**Js void操作符**

返回undefined，由于在js中，undefined不是保留关键字，那么它就可以被赋值，在任何条件下保证返回的是undefined，采用void 0。

在href和image中使用空的src，若值为空 src=””,任然会发起一个http的查询，若是src=”javascript:void(0);”，浏览器不会发起任何http请求。

Js sort排序  


Firefox下排序正常  


Chrome下sort排序错误

Js类型比较

!”” 与 !0 , !”0” 的异同 !variable 与 variable !==””的区别

Jquery data(“name”)和attr(“data-name”)的区别

val() 和 prop(“value”)的异同

<script>

function runAsync(){

var p = new Promise(function(resolve, reject){

//做一些异步操作

setTimeout(function(){

setTimeout(function(){

append('执行完成');

},10)

resolve('随便什么数据');

}, 1000);

});

return p;

}

function getA(){

var p = new Promise(function(resolvecall, rejectcall){

//做一些异步操作

setTimeout(function(){

append("getA console");

//reject更改状态 resolve不会响应

resolvecall("getA")

//rejectcall("getA")

}, 1000);

});

return p;

}

function getB(){

var p = new Promise(function(resolve, reject){

//做一些异步操作

//setTimeout(function(){

append("getB console");

resolve(function(){console.log("resolve getB")})

//}, 100);

});

return p;

}

function getC(){

var p = new Promise(function(resolve, reject){

//做一些异步操作

append("getA resolve console ");

resolve("getC")

// setTimeout(function(){

// append("getA resolve console ");

// resolve("getC")

// }, 1);

});

return p;

}

runAsync()

.then(function(data){

append(data)

if(typeof(data)==="function"){

data.call(data)

}

return getA()

})

.then(function(data){ // do promise A

reject()//这里reject 后 状态更改，后续的promise不会执行

append(data);

if(typeof(data)==="function"){

data.call(data)

}

return getB()

},function(data){

console.log(data);

if(typeof(data)==="function"){

data.call(data)

}

return getC();

})

.then(function(data){

append(data)

if(typeof(data)==="function"){

data.call(data)

}

return four

},function(){})

.then(function(data){

append(data)

if(typeof(data)==="function"){

data.call(data)

}

},function(){})

.catch(function(reason){

alert(reason)

})

</script>