Assignment - OOJS Javascript

Q1. Create a hierarchy of person, employee and developers.

Q2. Given an array, say [1,2,3,4,5]. Print each element of an array after 3 secs.

Q3. Explain difference between Bind and Call (example).

Q4. Explain 3 properties of argument object.

Q5. Create a function which returns number of invocations and number of instances of a function.

Q6. Create a counter using closures.

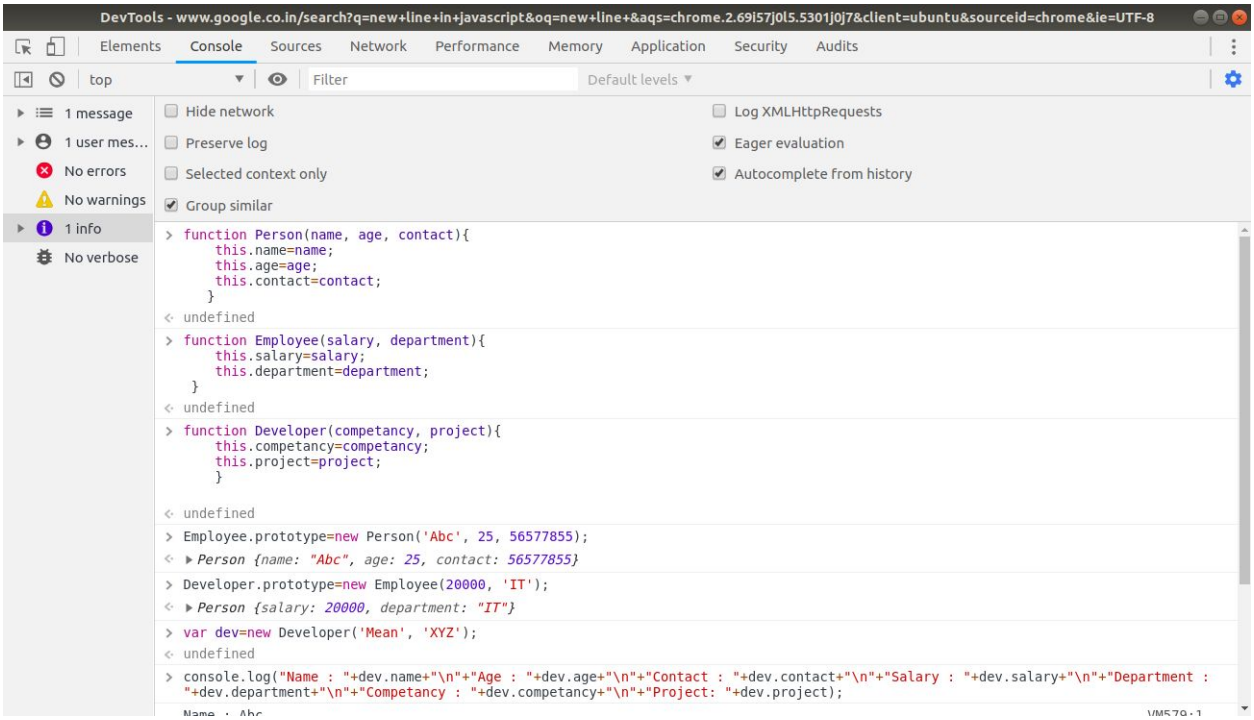
1. Code:

