**H5和CSS3新增内容（自己总结）**

HTML5新增标签及属性

CSS3新增样式

Web API

**选用标签的标准**

如果要支持比较老的浏览器（通常是桌面版网页）则不使用新标签

手机平板等网页开发可以使用新标签

不管什么情况下，都应该优先根据语义来选用标签

根据搜索引擎优化的要求和建议选用标签

**布局**

流式布局

定位

浮动/清除

Flex

布局框架（如Bootstrap栅格）

**CSS单位与计算**

px

%

rem/em

vw/vh

mm/cm

calc()

**CSS半透明/透明**

大部分标签元素默认是背景透明的，但也有不透明的（如iframe、button、图片等）

transparent

RGBA

HSLA

opacity

**动画**

transition：transition能过渡大部分样式（<http://dwz.cn/3NvjUN>）而且支持多样式同时过渡，注意标签元素上有多个样式类时，与过渡样式相关的最后2个样式会形成过渡效果。

animation：注意使用animation-play-state:paused可以使动画就地暂停，animation-play-state: ‘’可以使动画在暂停的状态继续进行

jQuery动画：jQuery动画与CSS3动画不同，jQuery动画是js驱动的，jQuery动画与CSS3动画是互补关系（而非谁取代谁）。jQuery内置了一些动画，如（fadeIn/slideUp等），也支持通过animate()方法自定义动画。而且提供了强大的动画控制API，如动画延迟、链式动画（上一个动画结束后开始下一下动画）、控制动画停止、关闭动画效果、动画频率调整、还可以指定回调函数参与每一个动画的每一步等。jQuery中的动画返回Deferred对象，因此支持Promise异步编程。jQuery默认不能过渡颜色，使用jQuery.Color插件可以解决问题。

触发动画：CSS一般在页面加载时会自动进行transition和animation，可以在浏览器的查看器（动画窗格）中观察到标签元素上的动画帧并对动画进行控制及设置不同的速度进行预览。使用:hover、:active伪类可以配合鼠标指针触发动画效果。通过js添加删除标签元素的样式类时也会触发相应的动画。jQuery动画的触发和控制由jQuery动画API控制。

动画结束事件/回调：CSS3动画支持DOM事件：animationstart（动画开始时）、animationend（动画结束时）、animationiteration（动画重复执行时），可以像普通click事件一样使用它们。jQuery动画通过回调函数获取执行完成通知，形如：$(..).animate({css}<， duration><， easing><， callbakc>)，因为jQuery动画返回Deferred对象，所以也可以使用.then(success，fail)或.done(function)等方法得到通知。

同时进行多个动画：在关键帧定义中可以定义多个样式，animation-name也可以指定多个关键帧动画名称。transition-property也可以指定多个样式。jQuery的animate方法中也可以指定多个样式。还可以在同一个标签元素上调多次animate方法（分开调用，而不是链式调用）

顺序进行多个动画：可以通过animation-delay、transition-delay延迟，安排出顺序执行效果。还可以通过监听动画事件在动画结束后再触发下一个动画。jQuery中animate()的链式调用本身就是顺序执行的。

requestAnimationFrame：会与浏览器呈现页面同步配合执行，因此比使用setInterval要更加流畅。requestAnimationFrame是H5新增的方法，与setInterval相比，没有提供频率的控制方法，因为它的频率与页面呈现频率是同步的，通常60次/s，相当于16.66667ms一次。

矩阵与图形变换

贝塞尔曲线与时间函数

**媒体查询与响应式页面**

媒体查询@media赋予CSS探测设备类型和设备属性（如屏幕宽高等）的能力，之前媒体查询主要用来控制页面打印效果（即针对页面上打印部分定义打印专用样式）；在移动互联网时代媒体查询对于页面适配各种大小的屏幕至关重要，可以说媒体查询是响应式页面的核心技术。Bootstrap就是以媒体查询技术为基础的响应式UI框架。

**浏览器兼容性和Web App页面（浏览器兼容性专题文档介绍的更详细）**

浏览器兼容性问题涉及的面比较广，从HTML标签、CSS样式到JS都存在兼容性问题。兼容性问题给Web开发造成了额外的成本支出，在兼容性上要求越高，付出的代价越大。

