

NÃO PODE SER VENDIDO

# INTRODUÇÃO À PROGRAMAÇÃO COM PYTHON ENUNCIADOS DOS EXERCÍCIOS

4ª edição - atualização do 26 de março 2024

**Nilo Ney Coutinho Menezes**

livro\_de\_python@nilo.pro.br

Telegram: <https://t.me/niloprogramacao>



**INDICE**

		Exercício 04-07	12
Sobre o livro	6	Exercício 04-08	13
Introdução	8	Exercício 04-09	13
<b>Exercício 02-01</b>	<b>8</b>	Exercício 04-10	13
Exercício 02-02	8	Exercício 04-11	13
Exercício 02-03	8	Exercício 04-12	13
Exercício 02-04	9	Exercício 04-13	14
Exercício 02-05	9	Exercício 04-14	14
Exercício 02-06	9	Exercício 04-15	14
Exercício 02-07	9	Exercício 04-16	14
<b>Exercício 03-01</b>	<b>9</b>	<b>Exercício 05-01</b>	<b>15</b>
Exercício 03-02	9	Exercício 05-02	15
Exercício 03-03	10	Exercício 05-03	15
Exercício 03-04	10	Exercício 05-04	15
Exercício 03-05	10	Exercício 05-05	15
Exercício 03-06	10	Exercício 05-06	15
Exercício 03-07	10	Exercício 05-07	15
Exercício 03-08	10	Exercício 05-08	16
Exercício 03-09	11	Exercício 05-09	16
Exercício 03-10	11	Exercício 05-10	16
Exercício 03-11	11	Exercício 05-11	16
Exercício 03-12	11	Exercício 05-12	16
Exercício 03-13	11	Exercício 05-13	16
Exercício 03-14	11	Exercício 05-14	17
Exercício 03-15	11	Exercício 05-15	17
<b>Exercício 04-01</b>	<b>12</b>	Exercício 05-16	17
Exercício 04-02	12	Exercício 05-17	17
Exercício 04-03	12	Exercício 05-18	17
Exercício 04-04	12	Exercício 05-19	17
Exercício 04-05	12	Exercício 05-20	18
Exercício 04-06	12	Exercício 05-21	18

Exercício 05-22	18	Exercício 07-02	23
Exercício 05-23	18	Exercício 07-03	23
Exercício 05-24	18	Exercício 07-04	23
Exercício 05-25	18	Exercício 07-05	23
Exercício 05-26	18	Exercício 07-06	23
Exercício 05-27-a	19	Exercício 07-07	23
Exercício 05-27-b	19	Exercício 07-08	23
Exercício 06-01	19	Exercício 07-09	24
Exercício 06-02	19	Exercício 07-10	24
Exercício 06-03	19	Exercício 07-11	24
Exercício 06-04	19	Exercício 07-12	25
Exercício 06-05	19	Exercício 08-01	25
Exercício 06-06	20	Exercício 08-02	25
Exercício 06-07	20	Exercício 08-03	25
Exercício 06-08	20	Exercício 08-04	25
Exercício 06-09	20	Exercício 08-05	26
Exercício 06-10	20	Exercício 08-06	26
Exercício 06-11	20	Exercício 08-07	26
Exercício 06-12	21	Exercício 08-08	26
Exercício 06-13	21	Exercício 08-09	26
Exercício 06-14	21	Exercício 08-10	26
Exercício 06-15	21	Exercício 08-11	26
Exercício 06-16	21	Exercício 08-12	27
Exercício 06-17	21	Exercício 08-13-a	27
Exercício 06-18	21	Exercício 08-13-b	27
Exercício 06-19	22	Exercício 08-14	27
Exercício 06-20-a	22	Exercício 08-15	27
Exercício 06-20-b	22	Exercício 08-16	27
Exercício 06-21	22	Exercício 08-17	28
Exercício 06-22	22	Exercício 08-18	28
Exercício 07-01	22	Exercício 08-19	28

Exercício 08-20	28	Exercício 09-27	33
Exercício 08-21	28	Exercício 09-28	33
Exercício 08-22	29	Exercício 09-29	33
Exercício 08-99	29	Exercício 09-30	33
Exercício 08-991	29	Exercício 09-31	33
Exercício 09-01	29	Exercício 09-32	33
Exercício 09-02	29	Exercício 09-33	33
Exercício 09-03	29	Exercício 09-34	33
Exercício 09-04	29	Exercício 09-35	34
Exercício 09-05	30	Exercício 09-36	34
Exercício 09-06	30	Exercício 09-37	34
Exercício 09-07	30	Exercício 09-38	34
Exercício 09-08	30	Exercício 09-39	34
Exercício 09-09	30	Exercício 09-40	34
Exercício 09-10	30	Exercício 09-41	34
Exercício 09-11	30	Exercício 09-42	34
Exercício 09-12	31	Exercício 09-43	35
Exercício 09-13	31	Exercício 09-44	35
Exercício 09-14	31	Exercício 10-01	35
Exercício 09-15	31	Exercício 10-02	35
Exercício 09-16	31	Exercício 10-03	35
Exercício 09-17	31	Exercício 10-04	35
Exercício 09-18	31	Exercício 10-05	36
Exercício 09-19	32	Exercício 10-06	36
Exercício 09-20	32	Exercício 10-07	36
Exercício 09-21	32	Exercício 10-08	36
Exercício 09-22	32	Exercício 10-09	36
Exercício 09-23	32	Exercício 10-10	36
Exercício 09-24	32	Exercício 10-11	36
Exercício 09-25	32	Exercício 10-12	36
Exercício 09-26	32	Exercício 10-13	37

Exercício 10-14	37
Exercício 10-15	37
Exercício 11-01	37
Exercício 11-02	37
Exercício 11-03	37
Exercício 11-04	37
Exercício 11-05	38
Exercício 11-06	38
Exercício 12-01	38
Exercício 12-02	38
Exercício 12-03	38
Exercício 12-04	38
Exercício 12-05	38
Exercício 12-06	39
Exercício 12-07	39
Exercício 12-08	39
Exercício 12-09	39
Exercício 12-10	39
Exercício 13-01	39
Exercício 13-02	40
Exercício 13-03	40
Exercício 13-04	40
Exercício 13-05	40
Exercício 13-06	40

## Exercícios Enunciados

Nilo Ney Coutinho Menezes <livro\_de\_python@nilo.pro.br>

2024-03-25

### SOBRE O LIVRO

Este livro é orientado ao iniciante em programação. Os conceitos básicos de programação, como expressões, variáveis, repetições, decisões, listas, dicionários, conjuntos, funções, arquivos, classes, objetos, banco de dados com SQLite 3, expressões regulares e interfaces gráficas com tkinter são apresentados um a um com exemplos e exercícios. A obra visa a explorar a programação de computadores como ferramenta do dia a dia. Ela pode ser lida durante um curso de introdução à programação de computadores e usada como guia de estudo para autodidatas. Para aproveitamento pleno do conteúdo, apenas conhecimentos básicos de informática, como digitar textos, abrir e salvar arquivos, são suficientes. Todo software utilizado no livro pode ser baixado gratuitamente, sendo executado em Windows, Linux e Mac OS X.

