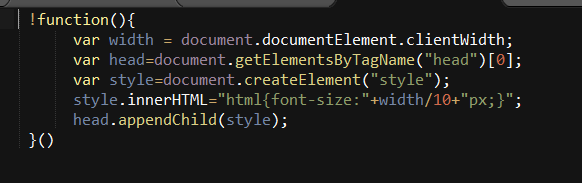
1. 浮动元素在使用的时候，注意设置高度一致；不然会导致下面的盒子掉下去；因为浮动元素会脱离文档流，会浮动，浮动的时候，会碰到元素就掉落；
2. 解决方案：固定盒子的高度；
3. 设置透明度的时候，如果使用opacity，那么父元素也会跟着一起改变；
4. 解决方案：使用rgba的方法设置透明度；
5. 动态创建移动端的元素，我们需要根据屏幕的宽度动态自动改变元素的宽高；首先要规定1个rem的相对大小；假设width/10,那么就是10个rem占满全屏，下面的元素的宽高就可以参照这个单位了；接下来就是根据UI提供的视觉稿来计算到底是多少rem，比如一个宽1000px的一个500px的盒子，那么适配的rem单位就是10/1000\*500，就是实际的rem；
6. 块级元素垂直居中的问题；display：table-cell; vertical-align: middle;
7. 使用display：table-cell，来使大小不确定的元素垂直居中；可以在内部使用span标签把文字包裹起来，设置inline-block，显示为图片的格式即可；
8. Js中的简写：

.三元操作符：const big = x > 10 ? " greater 10" : x

短路运算符：const variable2 = variable1  || 'new';

声明变量的写法：

let x, y, z=3;

If条件存在的简写方式;

if (likeJavaScript)

1. 循环的简写：

for (let index in allImgs)

1. 在使用递归的时候，考虑清楚边界条件，否则易导致内存泄露；这点一定要注意；
2. 比如：数组的元素求和；
3. **17.请描述一下 cookies，sessionStorage 和 localStorage 的区别？**
4. cookie在浏览器和服务器间来回传递。 sessionStorage和localStorage不会
5. sessionStorage和localStorage的存储空间更大；
6. sessionStorage和localStorage有更多丰富易用的接口；
7. sessionStorage和localStorage各自独立的存储空间；
8. **19.谈一谈JavaScript作用域链**
9. 当执行一段JavaScript代码（全局代码或函数）时，JavaScript引擎会创建为其创建一个作用域又称为执行上下文（Execution Context），在页面加载后会首先创建一个全局的作用域，然后每执行一个函数，会建立一个对应的作用域，从而形成了一条作用域链。每个作用域都有一条对应的作用域链，链头是全局作用域，链尾是当前函数作用域。
10. 作用域链的作用是用于解析标识符，当函数被创建时（不是执行），会将this、arguments、命名参数和该函数中的所有局部变量添加到该当前作用域中，当JavaScript需要查找变量X的时候（这个过程称为变量解析），它首先会从作用域链中的链尾也就是当前作用域进行查找是否有X属性，如果没有找到就顺着作用域链继续查找，直到查找到链头，也就是全局作用域链，仍未找到该变量的话，就认为这段代码的作用域链上不存在x变量，并抛出一个引用错误（ReferenceError）的异常。
11. **20.如何理解JavaScript原型链**
12. JavaScript中的每个对象都有一个prototype属性，我们称之为原型，而原型的值也是一个对象，因此它也有自己的原型，这样就串联起来了一条原型链，原型链的链头是object,它的prototype比较特殊，值为null。
13. 原型链的作用是用于对象继承，函数A的原型属性(prototype property)是一个对象，当这个函数被用作构造函数来创建实例时，该函数的原型属性将被作为原型赋值给所有对象实例，比如我们新建一个数组，数组的方法便从数组的原型上继承而来。
14. 当访问对象的一个属性时, 首先查找对象本身, 找到则返回; 若未找到, 则继续查找其原型对象的属性(如果还找不到实际上还会沿着原型链向上查找, 直至到根). 只要没有被覆盖的话, 对象原型的属性就能在所有的实例中找到，若整个原型链未找到则返回undefined。
15. **13.对前端工程师这个职位你是怎么样理解的？**
16. a. 前端是最贴近用户的程序员，前端的能力就是能让产品从 90分进化到 100 分，甚至更好
17. b. 参与项目，快速高质量完成实现效果图，精确到1px；
18. c. 与团队成员，UI设计，产品经理的沟通；
19. d. 做好的页面结构，页面重构和用户体验；
20. e. 处理hack，兼容、写出优美的代码格式；
21. f. 针对服务器的优化、拥抱最新前端技术。
22. **14.平时如何管理你的项目？**
23. 先期团队必须确定好全局样式（globe.css），编码模式(utf-8) 等；
24. 1编写习惯必须一致（例如都是采用继承式的写法，单样式都写成一行）；
25. 2标注样式编写人，各模块都及时标注（标注关键样式调用的地方）；
26. 3 页面进行标注（例如 页面 模块 开始和结束）；
27. 4 CSS跟HTML 分文件夹并行存放，命名都得统一（例如style.css）；
28. 5 JS 分文件夹存放 命名以该JS功能为准的英文翻译。
29. 6图片采用整合的 images.png png8 格式文件使用 尽量整合在一起使用方便将来的管理
30. Arguments的合并，例如，将用户输入的内容使用逗号连接起来，发送数据；
31. arguments不能用slice方法直接截取，需要先转换为数组，var args = Array.prototype.slice.call(arguments);合并参数可以使用concat方法，并且也需要将arguments先转换为数组才能使用concat进行合并。最用使用apply执行传入的函数即可。
32. **function** partialUsingArguments(fn) {
33. *//先获取p函数第一个参数之后的全部参数*
34. **var** args **=** Array.prototype.slice.call(arguments,1);
35. *//声明result函数*
36. **var** result **=** **function**(){
37. *//使用concat合并两个或多个数组中的元素*
38. **return** fn.apply(**null**, args.concat([].slice.call(arguments)));
39. }
40. **return** result;
41. }
42. **30.清除浮动有哪些方式？比较好的方式是哪一种？**
43. （1）父级div定义height。
44. （2）结尾处加空div标签clear:both。
45. （3）父级div定义伪类:after和zoom。
46. （4）父级div定义overflow:hidden。
47. （5）父级div定义overflow:auto。
48. （6）父级div也浮动，需要定义宽度。
49. （7）父级div定义display:table。
50. （8）结尾处加br标签clear:both。
51. 比较好的是第3种方式，好多网站都这么用。
52. **31.box-sizing常用的属性有哪些？分别有什么作用？**
53. (Q1)box-sizing: content-box|border-box|inherit;
54. (Q2)content-box:宽度和高度分别应用到元素的内容框。在宽度和高度之外绘制元素的内边距和边框(元素默认效果)。
55. border-box:元素指定的任何内边距和边框都将在已设定的宽度和高度内进行绘制。通过从已设定的宽度和高度分别减去边框和内边距才能得到内容的宽度和高度。