

20170512面向对象3

var | let :变量

const 常量:

一样没有预解析;

声明立马赋值 (不然会报错);

在同一域不允许有同名常量, 不然会报错;

hasOwnProperty();

某个属性是不是这个对象*自身的*属性;

返回值为布尔值:

是: true; 否: false;

--
constructor;

某个对象是那个? 构造函数构造出来的;

虚的 (提供查找构造函数的线索) 只能是参考;

因为constructor本身容易被修改, 所以说在有可能被修改的时候认为

主动添加一个constructor属性指向正确的构造函数;

instanceof

二元运算符;

某个对象是不是某个构造函数构造出来的

左边的值是不是右边的构造出来的;

继承:

自身带有父辈的一些特征, 自己也有自己的特征;

事件函数中的this指向该事件的对象。

函数直接调用默认走window.

单纯只要函数名 + 括号, this就是window;

方法的this默认为绑定方法的对象。arr.fn();

1. 事件调用;

2. 自己调用;

3. new;

4. call .apply;

5. 定时器

1 拷贝继承

1. 属性的继承 {

调用父类并且使用call来修改this的指向;

}

2. 方法的继承{

将父类的原型上的方法通过for in 赋值给子类

}

浅拷贝:

通过 for in 将简单类型值, 赋值给另一个对象(赋值);

但是如果for in 中遇到看复合类型依然是赋址;

类式继承:

1. 属性的继承{

调用父类并使用call来修改this指向

}

2. 方法的继承