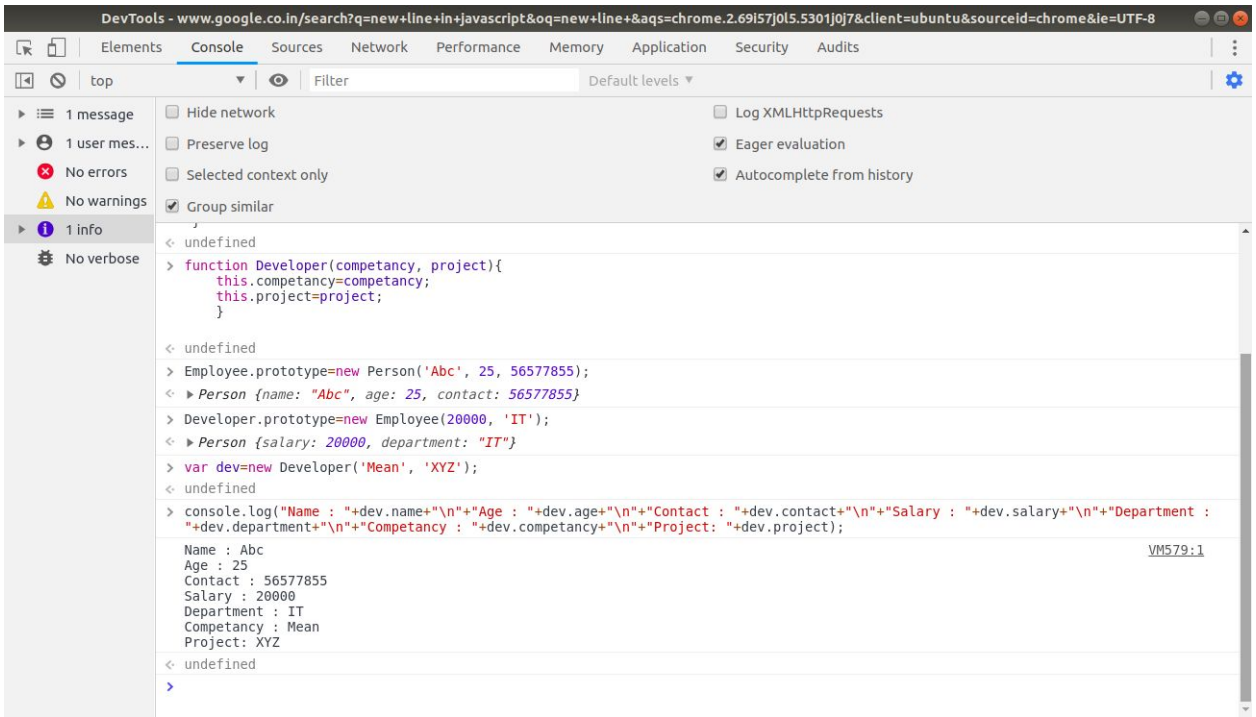
```
function Person(name, age, contact){
        this.name=name;
   this.age=age;
   this.contact=contact;
  }
function Employee(salary, department){
        this.salary=salary;
   this.department=department;
 }
function Developer(competancy, project){
        this.competancy=competancy;
   this.project=project;
   }

Employee.prototype=new Person('Abc', 25, 56577855);
Developer.prototype=new Employee(20000, 'IT');

var dev=new Developer('Mean', 'XYZ');

console.log("Name : "+dev.name+"\n"+"Age : "+dev.age+"\n"+"Contact :
"+dev.contact+"\n"+"Salary : "+dev.salary+"\n"+"Department :
"+dev.department+"\n"+"Competancy : "+dev.competancy+"\n"+"Project: "+dev.project);
```

