# FLEX原理和应用

## 单择题

1. 作用于flex项目上的属性有 ( )   
A. flex-direction

B. flex-wrap

C. order  
D. align-content

答案：C

2. flex容器属性flex-direction：row-reverse表示 ( )   
A. 主轴为水平方向，起点在左端

B. 主轴为水平方向，起点在右侧

C. 主轴为垂直方向，起点在上沿  
D. 主轴为垂直方向，起点在下沿

答案：B

**3. flex容器属性flex-direction：row表示 ( )**   
A. 主轴为水平方向，起点在左端

B. 主轴为水平方向，起点在右侧

C. 主轴为垂直方向，起点在上沿  
D. 主轴为垂直方向，起点在下沿

答案：A

**4. flex容器属性flex-direction：column表示 ( )**   
A. 主轴为水平方向，起点在左端

B. 主轴为水平方向，起点在右侧

C. 主轴为垂直方向，起点在上沿  
D. 主轴为垂直方向，起点在下沿

答案：C

**5. flex容器属性flex-direction：column-reverse表示 ( )**   
A. 主轴为水平方向，起点在左端

B. 主轴为水平方向，起点在右侧

C. 主轴为垂直方向，起点在上沿  
D. 主轴为垂直方向，起点在下沿

答案：D

**6. 2009年less首次用哪种语言编程 ( )**  
A. Ruby

B. JavaScript

C. Java  
D.C++

答案：A

**7. flex容器属性flex-wrap：wrap表示 ( )**   
A. 换行，第一行在上方

B. 不换行

C. 换行，第一行在下方  
D. 都不是

答案：A

**8. flex容器属性flex-wrap：wrap-reverse表示 ( )**   
A. 换行，第一行在上方

B. 不换行

C. 换行，第一行在下方  
D. 都不是

答案：C

**9. flex容器属性flex-flow是哪两个属性缩写 ( )**   
A. flex-direction flexwrap

B. justify-content align-items

C. justify-content flexwrap  
D. flex-direction align-items

答案：A

**10. flex容器属性justify-content:flex-start表示 ( )**   
A. 右对齐

B. 居中

C. 左对齐  
D. 两端对齐，项目之间的间隔都相等

答案：C

**11. flex容器属性justify-content:flex-end表示 ( )**   
A. 右对齐

B. 居中

C. 左对齐  
D. 两端对齐，项目之间的间隔都相等

答案：A

**12. flex容器属性justify-content:center表示 ( )**   
A. 右对齐

B. 居中

C. 左对齐  
D. 两端对齐，项目之间的间隔都相等

答案：B

**13. flex容器属性justify-content:space-between表示 ( )**   
A. 右对齐

B. 居中

C. 左对齐  
D. 两端对齐，项目之间的间隔都相等

答案：D

**14. flex容器属性align-items：flex-start表示 ( )**   
A. 交叉轴的终点位置

B. 每个项目两侧的间隔相等

C. 交叉轴的起点位置  
D. 项目的第一行文字的基线对齐

答案：C

**15. flex容器属性align-items：flex-end表示 ( )**   
A. 交叉轴的终点位置

B. 每个项目两侧的间隔相等

C. 交叉轴的起点位置  
D. 项目的第一行文字的基线对齐

答案：A

**16. flex容器属性align-items：center表示 ( )**   
A. 交叉轴的终点位置

B. 交叉轴的中点对齐

C. 交叉轴的起点位置  
D. 项目的第一行文字的基线对齐

答案：B

**17. flex容器属性align-items：baseline表示 ( )**   
A. 交叉轴的终点位置

B. 每个项目两侧的间隔相等

C. 交叉轴的起点位置  
D. 项目的第一行文字的基线对齐

答案：D

**18. flex容器属性align-items：stretch表示 ( )**   
A. 如果项目未设置高度或设为auto，将占满整个容器的高度

B. 每个项目两侧的间隔相等

C. 交叉轴的起点位置  
D. 项目的第一行文字的基线对齐

答案：A

**19. flex容器属性align-content：flex-end表示 ( )**   
A. 与交叉轴的终点对齐

B. 与交叉轴的起点对齐

