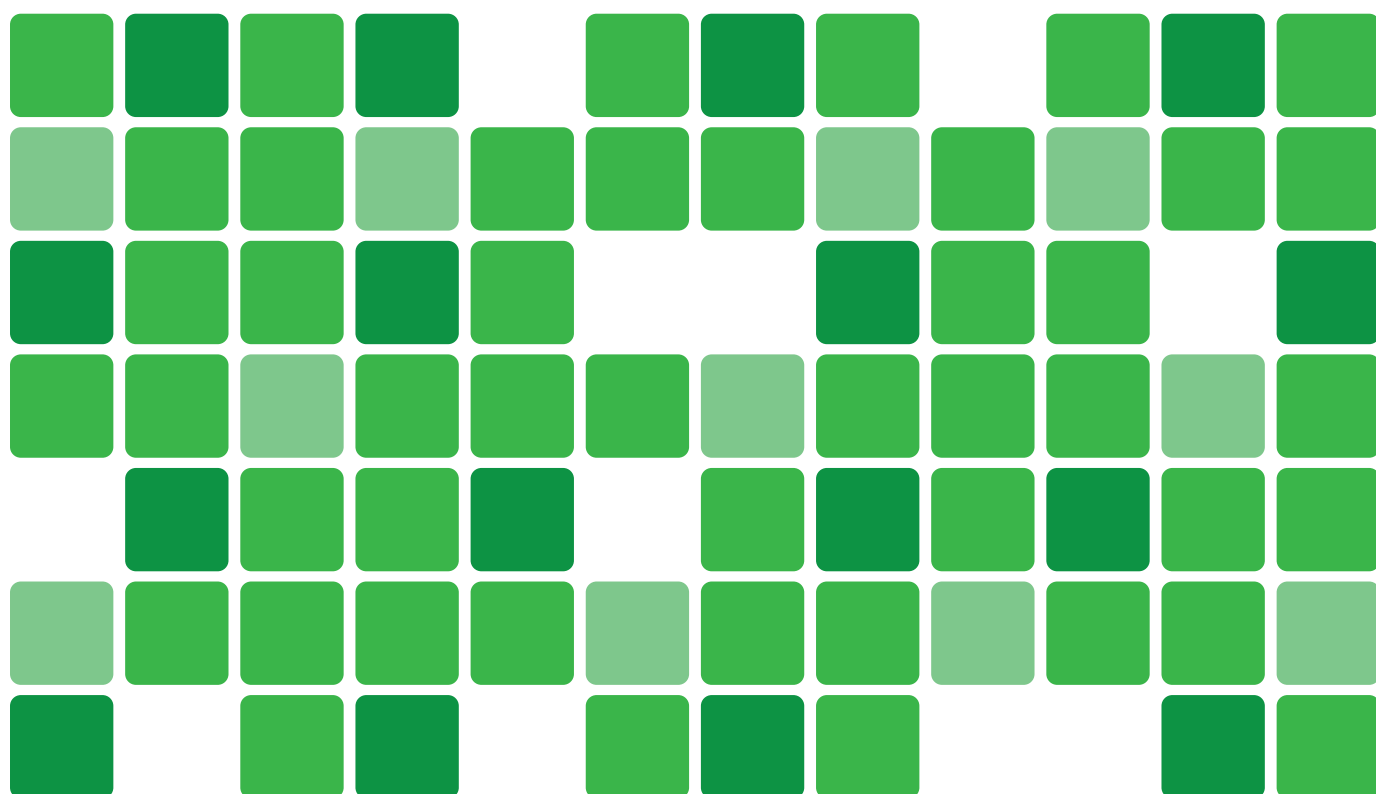




Programação para a Web

Técnico em Informática Integrado ao Ensino Médio

Prof. Dr. Ricardo Maroquio



Copyright © 2023 Ricardo Maroquio

INSTITUTO FEDERAL DO ESPÍRITO SANTO
CAMPUS CACHOEIRO

Disponível para download em:
[HTTP://MAROQUIO.COM](http://maroquio.com)

Esta apostila está sob a licença *Creative Commons Attribution-NonCommercial 4.0*. Você só pode usar esse material se concordar com essa licença. Você pode acessar a licença em <https://creativecommons.org/licenses/by-nc-sa/4.0>. A menos que seja aplicável por lei ou acordado por escrito, material digital sob esta licença deve ser distribuído “COMO ESTÁ”, SEM GARANTIAS OU CONDIÇÕES DE NENHUM TIPO. Para maiores informações sobre o uso deste material, consulte a licença disponível no link supracitado.

