



vue实现PC端分辨率适配

分类专栏：vue 2019-07-25 16:12:23 25237 已收藏 89

依赖

项目基础配置使用 **vue-cli** 生成
自适应方案核心：阿里可神端布局方案 **lib-flexible**
px转rem：**px2rem**，它有webpack loader **px2rem**

开始

先用vue脚手架初始化一个webpck项目

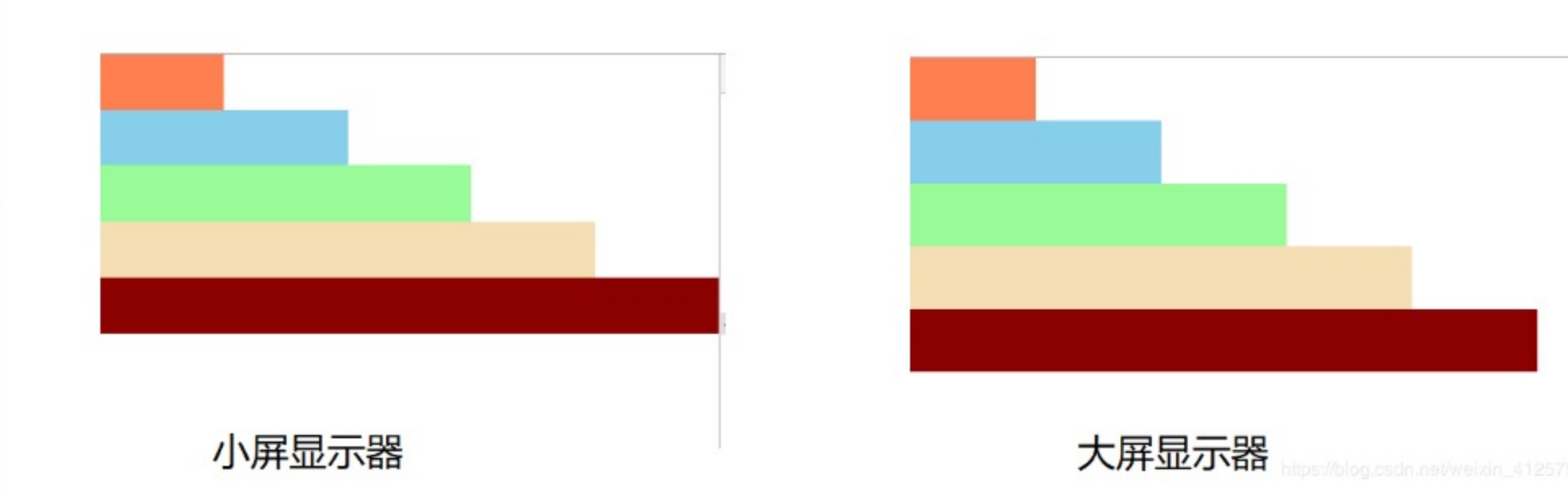
```
1 | vue init webpack 项目名称
项目初始化好了之后，进入项目目录中（cd 项目名）安装 lib-flexible 和 px2rem-loader
1 | npm i lib-flexible -S
1 | npm i px2rem-loader -D
```

安装好了之后还需要在项目的入口文件 main.js 里引入 lib-flexible

```
1 //...
2 import 'lib-flexible'
...
main.js × index.html × .gitignore × .editorconfig × package.json
1 import Vue from 'vue'
2 import App from './App'
3 import router from './router'
4
5 import 'lib-flexible'
6
7 Vue.config.productionTip = false
8
9 /* eslint-disable no-new */
10 new Vue({
11   el: '#app',
12   router,
13   components: { App },
14   template: 'App'
15 }).$mount('#app')
```

接下来为了验证 lib-flexible 是否生效，可以将app.vue中的内容改成：

```
1 <template>
2   <div class="wrapper">
3     <div class="box1"></div>
4     <div class="box2"></div>
5     <div class="box3"></div>
6     <div class="box4"></div>
7     <div class="box5"></div>
8   </div>
9 </template>
10
11 <style>
12   * {
13     margin: 0;
14     padding: 0;
15   }
16 </style>
17
18 <style scoped>
19   .wrapper div {
20     height: 1rem;
21   }
22   .box1 {
23     width: 2rem;
24     background-color: coral;
25   }
26   .box2 {
27     width: 3rem;
28     background-color: skyblue;
29   }
30   .box3 {
31     width: 6rem;
32     background-color: palegreen;
33   }
34   .box4 {
35     width: 8rem;
36     background-color: wheat;
37   }
38   .box5 {
39     width: 10rem;
40     background-color: darkred;
41   }
42 </style>
```