C. 与交叉轴的中点对齐  
D. 与交叉轴两端对齐，轴线之间的间隔平均分布

答案：A

**20. flex容器属性align-content：flex-center表示 ( )**   
A. 与交叉轴的终点对齐

B. 与交叉轴的起点对齐

C. 与交叉轴的中点对齐  
D. 与交叉轴两端对齐，轴线之间的间隔平均分布

答案：C

**21. flex容器属性align-content：space-between表示 ( )**   
A. 每跟轴线两侧的间隔都相等

B. 轴线占满整个交叉轴

C. 与交叉轴的中点对齐  
D. 与交叉轴两端对齐，轴线之间的间隔平均分布

答案：D

**22. flex容器属性align-content：space-around表示 ( )**   
A. 每根轴线两侧的间隔都相等

B. 轴线占满整个交叉轴

C. 与交叉轴的中点对齐  
D. 与交叉轴两端对齐，轴线之间的间隔平均分布

答案：A

**23. flex项目属性order ( )**   
A. 定义了项目的缩小比例，默认为1，如果空间不足，改项目将缩小

B. 定义项目的排列顺序，数值越小，排列越靠前，默认为0

C. 定义了在分配多余空间之前，该项目占据的主轴空间  
D. 定义项目的放大比例，默认是0，即如果存在剩余空间也不放大

答案：B

**24. flex项目属性flex-grow ( )**   
A. 定义了项目的缩小比例，默认为1，如果空间不足，改项目将缩小

B. 定义项目的排列顺序，数值越小，排列越靠前，默认为0

C. 定义了在分配多余空间之前，该项目占据的主轴空间  
D. 定义项目的放大比例，默认是0，即如果存在剩余空间也不放大

答案：D

**25. flex项目属性flex-shrink ( )**   
A. 定义了项目的缩小比例，默认为1，如果空间不足，改项目将缩小

B. 定义项目的排列顺序，数值越小，排列越靠前，默认为0

C. 定义了在分配多余空间之前，该项目占据的主轴空间  
D. 定义项目的放大比例，默认是0，即如果存在剩余空间也不放大

答案：A

**26. flex项目属性flex-basis ( )**   
A. 定义了项目的缩小比例，默认为1，如果空间不足，改项目将缩小

B. 定义项目的排列顺序，数值越小，排列越靠前，默认为0

C. 定义了在分配多余空间之前，该项目占据的主轴空间  
D. 定义项目的放大比例，默认是0，即如果存在剩余空间也不放大

答案：C

**27. flex项目属性flex ( )**   
A. 定义了项目的缩小比例，默认为1，如果空间不足，改项目将缩小

B. 定义项目的排列顺序，数值越小，排列越靠前，默认为0

C. 定义了在分配多余空间之前，该项目占据的主轴空间  
D. 是flex-grow，flex-shrink和flex-basis的缩写，默认是0 1 auto，后两个属性可选

答案：D

**28. flex项目属性align-self ( )**   
A. 允许项目有与其他项目不一样的对齐方式，可覆盖align-items属性

B. 不可覆盖align-items属性

C. 默认值是flex-auto  
D.不允许项目有与其他项目不一样的对齐方式

答案：A

**29. flex项目属性align-self，以下哪项不是align-self的属性值 ( )**  
A. auto

B. flex-start

C. center  
D.space-between

答案：D

**30. flex容器属性flex-wrap：nowrap表示 ( )**   
A. 换行，第一行在上方

B. 不换行

C. 换行，第一行在下方  
D. 都不是

答案：B

**31. flex容器属性align-content：flex-start表示 ( )**   
A. 与交叉轴的终点对齐

B. 与交叉轴的起点对齐

C. 与交叉轴的中点对齐  
D. 与交叉轴两端对齐，轴线之间的间隔平均分布

答案：B

**32. 作用于flex容器上的属性有( )**   
A. order

B. justify-content  
C. flex-grow  
D.以上都不正确。

答案：B

**33. flex容器属性align-content：stretch表示 ( )**   
A. 每跟轴线两侧的间隔都相等

B. 轴线占满整个交叉轴

C. 与交叉轴的中点对齐  
D. 与交叉轴两端对齐，轴线之间的间隔平均分布

答案：B

**34. flex容器属性justify-content:space-around表示 ( )**   
A. 右对齐

B. 每个项目两侧的间隔相等

C. 左对齐  
D. 两端对齐，项目之间的间隔都相等

答案：B

## 判断题

# 1.Flex 是 Flexible Box 的缩写，意为"弹性布局"，用来为盒状模型提供最大的灵活性。()

答案：√

