

# Licenciatura em Computação


# Programação de

# Computadores

---

Prof. Dr. Eduardo Ribeiro  
Universidade Federal do Tocantins





# Capítulo 1 - Desenvolvimento de Algoritmos em Linguagem de Programação

*"Programas devem ser escritos para que as pessoas possam ler, e apenas incidentalmente para  
que as máquinas possam executar."*

*Abelson e Sussman*



Bem-vindos ao Capítulo 1: Desenvolvimento de Algoritmos em Linguagem de Programação!


Neste capítulo, vamos dar início à nossa jornada no mundo da programação de computadores. Aqui, vocês serão apresentados aos conceitos fundamentais necessários para a criação de algoritmos usando uma linguagem de programação popular, como o Python.

Ao longo deste capítulo, exploraremos diversos tópicos cruciais que irão prepará-los para construir programas funcionais e eficientes. Vamos dar uma olhada em alguns dos principais temas que serão abordados:

- ☐ **Introdução à Programação:** Começaremos compreendendo o que é a programação de computadores, sua relevância e aplicações em diferentes áreas do conhecimento. Será fornecido um panorama sobre como os programas afetam nosso cotidiano e a sociedade como um todo.
- ☐ **Algoritmos:** Aprenderemos o conceito de algoritmo, que consiste em uma sequência lógica de instruções projetadas para executar uma tarefa específica. Veremos como é possível representar algoritmos de forma estruturada, utilizando pseudocódigo, fluxogramas ou mesmo a própria linguagem de programação.
- ☐ **Sintaxe Básica da Linguagem:** Será apresentada a sintaxe básica da linguagem de programação escolhida, neste caso, o Python. Você aprenderá como escrever um programa simples, incluindo informações sobre declaração de variáveis, estruturas de controle e exibição de resultados na tela.
- ☐ **Entrada e Saída de Dados:** Descobriremos como um programa pode interagir com o usuário, recebendo dados de entrada e exibindo resultados na tela. Você entenderá como utilizar funções de entrada, como a função `input()` do Python, e funções de saída, como a função `print()`.
- ☐ **Exemplos Práticos:** Para reforçar os conceitos aprendidos, trabalharemos com exemplos práticos de algoritmos simples. Esses exemplos incluirão cálculos matemáticos básicos, conversões de unidades e resolução de problemas simples do dia a dia.

Nosso objetivo é fornecer a vocês uma base sólida para o desenvolvimento de algoritmos e a tradução desses algoritmos para a linguagem de programação escolhida. Ao longo do capítulo, serão apresentados exemplos claros e práticos, estimulando-os a praticar e experimentar por conta própria.

Estejam preparados para uma jornada empolgante no universo da programação de computadores. O conhecimento adquirido nesta disciplina será a base para suas futuras conquistas como programadores competentes e confiantes.



Vamos iniciar essa fascinante jornada pelo mundo da programação. Aproveitem o Capítulo 1: Desenvolvimento de Algoritmos em Linguagem de Programação e bons estudos!

## **Seção 1.1 - Introdução a programação**

A programação de computadores é uma disciplina fundamental e empolgante que nos permite criar e desenvolver softwares, aplicativos e sistemas que impulsionam o mundo atual. É através da programação que podemos dar vida a ideias, resolver problemas complexos e aproveitar todo o potencial da tecnologia.


Mas, afinal, o que é a programação de computadores? Podemos defini-la como um conjunto de instruções lógicas e precisas que guiam um computador na execução de tarefas específicas. Através da escrita de algoritmos e do uso de linguagens de programação, somos capazes de projetar soluções que automatizam processos, processam dados, interagem com usuários e muito mais.

A relevância da programação na sociedade moderna é inegável. Ela está presente em praticamente todos os aspectos de nossas vidas. Através da programação, podemos desenvolver aplicativos móveis que facilitam nossa comunicação, jogos que nos entretêm e sistemas de gerenciamento que tornam nossas tarefas mais eficientes. A programação permeia a indústria, a medicina, a educação, as ciências, a engenharia, o entretenimento e praticamente todas as áreas do conhecimento.

