

5 Matrizes Multidimensionais

Neste capítulo você vai aprender como criar, preencher e utilizar matrizes com duas ou mais dimensões. Matrizes multidimensionais são ideais para armazenar e processar diferentes informações associadas a uma mesma entidade, facilitando a criação e visualização de combinações e correlações entre elas.

As linguagens de programação em geral permitem a criação de matrizes com duas ou mais dimensões, ou seja, estruturas cujos elementos possuem o mesmo nome mas são acessados através do uso de dois ou mais índices. Um jogo de batalha naval, o plano cartesiano e uma planilha de notas de um conjunto de alunos podem ser pensados como estruturas de dados bidimensionais; o potencial elétrico em uma região do espaço pode ser armazenado em uma estrutura tridimensional, onde cada dimensão corresponde a uma coordenada espacial.

Em JavaScript, matrizes multidimensionais são criadas em etapas. O código a seguir ilustra a criação de uma estrutura bidimensional com 3 elementos em uma dimensão e 4 na outra, a cada elemento sendo atribuído o valor $i+j$, para i variando de 0 a 2 e j variando de 0 a 3:

Listagem:

```
<script>
// primeiro bloco: cria e preenche a matriz 2d
var m2d = [];
for (var i=0;i<3;i++) {
  m2d[i] = [];
  for (var j=0;j<4;j++) {
    m2d[i][j] = i + j;
  }
}
// segundo bloco: imprime a matriz 2d
for (var i=0;i<3;i++) {
  for (var j=0;j<4;j++) {
    document.write(m2d[i][j] + " ");
  }
}
```

```
    }  
    document.write("<br>");  
  }  
</script>
```

Resultado:

```
0 1 2 3  
1 2 3 4  
2 3 4 5
```

Na primeira linha do primeiro bloco do script é criada a matriz unidimensional `m2d`, sem tamanho definido. Dentro do primeiro laço `for`, em cada i -ésimo elemento da matriz unidimensional é criada uma nova matriz unidimensional (`m2d[i] = []`) e em um novo laço `for` são atribuídos os valores $i + j$ aos elementos da matriz. Um segundo bloco é criado exclusivamente para imprimir os elementos da matriz em uma formatação apropriada. Note o uso de um espaço em branco após cada elemento e de uma quebra de linha ("`
`") após cada ciclo da variável `j`, ou seja, após cada linha.

No exemplo anterior, a criação, preenchimento e impressão dos elementos da matriz poderiam ser feitos em um único conjunto de laços `for` aninhados. Entretanto, para facilitar reaproveitamento de códigos, em geral é uma boa prática isolar as operações de entrada e saída de dados das operações lógicas e de cálculo (exceto quando a minimização do tempo de execução for uma prioridade).

Sua vez... (5-1)

Modifique o exemplo anterior de modo produza uma matriz 4×3 cujos elementos são definidos pelo produto dos índices i e j .

O exemplo a seguir mostra como utilizar matrizes para, por exemplo, calcular a média das notas de alguns alunos a partir de notas obtidas em avaliações parciais. Note, em particular, que todos os campos de cada linha são inicializados na mesma instrução e que, em JavaScript, uma matriz pode conter tipos diferentes de elementos, no caso literais (os nomes dos alunos) e números (as notas).

Listagem:

```
<script>  
  
// define matriz para nomes e notas parciais  
var a = [];  
  
// inicializa matriz com nomes e notas  
a[0] = ["Ana", 7.5, 5.0, 9.0];  
a[1] = ["Júlio", 9.0, 7.0, 8.0];  
a[2] = ["Paula", 8.0, 6.5, 7.0];  
a[3] = ["Rita", 8.5, 6.0, 8.5];
```

```

// cria e inicializa com zeros a matriz de médias
var m = [0,0,0,0];

// calcula as médias
for (var i=0;i<4;i++) {
    for (var j=1; j<=3; j++) {
        m[i] = m[i] + a[i][j];
    }
    m[i] = m[i] / 3;
}

// imprime resultados em modo pre-formatado
document.write("<pre>");

for (var i=0;i<4;i++)
    document.write(a[i][0] + "\t" + m[i].toFixed(2) + "\n");

document.write("</pre>");

</script>

```

Resultado:

```

Ana 7.17
Júlio 8.00
Paula 7.17
Rita 7.67

```

As últimas linhas do script ilustram mais uma característica do HTML: a primeira linha do último bloco chama o método `write()` do objeto `document`, passando como argumento o elemento `<pre>` do HTML; a última linha passa como argumento o elemento `</pre>`. Esses elementos demarcam o início e o fim de uma região do documento onde o texto será impresso com fonte de tamanho fixo (em que o "i" ocupa o mesmo espaço de um "m"), onde os espaços em branco adicionais não serão ignorados e onde certas combinações de caracteres, tais como a tabulação (`\t`) e a nova linha (`\n`) serão interpretados como sequências de escape. Esta estratégia é interessante quando se quer tentar tabular informações sem de fato utilizar uma tabela HTML (que gera resultados de qualidade editorial muito superior, mas que pode ser trabalhoso para programar).