# 2.flex-direction是作用于flex项目上的属性。()

答案：×

# 3.flex容器属性align-item：flex-end表示交叉轴的起点位置()

答案：×

# 4.flex容器属性align-item：center表示交叉轴的中心点()

答案：√

# 5.flex容器属性align-item：baseline表示项目的第一行文字的基线对齐()

答案：√

# 6.flex容器属性align-item：stretch表示如果项目未设置高度或设为auto，将占满整个容器的高度()

答案：√

# 7.flex容器属性align-content：space-between表示：轴线占满整个交叉轴()

答案：×

# 8.flex容器属性align-content：space-around表示：每根轴线两侧的间隔相等()

答案：√

# 9.flex项目属性align-self允许项目与其他项目不一样的对齐方式，可覆盖align-item属性()

答案：√

# 10.flex项目属性align-self的值不包括space-between()

答案：√

# 11.flex布局在移动端适配会更好

答案：√

# 12.flex布局编写的网页在不同的浏览器上会存在兼容性问题

答案：√

## 填空题

# 1.Flex 布局的元素，称为 Flex 容器（flex container），简称"容器"。它的所有子元素自动成为容器成员，称为 Flex 项目（flex item），简称"\_\_\_\_\_"。

答案： 项目**；**

# 2.移动端常用的单位有rem、em、vw和\_\_\_\_\_\_\_。

答案：vh**；**

# 3.flex项目的属性有order、flex-grow、flex-shrink、flex-basis、flex、\_\_\_\_\_\_\_。

答案：align-self**；**

# 4.flex中justify-content属性的作用是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

答案：定义了项目在主轴上的对齐方式**；**

# 5.Flex是Flexible Box的缩写，意为”\_\_\_\_\_\_\_\_\_\_\_\_”，用来为盒状模型提供最大的灵活性。

答案：弹性布局

# 6.设置弹性布局时，需要在容器中输入的代码为：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

答案：display:flex**；**

# 7.设为弹性布局时，子元素的float、clear和\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_属性将会失效。

答案：vertical-align**；**

# 8.容器默认存在两根主轴：\_\_\_\_\_\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

答案：水平主轴 垂直的交叉轴**；**

# 9.主轴的开始位置叫做main start，结束位置叫做\_\_\_\_\_\_\_\_\_\_\_。

答案：main end**；**

# 10.交叉轴的开始位置叫做cross start，结束位置叫做\_\_\_\_\_\_\_\_\_\_\_\_。

答案：cross end**；**

# 11.flex布局中项目默认沿\_\_\_\_\_\_\_\_\_排列。

答案：主轴**；**

# 12.单个项目占据的主轴空间叫做main size，占据的交叉轴空间叫做\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

答案：cross size**；**

## 简答题

# 1.什么是flex布局？

答案：Flex是Flexible Box的缩写，意为“弹性布局”，用来为盒状模型提供最大的灵活性。可以随着页面大小的改变自适应页面布局。目前被所有浏览器所支持。

# 2.flex的父级有哪些属性？

答案：flex-direction、flex-wrap、flex-flow、justify-content、align-items和align-center。

# 3.flex的子容器有哪些属性？

答案：order、flex-grow、flex-shrink、flex-basis、flex和align-self

# 4.如何设置成Flex布局？

答案：块级元素：.box{display:flex}行内元素也可以设置成flex布局，.box{display:inline-flex}；设置成Flex布局后，子元素的float、clear和vertical-align属性将失效。

