Comparison of TypeScript Array Methods:

Method	Definition	Syntax	Return Type
forEach()	Executes a function once for each array element (used for iteration).	array.forEach(function(element, index, array){})	void (no return value)
map()	Creates a new array with the results of calling a function on each element.	array.map(function(element, index, array){})	Array<T> (same length)
filter()	Creates a new array with all elements that pass the test implemented by the function.	array.filter(function(element, index, array){})	Array<T> (subset)
reduce()	Executes a reducer function on each element, resulting in a single output value.	array.reduce(function(accumulator, currentValue, index, array){})	Any (based on accumulator)
some()	Tests whether **at least one** element passes the provided function.	array.some(function(element, index, array){})	boolean
every()	Tests whether **all** elements pass the provided function.	array.every(function(element, index, array){})	boolean

Note:

- **forEach()** is purely for executing side-effects (like logging), it **doesn't return anything**.
- **map()** transforms data and **returns a new array** of the same length.
- **filter()** is used when you want to keep certain elements based on a condition.
- **reduce()** is powerful and used to accumulate values (e.g., sum, average, object merging).
- **some()** returns true if **any** element matches the condition.
- **every()** returns true only if **all** elements match the condition.