// 1.this三板斧

// 1.1

function show() {

console.log('this:', this)

}

var obj = {

show: show

}

obj.show();

答案：obj

解析：被别人点出来的函数，this是调用函数的那个对象

function show() {

console.log('this:', this)

}

var obj = {

show: function() {

show()

}

}

obj.show()

答案：window

解析：相当于直接调用函数show(),this指向window

// 1.2

var obj= {

show: function () {

console.log('this:', this)

}

};

(0, obj.show)()

答案：window

解析：相当于(obj.show)(), 直接调用函数，this指向window

// 1.3

var obj= {

sub: {

show: function () {

console.log('this:', this);

}

}

}

obj.sub.show()

答案：sub

解析：根据最后一个点，show函数是被sub点出来的，this指向sub

// 1.4

var obj = {

show: function () {

console.log('this:', this)

}

}

var newobj = new obj.show()

答案：newobj

解析：相当于构造函数的实例，this指向该实例

// 1.5&1.6

var obj = {

show: function() {

console.log('this:', this)

}

}

var newobj = new (obj.show.bind(obj))()

答案：newobj

解析：相当于构造函数的实例，this指向该实例

// 1.7

var obj = {

show: function (){

console.log('this:', this)

}

}

var elem = document.getElementById('test');

elem.addEventListener('click',obj.show);

答案：elem

解析：相当于shwo函数是被elem点出来的，this指向elem

elem.addEventListener('click',obj.show.bind(obj));

答案：obj

解析：通过bind绑定this指向obj

elem.addEventListener('click',function() {

obj.show()

})

答案：obj

解析：相当于show函数是被obj点出来的，this指向obj，但是show函数外面的this指向的是elem

// 2.作用域

// 2.1

var person = 1

function showPerson () {

var person = 2

console.log(person)

}

showPerson()

答案：2

解析：showPerson函数作用域中有定义了person变量，优先取当前函数中的变量值2

// 2.2

var person = 1

function showPerson () {

console.log(person);

var person = 2;

}

showPerson()

答案：undefined

解析：变量提升，showPerson函数内部相当于：

var person;

console.log(person)

person = 2

// 2.3

var person = 1;

function showPerson () {

console.log(person);

var person = 2;

function person() {}

}

showPerson()

答案：ƒ person(){}

解析：函数声明提升和变量提升，showPerson函数内部相当于：

function person() {}

var person ---> 重新声明并不会将person 重置为undefined

console.log(person)

person = 2

// 2.4

var person = 1;

function showPerson () {

console.log(person)

function person(){}

var person = 2;

}

showPerson()

答案：ƒ person(){}

解析：函数声明提升和变量提升，showPerson函数内部相当于：

function person() {}

var person ---> 重新声明并不会将person 重置为undefined

console.log(person)

person = 2

// 2.5

for(var i= 0;i<10;i++){

console.log(i)

}

答案：0-9

解析：每次循环都打印当时对应的i值

for (var i=0; i<10; i++) {

setTimeout(function() {

console.log(i);

},0)

}

答案：10个10

解析：定时器执行时，全局变量i已经被复制为10，此时有10个定时器，10个定时器执行时打印的均为10

for (var i=0; i<10; i++) {

(function (i) {

setTimeout(function () {

console.log(i)

},0);

})(i);

}

答案：0-9

解析：闭包形式将当次循环中的i传递给内部的函数，定时器在运行时获取的是闭包函数作用域中的变量i,而不是全局变量i

for (let i = 0; i<10; i++) {

console.log(i)

}

答案：0-9

// 3.面向对象

// 3.1

function Person () {

this.name = 1;

return {};

}

var person = new Person();

console.log('name:', person.name);

答案：name: undefined

解析：实例person 是一个空对象{} ,person.name为undefined

// 3.2

function Person () {

this.name = 1

}

Person.prototype = {

show: function() {

console.log('name is:', this.name)

}

}

var person = new Person()

person.show();

答案：name is: 1

解析：show函数中的this指向构造函数实例person对象，this.name即为person.name，值为1

// 3.3

function Person() {

this.name = 1;

}

function Person () {

this.name = 1;

}

Person.prototype = {

name: 2,

show: function() {

console.log('name is:', this.name)

}

}

var person = new Person()

Person.prototype.show = function () {

console.log('new show');

}

person.show();

答案：new show

解析：person是构造函数Person的实例

person实例上本身没有show函数， person.show其实是读取的person.\_\_proto\_\_.show，即Person.prototype.show

后面Person.prototype.show被覆盖，因此打印的是新的show函数“new show”

// 3.4

function Person () {

this.name = 1;

}

Person.prototype = {

name: 2,

show: function () {

console.log('name is:', this.name)

}

}

var person = new Person();

var person2 = new Person();

person.show = function () {

console.log('new show');

};

person2.show();

答案： name is: 1

person.show();

答案： new show

解析：person2和perosn 都是构造函数Person的实例对象

但person实例上有自己的show方法，person2上没有，搜索其原型链上的show方法，并执行

// 4.综合题

function Person() {

this.name = 1;

}

Person.prototype = {

name: 2,

show:function () {

console.log('name is:' ,this.name)

}

}

Person.prototype.show();

答案：2

解析：根据最后一个点，相当于prototype调用的show方法

(new Person()).show();

答案：1

解析：相当于构造函数的实例调用的show方法，用实例上本身的name值，如果没有再取原型上的name值