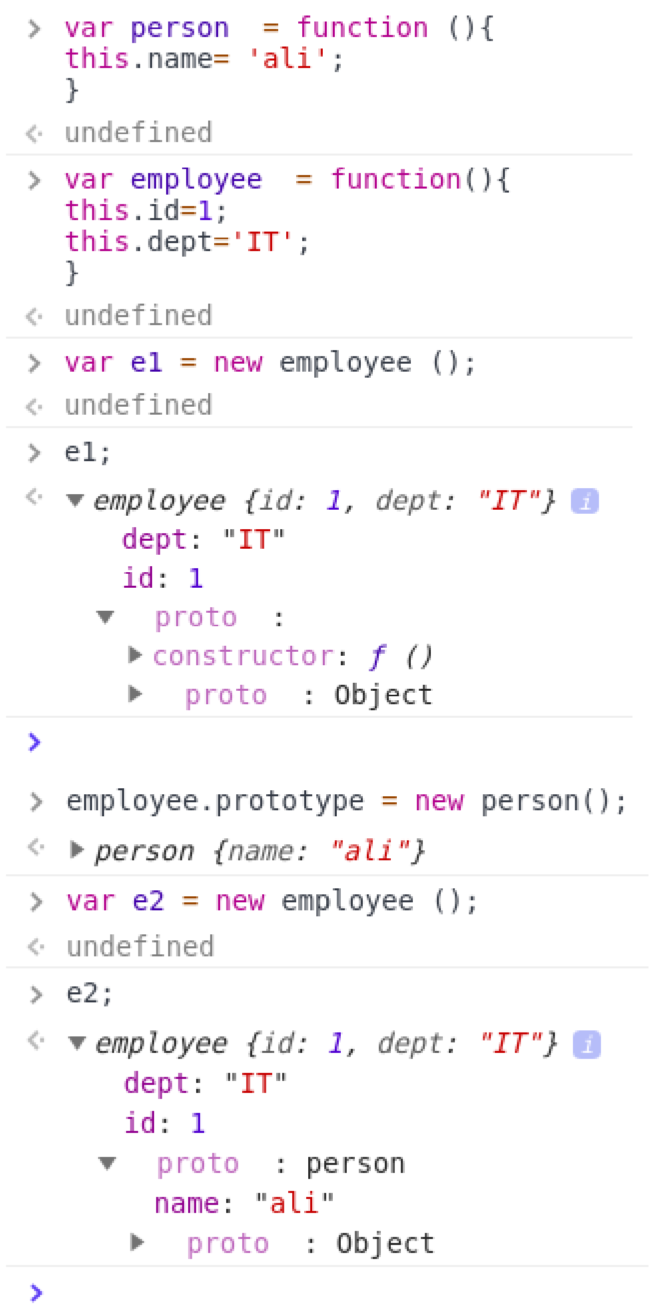
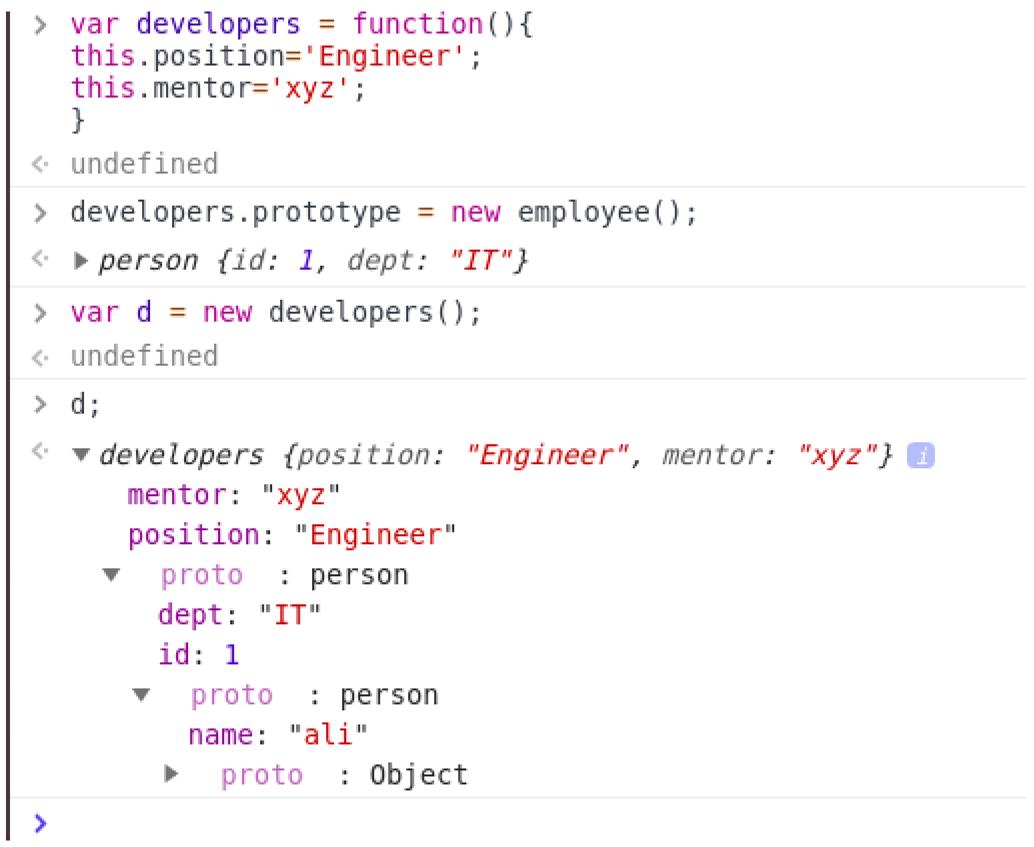
OOJS Javascript

Q-.Create a hierarchy of person, employee and developers.



