【HTML/CSS/DOM】

\*如何在一个CSS动画播放完成后继续播放另一个CSS动画？webkitAnimationEnd

\*简述下DOM中事件传递机制？捕获-》冒泡

阻止事件传递到父元素stopPropagation（本事件类型）、stopImmediatePropagation（所有事件）

阻止元素默认行为preventDefault（标签内的onclick等事件处理）区别是啥？

\*如何在站点里用自定义字体？如何解决自定义字体的包值问题？

\*SASS中如何计算百分比？用percentage函数

\*mixin、extend的使用方法及优缺点？

mixin配合@include生成代码块（会生成多余代码块）

@extend为继承某个类（不会生成多余代码，但是会重复出现类名，如果某个类名在html内并不会使用，只是用于被继承，则可以使用%占位符号来命名这个类解决，如%flex-col）

\*flex布局

【JS】

\*简述sessionStorag、localStorage的区别及作用域？1\*如何在两张html之间如何传递数据？如果用URL的话长度限制呢？

\*如何保证页面代码更新后，用户能不受缓存影响立刻刷新为最新的页面？

\*有一个列表内有1000个条目，如何去监听用户点击了某个条目内的某个按钮？监听父元素的事件然后判断事件发生目标的DOM是谁

\*如何禁止页面回退？window.history.forward(1), location.replace,捕获popstate事件

\*TAP事件处理？点透？fastclick

\*js模块化工具：AMD和CMD区别?(AMD预加载，CMD需要懒加载)，loadsh(vuecli内置)

\*简单描述下原型链，prototype和\_\_proto\_\_区别？JS继承方法？

1、实例化的对象是不存prototype，只有function存在prototype，function的prototype的构造方法指向的就是function自己

2、\_\_proto\_\_指向的是父类的prototype

3、call/apply区别,主要是传递参数的方式不同，作用都是执行father方法，但是father方法内的this全部替换为了apply/call后面第一个参数这个对象

function classFather(name,age){

this.name=name;

this.age=age;

}

function classSon(name,age){

father.apply(this,arguments);

father.call(this,name,age);

}

继承好了构造方法内所有属性后，接下来就需要继承父类里面prototype的所有属性方法，通过遍历父类的prototype所有属性和方法赋值给子类的\_\_proto\_\_[属性]下来继承所有prototype的方法

for( var param in father.prototype ){

if( !son.\_\_proto\_\_[param ]){

son.\_\_proto\_\_[param] = father.prototype[param];

}

}

\*document.createDocumentFragment 在生成大量数据的列表时利用它来进行缓存

\* for、map、foreach区别？

速度上map>foreach>for ,

map可以return处理后的新数组,foreach没有任何返回值

\*null,undefined的区别？该如何检测？==和===的区别？

\*document.write有什么特点？异步中会有什么问题？

\*如何解决ajax接口嵌套过多？promise+async/await

【兼容性】

\*移动端上实现DIV内指定区域流畅的滑动列表需要哪些参数？-webkit-overflow-scrolling: touch;

\*正确捕获用户输入事件？oninput,onkeyup,onchange,区别

\* img标签如果暂时没有src的值的话，需要做什么处理？

\* 如果需要将input标签设置禁止编辑的话，不要用disabled 用readOnly

\* new Date('12-01-02 12:02:02')格式化时间有什么问题？（IOS不行）

\* 以下执行结果是什么？

(function(){

a=100;

let a=5;

console.log(a);

})();

let a=200;

(function(){

let a=100;

console.log(a);

