课程简介

讲师：李立超

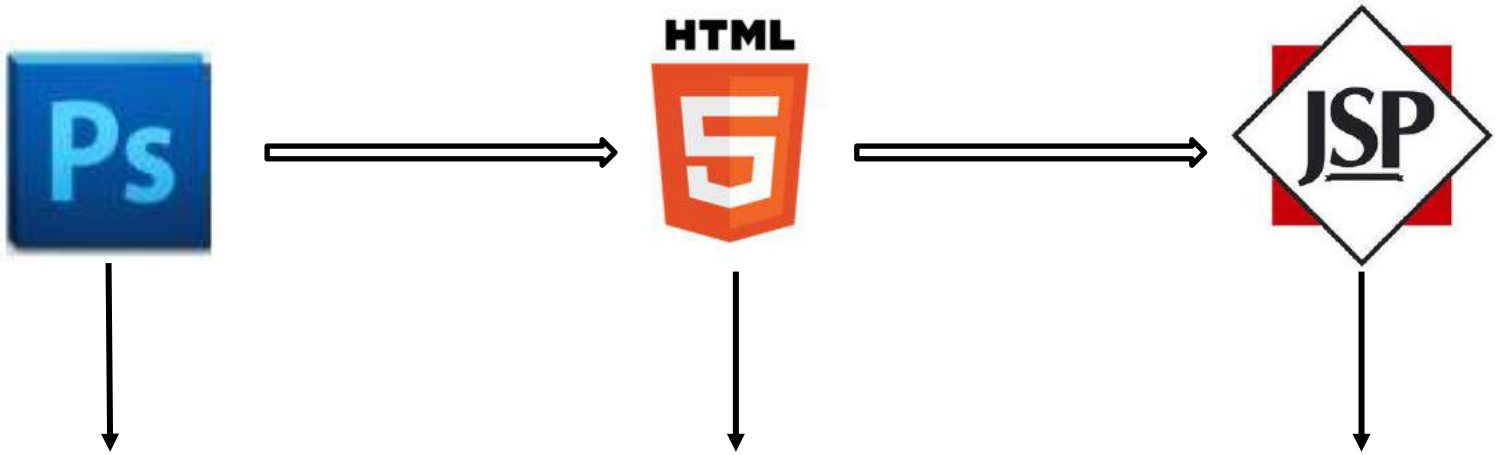
做什么？

- 其实不管是前端工程师还是后台工程师我们要做的工作无非就是软件的开发。
- 软件主要分两种架构C/S和B/S。
- 我们主要从事的是B/S的软件的开发。

什么是B/S？

- B/S中的B指的是browsers，是浏览器的意思，S值Server指服务器的意思。
- B/S架构的软件一般都是通过访问一个网页的形式来使用的，而将一些运算等操作放到远端的服务器上。
- 这样就降低了对客户端的要求，我们的计算机上只需要安装一个浏览器即可使用。
- 像我们常用的京东、taobao、12306等这些网站都是B/S架构的软件。

