# ARRAY-SYNTAX

map( (currentValue, currentIndex, arr)=>{………})

map( callbackFn, thisValue)

map( function(currentValue, currentIndex, arr){………}, thisValue )

The same syntax of map(), filter(), forEach(), every(), some(), find(), findLast(), findIndex(), findLastIndex(), flatMap(), group(), groupToMap(),

length

at(index)

includes(searchElement, fromIndex)

indexOf(searchElement, fromIndex)

lastIndexOf(searchElement, fromIndex)

toString()

toLocaleString(locales , options)

Join(separator)

reduce( (total, currentValue, currentIndex, arr)=>{………} )

reduce( callbackFn, initialValue)

reduce( function(total, currentValue, currentIndex, arr){………}, initialValue)

The same syntax of reduce(), reduceRight()

fill(value, start, end)

copyWithin(targetIndex, start, end)

flat(depthLevelOfNestedArray)

Array.isArray(value)

keys()

values()

entries()

Array.of(element0, element1, ……… , elementN)

reverse()

sort(compareFunction)

slice(start, end)

Array.from(arrayLike, (currentValue, currentIndex)=>{………})

Array.from*(* arrayLike, mapFunction, thisValue *)*

Array.from( arrayLike, function(currentValue, currentIndex){………}, thisValue )

pop()

push(element0, element1, ……… , elementN)

shift()

unshift(element0, element1, ……… , elementN)

splice(startIndex, deletedElementCount, element0,element1, ……… , elementN)

concat(value1,value2, …, valueX) **value may be element or array**

[ …array1 , …array2 , element1 , element2 ]  **for concat**

**array comes with three-dot(…) , element comes with normally**

**Spread operator**

**Call callBackFunction in Array Manipulaion** map, filter, forEach, every, some, find, findLast, findIndex, findLastIndex, flatMap, group, groupToMap, from, reduce, reduceRight,

**Length of Array** length,

**Reduce Array on a value element in array** reduce, reduceRight

**Find index of element** indexOf, lastIndexOf, findIndex, findLastIndex

**Check element in Array** some, every, includes, indexOf, lastIndexOf

**Filter element from array and return filtered array** filter,forEach

**Map element in Array and return mapped array** map, from , forEach

**Sort element in array** sort

**Reverse an array** reverse

**Sub-Array from array** slice

**Add element in array** push, unshift, spreadOperator, splice, arr[length+n]

**Remove element in array** pop, shift, splice

**LIFO behaviour** push, pop, splice

**FIFO behaviour** push, shift, splice

**Update/Replace element of Array** arr[index], splice, fill, copyWIthIn

**Add Two Array** spreadOperator, concat

**Convert Array To String** join, toString, toLocaleString

**Find element from Index** at, arr[index]

concat() Joins arrays and returns an array with the joined arrays

constructor Returns the function that created the Array object's prototype

copyWithin() Copies array elements within the array, to and from specified positions

entries() Returns a key/value pair Array Iteration Object

every() Checks if every element in an array pass a test

fill() Fill the elements in an array with a static value

filter() Creates a new array with every element in an array that pass a test

find() Returns the value of the first element in an array that pass a test

findIndex() Returns the index of the first element in an array that pass a test

forEach() Calls a function for each array element

from() Creates an array from an object

includes() Check if an array contains the specified element

indexOf() Search the array for an element and returns its position

isArray() Checks whether an object is an array

join() Joins all elements of an array into a string

keys() Returns a Array Iteration Object, containing the keys of the original array

lastIndexOf() Search the array for an element, starting at the end, and returns its position

length Sets or returns the number of elements in an array

map() Creates a new array with the result of calling a function for each array element

pop() Removes the last element of an array, and returns that element

prototype Allows you to add properties and methods to an Array object

push() Adds new elements to the end of an array, and returns the new length

reduce() Reduce the values of an array to a single value (going left-to-right)

reduceRight() Reduce the values of an array to a single value (going right-to-left)