})();

var objA={

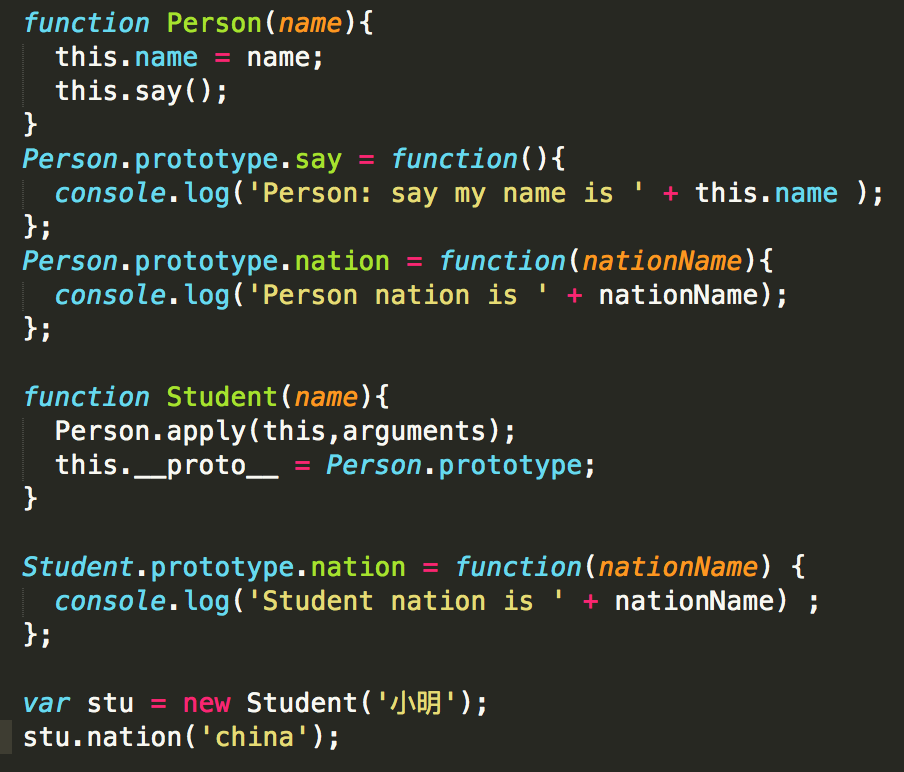
name:"老王",

do:()=>{console.log("名字:"+this.name)},

};

objA.do();//打印的是？

**类继承2**



1、打印结果是？

2、这端代码有哪些问题?

3、如果想要达到子类的方法可以调用父类的方法，打印结果是下面这样，该怎么办？

Person:say my name is 小明

Person nation is china

Student nation is china

==========================================================================

一个对象内有2个属性，改变一个属性的值，另外一个属性也同样发生改变，以下哪种方式好？

var obj = { phone: “13700000001”,myPhone:”13700000001”};

1、 obj.phone = '1380000001';

obj.myPhone = obj.phone;

2、 function setPhone(val) {

obj.phone = val;

obj.myPhone =val;

}

==========================================================================

请优化以下代码

var box = document.getElementById(‘box’);

setInterval(function(){

var dy = box.offsetTop + 10;

box.style.top = dy+’px’;

},33);

==========================================================================

var arr= ['wangchen','zhangsan','peter'];

var txt='peter';

请用最简单的方式判断txt的值，是否在arr内存在

==========================================================================

如何合并2个对象的属性，并且如有相同属性，则以obj2覆盖obj2

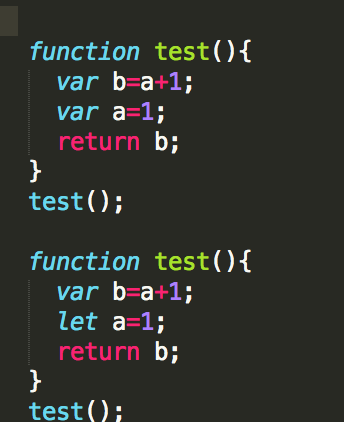
var obj = {session:’12346’, age:19};

var obj2 = {session:’aaa’, name:’cx’};

经过合并后

console.log(obj);// { session:’aaa’,name:’cx’,age:19};