软件开发流程



网页设计师根据需求设计网页

前端工程师将设计做成静态网页

后台工程师将静态网页修改为动态网页

设计师的网页往往是这样的……

而我们需要把它变成这样的……

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" class="">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>尚硅谷【官网】0费用java培训|不交1分钱java培训,起薪10042</title>
<meta name="keywords" content="java培训,北京Java培训,java就业培训,java课程,java视频教程" />
<meta name="description" content="尚硅谷java培训革命者，真正0费用Java培训，保证就业薪资5k-1w以上，向java培训机构潜规则宣战!比就
业、拼课程、论口碑，尚硅谷Java培训谁与争锋!" />
<link href="main.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="js/jquery-1.7.2.min.js"></script>
<script type="text/javascript" src="js/easySlider1.5.js"></script>
<script src="js/jquery.superslide.2.1.1.js" type="text/javascript"></script>
<script language="javascript" type="text/javascript" src="js/mainbanner.js"></script>
<script language="javascript" type="text/javascript" src="js/jiuye.js"></script>
<script src="js/jquery.KinSlideshow-1.2.1.min.js" type="text/javascript"></script>
<script type="text/javascript">
$(function(){
    $("#KinSlideshow").KinSlideshow();
})
</script>
</head>
<body>
<div class="RO_banner" style="margin:0 auto; width:1000px; height:60px;" ><a href="http://www.atguigu.com/news1203.shtml" target
="_blank" title="java培训"><img src="pics/sinaaddjava.jpg" alt="java培训 java视频" border=0/></a> </div>
<div class="header">
    <div id="top">
    <div class="whole">
        <div class="l"><a href="http://www.atguigu.com/">Java培训</a> | <a href="http://Android.atguigu.com/">Android培训</a> |
            <a href="http://web.atguigu.com/">HTML5培训</a> | <a href="http://www.atguigu.com/download.shtml">Java视频教程</a> | <a
            href="http://ke.atguigu.com/" target="_blank">免费在线课堂</a>
        </div>
        <div class="r">
        <div class="zi">
        <a href="http://www.atguigu.com/contant.shtml">联系我们</a> |
        </div>
```

现在你知道了吗？

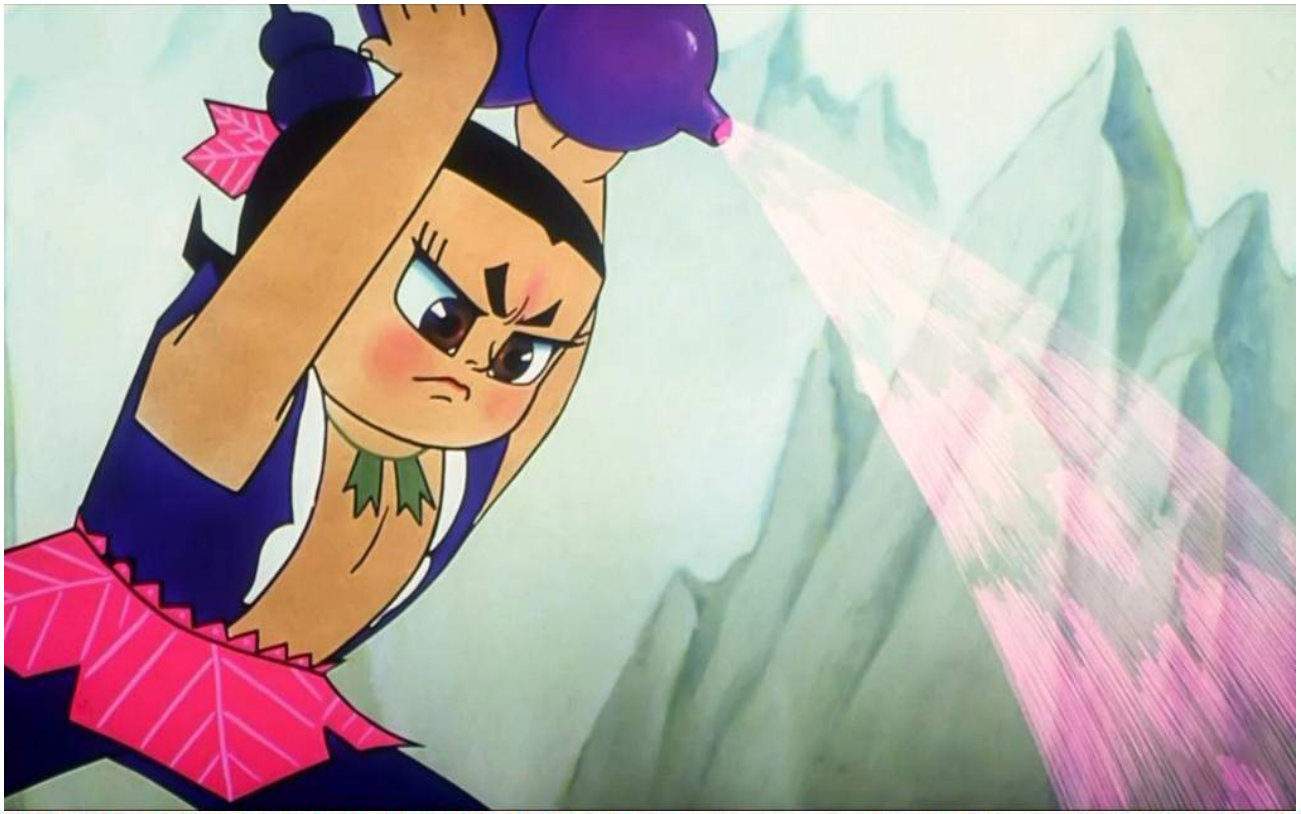
- 我们需要将设计师的设计转换为代码，然后交给后台工程师，在由他们去编写服务器的代码。
- 我们需要和设计师沟通，需要和产品经理沟通，需要和后台工程师沟通。
- 我们的编写的网页会在整个项目的最前端由用户查看。

前端技术好学吗？

- 前端技术简单好学，其实这是我们的一个误区。
- 首先，可以肯定的是前端技术不像Java那样有着较高的门槛。它入门很容易，so easy。
- 但是，刚才也说道了，前端工程师需要和设计师和后台工程师做衔接，这两方面技术我们都需要懂一些。
- 再来，前端技术虽然入门简单，但是深入起来也不是随便谁都能玩好的。所以学习前端技术必须要努力、努力、再努力。你准备好了吗？

