Educação Profissional Paulista

Técnico em Desenvolvimento de Sistemas

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

Estrutura básica

Operadores e expressões

Aula 4

Código da aula: [SIS]ANO1C1B2S15A4

Exposição

Objetivo da Aula

- Conhecer como os operadores de atribuição são usados para designar valores a variáveis, essenciais para armazenar e manipular dados em programas.

Competências da Unidade (Técnicas e Socioemocionais)

- Desenvolver sistemas computacionais utilizando ambiente de desenvolvimento;
- Migrar sistemas implementando rotinas e estruturas de dados mais eficazes;
- Trabalhar a criatividade e o comprometimento na resolução de problemas computacionais.

Recursos Didáticos

- Recurso audiovisual para a exibição de vídeos e imagens.
- Folhas sulfite, canetas coloridas e lápis.

Duração da Aula

50 minutos

Exposição

Operadores de atribuição e sua eficiência

Operadores de atribuição são usados para designar valores a variáveis, essenciais para armazenar e manipular dados em programas. Eles também influenciam a legibilidade e a eficiência do código.

- Tipos de operadores de atribuição (simples e composto).
- Uso eficiente em loops e funções.
- Impacto na legibilidade e na manutenção do código.

Exposição

Tipos de operadores de atribuição

Os operadores de atribuição são usados para atribuir valores a variáveis em muitas linguagens de programação. Existem dois tipos principais: simples e composto.

a) Simples: o operador de atribuição simples é o sinal de igual (=). É usado para atribuir o valor do lado direito à variável no lado esquerdo. Por exemplo: x = 5

Aqui, 5 é atribuído à variável x.

Exposição

Tipos de operadores de atribuição

b) **Composto: os operadores de atribuição compostos combinam um operador aritmético ou bitwise com o operador de atribuição simples.** Eles realizam uma operação entre a variável e o valor, e, depois, atribuem o resultado à variável.

Alguns exemplos incluem:

Adição: x += 1 (equivalente a x = x + 1)
Subtração: x -= 1 (equivalente a x = x - 1)
Multiplicação: x *= 2 (equivalente a x = x * 2)
Divisão: x /= 2 (equivalente a x = x / 2)
Módulo: x %= 3 (equivalente a x = x % 3)

Curiosidade

Em linguagens como JavaScript e Python, os operadores de atribuição compostos funcionam de maneira similar, simplificando a escrita de operações matemáticas comuns. Existem linguagens em que não é possível utilizar esse conceito.

Exposição

Impacto na legibilidade e na manutenção do código

a) Legibilidade

Operadores simples: são claros e diretos, mas podem tornar o código mais extenso em operações repetitivas.

Operadores compostos: tornam o código mais conciso, especialmente em operações matemáticas ou lógicas frequentes. Isso pode melhorar a legibilidade ao reduzir a quantidade de código necessário.

b) Manutenção

Código claro: o uso de operadores compostos em contextos apropriados pode tornar o código mais fácil de se entender e manter, pois as intenções do programador são expressas de maneira mais direta.

Consistência: manter um estilo consistente de operadores de atribuição em um projeto ajuda na manutenção, pois reduz a curva de aprendizado para novos desenvolvedores e facilita a revisão de código.

Em resumo, escolher o tipo certo de operador de atribuição pode ter um impacto significativo sobre a clareza, eficiência e facilidade de manutenção do código. É importante usar ambos os tipos de operadores adequadamente, dependendo do contexto e da necessidade específica.

Exposição

Uso eficiente em loops e funções

Os operadores de atribuição compostos são particularmente úteis em loops e funções, pois viabilizam a modificação de variáveis de maneira concisa e eficiente.

Por exemplo, **ao incrementar uma variável em um loop, x += 1 é mais curto e direto do que x = x + 1**. Isso não apenas economiza tempo de digitação, mas também torna o código mais legível.

Vamos fazer juntos: figurinhas

Desafio: "Ricardo e Vicente são aficionados por figurinhas. Nas horas vagas, eles arrumam um jeito de jogar um 'bafo' ou algum outro jogo que envolva tais figurinhas. **Ambos também têm o hábito de trocar as figuras repetidas com seus amigos. Certo dia, pensaram em uma brincadeira diferente.** Chamaram todos os amigos e propuseram o seguinte: com as figurinhas em mãos, cada um tentava fazer uma troca com o amigo que estivesse mais perto, de acordo com a seguinte regra – cada um contava quantas figurinhas tinha. Em seguida, eles tinham que dividir as figurinhas de cada um em pilhas do mesmo tamanho, no maior tamanho que fosse possível para ambos. Então, **cada um escolhia uma das pilhas de figurinhas do amigo para receber. Por exemplo, se Ricardo e Vicente fossem trocar as figurinhas e tivessem respectivamente 8 e 12 figuras, ambos dividiriam todas as suas figuras em pilhas de 4 figuras (Ricardo teria 2 pilhas e Vicente teria 3 pilhas) e escolheriam uma pilha do amigo para receber.**

Entrada: a primeira linha da entrada contém um único inteiro N ($1 \leq N \leq 3000$), indicando o número de casos de teste. **Cada caso de teste contém 2 inteiros F1** ($1 \leq F1 \leq 1000$) **e F2** ($1 \leq F2 \leq 1000$), indicando, respectivamente, a quantidade de figurinhas que Ricardo e Vicente têm para trocar.

Saída: para cada caso de teste de entrada, haverá um valor na saída, representando o tamanho máximo da pilha de figurinhas que poderia ser trocada entre dois jogadores."

(TONIN, [s.d.])

Vamos
fazer uma
atividade

Com base no caso apresentado, desenvolva um código de programação para a solução do problema.