Embora a linguagem Python (versão +3.12) seja muito poderosa e repleta de recursos modernos de programação, este livro não pretende ensinar a linguagem em si, mas ensinar a programar. Alguns recursos da linguagem não foram utilizados para privilegiar os exercícios de lógica de programação e oferecer uma preparação mais ampla ao leitor para outras linguagens. Essa escolha não impediu a apresentação de recursos poderosos da linguagem, e, embora o livro não seja fundamentalmente uma obra de referência, o leitor encontrará várias notas e explicações de características específicas do Python, além da lógica de programação.



Título: Introdução à Programação com Python

Autor: Nilo Ney Coutinho Menezes

Edição: Quarta

ISBN: 978-8575228869

Editora: Novatec

Ano: 2024

Páginas: 552

Para comprar o livro na Amazon, visite o link ou escaneie o qrcode com seu celular:



Na Amazon: <https://amzn.to/3Pjg4MZ>

Para comprar o livro na Editora Novatec, visite o link ou escaneie o qrcode com seu celular:



Na Editora Novatec: <http://www.novatec.com.br/livros/introducao-python-4ed/>

## INTRODUÇÃO

Este documento foi criado para disponibilizar todos os exercícios resolvidos do livro em um só arquivo. O site do livro pode ser acessado em <https://python.nilo.pro.br> ou pelo qr-code abaixo:



### *Site do livro*

Se você não conhece o livro, visite o site web e tenha acesso às listagens, exercícios resolvidos, dúvidas e correções (errata). Para comprar o livro na Amazon, visite o link ou escaneie o qr-code com seu celular:

Lembre-se que os exercícios foram feitos para que você aprenda sozinho. Não olhe a resposta antes de tentar sozinho algumas vezes ;-D.

Não pode ser vendido.

## EXERCÍCIO 02-01

Converta as seguintes expressões matemáticas para que possam ser calculadas usando o interpretador Python.  $10 + 20 \times 30$   $42 \div 30$   $(94 + 2) \times 6 - 1$

## EXERCÍCIO 02-02

Digite a seguinte expressão no interpretador:  $10 \% 3 * 10 ** 2 + 1 - 10 * 4 / 2$  Tente resolver o mesmo cálculo, usando apenas lápis e papel. Observe como a prioridade das operações é importante.

## EXERCÍCIO 02-03

Faça um programa que exiba seu nome na tela.

**EXERCÍCIO 02-04**

Escreva um programa que exiba o resultado de  $2a \times 3b$ , em que  $a$  vale 3 e  $b$  vale 5.

**EXERCÍCIO 02-05**

Escreva um programa que calcule a soma de três variáveis e imprima o resultado na tela.

**EXERCÍCIO 02-06**

Modifique o Programa 2.2, de forma que ele calcule um aumento de 15% para um salário de R\$ 750.

**EXERCÍCIO 02-07**

Usando as propriedades da divisão e da multiplicação, tente entender como estes resultados são iguais:

$$0.2 * 6 + 8 * 0.3 + 7 * 0.5 = (20 * 6 + 8 * 30 + 7 * 50) / 100$$

**EXERCÍCIO 03-01**

Complete a tabela a seguir, marcando inteiro ou ponto flutuante dependendo do número apresentado.

Número	Tipo numérico	
5	<input type="checkbox"/> inteiro	<input type="checkbox"/> ponto flutuante
5.0	<input type="checkbox"/> inteiro	<input type="checkbox"/> ponto flutuante
4.3	<input type="checkbox"/> inteiro	<input type="checkbox"/> ponto flutuante
-2	<input type="checkbox"/> inteiro	<input type="checkbox"/> ponto flutuante
100	<input type="checkbox"/> inteiro	<input type="checkbox"/> ponto flutuante
1.333	<input type="checkbox"/> inteiro	<input type="checkbox"/> ponto flutuante

**EXERCÍCIO 03-02**

Complete a tabela a seguir, respondendo True ou False. Considere  $a = 4$ ,  $b = 10$ ,  $c = 5.0$ ,  $d = 1$  e  $f = 5$ .

Expressão	Resultado	Expressão	Resultado
$a == c$	<input type="checkbox"/> True <input type="checkbox"/> False	$b > a$	<input type="checkbox"/> True <input type="checkbox"/> False
$a < b$	<input type="checkbox"/> True <input type="checkbox"/> False	$c >= f$	<input type="checkbox"/> True <input type="checkbox"/> False
$d > b$	<input type="checkbox"/> True <input type="checkbox"/> False	$f >= c$	<input type="checkbox"/> True <input type="checkbox"/> False
$c != f$	<input type="checkbox"/> True <input type="checkbox"/> False	$c <= c$	<input type="checkbox"/> True <input type="checkbox"/> False
$a == b$	<input type="checkbox"/> True <input type="checkbox"/> False	$c <= f$	<input type="checkbox"/> True <input type="checkbox"/> False
$c < d$	<input type="checkbox"/> True <input type="checkbox"/> False		

## EXERCÍCIO 03-03

Complete a tabela a seguir utilizando  $a = \text{True}$ ,  $b = \text{False}$  e  $c = \text{True}$ .

Expressão	Resultado	Expressão	Resultado
$a \text{ and } a$	[ ] True	$a \text{ or } c$	[ ] True [ ] False
$b \text{ and } b$	[ ] True [ ] False	$b \text{ or } c$	[ ] True [ ] False
$\text{not } c$	[ ] True [ ] False	$c \text{ or } a$	[ ] True [ ] False
$\text{not } b$	[ ] True [ ] False	$c \text{ or } b$	[ ] True [ ] False
$\text{not } a$	[ ] True [ ] False	$c \text{ or } c$	[ ] True [ ] False
$a \text{ and } b$	[ ] True [ ] False	$b \text{ or } b$	[ ] True [ ] False
$b \text{ and } c$	[ ] True [ ] False		

## EXERCÍCIO 03-04

Escreva uma expressão para determinar se uma pessoa deve ou não pagar imposto. Considere que pagam imposto pessoas cujo salário é maior que R\$ 1.200,00.

## EXERCÍCIO 03-05

Calcule o resultado da expressão  $A > B \text{ and } C \text{ or } D$ , utilizando os valores da tabela a seguir.

A	B	C	D	Resultado
1	2	True	False	
10	3	False	False	
5	1	True	True	

## EXERCÍCIO 03-06

Escreva uma expressão que será utilizada para decidir se um aluno foi ou não aprovado. Para ser aprovado, todas as médias do aluno devem ser maiores que 7. Considere que o aluno cursa apenas três matérias, e que a nota de cada uma está armazenada nas seguintes variáveis: `matéria1`, `matéria2` e `matéria3`.

## EXERCÍCIO 03-07

Faça um programa que peça dois números inteiros. Imprima a soma desses dois números na tela.

## EXERCÍCIO 03-08

Escreva um programa que leia um valor em metros e o exiba convertido em milímetros.

## EXERCÍCIO 03-09

Escreva um programa que leia a quantidade de dias, horas, minutos e segundos do usuário. Calcule o total em segundos.

## EXERCÍCIO 03-10

Faça um programa que calcule o aumento de um salário. Ele deve solicitar o valor do salário e a porcentagem do aumento. Exiba o valor do aumento e do novo salário.

## EXERCÍCIO 03-11

Faça um programa que solicite o preço de uma mercadoria e o percentual de desconto. Exiba o valor do desconto e o preço a pagar.

## EXERCÍCIO 03-12

Escreva um programa que calcule o tempo de uma viagem de carro. Pergunte a distância a percorrer e a velocidade média esperada para a viagem.