# 5.谈谈你对Flex布局的理解？

答案：web应用有不同设备尺寸和分辨率，这时需要响应式界面设计来满足复杂的布局需求，Flex弹性盒模型的优势在于开发人员只是声明布局应该具有的行为，而不需要给出具体的实现方式，浏览器负责完成实际布局，当布局涉及到不定宽度，分布对齐的场景时，就要优先考虑弹性盒布局。

# LESS原理和应用

## 单选题

**1. 在less中如何声明变量 ( )**  
A. 使用var声明变量

B. 使用let声明变量

C. 需要使用@符号声明变量，并使用（:）来分配变量中的特定值。还必须在变量的值之后添加分号；  
D. 使用$声明

答案：C

**2. less文件的扩展名是什么 ( )**  
A. .less

B. .css

C. .java  
D. .js

答案：A

**3. 目前less使用的语言是 ( )**  
A. Ruby

B. JavaScript

C. Java  
D.C++

答案：B

**4. 目前less使用的语言是 ( )**  
A. Ruby

B. JavaScript

C. Java  
D.C++

答案：B

**5. less编程语言首次发明是什么时候 ( )**  
A. 2008年

B. 2009年

C. 2010年  
D.2011年

答案：B

**6. npm安装less的时候执行命令npm install less -g中-g的作用是 ( )**   
A. 局部安装less

B. 全局安装less

C. 没什么作用

答案：B

**7. 安装less时执行npm i less --save-dev中dev表示 ( )**   
A. 在全局环境中配置less

B. 不在全局环境中安装less，只在当前文件夹中安装less

C. 以上都不对

答案：A

**8. 如下代码使用less写样式时,不用less写样式时为 ( )**

**.deme{**

**.demo2{**

**width:20px;**

**}**

**}**  
A. .demo>.demo2{ with:20px }

B. .demo,.demo2{ with:20px }

C. .demo .demo2{ width:20px }  
D. 以上都不是

答案：C

**9. 使用less的代码为下面代码 ，怎输出的css代码为( )**

@base: #f938ab;

.box-shadow(@style, @c) when (iscolor(@c)) { -webkit-box-shadow: @style @c; box-shadow: @style @c; }

.box-shadow(@style, @alpha: 50%) when (isnumber(@alpha)) { .box-shadow(@style, rgba(0, 0, 0, @alpha)); }

.box { color: saturate(@base, 5%); border-color: lighten(@base, 30%); div { .box-shadow(0 0 5px, 30%) } }  
A. .box{ color:#fe33ac; border-color:#fdcdea }

B. . box div { -webkit-box-shadow: 0 0 5px rgba(0, 0, 0, 0.3); box-shadow: 0 0 5px rgba(0, 0, 0, 0.3); }

C. . box { color: #fe33ac; border-color: #fdcdea; } .box div { -webkit-box-shadow: 0 0 5px rgba(0, 0, 0, 0.3); box-shadow: 0 0 5px rgba(0, 0, 0, 0.3); }  
D. 都不是

答案：C

**10. 使用less编写 @fnord: "I am fnord.";@var: 'fnord';content: @@var;，解析后的代码为( )**

A. content：“I am fnord”

B. content:''

C. content:'var'

答案：A

**11. less中代码为@width:20px; @height:@width+10px,则@height的值为 ( )**  
A. 10px

B. 20px

C. 30px  
D.50px

答案：C

**12. less中编写代码.fz{font-size:18px} div .demo{color:red; .fz() } ，则.demo的具体样式为( )**  
A. color:red

B. color:red；font-size:18px;

C. color:red;font-size:16px  
D.color:#fff;font-size:12px

答案：B

**13. 代码#nav{color:black} #nav .button{font-size:16px}与下面less代码中哪个编译后想等 ( ) ( )**

A. #nav{

color:black;

