sass依赖于 Ruby环境

注意：所有的环境中，不能有任何的中文、空格、特殊符号，换句话就是 路径必须是纯英文或英文数字的组合。

windows 默认没有需要安装

ruby官网：https://www.ruby-lang.org/en/downloads/

下载安装，一路下一步，（找一个叫xxx..PATH）

安装完成以后，打开window的cmd

（win+r） 输入 cmd

输入：ruby -v 如果出现版本号，代表成功，

如果出现 ruby -v 不是内部的命令 代表失败

ruby 安装成功后，在命令行中

切换源

1 添加、替换 gem sources --add https://gems.ruby-china.com/ --remove https://rubygems.org/

2 查看 gem sources -l

参考资料：gems.ruby-china.com/

然后安装sass

gem install sass

sass -v 出版本号表示正常

编译：

1 找到要编译的文件

怎么找到？

1 cmd找

常用的cmd指令：

dir 显示当前目录所有文件

cd 打开文件

cd 文件或文件夹名称

cd .. 返回上一层

2 使用地址栏复制，然后到cmd中粘贴（粘贴之前先cd空格一下）

3 右键管家(右键菜单) 添加 在此处打开命令窗口

2 编译：

单次编译： sass sass文件名:css文件名

例如： sass a.scss:b.css

单文件监听：

sass --watch a.scss:a.css

文件夹监听：

sass --watch 要编译的文件夹名称:输出的文件夹名称

3 编译的格式：

sass --watch a.scss:a.css --style 格式

可选择：

Nested //嵌套输出

\* expanded

\* compact

compressed

-->

<!--

设备尺寸：

414

元素：200px

移动端设备提供的可视区: 980px

移动端页面头部

meta

viewport

content可选参数

width – viewport宽度(数值/device-width) 也可以是具体的数值

height – viewport高度(数值/device-height)

initial-scale – 初始缩放比例 任意值

user-scalable – 是否允许用户缩放 yes/no

maximum-scale – 最大缩放比例

minimum-scale – 最小缩放比例

例如：

<meta id="viewport" name="viewport" content="width=640, initial-scale=1.0, user-scalable=1,minimum-scale=1,maximum-scale=1">

<meta id="viewport" name="viewport" content="width=device-width,minimum-scale=1,maximum-scale=1">

什么叫移动端？ （手持设备）

手机

pad

移动端页面大小

640

x3 960

1rem = 40px

750

x3 1125

1rem = 50

retina (视网膜屏幕)

把2px 缩小到 1px中显示

em 根据font-size来计算大小的

计算规则类似于js的变量

如果自己有font-size 根据自己的font-size来计算em的大小

如果没有，一直往上找

如果找不到font-size 那么用浏览器默认的字体大小 16px

rem

W3C在文档中有这么一句话：font size of the root element

根元素： html

retina屏幕适配

图片适配

背景

背景图片适配 background: url(pic.png) 0 0 no-repeat; background-image: -webkit-image-set(url(pic.png) 1x, url(pic@2x.png) 2x); 需要加前缀

img标签

<img src="pic.png" srcset="pic@2x.png 2x">

nodeJs环境：

https://nodejs.org/en/

npm 以后安装插件用的

问题：服务器在国外，速度上相对会慢

可以选择使用cnpm (淘宝镜像的服务器-每10分钟同步一次)

http://npm.taobao.org/

cnpm的安装：

npm install -g cnpm --registry=https://registry.npm.taobao.org

问题：可能会报错（管理员权限问题）

win: 开始菜单单击右键>搜索>输入cmd>在结果中右键 以管理员身份运行 再次输入代码就可以了

mac: 如果遇到管理员权限问题

sudo ....

安装插件：cnpm install xxx

cnpm i --save-dev gulp gulp-px2rem browser-sync gulp-sass node-sass 依赖

browser-sync安装：

sync 读音同 think

官网：http://www.browsersync.cn/

cnpm install -g browser-sync

注意：权限问题

如果手机访问不了，window的防火墙一定要关掉

fixScreen

viewport 方案需要些meta标签

优点：直接使用px作为单位，通过缩放视窗来适配

缺点：部分低版本浏览器不支持该方案

rem 方案

优点：兼容性相对完美

缺点：写的时候，得把px转成rem -->

flexible

在head中引入

flexible\_css.js

flexible.js

(可选配置) 可以手动的设置dpr <meta name="flexible" content="initial-dpr=2”>

目前Flexible会将视觉稿分成100份（主要为了以后能更好的兼容vh和vw），而每一份被称为一个单位a。同时1rem单位被认定为10a。针对我们这份视觉稿可以计算出：

