移动端适配

1. **媒体查询**
2. **rem**

01. 媒体查询

作用：

> 感受到屏幕的变化；可以根据屏幕不同的宽，从而获得不同的样式，然后实现不同的样式显示；

1. CSS3 新语法，是一个查询屏幕的过程，通过查询当前屏幕尺寸属于哪个范围，从而有哪个范围的样式生效；
2. 感受屏幕变化，屏幕变化就是宽度的变化，通过预设置，当屏幕到了我已经预设置的变化的范围，就会把我提前设置好的样式进行生效；

语法：

CSS样式：

```
/*
mediatype 查询类型：
-----------------
all   所有设备
print 用于打印机和打印预览
screen 用于电脑屏幕，平板电脑，智能手机等。

条件：
----
and 并且  not 不满足  only 仅仅满足

media feature  查询条件：
----------------------
width,min-with,max-width
*/
@media mediatype and|not|only (media feature) {
    CSS-Code;
}
```

引用资源：（了解）

```
<link rel="stylesheet" media="mediatype and|not|only (media feature)" href="mystylesheet.css">
```

例子：

如果文档宽度小于 500 像素则修改背景颜色(background-color):

```
/* min-width/max-width：最小界值，最大界值；查询条件包含等于号；*/
/* 宽度的最小界值500px，大于等于500px */
@media screen and (max-width:499px) {
    body {
        background-color: red;
    }
}
```

适配需求：

- 档位1：w<320 px　　w <= 319 px;
- 档位2：w>=320px　and　w< 640px;
- 档位3：w>=640px

02. rem

rem适配原理

> rem是一个相对单位，类似于em，
>
> 不同的是rem的基准是相对于html元素的字体大小，em是父元素字体大小。

作用：让一些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。

方案：使用媒体查询根据不同设备按比例设置html的字体大小，然后页面元素使用rem做尺寸单位，当html字体大小变化元素尺寸也会发生变化，从而达到等比缩放的适配。

用法:

- rem单位，可以控制整个页面所有元素有关PX类；（宽、高、padding、margin、top...）只要是你设置数值的地方都可以实现控制；
- root：1rem代表HTML的font-size大小；
- 语法：

```
/* 1.根html 为 10px */
html {
   font-size: 10px;
}

/* 2.此时 div 的宽就是 150px */
div {
    width: 15rem;
}
```

03. rem 应用

- rem布局的核心：rem+媒体查询；
- 媒体查询：把屏幕划分不同档位，等待变化；
    - 布局所有单位用rem，**当 HTML字体大小发生改变，使用rem单位元素都会发生改变；**
    - rem+媒体查询加在一起：划分屏幕，等待变化；变化谁？变化唯一控制 rem（HTML字体大小）

- 语法：

```
@media screen and (min-width: 320px) {
  html {
    font-size: 20px;
  }
}

@media screen and (min-width: 640px) {
  html {
    font-size: 40px;
  }
}

div {
  width:1rem;
  height:1rem;
}
```

- rem布局核心总结：
  - 媒体查询：屏幕到达不同的范围下，HTML的font-size大小会有不同的取值;
  - 1rem 背后的代表的px值 = 当前档位 HTML 字体大小 所代表的大小
  - 那么使用rem单位的元素就会发生等比的变化;

04. 实际应用

UI设计稿：

- 尺寸不同，页面在不同的尺寸下要等比缩放

设备	常见宽度
iphone 4.5	640px
iphone 678	750px
Android	常见320px、360px、375px、384px、400px、414px、500px、720px 大部分4.7~5寸的安卓设备为720px

- 档位划分：min-width 最小界值;
- 约定：把UI各种设计稿，从小到大，当前我们档位划分开始点，起始点;

1rem背后代表的值：

- 在这里，**我们约定设置HTML的字体大小：按照每个档位的最小界值，都划分相同的份数，得到当前档位的HTML的font-size大小;**

```
// 我们此次定义的划分的份数 为 10 这个过程在计算 1个rem 在不同档位下是多大;
@media screen and (min-width: 320px) {

    html {
```

```
        font-size: 32px;
    }
}
@media screen and (min-width: 360px) {
    html {
        font-size: 36px;
    }
}
@media screen and (min-width: 540px) {
    html {
        font-size: 54px;
    }
}
```

具体如何做?