Fevereiro de 2023



Sumário

I

Parte I: Python

1	Fundamentos da Linguagem	9
1.1	Variáveis	9
1.2	Tipos de Dados	11
1.2.1	Números	11
1.2.2	Textos	11
1.2.3	Listas	11
1.2.4	Dicionários	12
1.2.5	Conjuntos	13
1.3	Operadores Matemáticos	13
1.3.1	Adição	14
1.3.2	Subtração	14
1.3.3	Multiplicação	14
1.3.4	Divisão	15
1.3.5	Divisão Inteira	15
1.3.6	Resto da Divisão Inteira	15
1.3.7	Exponenciação	15
1.4	Entrada e Saída de Dados	15
1.4.1	Entrada de Dados	16
1.4.2	Saída de Dados	16
1.4.3	Saída Formatada	16

1.5	Exercícios Propostos	17
1.6	Considerações Sobre o Capítulo	20
2	Estruturas Condicionais	21
3	Estruturas de Repetição	23
4	Números, Textos e Coleções	25
5	Funções	27
6	Tratamento de Exceções	29
7	Arquivos e Módulos	31
8	Classes e Objetos	33
9	Manipulação de Bancos de Dados SQL	35
10	Manipulação de Bancos de Dados NoSQL	37
	Bibliografia	39
	Artigos	39
	Livros	39



Lista de Exemplos de Código



1.1	Atribuição de valores a variáveis	10
1.2	Reatribuição de valores a variáveis	10
1.3	Tipos numéricos	11
1.4	Tipo Texto	11
1.5	Tipo Lista	12
1.6	Tipo Dicionário	12
1.7	Tipo Conjunto	13
1.8	Operações Sobre Conjuntos	13
1.9	Usos do operador de adição	14
1.10	Operador de subtração	14
1.11	Operador de multiplicação	14
1.12	Operador de divisão	15
1.13	Operador de divisão	15
1.14	Operador de resto da divisão inteira	15
1.15	Operador de multiplicação	16
1.16	Entrada de dados em console	16
1.17	Saída de dados em console	16
1.18	O método format()	17
1.19	O operador de formatação %	17
1.20	O operador de formatação "f"	17

Parte I: Python

1	Fundamentos da Linguagem	9
1.1	Variáveis	9
1.2	Tipos de Dados	11
1.3	Operadores Matemáticos	13
1.4	Entrada e Saída de Dados	15
1.5	Exercícios Propostos	17
1.6	Considerações Sobre o Capítulo	20
2	Estruturas Condicionais	21
3	Estruturas de Repetição	23
4	Números, Textos e Coleções	25
5	Funcões	27
6	Tratamento de Exceções	29
7	Arquivos e Módulos	31
8	Classes e Objetos	33
9	Manipulação de Bancos de Dados SQL	35
10	Manipulação de Bancos de Dados NoSQL	37
	Bibliografia	39
	Artigos	39
	Livros	39



1. Fundamentos da Linguagem

O Python é uma linguagem de programação poderosa e versátil amplamente utilizada em aplicações de computação científica, análise de dados, desenvolvimento de aplicações e muito mais. Neste primeiro capítulo, revisaremos os conceitos básicos da programação em Python e veremos como aplicá-los para resolver problemas comuns do mundo da programação.

Vale ressaltar que este capítulo tem como proposta a realização de uma rápida revisão da linguagem Python, ou seja, pressupõe-se que você já tenha tido alguma experiência com programação em Python. Ao longo do capítulo, você aprenderá sobre tipos de dados, variáveis, estruturas de controle de fluxo, funções e muito mais. Também veremos muitos exemplos práticos que irão ajudá-lo a aplicar os conceitos abordados.