Sua vez... (5-2)

Para ter mais clareza do que isto significa, experimente comentar as duas linhas que imprimem o `<pre>` e o `</pre>` e veja o resultado. Experimente também aumentar o número de espaços em branco entre as aspas, o número de tabulações e de quebras de linhas com essas linhas comentadas ou não. Para finalizar, faça uma busca na internet sobre as propriedades do elemento `pre` do HTML.

Matrizes, no sentido computacional, são obviamente bastante apropriadas para a realização de operações matriciais no sentido matemático. Cálculos com matrizes são frequentemente necessários para a solução de problemas em física e engenharia (resolução de sistemas lineares, transformações de coordenadas etc.). O script a seguir realiza o produto de

uma matriz **A** com m_A linhas e n_A colunas com uma matriz **B** com m_B linhas e n_B colunas fazendo a soma do produto de cada elemento das linhas da matriz **A** com cada elemento das colunas da matriz **B**, resultando na matriz $C = A \cdot B$:

$$c_{ij} = a_{i1} b_{1j} + a_{i2} b_{2j} + \dots + a_{ik} b_{kj} + \dots + a_{in_A} b_{n_A j}$$

Listagem:

```
<script>
// matriz A
var mA = 3; // número de linhas
var nA = 2; // número de colunas
var matA= [];
for (i=0;i<mA;i++) matA[i] = [];
matA[0][0] = 1; matA[0][1] = 0;
matA[1][0] = 1; matA[1][1] = 1;
matA[2][0] = 0; matA[2][1] = 1;

// matriz B
var mB = 2; // número de linhas
var nB = 3; // número de colunas
var matB = [];
for (i=0;i<mB;i++) matB[i] = [];
matB[0][0] = -1; matB[0][1] = 0; matB[0][2] = -1;
matB[1][0] = -3; matB[1][1] = 1; matB[1][2] = -2;

// matriz C
var mC = mA; // número de linhas
var nC = nB; // número de colunas
var matC = [];
for (i=0;i<mC;i++) matC[i] = [];
for (i=0;i<mC;i++) for (j=0;j<nC;j++) matC[i][j] = 0;

// produto das matrizes
for (i=0;i<mA;i++)
  for (j=0;j<nB;j++)
    for (k=0;k<nA;k++)
      matC[i][j] = matC[i][j] + matA[i][k]*matB[k][j];

// imprime a matriz resultante no modo preformatado
document.write("<pre>");
for (i=0;i<mA;i++) {
  for (j=0;j<nB;j++) document.write(matC[i][j] + " &nbsp;&nbsp;&nbsp; ");
  document.write("<br>");
}
document.write("</pre>");
</script>
```

Resultado:

```
-1 0 -1
-4 1 -3
-3 1 -2
```

Todos os elementos da matriz resposta **C** foram inicializados com zero em uma instrução que tem dois laços `for` aninhados (um dentro do outro). Isto é necessário porque na estrutura seguinte, que faz o produto das matrizes, os elementos da matriz serão utilizados

como acumuladores e, como tal, precisam ter um valor inicial.

No trecho do script que faz o produto das matrizes existem três laços `for` aninhados: um para percorrer as linhas da primeira matriz, outro para percorrer as colunas da segunda matriz e um terceiro que faz a soma dos produtos dos elementos das linhas pelos elementos das colunas. Note que, por conterem apenas uma instrução em cada bloco, as `{...}` (chaves) não são necessárias.

A impressão da matriz resultante é feita utilizando dois laços `for` aninhados: um para as linhas e outro para as colunas. O primeiro tem duas instruções no bloco, portanto as chaves são obrigatórias; o segundo só tem uma instrução, dispensando as chaves.

Essa implementação de um simples produto de duas matrizes é relativamente longa e intrincada. Algumas linguagens mais dedicadas a aplicações científicas e tecnológicas têm instruções nativas que simplificam enormemente esse tipo de operação. JavaScript não tem, mas é possível encontrar na internet muitas bibliotecas em código aberto que, incorporadas ao seu programa com a inclusão de apenas uma linha, fazem o mesmo serviço.