.button{

font-size:16px

}

}

B. #nav{

color:black;

.button{

font-size:18px

}

}

C. #nav{

color:black;

#nav .button{

font-size:18px

}

}

D.都没有

答案：A

**14. less中编写代码.line1{ width:100%;color:red;font-size:18px;&:before{conten:"你好代码！"} }，编译后的代码为( )**  
A. .line1 { content:'你好代码' }

B. .line1 { width:100%; color:red;font-size:18px } line1:before{ content:'你好代码！' }

C. .line1 { width:100%; color:red; } line1:before{ content:'你好代码！' }  
D. .line1 { width:100%; color:red; font-size:16px } line1:before{ content:'你好代码！' }

答案：B

**15. less中编写代码@conversion-1:5cm+10mm,结果为：( )**  
A. 15cm

B. .10cm

C. 15mm  
D. 6cm

答案：D

**16. less中编写代码@conversion-2:2-3cm-5mm,结果为：( )**  
A. -1.5cm

B. .0cm

C. -6cm  
D. 6mm

答案：A

**17. less中编写代码@conversion-units:2+5px-3cm,结果为：( )**  
A. 4cm

B. 4px

C. -6cm  
D. 6mm

答案：B

**18. less中编写代码 @base:5%;@filler:@base\*2;结果为：( )**  
A. 10%

B. 5%

C. 7%  
D. null

答案：A

**19. less中编写代码 @base:5%;@filler:@base\*2;@other:@base+@filter结果为：( )**  
A. 10%

B. 15%

C. 14%  
D. null

答案：B

**20. less中编写代码 @base:2cm\*3mm结果为：( )**  
A. 6mm

B. 6cm

C. 5cm  
D. 5mm

答案：B

**21. less中编写代码@color: #224488/2;background-color:#112244+#111结果为：( )**  
A. #223355

B. #223366

C. #112244  
D. #445566

答案A

## 判断题

# 1.学习者样式表(Less)是一种向后兼容的语言, 可以编译为层叠样式表(CSS), 并且可以在客户端和服务器端运行。它是一种开源语言, 并且与跨浏览器兼容。

答案：√

# 2.Less编程语言创立于2009年。()

答案：√

# 3.在Less中, Mixins可以帮助你将一组规则集中的一组属性添加到另一组规则集中。它包括类名作为其属性。可以使用类或id选择器以与CSS样式相同的方式声明Mixins。它可以存储多个值, 并在必要时可以在代码中重用。()

答案：√

# 4.less嵌套用于允许代码遵循某些视觉层次结构, 从而使代码更简单, 更简洁, 更简单。例如, 可以在更少的编程中完成类的嵌套。()

答案：√

# 5.在Less中, 颜色通道功能是外置功能, 用于根据颜色定义设置有关通道的值。()

答案：×

# 6.在Less中, 借助数据URI, 可以将图像直接嵌入样式表中。()

答案：√

# 7.“Less源代码的源映射”选项表示必须将所有CSS文件都包含在源映射中。 ()

答案：√

# 8.Less和SASS的区别之一是SASS使用了红宝石编码 ()

答案：√

# 9.Less和SASS的区别还有是SASS使用$声明变量 ()

答案：√

# 10.Less和SASS的区别还有是SASS为了进行混编，它使用了指南针 ()

答案：√

# 11.Less和SASS两者的命名空间相似 ()

答案：√

# 12.Less和SASS两者的颜色功能不相似 ()

答案：×

# 13.Less和SASS两者的嵌套功能相似 ()

答案：√

# 14.Less和SASS两者的JavaScript评估都是相同的 ()

答案：√

# 15.在LESS中, ＆combinator用于组合嵌套选择器和父选择器 ()

答案：√

# 16.Less支持各种算术运算, 例如+ 、？ , \*, /可以对任何颜色或变量进行操作。()

答案：√

# 17.无法使用css或手写笔代替SCSS()

答案：×

# 18.e()函数的作用是从Less编译器中转义一个值()

答案：√

# 19.将Less预先编译为CSS的最简单方法是将Less.js与Node.js分开使用, 其中Less.js在浏览器外部运行。()

答案：×

