🔖

Day 4 - Loops/ Functions/ Callbacks

1. Not understanding the assignment problems.
2. Stuck on how to run test cases.
3. Learnt about Functions, Callbacks and Loops.
4. **Loops**
   Repeated Logic is used.
   To iterate over certain set of numbers.
   code:

```
let ans =0;
for (let i=1; i<=50 ;i++){
  ans = ans + i;
}
console.log(ans);
```

How does the code run?

FOR LOOP
1. let ans =0
2. let i =1 —> initialisation (we start from 1 and go to 2,3,4,5)
3. i. ≤ 50 —> Comparator Section (it is supposed to loop over a section)
4. ans = ans+i —> the Control reaches here then.
5. After the logic is run , then the control reaches to i++—> this means that you're increasing the value of i with +1.
6. This will keep happening until 1,2,3,4,5….51.

7. We have run this loop for 50 times, and it is the right way.
8. Now just change variable x of i≤x, to get the changed or desired answer.

5. Website to visualise:

http://latentflip.com/loupe/?code=JC5vbignYnV0dG9uJywgJ2NsaWNrJywgZnVuY3Rpb24gb25DbGljaygpIHsKCAgIHNldFRpbWVvdXQoZnVuY3Rpb24gdGltZXIoKSB7CiAgICAgICAgY29uc29sZS5sb2coJ1lvdSBjbGlja2VkIHRoZSBidXR0b24hJyk7ICAgIAogICAgfSwgMjAwMCk7Cn0pOwoKY29uc29sZS5sb2coIkhpISIpOwoKc2V0VGltZW91dChmdW5jdGlvbiB0aW1lb3V0KCkgewogICAgY29uc29sZS5sb2coIkNsaWNrIHRoZSBidXR0b24hIik7Cn0sIDUwMDApOwoKY29uc29sZS5sb2coIldlbGNvbWUgdG8gbG91cGUuIik7!!!PGJ1dHRvbj5DbGljayBtZSE8L2J1dHRvbj4%3D

6. **Function**
   Set of statements that perform a task or calculate a value.
   Bunch of lines of code that do some logic.

Declaration

```
function nameOfFunction(n){
  let ans = 0;
  for (let i=i ; i<n ; i++){
    ans = ans + i;
  }
return ans;
}
```

```
function sum (a+b){
  return a+b;
}
```

Calling the Function

```
let ans = nameOfFunction(100);
console.log(ans);
```

7. **Fundamental of Coding —>**
   DRY = DON’T REPEAT YOURSELF

1. Callback Function

   Writing the logic once and calling it back when needed.
   Design the logic according to your needs.

```
function square(n){
  return n * n
}

function sumOfSquares(a,b){
  const val1 = square(a);
  const val2 = square(b);

  return val1 + val2
}

console.log(sumOfSquares(1,2));
```

In this function we have made a function square and have used it inside our a different function “sumOfSqares”.

Write a simple single function and pass in 3 parameters instead of creating 3-4 functions with 2 parameters each.

```
//callback functions:

function square(n){
  return n*n;
}
function cube(n){
  return n*n*n;
}

function sumOfSomething(a,b,fn){
  const val1 = fn(a);
  const val2 = fn(b);
  return val1 + val2;
}

console.log(sumOfSomething(2,3,cube))

//hence this removes the DRY.
```

8. Anonymous Function
   copying the function without natively naming it, but just passing it.