29. POLYFILLS ON HIGHER ORDER METHODS

- map, filter & reduce
- Interview Questions on Functional Programming.

Polyfills

Legacy web browsers

[1,2].map

ES6 /2017.

- Fallback Code to run if some functionality does not exist in the web browser.
- Ensuring application remain consistent.

Filter

[1,2,3,4].filter(cb)

(PASS)
True → Pushes the element to new array

False
(FAIL)
Do not push elem to new array.

Returns a new array that passes the 'test' given by 'cb'.

Q given an array, returns only even no.s.

eg arr = [1,2,3,4,5, 6,7,8]

// [2,4,6,8].

```
result
=arr.filter( function (el) {
        if (el % 2 === 0)
                return true
        else
                return false
        }
    )
```

COMPARATOR operator

(>, <, ==)

always return a boolean

if (a<b)
T , F x

result ← [2,4,6,8]

$$r = arr.filter(el \Rightarrow \underbrace{el \% 2 === 0}_{boolean})$$

CB function of filter returns a 'boolean'.

true : push to result array

boolean

false : Don't push.

.reduce

powerful.

reduces the array into single value.

```
[1,2,3,4,5].reduce((accumulator,
                    currentValue) =>
```

```
[1,2,3].reduce((acc,curr) => acc+curr, 0)
```

↑ initial value

```
// 6
```

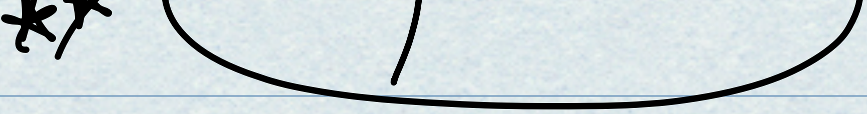
Dryrun

acc	curr	acc+curr
~~0~~	1	1
~~1~~	2	1+2=3
~~3~~	3	3+3=6
6		

Optional

```
arr.reduce(function(acc, currVal, index,
                                   arr),
           initialValue)
```

Required



FLATTEN

[1, 2, [3, 4]], [[5, 6]

→ iterate on array

If (el != arr)

3 4 push to result

5, 6

[1, 2, 3, 4], 5, 6

Else,

recursion

el = [3, 4]

[5, 6]