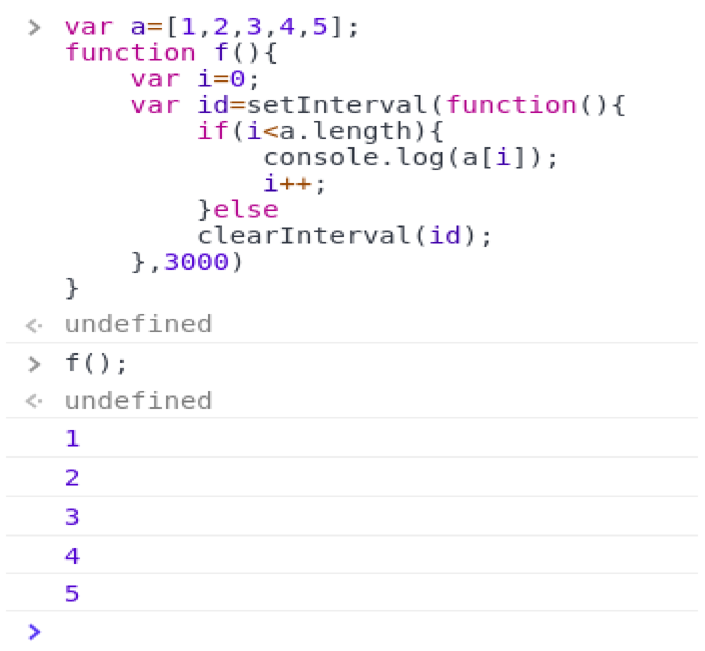
```
> var person  = function (){
  this.name= 'ali';
  }
<· undefined
> var employee  = function(){
  this.id=1;
  this.dept='IT';
  }
<· undefined
> var e1 = new employee ();
<· undefined
> e1;
<· ▼employee {id: 1, dept: "IT"} i
      dept: "IT"
      id: 1
    ▼  proto  :
     ▶constructor: ƒ ()
     ▶   proto  : Object
>

> employee.prototype = new person();
<· ▶person {name: "ali"}
> var e2 = new employee ();
<· undefined
> e2;
<· ▼employee {id: 1, dept: "IT"} i
      dept: "IT"
      id: 1
    ▼  proto  : person
      name: "ali"
     ▶   proto  : Object
>
```



```
> var developers = function(){
  this.position='Engineer';
  this.mentor='xyz';
  }
<· undefined
> developers.prototype = new employee();
<· ▶ person {id: 1, dept: "IT"}
> var d = new developers();
<· undefined
> d;
<· ▼ developers {position: "Engineer", mentor: "xyz"} i
      mentor: "xyz"
      position: "Engineer"
    ▼  proto  : person
        dept: "IT"
        id: 1
      ▼  proto  : person
          name: "ali"
        ▶  proto  : Object
>
```

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

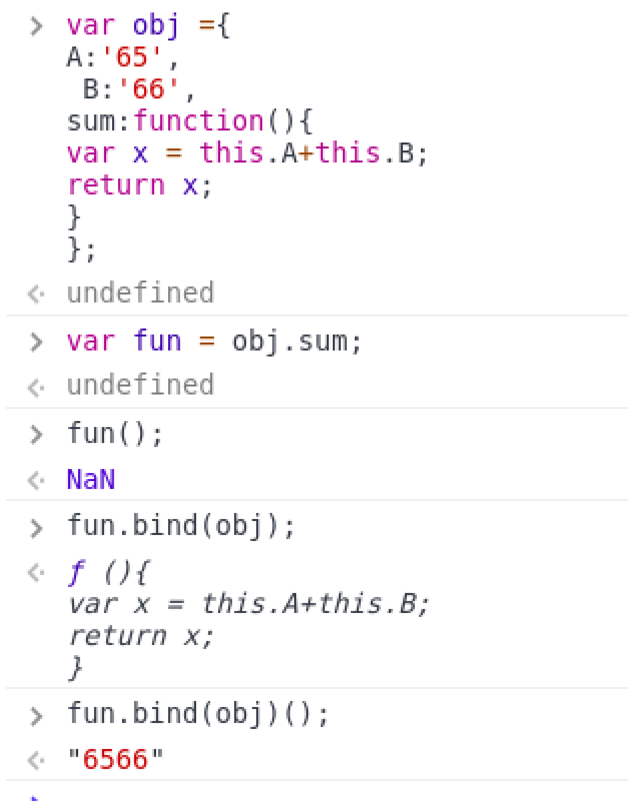


```
> var a=[1,2,3,4,5];
  function f(){
      var i=0;
      var id=setInterval(function(){
          if(i<a.length){
              console.log(a[i]);
              i++;
          }else
          clearInterval(id);
      },3000)
  }
<· undefined
> f();
<· undefined
  1
  2
  3
  4
  5
>
```