Ao concluir este capítulo, você terá revisto todos os recursos necessários para escrever programas simples em Python e estará preparado para se aprofundar em recursos adicionais de programação em Python com mais confiança.

1.1 Variáveis

Variáveis são consideradas componentes fundamentais da programação. Elas permitem armazenar valores que podem ser usados posteriormente ao longo do programa. Em Python, as variáveis são criadas usando o sinal de igual (=). O exemplo 1.1 mostra a criação de duas variáveis simples.

Neste exemplo, criamos duas variáveis: `nome` e `idade`. O nome da variável fica à esquerda do sinal de igual e o valor da variável fica à direita. O tipo da variável é determinado automaticamente com base no valor atribuído. Uma variável pode ter

```
1 | nome = "João"
2 | idade = 30
```

Exemplo de Código 1.1: Atribuição de valores a variáveis

seu valor modificado a qualquer momento. O exemplo 1.2

```
1 | nome = 'João'
2 | idade = 30
3 | print("Meu nome é", nome, "e tenho", idade, "anos.")
4 |
5 | nome = "Maria"
6 | idade = 35
7 | print('Meu nome é', nome, "e tenho", idade, "anos.")
```

Exemplo de Código 1.2: Reatribuição de valores a variáveis

Neste exemplo, alteramos o valor da variável `nome` de "João" para "Maria" e o valor da variável `idade` de 30 para 35. Quando o programa é executado, o texto "Meu nome é Maria e tenho 35 anos." é impresso na tela.

Lembre-se de que as variáveis em Python não precisam ser declaradas com antecedência. Basta atribuir um valor a uma variável para criá-la automaticamente. Além disso, é importante escolher nomes de variáveis descritivos e significativos para facilitar a leitura e manutenção do código. As regras para nomeação de variáveis em Python são as seguintes:

1. Nomes de variáveis devem começar com uma letra ou um sublinhado (`_`);
2. Após o primeiro caractere, podem conter letras, números e sublinhados;
3. Nomes de variáveis não podem ser uma palavra reservada do Python, como `if`, `else`, `for` etc.;
4. Nomes de variáveis devem ser descritivos e significativos;
5. Nomes de variáveis não podem conter espaços. Em vez disso, usa-se o sublinhado para separar palavras;
6. Nomes de variáveis devem ser escritos em minúsculas, exceto em casos de convenções de nomenclatura como `camelCase`, `PascalCase` ou `snake_case`.

Agora que você já sabe como declarar variáveis em Python, vamos quais são os possíveis tipos de valores que podemos atribuir a uma variável.

1.2 Tipos de Dados

Em programação, é importante compreender os diferentes tipos de dados que você pode trabalhar. Em Python, existem vários tipos de dados básicos que você precisa conhecer, incluindo números, strings, listas, dicionários, conjuntos, entre outros. Vamos explorar cada um desses tipos em detalhes.

1.2.1 Números

Existem dois tipos numéricos em Python: inteiros (*int*) e de ponto flutuante (*float*). No exemplo 1.3, veja que a variável *x* recebe um valor inteiro e que a variável *y* recebe um valor real.

```
1 | x = 10 # int
2 | y = 3.14 # float
```

Exemplo de Código 1.3: Tipos numéricos

1.2.2 Textos

Textos são uma sequência de caracteres. Em Python, são representados pelo tipo *string* seus valores podem ser delimitados por aspas simples ou duplas. O exemplo 1.4 mostra como criar strings usando aspas simples e aspas duplas.

```
1 | frase1 = "Olá, Python!" # string delimitado por aspas duplas
2 | frase2 = 'Olá, Python!' # string delimitado por aspas simples
```

Exemplo de Código 1.4: Tipo Texto

1.2.3 Listas

As listas são uma estrutura de dados em Python que permitem armazenar uma coleção de itens tipos iguais ou diferentes. As listas são delimitadas por colchetes `[]` e os itens são separados por vírgulas. O exemplo 1.5 mostra a criação de 3 listas diferentes.

