1.

function C1(name) {

if (name) {

this.name = name;

}

;

}

function C2(name) {

this.name = name;

}

function C3(name) {

this.name = name || 'join';

}

C1.prototype.name = 'Tom';

C2.prototype.name = 'Tom';

C3.prototype.name = 'Tom';

console.log(new C1().name)

console.log(new C2().name)

console.log(new C3().name)

------------------------------------------------------

2.

function fun() {

this.a = 0;

this.b = function () {

alert(this.a);

}

}

fun.prototype = {

b: function () {

this.a = 20;

alert(this.a);

},

c: function () {

this.a = 30;

alert(this.a)

}

}

var my\_fun = new fun();

my\_fun.b();

my\_fun.c();

-----------------------------------------------------

3.

function Foo() {

getName = function () {

console.log(1);

};

return this;

}

Foo.getName = function () {

console.log(2);

};

Foo.prototype.getName = function () {

console.log(3);

};

var getName = function () {

console.log(4);

};

function getName() {

console.log(5);

}

Foo.getName();

getName();

Foo().getName();

getName();

new Foo.getName();

new Foo().getName();

new new Foo().getName()

---------------------------------------------------

4.

function Fn() {

this.x = 100;

this.y = 200;

this.getX = function () {

console.log(this.x);

}

}

Fn.prototype.getX = function () {

console.log(this.x);

};

Fn.prototype.getY = function () {

console.log(this.y);

};

var f1 = new Fn;

var f2 = new Fn;

console.log(f1.getX === f2.getX);

console.log(f1.getY === f2.getY);

console.log(f1.\_\_proto\_\_.getY === Fn.prototype.getY);

console.log(f1.\_\_proto\_\_.getX === f2.getX);

console.log(f1.getX === Fn.prototype.getX);

console.log(f1.constructor);

console.log(Fn.prototype.\_\_proto\_\_.constructor);

f1.getX();

f1.\_\_proto\_\_.getX();

f2.getY();

Fn.prototype.getY()

-----------------------------------------------------------------

5.

var name = 'zhufengpeixun';

var Fn =function (name) {

var name = 'world';

this.name = 'zhufeng';

this.sex = function () {

this.name = 'hello';

}

}

var f1 = new Fn(name);

var f2 = new Fn('age');

console.log(f1.name);

console.log(f2.age);

f1.sex();

console.log(f1);

f1.sex === f2.sex

Fn.name == f1.name

------------------------------------------------------------------

6.

function Fn() {

var a = 1

this.a = a

}

Fn.prototype.say = function () {

this.a = 2

}

Fn.prototype = new Fn;

var f1 = new Fn;

f1.\_\_proto\_\_.b = function () {

this.a = 3

};

console.log(f1.a)

console.log(f1.prototype)

console.log(f1.b)

f1.hasOwnProperty('b')

'b' in f1

console.log(f1.constructor == Fn)

7-----------------------------------------------------------------------------

let n=m=10;

var obj={

n:10,//

f1:(function (m) {

n+=n;

return ()=>{

console.log(++this.n,this.m==undefined?m:++m);

}

})(n)

}

function Fn(){

this.n=n;

n++<21?n++:this.n++;

f1.call(this,this.n);

this.getN=function () {

console.log(this.n++);

}

}

function f1(n){

console.log(this.n++);

}

var f2=new Fn();//{n:21,getN:function(){}}

var f3=new Fn();//{n:22,getN:function(){}}

//Fn.prototype.f1=f1

f2.\_\_proto\_\_.f1=f1;

Fn.prototype.\_\_proto\_\_.f1=f1;

Fn.prototype.\_\_proto\_\_.getN=f2.getN;

console.log(f2.getN == f3.getN);

console.log(f2.getN === Fn.prototype.getN)

console.log(f2.f1 === Object.f1);

obj.f1();// NaN ,11

obj.\_\_proto\_\_.f1();

obj.getN();

f2.f1();

f3.f1.call(f2);

console.log(n, m);

8.let n=10,  
 obj = {  
 n:20  
 }  
let fn = obj.fn = (function () {  
 n++;  
 return function (m) {  
 n+=10+(++m);  
 this.n+=n;  
 console.log(n);  
 }  
})(obj.n)  
fn(10);  
obj.fn(10);  
console.log(n,obj.n)