Sua vez.. (5-3)

Modifique o exemplo anterior para apresentar o produto da matriz **B** pela matriz **A**, que resulta em uma matriz 2×2 . Incremente o seu exercício fazendo com que os elementos das matrizes sejam inicializados com funções dos índices (tipo $i^2 + j^2$ ou qualquer outra que você quiser inventar).

Matrizes com mais de duas dimensões são criadas propagando o método apresentado. Para criar uma matriz de três dimensões, cria-se os elementos da primeira dimensão; a seguir cria-se neles os elementos da segunda dimensão e finalmente cria-se nesses os da terceira dimensão.

Considere, por exemplo, duas estantes, cada uma com duas prateleiras e cada prateleira com quatro livros. A localização de cada livro pode ser dada por um código do tipo "124", indicando a primeira estante, a segunda prateleira e o quarto livro. Uma maneira de criar uma matriz tridimensional para guardar os nomes dos livros associados à sua localização seria:

Listagem:

```
var nEst = 2;
var nPra = 2;
var nLiv = 4;
var Bibl = [];
for (var i=0; i<nEst; i++) {
  Bibl[i] = [];
  for (var j=0; j<nPra;j++) {
    Bibl[i][j] = [];
  }
}
```

Preencher a matriz com os nomes dos livros seria trabalhoso, mas nada difícil, utilizando agora 3 índices:

Listagem:

```
Bibl[0][0][0] = "Livro 1";   Bibl[1][0][0] = "Livro 9";
Bibl[0][0][1] = "Livro 2";   Bibl[1][0][1] = "Livro 10";
Bibl[0][0][2] = "Livro 3";   Bibl[1][0][2] = "Livro 11";
Bibl[0][0][3] = "Livro 4";   Bibl[1][0][3] = "Livro 12";
Bibl[0][1][0] = "Livro 5";   Bibl[1][1][0] = "Livro 13";
Bibl[0][1][1] = "Livro 6";   Bibl[1][1][1] = "Livro 14";
Bibl[0][1][2] = "Livro 7";   Bibl[1][1][2] = "Livro 15";
Bibl[0][1][3] = "Livro 8";   Bibl[1][1][3] = "Livro 16";
```

A recuperação da informação é feita da mesma maneira. Por exemplo, para imprimir os nomes dos livros introduzindo indicações sobre a estante e a prateleira em que se encontram:

Listagem:

```
for (var i=0; i<nEst; i++) {
  document.write("Estante " + (i+1) + "<br>");
  for (var j=0; j<nPra; j++) {
    document.write("Prateleira " + (j+1) + ":");
    for (var k=0; k<nLiv; k++) {
      document.write(" " + Bibl[i][j][k]);
    }
    document.write("<br>");
  }
  document.write("<br><br>");
}
```

Resultado:

```
Estante 1:
Prateleira 1:  Livro 1  Livro 2  Livro 3  Livro 4
Prateleira 2:  Livro 5  Livro 6  Livro 7  Livro 8

Estante 2:
Prateleira 1:  Livro 9  Livro 10  Livro 11  Livro 12
Prateleira 2:  Livro 13  Livro 14  Livro 15  Livro 16
```

Sua vez.. (5-4)

Modifique o exemplo acima de modo que, ao invés de atribuir os nomes dos livros elemento a elemento individualmente, faça-o a partir de uma lista armazenada em uma variável tipo:

```
var livros = ["Livro 1", "Livro 2", ... , "Livro N"]
```

utilizando laços `for` que controlem a sua distribuição com uma instrução do tipo:

```
Bibl[i][j][k] = livros[f(i,j,k)];
```

onde $f(i, j, k)$ é uma combinação linear dos índices da distribuição `Bibl[i][j][k]` ($a \cdot i + b \cdot j + c \cdot k$) que gera um novo índice para recuperação da informação no vetor unidimensional `livros`.

É possível construir matrizes multidimensionais mais complexas. Por exemplo, considere uma estrutura de dados construída para armazenar informações sobre partículas emergindo de uma colisão num acelerador. Pode ser interessante armazenar, para cada partícula, a massa, a carga e as três componentes do momento linear. O exemplo a seguir mostra uma possível maneira de implementar esta estrutura para 5 partículas:

Listagem:

```
var nPart = 5;
var partícula = [];

// para cada partícula...
for (var i=0; i<nPart; i++) {
    // massa, carga e vetor momento
    partícula[i] = [];
    // para o 3o. elemento, nova matriz (px,py,pz)
    partícula[i][2] = [];
}

// preenchendo a matriz:
partícula[0][0] = ...; // massa
partícula[0][1] = ...; // carga
partícula[0][2][0] = ...; // px
partícula[0][2][1] = ...; // py
partícula[0][2][2] = ...; // pz
...

