

目 录

第一部分	1
1. HTML5 基础	1
2. CSS3 基础	5

达内IT培训集团

第一部分

1. HTML5 基础

1.1. 请描述一个网页从开始请求到最终显示的完整过程？

一个网页从请求到最终显示的完整过程一般可分为如下 7 个步骤：

1. 在浏览器中输入网址；
2. 发送至 DNS 服务器并获得域名对应的 WEB 服务器的 IP 地址；
3. 与 WEB 服务器建立 TCP 连接；
4. 浏览器向 WEB 服务器的 IP 地址发送相应的 HTTP 请求；
5. WEB 服务器响应请求并返回指定 URL 的数据，或错误信息，如果设定重定向，则重定向到新的 URL 地址。
6. 浏览器下载数据后解析 HTML 源文件，解析的过程中实现对页面的排版，解析完成后在浏览器中显示基础页面。
7. 分析页面中的超链接并显示在当前页面，重复以上过程直至无超链接需要发送，完成全部显示。

1.2. DOCTYPE 声明的作用是什么？严格模式与混杂模式如何区分？

HTML 语言已经存在太久了，目前必然会有一些不同版本的文档存在。为了能够让浏览器清楚你的文档的版本、类型和风格，需要在文档的起始用 DOCTYPE 声明指定当前文档的版本和风格。如果在网页中提供了版本信息，则可以有利于验证页面中的代码是否符合当前的版本和风格。

<!DOCTYPE> 声明位于文档中的最前面，处于 <html> 标签之前，告知浏览器的解析器，用什么文档类型规范来解析这个文档。

在严格模式（标准模式）中，浏览器根据规范呈现页面；在混杂模式中，页面以向后兼容的方式显示，以防止老站点无法工作。

如果 HTML 文档包含形式完整的 DOCTYPE，那么它一般以标准模式呈现。对于 HTML 4.01 文档，包含严格 DTD 的 DOCTYPE 常常导致页面以标准模式呈现。DOCTYPE 不存在或格式不正确会导致文档以混杂模式呈现。

1.3. 简要描述常见的浏览器内核。

浏览器内核负责对网页语法的解释并显示网页,它决定了浏览器如何显示网页的内容以及页面的格式信息。

常见的浏览器内核有：

- Trident：IE 浏览器；
- Gecko：Mozilla 浏览器，比如 Firefox；
- Webkit：Safari 浏览器，也是 Chrome 浏览器的内核原型；
- Blink：Chrome 浏览器，Opera 浏览器。

1.4. 如何理解 html 标签语义化？

语义化的主要目的在于，直观的认识标签(markup)和属性(attribute)的用途和作用。可以概括为：用正确的标签做正确的事情。

html 语义化可以让页面的内容结构化，便于浏览器解析，便于搜索引擎解析，并提高代码的可维护度和可重用性。

比如，尽可能少的使用无语义的标签 div，使用结构化标签<header>、<section>、<footer>。

1.5. 锚点的作用是什么？如何创建锚点？

锚点是文档中某行的一个记号，类似于书签，用于链接到文档中的某个位置。当定义了锚点后，我们可以创建直接跳至该锚点（比如页面中某个小节）的链接，这样使用者就无需不停地滚动页面来寻找他们需要的信息了。

在使用 <a> 元素创建锚点时，需要使用 name 属性为其命名，代码如下所示：

```
<a name=" anchorname1" >锚点一</a>
```

然后就可以创建链接，直接跳转到锚点，代码如下所示：

```
<a href="#" #anchorname1" >回到锚点一</a>
```

1.6. 使用 <label> 元素显示文本与使用其他文本标记显示文本有什么不同？

<label> 元素的直观效果是直接显示标记之间的文本，而且不会为文本呈现任何特殊效果。但是，它和其他文本标记所不同的是，它为鼠标用户改进了用户体验性。

这是因为，<label> 元素可以附带一个 for 属性，只要将该属性的值设置为表单中任何一个控件的 id 属性的值，则当用户点击该标签（文本）时，浏览器就会自动将焦点转到和标签相关的表单控件上。即：如果在 <label>元素内点击文本，就会触发此控件。

