Array []

ISEP / SWitCH_QA / DESOFT I

The problem with variables

Guardar uma garrafa de vinho

Guardar várias garrafas de vinho

The solution: Array

Guardar uma garrafa de vinho

Guardar várias garrafas de vinho

Array Dimensions

Variável simples

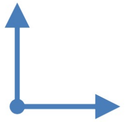
guarda um valor

Variável composta (Array)

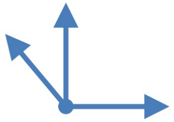
guarda vários valores do mesmo tipo

unidimensional

bidimensional



tridimensional



Array

Array Definition

- An array is a homogeneous collection of values
    - An array is a collection of values of the same data type.

- It is a user defined type
    - Be reusing existing types

Array Characteristics

- Once an array is initialized, it cannot be resized

- Each memory block represents an array element

- Array element values can be updated or modified but cannot be deleted

- REMARK
  - This information is related to simple Arrays in Java defined with [].

  - If you go looking online for examples and find the usage of Array() object, this is a different thing and we'll be discussing it later on.

One Dimension Array

Declaration and Initialisation

- //declaration
  - var array_name [:datatype];

- //initialisation
  - array_name = [val1,val2,valn..]

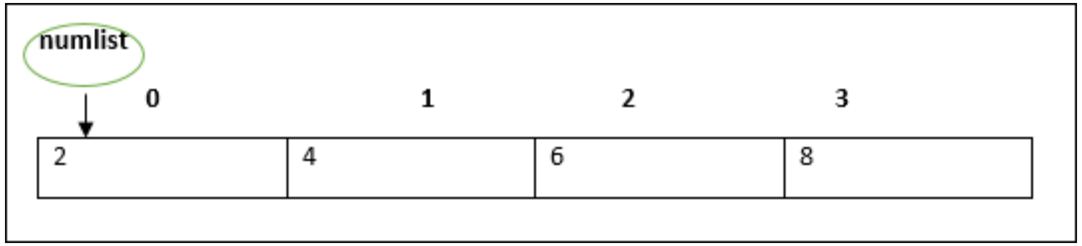
- //declaration and initialisation
  - var array_name [:data type] = [val1,val2…valn]

- Example
  - `let numlist: number[] = [2, 4, 6, 8];`

Accessing Elements

- Update an array value
  - array_name[index] = value
  - Index is the position to overwrite with value
    - Index starts at 0 position

- Retrieve an array value
  - let value [:data type] = array_name[index]



- Example

```
let numlist: number[] = [2, 4, 6, 8];
console.log(numlist);

console.log(numlist[2]);

numlist[2] = 10;
console.log(numlist[2]);

console.log(numlist);
```

- Output

```
[2, 4, 6, 8]
6
10
[2, 4, 10, 8]
```

Array Traversal

- Example

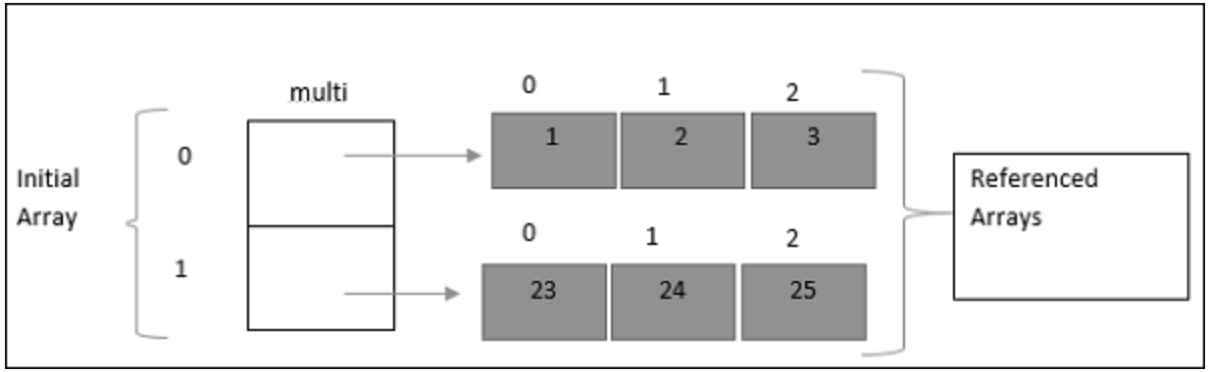
```
let j: any;
let nums: number[] = [1, 2, 3, 4]

for (j in nums) {
    console.log(nums[j])
}
```

