Web 开发之rem 适配布局

[1.rem基础 2](#_Toc45296307)

[2.媒体查询 2](#_Toc45296308)

[2.1 定义 2](#_Toc45296309)

[2.2 语法规范 3](#_Toc45296310)

[2.3 媒体查询 + rem 实现元素动态大小变化 4](#_Toc45296311)

[2.4 引入资源（理解） 5](#_Toc45296312)

[3. less基础 5](#_Toc45296313)

[3.1 维护css的弊端 5](#_Toc45296314)

[3.2 less 介绍 6](#_Toc45296315)

[3.3 less安装 6](#_Toc45296316)

[3.4 less使用 6](#_Toc45296317)

[4. rem 适配方案 7](#_Toc45296318)

[5. 苏宁首页案例制作 9](#_Toc45296319)

[6，rem 适配方案2 10](#_Toc45296320)

[6.1 简洁高效的rem适配方案flexible.js 10](#_Toc45296321)

[6.2 使用适配方案2 制作苏宁移动端首页 10](#_Toc45296322)

[1. 技术选型 10](#_Toc45296323)

[2. 搭建相关文件夹结构 11](#_Toc45296324)

[3. 设置视口标签以及引入初始化样式还有js文件 11](#_Toc45296325)

[4. body样式 11](#_Toc45296326)

[6.3 VSCode px 转换rem 插件 cssrem 11](#_Toc45296327)

rem 基础 媒体查询 Less 基础 rem 适配方案 苏宁首页案例制

能够掌握 **rem 单位**

能够掌握**媒体查询**的基本语法

能够掌握 Less 基本语法

能够掌握在 Less 中的嵌套使用

能够掌握在 Less 中的运算使用

能够掌握 2 种 rem 适配方案

能够独立完成**苏宁移动端**首页

方案？

1. 以前的**页面文字是不是随着屏幕大小变**化？
2. 流式布局和flex布局只要针对于宽度布局，那高度如何设置？
3. 怎么样让屏幕发生变化的时候元素**高度和宽度等比例缩放**？

# 1.rem基础

Rem单位：（root em）是一个相对单位，类似于em, em是**父元素字体大小**

不同的是**rem是基准层相对于html元素的字体大小**。比如，根元素（html）设置font-size=12px; 非根元素设置width：2rem; 则换成px表示就是**24px**;

=🡺 通过页面大小**整体控制字体全体的大小**，rem的优点就是可以通过修改HTML里边文字大小来改变页面中元素的大小可以整体控制

# 2.媒体查询

## 2.1 定义

**Media query** 是css3的新语法

使用@media媒体查询，可以针对不同的媒体类型定义不同的样式

@media可以针对不同的屏幕尺寸设置不同的样式

当你重置浏览器大小的过程中，页面也会根据浏览器宽高重新渲染页面

目前针对苹果手机，安卓，平板等设备都用得到媒体查询

## 2.2 语法规范

|  |
| --- |
| @media mediatype and | not | only (media features) {  css-code;  } |
| * @media 开头 * Mediatype 媒体类型 * 关键字： and not only * media features 媒体特性，比必须有小括号包含 |

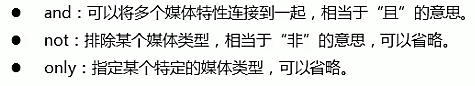
1. mediatype 查询类型：将**不同设备**划分成**不同的类型**，称为媒体类型

all : 用于所有设备

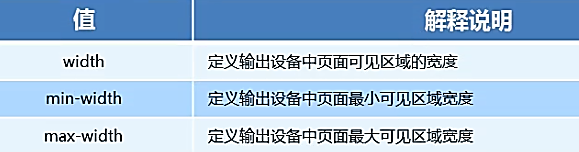
print: 用于打印机

**scree：用于电脑屏幕，平板电脑，智能手机屏幕（用的多）**

1. 关键字：媒体类型/多个媒体类型连接到一起做为媒体查询的条件



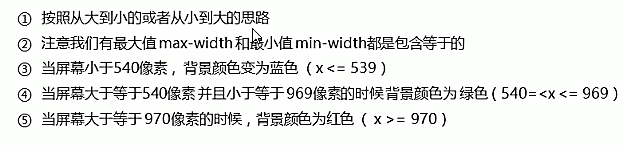
1. 媒体特性：每种媒体类型都有自己的特性，根据不同媒体类型的媒体特性设置不同的展示风格，



|  |
| --- |
| //在屏幕上，并且最大宽度是800想素=小于等于800px,屏幕才变色, 然后设置样式  @media screen and (max-width:800px){  body{  background-color:red;  }  } |

