

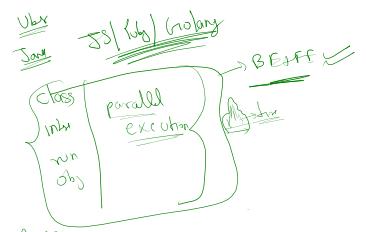
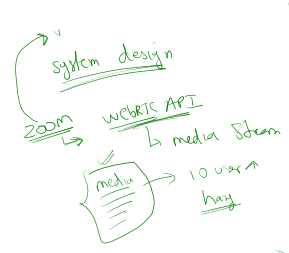
rebus
input proces → single value output
↓
var { 2, 3, arr }

code arr
access impl x
crux made
checks → is also

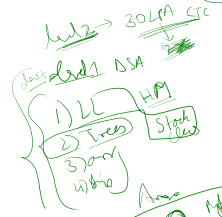
Prototypes
is an obj which contains some methods and properties
this obj is attached to arr, obj, fn, variable
as soon as they are declared/defined
Array prototype = Array.prototype
arr = [1, 2, 3]

JS obj
1) apply is a wrapper
2) let word is (function) → function
3) JS obj: simple binding or reference
4) JS obj: value is attached and method is attached
5) JS obj: spread, call, apply
6) JS obj: value is attached and method is attached
7) JS obj: value is attached and method is attached
8) JS obj: value is attached and method is attached
9) JS obj: value is attached and method is attached
10) JS obj: value is attached and method is attached
11) JS obj: value is attached and method is attached
12) JS obj: value is attached and method is attached
13) JS obj: value is attached and method is attached
14) JS obj: value is attached and method is attached
15) JS obj: value is attached and method is attached
16) JS obj: value is attached and method is attached
17) JS obj: value is attached and method is attached
18) JS obj: value is attached and method is attached
19) JS obj: value is attached and method is attached
20) JS obj: value is attached and method is attached
21) JS obj: value is attached and method is attached
22) JS obj: value is attached and method is attached
23) JS obj: value is attached and method is attached
24) JS obj: value is attached and method is attached
25) JS obj: value is attached and method is attached
26) JS obj: value is attached and method is attached
27) JS obj: value is attached and method is attached
28) JS obj: value is attached and method is attached
29) JS obj: value is attached and method is attached
30) JS obj: value is attached and method is attached
31) JS obj: value is attached and method is attached
32) JS obj: value is attached and method is attached
33) JS obj: value is attached and method is attached
34) JS obj: value is attached and method is attached
35) JS obj: value is attached and method is attached
36) JS obj: value is attached and method is attached
37) JS obj: value is attached and method is attached
38) JS obj: value is attached and method is attached
39) JS obj: value is attached and method is attached
40) JS obj: value is attached and method is attached
41) JS obj: value is attached and method is attached
42) JS obj: value is attached and method is attached
43) JS obj: value is attached and method is attached
44) JS obj: value is attached and method is attached
45) JS obj: value is attached and method is attached
46) JS obj: value is attached and method is attached
47) JS obj: value is attached and method is attached
48) JS obj: value is attached and method is attached
49) JS obj: value is attached and method is attached
50) JS obj: value is attached and method is attached
51) JS obj: value is attached and method is attached
52) JS obj: value is attached and method is attached
53) JS obj: value is attached and method is attached
54) JS obj: value is attached and method is attached
55) JS obj: value is attached and method is attached
56) JS obj: value is attached and method is attached
57) JS obj: value is attached and method is attached
58) JS obj: value is attached and method is attached
59) JS obj: value is attached and method is attached
60) JS obj: value is attached and method is attached
61) JS obj: value is attached and method is attached
62) JS obj: value is attached and method is attached
63) JS obj: value is attached and method is attached
64) JS obj: value is attached and method is attached
65) JS obj: value is attached and method is attached
66) JS obj: value is attached and method is attached
67) JS obj: value is attached and method is attached
68) JS obj: value is attached and method is attached
69) JS obj: value is attached and method is attached
70) JS obj: value is attached and method is attached
71) JS obj: value is attached and method is attached
72) JS obj: value is attached and method is attached
73) JS obj: value is attached and method is attached
74) JS obj: value is attached and method is attached
75) JS obj: value is attached and method is attached
76) JS obj: value is attached and method is attached
77) JS obj: value is attached and method is attached
78) JS obj: value is attached and method is attached
79) JS obj: value is attached and method is attached
80) JS obj: value is attached and method is attached
81) JS obj: value is attached and method is attached
82) JS obj: value is attached and method is attached
83) JS obj: value is attached and method is attached
84) JS obj: value is attached and method is attached
85) JS obj: value is attached and method is attached
86) JS obj: value is attached and method is attached
87) JS obj: value is attached and method is attached
88) JS obj: value is attached and method is attached
89) JS obj: value is attached and method is attached
90) JS obj: value is attached and method is attached
91) JS obj: value is attached and method is attached
92) JS obj: value is attached and method is attached
93) JS obj: value is attached and method is attached
94) JS obj: value is attached and method is attached
95) JS obj: value is attached and method is attached
96) JS obj: value is attached and method is attached
97) JS obj: value is attached and method is attached
98) JS obj: value is attached and method is attached
99) JS obj: value is attached and method is attached
100) JS obj: value is attached and method is attached

2nd
arr, string
objects → HashMap
Linked list → RS
arr, 2 variable
let arr, arr
1 → by 2 → by 3 → by 4



hackerearth
gfg



10 skill set
2, 3, 4 → apply
1 tech X
MNT → Ruby / Python / GoLang
D2C → unstop