reverse() Reverses the order of the elements in an array

shift() Removes the first element of an array, and returns that element

slice() Selects a part of an array, and returns the new array

some() Checks if any of the elements in an array pass a test

sort() Sorts the elements of an array

splice() Adds/Removes elements from an array

toString() Converts an array to a string, and returns the result

unshift() Adds new elements to the beginning of an array, and returns the new length

valueOf() Returns the primitive value of an array

# String-SYNTAX

trim()

trimEnd()

trimRight()

trimStart()

trimLeft()

toLowerCase()

toUpperCase()

toLocaleLowerCase(locales)

toLocaleUpperCase(locales)

slice(startIndex, endIndex)

substring(startIndex, endIndex)

substr(startIndex, lengthOfSubString)

endsWith(searchString, endPosition)

startsWith(searchString, startPosition)

includes(searchString, startPosition)

Constructor :- create string object

toString() :- convert string object into string

valueOf() :- return primitive value of string object

normalize(form)

raw(strings, ...substitutions)

Create string

at(index)

charAt(index)

charCodeAt(index)

codePointAt(index)

indexOf(searchString, start)

lastIndexOf(searchString, start)

search(regexp)

match(regExp)

matchAll(regExp)

localeCompare(compareString, locales, options)

concat(string1, string2, … , stringX)

repeat(count)

Concat string

padEnd(targetLength, padString)

padStart(targetLength, padString)

***blueColorText*** = manadatory *argunment*

***blackColorText*** = optional *argunment*

replace(pattern, replacement)

replaceAll(pattern, replacement)

Replace string to another string

fromCharCode(num1, num2, numN)

From Ascii code to string

fromCodePoint(num1, num2, numN)

From UTF-16 code to string

split(separator, limit)

String to array

# EventListener-SYNTAX

**Focus Event For focus-related events**

**blur focus focusin focusout**

**Composition Event**

**compositionend compositionstart compositionupdate**

**Clipboard Event For modification of the clipboard**

**copy cut paste**

**Gesture Event**

**gesturechange gestureend gesturestart**

**Animation Event For CSS animations**

**animationcancel animationend animationiteration**

**animationstart**

**Transition Event For CSS transition**

**transitioncancel transitionend transitionrun transitionstart**

**CSS Containment Event**

**contentvisibilityautostatechanged**

**XR Session Event For change of the state of an XRSession**

**beforexrselect**

**Printing Event**

**afterprint beforeprint**

**Security Policy Violation Event**

**securitypolicyviolation**

**Hash Change Event For changes in the anchor part of the URL**

**hashchange**

**Media Event For audio video related event**

**canplay canplaythrough durationchange ended fullscreenchange fullscreenerror waiting volumechange timeupdate suspend stalled seeking seeked load loadeddata loadedmetadata loadstart pause play playing progress ratechange**

**Generic Event General event related to DOM**

**abort afterprint afterscriptexecute beforeprint**

**beforeunload beforematch beforescriptexecute**

**canplay canplaythrough change durationchange**

**error fullscreenchange fullscreenerror input**

**invalid load loadeddata loadedmetadata message**

**open online offline pause play playing progress**

**ratechange reset resize submit search stalled**

**select scroll seeked seeking show suspend**

**toggle timeupdate unload waiting**

**UI Event For user interface interaction**

**error abort beforeunload fullscreenchange fullscreenerror unload scroll select load resize message open**

**Mouse Event For mouse interaction**

**auxclick click contextmenu dblclick mousedown mouseenter mouseleave mousemove mouseout mouseover mouseup mousewheel wheel**