Você pode acessar, alterar ou remover itens em uma lista usando seus índices (posições) e também pode realizar operações comuns em listas, como concatenação, repetição e pesquisa. Além disso, as listas são objetos mutáveis, o que significa que você pode modificar seus itens depois de criá-los. Veremos listas novamente mais adiante de forma mais aprofundada.

```
1  # criando uma lista de números inteiros
2  numeros = [1, 2, 3, 4, 5]
3
4  # criando uma lista de strings
5  frutas = ['maçã', 'banana', 'laranja']
6
7  # criando uma lista mista de diferentes tipos de dados
8  mista = [1, 'dois', 3.0, ['a', 'b', 'c']]
```

Exemplo de Código 1.5: Tipo Lista

1.2.4 Dicionários

Em Python, um dicionário é uma estrutura de dados que permite armazenar uma coleção de pares chave-valor. Ao contrário de uma lista, que armazena itens em uma ordem específica baseada em índices, um dicionário armazena itens com base nas suas chaves. As chaves são usadas para acessar seus valores correspondentes.

Os dicionários são delimitados por chaves e cada par chave-valor é separado por vírgulas. As chaves podem ser de diferentes tipos, como números, strings ou tuplas, mas precisam ser imutáveis. Os valores podem ser de qualquer tipo. O exemplo 1.6 mostra a criação e algumas operações básicas sobre um dicionário.

```
1  # criando um dicionário com pares chave-valor
2  pessoa = {'nome': 'João', 'idade': 30, 'cidade': 'São Paulo'}
3
4  # acessando um valor pelo seu nome de chave
5  print(pessoa['nome']) # imprime "João"
6
7  # alterando o valor de uma chave
8  pessoa['idade'] = 31
9  print(pessoa['idade']) # imprime 31
10
11 # adicionando um novo par chave-valor
12 pessoa['pais'] = 'Brasil'
13 print(pessoa) # imprime {'nome': 'João', 'idade': 31, 'cidade': 'São Paulo', 'pais': 'Brasil'}
14
15 # removendo um par chave-valor
16 del pessoa['cidade']
17 print(pessoa) # imprime {'nome': 'João', 'idade': 31, 'pais': 'Brasil'}
```

Exemplo de Código 1.6: Tipo Dicionário

1.2.5 Conjuntos

Em Python, conjuntos são coleções não ordenadas e não indexadas de elementos únicos. Eles são definidos usando chaves ou a função `set()`. O exemplo 1.7 mostra as duas formas de criação de conjuntos.

```
1 conjunto = {1, 2, 3, 4}
2 print(conjunto) # output: {1, 2, 3, 4}
3
4 conjunto = set([1, 2, 3, 4])
5 print(conjunto) # output: {1, 2, 3, 4}
```

Exemplo de Código 1.7: Tipo Conjunto

Os elementos em um conjunto não podem ser repetidos e não possuem uma ordem específica. Isso significa que você não pode acessar um elemento específico de um conjunto usando um índice, mas você pode verificar se um elemento está presente no conjunto usando o operador `in`. Você pode realizar operações comuns com conjuntos, como união, interseção e diferença. O exemplo 1.8 mostra a realização de algumas dessas operações.

```
1 conjunto1 = {1, 2, 3, 4}
2 conjunto2 = {3, 4, 5, 6}
3
4 # União
5 print(conjunto1 | conjunto2) # output: {1, 2, 3, 4, 5, 6}
6
7 # Interseção
8 print(conjunto1 & conjunto2) # output: {3, 4}
9
10 # Diferença
11 print(conjunto1 - conjunto2) # output: {1, 2}
```

Exemplo de Código 1.8: Operações Sobre Conjuntos

1.3 Operadores Matemáticos

Em Python, existem vários operadores matemáticos que permitem realizar operações matemáticas básicas com números, como adição, subtração, multiplicação, divisão, exponenciação, etc. As subseções a seguir apresenta os operadores matemáticos nativos mais comuns da linguagem Python, juntamente com exemplos uso.