1.7. 列举常用的结构标记，并描述其作用。

结构标记专门用于标识页面的不同结构，相对于使用 `<div>` 元素而言，可以实现语义化的标签。

常用的结构标记有：

- `<header>` 元素：用于定义文档的页眉；
- `<nav>` 元素：用于定义页面的导航链接部分；
- `<section>` 元素：用于定义文档中的节，表示文档中的一个具体的组成部分；
- `<article>` 元素：常用于定义独立于文档的其他部分的内容；
- `<footer>` 元素：常用于定义某区域的脚注信息；
- `<aside>` 元素：常用于定义页面的一些额外组成部分，如广告栏、侧边栏和相关引用信息等。

1.8. 超级链接有哪些常见的表现形式？

`<a>` 元素用于创建超级链接，常见的表现形式有：

1、普通超级链接，语法为：

```
<a href="" target="">文本</a>
```

2、下载链接，即目标文档为下载资源，语法如：

```
<a href="DAY02.zip">下载</a>
```

3、电子邮件链接，用于链接到 email，语法如：

```
<a href="mailto:tarena@tarena.com.cn">联系我们</a>
```

4、空链接，用于返回页面顶部，语法如：

```
<a href="#">...</a>
```

5、链接到 JavaScript，以实现特定的代码功能，语法如：

```
<a href="javascript : ...">JS 功能</a>
```

1.9. 简要描述行内元素和块级元素的区别。

块级元素的前后都会自动换行，如同存在换行符一样。默认情况下，块级元素会独占一

行。例如，<p>、<h1>、<div> 都是块级元素。在显示这些元素中间的文本时，都将从新行中开始显示，其后的内容也将在新行中显示。

行内元素可以和其他行内元素位于同一行，在浏览器中显示时不会换行。例如，<a>、 等。对于行内元素，不能设置其高度和宽度。

还有一种元素，为行内块级元素，比如 、<input> 元素等。这些元素可以和其他行内元素位于同一行，同时可以设置其高度和宽度。

1.10. 表单向服务器提交数据有几种方式？这些方式有什么区别？

将表单数据发送给服务器的常用方式有两种：Get 和 Post。

浏览器发送给服务器的 HTTP 请求分为请求头(header)和请求主体(body)两部分。其中，必须包含头部分，用于指定发送请求的方式、目的地以及其他关键信息；而主体是可选的。在头数据和主体数据之间用一个空白行来隔开。

比如，需要发送请求到页面 GetStockPrice.php，且需要附带数据 Symbol=MSFT。那么如果使用 Get 方式发送数据，则简化后的请求数据内容如下所示：

```
GET /Trading/GetStockPrice.aspx?Symbol=MSFT HTTP/1.1
Host: localhost
```

如果使用 Post 方式发送数据，则简化后的请求数据内容如下所示：

```
POST /Trading/GetStockPrice.aspx HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
Content-Length: 11

Symbol=MSFT
```

由此可见，两种方式的区别主要在于发送数据方式不同。

使用 Get 方式向服务器发送表单数据时，表单数据将附加在 URL 属性的末端；采用 POST 方法发送数据时，数据会放置在主体中发送。

2. CSS3 基础

2.1. link 和@import 都可以为页面引入 CSS 文件，其区别是？

将样式定义在单独的.css 的文件里，link 和@import 都可以在 html 页面引入 css 文件。有 link 和@import 两种方式，导入方式如下

link 方式：

```
<link rel="stylesheet" type="text/css" href="aa.css">
```

@import 方式:

```
<style type="text/css">  
  @import "aa.css";  
</style>
```

link 和@import 两种导入 css 文件的区别：

- 1) 祖先的差别。link 属于 XHTML 标签，而@import 完全是 CSS 提供的一种方式。link 标签除了可以加载 CSS 外，还可以做很多其它的事情，比如定义 RSS，定义 rel 连接属性等，@import 就只能加载 CSS 了；
- 2) 加载顺序的差别。当一个页面被加载的时候（就是被浏览者浏览的时候），link 引用的 CSS 会同时被加载，而@import 引用的 CSS 会等到页面全部被下载完再被加载。所以有时候浏览@import 加载 CSS 的页面时开始会没有样式（就是闪烁），网速慢时更为明显；
- 3) 兼容性的差别。由于@import 是 CSS2.1 提出的所以老的浏览器不支持，@import 只有在 IE5 以上的才能识别，而 link 标签无此问题；
- 4) 使用 DOM 控制样式时的差别。当使用 JavaScript 控制 DOM 去改变样式的时候，只能使用 link 标签，因为@import 不是 DOM 可以控制的；
- 5) @import 可以在 css 中再次引入其它样式表，比如可以创建一个主样式表，在主样式表中再引入其他的样式表。