# 20.在less中命名空间用于以通用名称将Mixins分组。你可以在Less文件中借助命名空间和访问器来执行封装。()

答案：√

# 21.less变量的范围是指变量存在的位置。首先在本地范围内搜索变量。如果它不在本地范围内, 则编译器将在父范围内搜索。()

答案：√

# 22.less中编译器对注释的反应不算什么。在编译代码时, 编译器将忽略注释。()

答案：√

# 23.Less文档中允许两种类型的注释：内联注释和块注释。但是, 如果你在代码中编写内联注释, 则编译器将忽略它, 但会将该代码添加到CSS文件中。仅将块注释添加到CSS文件中。 ()

答案：×

# 24.less编程中，导入用于将css或less文档导入less文档()

答案：√

# 25.less中，extend是less伪类，用于在一个选择器中选则另一个选择器样式()

答案：√

# 26.less中提供的不同类型功能有杂项功能、字符串函数和清单功能()

答案：√

# 27.less中提供的不同类型功能有数学功能、类型功能和颜色定义功能 ()

答案：√

# 28.less中提供的不同类型功能有色彩通道功能、色彩操作和色彩融合功能()

答案：√

# 29.使用less代码嵌套的时候，习惯最多嵌套3层()

答案：√

# 30.由于less使用嵌套，使得代码更短、更干净，并以特定方式组织()

答案：√

# 31.由于less使用了变量，所以更难维护（）

答案：X

# 32.less提供了合并属性()

答案：√

# 33.Less是用JavaScript编写的，它可以比CSS的其他预处理器更快地编译()

答案：√

## 填空题

# 1.less代码为@width:200px; div{width:@width},编译后div{width:\_\_\_\_\_\_\_\_}

答案：200px**；**

# 2.less代码为@height:50px; div{width:@height},编译后div{width:\_\_\_\_\_\_\_\_}。

答案：50px**；**

# 3.less中声明变量是以\_\_\_\_\_\_\_\_\_开头

答案：@**；**

# 4.less中的选择器可以使用变量，变量需要用\_\_\_\_\_\_\_\_\_\_包括。

答案：@{}**；**

# 5.less样式顶部声明变量后，可以在属性名称使用\_\_\_\_\_\_\_\_\_\_\_;

答案：变量**；**

# 6.媒less中的变量值为url时，需要将url加\_\_\_\_\_\_\_\_\_\_\_。

答案：引号**；**

# 7.媒less中的mixins可以用于引入\_\_\_\_\_\_\_\_\_\_\_代码块。

答案：公共**；**

# 8.媒less中的mixins的声明结构为\_\_\_\_\_\_\_\_\_\_\_。

答案：@name：{属性：值}**；**

# 9.媒less中的mixins声明的结构如何使用\_\_\_\_\_\_\_\_\_\_\_。

答案：@name（）**；**

## 简答题

# 1.Less支持操作运算吗？。

答案：是的，Less支持各种算术运算，例如+、？、\*、/，可以对任何颜色或变量进行操作。

# 2.在"Less"文档中可以进行多少种注释？。

答案：Less文档种允许两种类型的注释：内联注释和块注释。

但是，如果在代码种编写内联注释，则编译器将忽略它，并且不会将改代码添加到css文件中，仅将代码块注释添加到css文件中。

# 3.如何在Less中表示变量？

答案：需要使用@符号声明变量，并且使用（:）来分配变量中的特定值，还必须在变量的值之后添加分号（；）

# 4.嵌套在Less中起什么作用？

答案：嵌套用于允许代码遵循某些视觉层次结构，从而使代码更简单、更简洁。

# 5.我们使用后Less有多少种方式。

答案：有三种使用方式：

通过命令行

通过第三方工具

可以下载Less作为浏览器的脚本文件。

# 6.rem和em的区别？

答案：rem的特点：

rem的大小是根据html根目录下的字体大小进行计算的。

当我们改变根目录下的字大小的时候，下面的字体都改变属性。

rem不仅可以设置字体的大小，也可以设置元素的宽、高等属性。

em的特点：

字体大小是根据父元素大小设置的。

# 7.rem实现适配的原理？

答案：核心思想：百分比布局可实现相应式布局，而rem相当于百分比布局。

实现手段：动态获取当前视口宽度width，除以一个固定的数n，得到rem的值。

表达式为rem = width/n。

通过此方法，rem大小始终为width的n等分。

# 8.rem布局的优点？

答案：

在屏幕分辨率千差万别的时代，只要将rem与屏幕分辨率关联起来就可以实现页面的整体缩放，使得在设备上的展现都统一起来了。

而且现在浏览器基本都已经支持rem了，兼容性也非常好。

# 代码题

# 1.在代码中调用less编译器（Node平台）

var less = require('less');

less.render('.class{width:(1+1)}',function(e,css){

console.log(css)

})

输出的结果为：

.class{