1.3.1 Adição

O operador de adição, representado pelo caractere "+", adiciona dois números ou concatena duas strings. Veja o exemplo 1.9 a seguir.

```
1 | a = 5
2 | b = 3
3 | c = a + b
4 | print(c) # imprime 8
5 |
6 | s1 = "Olá"
7 | s2 = " mundo!"
8 | s3 = s1 + s2
9 | print(s3) # imprime "Olá mundo!"
```

Exemplo de Código 1.9: Usos do operador de adição

1.3.2 Subtração

O operador de subtração, representado pelo caractere "-", subtrai dois números. Veja o exemplo 1.10 a seguir.

```
1 | a = 5
2 | b = 3
3 | c = a - b
4 | print(c) # imprime 2
```

Exemplo de Código 1.10: Operador de subtração

1.3.3 Multiplicação

O operador de multiplicação, representado pelo caractere "*", multiplica dois números ou repetidamente concatena uma string. Veja o exemplo 1.11 a seguir.

```
1 | a = 5
2 | b = 3
3 | c = a * b
4 | print(c) # imprime 15
5 |
6 | s = "Oi"
7 | r = s * 3
8 | print(r) # imprime "OiOiOi"
```

Exemplo de Código 1.11: Operador de multiplicação

1.3.4 Divisão

O operador de divisão, representado pelo caractere "/", divide um número por outro. Veja o exemplo 1.13 a seguir.

```
1 | a = 6
2 | b = 3
3 | c = a / b
4 | print(c) # imprime 2.0
```

Exemplo de Código 1.12: Operador de divisão

1.3.5 Divisão Inteira

O operador de divisão inteira, representado pelos caracteres "//", divide um número por outro e retorna somente a parte inteira do resultado. Veja o exemplo 1.13 a seguir.

```
1 | a = 7
2 | b = 3
3 | c = a // b
4 | print(c) # imprime 2
```

Exemplo de Código 1.13: Operador de divisão

1.3.6 Resto da Divisão Inteira

O operador de resto da divisão inteira, representado pelo caractere "%", retorna o resto da divisão inteira de um número por outro. Veja o exemplo 1.14 a seguir.

```
1 | a = 7
2 | b = 3
3 | c = a % b
4 | print(c) # imprime 1
```

Exemplo de Código 1.14: Operador de resto da divisão inteira

1.3.7 Exponenciação

O operador de exponenciação, representado pelos caracteres "**", eleva um número a uma potência. Veja o exemplo 1.15 a seguir.

1.4 Entrada e Saída de Dados

A linguagem Python possui comandos nativos para entrada e saída de dados em programas console. Através desses comandos, você consegue permitir que o usuário

```
1 | a = 2
2 | b = 3
3 | c = a ** b
4 | print(c) # imprime 8
```

Exemplo de Código 1.15: Operador de multiplicação

interaja com seu programa, digitando valores para gerar resultados. Ainda, é possível criar saídas formatadas de diferentes maneiras. As subseções a seguir tratam desses tópicos.

1.4.1 Entrada de Dados

Em uma aplicação console em Python, a entrada de dados geralmente é feita usando a função `input()` e a saída de dados é feita usando a função `print()`. A função `input()` permite que o usuário forneça dados para o programa. Quando a função é chamada, ela exibe uma mensagem na tela e aguarda a entrada do usuário. O valor digitado pelo usuário é armazenado como uma *string*. O exemplo 1.16 ilustra o uso da função `input()`.

```
1 | nome = input("Digite seu nome: ")
2 | print("Olá, " + nome + "!")
```

Exemplo de Código 1.16: Entrada de dados em console

1.4.2 Saída de Dados

A função `print()` é usada para exibir resultados ou mensagens na tela console. Ela pode exibir uma *string* ou o resultado de uma expressão. O exemplo 1.17 ilustra o uso da função `print()`.

```
1 | nome = "Lucas"
2 | print("Olá, " + nome + "!") # imprime Olá, Lucas!
```

