

15 项目的建立

更新时间：2019-07-04 11:03:13



能够生存下来的物种,并不是那些最强壮的,也不是那些最聪明的,而是那些对变化作出快速反应的。

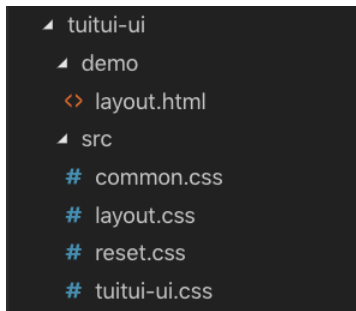
——达尔文

项目目录结构

接下来我们要做的就是建立好项目的目录结构。一个项目的目录层级尽量扁平，不宜太深。我们这个项目根目录就是项目名称“tuitui-ui”。根目录下面把样式库的核心CSS代码放在src目录下，里面包含主文件、公共样式和其他组件的样式文件。为了演示效果，我们在根目录下还会建立一个demo目录，来存放各个组件的演示代码。在后面还会在根目录下追加存放目标代码的dist目录和实战页面的page目录，这两个目录等用到的时候再建，现在可以按着下面的目录结构建立对应的文件夹和文件。

```
tuitui-ui      (推荐UI根目录)
├── src        (核心样式代码)
│   ├── tuitui-ui.css (推荐UI的主样式文件)
│   ├── common.css   (公共样式文件)
│   ├── reset.css     (CSS-Reset文件)
│   ├── layout.css    (组件文件：图标样式)
│   └── ...          (其他组件文件)
├── demo       (演示文件目录)
│   ├── layout.html  (布局样式文件)
│   └── ...          (其他组件的样式文件)
```

建立好的文件目录结构如下：



基础文件的建立

在刚才建好的目录结构里，我们要先完成几个基础的文件。

一、/src/reset.css文件

reset.css文件用来存放CSS-Reset的内容。我们在第二章里讲过，CSS-Reset的作用是覆盖浏览器的默认样式，从而排除浏览器默认样式的影响。在那一节也给出了一份需要reset的样式，这里我们可以直接把之前给出的CSS-Reset文件直接放在这个reset.css文件里：

reset.css文件内容：

```
/*
 * @Author: Rosen
 * @Date: 2019-04-12 10:06:29
 * @Last Modified by: Rosen
 * @Last Modified time: 2019-04-29 16:37:40
 */

/* 去掉所有元素的内外边距 */
html, body, div, span,
h1, h2, h3, h4, h5, h6, p, pre,
a, img, ul, li, form, label, input,
table, tbody, tfoot, thead, tr, th, td,
audio, video {
  margin: 0;
  padding: 0;
}
/* 统一全局字体 */
body {
  font-family: -apple-system-font, BlinkMacSystemFont, "Helvetica Neue", "PingFang SC", "Hiragino Sans GB", "Microsoft YaHei UI", "Microsoft YaHei", Arial, sans-serif;
}
/* 列表元素去掉默认的列表样式 */
ol, ul {
  list-style: none;
}
/* Table元素的边框折叠 */
table {
  border-collapse: collapse;
  border-spacing: 0;
}
/* 去掉默认的下划线 */
a {
  text-decoration: none;
}
/* 去掉input自带的边缘效果和背景颜色 */
input {
  outline: none;
  background: none;
}
```

在这份Reset文件里，每一条样式的作用都用注释进行了标注。其中要注意的是全局的字体，这里给body用font-family属性加了默认的字体。在运行的时候，浏览器会按着font-family所指定的字体从左到右依次在系统里查找，直到找到操作系统里存在的字体，然后把它作为网页使用的字体。如果找到最后一个都没有找到，就会使用浏览器的默认字体。

二、/src/common.css文件

第二个要建立的是common.css，这个文件用来存放项目里的公共的样式，像控制全局布局的，或者用的比较多的公共样式，都可以放在这个文件中。我们这里先把这个文件建立出来，并放上最基础的样式，到后面有需要提炼出来的公共样式，随时可以加在这个文件里。这一节我们要在common.css文件里处理全局的适配问题，这是一组全局的样式并且和业务关联不大，所以会把全局适配用的代码放在公共部分。

之前在第二章讲过经常使用的四种适配方式，包括百分比适配、多套样式分段适配、rem方式适配和弹性布局这几种方式，介绍了这四种适配方式的特点以及它们的优缺点。下来我们就一起看看怎么做一个移动端项目的适配。