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

答案：width:2;

# 2.使用less编写代码为

@width：100px;

@height:@width+50px;

#header {

width:@width;

height:@height

}

编译为：

#header{

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

答案：width:100px; height:150px;

# 3.用less编写的样式代码为

div{

.demo{

width:100%;

height:50px;

}

}

编译后的代码为：

\_\_\_\_\_\_\_\_\_\_\_{

width:100%;

height:50px;

}

答案：div .demo

# 4.less中将bordered属性规则混入到a标签中

.bordered{

border-top: dotted 1px black;

border-bottom: solid 2px baack;

}

#menu a{

color: #111;

\_\_\_\_\_\_\_\_\_\_\_\_;

}

.post a{

color:red;

\_\_\_\_\_\_\_\_\_\_

}

答案：. bordered() . bordered()

# 5.less编写的样式代码为：

.clearfix{

display:block;

zoom:1;

&:after{

content: " ";

display:block;

font-size:0;

height:0;

clear:both;

visibility:hidden;

}

}

编译后的代码为：

. clearfix{

display:block;

zoom:1;

}

\_\_\_\_\_\_\_\_\_\_{

content: " ";

display:block;

font-size:0;

height:0;

clear:both;

visibility:hidden;

}

答案：.clearfix:after

# 6.less样式代码为：

.component{

width:300px;

@media (min-width:768px){

width:600px;

@media (min-resolution:192dpi){

background-image:url(/img/retina2x/png);

}

}

@media(min-width:1280px){

width:800px;

}

}

编译为：

. component{

width:300px

}

@media(min-width:768px){

.component{

width:600px;

}

}

@media(min-width:768px) and (min-resolution:192dpi){

.component{

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

}

@media (min-width:1280px){

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

}

答案：background-image:url(/img/retina2x/png);

width:800px;

# 7.下面的calc()的代码

@var: 50vh/2;

width: calc(50% + (@var - 20px))

width结果是：

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

答案：calc(50%+25vh-10px)

# 8.如下代码

@min768:~"(min-width: 768px)";

.element {

@media @min768{

font-size:1.2rem

}

}

编译后为：

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_{

.element{

font-size:1.2rem

}

}

答案：min-width: 768px

# 9.less中代码为：

@var：red;

#page{

@var :white:

#header{

color:@var;

}

}

编译后代码为：

#page #header{

color:\_\_\_\_\_\_\_\_\_\_\_\_;

}

答案：white；

# 10.需要将一个div类名为group的设置flex布局，代码为

.group{

width:100%;

height:50px;

background-color:blue;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_；

}

答案：display：flex