Exemplo de Código 1.17: Saída de dados em console

1.4.3 Saída Formatada

Para imprimir uma saída formatada em Python, você pode usar o método `format()` ou o operador de formatação de *string* `%`. O método `format()` é uma forma fácil de formatar *strings* e incorporar valores de variáveis nelas. O exemplo 1.18 mostra como usá-lo.


```
1 | nome = "Lucas"
2 | idade = 30
3 | print("Meu nome é {} e tenho {} anos".format(nome, idade))
4 | # imprime Meu nome é Lucas e tenho 30 anos
```

Exemplo de Código 1.18: O método `format()`

O operador de formatação de string `%` é uma forma mais antiga de formatar strings. O exemplo 1.19 mostra como usá-lo.

```
1 | nome = "Lucas"
2 | idade = 30
3 | print("Meu nome é %s e tenho %d anos" % (nome, idade))
4 | # imprime Meu nome é Lucas e tenho 30 anos
```

Exemplo de Código 1.19: O operador de formatação `%`

O operador `"f"` é uma forma mais recente e conveniente de formatar strings em Python. Ele permite que você insira valores de variáveis diretamente na *string*, usando chaves e o prefixo `"f"`. O exemplo 1.20 mostra como usá-lo.

```
1 | nome = "Lucas"
2 | idade = 30
3 | print(f"Meu nome é {nome} e tenho {idade} anos")
4 | # imprime Meu nome é Lucas e tenho 30 anos
```

Exemplo de Código 1.20: O operador de formatação `"f"`

O operador `"f"` é uma forma mais legível e concisa de formatar strings em comparação com outros métodos, e é amplamente utilizado na comunidade Python. Ele também permite a execução de expressões dentro das chaves e a formatação avançada de números, entre outras coisas.

1.5 Exercícios Propostos

Essa série de exercícios envolve os conceitos abordados neste capítulo. É importante que, além de resolver o problema, você faça uso dos recursos de formatação de strings para apresentar uma saída textual bem apresentável ao usuário.

Exercício 1.1 Armazene seu nome e idade em variáveis separadas e imprima uma saída formatada com elas. ■

Exercício 1.2 Armazene dois números em variáveis e imprima a soma, subtração, multiplicação e divisão deles. ■

Exercício 1.3 Peça ao usuário para digitar três números, armazene-os em variáveis e imprima a média aritmética dos três números. ■

Exercício 1.4 Peça ao usuário para digitar seu peso e altura, calcule o índice de massa corporal (IMC) e imprima o resultado. ■

Exercício 1.5 Peça ao usuário para digitar três números e imprima a soma deles. ■

Exercício 1.6 Peça ao usuário para digitar um número e calcule o seu dobro. ■

Exercício 1.7 Peça ao usuário para digitar o nome, o preço de custo, o preço de venda e a quantidade em estoque de determinado produto e mostre o lucro que esse estoque pode gerar se todos os produtos forem vendidos. ■

Exercício 1.8 Peça ao usuário para digitar um número e calcule a raiz quadrada desse número. ■

Exercício 1.9 Peça ao usuário para digitar um número e calcule o seno, cosseno e tangente desse número. ■

Exercício 1.10 Peça ao usuário para digitar dois números e calcule a potência do primeiro número elevado ao segundo. ■

Exercício 1.11 Peça ao usuário para digitar três números e calcule a fórmula de Bhaskara para esses números. ■

Exercício 1.12 Peça ao usuário para digitar o raio de um círculo e calcule a área e o comprimento do círculo. ■

Exercício 1.13 Peça ao usuário para digitar as dimensões de um retângulo (largura e altura) e calcule a área e o perímetro desse retângulo. ■

Exercício 1.14 Peça ao usuário para digitar a base e a altura de um triângulo e calcule a área desse triângulo. ■

Exercício 1.15 Peça ao usuário para digitar a distância e a velocidade inicial de um objeto e calcule o tempo que leva para atingir o solo. ■

Exercício 1.16 Peça ao usuário para digitar o valor inicial de um investimento, a taxa de juros e o número de anos e calcule o valor final do investimento. ■

Exercício 1.17 Peça ao usuário para digitar o preço de uma mercadoria, o desconto e o imposto e calcule o preço final da mercadoria. ■