Os programas de computador afetam nosso cotidiano de maneiras significativas. Pense em como a internet, que é um resultado da programação, revolucionou a forma como nos comunicamos, compartilhamos informações e realizamos transações financeiras. Considere como os aplicativos de transporte mudaram a maneira como nos locomovemos nas cidades. Observe como os avanços na inteligência artificial estão transformando a maneira como interagimos com assistentes virtuais, sistemas de recomendação e carros autônomos.

Além disso, a programação nos permite resolver problemas complexos e enfrentar desafios globais. Por exemplo, podemos utilizar algoritmos para analisar grandes volumes de dados e obter insights valiosos para áreas como pesquisa científica, previsão de desastres naturais, descoberta de medicamentos e combate a crimes.

Em resumo, a programação de computadores é uma habilidade poderosa que abre portas para oportunidades ilimitadas. Compreender os princípios da programação não apenas permite que você desenvolva suas próprias ideias e projetos, mas também o capacita a ser um solucionador de problemas criativo e inovador.



Nesta seção, exploraremos os fundamentos da programação, desde a estruturação de algoritmos até o desenvolvimento de soluções práticas usando uma linguagem de programação. Prepare-se para embarcar em uma jornada emocionante e desafiadora, na qual você irá desvendar os segredos do mundo da programação de computadores.

Estamos entusiasmados para acompanhá-lo nesta jornada e vê-lo explorar o poder da programação em várias áreas do conhecimento. Vamos começar essa incrível aventura rumo ao domínio da programação de computadores!

À medida que avançamos na era digital, a importância da programação de computadores só tende a aumentar. As habilidades em programação são cada vez mais requisitadas em diferentes setores e profissões, seja no desenvolvimento de software, na análise de dados, na criação de sites, na automação de processos, entre tantas outras aplicações.


A programação não se limita apenas aos especialistas em computação. Ela se tornou uma habilidade essencial para uma ampla gama de profissionais, desde cientistas de dados e engenheiros até designers e profissionais de marketing. O conhecimento em programação permite que esses profissionais expandam suas capacidades, resolvam problemas de forma eficiente e se adaptem às rápidas mudanças tecnológicas.

Além disso, a programação proporciona um pensamento analítico e lógico aprimorado, além de desenvolver a capacidade de resolver problemas de maneira estruturada. O processo de programação envolve a decomposição de problemas complexos em partes menores, a identificação de padrões e a formulação de soluções algorítmicas. Essas habilidades de pensamento computacional são altamente valorizadas em diversas áreas do conhecimento.

Outro aspecto importante é o fato de que a programação permite a expressão criativa. Ao escrever código, você pode dar vida a suas ideias, criar aplicativos, jogos, arte interativa e muito mais. A programação é uma forma de expressão artística, onde sua imaginação e criatividade podem se manifestar em projetos concretos e tangíveis.

No entanto, a importância da programação vai além das oportunidades profissionais e do aspecto criativo. Ela também desempenha um papel crucial na capacitação das pessoas para a era digital. Compreender os princípios da programação nos torna cidadãos digitais mais informados e capacitados. Nos permite entender melhor como os sistemas tecnológicos funcionam, como os dados são processados e como a informação é transmitida.

Em uma sociedade cada vez mais dependente de tecnologia, é fundamental que todos tenham pelo menos uma compreensão básica da programação. Isso nos capacita a tomar decisões




informadas, a participar ativamente do desenvolvimento tecnológico e a contribuir para um futuro onde a tecnologia seja usada de forma responsável e ética.

Nesta disciplina, vamos explorar os conceitos essenciais da programação de computadores, desde os algoritmos e estruturas de controle até a resolução de problemas reais. Você terá a oportunidade de praticar e aplicar seus conhecimentos em projetos práticos e desafiadores.

A programação é uma ferramenta poderosa para resolver problemas do mundo real em diversas áreas. Aqui estão alguns exemplos práticos de problemas que podem ser solucionados por meio da programação:

- ☐ Gerenciamento de estoque: Uma empresa precisa acompanhar e controlar seu estoque de produtos. A programação pode ser usada para desenvolver um sistema automatizado de gerenciamento de estoque, que rastreia as quantidades disponíveis, registra vendas e emite alertas quando os níveis de estoque estão baixos.
- ☐ Análise de dados: Com a explosão de dados disponíveis atualmente, a programação é essencial para analisar e extrair insights significativos. Por exemplo, um cientista de dados pode escrever programas para analisar grandes conjuntos de dados, identificar padrões, realizar previsões ou criar modelos estatísticos.
- ☐ Automatização de tarefas: Tarefas repetitivas e demoradas podem ser automatizadas por meio da programação. Por exemplo, um programador pode escrever um script para automatizar o envio de e-mails personalizados para uma lista de contatos, economizando tempo e esforço.
- ☐ Aplicativos móveis: Os aplicativos móveis são uma parte integrante de nossas vidas atualmente. Através da programação, é possível desenvolver aplicativos que fornecem serviços e informações relevantes para os usuários, como aplicativos de entrega de comida, rastreamento de exercícios físicos ou plataformas de aprendizado online.
- ☐ Simulações e modelagem: Programas podem ser usados para simular e modelar fenômenos complexos em áreas como física, engenharia, medicina e finanças. Por exemplo, um engenheiro pode escrever um programa para simular o comportamento de uma estrutura durante um terremoto, a fim de avaliar sua resistência e identificar áreas de melhoria.
- ☐ Segurança da informação: A programação desempenha um papel fundamental na proteção de informações sensíveis. Programas podem ser desenvolvidos para implementar criptografia de dados, autenticação de usuários e detecção de ameaças de segurança em redes.
- ☐ Desenvolvimento web: A programação é essencial para o desenvolvimento de sites e aplicativos web. Programadores utilizam linguagens como HTML, CSS e JavaScript para



criar interfaces interativas, implementar funcionalidades avançadas e garantir uma experiência de usuário agradável.

Esses são apenas alguns exemplos de como a programação pode ser aplicada para resolver problemas do mundo real. A capacidade de criar soluções personalizadas e adaptadas às necessidades específicas de cada contexto torna a programação uma ferramenta versátil e valiosa em diversos setores e disciplinas. Através da programação, podemos impulsionar a inovação, melhorar processos e tornar nossas vidas mais eficientes e conectadas.

## **Seção 1.2 - Algoritmos - A Base da Programação**

Na programação de computadores, o conceito de algoritmo é fundamental. Um algoritmo pode ser definido como uma sequência lógica de instruções que visa resolver um problema específico ou realizar uma tarefa. Podemos compará-lo a uma receita de bolo, onde cada passo é cuidadosamente definido para atingir o resultado desejado.

Ao desenvolver algoritmos, buscamos estruturar as instruções de forma clara, coerente e lógica, de modo que qualquer pessoa - ou, no caso, um computador - possa entender e executar as ações necessárias para alcançar o objetivo proposto.

Para representar algoritmos de forma estruturada, existem diversas técnicas amplamente utilizadas. Uma delas é o pseudocódigo, que se trata de uma descrição em linguagem simples e semelhante à linguagem humana, permitindo a expressão dos passos necessários para a resolução de um problema. O pseudocódigo é uma abordagem mais informal e flexível, facilitando o entendimento e a comunicação do algoritmo.

Outra técnica comum é o uso de fluxogramas, que são diagramas visuais que representam a sequência de instruções do algoritmo por meio de formas geométricas e setas. Cada forma representa uma ação específica, como uma decisão ou um bloco de código. Os fluxogramas são particularmente úteis para ilustrar a lógica do algoritmo e ajudar na visualização e compreensão de sua estrutura.

Além disso, os algoritmos podem ser representados diretamente em uma linguagem de programação específica. Nesse caso, cada instrução é escrita seguindo a sintaxe da linguagem escolhida. Isso permite que o algoritmo seja traduzido diretamente para um programa executável.

Vejamos um exemplo simples de algoritmo para ilustrar esse conceito:

Unset

**Algoritmo: Calcular a média de dois números**

1. Ler o primeiro número
2. Ler o segundo número
3. Somar os dois números e armazenar o resultado em uma variável "soma"
4. Dividir a soma por 2 para obter a média e armazenar o resultado em uma variável "media"
5. Exibir o valor da média na tela
6. Fim do algoritmo

Nesse exemplo, temos um algoritmo que calcula a média de dois números. Cada passo é claramente definido, desde a leitura dos números até a exibição do resultado final. Esse algoritmo pode ser representado em pseudocódigo, em um fluxograma ou até mesmo em uma linguagem de programação como o Python.

