1. this
2. 关于this

* this既不指向函数自身，也不指向函数的词法作用域。
* this是在函数被调用时发生的绑定，它指向什么完全取决于函数在哪里被调用。

1. this全面解析

* this绑定的4条规则：

由new调用 绑定到新创建的对象

由call apply bind调用 绑定到指定对象

由上下文对象调用 绑定到那个上下文对象

默认：严格模式下绑定到undefined 非严格模式下绑定到全局对象

* 箭头函数会继承外层函数的this绑定，像es5的self=this机制
* 参数传递是一种隐式赋值 其和赋值容易导致丢失绑定对象
* call函数中，如果传入了一个原始值当作thisd 绑定对象，这个原始值会被转换为它的对象形式，称为装箱。string->new String()
* 如果把null undefined作为this的绑定对象传入call apply bind 这些值在调用时会被忽略，实际应用的是默认规则。更安全的是传入一个特殊的对象，把this绑定到这个对象上不会有影响，即MDZ-非军事区，一个空的非委托对象。

用Object.create(null)创建

function foo(a,b){

console.log(a,b);

}

foo.apply(null,[1,2]);//把数组展开成参数

var bar=foo.bind(null,2);//使用bind进行函数柯里化

bar(3);

1. other
2. this导致的隐式变量

function foo(num){

console.log("foo:"+num);

this.count++;

}

foo.count=0;

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

if(i>5){

foo(i);

}

}

console.log(foo.count);

console.log(window.count);//NaN

/\*foo 里面的this 指 window

左查询会隐式声明一个全局变量为undefined

对其++运算所以为NaN\*/

1. argument.callee已被弃用 严格模式下错误
2. 执行上下文在运行时确定，作用域在定义时确定。
3. 题
4. 设置表单元素为必填：required
5. p:nth-child(even){}选择其父元素下偶数位的子元素，且子元素为p元素，从1开始

p:last-child{}选择其父元素的最后一个元素，且该子元素为P元素

1. 在Microsoft Internet Explorer 中，盒模型默认使用的是 content-box
2. rem是相对于根节点html的
3. ++的优先级高于==
4. canvas 中绘制的元素不可以通过浏览器提供的接口获取到。
5. 同源是指，域名domain，协议protocal，端口port相同
6. 通过new出来的，调用的肯定不是同一个方法
7. http协议中可以控制缓存的



1. a标签里面不能有a标签
2. json xml

