**Módulo 1: Noções de Programação e Manipulação Singular de Dados**

• **Objetivos Iniciais**: O conteúdo visa definir instruções para manipulação simples de dados, distinguir a estrutura de repetição for e a estrutura condicional if, e reconhecer expressões usadas em códigos de computador.

• **Conceitos Fundamentais**: Aborda o conceito de códigos e instruções para computadores, representação digital de imagens e o desenvolvimento de habilidades de pensamento computacional como abstração, automação, reconhecimento de padrões, análise e avaliação.

• **Manipulação Singular de Dados**: Refere-se à prática de escrever códigos simples para tratar dados individualmente, sem eficiência para múltiplos dados. Envolve instruções básicas como carregar uma imagem (img = newSimpleImage("circulo.bmp")), aplicar zoom (img.setZoom(20)) e exibi-la na tela (print(img)).

• **Manipulação de Pixels Individuais**: Para manipular pixels específicos, são usadas instruções como pixel = img.getPixel(x,y) para obter a referência de um pixel e pixel.setRed(número), pixel.setGreen(número) e pixel.setBlue(número) para ajustar seus níveis de cor (valores entre 0 e 255).

• **Limitação da Manipulação Singular**: Esta abordagem é inviável para imagens grandes, pois exigiria repetir milhões de vezes as instruções de manipulação de pixels, resultando em um código difícil de ler, compreender e manter.

**Módulo 2: Estrutura de Repetição for**

• **Importância**: A estrutura de repetição for (ou "loop for") é essencial em programação para a automação de tarefas repetitivas e a manipulação eficiente de múltiplos dados, contrastando com a manipulação singular. Ela permite escrever poucas linhas de código que o computador executa repetidamente para cada dado, como os pixels de uma imagem.

• **Aplicação em Imagens**: É fundamental para processar imagens com milhares ou milhões de pixels (ex: uma imagem de 584x500 pixels tem 292.000 pixels) de forma prática e rápida.

• **Sintaxe (Exemplo em Python)**: Utiliza for i in range(image.width): e for j in range(image.height): para iterar sobre a largura e altura da imagem, processando cada pixel. Em Python, a indentação (recuo) define o bloco de instruções pertencente ao for.

• **Exemplo Prático**: Clarear uma imagem aumentando a intensidade de suas cores (vermelho, verde, azul) para cada pixel, garantindo que não ultrapassem o valor máximo de 255. Outros exemplos incluem transformar a imagem em preto removendo todas as cores ou obter canais de cores específicos (vermelho, verde, azul).

• **Benefícios**: Torna o código mais legível, fácil de manter e otimiza a execução de tarefas complexas com grandes volumes de dados.

**Módulo 3: Expressões em Código de Computador**

• **Função**: Expressões permitem manipular e processar dados de forma dinâmica, combinando valores para calcular parâmetros personalizados em vez de usar apenas valores fixos. Isso leva a resultados mais realistas e flexíveis nos programas.

• **Funcionamento**: Quando uma expressão é encontrada, o computador primeiro a avalia para obter um valor resultante e, em seguida, utiliza esse valor como parâmetro para a função correspondente.

• **Funções para Obter Valores de Pixels (JavaScript)**:

    ◦ pixel.getRed(): Obtém o valor atual do componente vermelho do pixel.

    ◦ pixel.getGreen(): Obtém o valor atual do componente verde do pixel.

    ◦ pixel.getBlue(): Obtém o valor atual do componente azul do pixel.

• **Exemplos de Uso**:

    ◦ **Duplicar/Reduzir Intensidade de Cor**: Para duplicar a intensidade de vermelho de um pixel, usa-se pixel.setRed(pixel.getRed()\*2). Para reduzir pela metade, pixel.setRed(pixel.getRed()/2) ou pixel.setRed(pixel.getRed()\*0.5).