浏览器的兼容性主要通过测试解决，桌面网页最为困难的是兼容IE6/7/8/等老浏览器（可以使用IE Tester或IE11的开发者工具测试）！解决浏览器兼容性没有什么难度，繁锁而已，如果要兼容老浏览器，则应放弃使用新功能或者对老浏览器显示友好的提示信息（使开发测试时间增长、成本增加）。

手机浏览器中最为困难的是Android版的UC浏览器，Android版的UC几乎不支持H5和CSS3，甚至是常用的CSS样式有些都不支持，如果项目要兼容UC手机浏览器，那写CSS时，就只能使用最基本的样式！也不要指望页面能做多漂亮多复杂了。

Android系统上除了UC这个奇葩之外，本身的碎片化、更新慢也是大问题。如很多手机现在的Android系统还是4.4，甚至更古老（尽管Adnroid7都发布了）。如果开发应用内嵌页面，可以建议Android工程师采用腾讯出的WebView控件，能够使页面对H5的支持和微信及手机QQ一样好，从而达到和iOS同级别的效果。

直接使用框架可以有效地避免兼容性问题，如各种UI框架和jQuery框架都会考虑兼容性。另外使用重置样式表也可以抹平各浏览器的差异。

在开发Web App时需要注意，移动设备，无论是iOS还是Android都使用Webkit内核，尤其是iOS，需要使用以下样式：

-webkit-appearance: none; 禁用浏览器特定外观（如按扭等表单元素）

-webkit-tap-highlight-color: rgba(0，0，0，0); 将触摸时的高亮颜色设为完全透明

-webkit-touch-callout: none; 禁用页面上的上下文菜单（上面有复制、粘贴等功能）

-webkit-text-size-adjust: none; 禁用页面自动调整文字大小

-webkit-user-select: none; 禁止用户选择页面上的文字

回弹滚动

overflow: auto; /\* auto | scroll \*/

-webkit-overflow-scrolling: touch;

另：有一些样式还需要使用-webkit前缀，如：

CSS滤镜要写成-webkit-filter

或者要兼容特别老的iOS或Android系统时CSS3样式要加-webkit前缀

另：如果需要对某些标签进行深度控制，则需要使用-webkit伪类，如：

input[type=range]::-webkit-slider-runnable-track{ }

input[type=range]::-webkit-slider-thumb{ }

可以控制滑动条的样式

progress::-webkit-progress-bar { }

progress::-webkit-progress-value { }

可以控制进度条的样式

另：使用meta标签可以要求浏览器支持或关闭某些行为

<meta name="format-detection" content="telephone=no" /> 禁用手机号探测

<meta name="format-detection" content="email=no"> 禁用email探测

固定到手机桌面（桌面书签）

<meta name="apple-mobile-web-app-title" content="常伟">

<link rel="apple-touch-icon" sizes="57x57" href="icon57.png" />

<link rel="apple-touch-startup-image" href="launch6.png">

<meta name="mobile-web-app-title" content="常伟">

<meta name="mobile-web-app-capable" content="yes">

关闭iOS键盘首字母自动大写，自动修正

<input type="text" autocapitalize="off" />

<input type="text" autocorrect="off" />

**LESS/SASS/SCSS**

Less 是 CSS 预处理语言，在CSS中引入了变量、函数、混入等编程机制，从而使CSS获得更高的灵活性和可扩展性。Less可以使用js直接在页面上转换成CSS，也可以使用命令行工具Lessc完成从less文件到css文件的转换，还有相应的自动化构建插件。注意Less中使用calc()或滤镜等函数时需要使用 ~””或e(””)进行转义，否则会被误识别为less函数，而less中并没有这些函数。

Sass （Sass3时提供了兼容CSS的语法，被称为SCSS）是对 CSS 的扩展，除支持变量、函数、混入外还支持条件判断和循环语句，Sass让 CSS 语言更强大、更优雅。

值得一提的是，Less是基于Node.js和js的，而Sass/SCSS是基于Ruby的（也需要Node.js），Sass/SCSS不能直接在页面上通过js转换成CSS。

**JS基本数据类型（typeof的返回结果）**

number（Infinity/NaN）

string

boolean

function

object（null、各种值装箱对象、内置对象、自定义对象）

undefined

判断对象是否为某个【类/构造函数】的实例（注意继承）用instanceof（对象与函数的关系）

获取对象的原型用Object.getPrototypeOf(obj)，如果原型中有constructor则可获得构造函数[获得的是函数而不是函数名字]