1a = 7.5px

1rem = 75px

那么我们这个示例的稿子就分成了100a，也就是整个宽度为10rem，对应的font-size为75px

px > rem

gulp

gulp

网址：https://www.gulpjs.com.cn/docs/getting-started/

px -> rem

插件：https://www.npmjs.com/package/gulp-px2rem -->

1 全局安装gulp

cnpm install --global gulp

2 项目依赖

- 安装到项目的根目录

- 在项目根目录中出现 node\_modules 代表成功了

cnpm install 插件的名称 --save-dev

3 在项目的根目录 新建 gulpfile.js

px2rem插件：

npmjs.com/package/gulp-px2rem

browser-sync

http://www.browsersync.cn/docs/gulp/

sass

https://www.npmjs.com/package/gulp-sass

gulp串行和并行

串行：一个执行完了，执行下一个

gulp.series(执行的任务, gulp.series...)

并行：多个任务同时执行

gulp.parallel()

查看css3属性兼容情况：

https://caniuse.com

vireport width

vw - 视窗宽度

vh - 视窗高度

vmin - 最小 - 宽和高中最小的那个

vmax - 最大 - 宽和高中最大的那个

页面640

元素：315?

315-> 49.21875vw

flexibox

把视觉稿分成100份

1份 1a

1rem = 10a

npm

bower 安装前端相关的框架的

npm i bower -g

bower i 框架的名字#info

选择版本

bower i jquery#xx.xx.xx

windows 需要安装git

https://www.swiper.com.cn/

jQuery的 scrollLoading.js

Jquery还有一个叫lazyload插件，但该插件有一些bug,之前加载过的图片会重新加载。（不用这个了就）

使用方法

<img class=“你的class名字” data-url="真实的图片地址" src="临时图片地址" width="120" height="90”>

$(“.你的class名”).scrollLoading();

版本选择：要求jq版本3.0+

配置参数

attr - 可以配置自定义的属性（可以把默认的data-url改成我们自己写的属性）

callback - 加载完成（相当于img的onload）

container - 默认是window下延迟加载，可以指定是某个容器内部的延迟加载

其他参数见资料 scrollLoading资料

4 不安全！！！

localStorage

特性

永久保存

大小5M

设置localStorage

window.localStorage.xxx = xxx;

localStorage.setItem("user","busy");

获取locaStorage

window.localStorage.xxx

localStorage.getItem("user");

清除

localStorage.removeItem() 单个

localStorage.clear(); 全部

Cookie：一般由服务器生成，可设置失效时间。如果在浏览器端生成Cookie，默认是关闭浏览器后失效、4K左右、需要开发者自己封装

localStorage：除非被清除，否则永久保存、一般为5M、提供原生api

sessionStorage：仅在当前会话下有效，关闭页面或浏览器后被清除、一般为5M、提供原生接口

function getCookie(attr){

var arr = document.cookie.split('; ');

var json = {};

arr.forEach(function(o,i){

var newarr = o.split('=');

json[newarr[0]] = newarr[1];

})

// if(json[attr]) return true;

return json;

}获取COOKIE

function removeCookie(attr){

setCookie(attr,'',-1);

}删除cookie

function setCookie(attr,value,day){

var d = new Date();

d.setDate(d.getDate() + day);

document.cookie = attr + '='+value+'; path=/; expires=' + d;

}设置cookie

cookie特点：

1 同一个域名的页面，公用一套cookie

新浪首页： http://www.sina.com.cn/

新浪NBA频道： http://sports.sina.com.cn/nba/

http://www.sina.com.cn:3000/news/index.html

http:// https:// - 协议

www 二级域名

sina.com.cn 一级域名

:3000 端口

news/index.html 具体地址了

2 一般是 4k （1024字节 = 1k）

3 过期 （一般是 随着你的页面关闭了，他自然就过期了。）

4 不安全！！！

**外阴影：**box-shadow: X轴  Y轴  Rpx  color;

**属性说明（顺序依次对应）：**阴影的X轴(可以使用负值)    阴影的Y轴(可以使用负值)    阴影模糊值（大小）    阴影的颜色

**内阴影：**box-shadow: X轴  Y轴  Rpx  color  **inset**;

       默认是外阴影   内阴影：inset 可以设置成内部阴影

background: linear-gradient(to left, red, yellow, green);线性渐变