**Two-Dimensional array in TS**

Declaring a Two-Dimensional array:

* let mat: number[][];
* let mat: Array<number>[];
* let mat: Array<Array<number>>;
* let mat: Array<number[]>;

Initializing a Two-Dimensional array:

1) Initialize a matrix with 0 rows:

|  |
| --- |
| let mat: number[][] = [];  document.write(mat.length + "<br/>"); // (output is: 0) |
| let mat: number[][] = new Array<number[]>();  document.write(mat.length + "<br/>"); // (output is: 0) |
| let mat: number[][] = new Array<number[]>(0);  document.write(mat.length + "<br/>"); // (output is: 0) |
| let mat: number[][] = new Array<Array<number>>();  document.write(mat.length + "<br/>"); // (output is: 0) |
| let mat: number[][] = new Array<Array<number>>(0);  document.write(mat.length + "<br/>"); // (output is: 0) |

Attention:

matrix3 = []; --> OK

matrix4 = new Array<number>(); --> COMPILATION ERROR

2) Initialize a matrix with N Empty rows :)N >0)

(In the following example N=3)

|  |
| --- |
| let mat: number[][] = new Array<number[]>(3);  document.write(mat.length + "<br/>"); // (output is: 3)  //document.write(mat [0][1] + "<br/>"); --> Uncaught TypeError: Cannot read property '1' of undefined |
| let mat: number[][] = new Array<Array<number>>(3);  document.write(mat.length + "<br/>"); // (output is: 3)  //document.write(mat [0][1] + "<br/>"); --> Uncaught TypeError: Cannot read property '1' of undefined |

3) Initialize a matrix with N rows – each row is an empty

array:)N >0)

(In the following example N=3)

|  |
| --- |
| let mat: number[][] = [[], [], []];  document.write(mat.length + "<br/>"); // (output is: 3)  document.write(mat[0][1] + "<br/>"); // (output is: undefined) |
| let mat: number[][] = [new Array<number>(), new Array<number>(), new Array<number>()];  document.write(mat.length + "<br/>"); // (output is: 3)  document.write(mat[0][1] + "<br/>"); // (output is: undefined) |
| let mat: number[][] = new Array<Array<number>>([], [], []);  document.write(mat.length + "<br/>"); // (output is: 3)  document.write(mat[0][1] + "<br/>"); // (output is: undefined) |
| let mat: number[][] = new Array<Array<number>>(new Array<number>(), new Array<number>(), new Array<number>());  document.write(mat.length + "<br/>"); // (output is: 3)  document.write(mat[0][1] + "<br/>"); // (output is: undefined) |

4) Initialize a matrix with N rows – each row with values:)N >0)

(In the following example N=3)

|  |
| --- |
| let mat: number[][] = [[1, 2, 3], [4, 5, 6], [7, 8, 9]];  document.write(mat.length + "<br/>"); // (output is: 3)  document.write(mat[2][1] + "<br/>"); // (output is: 8) |
| let mat: number[][] = [new Array<number>(1, 2, 3), new Array<number>(4, 5, 6), new Array<number>(7, 8, 9)]  document.write(mat.length + "<br/>"); // (output is: 3)  document.write(mat[2][1] + "<br/>"); // (output is: 8) |
| let mat: number[][] = new Array<number[]>([1, 2, 3], [4, 5, 6], [7, 8, 9]);  document.write(mat.length + "<br/>"); // (output is: 3)  document.write(mat[2][1] + "<br/>"); // (output is: 8) |
| let mat: number[][] = new Array<Array<number>>(new Array<number>(1, 2, 3), new Array<number>(4, 5, 6), new Array<number>(7, 8, 9));  document.write(mat.length + "<br/>"); // (output is: 3)  document.write(mat[2][1] + "<br/>"); // (output is: 8) |

5) Initialize a matrix with N rows – each row with an array variable:)N >0)

(In the following example N=3)

|  |
| --- |
| let arr1: number[] = [1, 2, 3];  let arr2: number[] = [4, 5, 6];  let arr3: number[] = [7, 8, 9];  let mat: number[][] = [arr1, arr2, arr3];  document.write(mat.length + "<br/>"); // (output is: 3)  document.write(mat[2][1] + "<br/>"); // (output is: 8) |
| let arr1: number[] = [1, 2, 3];  let arr2: number[] = [4, 5, 6];  let arr3: number[] = [7, 8, 9];  let mat: number[][] = new Array<Array<number>>(arr1, arr2, arr3);  document.write(mat.length + "<br/>"); // (output is: 3)  document.write(mat[2][1] + "<br/>"); // (output is: 8) |
| let arr1: number[] = [1, 2, 3];  let arr2: number[] = [4, 5, 6];  let arr3: number[] = [7, 8, 9];  let mat: number[][] = new Array<number[]>(arr1, arr2, arr3);  document.write(mat.length + "<br/>"); // (output is: 3)  document.write(mat[2][1] + "<br/>"); // (output is: 8) |

Full code example

let mat: number[][] = [[1, 2, 3], [1, 2], [1]];

document.write("<br/>row 0: ");

document.write(mat[0][0] + " ");

document.write(mat[0][1] + " ");

document.write(mat[0][2] + " ");

document.write("<br/>row 1: ");

document.write(mat[1][0] + " ");

document.write(mat[1][1] + " ");

document.write("<br/>row 2: ");

document.write(mat[2][0] + " ");

let matrixLength: number = mat.length; //There are two rows in the matrix, so the length is 2

let firstRowLength: number = mat[0].length; // first row has 3 elements, so the length is 3

let secondRowLength: number = mat[1].length; // second row has 2 elements, so the length is 2

let thirdRowLength: number = mat[2].length; // third row has 1 elements, so the length is 1

document.write("<br/><br/>Rows in mat:"+matrixLength + "<br/>");

document.write("Elements in first row (mat[0]): " +firstRowLength + "<br/>");

document.write("Elements in second row (mat[1]): " +secondRowLength + "<br/>");

document.write("Elements in third row (mat[2]): " + thirdRowLength + "<br/>");

for (let i: number = 0; i < mat.length; i++) {

document.write("<br/>row " + i + ": ");

for (let j: number = 0; j < mat[i].length; j++) {

document.write(mat[i][j] + " ");

}

}

**The output is:**

**row 0: 1 2 3   
row 1: 1 2   
row 2: 1   
  
Rows in mat:3  
Elements in first row (mat[0]): 3  
Elements in second row (mat[1]): 2  
Elements in third row (mat[2]): 1  
  
row 0: 1 2 3   
row 1: 1 2   
row 2: 1**