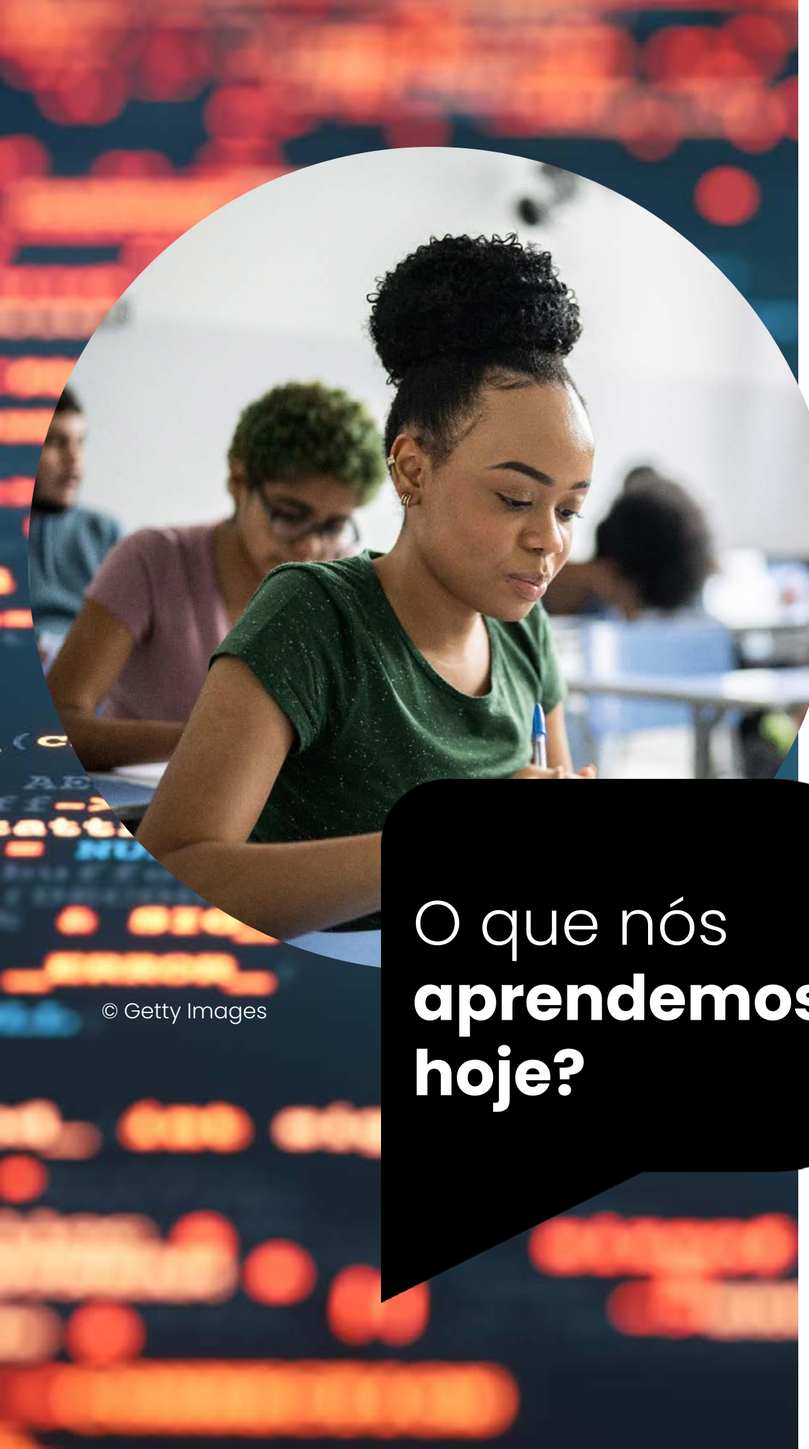
Tempo estimado: 20 minutos

Atividade pode ser realizada em grupos

Vamos fazer juntos: figurinhas

1. A atividade consiste em desenvolver esse código em Python junto ao seu grupo.

2. Considere os conceitos aprendidos hoje na aula e com os professores.



O que nós **aprendemos hoje?**

Hoje desenvolvemos:

1 O conhecimento sobre **os principais tipos de operadores de atribuição (simples e composto);**

2 A compreensão sobre a **importância do uso eficiente em loops e funções;**

3 O aprendizado na prática por meio de uma atividade de **análise de impacto na legibilidade e na manutenção do código.**

Saiba mais

Com o conceito de operadores lógicos, podemos avançar nossos estudos para o conceito de estruturas condicionais. Vamos analisar essas estruturas a partir dos conceitos que já aprendemos?

CURSO EM VÍDEO. *Estruturas condicionais 1 – curso de algoritmos #07* – Gustavo Guanabara. Disponível em: https://www.youtube.com/watch?v=_g05aHdBAEY. Acesso em: 10 abr. 2024.

Referências da aula

CURSO EM VÍDEO. *Estruturas condicionais 1 – Curso de Algoritmos #07* – Gustavo Guanabara. Disponível em: https://www.youtube.com/watch?v=_g05aHdBAEY. Acesso em: 10 abr. 2024.

DELGADO, A. L. N. *6 operadores lógicos*. CI067 – Oficina de Computação, 21 out. 2013. Disponível em: https://www.inf.ufpr.br/cursos/ci067/Docs/NotasAula/notas-6_Operadores_Logicos.html. Acesso em: 10 abr. 2024.

TONIN, N. *Figurinhas*. Beecrowd, [s.d.]. Disponível em: https://www.beecrowd.com.br/repository/UOJ_1028.html. Acesso em: 16 abr. 2024.

Identidade visual: imagens © Getty Images.

Educação Profissional Paulista

Técnico em Desenvolvimento de Sistemas

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO