

移动Web 第五天

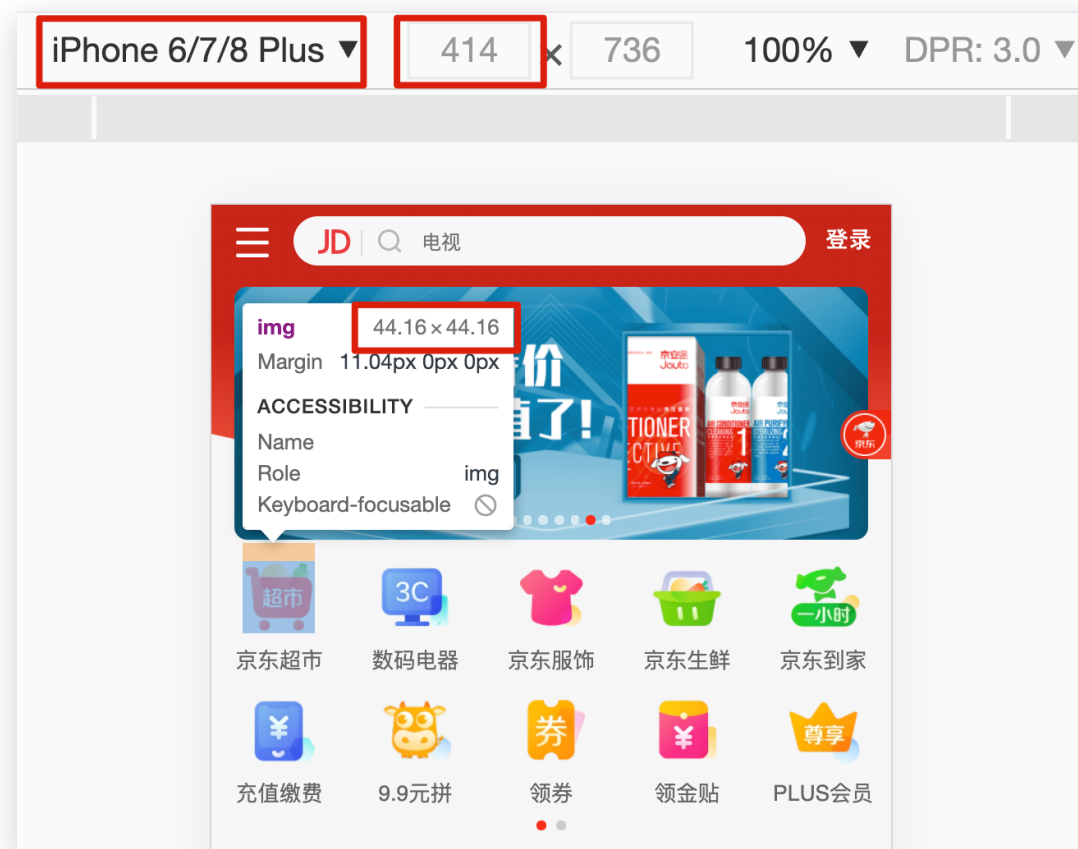
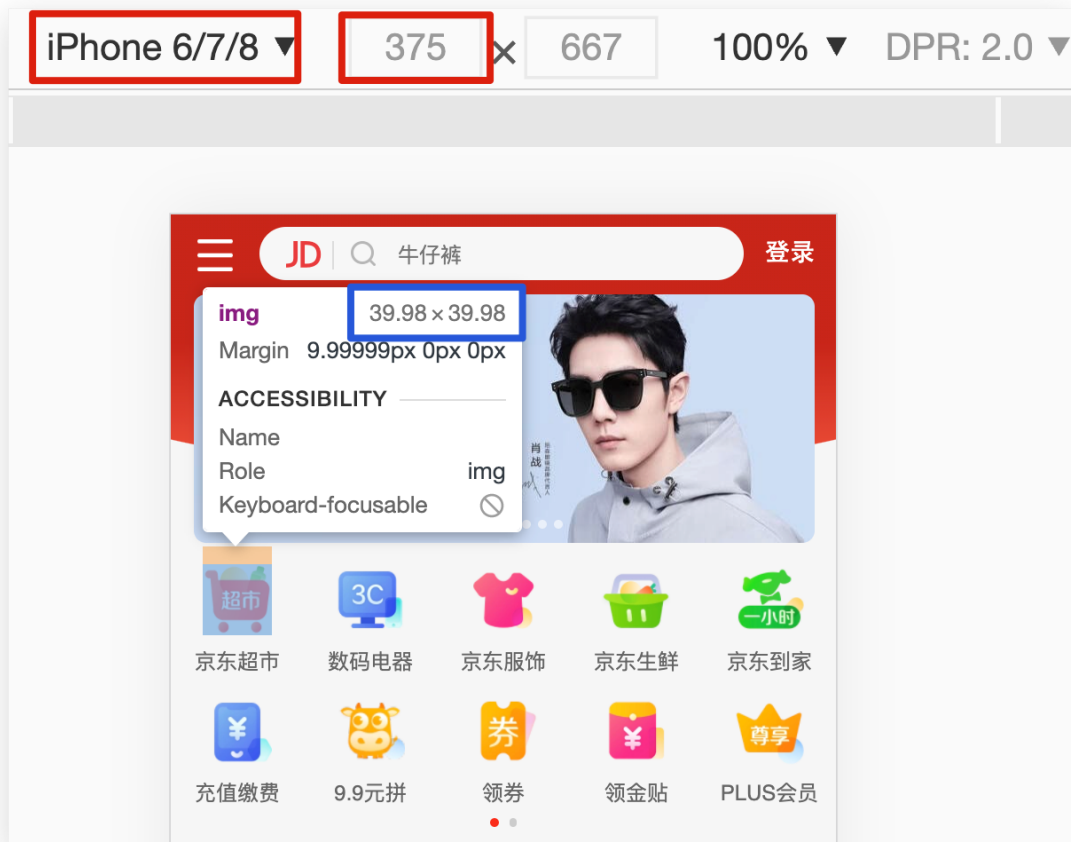
成长必经之路

移动适配

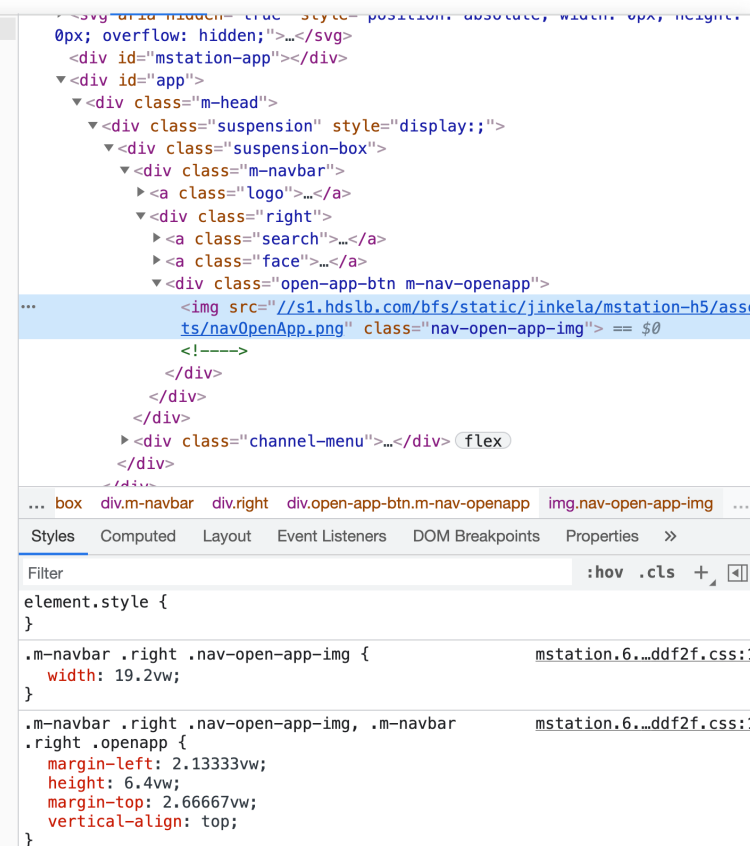
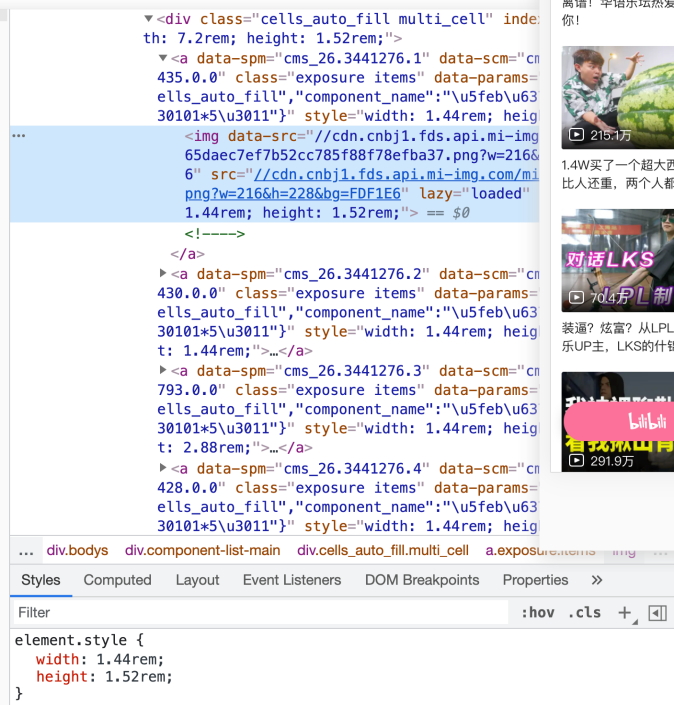


黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌



- rem : 目前多数企业在用的解决方案
- vw / vh : 未来的解决方案





目录

Contents

◆ rem

- 媒体查询
- flexible.js

◆ less

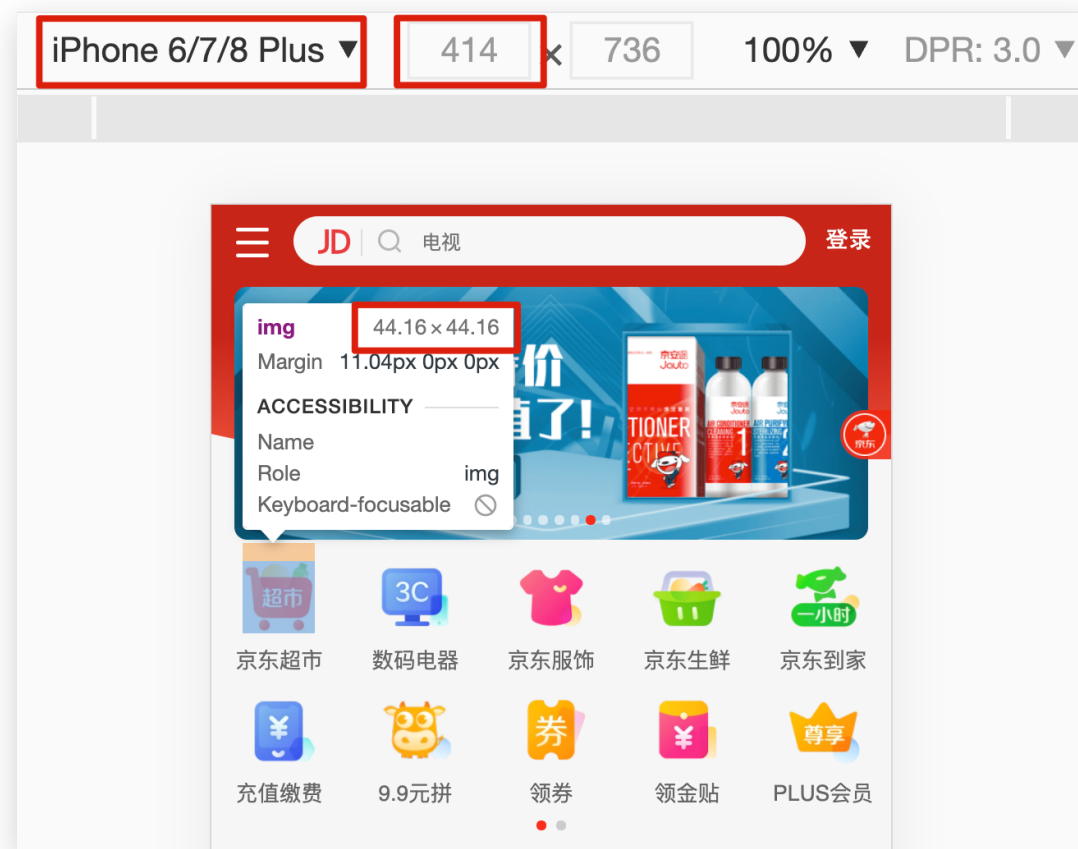
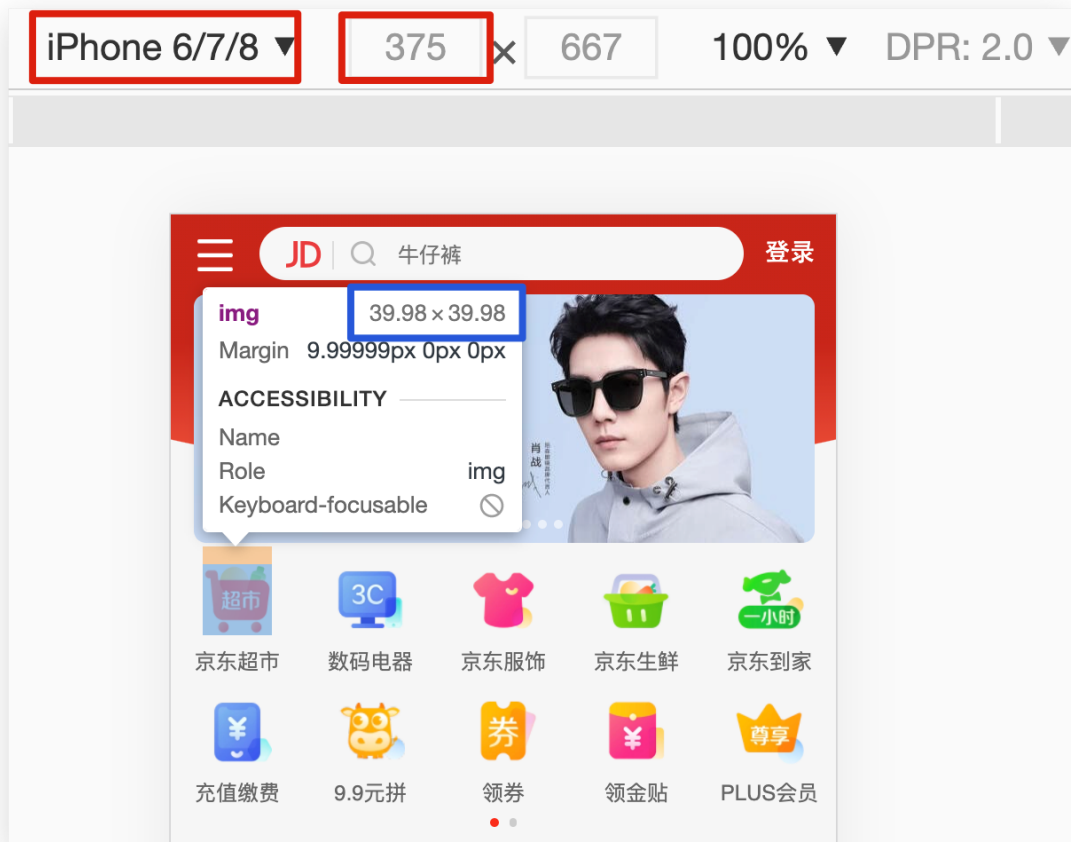
◆ 项目演练



长度单位

- rem
- vw/vh

目标：能够使用rem单位设置网页元素的尺寸



目标：能够使用rem单位设置网页元素的尺寸

- 网页效果
 - 屏幕宽度不同，网页元素尺寸不同（等比缩放）
- px单位或百分比布局可以实现吗？
 - px单位是绝对单位
 - 百分比布局特点宽度自适应，高度固定
- 适配方案
 - rem
 - vw / vh

目标：能够使用rem单位设置网页元素的尺寸

- rem单位
 - 相对单位
 - rem单位是相对于HTML标签的字号计算结果
 - $1\text{rem} = 1\text{HTML字号大小}$

目标：能够使用rem单位设置网页元素的尺寸

- 思考

- 1. 手机屏幕大小不同，分辨率不同，如何设置不同的HTML标签字号？
- 2. 设备宽度不同，HTML标签字号设置多少合适？

目标：能够使用**媒体查询**设置**差异化**CSS样式

- 思考
 - 手机屏幕大小不同，分辨率不同，如何设置不同的HTML标签字号？
 - **媒体查询**
- **媒体查询能够检测视口的宽度，然后编写差异化的 CSS 样式**
- **当某个条件成立, 执行对应的CSS样式**

目标：能够使用**媒体查询**设置**差异化**CSS样式

- 写法

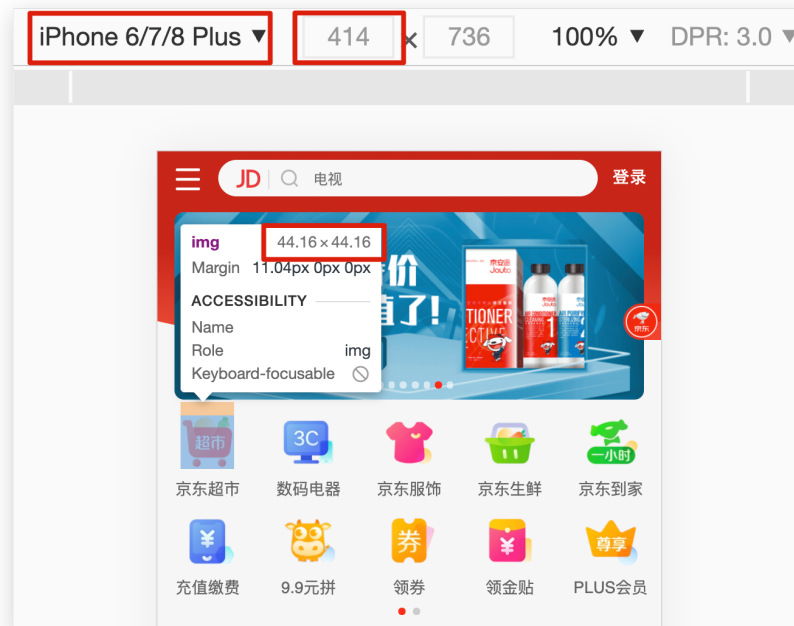
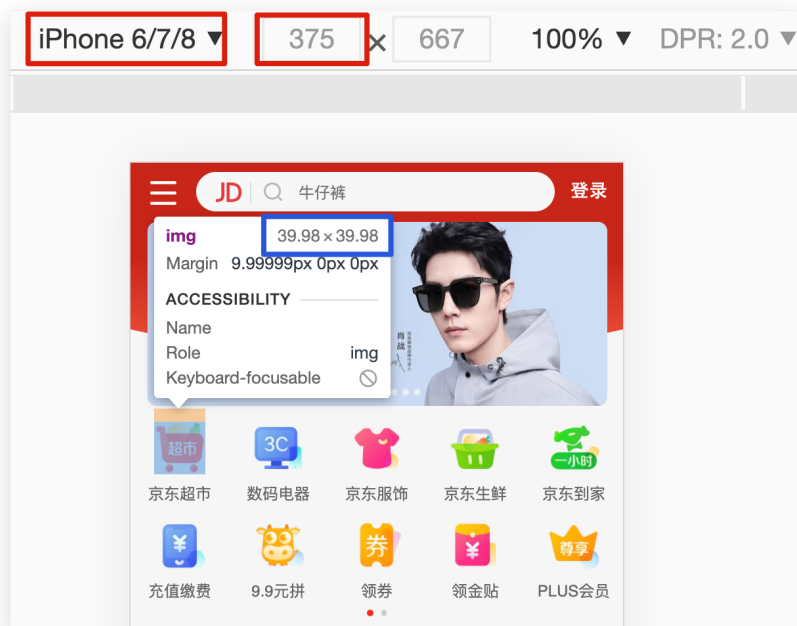
```
@media (媒体特性) {  
    选择器 {  
        CSS属性  
    }  
}
```

```
@media (width:320px) {  
    html {  
        font-size: 32px;  
    }  
}
```

目标：能够使用rem单位设置网页元素的尺寸

- 思考

- 手机屏幕大小不同，分辨率不同，如何设置不同的HTML标签字号？
- 设备宽度不同，HTML标签字号设置多少合适？
 - 设备宽度大，元素尺寸大
 - 设备宽度小，元素尺寸小



目标：能够使用rem单位设置网页元素的尺寸

- 目前rem布局方案中，将网页等分成10份，HTML标签的字号为视口宽度的 1/10

```
/* 视口宽度320px, 根字号为32px */
@media (width: 320px) {
  html {
    font-size: 32px;
  }
}

/* 视口宽度375px, 根字号为37.5px */
@media (width: 375px) {
  html {
    font-size: 37.5px;
  }
}

/* 视口宽度414px, 根字号为41.4px */
@media (width: 414px) {
  html {
    font-size: 41.4px;
  }
}
```

目标：实现在不同宽度的设备中，网页元素尺寸等比缩放效果

- 思考：
 - 工作中，书写代码的尺寸要参照设计稿尺寸，设计稿中是px还是rem？
 - 如何确定rem的数值？



目标：实现在不同宽度的设备中，网页元素尺寸等比缩放效果

- rem单位尺寸

1. 根据视口宽度，设置不同的HTML标签字号
2. 书写代码，尺寸是rem单位

- 2.1 确定设计稿对应的设备的HTML标签字号

➤ 查看设计稿宽度 → 确定参考设备宽度(视口宽度) → 确定基准根字号 (1/10视口宽度)

- 2.2 rem单位的尺寸

- $N * 37.5 = 68 \rightarrow N = 68 / 37.5$
- rem单位的尺寸 = px单位数值 / 基准根字号



目标：使用flexible js配合rem实现在不同宽度的设备中，网页元素尺寸等比缩放效果

- 思考：咱们检测了3个视口，分别设置了根字号，有什么弊端吗？
- 答：手机设备多，屏幕尺寸不一，视口不仅仅有这3个

```
/* 视口宽度320px, 根字号为32px */
@media (width: 320px) {
    html {
        font-size: 32px;
    }
}

/* 视口宽度375px, 根字号为37.5px */
@media (width: 375px) {
    html {
        font-size: 37.5px;
    }
}

/* 视口宽度414px, 根字号为41.4px */
@media (width: 414px) {
    html {
        font-size: 41.4px;
    }
}
```

目标：使用flexible js配合rem实现在不同宽度的设备中，网页元素尺寸等比缩放效果

- flexible.js是手淘开发出的一个用来适配移动端的js框架。
- 核心原理就是根据不同的视口宽度给网页中html根节点设置不同的font-size。

```
<body>
  <div class="box"></div>
  <script src="./js/flexible.js"></script>
</body>
```