判断A对象是否为B对象原型链上的一环A.isPrototypeOf(B) （对象与对象之间的关系）

**原型与原型链**

如果在构造函数中直接为对象添加属性和方法，则会导致每个对象都包含一份自己的属性和方法，当大量创建对象时就会浪费很多存储空间。而将属性和方法添加到构造函数的原型中就可以使通过该构造函数创建出来对象共用这些属性和方法。即原型实现了属性和方法的**共享**。

为构造函数指定原型：

ObjFunc.prototype = object.create(baseFunc)

ObjFunc.prototype.constructor = baseFunc

ES6中支持 class 和 extends 关键字，就不需要用上面的方式了（但本质上还是一样的）。

当访问对象的属性和方法时，JS运行时会首先在该对象自身中查找要使用的属性和方法，找到就调用；如果找不到，则会到原型中查找。因此原型中的属性和方法虽然是共享的，但它们的优先级要低一点。

JS运行时在查找可调用的属性和方法时并不只查找一层原型，因为原型本身也是对象，因此JS运行时还会查找原型对象的原型，一直查找到Object的原型为止。这样就构成了一个链式结构，称为原型链。通过原型链可以形成类似**继承**的效果。

但JS中的继承与其它编程语言中的继承还是有区别的！在其它编程语言中，任何类的对象都是由一个实例构成的，属性值也只有一个值，不管继承多少层。而JS通过原型链形成的继承是由多个实例构成的链，属性值也可以有多个值（每一层原型上都可以有一个不同的值），继承的层次越多，涉及的实例就越多，值也越多！

JS的原型链在Angular中还被用于构造作用域链。

**this**

js中的this与其它编程语言中的不同，js中的this指向的对象不是固定的，而是可以改变的（通过function对象的call、apply、bind，jQuery的proxy等可以改变this指向）。以下是this的默认指向：

1. 定义在全局作用域中的普通函数中的this指向window对象（严格模式下本条无效，参见后面严格模式的说明）
2. 事件处理函数中的this指向触发事件的标签元素
3. 构造函数中的this指向当前正在创建的对象（严格模式下必须使用new）
4. 原型中的函数内的this指向当前实例
5. 其它框架中的函数内的this都有特定的指向，如jQuery.each(func)中func中的this指向当前迭代的标签元素对象。

**DOM**

文档对象模型，包括document本身和页面上的各种标签元素。DOM是访问和控制文档及标签元素的API，通过DOM可以查找、修改、添加、删除标签元素，还可以添加事件监听函数以响应页面上的各种事件。H5新增了一些DOM API（如querySelector）在一程度上增强了DOM，但实际开发中，使用jQuery进行开发还是非常普遍的。因为jQuery不仅提供了强大的DOM操作能力，而且能够解决浏览器兼容性问题。此外jQuery还提供了Ajax请求、动画等实用功能，还提供了Callbacks、Deferred等高级API。

**BOM**

浏览器对象模型，包括window、location、history、navigator等

window对象是全局作用域，window对象会随页面的加载或刷新而重置，即全局作用域生命周期相关的内容都会被销毁。window对象提供了窗口相关的API，用于控制窗口及窗口与窗口之间的关系。window对象还提供了XHR、setTimeout、alert、WebWorker等各种编程接口和功能支持。实际上从window对象出发可以访问几乎所有的Web API。

location提供了与url相关的API，通过location提供的属性可以获取url的各个组成部分。location还提供了对页面进行导航控制的基本功能，如对href赋值、assign()、reload()、replace()方法等。值得注意的是location提供了hashchange事件，可以监听hash fragment的变化，这对于单页Web应用中的路由机制提供了重要的支持。

history对象提供了浏览历史记录控制功能，如back()、forward()、go()等，H5还新增了状态相关的API：state、pushState、replaceState配合window的popstate事件可以为路由机制提供重要的支持。

navigator对象提供了浏览器和操作系统信息的描述，如userAgent、H5还在navigator中加入了很多设备访问功能，如geolocation、battery、vibrate等（浏览器目前支持不太好）。

**JS语言核心对象**

包括Object、Function、Array、与值类型对应的包装对象（Boolean、Number、String）、Math、Date、Error、Regexp、JSON，这些对象是JS编程语言自带的，无论JS在何处（如Photoshop、Mongodb等）运行都会包含这些对象。