## EXERCÍCIO 03-13

Escreva um programa que converta uma temperatura digitada em °C em °F. A fórmula para essa conversão é:

$$F = \frac{9 \times C}{5} + 32$$

## EXERCÍCIO 03-14

Escreva um programa que pergunte a quantidade de km percorridos por um carro alugado pelo usuário, assim como a quantidade de dias pelos quais o carro foi alugado. Calcule o preço a pagar, sabendo que o carro custa R\$ 60 por dia e R\$ 0,15 por km rodado.

## EXERCÍCIO 03-15

Escreva um programa para calcular a redução do tempo de vida de um fumante. Pergunte a quantidade de cigarros fumados por dia e quantos anos ele já fumou. Considere que um fumante perde 10 minutos de vida a cada cigarro, e calcule quantos dias de vida um fumante perderá. Exiba o total em dias.

## EXERCÍCIO 04-01

Analise o Programa 4.1. Responda o que acontece se o primeiro e o segundo valores forem iguais? Explique.

## EXERCÍCIO 04-02

Escreva um programa que pergunte a velocidade do carro de um usuário. Caso ultrapasse 80 km/h, exiba uma mensagem dizendo que o usuário foi multado. Nesse caso, exiba o valor da multa, cobrando R\$ 5 por km acima de 80 km/h.

## EXERCÍCIO 04-03

Escreva um programa que leia três números e que imprima o maior e o menor.

## EXERCÍCIO 04-04

Escreva um programa que pergunte o salário do funcionário e calcule o valor do aumento. Para salários superiores a R\$ 1.250,00, calcule um aumento de 10%. Para os inferiores ou iguais, de 15%.

## EXERCÍCIO 04-05

Execute o Programa 4.5 e experimente alguns valores. Verifique se os resultados foram os mesmos do Programa 4.2.

## EXERCÍCIO 04-06

Escreva um programa que pergunte a distância que um passageiro deseja percorrer em km. Calcule o preço da passagem, cobrando R\$ 0,50 por km para viagens de até de 200 km, e R\$ 0,45 para viagens mais longas.

## EXERCÍCIO 04-07

Analise o Programa 4.3. Faz sentido usar o else nesse programa? Explique sua resposta.

**EXERCÍCIO 04-08**

Reescreva o Programa 4.4 e calcule a conta da operadora Tchau usando else.

**EXERCÍCIO 04-09**

Rastreie o Programa 4.8. Compare seu resultado ao apresentado na Tabela 4.2.

**EXERCÍCIO 04-10**

Escreva um programa que leia dois números e que pergunte qual operação você deseja realizar. Você deve poder calcular soma (+), subtração (-), multiplicação (\*) e divisão (/). Exiba o resultado da operação solicitada.

**EXERCÍCIO 04-11**

Escreva um programa para aprovar o empréstimo bancário para compra de uma casa. O programa deve perguntar o valor da casa a comprar, o salário e a quantidade de anos a pagar. O valor da prestação mensal não pode ser superior a 30% do salário. Calcule o valor da prestação como sendo o valor da casa a comprar dividido pelo número de meses a pagar.

**EXERCÍCIO 04-12**

Escreva um programa que calcule o preço a pagar pelo fornecimento de energia elétrica. Pergunte a quantidade de kWh consumida e o tipo de instalação: R para residências, I para indústrias e C para comércios. Calcule o preço a pagar de acordo com a tabela a seguir.

Preço por tipo e faixa de consumo		
Tipo	Faixa (kWh)	Preço
Residencial	Até 500	R\$ 0,40
	Acima de 500	R\$ 0,65
Comercial	Até 1000	R\$ 0,55
	Acima de 1000	R\$ 0,60
Industrial	Até 5000	R\$ 0,55
	Acima de 5000	R\$ 0,60

## EXERCÍCIO 04-13

No programa a seguir, inverta as linhas do if e else, negando a condição. Adicione as linhas necessárias para fazê-lo funcionar em Python.

```
if a > b:
    print("a é maior que b")
else:
    print("b é maior que a")
```

## EXERCÍCIO 04-14

Reescreva o programa a seguir com if-elif-else. Adicione as linhas necessárias para fazê-lo funcionar em Python.

```
if a < 10:
    print("a é menor que 10")
if a >= 10 and a < 20:
    print("a é maior que 10 e menor que 20")
if a >= 20:
    print("a é maior que 20")
```

## EXERCÍCIO 04-15

Reescreva o programa a seguir com if-elif-else.

```
hora = int(input("Digite a hora atual:"))
if hora < 12:
    print("Bom dia!")
if hora >=12 and hora <=18:
    print("Boa tarde!")
if hora >=18:
    print("Boa noite!")
```

## EXERCÍCIO 04-16

Corrija o programa a seguir:

```
média = input("Digite sua média:")
if média < 4:
    print("Infelizmente você reprovou")
if média < 7:
    print("Você ficou de recuperação")
if média > 7:
    print("Você passou de ano")
```

## EXERCÍCIO 05-01

Modifique o programa para exibir os números de 1 a 100.

## EXERCÍCIO 05-02

Modifique o programa para exibir os números de 50 a 100.

## EXERCÍCIO 05-03

Faça um programa para escrever a contagem regressiva do lançamento de um foguete. O programa deve imprimir 10, 9, 8, ..., 1, 0 e Fogo! na tela.

## EXERCÍCIO 05-04

Modifique o programa anterior para imprimir de 1 até o número digitado pelo usuário, mas, dessa vez, apenas os números ímpares.

## EXERCÍCIO 05-05

Reescreva o programa anterior para escrever os 10 primeiros múltiplos de 3.

## EXERCÍCIO 05-06

Altere o programa anterior para exibir os resultados no mesmo formato de uma tabuada:  $2 \times 1 = 2$ ,  $2 \times 2 = 4$ , ...

## EXERCÍCIO 05-07

Modifique o programa anterior de forma que o usuário também digite o início e o fim da tabuada, em vez de começar com 1 e 10.

## EXERCÍCIO 05-08

Escreva um programa que leia dois números. Imprima o resultado da multiplicação do primeiro pelo segundo. Utilize apenas os operadores de soma e subtração para calcular o resultado. Lembre-se de que podemos entender a multiplicação de dois números como somas sucessivas de um deles. Assim,  $4 \times 5 = 5 + 5 + 5 + 5 = 4 + 4 + 4 + 4 + 4$ .

## EXERCÍCIO 05-09

Escreva um programa que leia dois números. Imprima a divisão inteira do primeiro pelo segundo, assim como o resto da divisão. Utilize apenas os operadores de soma e subtração para calcular o resultado. Lembre-se de que podemos entender o quociente da divisão de dois números como a quantidade de vezes que podemos retirar o divisor do dividendo. Logo,  $20 \div 4 = 5$ , uma vez que podemos subtrair 4 cinco vezes de 20.

## EXERCÍCIO 05-10

Modifique o programa anterior para que aceite respostas com letras maiúsculas e minúsculas em todas as questões.

## EXERCÍCIO 05-11

Escreva um programa que pergunte o depósito inicial e a taxa de juros de uma poupança. Exiba os valores mês a mês para os 24 primeiros meses. Escreva o total ganho com juros no período.

## EXERCÍCIO 05-12

Altere o programa anterior de forma a perguntar também o valor depositado mensalmente. Esse valor será depositado no início de cada mês, e você deve considerá-lo para o cálculo de juros do mês seguinte.

