Curso de Computação 1
Introdução à Programação em Python

AULA 9

Laços aninhados e matrizes

1. Laços aninhados

Vimos nas aulas anteriores dois comandos presentes na linguagem Python que implementam estruturas de repetição: **while** e **for**. Como em toda estrutura, no corpo de um **while** ou de um **for** podemos ter um bloco de instruções qualquer, a ser repetido a cada iteração do comando, no qual podem aparecer outras estruturas. Já vimos exemplos onde onde usamos o condicional **if,** mas podemos ter também outros comandos de repetição. Quando temos um comando de repetição como um dos comandos de um bloco de comandos sendo repetido, dizemos que temos **laços aninhados**.

No vídeo a seguir mostramos um exemplo de como o uso de laços aninhados pode aparecer em nossos algoritmos:

Vídeo: Introdução aos laços aninhados

E nesse próximo vídeo, executamos a função `conta_ocorrencias_v3` no Python Tutor. Se você preferir, pode executar você mesmo. Atente para o número total de vezes que o bloco de comandos do corpo do laço mais interno é executado. Se surgir alguma dúvida, anote e pergunte ao seu professor em aula síncrona.

Vídeo: Laços aninhados no Python Tutor

Laços são ferramentas algorítmicas bastante poderosas, pois permitem que sejam especificadas sequências de ações que devem ser repetidas sem sabermos no momento da programação com quais valores eles deverão ser executadas e muitas vezes nem mesmo quantas vezes elas deverão ser repetidas. A possibilidade de executar novas repetições de ações (ações do laço interno) em cada uma das iterações de um laço mais externo, aumenta ainda mais a nossa capacidade de criar códigos que executam cálculos complexos a partir de sequências de ações relativamente simples. Veja a seguir outro exemplo de código que faz uso de laços aninhados. Vamos construir uma lista contendo a tabuada de multiplicação dos números de 1 a 9.

Vídeo: Construindo uma tabuada

Atividade: Vamos agora treinar um pouco com exercícios de fixação. Responda as perguntas da atividade “Introdução aos laços aninhados”. Fique atento ao prazo de entrega dessa atividade!

Curso de Computação 1
Introdução à Programação em Python

2. Matrizes

Usualmente, o termo ***matriz*** refere-se à organização de informações de forma tabular, em linhas e colunas. Trabalhar com informações organizadas em matrizes é algo muito útil em computação. Primeiro, porque a computação pode ser usada para resolver vários problemas matemáticos de diferentes áreas do conhecimento (engenharia, física, genética, geografia, para citar algumas) em que os dados são usualmente organizados em matrizes. Segundo, porque mesmo dados não numéricos podem receber tratamento sistemático se tabulados no formato de matriz. Nesta aula, abordaremos maneiras simples de representar matrizes em Python e de como podemos manipular dados armazenados em matrizes.

2.1 Matrizes como listas de listas

Uma das maneiras de se trabalhar com matrizes em Python é representando-as como listas de listas. Veja no vídeo a seguir como isso é feito:

Vídeo: Representação de matrizes como listas de listas

Atividade: Execute o seguinte código no Python tutor, observando como é feito o acesso e armazenamento dos elementos da matriz. O código foi disponibilizado no arquivo de códigos desta aula.

```
Mat=[ [0,0,0],[0,0,0] ]

lin=len(Mat)
col=len(Mat[0])

# Elementos da primeira linha:
Mat[0][0]=2
Mat[0][1]=-3
Mat[0][2]=4

# Elementos da segunda linha:
Mat[1][0]=0
Mat[1][1]=7
Mat[1][2]=5
```

Curso de Computação 1
Introdução à Programação em Python

2.2 Laços aninhados e matrizes

A grande vantagem do uso de matrizes é que a organização de elementos de forma tabular, em linhas e colunas, nos permite manuseá-los de forma sistemática com o uso de dois índices. A ideia é que, se todas as linhas irão receber o mesmo tratamento, podemos escrever o código para tratar os dados de uma linha e associar esse código a um comando de repetição, segundo o esquema:

```
Repete para todas as linhas:
     <bloco de código que trata os dados de uma linha>
```