==========================================================================



第一题

NaN 因为var a=1:变量提升后变成

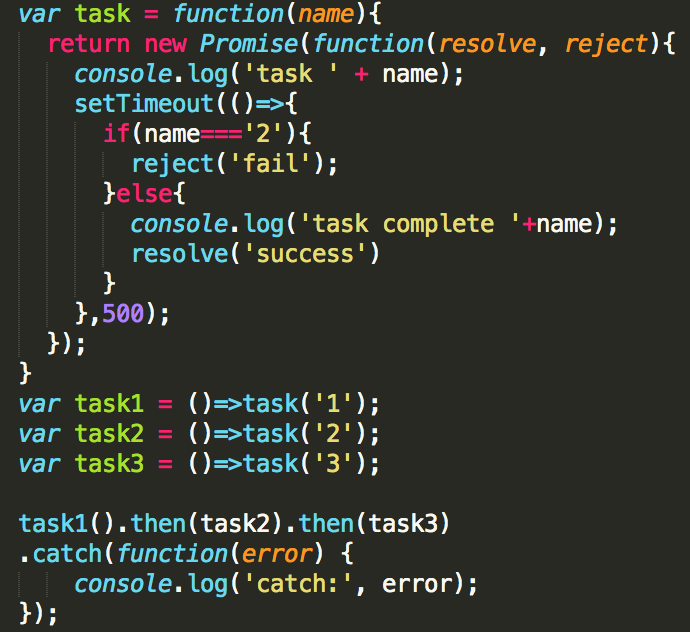
Var a;

B=a+1;

A=1;

