

Lógica de Programação - Aula 14 - Fundamentos da Construção do Algoritmo

Plano de Aula: Prática Integrada em Python - Aula 14

Disciplina: Lógica de Programação

Módulo: Fundamentos de Construção de Algoritmo

Aula: 14 - Fundamentos da Construção do Algoritmo

Conteúdos:

- Atividade prática de fixação dos conceitos de variáveis, tipos de dados e operadores.
 - Exercícios de conversão de pseudocódigo para Python.
-

1. Abertura (5 min) – Engajamento Inicial

Olá a todos! Nas últimas aulas, nós nos aprofundamos nos fundamentos teóricos da programação, e na aula 12, demos o nosso primeiro passo em **Python**. Vocês aprenderam sobre **variáveis**, **tipos de dados** e **operadores aritméticos**, que são os blocos de construção de qualquer programa.

Pensem em um atleta que acabou de aprender os fundamentos do seu esporte: como segurar a raquete, como posicionar os pés, como lançar a bola. Ele não vai para a competição sem antes praticar, praticar e praticar. Da mesma forma, a nossa aula de hoje será 100% prática.

O objetivo é que vocês consolidem o que aprenderam sobre os conceitos básicos do Python e que comecem a pensar "na linguagem do computador" de forma mais fluida. Ao final da aula, vocês não apenas terão resolvido os problemas, mas também terão uma base sólida para enfrentar os desafios mais complexos que vêm por aí, como as estruturas de decisão e repetição.

2. Desenvolvimento (30 min) – Conteúdo + Prática

Parte Expositiva (5 min): Da Teoria à Prática

Antes de mergulharmos nos exercícios, vamos fazer uma revisão rápida e um lembrete crucial:

- **Variáveis:** São como caixas nomeadas para guardar informações. Em Python, você as cria simplesmente dando um nome e atribuindo um valor, como `nome = "João"`.
- **Tipos de Dados:** O tipo de informação que a variável guarda. Lembrem-se dos principais: `str` (texto), `int` (números inteiros), `float` (números decimais) e `bool` (verdadeiro ou falso).
- **Operadores Aritméticos:** Os símbolos para realizar cálculos, como `+`, `-`, `*`, `/`.

Agora, a parte mais importante: a conversão. Para cada problema, a nossa mente deve seguir um processo claro:

1. **Entender o Problema:** O que ele pede? Quais são os dados de entrada e saída?
2. **Planejar em Pseudocódigo:** Esboçar a solução, passo a passo, como fizemos nas aulas anteriores.
3. **Traduzir para Python:** Transformar cada linha do pseudocódigo em código Python.

Parte Interativa (25 min): Desafios de Conversão

Agora, vamos colocar a mão na massa. Em duplas, vocês terão dois desafios que já resolvemos em pseudocódigo e fluxograma. A tarefa é traduzi-los e executá-los em Python.

Desafio 1: Conversor de Temperatura

- **Problema:** Crie um programa Python que converta uma temperatura de Celsius para Fahrenheit.
- **Fórmula:** $Fahrenheit = (Celsius * 9/5) + 32$.
- **Tarefas:**
 1. Peça ao usuário para digitar a temperatura em Celsius.
 2. Converta o valor de entrada para o tipo numérico correto.
 3. Aplique a fórmula de conversão.
 4. Exiba o resultado com uma mensagem clara.

Desafio 2: Calculadora de Área do Círculo

- **Problema:** Crie um programa Python que calcule a área de um círculo.
- **Fórmula:** $A_{\text{rea}} = \pi * R^2$. Use $\pi = 3.14$.
- **Tarefas:**
 1. Peça ao usuário para digitar o raio do círculo.
 2. Converta o valor de entrada para o tipo numérico correto.
 3. Aplique a fórmula da área.
 4. Exiba o resultado com uma mensagem clara.

Metodologia:

1. Formem as duplas e abram o **Google Colab** ou o **IDLE**.
2. Comecem pelo Desafio 1 e, ao final de 10 minutos, passem para o Desafio 2.
3. O professor circulará pela sala, auxiliando e tirando dúvidas. O foco da ajuda será na sintaxe do Python e na conversão de tipos.
4. Ao final da atividade, pedirei para algumas duplas compartilharem a tela e executarem seus códigos, e faremos uma breve análise da solução.

Esta atividade é a peça-chave para fixar os conceitos. O ato de traduzir a lógica de uma linguagem para outra é um dos exercícios mais importantes para qualquer programador iniciante.

3. Fechamento (10 min) – Revisão e Reflexão

Revisão Rápida (5 min):

- Qual foi a principal dificuldade ao traduzir os algoritmos para Python? (Provavelmente a conversão de tipo com `float(input())` ou a sintaxe de exibição com `print()`).
- Por que o Python é considerado uma linguagem de "alto nível" quando comparado ao pseudocódigo? (Porque ele executa as instruções de forma real e direta, enquanto o pseudocódigo é apenas um plano).
- Qual a importância de um nome de variável claro, como `temperatura_celsius` em vez de apenas `t`? (Melhora a legibilidade do código, facilitando a manutenção futura).

Aplicação (5 min):

A partir desta aula, a sua confiança para enfrentar a programação vai aumentar. Vocês já podem resolver problemas básicos, recebendo dados, realizando cálculos e exibindo resultados. Isso é a fundação para tudo que faremos a seguir. Na próxima aula, vamos dar o passo mais importante: aprender a criar programas que **tomam decisões** com as estruturas de condição (`if`, `elif`, `else`).

4. Finalização (5 min) – Conclusão e Feedback

Para encerrar, pensem na seguinte pergunta:

"Se um programa sem variáveis e operadores é como uma calculadora sem números e símbolos, qual a importância de dominar esses conceitos para construir qualquer coisa mais complexa?"

Atividade de Aprofundamento (Para Casa)

Atividade 1 - Desafio "Calculadora de Imposto de Renda":

Crie um programa em Python que calcule o imposto de renda de um salário. A taxa de imposto é de 15% sobre o salário. O programa deve pedir ao usuário para digitar o salário atual, calcular o valor do imposto e exibir o salário líquido (salário atual - imposto).

Fontes de Pesquisa e Dicas de Leitura

- **Livros:**
 - **Python para Todos** (disponível online): É um excelente livro gratuito que cobre os conceitos básicos de forma detalhada e com exemplos práticos.
 - **Algoritmos e Lógica de Programação** (2ª ed.) - Marco A. Furlan de Souza: Revise o apêndice sobre Python para ter uma referência rápida de sintaxe.
- **Dicas de Conteúdo Complementar:**
 - **YouTube:** Assista a vídeos sobre "Python float function" ou "Python arithmetic operators" para ver diferentes exemplos e entender como eles se comportam em diversas situações.
 - **Google Colab:** Continue usando essa ferramenta para praticar e compartilhar seus códigos com facilidade.
 - **Exercícios Online:** Visite plataformas como a **w3schools** para fazer exercícios interativos sobre os conceitos básicos de Python.

A prática é a única forma de aprender a programar. Dediquem tempo aos exercícios e não se frustrem com os erros. Cada erro é uma oportunidade de aprendizado.