E se todos os elementos de uma linha (ou seja, os elementos de cada coluna) receberão o mesmo tratamento, podemos escrever o código correspondente uma única vez, e associá-lo, por sua vez, a um outro comando de repetição, seguindo o esquema de laços aninhados que vimos no início desta aula:

```
Repete para todas as linhas:
     Repete para os elementos de cada coluna de uma linha:
          <bloco de código que trata um único elemento da matriz>
```

Estes esquemas se mostram particularmente úteis quando estamos lidando com matrizes de dados numéricos, visto que:

- Os dados são homogêneos, o que aumenta a probabilidade de que eles sejam tratados da mesma forma
- Usar um dispositivo programável como o computador para realizar cálculos matriciais diminui a probabilidade que se cometam erros de cálculo.

Matrizes de dados numéricos são comuns quando trabalhamos em áreas das ciências exatas, e por isso mesmo suas propriedades são muito estudadas. Provavelmente em outras disciplinas de seu curso você terá que lidar com matrizes, e poderá aproveitar bastante o que aprendeu neste curso de forma a contar com o computador para fazer os cálculos pra você ;-)

Veja o vídeo logo abaixo onde são mostrados exemplos sobre como podemos percorrer todos os elementos de uma matriz, fazendo uso ou não de índices para acessar cada elemento.

Vídeo: Acesso aos elementos de matrizes por laços aninhados

2.3 Criação de matrizes

Podemos criar as listas que representam uma matriz diretamente com um comando de atribuição, explicitando os elementos de cada posição da matriz. Foi assim que trabalhamos nos exemplos anteriores. Em muitas situações, porém, é necessário trabalhar com matrizes muito grandes e esparsas (que contém muitas vezes o elemento zero), ou matrizes cujos dados vão sendo fornecidos paulatinamente (e não todos de uma só vez). Nesses casos, temos que recorrer a outras estratégias para criar nossa matriz. Vamos aprender essas estratégias no vídeo a seguir. Preste bastante atenção, pois **erros no processo de criação das matrizes** que você pretende usar **comprometem todo o resultado do código!**

Vídeo: Cuidados com criação de matrizes

2.4 Usando matrizes

Façamos agora uma revisão sobre manipulação de matrizes analisando a execução de um exercício mais complexo no python tutor. O vídeo a seguir mostra como fazer a multiplicação entre duas matrizes, retornando a matriz produto. O resultado é escrito em uma nova matriz. O exemplo em questão é interessante por vários aspectos:

- Para resolver este problema, indexar sistematicamente os elementos é particularmente importante, pois os cálculos a serem feitos para cada posição da matriz de resultado envolvem mais de um elemento das matrizes de entrada: para calcular o elemento da posição i,j da matriz de resultado, usamos:
    - *todos os elementos da **linha i da matriz1** de entrada e*
    - *todos os elementos da **coluna j da matriz2** de entrada.*

  Dica: dê uma olhada em https://pt.wikipedia.org/wiki/Produto_de_matrizes para revisar o algoritmo de multiplicação de matrizes
- Precisaremos mais que dois laços aninhados para calcular a multiplicação entre matrizes. Você consegue ver por quê?
- Por fim, precisamos criar a matriz de retorno corretamente para que seja possível escrever cada um de seus elementos sem atualizar incorretamente outras linhas, como ocorre quando temos o problema do *aliasing* de listas.

O vídeo a seguir apresenta o passo a passo de uma função para multiplicação de matrizes.

Vídeo: Multiplicação de matrizes no Python Tutor

Dica: você pode rever o passo-a-passo da multiplicação de matrizes com o exemplo numérico utilizado no link: https://www.somatematica.com.br/emedio/matrizes/matrizes4.php

Atividade: Vamos agora treinar um pouco com exercícios de fixação. Responda as perguntas da atividade “Matrizes e laços aninhados”. Fique atento ao prazo de entrega dessa atividade!

Prática em Programação

Após concluir as etapas anteriores deste roteiro, faça as atividades práticas desta aula, disponíveis no Google Classroom da turma.
Até a próxima aula!