Os algoritmos são a base da programação, pois são eles que nos permitem solucionar problemas de forma estruturada e lógica. A capacidade de desenvolver algoritmos eficientes e bem projetados é essencial para o sucesso na programação. Ao compreender o conceito de algoritmo e sua representação, você estará preparado para avançar na construção de programas mais complexos e sofisticados.


Ao longo deste curso, você terá a oportunidade de aprimorar suas habilidades na criação e implementação de algoritmos em Python. Lembre-se de praticar, experimentar e refinar suas soluções algorítmicas. Afinal, a programação é um processo contínuo de aprendizado e aprimoramento.

Prepare-se para dominar a arte dos algoritmos e explorar todo o potencial da programação de computadores. Estamos empolgados para acompanhá-lo nessa jornada emocionante!

### **Seção 1.3 - A linguagem de Programação Python**

Considerando que vocês já tiveram contato com o conceito de algoritmos em disciplinas anteriores, estamos empolgados em iniciar diretamente a programação em uma linguagem





específica: o Python. Nesta disciplina, daremos continuidade ao seu aprendizado, aplicando os conceitos de algoritmos em uma linguagem de programação real.

Linguagens de programação são conjuntos de regras e símbolos que permitem aos programadores escreverem códigos para instruir os computadores a realizar tarefas específicas. Elas fornecem uma forma estruturada e precisa de se comunicar com a máquina, permitindo que você crie programas e aplicativos.

Existem diversas linguagens de programação disponíveis, cada uma com suas características, sintaxe e propósitos específicos. Algumas linguagens são mais adequadas para o desenvolvimento de aplicativos móveis, enquanto outras são mais utilizadas para a análise de dados ou desenvolvimento web. A escolha da linguagem de programação depende do contexto e das necessidades do projeto em questão.


Uma das linguagens de programação mais populares e amplamente utilizadas é o Python. O Python é uma linguagem de programação de alto nível, de código aberto e com uma sintaxe simples e legível. É conhecida por sua facilidade de aprendizado e expressividade, o que a torna uma excelente escolha para iniciantes e programadores experientes.

O Python foi escolhido como a linguagem de programação principal para este curso devido à sua popularidade, versatilidade e facilidade de aprendizado. Ele é amplamente utilizado em diferentes áreas, desde desenvolvimento web e análise de dados até automação e inteligência artificial.

Ao nos concentrarmos diretamente na programação em Python, vocês terão a oportunidade de aplicar os algoritmos que já aprenderam e traduzi-los em código real. Vamos explorar a sintaxe e as estruturas de controle específicas do Python, permitindo que vocês criem programas funcionais e executáveis.

Durante o curso, forneceremos exemplos práticos e exercícios para consolidar seu conhecimento em programação Python. Vocês terão a oportunidade de desenvolver projetos reais e resolver problemas do mundo real, utilizando todas as habilidades que adquiriram nas disciplinas anteriores.

Embora estejamos partindo diretamente para a programação em Python, não deixaremos de revisar os conceitos essenciais, como algoritmos, estruturas de controle e boas práticas de programação. Essa revisão será integrada ao longo das aulas, fornecendo a vocês um entendimento completo da linguagem e garantindo que vocês se tornem programadores proficientes.



Lembre-se de que a programação é uma habilidade que se desenvolve com prática e experimentação. À medida que avançamos no curso, incentivamos vocês a explorar, criar e solucionar problemas utilizando a linguagem Python. Através da programação, vocês estarão capacitados para criar soluções inovadoras e contribuir para o mundo da tecnologia.

Estejam preparados para uma jornada emocionante na programação em Python. Nossos objetivos são aprofundar seu conhecimento em algoritmos e capacitar vocês a desenvolverem programas reais e funcionais. Estamos ansiosos para acompanhá-los nessa nova etapa e ajudá-los a se tornarem programadores habilidosos em Python.

Vamos mergulhar de cabeça na programação em Python e explorar todas as possibilidades dessa linguagem poderosa!

Python possui uma ampla gama de bibliotecas e frameworks que tornam a programação mais eficiente e produtiva. Com o Python, é possível desenvolver aplicativos web, científicos, de automação, aprendizado de máquina, entre outros. Sua popularidade é atribuída à sua versatilidade, comunidade ativa e vasta documentação.