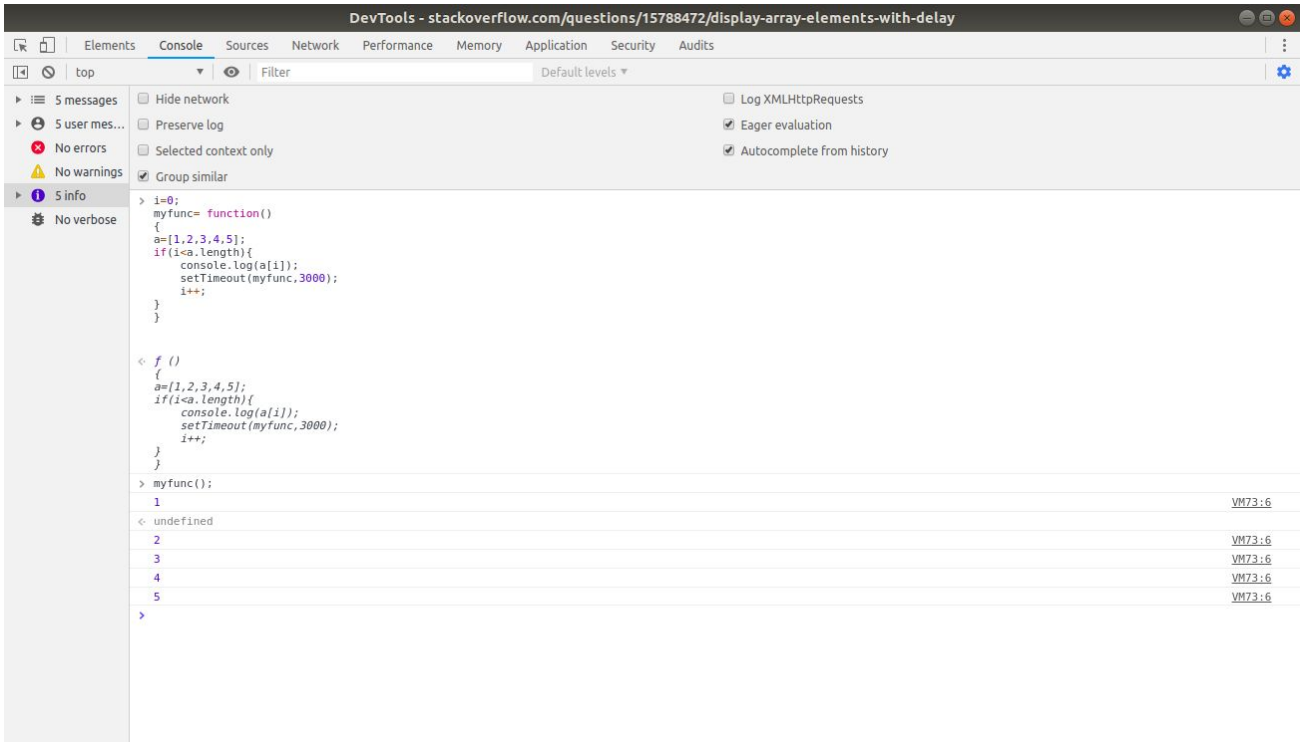

DevTools - www.google.co.in/search?q=new+line+in+javascript&oq=new+line+&aqs=chrome.2.69i57j0l5.5301j0j7&client=ubuntu&sourceid=chrome&ie=UTF-8
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Audits
top
Filter
Default levels
1 message
1 user mes...
No errors
No warnings
1 info
No verbose
Hide network
Preserve log
Selected context only
Group similar
Log XMLHttpRequests
Eager evaluation
Autocomplete from history
function Person(name, age, contact){
this.name=name;
this.age=age;
this.contact=contact;
}
undefined
function Employee(salary, department){
this.salary=salary;
this.department=department;
}
undefined
function Developer(competancy, project){
this.competancy=competancy;
this.project=project;
}
undefined
Employee.prototype=new Person('Abc', 25, 56577855);
Person {name: "Abc", age: 25, contact: 56577855}
Developer.prototype=new Employee(20000, 'IT');
Person {salary: 20000, department: "IT"}
var dev=new Developer('Mean', 'XYZ');
undefined
console.log("Name : "+dev.name+"\n"+"Age : "+dev.age+"\n"+"Contact : "+dev.contact+"\n"+"Salary : "+dev.salary+"\n"+"Department : "+dev.department+"\n"+"Competancy : "+dev.competancy+"\n"+"Project: "+dev.project);


DevTools - www.google.co.in/search?q=new+line+in+javascript&oq=new+line+&aqs=chrome.2.69i57j0l5.5301j0j7&client=ubuntu&sourceid=chrome&ie=UTF-8
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Audits
top
Filter
Default levels
1 message
1 user mes...
No errors
No warnings
1 info
No verbose
Hide network
Preserve log
Selected context only
Group similar
Log XMLHttpRequests
Eager evaluation
Autocomplete from history
undefined
function Developer(competancy, project){
this.competancy=competancy;
this.project=project;
}
undefined
Employee.prototype=new Person('Abc', 25, 56577855);
Person {name: "Abc", age: 25, contact: 56577855}
Developer.prototype=new Employee(20000, 'IT');
Person {salary: 20000, department: "IT"}
var dev=new Developer('Mean', 'XYZ');
undefined
console.log("Name : "+dev.name+"\n"+"Age : "+dev.age+"\n"+"Contact : "+dev.contact+"\n"+"Salary : "+dev.salary+"\n"+"Department : "+dev.department+"\n"+"Competancy : "+dev.competancy+"\n"+"Project: "+dev.project);
Name : Abc
Age : 25
Contact : 56577855
Salary : 20000
Department : IT
Competancy : Mean
Project: XYZ
VM579:1
undefined