1. 区分屏幕宽度的方式

适配中最先考虑的是要区分页面宽度，对不同的宽度做适配，在比较小的屏幕上把元素设置的小一些可以多显示一些内容；而在大一些的屏幕上把元素设置的也大一些，可以让内容更易读。在做宽度适配的时候有两种方式，一种是把宽度划分为几个区间，在每个区间里对需要适配的元素指定一个大小；另一种方式是根据屏幕宽度按比例计算元素的大小。这两种方式都没有什么致命的缺点，在实际项目中也都有在用的。但是相比之下，分区间适配在测试的时候会相对简单些，每个宽度区间取一个代表值来测试就可以了，而按比例适配的话不可能测试到每个尺寸，但也要把市面上常用的设备都测试一下。我们为了简便起见，准备采用分区间适配的方式。在做这个区间的分隔时，就可以使用CSS3中媒体查询的方式了。

@ Tips:

媒体查询是CSS3中的一个功能，它可以在不同类型的设备和不同的设备尺寸的情况下使用不同的样式。媒体查询的语法如下：

@media 设备类型 and 设备尺寸 { 样式部分 }

这条语句中的三个可设置的变量如下：

设备类型是指用的什么设备，这个值最常采用的是screen（屏幕设备，幕设如手机电脑等都算屏备）和print（打印类型的设备，打印机等），而其他的像braille（盲文）、handheld（手持设备）和projection（项目演示）之类的取值基本很难用到。另外这个字段还可以使用all来代替所有设备类型，不写这个字段的话默认就是指定所有设备类型。

设备尺寸是指屏幕的宽度，这个字段的取值就是适配的关键了。它使用“max-width”表示小于的关系，使用“min-width”表示大于的关系，通过用这两个值做配合，就可以给宽度做不同的分区了。

样式部分和普通的CSS样式没有什么区别，如果某个设备复合这条媒体查询的规则，那么这条规则下的样式就会生效，如果样式优先级够高就会覆盖其他的样式从而达到适配的目的。

在这个项目里，我们直接忽略“设备类型”字段，默认对所有媒体都生效。然后把宽度分成三个区间，340px到410px的宽度内的屏幕定为基准区间（以iPhone6/7/8屏幕的375px作为基准，向两边浮动35px就是这个基准区间），而另外两个区间就是小于340px的部分和大于410px的部分，用代码表示出来就是：

```
/* 屏幕宽度小于340px时的样式 */
@media (max-width: 340px){ /* 样式1 */}

/* 屏幕宽度在340px至410px时的样式 */
@media (min-width: 340px) and (max-width: 410px){ /* 样式2 */}

/* 屏幕宽度大于410px时的样式 */
@media (min-width: 410px){ /* 样式3 */}
```

因为媒体查询里的样式生效的原理是样式的覆盖，所以上面三条语句中可以做一下精简，用一个区间内的样式当做基础样式，这里我们就可以把样式2拿出来。这样做的好处除了能精简代码外，在对媒体查询支持不好的设备上也有一个基础的样式可以用。改造后就可以变成：

```
/* 屏幕宽度在340px至410px时的样式 */
/* 样式2 */

/* 屏幕宽度小于340px时的样式 */
@media (max-width: 340px){ /* 样式1 */}

/* 屏幕宽度大于410px时的样式 */
@media (min-width: 410px){ /* 样式3 */}
```

这样屏幕适配的框架就完成了

2. 元素尺寸的指定

确定好整体的适配方式后，就可以对每个宽度区间里的元素做适配了。我们在指定的时候可以用px，那么每个区间的所有元素都需要适配一遍。当然也可以使用之前介绍过的rem单位来确定所有尺寸，这样只要控制不同区间下html元素的“font-size”就可以达到适配的效果，也就是在屏幕适配部分讲到的rem方式适配。

这里我们要在刚才媒体查询的基础上，指定html元素的“font-size”属性。在基准区间内，我们使用20px作为“font-size”的值，在小屏幕上使用18px，在大屏幕上使用22px。用代码表示就是：

```
/* 屏幕宽度在340px至410px时，基准尺寸使用20px */
html{
  font-size: 20px;
}
/* 屏幕宽度小于340px时，基准尺寸使用18px */
@media (max-width: 340px){
  html{
    font-size: 18px;
  }
}
/* 屏幕宽度大于410px时，基准尺寸使用22px */
@media (min-width: 410px){
  html{
    font-size: 22px;
  }
}
```