小屏显示器 大屏显示器

但是在实际开发中，我们通过设计稿得到的单位是px，所以要将其转换成rem再进行样式中。但如果都需要手动转换的话，就很麻烦，所以需要一个工具帮我们完成这项工作，这个时候就需要配置px2rem了，当然，编辑器可能也要对应的插件。

在 vue-cli 生成的 webpack 配置中，vue-loader 的 options 和其他样式文件 loader 最终都是由 build/utils.js 里的一个方法生成的。我们只要在 cssLoader 后再加上一个 px2remLoader 即可，px2rem-loader 的 remUnit 选项意思是 1rem=多少像素，结合 lib-flexible 的方案，我们将 px2remLoader 的 options.remUnit 设置成设计稿宽度的 1/10，这里我们假设设计稿宽为 1920px。

```
1 //utils.js
2 const cssLoader = {
3   loader: 'css-loader',
4   options: {
5     sourceMap: options.sourceMap
6   }
7 }
8
9 const px2remLoader = {
10   loader: 'px2rem-loader',
11   options: {
12     remUnit: 192
13   }
14 }
15 //...
```

故进入loaders数组中

```
1 //utils.js
2 function generateLoaders (loader, loaderOptions) {
3   const loaders = options.usePostCSS ? [cssLoader, postcssLoader, px2remLoader] : [cssLoader, px2remLoader]
4
5   if (loader) {
6     loaders.push({
7       loader: loader + "-loader",
8       options: Object.assign({}, loaderOptions, {
9         sourceMap: options.sourceMap
10      })
11    })
12  }
13  //...
```

修改配置以后 **重启服务器**，将原来app.vue文件中的样式改成：

```
1 <style scoped>
2   .wrapper div {
3     height: 1rem;
4   }
5   .box1 {
6     /* 750 * 200 */
7     width: 150px;
8     background-color: coral;
9   }
10  .box2 {
11    /* 750 * 400 */
12    width: 300px;
13    background-color: skyblue;
14  }
15  .box3 {
16    /* 750 * 800 */
17    width: 600px;
18    background-color: palegreen;
19  }
20  .box4 {
21    /* 750 * 800 */
22    width: 600px;
23    background-color: wheat;
24  }
25  .box5 {
26    /* 750 * 1000 */
27    width: 750px;
28    background-color: darkred;
29  }
30 </style>
```

可以看到，适配依然生效。

以上是vue-cli配置px2rem的方式，如果rem单位使用的是vue-cli3的话，由于vue-cli3 生成的目录结构没有build文件，所以在配置上也有些不同。

vue-cli3配置方式：

找到文件 node_modules/@vue/cli-service/lib/config/css.js，添加规则：

```
1 config.module
2   .rule('css')
3     .test(/\.css$/)
4     .oneOf('vue')
5     .resourceQuery(/!vue/)
6     .use('px2rem')
7     .loader('px2rem-loader')
8     .options({
9       remUnit: 75
10    })
```

```
html × css.js ×
const fs = require('fs')
const path = require('path')

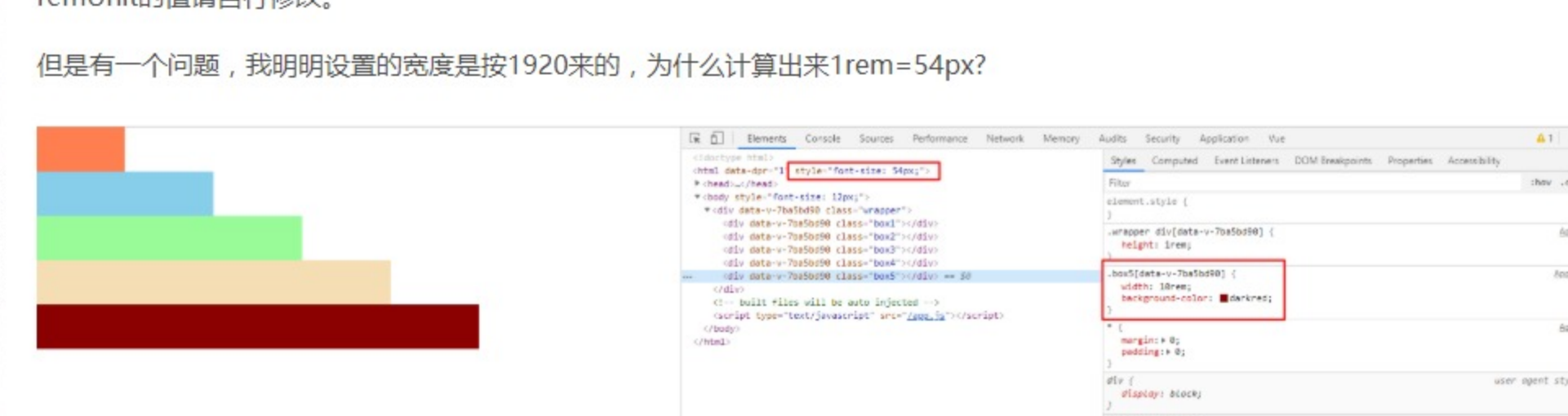
const findExisting = (context, files) => {
  for (const file of files) {
    if (fs.existsSync(path.join(context, file))) {
      return file
    }
  }
}

module.exports = (api, options) => {
  api.chainWebpack(config) => {
    const getAssetPath = require('../util/getAssetPath')
    const shadowNode = !process.env.VUE_CLI_CSS_SHADOW_NODE
    const isProd = process.env.NODE_ENV === 'production'

    config.module
      .rule('css')
      .test(/\.css$/)
      .oneOf('vue')
      .resourceQuery(/!vue/)
      .use('px2rem')
      .loader('px2rem-loader')
      .options({
        remUnit: 75
      })
  }
}
```

