1. sass中使用\_\_\_%\_+name\_\_\_声明一个占位符
2. sass中通过\_\_@extend\_\_\_\_来调用占位符
3. sass中实现大段代码的重用，可以通过\_\_混合宏\_\_\_\_\_和\_\_继承\_\_\_\_和 \_\_占位符\_\_\_\_来实现
4. p {color: #010203 + #040506;}编译后CSS中的P的颜色是\_\_#050709\_\_\_\_
5. sass允许使用继承，编译之后的结果以\_\_\_@extend\_\_\_\_方式呈现
6. sass中变量\_\_\_\_有\_\_\_作用域的问题（有/无）
7. 如果变量需要镶嵌在字符串中，那么需要\_\_@include\_\_\_\_的方式来引用
8. sass中实现继承的关键字是\_\_@extend\_\_\_\_\_\_
9. sass的变量包括\_\_\_&\_\_\_\_\_和\_\_\_名字\_\_\_\_和\_\_\_\_\_值\_\_\_\_三部分
10. #{}的作用是\_\_\_插值\_\_\_\_
11. sass中的\_\_.scss\_\_\_\_\_方法即使不调用依旧会编译在css文件中
12. sass中是css的\_\_\_\_预处理\_\_\_,不可以\_\_\_\_\_\_（可以/不可以）直接运用到 项目中
13. css编译之后的结果是header a,footer a{},通过sass怎么书写 \_\_\_\_\_\_\_\_
14. sass的嵌套分为\_\_\_属性嵌套\_\_\_\_和\_\_\_\_\_\_选择器嵌套\_\_\_\_\_
15. \_\_\_&\_\_\_符在sass中标识选择父选择器
16. \_\_继承\_\_\_和\_\_占位符\_\_\_\_需要关键字@extend来调用
17. 简答题
18. 阐述继承或者扩展的作用

减少代码的重复量

1. 阐述混合宏/继承/占位符的优缺点

优点：减少代码的重复量

1. 阐述选择器占位符%placeholder和@mixin从功能上来说（在浏览器上渲染的效果是完全相同的）是相同，编译出来的css也相同？为什么？

他们都是通过@extend调用

1. sass中属性混合宏有几种格式？
2. 程序题
3. 已知代码 .box {

width: ((220px + 720px) - 11 \* 20 ) / 12 ;

}

编译出的CSS中的.box的width是\_\_\_\_60px;\_\_\_\_

1. 已知变量$w:45px,SASS中font-size:$w/(30/2+”px”)+”px”编译后的结果是font-size:\_\_\_3px\_\_\_
2. 已知声明一个混合宏

@mixin flex{

Width:100%;

Height:10px;

Background:skyblue;}，要求header与section与footer都实现以上代码，有几种方式实现？并写出代码

Header{@include flex;} footer{@include flex;}

section{@include flex;}

Header,footer,section{@include flex;}

1. body{width:200px},在SASS中可以怎么实现？(利用sass的加减乘除来实现四种运算来实现，自由发挥)

body{width:100+100+px}

body{width:400/2+px}

body{width:100\*2+px}