Ao converter um algoritmo para uma linguagem de programação como o Python, você está traduzindo as instruções lógicas em uma forma que a máquina pode entender e executar. Isso envolve utilizar a sintaxe e as estruturas específicas da linguagem para implementar cada passo do algoritmo.

Aqui está o mesmo exemplo do algoritmo mostrado na seção anterior traduzido para a linguagem Python:

Python

**# Calcular a média de dois números**

**# Passo 1: Ler o primeiro número**

**numero1 = float(input("Digite o primeiro número: "))**

**# Passo 2: Ler o segundo número**

**numero2 = float(input("Digite o segundo número: "))**

**# Passo 3: Somar os dois números e armazenar o resultado em uma variável "soma"**

```
soma = numero1 + numero2

# Passo 4: Dividir a soma por 2 para obter a média e armazenar o
# resultado em uma variável "media"

media = soma / 2

# Passo 5: Exibir o valor da média na tela

print("A média dos números é:", media)

# Passo 6: Fim do algoritmo
```

Nesse código em Python, as etapas do algoritmo são traduzidas em instruções específicas. O programa solicita ao usuário que digite os dois números, realiza a soma deles e armazena o resultado na variável "soma". Em seguida, a média é calculada dividindo a soma por 2 e armazenada na variável "media". Por fim, o resultado é exibido na tela utilizando a função print().

Agora você pode executar esse código Python e obter a média dos dois números inseridos. Lembre-se de que, ao ler os números, utilizamos a função float(input()) para garantir que os valores sejam tratados como números decimais (ponto flutuante).

No exemplo anterior do cálculo da média de dois números, cada passo foi convertido para código Python. Os comandos input() são usados para solicitar a entrada do usuário, os operadores matemáticos + e / são utilizados para realizar as operações de soma e divisão, e a função print() é usada para exibir o resultado na tela.

Ao converter algoritmos para uma linguagem de programação, é importante levar em consideração a sintaxe, as estruturas de controle e as convenções da linguagem escolhida. Cada linguagem possui suas peculiaridades, e é necessário se familiarizar com sua documentação e recursos para escrever código correto e eficiente.

A conversão de algoritmos para uma linguagem de programação permite que você transforme suas ideias em programas executáveis. Ela capacita você a criar soluções práticas para problemas do mundo real e explorar todo o potencial da programação.

À medida que avançamos neste curso, você terá a oportunidade de aprimorar suas habilidades de conversão de algoritmos para a linguagem Python. Você aprenderá a utilizar a sintaxe e as

estruturas da linguagem para implementar algoritmos complexos e desenvolver soluções funcionais.

Esteja preparado para mergulhar no mundo da programação com Python e experimentar todo o potencial dessa linguagem poderosa. A conversão de algoritmos para código Python abrirá as portas para um novo nível de criatividade e solução de problemas. Estamos empolgados para acompanhá-lo nessa jornada emocionante!

Para aqueles que estão entrando no mundo da programação pela primeira vez e ainda não tiveram experiência com compiladores ou ambientes de desenvolvimento, gostaríamos de apresentar uma ferramenta bastante útil: o Google Colab.

O Google Colab é uma plataforma online gratuita que permite escrever, executar e compartilhar



código em Python sem a necessidade de instalar qualquer software no seu computador. Ele é baseado no Jupyter Notebook, uma interface interativa que combina código, texto explicativo e elementos visuais em um único documento.