## EXERCÍCIO 05-13

Escreva um programa que pergunte o valor inicial de uma dívida e o juro mensal. Pergunte também o valor mensal que será pago. Imprima o número de meses para que a dívida seja paga, o total pago e o total de juros pago.

## EXERCÍCIO 05-14

Escreva um programa que leia números inteiros do teclado. O programa deve ler os números até que o usuário digite 0 (zero). No final da execução, exiba a quantidade de números digitados, assim como a soma e a média aritmética.

## EXERCÍCIO 05-15

Escreva um programa para controlar uma pequena máquina registradora. Você deve solicitar ao usuário que digite o código do produto e a quantidade comprada. Utilize a tabela de códigos a seguir para obter o preço de cada produto:

Código	Preço
1	0,50
2	1,00
3	4,00
5	7,00
9	8,00

Seu programa deve exibir o total das compras depois que o usuário digitar 0. Qualquer outro código deve gerar a mensagem de erro “Código inválido”.

## EXERCÍCIO 05-16

Execute o Programa 5.1 para os seguintes valores: 501, 745, 384, 2, 7 e 1.

## EXERCÍCIO 05-17

O que acontece se digitarmos 0 (zero) no valor a pagar?

## EXERCÍCIO 05-18

Modifique o programa para também trabalhar com notas de R\$ 100.

## EXERCÍCIO 05-19

Modifique o programa para aceitar valores decimais, ou seja, também contar moedas de 0,01, 0,02, 0,05, 0,10 e 0,50

## EXERCÍCIO 05-20

O que acontece se digitarmos 0,001 no programa anterior? Caso ele não funcione, altere-o de forma a corrigir o problema.

## EXERCÍCIO 05-21

Reescreva o Programa 5.1 de forma a continuar executando até que o valor digitado seja 0. Utilize repetições aninhadas.

## EXERCÍCIO 05-22

Escreva um programa que exiba uma lista de opções (menu): adição, subtração, divisão, multiplicação e sair. Imprima a tabuada da operação escolhida. Repita até que a opção saída seja escolhida.

## EXERCÍCIO 05-23

Escreva um programa que leia um número e verifique se é ou não um número primo. Para fazer essa verificação, calcule o resto da divisão do número por 2 e depois por todos os números ímpares até o número lido. Se o resto de uma dessas divisões for igual a zero, o número não é primo. Observe que 0 e 1 não são primos e que 2 é o único número primo que é par.

## EXERCÍCIO 05-24

Modifique o programa anterior de forma a ler um número n. Imprima os n primeiros números primos.

## EXERCÍCIO 05-25

Escreva um programa que calcule a raiz quadrada de um número. Utilize o método de Newton para obter um resultado aproximado. Sendo n o número a obter a raiz quadrada, considere a base  $b=2$ . Calcule p usando a fórmula  $p=(b+(n/b))/2$ . Agora, calcule o quadrado de p. A cada passo, faça  $b=p$  e recalcule p usando a fórmula apresentada. Pare quando a diferença absoluta entre n e o quadrado de p for menor que 0,0001.

## EXERCÍCIO 05-26

Escreva um programa que calcule o resto da divisão inteira entre dois números. Utilize apenas as operações de soma e subtração para calcular o resultado.

## EXERCÍCIO 05-27-A

Escreva um programa que verifique se um número é palíndromo. Um número é palíndromo se continua o mesmo caso seus dígitos sejam invertidos. Exemplos: 454, 10501

## EXERCÍCIO 05-27-B

Escreva um programa que verifique se um número é palíndromo. Um número é palíndromo se continua o mesmo caso seus dígitos sejam invertidos. Exemplos: 454, 10501

## EXERCÍCIO 06-01

Modifique o Programa 6.2 para ler 7 notas em vez de 5.

## EXERCÍCIO 06-02

Faça um programa que leia duas listas e que gere uma terceira com os elementos das duas primeiras.

## EXERCÍCIO 06-03

Faça um programa que percorra duas listas e gere uma terceira sem elementos repetidos.

## EXERCÍCIO 06-04

O que acontece quando não verificamos se a lista está vazia antes de chamarmos o método pop?

## EXERCÍCIO 06-05

Altere o Programa 6.7 de forma a poder trabalhar com vários comandos digitados de uma só vez. Atualmente, apenas um comando pode ser inserido por vez. Altere-o de forma a considerar operação como uma string.

Exemplo: FFFAAAS significaria três chegadas de novos clientes, três atendimentos e, finalmente, a saída do programa.

## EXERCÍCIO 06-06

Modifique o programa para trabalhar com duas filas. Para facilitar seu trabalho, considere o comando A para atendimento da fila 1; e B, para atendimento da fila 2. O mesmo para a chegada de clientes: F para fila 1; e G, para fila 2.

## EXERCÍCIO 06-07

Faça um programa que leia uma expressão com parênteses. Usando pilhas, verifique se os parênteses foram abertos e fechados na ordem correta.

Exemplo:

```
(( )) OK  
( ) ( ( ) ) OK  
( ) Erro
```

Você pode adicionar elementos à pilha sempre que encontrar abre parênteses e desempilhá-la a cada fecha parênteses. Ao desempilhar, verifique se o topo da pilha é um abre parênteses. Se a expressão estiver correta, sua pilha estará vazia no final.

## EXERCÍCIO 06-08

Modifique o primeiro exemplo (Programa 6.9) de forma a realizar a mesma tarefa, mas sem utilizar a variável `achou`. Dica: observe a condição de saída do `while`.

## EXERCÍCIO 06-09

Modifique o exemplo para pesquisar dois valores. Em vez de apenas `p`, leia outro valor `v` que também será procurado. Na impressão, indique qual dos dois valores foi achado primeiro.

## EXERCÍCIO 06-10

Modifique o programa do Exercício 6.9 de forma a pesquisar `p` e `v` em toda a lista e informando o usuário a posição onde `p` e a posição onde `v` foram encontrados.

## EXERCÍCIO 06-11

Modifique o Programa 6.6 usando `for`. Explique por que nem todos os `while` podem ser transformados em `for`.

## EXERCÍCIO 06-12

Altere o Programa 6.11 de forma a imprimir o menor elemento da lista.

## EXERCÍCIO 06-13

A lista de temperaturas de Mons, na Bélgica, foi armazenada na lista  $T = [-10, -8, 0, 1, 2, 5, -2, -4]$ . Faça um programa que imprima a menor e a maior temperatura, assim como a temperatura média.

## EXERCÍCIO 06-14

Modifique o Programa 6.13 de forma a mostrar quantos ingressos foram vendidos em cada sala. Utilize uma lista do mesmo tamanho da quantidade de salas e utilize seus elementos para contar quantos ingressos foram vendidos em cada sala. Imprima na tela o total das vendas no fim do programa.

## EXERCÍCIO 06-15

Modifique o Programa 6.13 de forma a perguntar o número de salas disponíveis no cinema, assim como a quantidade de lugares em cada uma delas.

## EXERCÍCIO 06-16

O que acontece quando a lista já está ordenada? Rastreie o Programa 6.20, mas com a lista  $L = [1, 2, 3, 4, 5]$ .

## EXERCÍCIO 06-17

O que acontece quando dois valores são iguais? Rastreie o Programa 6.20, mas com a lista  $L = [3, 3, 1, 5, 4]$ .

## EXERCÍCIO 06-18

Modifique o Programa 6.20 para ordenar a lista em ordem decrescente.  $L = [1, 2, 3, 4, 5]$  deve ser ordenada como  $L = [5, 4, 3, 2, 1]$ .

## EXERCÍCIO 06-19

Altere o Programa 6.22 de forma a solicitar ao usuário o produto e a quantidade vendida. Verifique se o nome do produto digitado existe no dicionário, e só então efetue a baixa em estoque.

## EXERCÍCIO 06-20-A

Escreva um programa que gere um dicionário, em que cada chave seja um caractere, e seu valor seja o número desse caractere encontrado em uma frase lida. Exemplo: O rato → { "O":1, "r":1, "a":1, "t":1, "o":1 }

## EXERCÍCIO 06-20-B

Escreva um programa que gere um dicionário, em que cada chave seja um caractere, e seu valor seja o número desse caractere encontrado em uma frase lida. Exemplo: O rato → { "O":1, "r":1, "a":1, "t":1, "o":1 }

## EXERCÍCIO 06-21

Escreva um programa que compare duas listas. Utilizando operações com conjuntos, imprima: • os valores comuns às duas listas • os valores que só existem na primeira • os valores que existem apenas na segunda • uma lista com os elementos não repetidos das duas listas. • a primeira lista sem os elementos repetidos na segunda

## EXERCÍCIO 06-22

Escreva um programa que compare duas listas. Considere a primeira lista como a versão inicial e a segunda como a versão após alterações. Utilizando operações com conjuntos, seu programa deverá imprimir a lista de modificações entre essas duas versões, listando: • os elementos que não mudaram • os novos elementos • os elementos que foram removidos

## EXERCÍCIO 07-01

Escreva um programa que leia duas strings. Verifique se a segunda ocorre dentro da primeira e imprima a posição de início. 1ª string: AABBEFAATT 2ª string: BE Resultado: BE encontrado na posição 3 de AABBEFAATT

## EXERCÍCIO 07-02

Escreva um programa que leia duas strings e gere uma terceira com os caracteres comuns às duas strings lidas. 1ª string: AAACCTBF 2ª string: CBT Resultado: CBT A ordem dos caracteres da string gerada não é importante, mas deve conter todas as letras comuns a ambas.

## EXERCÍCIO 07-03

Escreva um programa que leia duas strings e gere uma terceira apenas com os caracteres que aparecem em uma delas. 1ª string: CTA 2ª string: ABC 3ª string: BT A ordem dos caracteres da terceira string não é importante.

## EXERCÍCIO 07-04

Escreva um programa que leia uma string e imprima quantas vezes cada caractere aparece nessa string. String: TTAAC Resultado: T: 2x A: 2x C: 1x

## EXERCÍCIO 07-05

Escreva um programa que leia duas strings e gere uma terceira, na qual os caracteres da segunda foram retirados da primeira. 1ª string: AATTGGAA 2ª string: TG 3ª string: AAAA

## EXERCÍCIO 07-06

Escreva um programa que leia três strings. Imprima o resultado da substituição na primeira, dos caracteres da segunda pelos da terceira. 1ª string: AATTCGAA 2ª string: TG 3ª string: AC Resultado: AAAACCAA

## EXERCÍCIO 07-07

Escreva um programa que peça ao usuário que digite uma frase e imprima quantas vogais ela contém. Não considere maiúsculas e minúsculas como diferentes. Exemplo: uma frase como “A casa” deve imprimir três “as”.

## EXERCÍCIO 07-08

Escreva um programa para exibir todas as palavras de uma frase. Considere que uma palavra termina com um espaço em branco ou quando a string terminar. Exemplo: “O rato roeu a roupa” deve imprimir 5.

## EXERCÍCIO 07-09

Modifique o o jogo da forca (Programa 7.2) de forma a escrever a palavra secreta caso o jogador perca.

## EXERCÍCIO 07-10

Modifique o Programa 7.2 de forma a utilizar uma lista de palavras. No início, pergunte um número e calcule o índice da palavra a utilizar pela fórmula:  $\text{índice} = (\text{número} * 776) \% \text{len}(\text{lista\_de\_palavras})$ .

## EXERCÍCIO 07-11

Modifique o Programa 7.2 para utilizar listas de strings para desenhar o boneco da forca. Você pode utilizar uma lista para cada linha e organizá-las em uma lista de listas. Em vez de controlar quando imprimir cada parte, desenhe nessas listas, substituindo o elemento a desenhar.

Exemplo:

```
>>> linha = list("X-----")
>>> linha
['X', '-', '-', '-', '-', '-', '-']
>>> linha[6] = "|"
>>> linha
['X', '-', '-', '-', '-', '-', '|']
>>> "".join(linha)
'X-----|'
```

## EXERCÍCIO 07-12

Escreva um jogo da velha para dois jogadores. O jogo deve perguntar onde você quer jogar e alternar entre os jogadores. A cada jogada, verifique se a posição está livre. Verifique também quando um jogador venceu a partida. Um jogo da velha pode ser visto como uma lista de 3 elementos, na qual cada elemento é outra lista, também com três elementos.

Exemplo do jogo:

```
X | O |  
---+---+---  
 | X | X  
---+---+---  
 |  | O
```

Em que cada posição pode ser vista como um número. Confira a seguir um exemplo das posições mapeadas para a mesma posição de seu teclado numérico.

```
7 | 8 | 9  
---+---+---  
4 | 5 | 6  
---+---+---  
1 | 2 | 3
```

## EXERCÍCIO 08-01

Escreva uma função que retorne o maior de dois números. Valores esperados:  $\text{máximo}(5, 6) == 6$   
 $\text{máximo}(2, 1) == 2$   $\text{máximo}(7, 7) == 7$

## EXERCÍCIO 08-02

Escreva uma função que receba dois números e retorne True se o primeiro número for múltiplo do segundo. Valores esperados:  $\text{múltiplo}(8, 4) == \text{True}$   $\text{múltiplo}(7, 3) == \text{False}$   $\text{múltiplo}(5, 5) == \text{True}$

## EXERCÍCIO 08-03

Escreva uma função que receba o lado de um quadrado e retorne sua área ( $A = \text{lado}^2$ ). Valores esperados:  $\text{área\_quadrado}(4) == 16$   $\text{área\_quadrado}(9) == 81$

## EXERCÍCIO 08-04

Escreva uma função que receba a base e a altura de um triângulo e retorne sua área ( $A = (\text{base} \times \text{altura}) / 2$ ). Valores esperados:  $\text{área\_triângulo}(6, 9) == 27$   $\text{área\_triângulo}(5, 8) == 20$

## EXERCÍCIO 08-05

Reescreva a função do Programa 8.1 de forma a utilizar os métodos de pesquisa em lista, vistos no Capítulo 7.

## EXERCÍCIO 08-06

Reescreva o Programa 8.2 de forma a utilizar for em vez de while.

## EXERCÍCIO 08-07

Defina uma função recursiva que calcule o maior divisor comum (M.D.C.) entre dois números a e b, em que  $a > b$ . Ver representação no livro

## EXERCÍCIO 08-08

Usando a função mdc definida no exercício anterior, defina uma função para calcular o menor múltiplo comum (M.M.C.) entre dois números.  $mmc(a, b) = |a \times b| / mdc(a, b)$  Em que  $|a \times b|$  pode ser escrito em Python como: `abs(a * b)`.

## EXERCÍCIO 08-09

Rastreie o Programa 8.6 e compare o seu resultado com o apresentado

## EXERCÍCIO 08-10

Reescreva a função para cálculo da sequência de Fibonacci, sem utilizar recursão.