Less

目标：使用Less运算写法完成px单位到rem单位的转换

- 思考：在px单位转换到rem单位过程中，哪项工作是最麻烦的？
- 答：除法运算。CSS不支持计算写法。
- 解决方案：可以通过Less实现。

目标：使用Less语法快速编译生成CSS代码

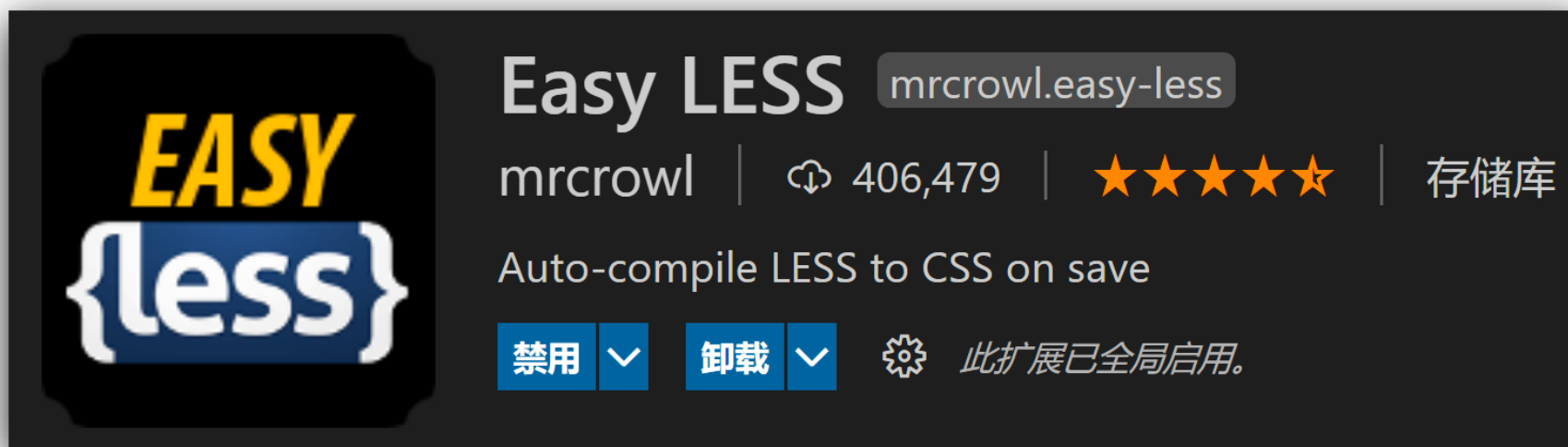
- Less是一个CSS预处理器, Less文件后缀是.less
- 扩充了 CSS 语言, 使 CSS 具备一定的逻辑性、计算能力。
- 注意：浏览器不识别Less代码，目前阶段，网页要引入对应的CSS文件。



目标：使用Less语法快速编译生成CSS代码

Easy Less：

- vscode插件
- 作用：less文件保存自动生成css文件



目标：使用Less语法快速编译生成CSS代码

注释：

- 单行注释
 - 语法：`// 注释内容`
 - 快捷键：`ctrl + /`
- 块注释
 - 语法：`/* 注释内容 */`
 - 快捷键：`shift + alt + A`

目标：使用Less运算写法完成px单位到rem单位的转换

运算：

- 加、减、乘直接书写计算表达式
- 除法需要添加 小括号 或 .
- 注意：
 - 表达式存在多个单位以第一个单位为准

```
.box {  
  width: 100 + 50px;  
  height: 5 * 32px;  
  
  width: (100 / 4px);  
  height: 100 ./ 4px;  
}
```

目标：能够使用Less嵌套写法生成后代选择器

嵌套：

- 思考：书写CSS选择器时，如何避免样式冲突？

```
.xtx_body_aside .user_meta .avatar {  
    width: 85px;  
    height: 85px;  
    margin: 0 auto;  
    border-radius: 50%;  
    overflow: hidden;  
}  
  
.xtx_body_aside .user_meta h4 {  
    font-size: 18px;  
    font-weight: 500;  
    margin: 14px 0 12px;  
}
```

目标：能够使用Less嵌套写法生成后代选择器

嵌套：

- 作用：快速生成后代选择器。

- 语法：

```
.父级选择器 {  
  // 父级样式  
  .子级选择器 {  
    // 子级样式  
  }  
}
```



```
.father {  
  color: ■ red;  
  .son {  
    width: 100px;  
  }  
}
```



```
.father {  
  color: ■ red;  
}  
.father .son {  
  width: 100px;  
}
```

- 注意：&不生成后代选择器，表示当前选择器，通常配合伪类或伪元素使用

```
.father {  
  color: ■ red;  
  &:hover {  
    color: ■ green;  
  }  
}
```



```
.father {  
  color: ■ red;  
}  
.father:hover {  
  color: ■ green;  
}
```

目标：能够使用Less变量设置属性值

变量

● 思考：

- 网页中, 文字颜色基本都是统一的，如果网站改版，变换文字颜色，如何修改代码？
- 方法一：逐一修改属性值（太繁琐）
- 方法二：把颜色提前存储到一个容器，设置属性值为这个容器名

```
.box {  
  color: pink;  
}  
  
a {  
  color: pink;  
}  
  
p {  
  color: pink;  
}
```

目标：能够使用Less变量设置属性值

变量

- 方法二：把颜色提前存储到一个容器，设置属性值为这个容器名
- 变量：存储数据，方便使用和修改。
- 语法：
 - 定义变量：**@变量名: 值;**
 - 使用变量：**CSS属性: @变量名;**

