1. H5缓存

1>概念

通过创建 cache manifest 文件, 可以轻松地创建 web 应用的离线版本

2>基本使用

(1)如需启用应用程序缓存，请在文档的 <html> 标签中包含 manifest 属性 -->

<!DOCTYPE HTML>

<html manifest="demo.appcache">

...

</html>

(2)创建manifest 文件的建议的文件扩展名是：".appcache"

注意：manifest 文件需要配置正确的 MIME-type，即 "text/cache-manifest"。必须在 web 服务器上进行配置

(3)每个指定了 manifest 的页面在用户对其访问时都会被缓存。如果未指定 manifest 属性，则页面不会被缓存（除非在 manifest 文件中直接指定了该页面）

3>Manifest 文件

(1)概念：

manifest文件是简单的文本文件, 它告知浏览器被缓存的内容(以及不缓存的内容)

(2)manifest 文件可分为三个部分

##CACHE MANIFEST – 开始

CACHE在此标题下列出的文件将在首次下载后进行缓存

##NETWORK - 在此标题下列出的文件需要与服务器的连接，且不会被缓存

##FALLBACK - 在此标题下列出的文件规定当页面无法访问时的回退页面(比如 404 页面)

(3)三部分说明

##CACHE MANIFEST说明：

CACHE MANIFEST，放置在第一行，是必需的 -->

CACHE ：

/theme.css

/logo.gif

/main.js

上面的 manifest 文件列出了三个资源：一个 CSS 文件，一个 GIF 图像，以及一个 JavaScript 文件。当 manifest 文件加载后，浏览器会从网站的根目录下载这三个文件。然后，无论用户何时与因特网断开连接，这些资源依然是可用的

##NETWORK说明：

NETWORK小节规定文件"login.asp"永远不会被缓存, 且离线时是不可用的 -->

NETWORK:

login.asp

##FALLBACK说明：

FALLBACK小节规定如果无法建立因特网连接, 就使用指定的资源代替所请求的url的资源, -->

FALLBACK:

/html5/ /404.html

注释：当html5资源在离线状态下无法请求的时候，就使用404.html代替

(4)总结说明

##CACHE: 可以省略，这种情况下将需要缓存的资源写在CACHE MANIFEST

##可以指定多个CACHE: NETWORK: FALLBACK:，无顺序限制

## #表示注释, 只有当demo.appcache文件内容发生改变时或者手动清除缓存后，才会重新缓存。

##chrome 可以通过chrome://appcache-internals/工具和离线（offline）模式来调试管理应用缓存

(5)更新缓存

一旦文件被缓存，则浏览器会继续展示已缓存的版本，即使修改了服务器上的文件。为了确保浏览器更新缓存，也需要更新 manifest 文件，也就意味着一旦应用被缓存，它就会保持缓存直到发生下列情况：

1. 用户清空浏览器缓存
2. manifest 文件被修改（参阅下面的提示）
3. 由程序来更新应用缓存

说明：更新注释行中的日期和版本号是一种使浏览器重新缓存文件的办法

后面查看HTML5-三~五天笔记

1. 适配方案(通过视口, 配合流式布局)
2. 流式布局

宽度使用百分比(百分比布局), 高度用px来固定住

1. viewport(视觉窗口, 即视口)

(1)简介

viewport是移动端特有的, 是一个虚拟的区域, 承载网页, 浏览器承载视口, 视口承载网页(使得PC端页面放到移动端不出现滚动条, 原理是缩放)

缺点：内容也会缩放(采用viewport默认值会)

(2)标准适配方案(改变viewport默认设置以达到内容正常显示, 且无滚动条)

##注意点：

++设置viewport需紧跟在编码(<meta charset=”UTF-8”>)后面

++s通过meta标签的content属性来传递参数来改变其默认值 -->

<meta content=”里面设置参数”>

++开启默认缩放(user-scalable=yes)之后maximum和maximum才有用

##viewport的主要功能(参数)

++width 设置宽度 (device-width当前设备的宽度)

++height 设置高度

++initial-scale 缩放比例

++user-scalable 是否允许用户缩放 (no, 0用户不能缩放, yes, 1用户能缩放)

++maximum 最大缩放比

++maximum 最小缩放比

(3)国际标准适配方案



(4)生成标准适配方案快捷键meta:vp+tab

1. 移动开发注意点
2. 屏幕像素和px区别

屏幕像素：物理像素, 即像素点, 设备屏幕的最小可视颗粒

px:页面的尺寸单位

1. 移动开发可以使用jquery但不推荐使用

原因：jquery做了很多PC端浏览器兼容, 特别是ie, 移动端没有ie

1. 移动端内核都是webkit或者blink 兼容都是加-webkit-前缀
2. 建议使用h5的pai或者使用 zepto.js的库(基于高版本浏览器开发)
3. 常用box-sizing原因

(1)移动开发常用流式布局(宽度百分比, 高度固定)

(2)普通盒模型无法准确计算盒子大小 -->

.box{ wdith: 100%; height: 200px; border: 20px solid red; }

普通盒子无法计算盒子大小会导致出现横向滚动条

ie盒模型其大小为100%, 200px无横向滚动条

(3)设置IE盒模型之后都是从边框开始计算盒子尺寸的

6>在项目中初始化文件中常见设置

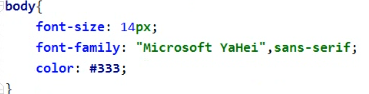
(1)设置成IE盒模型

\*, \*::before, \*::after{ box-sizing: border-box; -webkit-box-sizing: border-box}

(2)清除高亮属性(移动端特有的默认样式, 注意和伪类:active区分)

-webkit-tap-highlight-color: transparent;清除高亮

(3)设置字体

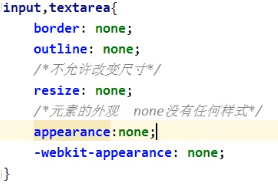


没有Microsoft YaHei就用系统自带的字体

(4)去列表原点



(5)输入框



7>px是css单位, 不是真机上的分辨率,

8>1px单位里面可能可以放多个物理像素点

9>物理像素比 非矢量网站可能会失真, (标准视口使用二倍图可以解决失真问题)

10>移动端屏幕划分区间

(1)<768px 超小屏设备(手机)

(2)[769px, 992px) 小屏设备(平板电脑)

(3)[992,px 1200px) 中屏设备(台式电脑)

(4)>=1200px 大型设备(台式电脑)

1. 移动端常见多栏布局
2. 双飞翼(圣杯模式)

两边固定中间不固定

2>弹性盒子布局flex

用法见HTML第五天笔记

3>响应式

1. iScroll插件区域滚动插件注意点
2. 使用插件new(“父容器(小容器)”, function () { 配置参数} )
3. 需要滚动的子容器必须要大于父容器
4. 父容器和子容器之前需要夹一层容器 -->

div.wrap(父容器)>ul>li(需要滚动的容器)