2.2. 如何理解 CSS 样式表的层叠性？

CSS 使用层叠（Cascade）的原则来考虑继承、层叠次序和优先级等重要特征，从而判断相互冲突的规则中哪个规则应该起作用。

继承性是指，许多 CSS 的样式规则不但影响选择器所定义的元素，而且会被这些元素的后代继承。

层叠性是指，当一个 Web 页面使用多个样式表，多个样式表中的样式可层叠为一个。在多个样式表之间所定义的样式没有冲突的时候，浏览器会显示所有的样式。

优先级是指,当发生样式定义冲突时,浏览器首先会按照不同样式规则的优先级来应用样式。CSS 样式的优先级如下所示(其中数字 3 拥有最高的优先权):

1. 浏览器缺省设置;
2. 外部样式表(.css 文件)或者内部样式表(位于 <head> 元素内部);
3. 内联样式(作为某个元素的 style 属性的值)。

同等优先级下,以最后定义的样式为准,important 比内联高。

2.3. 哪些属性可以继承?

CSS 中可以继承的属性如下:

- 1) 文本相关属性:font-family、font-size、font-style、font-variant、font-weight、font、letter-spacing、line-height、text-align、text-indent、text-transform、word-spacing、color;
- 2) 列表相关属性:list-style-image、list-style-position、list-style-type、list-style;
- 3) 表格相关属性:border-collapse、border-spacing、caption-side、table-layout;
- 4) 其他属性:Cursor、visibility。

2.4. CSS 选择器中,元素选择器和类选择器的区别是什么?

元素选择器是最常见的 CSS 选择器,即,文档的元素就是最基本的选择器。选择器通常是某个 HTML 元素,比如 <p>、<h1>、、<a>等,甚至可以是 <html> 元素本身。

类选择器用于将样式规则与附带 class 属性的元素匹配,其中该 class 属性的值为类选择器中指定的值。使用类选择器时,首先需要定义样式类,其语法为:

```
.className { }
```

所有能够附带 class 属性的元素都可以使用此样式声明。只需要将 class 属性的值设置为“className”,则可以将类选择器的样式与元素关联。

在实际使用时,如果需要为某种元素定义样式,则往往使用元素选择器;如果要应用样式而不考虑具体设计的元素,最常用的方法就是使用类选择器。

2.5. 简要描述 CSS 中的定位机制。

CSS 中,除了默认的流定位方式以外,还有如下几种定位机制:浮动定位、相对定位、绝对定位和固定定位。

浮动定位是指将元素排除在普通流之外,并且将它放置在包含框的左边或者右边,但是依旧位于包含框之内。

相对定位将元素相对于它在普通流中的位置进行定位。

绝对定位是指将元素的内容从普通流中完全移除,并且可以使用偏移属性来固定该元素的位置。

固定定位是指将元素的内容固定在页面的某个位置。

2.6. display 属性 和 visibility 属性的区别？

可以使用 display 属性定义建立布局时元素生成的显示框类型。

1. 如果将 display 属性设置为 block,可以让行内元素(比如 <a> 元素)表现得像块级元素一样;

2. 如果将 display 属性设置为 inline,可以让块级元素(比如 <p> 元素)表现得像内联元素一样;