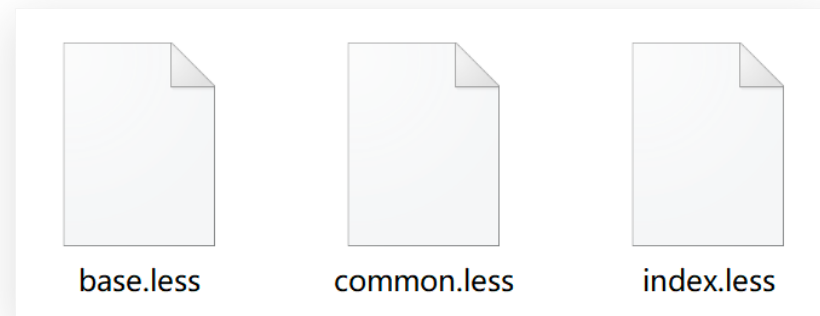
```
// 1. 定义
@fontColor: pink;

// 2. 使用
.box {
    color: @fontColor;
}

a {
    color: @fontColor;
}
```

目标：能够使用Less导入写法引用其他Less文件

- 思考：
 - 开发网站时，网页如何引入公共样式？
 - CSS：书写link标签
 - Less：导入
- 导入: `@import` “文件路径”；



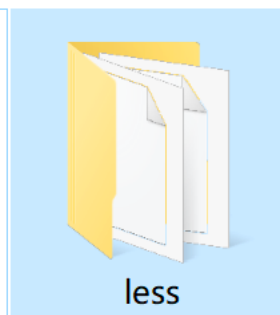
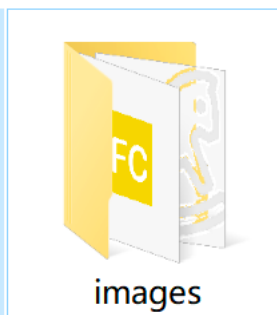
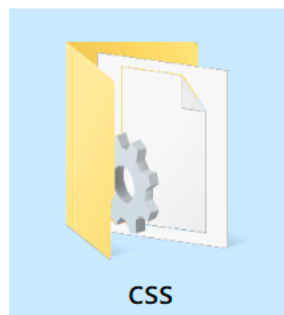
```
// 如果是less文件， 可以省略后缀.less
@import 'base.less';
@import '01-体验less';
```

目标：使用Less语法导出CSS文件

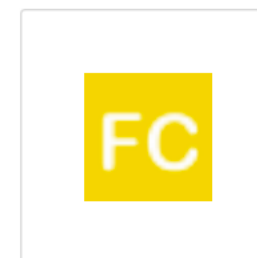
- 思考：

- 目前，Less文件导出的CSS文件位置是哪里？

- 01-体验less.css
- 01-体验less.less
- 02-注释.css
- 02-注释.less
- 03-嵌套.css
- 03-嵌套.less
- 04-计算.css
- 04-计算.less



uploads



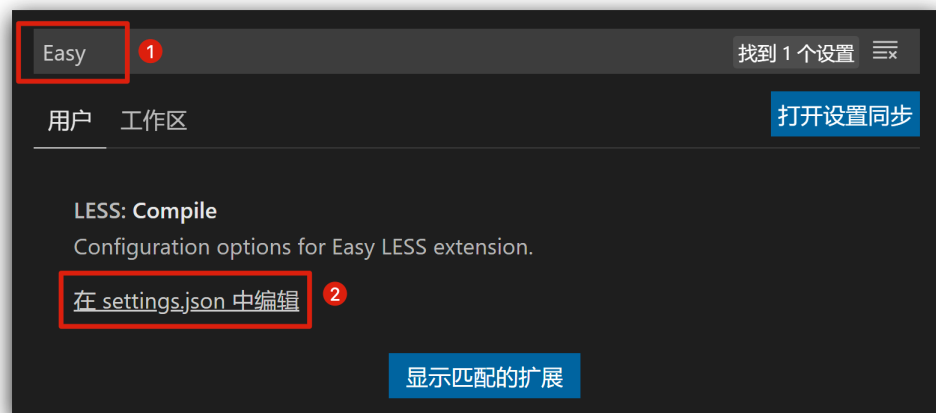
favicon.ico



index.html

目标：使用Less语法导出CSS文件

- 方法一：
 - 配置EasyLess插件，实现所有Less有相同的导出路径
- 配置插件：设置 → 搜索EasyLess → 在setting.json中编辑 → 添加代码（注意，必须是双引号）



```
{  
  "window.zoomLevel": 2,  
  "editor.minimap.enabled": false,  
  "workbench.statusBar.visible": false,  
  "editor.suggest.snippetsPreventQuickSuggestions": false,  
  "explorer.confirmDelete": false,  
  "less.compile": {  
    "out": "../css/"  
  }  
}
```

