimentos longos, a fim de explorar alguns dos conceitos básicos de programação.

Este texto é fruto das minhas atividades de monitoria nos cursos de Computação I (MAB125) na Universidade Federal do Rio de Janeiro (UFRJ) entre os anos de 2018 e 2019. Entre as turmas que trabalhei estão os cursos de Engenharia Naval, Engenharia de Produção e o Bacharelado em Ciências Matemáticas e da Terra (BCMT). Por conta disso, você deve encontrar referências a temas bastante diversos, dadas as inúmeras aplicações distintas que foram exercitadas por mim e pelos professores nestes diferentes contextos.

O curso é pensado para alunos que acabaram de ingressar na Graduação e que possivelmente não tiveram contato com computação anteriormente.

Vamos utilizar como referência a linguagem Python. No entanto, este material não pretende ser um curso ou referência da linguagem. Ela nos dará suporte para enunciar os fundamentos de programação, assim como noções de algoritmos e estruturas de dados. De maneira geral, as especificidades da linguagem são discutidas com maior profundidade nos tópicos finais de cada capítulo.

Além disso, deve ficar claro o meu interesse pessoal por tópicos de Matemática. Isso certamente contribuiu para a presença do assunto nos exercícios, uma vez que este conta não só com sua relação estreita com a computação como também com o meu apreço e curiosidade. A notação e os temas relacionados não devem ser um percalço para quem não tem a mesma afinidade

Petrópolis, março de 2021

Notação

Como referência, a notação matemática segue o padrão encontrado na maioria dos livros e artigos do assunto. No entanto, principalmente na hora de falar de código utilizaremos alguns recursos tipográficos para auxiliar na comunicação.

Python

Quando falando sobre os elementos da linguagem, estes aparecerão em fonte monoespaçada, tal como `x`, `y` e `math`. As palavras reservadas estão escritas em fonte alaranjada, como em `if`, `return` e `del`. Funções, tipos e exceções da biblioteca padrão são mostrados em tom violeta: `list`, `range`, `open`, etc. Por fim, temos as *strings* em verde, `'Oi Mundão'`, `"Adeus, Mundinho"`; e os comentários cor-de-vinho, `# Voltei Mundão`.

Sumário

Prefácio	i
Notação	iii
<i>Python</i>	iii
Sumário	1
Introdução	3
1 Tipos, Variáveis, Operadores e Funções	5
Introdução	5
Variáveis e atribuição	6
Funções	6
Condicionais	8
Repetição	9
Decoradores	9
Parâmetros	9
2 Estruturas e Sequências	11
Encadeamento de ideias	11
3 Números	13
Inteiros	13
Reais	13
Complexos	13
Vetores	13
4 <i>Strings</i> e Texto	15
Codificação	15
Operações em texto	15
Expressões Regulares	15

5	Entrada e Saída	17
	Interação com o usuário	17
	Arquivos	17
	A linha de comandos	17
6	Estruturas e Sequências	19
6.1	Encadeamento de ideias	19

Introdução

Capítulo 1

Tipos, Variáveis, Operadores e Funções

Introdução

É interessante, de fato, iniciar o estudo da computação através das funções. As funções trazem consigo os conceitos de tipos e variáveis de maneira muito natural. De maneira simples, uma função pode ser compreendida como a transformação dos dados de entrada nos dados de saída, cada qual com seus respectivos tipos.

A noção de tipo está também relacionada à ideia de conjunto. Ser de um determinado tipo é pertencer a um conjunto, mais especificamente ao conjunto daquele tipo. Isto fica bem claro quando se pensa nos números, embora seja um conceito adequado a uma miríade de situações.



Figura 1.1: Conjuntos numéricos

Em textos de matemática a nível superior, é comum que as funções recebam outros nomes a depender do contexto. Uma nomenclatura um tanto elucidativa

é chamar funções de *aplicações*, em inglês, *mappings*. É de bom gosto observar que as funções são *mapas* que levam de um ponto em um conjunto noutro ponto de outro conjunto.

Variáveis e atribuição

```
1 >>> x = 0
2 >>> y = 1
3 >>> x + y
4 1
```

Funções

Para definir uma função vamos utilizar o comando `def`, seguido do nome da função, dos seus parâmetros e, por fim, o código que deve executar.

```
1 >>> def f(x, y):
2 ...     return x + y
3 ...
4 >>> z = f(2, 3)
5 >>> z
6 5
```

1.1 Cálculo I - Limites

Seja $f : \mathbb{R} \rightarrow \mathbb{R}$ uma função diferenciável.

1.2 Música I - As notas e os sons

Cada nota musical corresponde a uma frequência distinta (em Hz). Tomando o lá central (A4) como referência, em 440Hz, podemos calcular a frequência das outras notas com base na distância relativa a essa nota.

$$f(n) = 440 \times 2^{(n/12)}$$

Na tabela abaixo, vemos as notas musicais, seus símbolos, e a distância em semitons¹ para o lá central (A4).

¹Dois semitons equivalem a um tom. No violão, cada casa de uma mesma corda está a um semitom da casa adjacente. No piano, quando há uma tecla preta entre as brancas, há uma distância de um tom entre elas. Quando a tecla preta não está, a distância é de meio tom, ou um semitom.