    ◦ **Ajustar Tonalidade**: Para tornar uma imagem mais laranja, pode-se reduzir a intensidade do verde de cada pixel, por exemplo, pixel.setGreen(pixel.getGreen()\*0.7) dentro de um loop for.

    ◦ **Converter para Escala de Cinza**: Calcula-se a média dos componentes RGB de um pixel (media = (pixel.getRed()+pixel.getGreen()+pixel.getBlue())/3) e atribui-se essa média a cada um dos três componentes (vermelho, verde, azul) do pixel.

• **Importância no Pensamento Computacional**: A capacidade de traduzir um objetivo (ex: "tornar a imagem um pouco mais laranja") para um código que o computador pode executar rapidamente é uma habilidade chave do pensamento computacional.

**Módulo 4: Estrutura Condicional if**

• **Definição e Propósito**: A estrutura condicional if é fundamental para a tomada de decisões no código, permitindo que o programa execute ações diferentes com base em condições específicas (verdadeiro ou falso). É essencial para criar programas dinâmicos e adaptativos.

• **Aplicação em Imagens**: Frequentemente usada dentro de estruturas de repetição (for) para resolver problemas complexos, como a modificação seletiva de pixels em uma imagem.

• **Sintaxe (Exemplo em Python)**: A sintaxe básica é if condição: // instruções se verdadeiro else: // instruções se falso. O recuo (indentação) das instruções indica quais blocos de código pertencem ao if ou ao else.

• **Exemplos Práticos**:

    ◦ **Alteração Seletiva de Cor**: Para mudar apenas a região vermelha de uma imagem para cinza, uma condição if mais específica é necessária: if( pixel.getRed() == 255 && pixel.getGreen() == 0 && pixel.getBlue() == 0 ) garante que apenas pixels puramente vermelhos (RGB(255,0,0)) sejam selecionados para alteração de cor.

    ◦ **Meio-fio Amarelo para Cinza**: Para identificar pixels amarelos em um meio-fio e transformá-los em cinza, observa-se que tons de amarelo têm níveis equivalentes de vermelho e verde, e um nível de azul muito mais baixo, com os componentes vermelho e verde maiores do que a média dos três. Uma solução mais inteligente utiliza a média dos componentes RGB: media=( pixel.getRed()+pixel.getGreen()+pixel.getBlue() ) / 3; if( pixel.getRed() > media && pixel.getGreen() > media ). Os pixels selecionados são então ajustados para o valor da media em R, G e B para se tornarem cinza.

• **Habilidade de Análise e Avaliação**: O processo de refinar as condições do if através de tentativas e observação dos resultados na imagem é uma prática fundamental da análise e avaliação no pensamento computacional.

**Módulo 5: Pensamento Computacional e Conclusão**

• **Pensamento Computacional**: É uma habilidade crucial, comparável à leitura, escrita e aritmética, indispensável em diversas áreas.

• **Pilares do Pensamento Computacional**:

    ◦ **Abstração**: Focar nos elementos essenciais (posição e cor do pixel) para representar conceitos do mundo real numericamente, desconsiderando detalhes não fundamentais.

    ◦ **Automação**: Escrever instruções (códigos) que o computador pode seguir automaticamente para manipular esses números e ajustar imagens.

    ◦ **Análise/Avaliação**: Observar e analisar os resultados, refletir sobre melhorias e ajustar o código para alcançar os objetivos desejados.

    ◦ **Decomposição**: Decompor um problema complexo em partes menores e mais simples, que podem ser resolvidas isoladamente (ex: carregar imagem, processar pixels com for, selecionar com if, alterar cor, imprimir).

    ◦ **Detecção de Padrões**: Identificar padrões naturais ou orgânicos (ex: características de cores amareladas) para construir soluções inteligentes.

• **Conclusão**: O conteúdo abordou estruturas simples de programação, o uso de for para repetição e if para condições na manipulação de imagens digitais, consolidando habilidades essenciais do pensamento computaci

Exercícios:

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Tabela

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Site

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.