目标：使用Less语法导出CSS文件

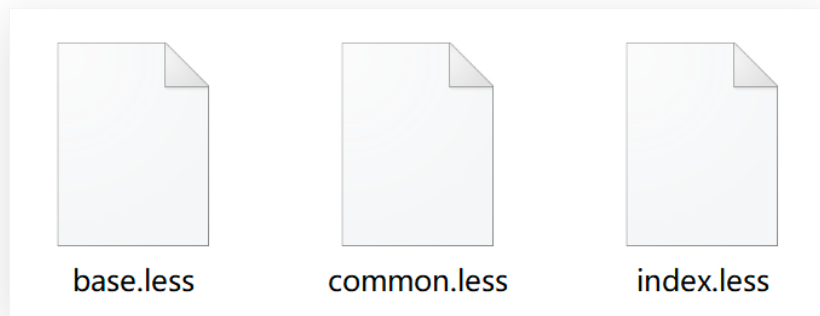
- 方法二：控制当前Less文件导出路径
 - Less文件第一行添加如下代码, 注意文件夹名称后面添加 /

```
1 // out: ./css/
```

```
1 // out: ./css/common.css
```

目标：使用Less语法导出CSS文件

- 思考：所有的Less文件都需要导出CSS文件吗？



- 禁止导出
 - 在less文件第一行添加: `// out: false`

```
1 // out:false
```



实战演练

- 游乐园项目

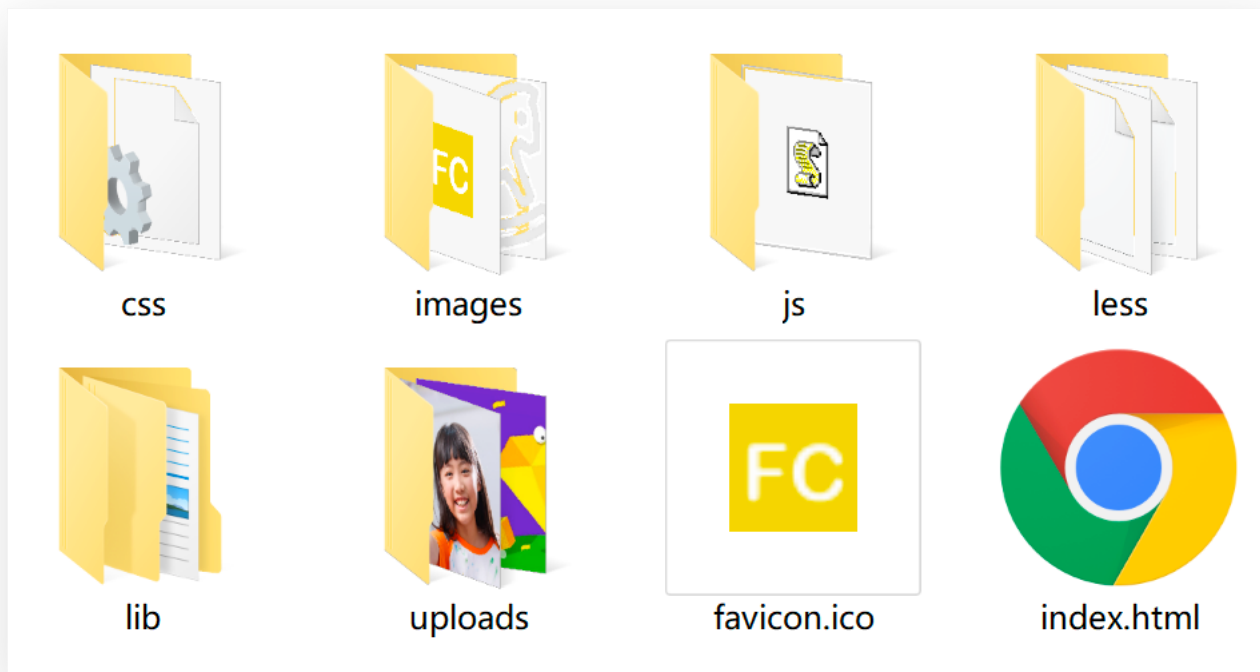
目标：实现在不同宽度设备中等比缩放的网页效果



目标：实现在不同宽度设备中等比缩放的网页效果

准备工作：项目目录及文件

- 根目录 (paradise / FC)



目标：实现在不同宽度设备中等比缩放的网页效果

整体布局：网页为灰色背景色

- 主体内容
 - 底部间距（高度与工具栏相同）
- 底部工具栏
 - 固定定位
 - 尺寸
 - 背景色

注意：需要设置基准根字号 @rootSize: 37.5 ;



四. 今日复习路线

1. 晚自习回来每个同学先必须xmind梳理今日知识点 (md 笔记也行)
2. 晚上8点是做测试题 (测试时间最多30分钟)
3. 做完考题的同学开始：独立书写今日综合案例 (关键代码要有注释)
游乐园
4. 做每日反馈，复盘知识吸收程度
5. 预习下次课程内容
6. 每日一句鼓励自己的话：

学在苦中求， 艺在勤中练



传智教育旗下高端IT教育品牌