Símbolo		F3	G3	A4	B4	C4	D4	E4	F4	G4	
Nome	...	fá	sol	lá	si	dó	ré	mi	fá	sol	...
Semitons		-4	-2	0	2	3	5	7	8	10	

Você pode notar que a cada 12 semitons, a nota se repete com o dobro da frequência. Chamamos este intervalo entre notas de oitava. Na notação acima, a cada letra indica uma nota diferente, enquanto o número diz a oitava em que ela se encontra.

Proposta

Faça uma função $f(n)$ que retorne a frequência em Hertz de uma nota que se encontra a n semitons de distância do lá central. Arredonde o resultado para o número inteiro mais próximo usando as funções `round` e `int`.

```
1 >>> f(0), f(2), f(3)
2 (440, 494, 523)
3 >>> f(12) # frequência dobra ...
4 880
```

1.3 Coordenadas polares

Estamos acostumados a pensar em coordenadas cartesianas na hora de descrever a geometria de um determinado objeto. No entanto, o sistema de coordenadas deve ser escolhido conforme o cenário em que se está trabalhando.

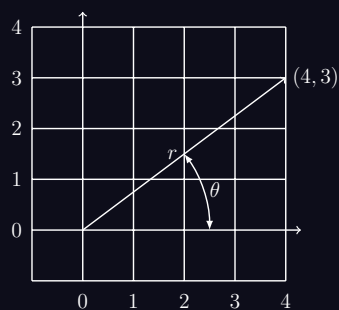


Figura 1.2: Coordenadas cartesianas e polares.

O ponto $(4, 3)$, quando escrito em coordenadas polares, nos dá:

$$r = \sqrt{4^2 + 3^2} = \sqrt{16 + 9} = \sqrt{25} = 5$$

$$\theta = \arctan \frac{3}{4} = 0.6435 \text{ rad} \approx 36.87^\circ$$

Proposta

Construa duas funções: `polar(x, y)` levará um ponto em coordenadas cartesianas (x, y) para a forma polar (r, θ) e `cart(r, theta)`, que fará o caminho contrário.

```
1 >>> import math
2 >>> polar(-1, 0)
3 (1.0, 3.141592653589793)
4 >>> cart(2, math.pi)
5 (-2.0, 0.0)
```

Condicionais

Uma das ferramentas mais simples e poderosas dos computadores é a sua capacidade de tomar diferentes atitudes mediante uma condição.

1.4 Meia-entrada

A Lei Federal nº 12933/2013, também conhecida como Lei da Meia-Entrada, garante o benefício do pagamento de Meia-Entrada para estudantes, pessoas com deficiência e jovens, de baixa renda, com idade entre 15 e 29 anos. Já a Lei Federal nº 10741/2003, mais conhecida como Estatuto do Idoso, as pessoas com idade igual ou superior a 60 anos tem direito à Meia-Entrada para eventos artísticos e de lazer. Aqui no estado do Rio de Janeiro, contamos ainda com a Lei Estadual RJ nº 3.364/00, que garante o benefício a todos os menores de 21 anos.

Proposta

Escreva a função `meia_entrada` que receba os parâmetros:

1. `idade (int)`
2. `estudante (bool)`
3. `deficiencia (bool)`
4. `baixa_renda (bool)`

informando com `True` ou `False` se a pessoa tem direito ao desconto. normal,

```
1 >>> meia_entrada(60, False, False, False)
2 True
3 >>> meia_entrada(30, True, False, False)
4 False
5 >>> meia_entrada(20, False, False, False)
6 True
```

Repetição

1.5 Sequência de *Collatz*

A sequência de *Collatz* é obtida aplicando sucessivamente a função

$$f(n) = \begin{cases} 3n + 1, & \text{se } n \text{ for ímpar} \\ n \div 2, & \text{se } n \text{ for par} \end{cases}$$

Por exemplo, começamos com $n = 26$. Após sucessivas aplicações temos:

$$26 \rightarrow 13 \rightarrow 40 \rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$$

Isso nos dá uma sequência com 11 números. 40 é maior que 26, mas sua sequência só teria 9 números.

Ainda não se sabe se todos os números induzem uma sequência que termina em 1. No entanto, até agora não foi encontrado um número sequer em que isso não tenha acontecido!

Proposta

Faça uma função que calcule o comprimento da sequência gerada a partir de um número natural n qualquer.

```
1 >>> collatz(26)
2 11
3 >>> collatz(40)
4 9
5 >>> collatz(1)
6 1
```

1.6 *Chatbot I*

Existe uma história corrente de que o grande motor da inteligência artificial em escala industrial é o uso indiscriminado do comando `if` e seus derivados.

Decoradores

Agora é hora de voltar a falar em funções.

Parâmetros

Agora é hora de voltar a falar em funções.

Capítulo 2

Estruturas e Sequências

Encadeamento de ideias

Capítulo 3

Números

Inteiros

Reais

Complexos

Vetores

Capítulo 4

Strings e Texto

Codificação

Operações em texto

Expressões Regulares

Capítulo 5

Entrada e Saída

Interação com o usuário

Arquivos

A linha de comandos

Capítulo 6

Estruturas e Sequências

6.1 Encadeamento de ideias

