

Exercícios Enunciados

Nilo Ney Coutinho Menezes <livro_de_python@nilo.pro.br>

2022-03-07: Gerado automaticamente

Conteúdo

Sobre o livro	1
Introdução	3
Exercício 02-01.	4
Exercício 02-02.	4
Exercício 02-03.	4
Exercício 02-04.	4
Exercício 02-05.	4
Exercício 02-06.	4
Exercício 03-01.	4
Exercício 03-02.	5
Exercício 03-03.	5
Exercício 03-04.	5
Exercício 03-05.	6
Exercício 03-06.	6
Exercício 03-07.	6
Exercício 03-08.	6
Exercício 03-09.	6
Exercício 03-10.	6
Exercício 03-11.	7
Exercício 03-12.	7
Exercício 03-13.	7
Exercício 03-14.	7
Exercício 03-15.	7
Exercício 04-01.	7
Exercício 04-02.	8
Exercício 04-03.	8
Exercício 04-04.	8
Exercício 04-05.	8
Exercício 04-06.	8
Exercício 04-07.	8
Exercício 04-08.	8
Exercício 04-09.	9
Exercício 04-10.	9
Exercício 05-01.	9
Exercício 05-02.	9

Exercício 05-03.....	9
Exercício 05-04.....	10
Exercício 05-05.....	10
Exercício 05-06.....	10
Exercício 05-07.....	10
Exercício 05-08.....	10
Exercício 05-09.....	10
Exercício 05-10.....	11
Exercício 05-11.....	11
Exercício 05-12.....	11
Exercício 05-13.....	11
Exercício 05-14.....	11
Exercício 05-15.....	11
Exercício 05-16.....	12
Exercício 05-17.....	12
Exercício 05-18.....	12
Exercício 05-19.....	12
Exercício 05-20.....	12
Exercício 05-21.....	12
Exercício 05-22.....	13
Exercício 05-23.....	13
Exercício 05-24.....	13
Exercício 05-25.....	13
Exercício 05-26.....	13
Exercício 05-27-a.....	13
Exercício 05-27-b.....	14
Exercício 06-01.....	14
Exercício 06-02.....	14
Exercício 06-03.....	14
Exercício 06-04.....	14
Exercício 06-05.....	14
Exercício 06-06.....	14
Exercício 06-07.....	15
Exercício 06-08.....	15
Exercício 06-09.....	15
Exercício 06-10.....	15
Exercício 06-11.....	15
Exercício 06-12.....	16

Exercício 06-13.....	16
Exercício 06-14.....	16
Exercício 06-15.....	16
Exercício 06-16.....	16
Exercício 06-17.....	16
Exercício 06-18-a.....	16
Exercício 06-18-b.....	17
Exercício 06-19.....	17
Exercício 06-20.....	17
Exercício 07-01.....	17
Exercício 07-02.....	17
Exercício 07-03.....	17
Exercício 07-04.....	18
Exercício 07-05.....	18
Exercício 07-06.....	18
Exercício 07-07.....	18
Exercício 07-08.....	18
Exercício 07-09.....	18
Exercício 07-10.....	19
Exercício 08-01.....	19
Exercício 08-02.....	20
Exercício 08-03.....	20
Exercício 08-04.....	20
Exercício 08-05.....	20
Exercício 08-06.....	20
Exercício 08-07.....	20
Exercício 08-08.....	20
Exercício 08-09.....	21
Exercício 08-10.....	21
Exercício 08-11.....	21
Exercício 08-12.....	21
Exercício 08-13-a.....	21
Exercício 08-13-b.....	21
Exercício 08-13-c.....	21
Exercício 08-14.....	22
Exercício 08-15.....	22
Exercício 08-16.....	22
Exercício 08-17.....	22

Exercício 09-01.....	22
Exercício 09-02.....	22
Exercício 09-03.....	22
Exercício 09-04.....	23
Exercício 09-05.....	23
Exercício 09-06.....	23
Exercício 09-07.....	23
Exercício 09-08.....	23
Exercício 09-09.....	23
Exercício 09-10.....	23
Exercício 09-11.....	24
Exercício 09-12.....	24
Exercício 09-13.....	24
Exercício 09-14.....	24
Exercício 09-15.....	24
Exercício 09-16.....	24
Exercício 09-17.....	25
Exercício 09-18.....	25
Exercício 09-19.....	25
Exercício 09-20.....	25
Exercício 09-21.....	25
Exercício 09-22.....	25
Exercício 09-23.....	25
Exercício 09-24.....	25
Exercício 09-25.....	26
Exercício 09-26.....	26
Exercício 09-27.....	26
Exercício 09-28.....	26
Exercício 09-29.....	26
Exercício 09-30.....	26
Exercício 09-31.....	26
Exercício 09-32.....	26
Exercício 09-33.....	27
Exercício 09-34.....	27
Exercício 09-35.....	27
Exercício 09-36.....	27
Exercício 10-01.....	27
Exercício 10-02.....	27

Exercício 10-03.....	27
Exercício 10-04.....	28
Exercício 10-05.....	28
Exercício 10-06.....	28
Exercício 10-07.....	28
Exercício 10-08.....	28
Exercício 10-09.....	29
Exercício 10-10.....	29
Exercício 10-11.....	29
Exercício 10-12.....	29
Exercício 11-01.....	29
Exercício 11-02.....	29
Exercício 11-03.....	30
Exercício 11-04.....	30
Exercício 11-05.....	30
Exercício 11-06.....	30

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 e banco de dados com SQLite 3 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 +3.7) 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: Terceira

ISBN: 978-85-7522-718-3

Editora: Novatec

Ano: 2019

Páginas: 328

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



Figura 1. Na Amazon: <https://amzn.to/2swVLAH>

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



Figura 2. Na Editora Novatec: <http://www.novatec.com.br/livros/introducao-python-3ed/>

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:



Figura 3. 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.

ATENÇÃO | 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 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	$b > a$	<input type="checkbox"/> True
$a < b$	<input type="checkbox"/> True	$c >= f$	<input type="checkbox"/> True
$d > b$	<input type="checkbox"/> True	$f >= c$	<input type="checkbox"/> True
$c != f$	<input type="checkbox"/> True	$c <= c$	<input type="checkbox"/> True
$a == b$	<input type="checkbox"/> True	$c <= f$	<input type="checkbox"/> True
$c < d$	<input type="checkbox"/> True		

Exercício 03-03

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

Expressão	Resultado	Expressão	Resultado
$a \text{ and } a$	<input type="checkbox"/> True	$a \text{ or } c$	<input type="checkbox"/> True
$b \text{ and } b$	<input type="checkbox"/> True	$b \text{ or } c$	<input type="checkbox"/> True
$\text{not } c$	<input type="checkbox"/> True	$c \text{ or } a$	<input type="checkbox"/> True
$\text{not } b$	<input type="checkbox"/> True	$c \text{ or } b$	<input type="checkbox"/> True
$\text{not } a$	<input type="checkbox"/> True	$c \text{ or } c$	<input type="checkbox"/> True
$a \text{ and } b$	<input type="checkbox"/> True	$b \text{ or } b$	<input type="checkbox"/> True
$b \text{ and } c$	<input type="checkbox"/> True		

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 $A > B$ and C 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 4.4 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

Rastreie o Programa 4.6. Compare seu resultado ao apresentado na Tabela 4.2.

Exercício 04-08

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 04-09

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 04-10

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 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

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 06-15

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

Exercício 06-16

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 06-17

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 06-18-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 $\rightarrow \{ \text{"O"}:1, \text{"r"}:1, \text{"a"}:1, \text{"t"}:1, \text{"o"}:1 \}$

Exercício 06-18-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 \rightarrow { "O":1, "r":1, "a":1, "t":1, "o":1 }

Exercício 06-19

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 06-20

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

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

Exercício 07-08

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 07-09

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 07-10

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: `máximo(5, 6) == 6`
`máximo(2, 1) == 2` `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: `múltiplo(8, 4) == True` `múltiplo(7, 3) == False` `múltiplo(5, 5) == True`

Exercício 08-03

Escreva uma função que receba o lado de um quadrado e retorne sua área ($A = \text{lado}^2$). Valores esperados: `área_quadrado(4) == 16` `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: `área_triângulo(6, 9) == 27` `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 $a > b$.

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. $\text{mmc}(a, b) = |a \times b| / \text{mdc}(a, b)$ 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

Altere o Programa 8.20 de forma que o usuário tenha três chances de acertar o número. O programa termina se o usuário acertar ou errar três vezes.

Exercício 08-13-b

Altere o Programa 8.20 de forma que o usuário tenha três chances de acertar o número. O programa termina se o usuário acertar ou errar três vezes.

Exercício 08-13-c

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

Exercício 08-15

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 08-16

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

Exercício 08-17

Escreva um generator capaz de gerar a série de Fibonacci.

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 `paresempares.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.

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: `Item1Item2Item3`

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 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.

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 min e max sejam parâmetros opcionais, em que min vale 2 e 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 min e max por nome.

Exercício 10-06

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

Exercício 10-07

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

Exercício 10-08

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

Exercício 10-09

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 10-10

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

Exercício 10-11

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

Exercício 10-12

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 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.