1.var ary = [1, 2, 3, 4];  
function ff(ary) {  
 ary[0] = 0;  
 ary = [0];  
 ary[0] = 100;  
 return ary  
}  
var s = ff(ary);  
console.log(ary);\_\_\_\_\_\_\_\_\_\_\_\_  
console.log(s);\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
2.var a = 10, b = 11, c = 12;  
function test(a) {  
 a = 1;  
 var b = 2;  
 c = 3;  
}  
test(10);  
console.log(a);\_\_\_\_\_\_\_\_\_\_\_\_  
console.log(b);\_\_\_\_\_\_\_\_\_\_\_\_  
console.log(c);\_\_\_\_\_\_\_\_\_\_\_\_

3.console.log(f1);\_\_\_\_\_\_\_\_\_\_  
if("a1" in window){  
 f1();  
 var a1=1;  
 function f1() {  
 a1++;  
 console.log(a1);\_\_\_\_\_\_  
 };  
}  
4.  
var n=10;  
var obj1={  
 n:1,  
 f:function () {   
 this.n++;  
 n=this.n++;  
 }  
};  
obj1.f();  
console.log(n);\_\_\_\_\_\_\_\_\_\_\_\_\_  
console.log(obj1.n);\_\_\_\_\_\_\_\_\_\_\_\_

5.

var aa=20;  
function bind() {

console.log(this);\_\_\_\_\_\_\_  
 var aa=12;  
 function fn() {  
 console.log(this.aa);\_\_\_\_\_  
 }  
 box.onclick=function () {  
 console.log(this);\_\_\_\_\_  
 fn();  
 };  
}  
bind()

6.

var a=0;  
function FF() {  
 this.a=1;  
 a++;  
 return a;  
}  
var f1=FF();  
console.log(f1,a);\_\_\_\_\_\_\_\_\_\_\_  
var f2=new FF();  
console.log(f2);\_\_\_\_\_\_\_\_\_  
console.log(a);\_\_\_\_\_\_\_\_\_\_

7

function FF() {  
 this.x=1;  
}  
FF.prototype.getX=function () {};  
var f1=new FF;  
var f2=new FF;  
console.log(f1.\_\_proto\_\_ == FF.prototype);  
  
console.log(FF.prototype.\_\_proto\_\_==Object.prototype);  
  
console.log(FF.prototype.constructor==FF);  
  
console.log(Object.prototype.\_\_proto\_\_);  
  
console.log(f1.getX==f2.getX);

1. 数组中的方法实现
2. 循环绑定事件,弹出对应的索引

Var oLis=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

10.原型图