remUnit的值自行修改。

但是有一个问题，我明明设置的宽度是按1920来的，为什么计算出来1rem=54px?



是不是插件哪里出了问题，或者在配置哪里过了54或者540相关的东西？
(我这里全局搜索了54找到flexible.js文件)

找到flexible.js文件看：

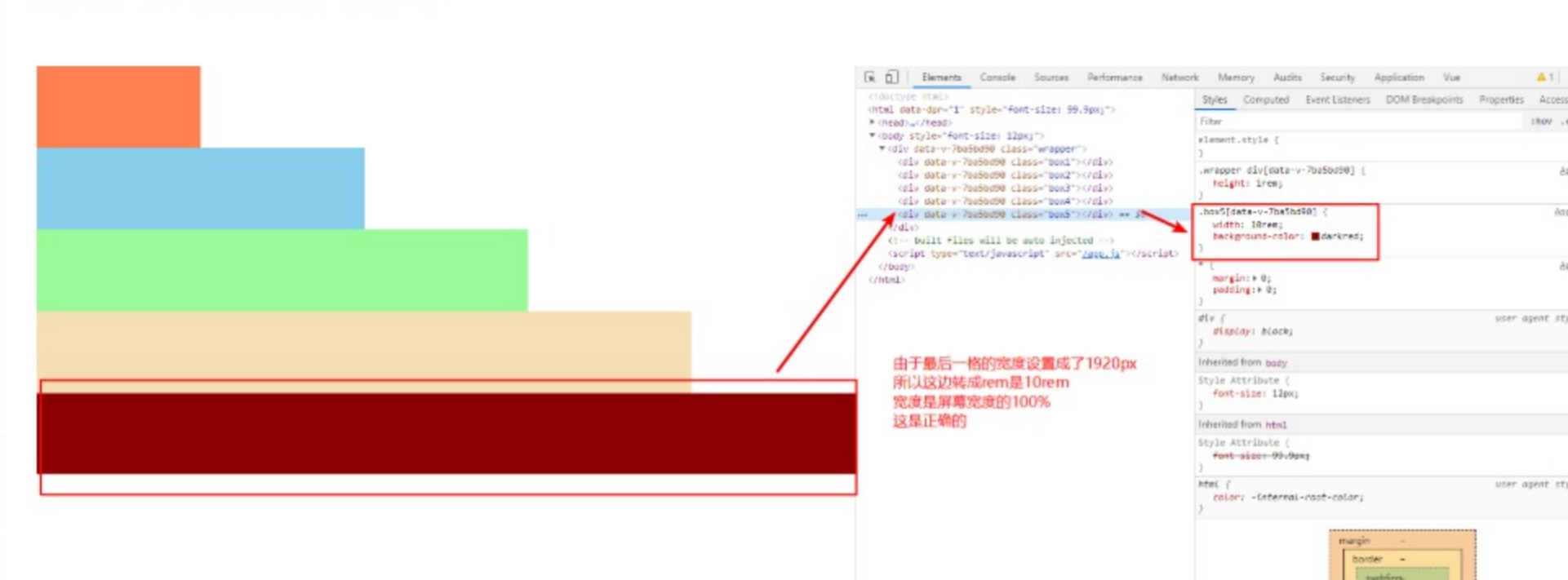
```
function refreshRem(){
  var width = docEl.getBoundingClientRect().width;
  if (width / dpr > 540) {  // 获取屏幕宽度
    width = 540 * dpr;      // 但是当屏幕宽度大于540的时候
                             // 把屏幕宽度与540对齐
  }
  var rem = width / 10;
  docEl.style.fontSize = rem + 'px';
  flexible.rem = win.rem = rem;
}

win.addEventListener('resize', listener: function() {
  clearTimeout(tid)
  tid = setTimeout(refreshRem, timeout: 300);
}, options: false);
win.addEventListener('pageshow', listener: function(e) {
  if (e.persisted) {
    clearTimeout(tid)
    tid = setTimeout(refreshRem, timeout: 300);
  }
}, options: false);
```