3. 可以通过把 display 属性设置为 none,让生成的元素根本没有框。这样的话,该框及其所有内容就不再显示,不占用文档中的空间。

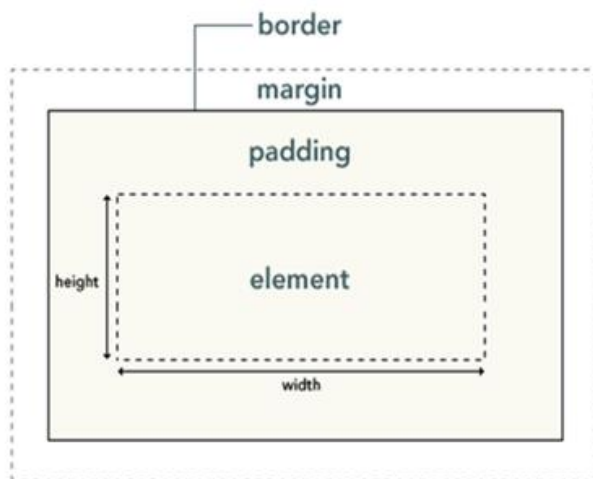
在 DIV 设计中,使用 display:none 属性后,HTML 元素(对象)的宽度、高度等各种属性值都将“丢失”;而使用 visibility:hidden 属性后,HTML 元素(对象)仅仅是在视觉上看不到(完全透明),而它所占据的空间位置仍然存在,也即是说它仍具有高度、宽度等属性值。

2.7. 简述对 CSS 的盒子模型理解？

CSS 盒子模型也叫做框模型,具备内容(content)、填充(padding)、边框(border)、边界(margin)这些属性。在 CSS 中,每一个元素都被视为一个框,而每个框都有三个属性:

- border: 元素的边框(可能不可见),用于将框的边缘与其他框分开;
- margin: 外边距,表示框的边缘与相邻框之间的距离,也称为页边空白;
- padding: 内边距,表示框内容和边框之间的空间。

盒子模型的结构如下图所示:



由上图可以看出，width 和 height 指的是内容区域的宽度和高度。增加内边距、边框和外边距不会影响内容区域的尺寸，但是会增加元素框的总尺寸。

因此，如果在一个具有边框的元素中放置文本，往往需要设置一些内边距，以便文本的边缘不要接触边框，这样更便于阅读。而外边距则可以在多个元素框之间创建空白，避免这些框都挤在一起。因此，在设计页面时，经常会使用 padding 属性和 margin 属性来设置页面的布局。但是，必须注意的是，一旦用了 padding 属性或者 margin 属性设置了元素的边距以后，会增加元素在页面布局中所占的面积。

2.8. 简述 CSS3 中的伪类选择器。

CSS3 提供了大量伪类选择器，浏览器对于有些伪类选择器的支持还不太好。目前，常用的伪类选择器有：

- 目标为类：即 :target，突出显示活动的 HTML 锚，用于选取当前活动的目标元素；
- 元素状态伪类：比如 :enabled、:disabled、:checked；
- 结构伪类：比如 :first-child、:last-child、:empty、:only-child；
- 否定伪类：即 :not(selector)，匹配非指定元素/选择器的每个元素。

2.9. 为什么建议设置背景图像的同时还设置背景颜色？

一般建议在使用背景图像的同时提供 background-color 属性，并且将其设置为和图像主要颜色类似的颜色。这样，如果正在加载页面，或者因为各种原因无法显示背景图像时，页面可以使用这种颜色作为背景色。

2.10. 如何居中 div？如何居中一个浮动元素？

给 div 设置一个宽度，然后设置元素的左右外边距为 auto，比如，margin:0px auto。

则可以实现 div 居中显示。

对于浮动元素，设置其左右外边距为关键字 auto 是无效的。此时，如果需要设置其居中显示，可以：

- 1、精确计算其左右外边距并进行设置，实现居中显示；
- 2、使用一个居中显示的 div 元素包含此浮动元素，代码如下：

```
<div style="margin:0px auto;"><div style="float:left;"></div></div>
```

2.11. 在设置文本的字体时，为什么建议设置替换字体？

可以使用 font-family 属性来指定文本的字体，代码如下所示：

```
font-family:name/inherit;
```

此时，name 为首选字体的名称。如果字体名称有多个单词，即中间有空格，则需要将字体名称用一对单引号或者双引号包围起来。

但是，如果用户机器上并没有安装 name 所指定的字体，则会显示默认字体。因此，如果可以指定一种替代字体，替代字体可以和指定字体不完全相同，相似且不会影响页面的布局，就可以解决问题了。