## EXERCÍCIO 08-11

Escreva uma função para validar uma variável string. Essa função recebe como parâmetro a string, o número mínimo e máximo de caracteres. Retorne verdadeiro se o tamanho da string estiver entre os valores de máximo e mínimo, e falso, caso contrário.

## EXERCÍCIO 08-12

Escreva uma função que receba uma string e uma lista. A função deve comparar a string passada com os elementos da lista, também passada como parâmetro. Retorne verdadeiro se a string for encontrada dentro da lista, e falso, caso contrário.

## EXERCÍCIO 08-13-A

Escreva uma função que receba uma string com as opções válidas a aceitar (cada opção é uma letra). Converta as opções válidas para letras minúsculas. Utilize input para ler uma opção, converter o valor para letras minúsculas e verificar se a opção é válida. Em caso de opção inválida, a função deve pedir ao usuário que digite novamente outra opção.

## EXERCÍCIO 08-13-B

Escreva uma função que receba uma string com as opções válidas a aceitar (cada opção é uma letra). Converta as opções válidas para letras minúsculas. Utilize input para ler uma opção, converter o valor para letras minúsculas e verificar se a opção é válida. Em caso de opção inválida, a função deve pedir ao usuário que digite novamente outra opção.

## EXERCÍCIO 08-14

Altere o Programa 8.22 de forma que o usuário tenha três chances de acertar o número. O programa termina se o usuário acertar ou errar três vezes.

## EXERCÍCIO 08-15

Altere o Programa 7.2, o jogo da forca. Escolha a palavra a adivinhar utilizando números aleatórios.

## EXERCÍCIO 08-16

Modifique o jogo do alienígena. Crie uma variável que represente a vida do jogador, começando com 100. A partida termina quando você encontrar o alienígena ou quando a vida acabar (0). A cada erro, diminua a vida por um valor aleatório entre 5 e 20, representando um ataque do alienígena. Você pode retirar a parte do jogo que limita o número de tentativas e deixar apenas a vida do jogador ou do alienígena decidirem quando a partida termina. Exiba a vida do jogador antes de perguntar a próxima árvore.

## EXERCÍCIO 08-17

Melhore o programa do exercício anterior perguntando ao jogador o nível de dificuldade desejado. No modo fácil, a vida começa com 100 e o alienígena pode causar entre 5 e 20 de dano, como anteriormente. No modo médio, a vida começa com 80 e o alienígena pode causar danos entre 10 e 25. Já no modo difícil, a vida começa com 75 e o alienígena causa danos entre 20 e 30. Adicione mensagens e caracteres para deixar o jogo mais divertido.

## EXERCÍCIO 08-18

Modifique o Programa 8.26 para que receba dois parâmetros opcionais. Um para indicar o caractere a imprimir antes do número, sendo o espaço em branco o valor padrão, caso este não seja passado. O segundo parâmetro opcional é quantos caracteres adicionar por nível, tendo 2 como valor padrão.

## EXERCÍCIO 08-19

Escreva um generator capaz de gerar a sequência dos números primos.

## EXERCÍCIO 08-20

Escreva um generator capaz de gerar uma sequência com o fatorial de 1 até n, em que n é passado como parâmetro para o gerador.

## EXERCÍCIO 08-21

Escreva uma função que gere os números como a função range do Python. Essa função recebe três parâmetros e seu comportamento muda se passarmos um, dois ou três parâmetros. Chame-a de faixa.

Exemplos:

```
list(faixa(1))
[0, 1]
list(faixa(1, 10))
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
list(faixa(0, 10, 2))
[0, 2, 4, 6, 8, 10]
```

Você deve ter percebido que, diferente de range, a função faixa considera o fim do intervalo fechado, ou seja, o último número faz parte da faixa.

## EXERCÍCIO 08-22

Modifique o programa da calculadora que usa `partial` para suportar mais duas operações: raiz para raiz quadrada e potência para exponenciação.

## EXERCÍCIO 08-99

Escreva um generator capaz de gerar a série de Fibonacci.

## EXERCÍCIO 08-991

Utilizando a função `type`, escreva uma função recursiva que imprima os elementos de uma lista. Cada elemento deve ser impresso separadamente, um por linha. Considere o caso de listas dentro de listas, como `L = [1, [2, 3, 4, [5, 6, 7]]]`. A cada nível, imprima a lista mais à direita, como fazemos ao indentar blocos em Python. Dica: envie o nível atual como parâmetro e utilize-o para calcular a quantidade de espaços em branco à esquerda de cada elemento.

## EXERCÍCIO 09-01

Escreva um programa que receba o nome de um arquivo pela linha de comando e que imprima todas as linhas desse arquivo.

## EXERCÍCIO 09-02

Modifique o programa do Exercício 9.1 para que receba mais dois parâmetros: a linha de início e a de fim para impressão. O programa deve imprimir apenas as linhas entre esses dois valores (incluindo as linhas de início e fim).

## EXERCÍCIO 09-03

Crie um programa que leia os arquivos `pares.txt` e `ímpares.txt` e que crie um só arquivo `pareseimpares.txt` com todas as linhas dos outros dois arquivos, de forma a preservar a ordem numérica.

## EXERCÍCIO 09-04

Crie um programa que receba o nome de dois arquivos como parâmetros da linha de comando e que gere um arquivo de saída com as linhas do primeiro e do segundo arquivo. O nome do arquivo de saída também pode ser passado como parâmetro na linha de comandos.

## EXERCÍCIO 09-05

Crie um programa que inverta a ordem das linhas do arquivo pares.txt. A primeira linha deve conter o maior número; e a última, o menor.

## EXERCÍCIO 09-06

Modifique o Programa 9.5 para imprimir 40 vezes o símbolo de = se este for o primeiro caractere da linha. Adicione também a opção para parar de imprimir até que se pressione a tecla Enter cada vez que uma linha iniciar com . (ponto) como primeiro caractere.

## EXERCÍCIO 09-07

Crie um programa que leia um arquivo-texto e gere um arquivo de saída paginado. Cada linha não deve conter mais de 76 caracteres. Cada página terá no máximo 60 linhas. Adicione na última linha de cada página o número da página atual e o nome do arquivo original.

## EXERCÍCIO 09-08

Modifique o programa do Exercício 9.7 para também receber o número de caracteres por linha e o número de linhas por página pela linha de comando.

## EXERCÍCIO 09-09

Crie um programa que receba uma lista de nomes de arquivo e os imprima, um por um.

## EXERCÍCIO 09-10

Crie um programa que receba uma lista de nomes de arquivo e que gere apenas um grande arquivo de saída.

## EXERCÍCIO 09-11

Crie um programa que leia um arquivo e crie um dicionário em que cada chave é uma palavra e cada valor é o número de ocorrências no arquivo.

## EXERCÍCIO 09-12

Modifique o programa do Exercício 9.11 para também registrar a linha e a coluna de cada ocorrência da palavra no arquivo. Para isso, utilize listas nos valores de cada palavra, guardando a linha e a coluna de cada ocorrência.

## EXERCÍCIO 09-13

Crie um programa que imprima as linhas de um arquivo. Esse programa deve receber três parâmetros pela linha de comando: o nome do arquivo, a linha inicial e a última linha a imprimir

## EXERCÍCIO 09-14

Crie um programa que leia um arquivo-texto e elimine os espaços repetidos entre as palavras e no fim das linhas. O arquivo de saída também não deve ter mais de uma linha em branco repetida.