**Keyboard Event For keyboard interaction**

**keydown keypress keyup**

**Wheel Event For mouse wheel interaction**

**wheel mousewheel**

**Pointer Event For mouse pointer interaction**

**gotpointercapture lostpointercapture pointercancel pointerdown pointerenter**

**pointerleave pointermove pointerout pointerover**

**pointerup**

**Touch Event For touch interaction**

**touchcancel touchend touchmove touchstart**

**Form Event For user form**

**input search submit reset message**

**Submit Event For user submit**

**submit**

**Input Event For user input**

**Input , callback (for angular)**

**Message Event**

**message**

**Page Transition Event For navigating to, and away from, web pages**

**pagehide pageshow**

**Drag Event For drag and drop interaction**

**drag dragend dragenter dragleave dragover dragstart drop**

**Storage Event**

**storage**

**Browser Event**

**online offline suspend canplay loadstart progress stalled canplaythrough**

**Progress Event For the progress of loading external resources**

**error loadstart**

**Pop State Event For changes in the history entry**

**popstate**

**Pointer Event For mouse pointer interaction**

**gotpointercapture** when an element captures a pointer using setPointerCapture().

The setPointerCapture() method of the Element interface is used to designate a specific element as the capture target of future pointer events

**lostpointercapture** when a captured pointer is released.

**pointercancel** when the browser determines that there are unlikely to be any more pointer events, or if after the pointerdown event is fired.

**pointerdown** when a pointer becomes active.

**pointerenter** when a pointing device is moved into the hit test boundaries of an element or one of its descendants, including as a result of a pointerdown event from a device that does not support hover

**pointerleave** when a pointing device is moved out of the hit test boundaries of an element.

**pointermove** when a pointer changes coordinates, and the pointer has not been canceled by a browser touch-action.

**pointerout** pointing device is moved out of the hit test boundaries of an element;

**pointerover** when a pointing device is moved into an element's hit test boundaries.

**pointerup** when a pointer is no longer active.

**Submit Event For user submit**

**submit** A form is submitted

**Input Event For user input**

**input** An element gets user input

**Message Event**

**message** A message is received through the event source

**Page Transition Event For navigating to, and away from, web pages**

**pagehide** User navigates away from a webpage

**pageshow** User navigates to a webpage

**Drag Event For drag and drop interaction**

**drag** An element is being dragged

**dragend** Dragging of an element has ended

**dragenter** A dragged element enters the drop target

**dragleave** A dragged element leaves the drop target

**dragover** A dragged element is over the drop target

**dragstart** Dragging of an element has started

**drop** A dragged element is dropped on the target

**Storage Event**

**storage** A Web Storage area is updated

**UiEvent For user interface interaction**

**abort** The loading of a media is aborted

**beforeunload** Before a document is about to be unloaded

**error** An error occurs while loading an external file

**fullscreenchange** An element is displayed in fullscreen mode

**fullscreenerror** An element can not be displayed in fullscreen mode

**load** An object has loaded

**message** A message is received through the event source

**open** A connection with the event source is opened

**resize** The document view is resized

**scroll** An element's scrollbar is being scrolled

**select** User selects some text

**unload** A page has unloaded (for <body>)

**Mouse Event For mouse interaction**

**auxclick** when a non-primary pointing device button (e.g. any non-left mouse button) has been pressed and released on an element.