我们可以为 font-family 属性指定多种字体，且多种字体之间用逗号隔开，这样可以为页面指定一个字体列表。如果用户机器没有第一种字体，则浏览器会查找字体列表中的下一种字体作为替代字体显示。如果找遍了字体列表还是没有可以使用的字体，浏览器才会使用默认字体显示页面。代码如下所示：

```
h1 {font-family: Georgia, serif;}
```

此时，如果用户机器上没有安装 Georgia，但安装了 Times 字体（serif 字体系列中的一种字体），浏览器就可能对 <h1> 元素使用 Times。尽管 Times 与 Georgia 并不完全匹配，但至少足够接近。

因此，我们建议在所有 font-family 规则中都提供一个通用字体系列。这样就提供了一条后路，在用户机器无法提供与规则匹配的特定字体时，就可以选择一个通用字体作为替换。

2.12. 内联元素可以实现浮动吗？

在 CSS 中，任何元素都可以浮动。浮动元素会生成一个块级框，而不论它本身是何种元素。因此，对于内联元素，如果设置为浮动，会产生和块级框相同的效果。

2.13. 什么情况下需要额外设置表格的显示规则？

默认情况下（不额外设置表格的显示规则时），表格按照自动表格布局进行显示，即，浏览器在显示表之前查看每一个单元格，然后基于所有单元格的设置计算表的大小，而列的宽度是由列单元格中没有折行的最宽的内容设定的。此时，单元格的大小会适应内容的大小。

自动表格布局的算法在表格复杂时会比较慢，这是由于它需要在确定最终的布局之前访问表格中所有的内容。在不能提前确定每一列的大小时，这种方式会非常适用。

如果额外设置表格的显示规则，即，设置 `table-layout` 属性的值为 `fixed`，则称为固定表格布局。在固定表格布局中，水平布局仅取决于表格宽度、列宽度、表格边框宽度、单元格间距，而与单元格的内容无关。浏览器将使用某列指定的宽度来计算布局（如果给了宽度的话），并使用该宽度计算该列中所有其他单元格的宽度。

固定表格布局与自动表格布局相比，允许浏览器更快地对表格进行布局。因为如果指定使用固定表格布局，浏览器在接收到第一行后就可以显示表格。如果表格庞大且已经指定了大小，则会加速表的显示。

2.14. 简要描述 CSS 中 `content` 属性的作用。

`content` 属性与 `:before` 及 `:after` 伪元素配合使用，来插入生成内容，可以在元素之前或之后放置生成的内容。可以插入文本、图像、引号，并可以结合计数器为页面元素插入编号。比如，查看如下代码：

```
body {counter-reset:chapter;}
h1:before {content: "第" counter(chapter) "章 "};
h1 {counter-increment:chapter;}
```

使用 `content` 属性，并结合 `:before` 选择器和计数器 `counter`，可以在每个 `<h1>` 元素前插入新的内容。

2.15. CSS Sprite 是什么，谈谈这个技术的优缺点

CSS Sprites 是一种网页图片应用处理方式，就是把网页中一些背景图片整合到一张图片文件中，再利用 CSS 的 `"background-image"`，`"background-repeat"`，`"background-position"` 的组合进行背景定位。

其优点在于：

- 减少网页的 http 请求，提高性能，这也是 CSS Sprites 最大的优点，也是其被广泛传播和应用的主要原因；
- 减少图片的字节 多张图片合并成 1 张图片的字节小于多张图片的字节总和；
- 减少了命名困扰：只需对一张集合的图片命名，不需要对每一个小元素进行命名