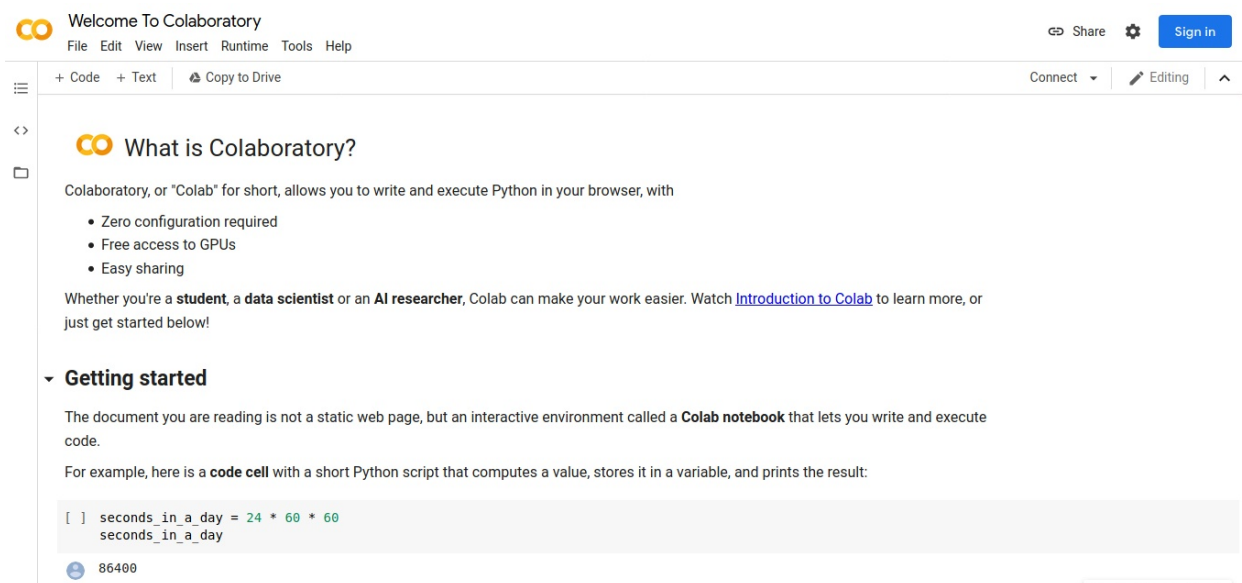
Uma das principais vantagens do Google Colab é a sua facilidade de uso. Você pode acessá-lo diretamente através do navegador, seja em um computador, tablet ou até mesmo em um smartphone. Isso significa que você não precisa se preocupar em configurar ambientes de desenvolvimento complexos em sua máquina pessoal.

Ao usar o Google Colab, você pode escrever seu código Python em células individuais, que podem ser executadas independentemente. Isso permite testar e depurar seu código de forma incremental, facilitando a compreensão e a visualização do fluxo do programa.

Além disso, o Google Colab oferece recursos adicionais, como a possibilidade de adicionar texto explicativo, gráficos, imagens e até mesmo vídeos aos seus projetos. Essa combinação de código e elementos visuais torna o processo de aprendizado mais interativo e envolvente.

Para utilizar o Google Colab, basta acessar o site [colab.research.google.com](https://colab.research.google.com) e criar um novo notebook. Você pode escrever seu código Python em cada célula, executá-lo clicando no botão de reprodução ou usando o atalho de teclado "Shift + Enter". O resultado da execução será exibido logo abaixo da célula correspondente.

Outra vantagem do Google Colab é a capacidade de compartilhar seus notebooks com outras pessoas. Você pode colaborar com colegas de classe, compartilhar seus projetos com instrutores e até mesmo acessar notebooks compartilhados por outros programadores em comunidades online.




Portanto, se você nunca viu um compilador na vida ou está apenas começando sua jornada na programação, o Google Colab é uma ferramenta valiosa para você experimentar, praticar e desenvolver suas habilidades em Python. Ele oferece um ambiente amigável e acessível para aprender e escrever código, sem as complicações iniciais de configurações de desenvolvimento.

Nosso curso utilizará o Google Colab para fornecer exemplos, exercícios e práticas em Python. Estaremos disponíveis para ajudá-los a aproveitar ao máximo essa ferramenta ao longo de nosso tempo juntos.

Agora vocês estão prontos para iniciar sua jornada no mundo da programação com o auxílio do Google Colab. Preparem-se para se divertir e se envolver na criação de código Python de forma prática e interativa.

## Seção 1.4- Sintaxe Básica da Linguagem - Dominando os Fundamentos do Python

Nesta seção, vamos mergulhar na sintaxe básica da linguagem de programação Python. A sintaxe é o conjunto de regras e estruturas que definem como o código em Python deve ser escrito. Compreender e dominar a sintaxe básica é essencial para escrever programas funcionais e eficientes.