【案例】根据页面宽度改变背景颜色

【实现思路】：



|  |  |
| --- | --- |
| <style>          /\* 从大到小 / 从小到大          1，选择从小到大（一般）          2，<540， 变蓝; 尽量不要包含540，第二章>=540           \*/          @media screen and (max-width: 539px){              body{                  background-color: blue;              }          }          /\* 3. 540-970 绿色 \*/          @media screen and (min-width:540px) and (max-width: 969px){              body{                  background-color: green;              }          }          /\* 4.> = 970  red  \*/          @media screen and (min-width:970px){              body{                  background-color: red;              }          }      </style> | 优化代码：  <style>          /\* 从大到小 / 从小到大          1，选择从小到大          2，<540， 变蓝; 尽量不要包含540，第二章>=540 \*/          @media screen and (max-width: 539px){              body{                  background-color: blue;              }          }          /\* 3. 540-970 绿色 \*/          @media screen and (min-width:540px){              body{                  background-color: green;              }          }          /\* 4.> = 970  red  \*/          @media screen and (min-width:970px){              body{                  background-color: red;              }          }      </style> |

## 2.3 媒体查询 + rem 实现元素动态大小变化

Rem 单位是跟着html 来的，有了rem 页面元素可以设置不同大小的尺寸

媒体查询可以根据不同的设备宽度来修改样式

**媒体查询+rem就可以实现不同设备宽度**，实现页面元素大小的动态变化

【案例】：根据页面大小实现元素变化：320 + 640

|  |
| --- |
| <style>          /\* .html{              font-size: 100px;          } \*/          /\* 从小到大顺寻 \*/          @media screen and (min-width:320px){              html{                  font-size: 50px;              }          }          @media screen and (min-width:640px){              html{                  font-size: 100px;              }          }          .top{              height: 1rem;              font-size: .5rem;              background-color: cadetblue;              color: white;              text-align: center;              /\* 行高 = 高： 文字垂直居中 \*/              line-height: 1rem;          }      </style>  </head>  <body>      <div class="top">shopping cart</div>  </body> |

## 2.4 引入资源（理解）

屏大，屏小的时候，样式文件不一样

当样式比较多，针对不同媒体 使用不同的css文件

原理：在link中判断设备的尺寸，然后引入不同css-文件

<link rel="stylesheet" href="./css/style320.css" media="screen and (min-width: 320px)">

<link rel="stylesheet" href="./css/style640.css" media="screen and (min-width: 640px)">

# 3. less基础

## 3.1 维护css的弊端

Css需要大量没有逻辑的代码，冗余比较高

不方便维护，不利于复用

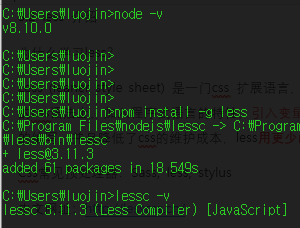
Css没有很好的计算能力

非前端开发工程师来讲，往往会因为缺少css编写经验很难写出有组织的良好还易于维护的css代码项目

## 3.2 less 介绍

为什么学习less？

Less (leaner style sheet) 是一门css 扩展语言，也成为**css处理器**。作为css的一种形式的扩展，再现有的css语法上，加入程序式语言的特性，引入变量，mixin(混入)，运算以及函数等功能，大大简化了css的编写，并且降低了css的维护成本，less**用更少的的代码做更多的事情**

