

## Anubhav Singh –Evening Assignment

### 2.Slide 52 Explanation:

```
> var add=(function(){  
    var counter=0;  
    return function(){return counter+=1;}  
})();  
add();  
add();  
add();  
3  
>
```

Here, counter although a local variable, due to closure property acts as a global variable and the variable “add” acts as a function. So the first execution sets counter to 0 after which counter variable is protected by the scope of anonymous function. Hence the value of counter keeps on incrementing rather than setting back to 0.

3. Create a function which takes input as a string and returns true if

a) String starts with lion

b) String ends with cat

c) String has “abc” (b can be  $n \geq 1$  times) anywhere in between the string. And also print the location of a/b/c if true or else return false.

```

> function checkstr(str)
{
  var a=/^lion/i;
  var b=/cat$/i;
  var c=/ab+c/i;
  if(a.test(str)){
    return true+"\n"+a.exec(str).index;}
  else if(b.test(str)){
    return true+"\n"+b.exec(str).index;}
  else if(c.test(str)){
    return true+"\n"+c.exec(str).index;}
  else return false;}
< undefined

> checkstr("Lion is the king of jungle");
< "true
0"

> checkstr("Rahul is having a cat");
< "true
18"

> checkstr("Hello Mr abbbccc");
< "true
9"

> checkstr("Welcome to Accolite");
< false
>

```

4. Create a function which takes array as an input and

a) Sort array in ascending order

b) Multiply each number by 10

c) Return those numbers which are divisible by 3

```

> arr=[]
function ArrayManipulation(arr)
{
  var brr=arr.sort();
  var crr=brr.map((item)=>{return item*10;});
  var drr=crr.filter((item)=>{return item%3===0;});
  return drr;
};
< ▶ []

> ArrayManipulation([20,10,40,60,50,90,30])
< ▶ (3) [300, 600, 900]
> |

```

First the sorted elements are stored in brr array, then the numbers are multiplied and stored in crr array and then only

those elements whose value is multiple of 3 are displayed as result.

5. Difference between == and === with a small example.

The basic difference between a “==” and “===” is that “==” will do the necessary type conversions and then will check for equality and then return a Boolean value whereas “===” won’t do type conversion before checking for equality.

Hence,

5=="5" will return true

Whereas

5==="5" will return false.