**click** An element is clicked on

**contextmenu** An element is right-clicked to open a context menu

**dblclick** An element is double-clicked

**mousedown** The mouse button is pressed over an element

**mouseenter** The pointer is moved onto an element

**mouseleave** The pointer is moved out of an element

**mousemove** The pointer is moved over an element

**mouseover** The pointer is moved onto an element

**mouseout** The pointer is moved out of an element

**mouseup** A user releases a mouse button over an element

**mousewheel** The mouse wheel rolls up or down over an element

**wheel** The mouse wheel rolls up or down over an element

**Keyboard Event For keyboard interaction**

**keydown** A key is down

**keypress** A key is pressed

**keyup** A key is released

**Wheel Event For mouse wheel interaction**

**wheel** The mouse wheel rolls up or down over an element

**mousewheel** The mouse wheel rolls up or down over an element

**Touch Event For touch interaction**

**touchcancel** The touch is interrupted

**touchend** A finger is removed from a touch screen

**touchmove** A finger is dragged across the screen

**touchstart** A finger is placed on a touch screen

**Form Event For user form**

**input** An element gets user input

**message** A message is received through the event source

**search** Something is written in a search field

**submit** A form is submitted

**reset** A form is reset

**Transition Event For CSS transition**

**transitioncancel** A CSS transition has canceled

**transitionend** A CSS transition has completed

**transitionrun** when a CSS transition is first created, i.e. before any transition-delay has begun.

**transitionstart** when a CSS transition has actually started, i.e., after any transition-delay has ended.

**CSS Containment Event**

**contentvisibilityautostatechanged** Any element with content-visibility: auto set on it when it starts or stops being relevant to the user and skipping its contents.

**XR Session Event For change of the state of an XRSession**

**beforexrselect** Before WebXR select events (select, selectstart, selectend) are dispatched. It can be used to suppress XR world input events while the user is interacting with a DOM overlay UI.

**Printing Event**

**afterprint** A page has started printing, or if the print dialogue box has been closed

**beforeprint** A page is about to be printed

**Security Policy Violation Event**

**securitypolicyviolation** when a Content Security Policy is violated.

**Hash Change Event For changes in the anchor part of the URL**

**hashchange** There has been changes to the anchor part of a URL

**Media Event For audio video related event**

**canplay** The browser can start playing the media (has buffered enough to begin)

**canplaythrough** The browser can play through the media without stopping for buffering

**durationchange** The duration of a media is changed

**ended** A media has reach the end ("thanks for listening")

**fullscreenchange** An element is displayed in fullscreen mode

**fullscreenerror** An element can not be displayed in fullscreen mode

**load** An object has loaded

**loadeddata** Media data is loaded

**loadedmetadata** Meta data (like dimensions and duration) are loaded

**loadstart** The browser starts looking for the specified media

**pause** A media is paused

**play** The media has been started or is no longer paused

**playing** The media is playing after having been paused or stopped for buffering.

**progress** The browser is downloading media data

**ratechange** The playing speed of the media is changed

**Browser Event**

**online** The browser starts to work online

**offline** The browser starts to work offline

**suspend** The browser is intentionally not getting media data

**canplay** The browser can start playing the media (has buffered enough to begin)

**loadstart** The browser starts looking for the specified media

**progress** The browser is downloading media data

**stalled** The browser is trying to get media data, but data is not available

**canplaythrough** The browser can play through the media without stopping for buffering

**Progress Event For the progress of loading external resources**

**error** An error occurs while loading an external file

**loadstart** The browser starts looking for the specified media

**Pop State Event For changes in the history entry**

**popstate** The window's history changes

**Focus Event For focus-related events**

**blur** An element loses focus

**focus** An element gets focus

**focusin** An element is about to get focus

**focusout** An element is about to lose focus

**Composition Event**

**compositionend** when a text composition system such as an input method editor completes or cancels the current composition session.

**compositionstart** when a text composition system such as an input method editor starts a new composition session.

**compositionupdate** when a new character is received in the context of a text composition session controlled by a text composition system