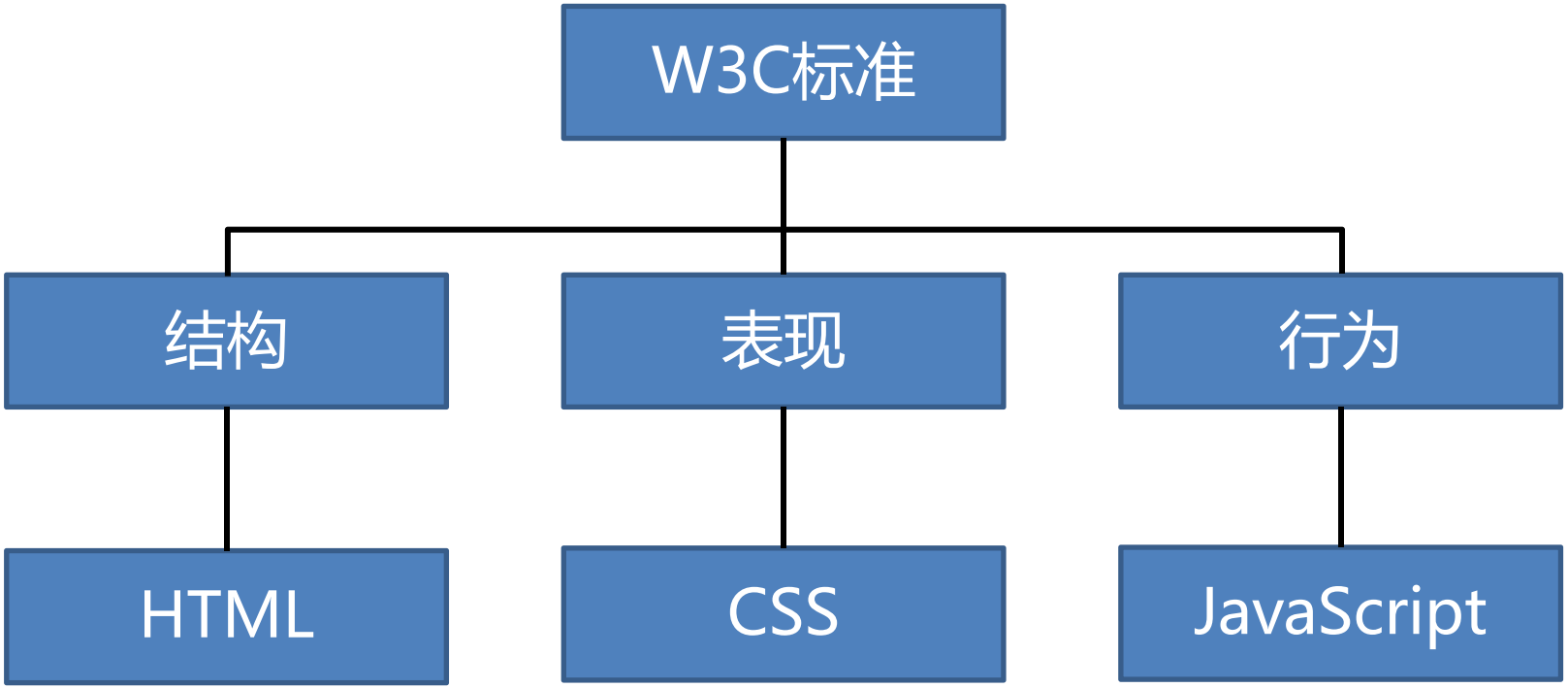
所以…

- 选择了做程序员，我们就要不断的学习，不断的吸取知识。



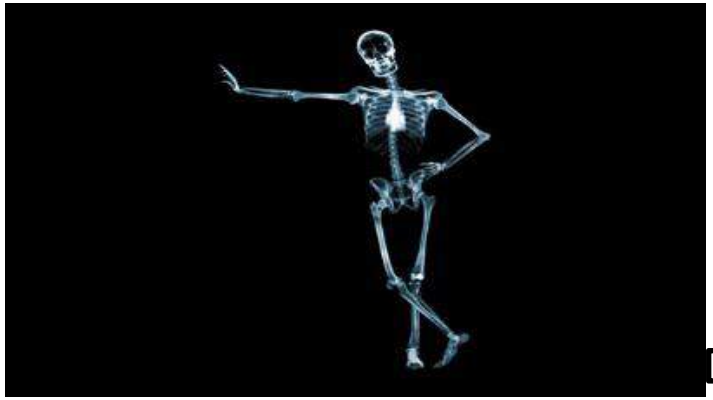
我们主要学习哪些内容？

- 根据W3C标准，一个网页主要由三部分组成：结构、表现还有行为。



什么是结构、表现、行为

- 结构
  - HTML用于描述页面的结构
- 表现
  - CSS用于控制页面中元素的样式
- 行为
  - JavaScript用于响应用户操作



HTML负责页面的结构

CSS负责页面的样式，美化页面

JavaScript负责页面的行为

这一阶段我们学什么？

- 这一阶段我们来学习HTML和CSS的基础知识，也就是我们所说的so easy的那部分。
- 主要内容有：
  - HTML
  - CSS
  - 网页布局
  - JavaScript

我们要用到哪些工具？

- 学习HTML和CSS开发我们不需要太复杂的工具有其是前一阶段，我们主要使用的工具有：
- 浏览器：
  - 火狐、IE、Chrome
- 编辑器
  - 记事本、NotePad++、HBuilder
- 图片工具
  - Photoshop

万维网联盟（W3C）

- 万维网联盟World Wid Web Consortium。
- W3C专门为了定义网页相关的标准而成立。
- W3C定义了网页中的HTML、CSS、DOM、HTTP、XML等标准。

WHATWG

- 网页超文本应用技术工作小组（WHATWG）
- 是一个以推动网络HTML 5 标准为目的而成立的组织。在2004年，由Opera、Mozilla基金会和苹果这些浏览器厂商组成。


尚硅谷
www.atguigu.com

HTML和CSS简介

讲师：李立超

问HTML为何物，其实就是标记语言

HTML

HTML

- HTML（Hypertext Markup Language）超文本标记语言。
- 它负责网页的三个要素之中的结构。
- HTML使用标签的的形式来标识网页中的不同组成部分。
- 所谓超文本指的是超链接，使用超链接可以让我们从一个页面跳转到另一个页面。

- 一个最基本的HTML页面：

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>网页标题</title>
</head>
<body>
    <h1>网页正文</h1>
</body>
</html>
```