- 档位和根基设置按照约定设置好了;
- UI给的图的已经拿到了，上面标注的是px单位;
- 目标：把px单位换算 成rem单位;
- 如何做：选择当前设计稿在档位下的rem背后代表的值进行计算;
    - 如果给你的540px宽度设计稿怎么换算rem，宽度=300px = 300/54 rem;

06. less的使用

- less : 让你写更少的代码，实现相同的效果;
- less :是一门 CSS 扩展语言，它扩展了CSS的动态特性。 CSS 预处理言语。
- 常见的CSS预处理器：Sass、Less、Stylus 。
- Less中文网址：http://lesscss.cn/

安装

- vscode插件安装:
    - 搜索 Easy LESS
    - 安装完毕插件，重新加载下 vscode。
    - 测试：保存一下 .less 文件，会自动生成 .css 文件。

变量

- 变量是指没有固定的值，可以改变的。
- 我们CSS中的一些颜色和数值等经常使用，可以设置为变量;
- 语法:

```
//@变量名:值;
@bg:#333;
.box_1 {
  background-color: @bg;
}

.box_2 {
  background-color: @bg;
}
```

- 命名规则：
  - 必须有@为前缀
  - 不能包含特殊字符~=+、不能以数字开头
  - 大小写敏感区分；

嵌套

- 类似HTML一样写less结构；
- 语法：

```
/* css 写法 */
#header .logo {
  width: 300px;
}

/* less 写法 */
#header {
  .logo {
      width: 300px;
  }
}
```

- 交集|伪类|伪元素选择器，语法：

```
/* css写法 */
a:hover{
    color:red;
}

/* less写法 */
a{
  &:hover{
      color:red;
  }
}
```

运算

- 任何数字、颜色或者变量都可以参与运算。

- Less提供了加（+）、减（-）、乘（*）、除（/）算术运算。
- 语法：

```
// 数字
width: 200px - 50;

// 颜色
background-color: #666 - #222;

// 注意：运算符中间左右有个空格隔开 1px * 5
```

- 单位选择：
  - 如果两个值之间只有一个值有单位，则运算结果就取该单位
  - 对于两个不同的单位的值之间的运算，运算结果的值取第一个值的单位

05. 解决方案

方案1

rem+媒体查询+less 方案

750px 操作过程：

- 第一步：
  - **原稿实现：**先拿到设计稿:750px ; 页面上所有的元素，在750px设计稿上进行测量，代码实现；
- 第二步：
  - **2.1 准备各个档位下的rem** ： 提前准备好各个档位下的HTML 的font-size大小；
  - **2.2 拿到当前尺寸的rem**：因为我现在是750px的设计稿，所以可以得到750px这个尺寸属于的档位下的HTML 的font-size大小，也就是750px设计稿下的1rem值。
  - **2.3 计算比例**：把页面刚才所有的元素的PX值替换为 rem
  - **达到目标**：那么，屏变化时，1rem(基础块)也会变化，自然就是等比缩放；

方案2（推荐!!!）

rem+flexible.js+less

- 和上个方案实现原理一样，都是通过改变1rem(基础块)大小实现页面整体元素改变；

flexible.js

- 简介：手机淘宝团队出的 简洁高效 移动端适配库；**和flex布局没有任何关系**
- github地址：https://github.com/amfe/lib-flexible
- 不是通过设置CSS媒体查询设置font-size，通过 JS 设置font-size，效果是屏幕变化一点，就有一个rem重新计算；

1rem背后代表的值

- **划分10份；**
- 设置在HTML标签上；

```
function setRem () {
  // docEl.clientWidth JS获取当前屏幕的宽度
  // 除以10,得到基础块
  var rem = docEl.clientWidth / 10;
  docEl.style.fontSize = rem + 'px'
}
```

步骤！！！

- 假如设计稿430px
- 拿到UI设计稿，**原稿实现**；页面上所有的元素，在设计稿上进行测量，代码实现；（流式、flex）只要是UI给图上有标注，就是写出来；先全部实现出来，一会统一替换；
  - 在哪里写？less文件
  - 需要把生成的css文件进行引入index.html；
- 设计稿宽度/10 ： 1rem=43px；
- 统一替换：100px=100/43rem;

方案对比

- 相同：都是对font-size实现控制，1rem（基础块）变化，实现等比效果；
- 不同：
  - rem+媒体查询+less：通过设置不同的档位下，设置不同的1rem值；效果为阶梯式变化；
  - flexible.js+rem：通过js设置不同的1rem值，效果为连续变化；这个看起来更为连贯，适配任何屏幕。

苏宁易购（两种方案实现）

-