Vamos começar explicando como escrever um programa simples em Python. Um programa em Python consiste em uma sequência de instruções que são executadas de forma linear, a menos que haja estruturas de controle para modificar o fluxo do programa. Aqui está um exemplo de um programa simples em Python:

Python

```
# Exemplo de um programa simples em Python

# Declaração de uma variável nome = "Maria"

# Exibição de uma mensagem na tela

nome="Maria"

print("Olá, " + nome + "! Bem-vindo ao mundo da programação em
Python!")
```

Neste exemplo, declaramos uma variável chamada nome e atribuímos a ela o valor "Maria". Em seguida, usamos a função print() para exibir uma mensagem na tela, concatenando a variável nome com a frase de boas-vindas. A função print() é utilizada para exibir resultados ou mensagens para o usuário.

Uma característica importante em Python é que não é necessário declarar o tipo da variável explicitamente. O tipo da variável é inferido automaticamente com base no valor atribuído a ela. No exemplo acima, a variável nome é do tipo string (texto), pois atribuímos a ela uma sequência de caracteres entre aspas.

Lembre-se de que a indentação (recuo) é fundamental em Python. A estrutura do código é definida pela indentação, ou seja, pela quantidade de espaços ou tabulações no início de cada linha. A indentação correta é crucial para que o programa funcione corretamente.

À medida que avançamos no curso, vocês terão a oportunidade de praticar e aplicar esses conceitos em exercícios e projetos práticos. É fundamental praticar a escrita de código, experimentar diferentes estruturas e desafiar-se com problemas do mundo real.

A sintaxe básica da linguagem Python é a base para a construção de programas complexos e soluções eficientes. Estejam preparados para se aprofundar nesses conceitos e desenvolver habilidades sólidas de programação em Python.

## Seção 1.5- Entrada e Saída de Dados - Interagindo com o Usuário em Python

Nesta seção, vamos explorar como os programas em Python podem interagir com o usuário, permitindo a entrada de dados e exibindo resultados na tela. A capacidade de interagir com o usuário é fundamental para criar programas dinâmicos e úteis.

Para receber dados de entrada do usuário, podemos utilizar funções de entrada, como a função `input()` em Python. Essa função permite que o programa solicite ao usuário que digite um valor ou uma informação específica. Aqui está um exemplo simples de como utilizar a função `input()`:

Python

```
# Solicitar ao usuário que digite seu nome
nome = input("Digite seu nome: ")

# Exibir uma mensagem de boas-vindas com o nome do usuário
print("Olá,", nome, "! Bem-vindo ao nosso programa!")
```

Nesse exemplo, a função `input()` é utilizada para solicitar ao usuário que digite seu nome. O valor digitado pelo usuário é então armazenado na variável `nome`. Em seguida, utilizamos a função `print()` para exibir uma mensagem de boas-vindas, incluindo o nome do usuário.

Além de receber dados de entrada, também é essencial exibir resultados ou mensagens na tela para que o usuário possa visualizá-los. A função `print()` é a principal forma de exibir informações na tela em Python. Vejamos alguns exemplos de como utilizar a função `print()`:

Python

```
# Exibir uma mensagem simples na tela
print("Olá, mundo!")
```

Python

```
# Exibir o resultado de um cálculo

numero1 = 10

numero2 = 5

soma = numero1 + numero2

print("A soma dos números é:", soma)
```

Python

```
# Exibir múltiplos valores separados por vírgula

nome = "Maria"

idade = 25

print("Nome:", nome, "| Idade:", idade)
```

Nesses exemplos, utilizamos a função `print()` para exibir mensagens na tela. Na primeira linha, temos uma mensagem simples "Olá, mundo!". Em seguida, realizamos um cálculo de soma e exibimos o resultado junto com uma mensagem descritiva. Por fim, exibimos múltiplos valores separados por vírgula, permitindo a exibição de informações concatenadas.

Lembrando que a função `print()` aceita diferentes tipos de dados como argumentos e os converte automaticamente para sua representação em texto. Isso nos permite exibir resultados de cálculos, informações armazenadas em variáveis e até mesmo combinar textos e valores em uma única mensagem.

A capacidade de interagir com o usuário através da entrada de dados e da exibição de resultados é fundamental para tornar os programas mais interativos e úteis. Você pode criar programas que solicitem informações específicas, realizem cálculos com base nos dados de entrada e exibam resultados personalizados para o usuário.



Ao longo do curso, vocês terão a oportunidade de explorar mais recursos e funcionalidades avançadas relacionadas à entrada e saída de dados em Python. Vocês aprenderão a formatar saídas, lidar com diferentes tipos de dados e realizar interações mais complexas com o usuário.