标签

- HTML中的标记指的就是标签。
- HTML使用标记标签来描述网页。
- 结构：

<标签名>标签内容</标签名>

<标签名 />

元素

- 我们还将一个完整的标签称为元素。
- 这里我们可以将元素和标签认为是一个同义词。

<h1>一级标题</h1>

上边的h1我们就称为元素

我是一个段落

p也是一个元素，em是p的子元素，p是em的父元素。

```
<body>
    <p><em>内容</em></p>
</body>
```

- body也是一个元素。
- body是p和em的祖先元素。
- p和em是body的后代元素。

属性

- 可以为HTML标签设置属性。
- 通过属性为HTML元素提供附加信息。
- 属性需要设置在开始标签或自结束标签中。
- 属性总是以名称/值对的形式出现。
- 比如：name= “value”
- 有些属性可以是任意值，有些则必须是指定值。

```
<h1 title="我是一个标题">标题</h1>
<img src="" alt="" />
```

常见属性

- id
  - id属性作为标签的唯一标识，在同一个网页中不能出现相同的id属性值。
- class
  - class属性用来为标签分组，拥有相同class属性的标签我们认为就是一组，可以出现相同的class属性，可以为一个元素指定多个class。
- title
  - title属性用来指定标签的标题，指定title以后，鼠标移入到元素上方时，会出现提示文字。

注释

- HTML注释中的内容不会在网页中显示。
- 格式:

  <!-- 注释内容 -->
- 合理的使用注释可以帮助开发人员理解网页的代码。
- 注释不能嵌套！

HTML从哪来，又会到哪去

HTML的发展

HTML的发展

- 1993年6月：HTML第一个版本发布。
- 1995年11月：HTML2.0
- 1997年1月：HTML3.2（W3C推荐）
- 1999年12月：HTML4.01（W3C推荐）
- 2000年底：XHTML1.0（W3C推荐）
- 2014年10月：HTML5（W3C推荐）

doctype

- HTML总共有那么多的版本，而且这其中至少有三个版本在广泛使用，那么浏览器怎么知道我们在使用哪个版本呢？
- 为了让浏览器知道我们使用的HTML版本我们还需要在网页的最上边添加一个doctype声明，来告诉浏览器网页的版本。

html4

- 过渡版

  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">

- 严格版

  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">

- 框架集

  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
  "http://www.w3.org/TR/html4/frameset.dtd">

xhtml1.0

- 过渡版

  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "
  http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

- 严格版

  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

- 框架集

  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">

html5

- 我们会发现html4.01和xhtml的文档声明十分的麻烦。不过不用担心，以上的内容都不是我们使用的，我们使用的是html5的文档声明，而且非常简单：

```
<!DOCTYPE html>
```

怪异模式

- 为了兼容一些旧的页面，浏览器中设置了两种解析模式：
  - 标准模式（Standards Mode）
  - 怪异模式（Quirks Mode）
- 怪异模式解析网页时会产生一些不可预期的行为，所以我们应该避免怪异模式的出现。
- 避免的最好方式就是在页面中编写正确的doctype。

唉唉唉~ 我写的中文怎么都变成鸟语了！

编码问题

编码问题

- 在计算机的内部，文件都是以二进制编码保存的。
- 所谓的二进制编码就是指1和0，也就是我们的所有内容都需要转换为1和0。
- 中国两个字在计算机的底层保存的可能要转换为10100101这种二进制码，这一过程称为编码。
- 计算机在读取文件时需要将10100101在转换为中国给我们显示这一过程称为解码。