Exercício 1.18 Peça ao usuário para digitar a massa e a aceleração de um objeto e calcule a força resultante. ■

Exercício 1.19 Peça ao usuário para digitar a velocidade final, a velocidade inicial e o tempo e calcule a aceleração. ■

Exercício 1.20 Peça ao usuário para digitar o valor da medida de um ângulo em radianos e calcule o valor desse ângulo em graus. ■

Exercício 1.21 Peça ao usuário para digitar o comprimento de dois lados de um triângulo retângulo e calcule o comprimento da hipotenusa. ■

Exercício 1.22 Peça ao usuário para digitar a distância percorrida por um objeto e o tempo gasto e calcule a velocidade média do objeto. ■

Exercício 1.23 Peça ao usuário para digitar a distância percorrida, o tempo gasto e aceleração e calcule a velocidade inicial e final do objeto. ■

Exercício 1.24 Calcular o perímetro de um círculo dado o seu raio como entrada. ■

Exercício 1.25 Calcular o volume de uma esfera dado o seu raio como entrada. ■

Exercício 1.26 Calcular a área de um triângulo retângulo dado as medidas dos seus catetos como entrada. ■

Exercício 1.27 Receber o nome, o salário e o valor do imposto de uma pessoa como entrada e imprimir o salário líquido. ■

Exercício 1.28 Crie uma lista com os nomes de 5 membros da sua família. ■

Exercício 1.29 Crie um dicionário com os nomes, idades e cor dos olhos de 5 membros da sua família. ■

Exercício 1.30 Crie dois conjuntos e combine-os usando as operações de união, interseção e diferença, apresentando os resultados de cada operação. ■

1.6 Considerações Sobre o Capítulo

Este capítulo apresentou os fundamentos da programação na linguagem Python. Foi mostrado como criar variáveis de diferentes tipos, como utilizar os operadores matemáticos básicos e como usar os comandos de entrada e saída de dados para se criar programas mais interativos. Por fim, foram propostos diversos exercícios para você colocar em prática aquilo que aprendeu ao longo do capítulo. No próximo capítulo, você verá como realizar operações mais avançadas em números, textos e coleções (listas, dicionários e conjuntos).



2. Estruturas Condicionais

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



3. Estruturas de Repetição



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



4. Números, Textos e Coleções

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



5. Funções

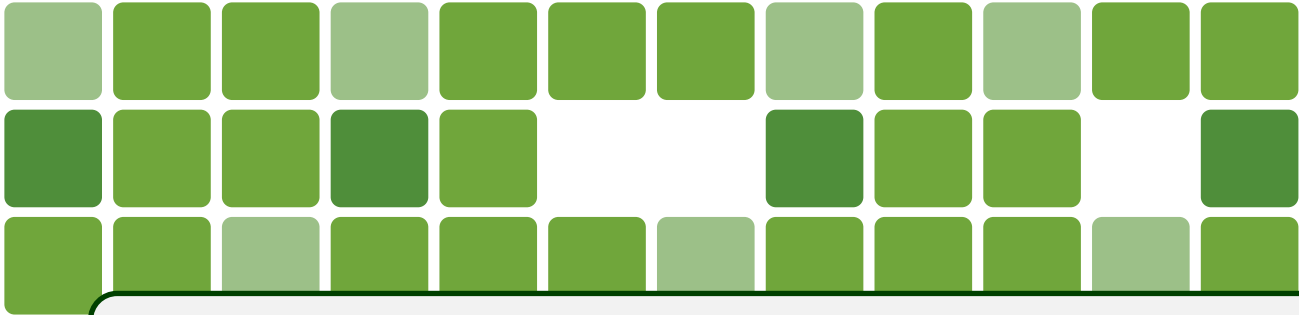


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



6. Tratamento de Exceções

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



7. Arquivos e Módulos



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



8. Classes e Objetos



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



9. Manipulação de Bancos de Dados SQL

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



10. Manipulação de Bancos de Dados NoSQL

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Bibliografia

Artigos

Livros