RxJs stands for Reactive Extensions for Javascript, and it's an implementation of Observables for Javascript.

Functional Reactive Programming (FRP) is a paradigm for software development that says that entire programs can be built uniquely around the notion of streams. Not only frontend programs, but any program in general.

Programs such as this can be built with very little to no application state variables, which are in general a source of errors. To make it clearer: the application does have state, but that state it's typically stored on certain streams or in the DOM, not on the application code itself.

The important is to realize that we should keep two things in mind at all times when dealing with observables:

* is the observable hot or cold?
* is the observable shared or not?

<https://www.tutorialsteacher.com/typescript/arrow-function>

Fat arrow notations are used for anonymous functions, i.e for function expressions.

They are called lambda functions in other languages.

https://www.learnrxjs.io/ for observable

TypeScript Study Notes:

TypeScript is an open-source pure object-oriented programing language. It is a strongly typed superset of JavaScript which compiles to plain JavaScript. It contains all elements of the JavaScript. It is a language designed for large-scale JavaScript application development, which can be executed on any browser, any Host, and any Operating System. The TypeScript is a language as well as a set of tools. TypeScript is the ES6 version of JavaScript with some additional features.

This process is known as **Trans-piled.**

The TypeScript compiler configuration is given in **tsconfig.json** file

Wild Web Developer for Eclipse???

1. interface Student {
2. name: string;
3. code: number;
4. }
5. let student = **<Student>** { };
6. student.name = "Rohit"; // Correct
7. student.code = 123; // Correct

**1. Using square brackets.**

1. let array\_name[:datatype] = [val1,val2,valn..]

**Example:**

1. let fruits: string[] = ['Apple', 'Orange', 'Banana'];

**2. Using a generic array type.**

1. let array\_name: Array**<elementType>** = [val1,val2,valn..]

var mArray:number[][] = [[1,2,3],[5,6,7]] ;

Array Methods

The list of array methods with their description is given below.

|  |  |  |
| --- | --- | --- |
| **SN** | **Method** | **Description** |
| 1. | concat() | It is used to joins two arrays and returns the combined result. |
| 2. | copyWithin() | It copies a sequence of an element within the array. |
| 3. | every() | It returns true if every element in the array satisfies the provided testing function. |
| 4. | fill() | It fills an array with a static value from the specified start to end index. |
| 5. | indexOf() | It returns the index of the matching element in the array, otherwise -1. |
| 6. | includes() | It is used to check whether the array contains a certain element or not. |
| 7. | Join() | It is used to joins all elements of an array into a string. |
| 8. | lastIndexOf() | It returns the last index of an element in the array. |
| 9. | Pop() | It is used to removes the last elements of the array. |
| 10. | Push() | It is used to add new elements to the array. |
| 11. | reverse() | It is used to reverse the order of an element in the array. |
| 12. | Shift() | It is used to removes and returns the first element of an array. |
| 13. | slice() | It returns the section fo an array in the new array. |
| 14. | sort() | It is used to sort the elements of an array. |
| 15. | splice() | It is used to add or remove the elements from an array. |
| 16. | toString() | It returns the string representation of an array. |
| 17. | unshift() | It is used to add one or more elements to the beginning of an array. |