## EXERCÍCIO 09-15

Altere o Programa 7.2, o jogo da forca. Utilize um arquivo em que uma palavra seja gravada a cada linha. Use um editor de textos para gerar o arquivo. Ao iniciar o programa, utilize esse arquivo para carregar (ler) a lista de palavras. Experimente também perguntar o nome do jogador e gerar um arquivo com o número de acertos dos cinco melhores.

## EXERCÍCIO 09-16

Explique como os campos nome e telefone são armazenados no arquivo de saída.

## EXERCÍCIO 09-17

Altere o Programa 9.6 para exibir o tamanho da agenda no menu principal.

## EXERCÍCIO 09-18

O que acontece se nome ou telefone contiverem o caractere usado como separador em seus conteúdos? Explique o problema e proponha uma solução.

## EXERCÍCIO 09-19

Altere a função lista para que exiba também a posição de cada elemento.

## EXERCÍCIO 09-20

Adicione a opção de ordenar a lista por nome no menu principal.

## EXERCÍCIO 09-21

Nas funções de altera e apaga, peça que o usuário confirme a alteração e exclusão do nome antes de realizar a operação em si.

## EXERCÍCIO 09-22

Ao ler ou gravar uma nova lista, verifique se a agenda atual já foi gravada. Você pode usar uma variável para controlar quando a lista foi alterada (novo, altera, apaga) e reinicializar esse valor quando ela for lida ou gravada.

## EXERCÍCIO 09-23

Altere o programa para ler a última agenda lida ou gravada ao inicializar. Dica: utilize outro arquivo para armazenar o nome.

## EXERCÍCIO 09-24

O que acontece com a agenda se ocorrer um erro de leitura ou gravação? Explique.

## EXERCÍCIO 09-25

Altere as funções pede\\_nome e pede\\_telefone de forma a receberem um parâmetro opcional. Caso esse parâmetro seja passado, utilize-o como retorno caso a entrada de dados seja vazia.

## EXERCÍCIO 09-26

Altere o programa de forma a verificar a repetição de nomes. Gere uma mensagem de erro caso duas entradas na agenda tenham o mesmo nome

## EXERCÍCIO 09-27

Modifique o programa para também controlar a data de aniversário e o email de cada pessoa.

## EXERCÍCIO 09-28

Modifique o programa de forma a poder registrar vários telefones para a mesma pessoa. Permita também cadastrar o tipo de telefone: celular, fixo, residência ou trabalho.

## EXERCÍCIO 09-29

Modifique o Programa 9.8 para utilizar o elemento p em vez de h2 nos filmes.

## EXERCÍCIO 09-30

Modifique o Programa 9.8 para gerar uma lista HTML, usando os elementos ul e li. Todos os elementos da lista devem estar dentro do elemento ul, e cada item dentro de um elemento li. Exemplo:  
`<ul><li>Item1</li><li>Item2</li><li>Item3</li></ul>`

## EXERCÍCIO 09-31

Crie um programa que corrija o Programa 9.9 de forma a verificar se z existe e é um diretório.

## EXERCÍCIO 09-32

Modifique o Programa 9.9 de forma a receber o nome do arquivo ou diretório a verificar pela linha de comando. Imprima se existir e se for um arquivo ou um diretório.

## EXERCÍCIO 09-33

Crie um programa que gere uma página HTML com links para todos os arquivos jpg e png encontrados a partir de um diretório informado na linha de comando.

## EXERCÍCIO 09-34

Altere o Programa 7.2, o jogo da forca. Dessa vez, utilize as funções de tempo para cronometrar a duração das partidas.

## EXERCÍCIO 09-35

Utilizando a função `os.walk`, crie uma página HTML com o nome e tamanho de cada arquivo de um diretório passado e de seus subdiretórios.

## EXERCÍCIO 09-36

Utilizando a função `os.walk`, crie um programa que calcule o espaço ocupado por cada diretório e subdiretório, gerando uma página HTML com os resultados.

## EXERCÍCIO 09-37

Escreva um programa que leia o nome do aluno e quatro notas. No final, o programa deve gravar os dados lidos em um arquivo em disco, usando o formato JSON.

## EXERCÍCIO 09-38

Modifique o programa anterior para que leia o mesmo arquivo, permitindo adicionar mais dados ao arquivo. Se o mesmo nome for digitado duas vezes, altere os dados para a nova entrada.

## EXERCÍCIO 09-39

Modifique o Programa 9.6 da agenda. Faça-o ler e gravar um arquivo, mas no formato JSON.

## EXERCÍCIO 09-40

Modifique o programa `visualiza.py` para imprimir apenas os 512 primeiros bytes do arquivo.

## EXERCÍCIO 09-41

Altere o programa `visualiza.py` para receber o número máximo de bytes a imprimir e quantos bytes por linha pela linha de comando.

## EXERCÍCIO 09-42

Modifique o Programa 9.20 para que receba o nome da imagem a gerar pela linha de comando.

## EXERCÍCIO 09-43

Modifique o programa do exercício anterior para receber um segundo parâmetro com o nome do arquivo com o desenho. A ideia é ler o desenho desse arquivo.

## EXERCÍCIO 09-44

Modifique o programa anterior para receber um terceiro parâmetro com a tabela de conversão de cores no formato JSON.

## EXERCÍCIO 10-01

Adicione os atributos tamanho e marca à classe Televisão. Crie dois objetos Televisão e atribua tamanhos e marcas diferentes. Depois, imprima o valor desses atributos de forma a confirmar a independência dos valores de cada instância (objeto).

## EXERCÍCIO 10-02

Atualmente, a classe Televisão inicializa o canal com 2. Modifique a classe Televisão de forma a receber o canal inicial em seu construtor como parâmetro opcional.

## EXERCÍCIO 10-03

Modifique a classe Televisão de forma que, se pedirmos para mudar o canal para baixo, além do mínimo, ela vá para o canal máximo. Se mudarmos para cima, além do canal máximo, que volte ao canal mínimo. Exemplo:

```
>>> tv = Televisão(2, 10)
>>> tv.muda_canal_para_baixo()
>>> tv.canal
10
>>> tv.muda_canal_para_cima()
>>> tv.canal
2
```

## EXERCÍCIO 10-04

Utilizando o que aprendemos com funções, modifique o construtor da classe Televisão de forma que canal\\_min e canal\\_max sejam parâmetros opcionais, em que canal\\_min vale 2 e canal\\_max vale 14, caso outro valor não seja passado.

## EXERCÍCIO 10-05

Utilizando a classe Televisão modificada no exercício anterior, crie duas instâncias (objetos), especificando o valor de canal\\_min e canal\\_max por nome.

## EXERCÍCIO 10-06

Modifique a classe Televisão de forma que os métodos muda\\_canal\\_para\\_cima e muda\\_canal\\_para\\_baixo retornem o canal após a mudança.

## EXERCÍCIO 10-07

Altere a classe Televisão. Ela só deve aceitar os comandos de trocar de canal se estiver ligada.

## EXERCÍCIO 10-08

Altere o programa de forma que a mensagem saldo insuficiente seja exibida caso haja tentativa de sacar mais dinheiro que o saldo disponível.

## EXERCÍCIO 10-09

Modifique o método resumo da classe Conta para exibir o nome e o telefone de cada cliente.