Css常见预处理器：Sass, less, stylus

中文网址：<http://lesscss.cn/>

## 3.3 less安装

下载node

Node –v

npm install -g less

lessc -v

## 3.4 less使用

3.4.1 less变量

变量是没有固定的值，可以改变得，后边一些颜色和数值要经常使用

@变量名：值；

变量名规范：**@开头**，**不包含特殊字符**，**不能以数字开头，大小写敏感**

3.4.2 less编译

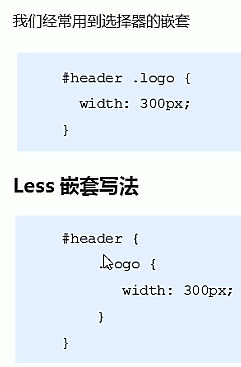
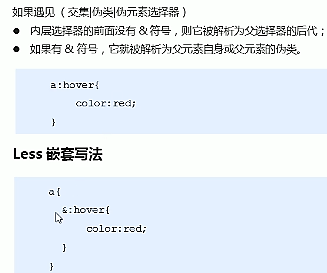
本质上，less就是一套自定义得语法及解析器，用户根据语法定义自己得样式规则，这些规则最终还是会通过解析器，编译成对应得css-文件。

Vscode less 插件 –Easy LESS

<link ref……

3.4.3 less嵌套

1. 子元素的样式直接写道父元素里边（儿子住到爸爸家里）

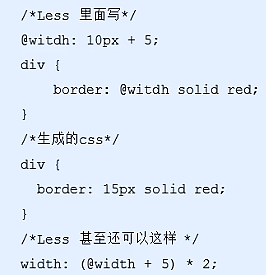
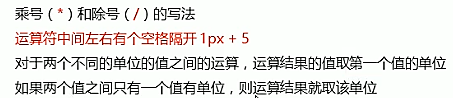
 

2.伪类/伪元素/交集选择器：hover等;

**&:** 如果没有这个符号，就属于父元素后代；如果有，就是连在一起的，给这个元素加伪元素

3.4.4 less运算 **\*\*\*\*\***

任何数字，颜色或者变量都可以运算，加减乘除

# 4. rem 适配方案

1. 适配的目标？
2. 如何达到目标？
3. 实际开发中的使用？

【答案】：不能等比自适应的元素，达到当设备尺寸发生改变的时候，等比例适配当前设备

**使用媒体查询根据不同设备比例设置html字体大小**，页面用**rem做尺寸单位**，当html字体大小变化，元素的尺寸也会发生变化，从而达到等比例缩放的适配

4.1 rem 实际开发适配方案

1. 按照**设计稿与设备宽度的比例**，动态计算并设置htm标签的font-size大小：（媒体查询）
2. Css中，设计稿元素的宽，高，相对位置等取值，按照**同比例换算为rem**为单位的值

4.2 rem 适配方案技术使用(市场主流)

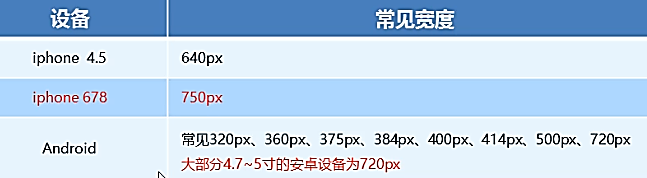


总结：两种方案都存在，第二种更加简单，无需了解里边的js代码

4.3 rem实际开发方案一

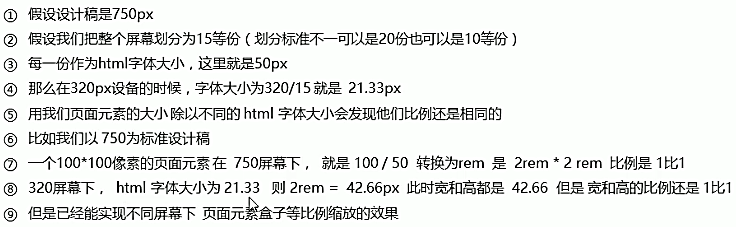
Rem + 媒体查询 + less技术

1. 设计稿常见尺寸



一般情况下，我们以一套或者两套效果图使应大部分屏幕，放弃极端屏或者对其优雅降级，牺牲一些效果。**现在基本以750**为准

1. 动态设置html标签font-size



1. 元素大小得取值

最后公式：页面元素的rem值=**页面元素值（px**）/(**屏幕宽度/划分的**份数)