**Clipboard Event For modification of the clipboard**

**copy** The content of an element is copied

**cut** The content of an element is cutted

**paste** Some content is pasted in an element

**Gesture Event**

**gesturechange** when digits move during a touch gesture.

**gestureend** when there are no longer multiple fingers contacting the touch surface, thus ending the gesture.

**gesturestart**  when multiple fingers contact the touch surface, thus starting a new gesture.

**Animation Event For CSS animations**

**animationcancel** when a CSS Animation unexpectedly aborts

**animationend** A CSS animation has completed

**animationiteration** A CSS animation is repeated

**animationstart** A CSS animation has started

**Generic Event General event related to DOM**

**ratechange** The playing speed of the media is changed

**reset** A form is reset

**resize** The document view is resized

**submit** A form is submitted

**search** Something is written in a search field

**stalled** The browser is trying to get media data, but data is not available

**select** User selects some text

**scroll** An element's scrollbar is being scrolled

**seeked** Skipping to a media position is finished

**seeking** Skipping to a media position is started

**show** A <menu> element is shown as a context menu

**suspend** The browser is intentionally not getting media data

**toggle** The user opens or closes the <details> element

**timeupdate** The playing position has changed (like when the user fast forwards to a different point in the media)

**unload** A page has unloaded (for <body>)

**waiting** The media has paused but is expected to resume (like when the media pauses to buffer more data)

**Media Event For audio video related event**

**ratechange** The playing speed of the media is changed

**suspend** The browser is intentionally not getting media data.

**stalled** The browser is trying to get media data, but data is not available.

**seeking** Skipping to a media position is started.

**seeked** Skipping to a media position is finished.

**timeupdate** The playing position has changed (like when the user fast forwards to a different point in the media)

**volumechange** The volume of the media has changed (includes setting the volume to "mute")

**waiting** The media has paused but is expected to resume (like when the media pauses to buffer more data)

**Generic Event General event related to DOM**

**abort** The loading of a media is aborted

**afterprint** A page has started printing, or if the print dialogue box has been closed

**afterscriptexecute** after a script has been executed.

**beforeprint** A page is about to be printed

**beforeunload** Before a document is about to be unloaded

**beforematch** when it is in the hidden until found state and the browser is about to reveal its content because the user has found the content through the "find in page" feature or through fragment navigation.

**beforescriptexecute** when a script is about to be executed. Cancelling the event prevents the script from executing.

**canplay** The browser can start playing the media (has buffered enough to begin)

**canplaythrough** The browser can play through the media without stopping for buffering

**change** The content of a form element has changed

**durationchange** The duration of a media is changed

**error** An error occurs while loading an external file.

**fullscreenchange** An element is displayed in fullscreen mode.

**fullscreenerror** An element can not be displayed in fullscreen mode.

**input** An element gets user input

**invalid** An element is invalid

**load** An object has loaded

**loadeddata** Media data is loaded

**loadedmetadata** Meta data (like dimensions and duration) are loaded

**message** A message is received through the event source

**open** A connection with the event source is opened

**online** The browser starts to work online

**offline** The browser starts to work offline

**pause** A media is paused

**play** The media has been started or is no longer paused

**playing** The media is playing after having been paused or stopped for buffering

**progress** The browser is downloading media data

**ratechange** The playing speed of the media is changed

**AnimationEvent For CSS animations**

**TransitionEventFor CSS transition**

**FocusEvent For focus-related events**

**InputEvent For user input**

**KeyboardEvent For keyboard interaction**

**PageTransitionEvent For navigating to, and away from, web pages**

**PopStateEvent For changes in the history entry**

**ProgressEvent For the progress of loading external resources**

**TouchEvent For touch interaction**

**DragEvent For drag and drop interaction**

**ClipboardEvent**

**For modification of the clipboard**

**HashChangeEvent For changes in the anchor part of the URL**

**WheelEvent For mousewheel interaction**

**MouseEvent For mouse interaction**

**UiEvent For user interface interaction**

**PopStateEvent For changes in the history entry**

**MediaEvent For audio video related event**

**AnimationEvent For CSS animations**

**TransitionEventFor CSS transition**

**FocusEvent For focus-related events**

**InputEvent For user input**

**KeyboardEvent For keyboard interaction**

**PageTransitionEvent For navigating to, and away from, web pages**

**PopStateEvent For changes in the history entry**

**ProgressEvent For the progress of loading external resources**

**TouchEvent For touch interaction**

**DragEvent For drag and drop interaction**

**ClipboardEvent**

**For modification of the clipboard**

**HashChangeEvent For changes in the anchor part of the URL**

**WheelEvent For mousewheel interaction**

**MouseEvent For mouse interaction**

**UiEvent For user interface interaction**

**PopStateEvent For changes in the history entry**

**MediaEvent For audio video related event**