1. HTML表单中的元素：

如果没有name属性，那么这个input元素不会放入提交的表单数据中。

如果没有disabled属性，那么这个input元素不会放入提交的表单数据中。但是readonly属性的元素会放入提交的表单数据中。

2.

HTML表单中的checkbox, 如果没有设置value属性而被选定，那么在选定的情况下返回的值为on. 如果设置了value值，那么返回的值为设置的value值。如果没有选定，那么直接不会显示在提交的数据字典当中。

3.

HTML的input元素属性设置中，required与readonly元素如果同时设置，将会有冲突。即使元素的值为空，也不会被required属性认为是违反规定，表单仍可提交。

4.

white-space:nowrap 用于告诉浏览器如何处理空格。如果设置为nowarp,则文本不会换行，文本会在在同一行上继续，直到遇到 <br> 标签为止

5. 在HTML5中，新加入了一个localStorage特性，这个特性主要是用来作为本地存储来使用的，解决了cookie存储空间不足的问题(cookie中每条cookie的存储空间为4k) ，localStorage中一般浏览器支持的是5M大小，这个在不同的浏览器中localStorage会有所不同。

localStorage的优势：

localStorage拓展了cookie的4K限制

localStorage会可以将第一次请求的数据直接存储到本地，这个相当于一个5M大小的针对于前端页面的数据库，相比于cookie可以节约带宽，但是这个却是只有在高版本的浏览器中才支持的

localStorage的局限：

浏览器的大小不统一，并且在IE8以上的IE版本才支持localStorage这个属性

目前所有的浏览器中都会把localStorage的值类型限定为string类型，这个在对我们日常比较常见的JSON对象类型需要一些转换（序列化的转换）

localStorage在浏览器的隐私模式下面是不可读取的

localStorage本质上是对字符串的读取，如果存储内容多的话会消耗内存空间，会导致页面变卡

localStorage不能被爬虫抓取到

localStorage的语句操作，主要是增删改查：

let storage = window.localStorage;

storage.setItem(“C”,1) #增加key “C”值为3

storage.removeItem(“C”) #删除key “C”

storage.setItem(“C”,2) #修改key “C”值为2

storage.getItem(“C”) #获取key “C”的对应值

一般我们会将整个JSON存入localStorage中，但是在localStorage会自动将localStorage转换成为字符串形式。 这个时候我们可以使用JSON.stringify()这个方法来进行序列化，来将JSON转换成为JSON字符串然后存入localStorage.

读取的时候使用JSON.parse(字符串)来反序列化，变为JSON对象。

6.

HTML开头信息生命，<!DOCTYPE> 不是 HTML 标签。它为浏览器提供一项信息（声明），即 HTML 是用什么版本编写的。

作为表明HTML5则为：<!DOCTYPE html>

7.

HTML5中不再支持<frame>, <frameset>元素。HTML5中使用<iframe>

8.

HTML5<picture>元素。

picture 元素允许我们在不同的设备上显示不同的图片，一般用于响应式。

HTML5 引入了 <picture> 元素，该元素可以躺图片资源的调整更加灵活。

<picture> 元素零或多个 <source> 元素和一个 <img> 元素，每个 <source> 元素匹配不同的设备并引用不同的图像源，如果没有匹配的，就选择 <img> 元素的 src 属性中的 url。

**注意**:<img> 元素是放在最后一个 <picture> 元素之后，如果浏览器不支持该属性则显示 <img> 元素的的图片。

例如：

<picture>

<source media="(min-width: 650px)" srcset="demo1.jpg">

<source media="(min-width: 465px)" srcset="demo2.jpg">

<img src="img\_girl.jpg">

</picture>

9.

<a>超链接属性hreflang表明超链接所指向的文是什么语言（使用两个字母的语言代码表示，例如en, zh）。

10.