找到问题了解决问题，既然文件把屏幕宽度写死了，那就写不死：

```
1 function refreshRem(){
2   var width = docEl.getBoundingClientRect().width;
3   if (width / dpr > 540) {
4     width = width * dpr;
5   }
6   var rem = width / 10;
7   docEl.style.fontSize = rem + 'px';
8   flexible.rem = win.rem = rem;
9 }
```

现在再重启项目，看一下运行结果：



end

有问题欢迎指正！

vue项目PC端屏幕分辨率与窗口大小自适应（五步完成） 519kDA · @ 2020

<template> <div class="big"> <div class="box1"></div> <div class="box2"></div> <div class="box3"></div> </div> </template>

PC端自适应使用rem weixin_44999830的博文 · @ 4399

做一个PC端的网页，设计图是1920X1080，要在浏览器上正常显示，比如：1280X720 1366X768 1440X900 1920X1080。要求使用REM，width，height...

新建一个Vue项目，并成功运行Vue init webpack mp-demo项目结构因为是在一个项目中，分了移动、PC端，所以我们在入口、页面、路由等均为...>

px转rem的兼容性 weixin_34009162的博文 · @ 4241

vue移动端适配的最佳解决方案，亲测有效 慕高丹的博文 · @ 37+

vue项目自适应桌面端处理 一个千万级用户的项目 · @ 7504

移动端适配（个人理解） 少小无猜 · @ 17+

vue实现PC端适配 lib-flexible+webpack px2rem favurite23的博文 · @ 1900

vue PC端适配方案flexible+px2rem-loader weixin_39595959的博文 · @ 3407

利用webpack配置px2rem-loader和lib-flexible实现px转rem1. 安装npm install px2rem-loader -Dnpm install lib-flexible -S2.参照官网配置...>

vue移动端适配配置 qq1345930282的博文 · @ 4708

vue实现PC端自适应 vue cli3 实现 weixin_43365995的博文 · @ 461

使用Excel导出表格，能获取到导出文件的地址，此时就相当于一页数据打开一个新的链接<button type="button" id="export" class="btn">导出</button>...>

vue PC端适配：使用Px2rem-loader和lib-flexible 胡小小白 · @ 2625

* 程序员的新技能树 04-24

获取付费课程方式：在付费课程最后一节课弹幕命令 ALLIT开头有二维码及拉群教程介绍 原价99元，今日仅收39元！购买请点小助手（微信名：c...

* Windows/Excel/Win10目标检测实践：训练自己的数据集 04-26

课程展示课：YOLOv4 cuda 10.2; cudnn 7.6.5; Python3.7; VisualStudio2019; OpenCV3.4需要学习ubuntu系统上YOLOv4的网子请前往： CYO...

© 2020 CSDN 皮肤主题：技术黑板 设计师：CSDN官方博客 返回顶部

关于我 招聘 网站地图 kefu@cson.net 底部论坛 400-660-0108 QQ客服 (8:30-22:00)
公司地址 北京市海淀区中关村大街188号10层1018号 联系电话 010-60457562 邮箱 11980262@qq.com
网络违法和不良信息举报：举报电话：010-12377 网络举报中心：www.12377.cn 010-60457562 北京新网数码信息技术有限公司