Q2. Explain with examples the remaining methods of string and array.

String Methods

The following table lists the standard methods of the String object.

Method	Description
`charAt()`	Returns the character at the specified index.
`charCodeAt()`	Returns the Unicode of the character at the specified index.
`concat()`	Joins two or more strings, and returns a new string.

`endsWith()`	Checks whether a string ends with a specified substring.
`fromCharCode()`	Converts Unicode values to characters.
`includes()`	Checks whether a string contains the specified substring.
`indexOf()`	Returns the index of the first occurrence of the specified value in a string.
`lastIndexOf()`	Returns the index of the last occurrence of the specified value in a string.
`localeCompare()`	Compares two strings in the current locale.

`match()`	Matches a string against a regular expression, and returns an array of all matches.
`repeat()`	Returns a new string which contains the specified number of copies of the original string.
`replace()`	Replaces the occurrences of a string or pattern inside a string with another string, and return a new string without modifying the original string.
`search()`	Searches a string against a regular expression, and returns the index of the first match.

`slice()`	Extracts a portion of a string and returns it as a new string.
`split()`	Splits a string into an array of substrings.
`startsWith()`	Checks whether a string begins with a specified substring.
`substr()`	Extracts the part of a string between the start index and a number of characters after it.
`substring()`	Extracts the part of a string between the start and end indexes.
`toLocaleLowerCase()`	Converts a string to lowercase letters,

	according to host machine's current locale.
`toLocaleUpperCase()`	Converts a string to uppercase letters, according to host machine's current locale.
`toLowerCase()`	Converts a string to lowercase letters.
`toString()`	Returns a string representing the specified object.
`toUpperCase()`	Converts a string to uppercase letters.
`trim()`	Removes whitespace from both ends of a string.
`valueOf()`	Returns the primitive value of a String object.

Array Methods

Constructor

`Array()`

Creates a new `Array` object.

Static properties

`get Array[@@species]`

The constructor function is used to create derived objects.

Static methods

`Array.from()`

Creates a new `Array` instance from an array-like or iterable object.

`Array.isArray()`

Returns `true` if the argument is an array, or `false` otherwise.

`Array.of()`

Creates a new `Array` instance with a variable number of arguments, regardless of number or type of the arguments.

Instance properties

`Array.prototype.length`

Reflects the number of elements in an array.

`Array.prototype[@@unscopables]`

A symbol containing property names to exclude from a `with` binding scope.

Instance methods

`Array.prototype.concat()`

Returns a new array that is this array joined with other array(s) and/or value(s).

`Array.prototype.copyWithin()`

Copies a sequence of array elements within the array.

`Array.prototype.entries()`

Returns a new `Array Iterator` object that contains the key/value pairs for each index in the array.

`Array.prototype.every()`

Returns `true` if every element in this array satisfies the testing function.

`Array.prototype.fill()`

Fills all the elements of an array from a start index to an end index with a static value.

`Array.prototype.filter()`

Returns a new array containing all elements of the calling array for which the provided filtering function returns `true`.

`Array.prototype.find()`

Returns the found `element` in the array, if some element in the array satisfies the testing function, or `undefined` if not found.

`Array.prototype.findIndex()`

Returns the found index in the array, if an element in the array satisfies the testing function, or `-1` if not found.

`Array.prototype.forEach()`

Calls a function for each element in the array.

`Array.prototype.includes()`

Determines whether the array contains a value, returning `true` or `false` as appropriate.

`Array.prototype.indexOf()`

Returns the first (least) index of an element within the array equal to an elemnet, or `-1` if none is found.

`Array.prototype.join()`

Joins all elements of an array into a string.

`Array.prototype.keys()`

Returns a new `Array Iterator` that contains the keys for each index in the array.

`Array.prototype.lastIndexOf()`

Returns the last (greatest) index of an element within the array equal to an element, or `-1` if none is found.

`Array.prototype.map()`

Returns a new array containing the results of calling a function on every element in this array.

`Array.prototype.pop()`

Removes the last element from an array and returns that element.

`Array.prototype.push()`

Adds one or more elements to the end of an array, and returns the new `length` of the array.

`Array.prototype.reduce()`

Apply a function against an accumulator and each value of the array (from left-to-right) as to reduce it to a single value.

`Array.prototype.reduceRight()`

Apply a funciton against an accumulator> and each value of the array (from right-to-left) as to reduce it to a single value.

`Array.prototype.reverse()`

Reverses the order of the elements of an array *in place*. (First becomes the last, last becomes first.)

`Array.prototype.shift()`

Removes the first element from an array and returns that element.

`Array.prototype.slice()`

Extracts a section of the calling array and returns a new array.

`Array.prototype.some()`

Returns `true` if at least one element in this array satisfies the provided testing function.

`Array.prototype.sort()`

Sorts the elements of an array in place and returns the array.

`Array.prototype.splice()`

Adds and/or removes elements from an array.

`Array.prototype.toLocaleString()`

Returns a localized string representing the array and its elements. Overrides the `Object.prototype.toLocaleString()` method.

`Array.prototype.toString()`

Returns a string representing the array and its elements. Overrides the

`Object.prototype.toString()` method.

`Array.prototype.unshift()`

Adds one or more elements to the front of an array, and returns the new `length` of the array.

`Array.prototype.values()`

Returns a new `Array Iterator` object that contains the v

alues for each index in the array.

`Array.prototype[@@iterator]()`

Returns a new `Array Iterator` object that contains the values for each index in the array.