2. Code:

```
i=0;
myfunc= function()
{
a=[1,2,3,4,5];
if(i<a.length){
        console.log(a[i]);
        setTimeout(myfunc,3000);
        i++;
}
}

myfunc();
```



3. Bind and Call

Ans: Bind

(i) It is used to fix the value of this.
(ii) It returns a copy of function with different context.
(iii) It creates the function.
(iv) It is used when we have to call the function later in some different context.

Example:

```
let obj={ temp: 3};
let add=function(a,b){
        result= this.temp+a+b;
        return result;
};

var p=add.bind(obj,2,2);
p; // returns a function
p(); // 7
```

Call

(i) Used to invoke the method and provides new value of this to method.
(ii) It takes give this argument and other required individual arguments;
(iii) Call is basically used when function needs to be called immediately.

Example:

```
let obj={ temp: 3};
let add=function(a,b){
        result= this.temp+a+b;
        return result;
};

var p=add.bind(obj,2,2);
p; // 7
```

4. Explain 3 properties of argument object.

(i) Callee :- It is basically used in case of anonymous function. It is used to refer the currently executing function in the function body.

Example-

```
var sum= function(x){
   if(x<1) return 0;
   return x+ arguments.callee(x-1);
}
sum(5); // 15
```

(ii) Length :- It basically returns the number of arguments passed to the function.

Example:-

```
function sum(x1,x2){
   var total= x1+x2;
         console.log(total);            // 7
         var arg=arguments.length;
         console.log(arg);              // 2
}
sum(2,5);
```

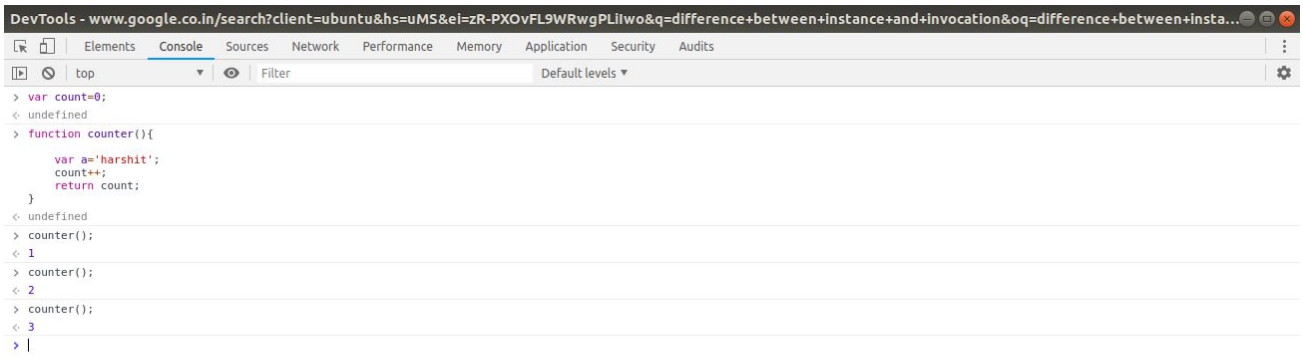
(iii) Caller:- It is used to provide the name of the function that invoked the current function being executed.

5. Create a function which returns number of invocations and number of instances of a function.

Code:

```
var count=0;
function counter(){

    var a='harshit';
    count++;
    return count;
}
counter();
counter();
counter();
```



6. Create a counter using closures.

Code:

```
function demo(){
    var count=0;
    function mycounter(n){
        while(count<n){
            console.log(count);
                        count++;
        }
    }
mycounter(5);
}

demo();
```