## EXERCÍCIO 10-10

Crie uma nova conta, agora tendo João e José como clientes e saldo igual a 500.

## EXERCÍCIO 10-11

Crie classes para representar estados e cidades. Cada estado tem um nome, sigla e cidades. Cada cidade tem nome e população. Escreva um programa de testes que crie três estados com algumas cidades em cada um. Exiba a população de cada estado como a soma da população de suas cidades.

## EXERCÍCIO 10-12

Modifique as classes Conta e ContaEspecial para que a operação de saque retorne verdadeiro se o saque foi efetuado e falso, caso contrário.

## EXERCÍCIO 10-13

Altere a classe `ContaEspecial` de forma que seu extrato exiba o limite e o total disponível para saque.

## EXERCÍCIO 10-14

Observe o método `saque` das classes `Conta` e `ContaEspecial`. Modifique o método `saque` da classe `Conta` de forma que a verificação da possibilidade de saque seja feita por um novo método, substituindo a condição atual. Esse novo método retornará verdadeiro se o saque puder ser efetuado, e falso, caso contrário. Modifique a classe `ContaEspecial` de forma a trabalhar com esse novo método. Verifique se você ainda precisa trocar o método `saque` de `ContaEspecial` ou apenas o novo método criado para verificar a possibilidade de saque.

## EXERCÍCIO 10-15

Modifique a classe `ListaÚnica` para sobrescrever o método `extend` de `UserList`. `extend` funciona como o `append`, mas recebe uma lista como parâmetro. Verifique o tipo de cada elemento na lista antes de adicioná-lo à lista.

## EXERCÍCIO 11-01

Faça um programa que crie o banco de dados `preços.db` com a tabela `preços` para armazenar uma lista de preços de venda de produtos. A tabela deve conter o nome do produto e seu respectivo preço. O programa também deve inserir alguns dados para teste.

## EXERCÍCIO 11-02

Faça um programa para listar todos os preços do banco `preços.db`.

## EXERCÍCIO 11-03

Escreva um programa que realize consultas do banco de dados `preços.db`, criado no Exercício 11.1. O programa deve perguntar o nome do produto e listar seu preço.

## EXERCÍCIO 11-04

Modifique o programa do Exercício 11.3 de forma a perguntar dois valores e listar todos os produtos com preços entre esses dois valores.

## EXERCÍCIO 11-05

Escreva um programa que aumente o preço de todos os produtos do banco preços.db em 10%.

## EXERCÍCIO 11-06

Escreva um programa que pergunte o nome do produto e um novo preço. Usando o banco preços.db, atualize o preço desse produto no banco de dados.

## EXERCÍCIO 12-01

Modifique o programa anterior para reconhecer sequências de letras. Uma letra é um caractere entre A e Z ou entre a e z, considerando letras maiúsculas e minúsculas. Ignore caracteres acentuados. Imprima uma lista com as sequências de letras encontradas.

## EXERCÍCIO 12-02

Reescreva a função que mostra os números na entrada ABC431DEF-901c431203FXEW9, mas usando a função `verifica\_padrão`.

## EXERCÍCIO 12-03

Utilizando a função `verifica\_padrão`, escreva uma função que detecte uma data no formato dd/mm/aa em que dd é o dia, mm o mês e aa o ano. A função deve apenas detectar o padrão da data e não precisa verificar se a data é válida.

## EXERCÍCIO 12-04

Utilizando a função `verifica\_padrão`, escreva uma função que detecte um valor em reais no formato: R\$999,99 em que 9 representa qualquer dígito. O primeiro número pode ter um ou mais dígitos, mas a segunda parte (centavos) deve ter no máximo dois dígitos.

## EXERCÍCIO 12-05

Crie uma função sequências que recebe `qmax` e `qmin`. Ela deve funcionar de forma semelhante a número, mas chamando a função `sequência`. Ela deve também funcionar quando `qmin` é 0, ou seja, quando a sequência é opcional.

## EXERCÍCIO 12-06

Crie uma função usando `verifica_padrao` que valide números de celulares. Um celular tem 9 dígitos depois do DDD. Por exemplo: (92)99812-1103.

## EXERCÍCIO 12-07

Escreva um programa que valide a entrada de dados do usuário. O programa deve aceitar números de CPF no seguinte formato: 999.999.999-99, em que cada 9 representa um dígito. Exija os pontos e o traço no final, verificando a correta quantidade de dígitos.

## EXERCÍCIO 12-08

Escreva um programa que valide a entrada de dados do usuário. O programa deve aceitar números de CNPJ no seguinte formato: 99.999.999/9999-99, em que cada 9 representa um dígito. Exija os pontos e o traço no final, verificando a correta quantidade de dígitos.

## EXERCÍCIO 12-09

Escreva um programa que valide a entrada de dados do usuário. Tente achar um número válido de CPF ou de CNPJ como definido nos exercícios anteriores. Exiba uma mensagem dizendo se o número é válido e se este é um CNPJ ou um CPF.

## EXERCÍCIO 12-10

Escreva uma função que aceite preços em reais. O programa deve ignorar espaços em branco e aceitar valores prefixados com R\$ ou não (com r ou R). O usuário deve entrar valores corretamente formatados com o ponto separando os milhares e a vírgula, os centavos. Se o usuário digitar centavos, estes devem ter dois dígitos. Valores válidos:

```
R$500
R$500
R$500,10
R$7.312,10
```

## EXERCÍCIO 13-01

Modifique o programa 13.7 para gravar e carregar o desenho em formato JSON. Você pode percorrer os objetos do canvas e guardar a forma, as coordenadas e as cores escolhidas.

## EXERCÍCIO 13-02

Modifique o programa anterior e grave a imagem no formato SVG. O formato SVG é um arquivo texto, que segue um formato bem definido.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<svg width="190mm" height="63mm" viewBox="0 0 190 63" version="1.1" id="svg1"
xmlns="http://www.w3.org/2000/svg" xmlns:svg="http://www.w3.org/2000/svg">
  <g id="layer1">
    <rect style="fill:#cccccc;stroke:#000000;stroke-width:0.264583" id="rect1"
width="42.451897" height="29.063223" x="14.858164" y="16.001099" />
    <ellipse style="fill:#cccccc;stroke:#000000;stroke-width:0.264583"
id="path1" cx="93.067612" cy="18.940079" rx="15.837823" ry="12.082462" />
    <path style="fill:none;stroke:#000000;stroke-width:0.264583px;stroke-l
inecap:butt;stroke-linejoin:miter;stroke-opacity:1" d="m 172.90984,16.654206
-54.69763,37.5536" id="path2" />
  </g>
</svg>
```

Que desenha a imagem mostrada na Figura 13.14.

## EXERCÍCIO 13-03

Modifique o código de Janela.ok (programa 13.12) para validar as datas ao editar ou adicionar novos links. Exiba uma mensagem de erro caso a data seja inválida.

## EXERCÍCIO 13-04

Modifique o código de Janela.ok (programa 13.12) para não aceitar URLs em branco. Exiba uma mensagem de erro caso data seja inválida.

## EXERCÍCIO 13-05

Modifique o código de Janela.ok (programa 13.12) para aceitar somente URLs iniciando por http:// ou https://.

## EXERCÍCIO 13-06

Modifique o programa 13.13 para ler e gravar dados de um banco de dados. Modificando a classe GerenteDeSites ou criando outra, você deve facilmente poder trocar o armazenamento de JSON para um banco de dados como o Sqlite.