名提高制作效率；

- 更换风格方便：只需要在一张或少张图片上修改图片的颜色或样式，整个网页的风格就可以改变，维护起来更加方便。

诚然 CSS Sprites 是如此的强大，但是也存在一些不可忽视的缺点：

- 图片合成比较麻烦；
- 背景设置时，需要得到每一个背景单元的精确位置；
- 维护合成图片时，最好只是往下加图片，而不要更改已有图片。

2.16. 对 CSS3 有了解吗？列举几个 CSS3 的新特性并简要描述

CSS3 作为 CSS 技术的升级版本，着力于模块化发展，将规范分解为一些小的模块，如选择器、盒子模型、列表模块、背景和边框等；并加入了很多新的模块和属性，比如复杂选择器、文字阴影、边框圆角、边框阴影、渐变、过渡、多栏布局、2D/3D 转换、动画等。

其中，CSS3 提供了一些复杂选择器，用于实现页面复杂情况下的元素选择，如属性选择器、一些伪类和伪元素选择器；渐变用于为元素设置渐变效果的背景；转换可以实现元素的变换，比如位移、缩放、旋转等；过渡可以实现简单的动画效果；而动画属性则可以实现复杂的动画，可以实现逐帧制作动画。

2.17. 过渡与动画的区别是什么？

过渡属性 transition 可以在一定的时间内实现元素的状态过渡为最终状态，用于模拟一种过渡动画效果，但是功能有限，只能用于制作简单的动画效果；而动画属性 animation 可以制作类似 Flash 动画，通过关键帧控制动画的每一步，控制更为精确，从而可以制作更为复杂的动画。

2.18. 什么是 CSS reset？

CSS reset，又叫做 CSS 重写或者 CSS 重置，用于改写 HTML 标签的默认样式。

有些 HTML 标签在浏览器里有默认的样式，例如 p 标签有上下边距，li 标签有列表标识符号等。这些默认样式在不同浏览器之间也会有差别，例如 ul 默认带有缩进的样式，在 IE 下，它的缩进是通过 margin 实现的，而 Firefox 下，它的缩进是由 padding 实现的。这必然会带来浏览器兼容问题。

因此，在 CSS 代码中，可以使用 CSS 代码去掉这些默认样式，即重新定义标签样式，从而覆盖浏览器的 CSS 默认属性，即 CSS reset。

需要注意的是，在进行样式重写时，不建议使用 * 选择器进行重写，这样会降低效率，影响性能。

2.19. 如何清除浮动元素所带来的影响？

浮动定位是指将元素排除在普通流之外，并且将它放置在包含框的左边或者右边，但是依旧位于包含框之内。也就是说，浮动的框可以向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止。

由于浮动框不在文档的普通流中，所以元素浮动之后，其原有位置不再保留，其他元素的位置会受到影响。

如果需要清除左侧或者右侧浮动元素带来的影响，则可以使用 `clear` 属性来设置。另外，包含框内的子元素浮动后，如果包含框没有设置具体的高度，则其高度会发生变化，此时，可以使用 `overflow` 属性来清除子元素浮动后带来的影响。

2.20. 谈谈你对浏览器兼容性问题的理解

浏览器的类型及版本的不同会造成 CSS 效果不尽相同，因此需要实现浏览器兼容，也可以针对不同的浏览器编写不同的 CSS。

目前，各主流浏览器的新版本，对于 W3C 的标准支持很好，因此，首先保证代码符合 W3C 的标准，这是解决浏览器兼容问题的前提。

其次，对于某些支持受限的属性，针对不同的浏览器添加相应的前缀，比如 `-webkit-`、`-o-`、`-moz-`。

第三，对于 IE 的低版本，可以编写带有特定前缀的代码，实现版本识别。比如：

```
.bb{
    background-color:#f1ee18;/*所有识别*/
    .background-color:#00deff\9; /*IE6、7、8 识别*/
    +background-color:#a200ff;/*IE6、7 识别*/
    _background-color:#1e0bd1;/*IE6 识别*/
}
```

另外，对于特定的兼容性问题，特殊解决。常见的特殊问题有：

- 1、使用 CSS reset :对于有些 HTML 标签，浏览器默认的 margin 和 padding 不同，可以使用 CSS 代码改写默认的样式效果，从而实现统一；
- 2、IE 低版本中，不能使用 `auto` 关键字实现块级元素居中显示，可以改用设置父元素的 `text-align`；
- 3、子元素设置上外边距时，父元素需要设置边框或者外边距；
- 4、外边距合并问题。