```

Note que neste exemplo a matriz tem 5 linhas (que indexam as partículas) e 3 colunas (massa, carga e momento), mas somente o terceiro elemento (momento) possui profundidade. A matriz é, portanto, parcialmente bidimensional e parcialmente tridimensional.

Sua vez.. (5-5)

Modifique o exemplo acima de modo que use o método `Math.random()` para atribuir valores para a massa e a carga das partículas escolhidas aleatoriamente entre a massa do elétron e do próton (9.1×10^{-31} kg e 1.7×10^{-27} kg) e suas respectivas cargas (-1.6×10^{-19} C e $+1.6 \times 10^{-19}$ C). Para cada componente do momento, seu programa deve atribuir números aleatórios entre 0 e 1. Para imprimir os resultados, o seu programa deve utilizar uma estrutura do tipo:

```
for (var i=0; i<nPart; i++) {
    document.write(partícula[i]);
    document.write("<br>");
}
```

que deve produzir uma listagem como a que segue:

```
1.7e-27,1.6e-19,0.2,0.6,0.1
1.7e-27,1.6e-19,0.0,0.2,0.9
1.7e-27,1.6e-19,0.3,0.1,0.8
9.1e-31,-1.6e-19,0.7,0.4,0.1
9.1e-31,-1.6e-19,0.8,0.1,0.9
```

Exercícios

1. Faça um script que crie e defina uma matriz bidimensional $n \times m$ cujos elementos são definidos por $A_{i,j} = (i^2 + j^2)^{1/2}$ onde $i = 1, \dots, n$ e $j = 1, \dots, m$. Para $n = 2$ e $m = 3$, o resultado deve ser formatado como apresentado a seguir:

```
1.41  2.24  3.16
2.24  2.83  3.61
```

2. Faça um script que crie uma matriz quadrada $n \times n$ cujos elementos são definidos por $A_{i,j} = 2i + j$ onde $i, j = 1, \dots, n$, e imprima o valor da soma dos elementos da diagonal da matriz. Para $n = 3$ uma possível formatação da resposta seria:

Dada a matriz:

```
3 4 5
5 6 7
7 8 9
```

A soma dos elementos da diagonal é 18.

3. Faça um script que reproduza o seguinte quadro inicial de uma "Batalha Naval", onde os "-" significam água os "X" significam navios:

```
X - - - - -
X - - - - - X
X - X X X - - X
- - - - - - X
- - - - - -
- - - X X X - X
- - - - - - X
- X X X - - - X
```

4. Modifique o script que calcula a média dos alunos de modo que ele:
 - a. atribua pesos diferentes para as avaliações.
 - b. imprima as notas e a média dos alunos.
 - c. imprima a média de cada aluno e, em seguida, a menção "Aprovado" se a média for maior ou igual a 6.00; caso contrário, imprima "Reprovado".
5. Modifique o script do produto de matrizes para que:
 - a. faça a soma de duas matrizes.

- b. imprima e a matriz transposta de uma matriz dada.
 - c. encontre o determinante de uma matriz dada.
6. Termine de desenvolver o último exemplo do texto de modo que sejam atribuídas aleatoriamente as massas, cargas e componentes do momento das partículas. Faça o script de tal modo que existam partículas de massa $m = 1$ e $m = 2$ com igual probabilidade mas que partículas com carga $q = 1$ sejam 5 vezes mais prováveis que partículas com carga $q = 2$. O script deve atribuir componentes p_x, p_y e p_z entre 0 e 1. O resultado final deve ser formatado de modo a produzir uma listagem que se assemelhe à que segue:

```
Partícula 0: 2, 1, 0.40, 0.58, 0.68
Partícula 1: 1, 1, 0.72, 0.96, 0.26
Partícula 2: 1, 1, 0.62, 0.50, 0.48
Partícula 3: 2, 1, 0.58, 0.35, 0.37
Partícula 4: 2, 1, 0.84, 0.59, 0.36
...
```

7. Modifique o script do problema anterior de modo que as partículas continuem tendo valores aleatórios entre 0 e 1 mas que tenham momento total sempre igual à unidade ($|\mathbf{p}| = 1$).