屏幕宽度/划分的份数就是html font-size 的大小

或者：页面元素的rem值 = 页面元素值（px）/ html font-size字体大小

1. 首先，选一套以750为标准

我们用屏幕尺寸/划分的分数=HTML里的文字大小，但是不同屏幕文字大小不同

页面元素的rem值= 页面元素在750想素下的px值 / html里边的文字大小

# 5. 苏宁首页案例制作

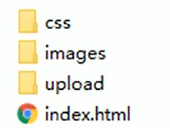
5.1 技术选型

方案：采取单独制作移动端页面方案

技术：布局采取rem 适配布局（less+rem+媒体查询）

设计图：本设计图采用750px设计尺寸

5.2 建立文件夹结构



5.3 设置视口标签以及引入初始化样式

|  |
| --- |
| <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no,maximum-scale=1.0,minimum-scale=1.0">   <meta http-equiv="X-UA-Compatible" content="ie=edge">   <link rel="stylesheet" href="css/normalize.css">   <link rel="stylesheet" href="css/index.css"> |

5.4 设置公共common.less文件

1）新建**common.less** 设置好最常见的尺寸，利用媒体查询设置不同的html字体大小，因为除了手业其他页面也需要

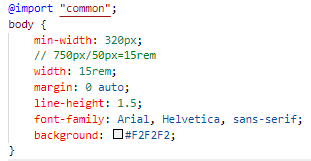
2）我们关心的尺寸有320px, 360px, 375px, 400px, 414px,424px, 480px, 540px,720px,750px

3) 划分为15等分

4）因为我们PC端可以打开苏宁移动端首页，我们默认html字体大小为50px, 注意这句话写到最上边

5.5 新建index.less文件

1）在里边写首页样式

2）将刚才设置好的common.less引入到index.less里边，语法如下：@import "common";

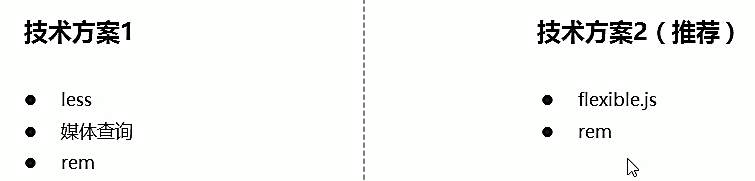
3）生成index.css 引入到html里边即可

5.6 body样式

**问题：图片显示不出来？？**

# 6，rem 适配方案2

## 6.1 简洁高效的rem适配方案flexible.js



手机淘宝团队出的简洁高效 移动端适配库

再也不用写不同屏幕的媒体查询，因为里边js做了处理

他的原理是把当前设备化成10等分，但是不同设备下，比例还是一致的

我们要做的，就是确定好我们当前设备的HTML文字大小就可以了

比如当前设计稿是750px，那么我们只需要把html文字大小设置成75px（750/10）

里边页面元素的rem值：页面元素的px值 / 75

剩余的，让flexible.js 去计算

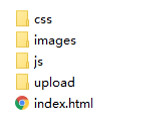
下载地址： <https://github.com/amfe/lib-flexible>

## 6.2 使用适配方案2 制作苏宁移动端首页

### 1. 技术选型

方案：我们采取**单独制作**移动页面方案

技术：布局采取**rem适配布局2**（flexible.js + rem）

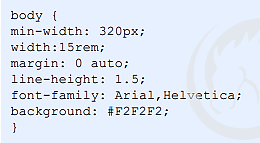
设计图： 本设计图采用 **750px** 设计尺寸

### 2. 搭建相关文件夹结构

### 3. 设置视口标签以及引入初始化样式还有js文件

|  |
| --- |
| <meta name="viewport" content="width=device-width, user-scalable=no,  initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  <link rel="stylesheet" href="css/normalize.css">  <link rel="stylesheet" href="css/index.css"> |
| 我们页面需要引入 这个js文件； 在 index.html 中 引入 flexible.js 这个文件  <script src=“js/flexible.js”> </script> |

### 4. body样式



## 6.3 VSCode px 转换rem 插件 cssrem

这是一个神奇的插件

设置html字体大小基准值 1. 打开 设置 快捷键是 **ctrl + 逗号**