**JavaScript Methods**

**map()**

Do something with each item in an array and put the amended items in a new array

**filter()**

Select items from an array that meet specific conditions and return them as a new array

**reduce()**

Transform an array into a single value (not an array).

*Supply the logic to produce the final value using the reducer function and a starting value*

array.reduce (reducerFunction, startingValue)

The starting value is the first value for the calculation you want to use

The reducerFunction has the form:

reducerFunction (accumulator, currentValue)

accumulator = stores the ongoing reduced calculation output

currentValue = stores the item currently processed (updates like “i” in a loop

**Example reduce ()**

Find the factorial of [1,2,3,4,5] – FACTORIAL = 1 x 2 x 3 x 4 x 5

const numbers = [1,2,3,4,5]

const factorial = numbers.reduce ((acc, item) => acc \* item, 1 )

console.log (factorial) //120

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**some()**

Custom filtering which returns true or false

**every()**

Check if every item in an array meets a given condition

**includes()**

Look for sub-strings and array elements

***\*Won’t work on objects\****

**slice ()**

Extract a segment from an array/string

Use with indexOf()

Does not include ending index item

**splice()**

remove/change/add elements. ***Modifies the original array***

**shift()**

Chop off the first element in an array

**unshift()**

add a new element to the start of an array

**SHIFT AND UNSHIFT ARE INEFFICIENT WHEN OPERATING ON LARGE ARRAYS**

**fill()**

change some or all items in an array to a single value

**Example fill()**

const heights = [1,2,3,4,5,1,7]

heights.fill(0)

console.log(heights) // [0,0,0,0,0,0,0]

OR

const heights2 = [1,2,3,4,5,7,1,1]

heights2.fill(0,4) \*second value is index start position (fill from)

console.log(heights2) // [1,2,3,4,0,0,0,0]

**reverse()**

Reverse the order of elements in an array

**sort()**

Sort array elements alphabetically

**Entries()**

Create key : value pairs from an array using index : array/item

**find()**

get the first element of an array that passes as test (provided by a function)

**Example find()**

const ages = [3,10,18,20]

function checkage (age) {

return age > 18

}

ages.find (checkAge)

**flat()**

reduce nesting of an array

**flat() example**

arr.flat([depth]) 🡪 the depth refers to how many levels to flatten the array. Higher numbers will remove more array brackets

const numbers = [['1', '2'], ['3', '4',

['5',['6'], '7']]];

const flatNumbers= numbers.flat(Infinity);

document.write(flatNumbers); // 1,2,3,4,5,6,7