

Table of Contents

1. [2016年4月5日 面向对象高级复习-by jk](#)

2016年4月5日 面向对象高级复习-by jk

1, 面向对象的基本概念

1> 创建对象的方法有哪些? (至少写出 2 中)

2> 使用字面量的方法创建对象下面两种方式有什么不同

```
var o1 = {  
  name: "jim",  
  age: 19,  
  gender: "男"  
};  
var o2 = {  
  "name": "tom",  
  "age": 20,  
  "gender": "男"  
};
```

3> 使用下面代码创建对象

```
var Person = function () {  
  name = "jim";  
  age = 19;  
  gender = "男";  
};  
var p = new Person();
```

代码: `console.log(p.name)` 执行的结果是多少?

4> 创建对象 Student(name, age, gender)

5> 对象

```
var o = {  
  name: 'jim',  
  age: 19,  
  gender: '男',  
  sayHello: function () {  
    alert( 'Hello World' );  
  }  
};
```

访问对象的 name 成员. 代码如何编写? 调用方法 sayHello 可以如何编写代码?

6> 已知 电压 / 电阻 的值 等于 电流. 写一个函数, 给定电压和电阻, 求电流

7>

写一个函数, 参数要求是一个字符串, 如果不是字符串则报错如果参数是 #开头, 则返回页面中指定 id 的元素, 例如

```
get( "#dv" )
```

返回页面中 id 为 dv 的元素;

如果参数是 .开头, 则返回页面中 class 属性为该参数的所有元素, 例如

```
get( ".cName" )
```

返回页面中 class 属性为 cName 的所有元素;

如果参数没有 # 和 . 则返回指定标签名的元素, 例如

```
get( "p" )
```

返回页面中所有的 p 元素.

2, 原型的基本使用

1> 如何验证写在构造函数中的方法是都占内存的?

2> 构造函数 **Person** 创建的对象 **p**. 对象 **p** 的原型对象如何获取?

3> 给对象添加属性的办法有那些?

4> 代码

```
var Person = function () {  
    this.name = "张三";  
};  
var p1 = new Person();  
p1.name = "李四";  
var p2 = new Person();  
  
console.log( p2.name );
```

代码的执行结果为多少?

5> 给对象添加方法的办法有那些?

6> 使用继承让 **Student** 对象有 **name**, **age** 和 **gender** 属性, 以及 **study** 的方法

7> 代码

```
var Person = function () {  
    this.name = "tom";  
};  
Person.prototype.name = "jim";  
  
var p = new Person();  
console.log( p.name );
```

代码执行的结果是多少?

8> 如何兼容的获得对象的类型名?

9> 代码

```
var Car = function( name ) {  
    this.carName = name;  
};  
var Person = function( pName, cName ) {  
    this.name = pName;  
    this.car = new Car( cName );  
};  
var extend = function( obj ) {  
    var o = {};  
    for ( var k in obj ) {  
        o[ k ] = obj[ k ];  
    }  
    return o;  
}  
var p1 = new Person( "jim", "BMW" );  
var p2 = p1;  
var p3 = extend( p1 );
```

```
console.log( p1.name + ", " + p1.car.carName );
console.log( p2.name + ", " + p2.car.carName );
console.log( p3.name + ", " + p3.car.carName );

p2.name = "tom";
p2.car.carName = "奥迪";

console.log( p1.name + ", " + p1.car.carName );
console.log( p2.name + ", " + p2.car.carName );
console.log( p3.name + ", " + p3.car.carName );
```

执行的结果是多少？

10> 如何判断属性或方法是被继承的, 而不是被自己实现的?

11> 如何将一个 json 格式的字符串转换成对象? 有多少种方法?

12> 需要实现继承, 有几种实现方法?

13> 如果需要所有的对象都具有一个 inherit 方法. 该方法有一个对象参数.

例如

```
p.inherit( obj )
```

返回一个新的对象, 该对象继承自 p 和 obj

14> 如果需要多有的函数都有方法 inherit. 该方法有

一个对象参数.

例如

```
Person.inherit()
```

该方法返回一个对象, 该对象继承自 Person.prototype

15> 函数的原型属性和原型对象有什么异同?

3, 原型链

1> 什么是原型继承?

2> 分析一下原型继承的优缺点?

3> 解释一下属性搜索原则

4> 代码

```
var Person = function () {  
    this.name = "jim";  
};  
Person.prototype.name = "tom";  
var p = new Person();  
p.name = "jack";  
delete p.name;  
console.log( p.name );
```

执行的结果是多少?

5> 原型对象有什么用？

6> 函数的原型对象是什么？

7> Object 继承自什么？

8> 代码

```
function fn() {}
```

函数 fn 继承自什么？

9> 代码

```
function Foo() {}  
function Func() {}  
Func.prototype = new Foo();  
var p = new Func();
```

描述出代码中对象的原型链

10> 代码

```
function Foo1() {}  
function Foo2() {}  
Foo1.prototype = new Foo2();  
var o = new Foo1();  
console.log( o instanceof Foo1 );
```

代码的执行结果多少？

11> 代码

```
function Foo1() {}  
function Foo2() {}  
var o = new Foo1();  
Foo1.prototype = new Foo2();  
console.log( o instanceof Foo1 );
```

代码的执行结果多少？

4, 作用域

1> 解释一下什么是词法作用域

2> 代码

```
var num = 123;  
function func() {  
    num = 456;  
}  
console.log( num );
```

执行结果是什么？

3> 代码

```
var num = 123;  
function num() {  
    console.log( num );  
}  
console.log( num );
```

代码的执行结果是什么？

4> 代码

```
var num = 123;
var func = function () {
  console.log( num );
  var num = 123;
  console.log( num );
};
func();
```

执行结果是多少？

5> 代码

```
var num = 123;
function foo1() {
  console.log( num );
}
function foo2() {
  var num = 456;
  foo1();
}
foo2();
```

执行结果是多少？

5, 闭包的概念

1> 解释一下什么是闭包？

2> 代码

```
function Foo() {  
    var o = { name: "jim" };  
    return {  
        o: o,  
        set_Name: function ( name ) {  
            o.name = name;  
        };  
    };  
}  
var o = Foo();  
console.log( o.name );  
console.log( o.o.name );  
o.set_Name( "tom" );  
console.log( o.o.name );
```

代码的执行结果是多少？

3> 代码

```
var arr = [ { name: 'jim' },  
            { name: 'tom' },  
            { name: 'jack' },  
            { name: 'clark' } ];  
for ( var i = 0; i < arr.length; i++ ) {  
    arr[ i ].sayHello = function () {  
        console.log( arr[ i ].name );  
    };  
}  
  
for ( var k in arr ) {  
    arr[ k ].sayHello();  
}
```

代码的执行结果为多少？如果不行为什么？

4>

写一个函数, 返回一个有姓名, 性别, 和年龄的对象. 要求 对象的参数有函数参数调用传递得到. 并且要求对象的数据 不允许被修改.