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

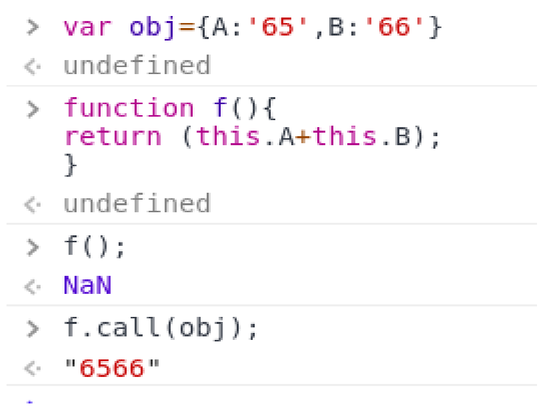
Whenever we want to to execute a function which is not a method of an object(not included in the constructor function),these functions make possible to use external function to particular object instances.

Bind() method invoke a function with object as an argument and return function definition.



```
> var obj ={
  A:'65',
   B:'66',
  sum:function(){
  var x = this.A+this.B;
  return x;
  }
  };
< undefined
> var fun = obj.sum;
< undefined
> fun();
< NaN
> fun.bind(obj);
< ƒ (){
  var x = this.A+this.B;
  return x;
  }
> fun.bind(obj)();
< "6566"
```

Call:-



```
> var obj={A:'65',B:'66'}
< undefined
> function f(){
  return (this.A+this.B);
  }
< undefined
> f();
< NaN
> f.call(obj);
< "6566"
```

:

Q. Explain 3 properties of argument object.

Argument are Objects which are accessible inside functions which contains parameters passed to the function.

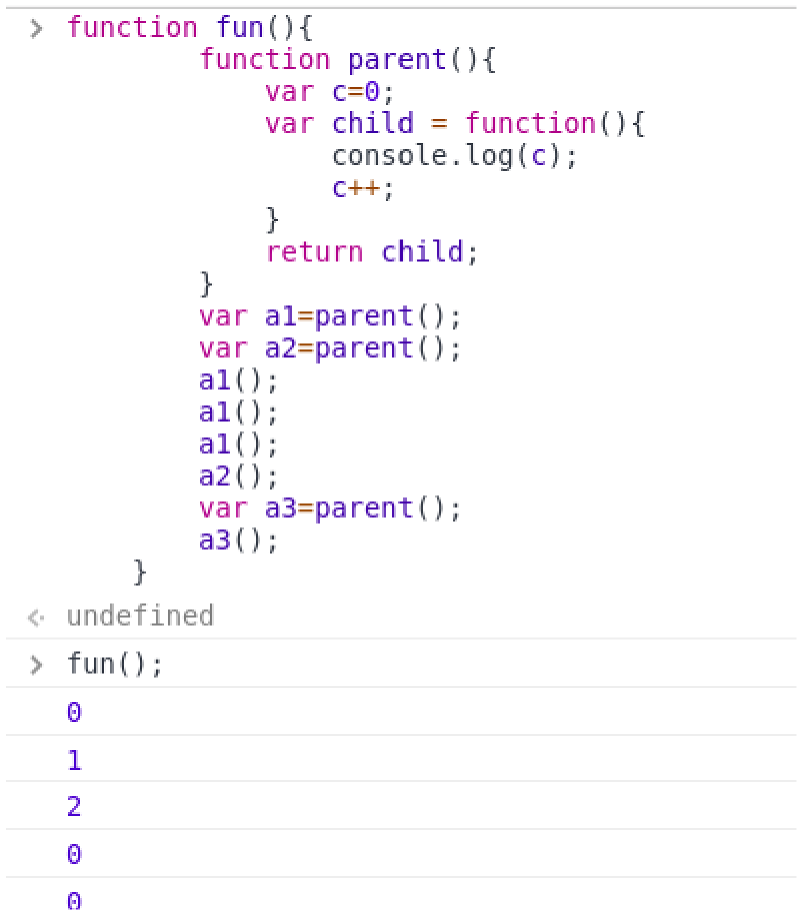
Argument is array-like object.It does not support Array properties except length.

For Arrow functions arguments are not defined.

The argument.callee property contains the currently executing function

Argument.caller refers to the function which currently called the executing function.

Q-Counter using closures.



```
> function fun(){
          function parent(){
              var c=0;
              var child = function(){
                  console.log(c);
                  c++;
              }
              return child;
          }
          var a1=parent();
          var a2=parent();
          a1();
          a1();
          a1();
          a2();
          var a3=parent();
          a3();
      }
<· undefined
> fun();
  0
  1
  2
  0
  0
```