Aqui estão mais alguns exemplos simples de códigos em Python:

Calcular a área de um retângulo:

Python

```
# Calcular a área de um retângulo

largura = 5

altura = 3

area = largura * altura

print("A área do retângulo é:", area)
```

Python

```
# Encontrar o maior entre três números

num1 = 10

num2 = 7

num3 = 12

maior = max(num1, num2, num3)

print("O maior número é:", maior)
```

Esses exemplos abordam diferentes aspectos e conceitos básicos da programação em Python, como cálculos matemáticos, entrada de dados e manipulação de strings. Eles fornecem uma base para você explorar e praticar a escrita de código em Python.

Lembre-se de executar esses códigos em um ambiente Python adequado, como o Google Colab, ou em seu próprio ambiente de desenvolvimento local. Pratique a modificação desses exemplos, experimente diferentes valores e desafie-se com novos problemas.

À medida que você avança no aprendizado, os exemplos de código se tornam mais complexos e abrangem uma variedade maior de tópicos. Continue se dedicando, explorando e aprofundando seu conhecimento em Python. A prática constante é a chave para se tornar um programador habilidoso.

Aqui estão alguns exemplos práticos de algoritmos simples em Python para reforçar os conceitos aprendidos:

Python

```
# Cálculo da área de um círculo

raio = float(input("Digite o raio do círculo: "))

area = 3.14159 * raio**2

print("A área do círculo é:", area)
```

Python

```
# Conversão de temperatura de Celsius para Fahrenheit e Kelvin

celsius = float(input("Digite a temperatura em Celsius: "))

fahrenheit = (celsius * 9/5) + 32

kelvin = celsius + 273.15

print("A temperatura em Fahrenheit é:", fahrenheit)

print("A temperatura em Kelvin é:", kelvin)
```

Python

```
# Cálculo do desconto em uma compra

valor_compra = float(input("Digite o valor da compra: "))
desconto = float(input("Digite o percentual de desconto: "))
valor_desconto = valor_compra * desconto / 100
valor_final = valor_compra - valor_desconto

print("O valor do desconto é:", valor_desconto)
print("O valor final da compra com desconto é:", valor_final)
```

Python

```
# Cálculo do IMC (Índice de Massa Corporal)

peso = float(input("Digite o seu peso em kg: "))
altura = float(input("Digite a sua altura em metros: "))

imc = peso / altura**2


print("Seu IMC é:", imc)
```

Esses exemplos abrangem diferentes cenários, desde cálculos matemáticos básicos até conversões de unidades e resolução de problemas simples do dia a dia. Eles permitem que você aplique os conceitos aprendidos, pratique a escrita de código em Python e obtenha resultados úteis.

Lembre-se de que a programação é uma habilidade prática que se desenvolve com a prática constante. Desafie-se a modificar esses exemplos, experimentar diferentes valores e criar soluções personalizadas. Quanto mais você praticar, mais confiança e habilidade você ganhará na resolução de problemas com programação.

## EXERCÍCIOS DE FIXAÇÃO

Aqui está uma lista de seis exercícios para você praticar em casa utilizando o Google Colab:

- 
1. Exercício de Cálculo de Média:
    - a. Escreva um programa em Python que solicite ao usuário três notas (de 0 a 10) e calcule a média dessas notas. Em seguida, exiba a média na tela.
  2. Exercício de Conversão de Horas:
    - a. Escreva um programa em Python que solicite ao usuário um número inteiro que represente a quantidade de horas e converta esse valor para dias e horas. Exiba o resultado na tela.
  3. Exercício de Aumento de Salário
    - a. 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.
  4. Exercício de Tempo de Viagem
    - a. 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.
  5. Exercício de Tempo de Vida
    - a. 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.

Esses exercícios abordam diferentes aspectos da programação em Python, como cálculos matemáticos, manipulação de strings e estruturas de repetição. Eles permitem que você pratique os conceitos aprendidos e desenvolva suas habilidades de programação.

Lembre-se de que a prática é essencial para o aprendizado da programação. Tente resolver os exercícios por conta própria, utilizando os conceitos e técnicas aprendidos durante o curso. Caso tenha alguma dificuldade, você sempre pode consultar a documentação do Python ou pedir ajuda aos seus colegas de estudo.