第二题

报错：a没有定义



Task 1

阻塞等待

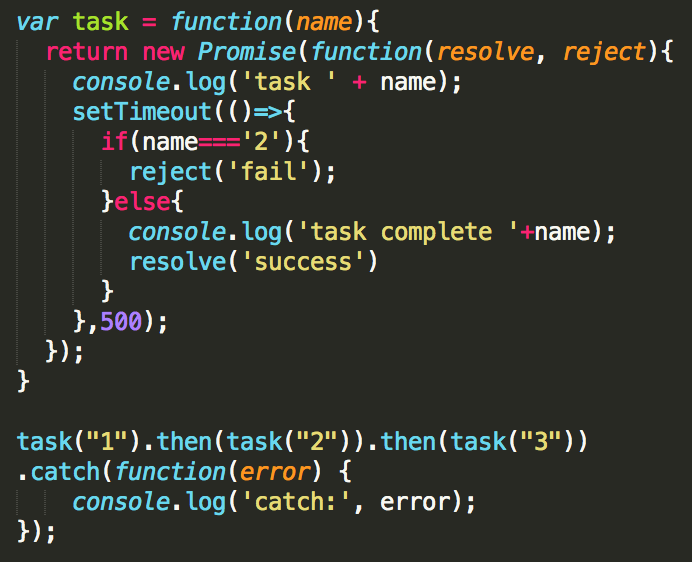
Task complete 1

Task 2

阻塞等待

catch fail

结束



task 1

task 2

task 3

阻塞等待

task complete 1

task complete 3

uncatch fail



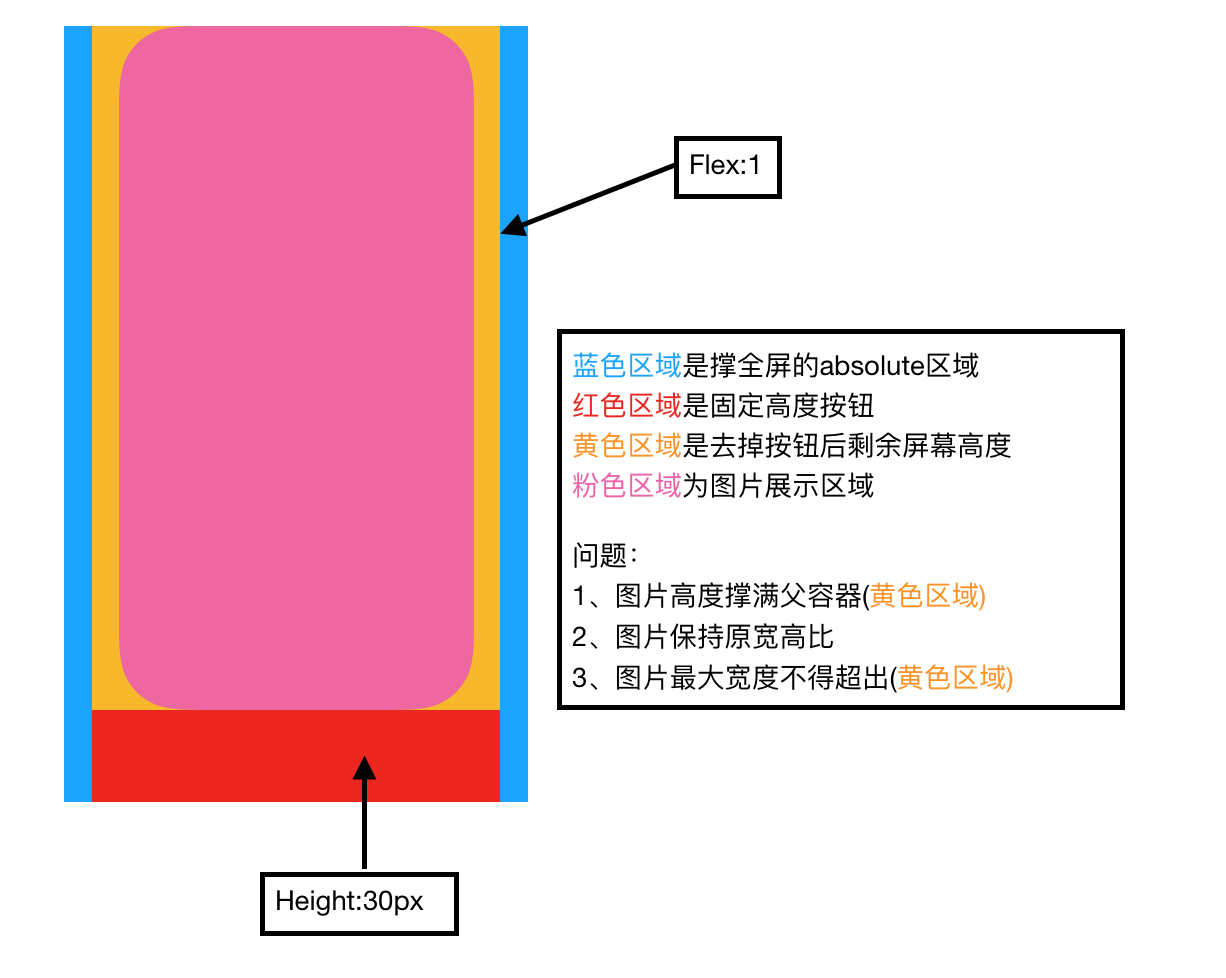
Task 2

Task 2 fail

Task 1

Task 1 success

complete



**类继承1**



1、打印结果是？

2、请使用ES5方法将Student继承自Person，提示:call/apply, \_\_proto\_\_, prototype

3、继承后打印结果是?

4、如果想要达到子类的方法可以调用父类的方法，打印结果是下面这样，该怎么办？

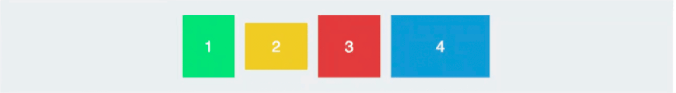
Person:say my name is 小明

Person nation is china

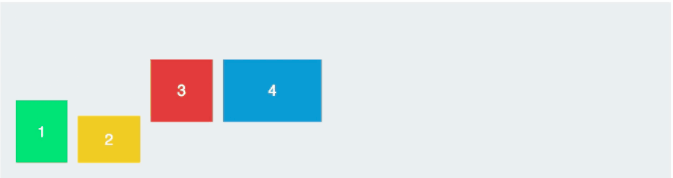
Student nation is china

实现如下2个布局？flex-direction\flex-pack\flex-align\align-self\flex-wrap

1.body内div横向排列，并且垂直横向居中



1. 单独设置1和2对齐到底部，其他模块保持垂直居中



==========================================================================

以下布局如何实现？

当main比较长时，footer跟随在main尾部 当main比较短时，footer置底

