

CALLBACK

```
5  ✓ let arr = [  
6      "My name is Sumeet Malik. I am a teacher. I teach programming.",  
7      "India won in Australia. England won't.",  
8      "India has gone to SA. India started well. But it rains.",  
9      "Lorem ipsum dolor. ipsum dolor lorem.",  
10 ];  
11  
12  ✓ // add a fn to all arrays which takes as input two callbacks  
13      // one for small string (< 50 in length) and the other for long strings  
14  
15      // short string callback should do the following  
16      // My name is Sumeet Malik. I am a teacher. I teach programming.  
17      // => Malik Sumeet is name My. teacher a am I. programming teach I.  
18  
19      // long string callback should to the following  
20      // I teach programming. I am a teacher. My name is Sumeet Malik.
```

```
1  Array.prototype.processStrings = function(sscb, lscb, criteria) {  
2      let oarr = this;  
3      for(let i = 0; i < oarr.length; i++){  
4          let str = oarr[i];  
5          if (str.length > criteria){  
6              lscb(str);  
7          }else{  
8              sscb(str);  
9          }  
10     }  
11     return this;  
12 }  
13  
14 let arr = [  
15     "My name is Hemakshi Pandey. I am a student. I am learning program",  
16     "India won in Australia. England won't.",  
17     "India has gone to SA. India started well. But it rains.",  
18     "Lorem ipsum dolor. ipsum dolor lorem.",  
19 ];  
20  
21 arr.processStrings(handleShortString, handleLongString, 40);  
22  
23 function handleShortString(str) {  
24     let res1 = str.split(".");  
25     let res2 = res1.filter(s => s.length > 0);  
26     let res3 = res2.map(s => s.trim());  
27     let res4 = res3.map(s => s.split(" "));  
28     let res5 = res4.map(a => a.reverse());  
29     let res6 = res5.map(a => a.join(" "));  
30     let res7 = res6.join(". ");  
31     let res8 = res7 + ".";  
32     console.log("Short string = " + res8);  
33 }  
34  
35  
36 function handleLongString(str) {  
37     let res1 = str.split(".");
```

```
25     let res2 = res1.filter(s => s.length > 0);  
26     let res3 = res2.map(s => s.trim());  
27     let res4 = res3.map(s => s.split(" "));  
28     let res5 = res4.map(a => a.reverse());  
29     let res6 = res5.map(a => a.join(" "));  
30     let res7 = res6.join(". ");  
31     let res8 = res7 + ".";  
32     console.log("Short string = " + res8);  
33 }  
34  
35
```

```
36 function handleLongString(str) {  
37     let res1 = str.split(".");  
38     let res2 = res1.filter(s => s.length > 0);  
39     let res3 = res2.map(s => s.trim());  
40     let res4 = res3.reverse();  
41     let res5 = res4.join(". ");  
42     let res6 = res5 + ".";  
43     console.log("Long string = " + res6);  
44 }  
45  
46 // add a fn to all arrays which takes as input two c  
47 // one for small string (< 50 in length) and the oth  
48  
49 // short string callback should do the following  
50 // My name is Sumeet Malik. I am a teacher. I teach  
51 // => Malik Sumeet is name My. teacher a am I. progr  
52  
53 // long string callback should to the following  
54 // I teach programming. I am a teacher. My name is S
```

→ Lecture_45 git:(main) ✖ node 1_ques_callback.js

Long string = I am learning programming. I am a student. My name is Hemakshi Pandey.

Short string = Australia in won India. won't England.

Long string = But it rains. India started well. India has gone to SA.

Short string = dolor ipsum Lorem. lorem dolor ipsum.