3. 按比例划分屏幕

最后还有一类问题，就是希望几个元素平分屏幕宽度的情况。这种情况下，无论px还是rem都无法准确给元素设定宽度。所以就会用到另外两种适配方式，一个是使用百分比，另一种方式就是使用弹性布局。在平分屏幕宽度的需求中，弹性布局可以覆盖百分比的功能，所以在后面的开发中Flex弹性布局会用的多一些，有些比较简单的情况也会使用百分比来适配。

平分屏幕的情况一般也都是在组件中才会用到。所以我们这里只做一下介绍，不需要在common.css文件中加入什么样式。

最后，把刚才第二步得出的代码加上文件头注释，最终的common.css的内容会如下：

```
/*
 * @Author: Rosen
 * @Date: 2019-06-20 22:16:01
 * @Last Modified by: Rosen
 * @Last Modified time: 2019-06-20 23:03:17
 */

/* 屏幕宽度在340px至410px时，基准尺寸使用20px */
html{
  font-size: 20px;
}
/* 屏幕宽度小于340px时，基准尺寸使用18px */
@media (max-width: 340px){
  html{
    font-size: 18px;
  }
}
/* 屏幕宽度大于410px时，基准尺寸使用22px */
@media (min-width: 410px){
  html{
    font-size: 22px;
  }
}
```

三、/src/tuitui-ui.css文件

最后一个要做的文件就是这个框架的主文件，以后使用框架的时候都会通过这个文件来引用所有的样式，所以这个文件的功能就是集成其他的样式。目前，我们只有/src/reset.css和/src/common.css这两个文件，所以集成的时候只需要把这两个文件引入即可。代码如下：

```
/*
 * @Author: Rosen
 * @Date: 2019-06-20 23:08:44
 * @Last Modified by: Rosen
 * @Last Modified time: 2019-06-20 23:08:44
 */

/* css reset */
@import './reset.css';
/* 公共样式 */
@import './common.css';
```

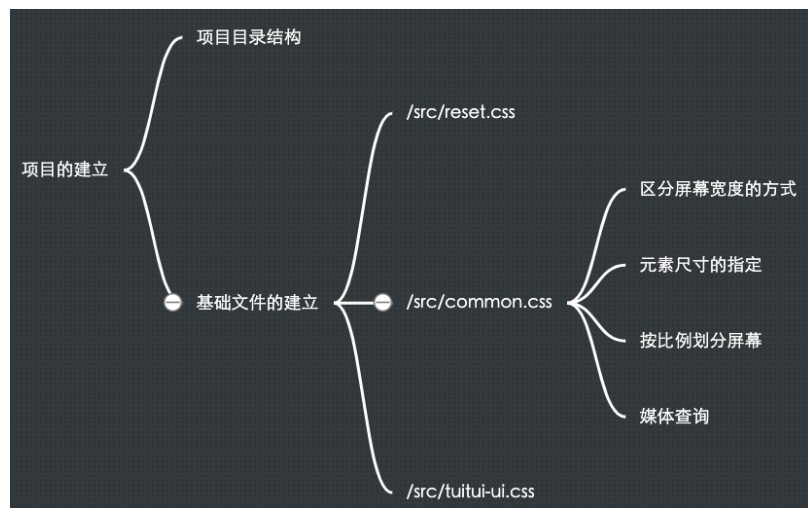
@ Tips:

在第二章的CSS规范一节，我们讲过使用@import引入文件会使页面加载变慢，但这里为什么还是用@import来集成呢？这是因为我们后面还会使用工具对代码进行合并和压缩，处理以后的代码会把@import方式集成的代码都打包到一个文件里，并且还会做代码压缩。所以，在开发阶段先使用@import方式组织代码不会有什么问题。

到此，这个项目的基础文件就建立完成了，后面整个框架的开发都会在这几个文件的基础上进行扩展。

结语

我们这一节负担了两部分的内容，一个是基础目录的搭建，另一个是填充了目录里几个基础的文件。这一章的整体结构如下：



这一节我们就开始有代码的开发了。虽然我们这次课程是以专栏的形式呈现，但本质还是一个实战课程。同学们在学习的时候一定把课程里的代码自己敲一遍，最起码也要看懂并且运行一遍，像看书一样从头看到尾肯定是有太多收获的。我们这一节的内容就先到这里，下节开始介绍字体图标相关的内容。

超清加密 更多 视频资料 原价一折 加Q176223832