- Output
  - 1, 2, 3, 4

Multi-Dimensional Array

Declaration and Initialisation



- //declaration
  - var array_name: datatype[][];
- //initialisation
  - array_name = [[val1,val2,val3],[v1,v2,v3]];
- //declaration and initialisation
  - let array_name :datatype[][] = [[val1,val2,val3], [v1,v2,v3]]
- Example
  - `let multi:number[][] = [[1,2,3],[23,24,25]]`

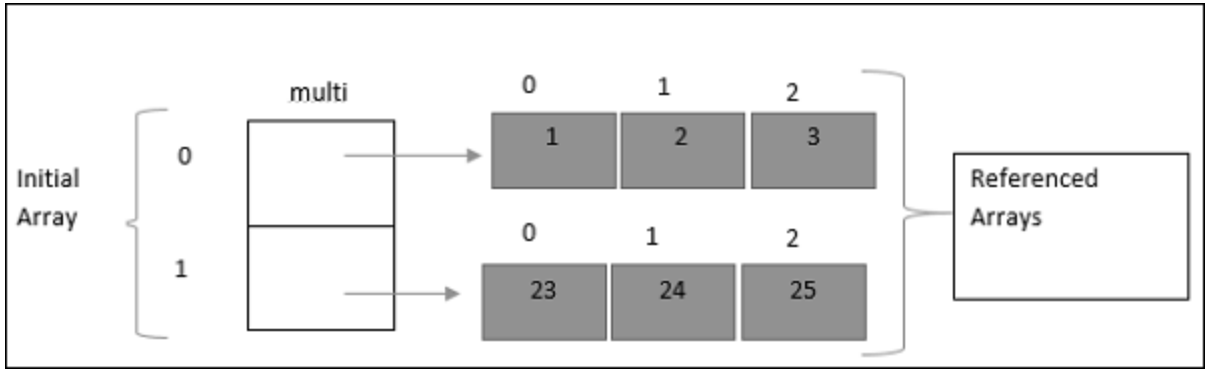
Accessing Elements

- Example (just like one-dimension array)

```
let multi:number[][] = [[1,2,3],[23,24,25]]
console.log(multi[0][0])
console.log(multi[0][1])
console.log(multi[0][2])
console.log(multi[1][0])
console.log(multi[1][1])
console.log(multi[1][2])
```

- Example (just like one-dimension array)

```
1
2
3
23
24
25
```



Array as Function Argument

- Example

```
function displayNames(array_names: string[]) {
    for (var i = 0; i < array_names.length; i++) {
        console.log(array_names[i]);
    }
}

let names: string[] = ["Mary", "Tom", "Jack", "Jill"]
displayNames(names);
```

- Output
  - Mary
  - Tom
  - Jack
  - Jill

Array as Function Return

- Example

```
function disp(): string[] {
    return ["Jill", "Tom", "Jack", "Mary"]
}

let newNames: string[] = disp();
displayNames(newNames);
```

- Output
  - Jill
  - Tom
  - Jack
  - Mary

Summary

- So far,
- We’ve understood how to use native array types on Typescript.
- You should try and apply this to your third chunk of exercises.
- On the next classes, we’ll explain how to handle objects on TypeScript.

Bibliography

- Adapted from LEI/APROG’s, LEI/ESOFT’s & SWITCH/DESOFT’s slide decks
- “Programação, algoritmos e estruturas de dados”; João Pedro Neto; Escolar Editora
- https://www.tutorialspoint.com/typescript/typescript_arrays.htm
  - Some of the images were obtained from this website