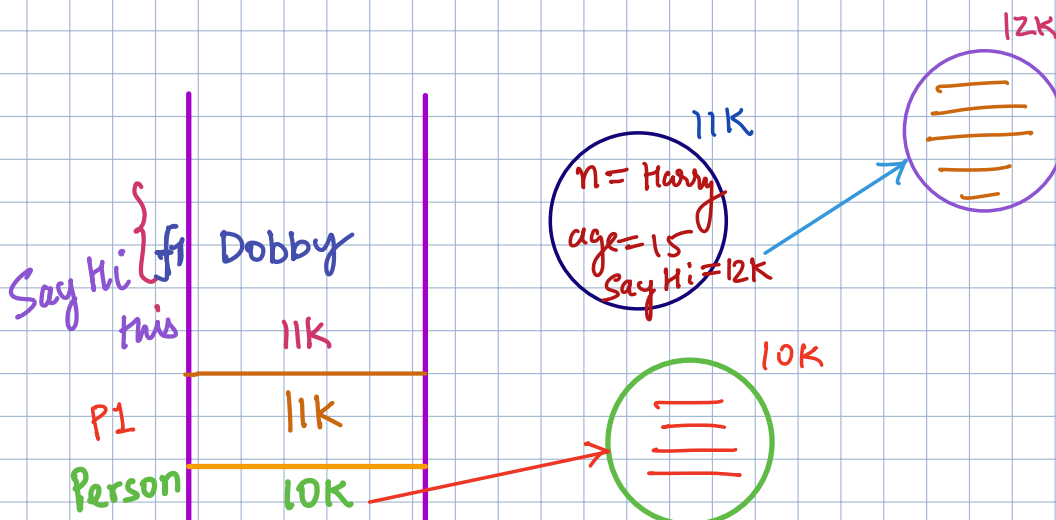
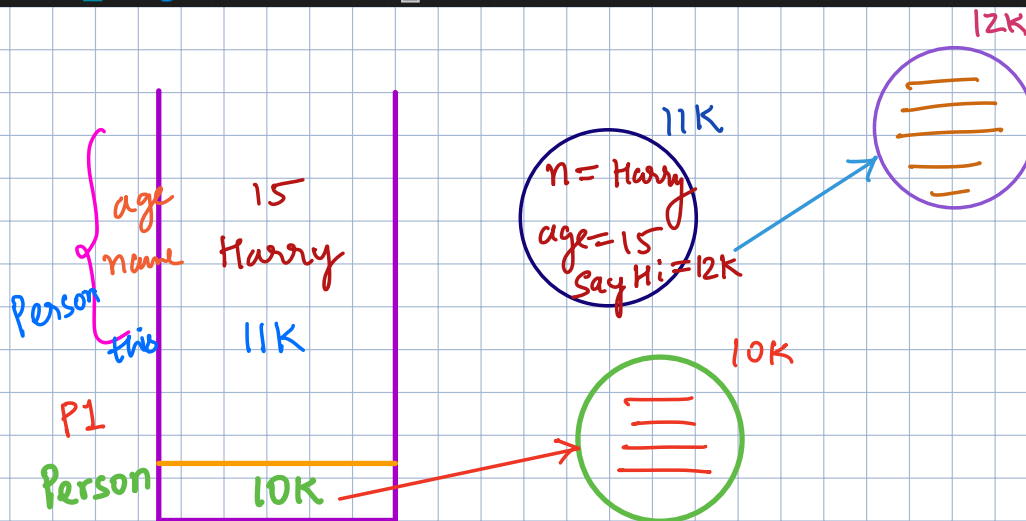
→ Lecture_45 git:(main) ✖

CONSTRUCTOR

```
2  function Person(name, age) {  
3      this.name = name;  
4      this.age = age;  
5      this.sayHi = function(f1) {  
6          console.log(this.name + "[" + this.age + "] says hi to " + f1);  
7      }  
8  }  
9  
10 let p1 = new Person("Harry", 15);  
11 p1.sayHi("Dobby");  
12
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

→ Lecture_45 git:(main) x node 2_constructor.js
Harry[15] says hi to Dobby ✓
→ Lecture_45 git:(main) x



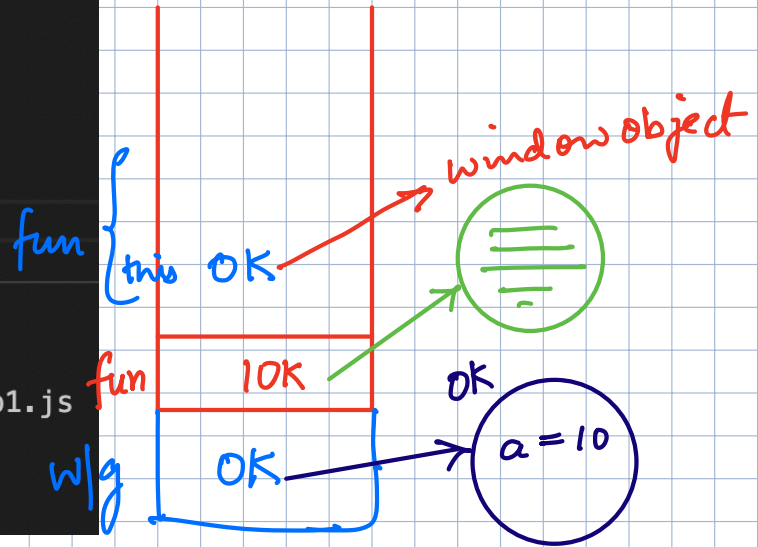
Uses of this

Part 1

```
1
2 function fun() {
3     this.a = 10;
4     console.log(a);
5 }
6
7 fun();
8 console.log(a);
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→ Lecture_45 git:(main) x node 3_Use_of_this_p1.js
10
10
→ Lecture_45 git:(main) x
```



```
1 function fun() {
2     a = 10;
3     console.log(a);
4 }
5
6 fun();
7 console.log(a);
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→ Lecture_45 git:(main) x node 4_Use_of_this_p2.js
10
10
→ Lecture_45 git:(main) x
```

- * ① Obj.fun()
- * ② pl = new Person()
- ③ fun() this points to global/window object

But we can control this using call, bind & apply

```
1 function fun() {
2     let a = 10;
3     console.log(a);
4 }
5
6 fun();
7 console.log(a);
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→ Lecture_45 git:(main) x node 5_Use_of_this.js
10
/Users/hemakshipandey/Desktop/Pepcoding-FJP1-Development/Javascript/Lecture_45/5_Use_of_this.js:7
console.log(a);
^
```