字符集

- 这就带来一个问题，中国到底是10100101还是01011010到底由谁说了算。
- 所以我们还需要一个东西称为字符集，字符集规定了如何将文本转换为二进制编码。
- 常见的字符集：ASCII、ISO8859-1、GBK、GB2312、UTF-8。

乱码

- 如果我们保存文件时使用的是utf-8进行编码，而浏览器读取页面时使用gb2312，这样就会导致页面中的内容不能正常显示，也就是我们所说的乱码。
- 所以我们只需要统一两者使用的字符集就可以解决乱码问题。
- 这里为了页面有更好的使用性，我们一般使用utf-8。

解决

- 保存文件的编码我们直接通过编辑器即可指定，接下来就是要告诉浏览器使用什么字符集去解析文件。
- 在html5中只需要使用meta标签即可完成这个任务：

```
<meta charset="utf-8" />
```

<meta>

- 作用：
  - <meta> 标签可提供有关页面的元信息，比如针对搜索引擎和更新频度的描述和关键词。
  - <meta> 标签位于文档的头部，不包含任何内容。<meta> 标签的属性定义了与文档相关联的名称/值对。

这么多标签都是干啥的啊？

常用标签

<html>

- 作用：
  - <html>标签用于告诉浏览器这个文档中包含的信息是用HTML编写的。
- 用法：
  - 所有的网页的内容都需要编写到html标签中，一个页面中html标签只能有一个。
  - html标签中有两个子标签head和body 。

<head>

- 作用：
  - <head>标签用来表示网页的元数据，head中包含了浏览器和搜索引擎使用的其他不可见信息。
- 用法：
  - head标签作为html标签的子元素的出现，一个网页中只能有一个head。

<title>

- 作用：
  - <title>标签表示网页的标题，一般会在网页的标题栏上显示。
  - title标签中的文字，是页面优化的最重要因素。在搜索引擎的搜索时最先看到的、最醒目的内容。
- 用法：
  - 建议将title标签紧贴着head标签编写，这样搜索引擎可以快速检索到标题标签。
  - 网站中的多个页面的title也不应该重复，这样不利于搜索隐藏检索。

<body>

- 作用：
  - <body>标签用来设置网页的主体，所有在页面中能看到的内容都应该编写到body标签中。
- 用法：
  - body标签作为html的子标签使用。

<h1>~<h6>

- 作用：
  - h1~h6都是网页中的标题标签，用来表示网页中的一个标题，不同的是，从h1~h6重要性越来越低。
  - 标题标签相当于正文的标题，通常认为重要性仅次于页面的title。
  - 一般标题标签我们只会使用到h3，h3以后的标题标签对于搜索引擎就没有什么意义了。
  - 一个页面中只会使用一个h1标签。

<p>

- 作用：
  - <p>标签表示网页中的一个段落。
  - 一般浏览器会在段落的前和后各加上一个换行，也就是段落会在页面中自成一行。

- 作用
  -
标签表示一个换行标签，使用br标签可以使br标签后的内容另起一行。

<hr />

- 作用：
  - <hr />标签是水平线标签，使用hr标签可以在页面中打印一条水平线，水平线可以将页面分成上下两个部分。

- 作用：
  - < img />标签是图片标签，可以用来向页面中引入一张外部的图片。
- 属性：
  - src
    - 指向一个外部的图片的路径。
  - alt
    - 图片的描述

<a>

- 作用：
  - <a>标签是超链接标签，通过a标签，可以快速跳转到其他页面。
- 属性：
  - href
    - 指向一个链接地址
  - target
    - 设置打开目标页面的位置，可选值：_blank新窗口、_self当前窗口。

有序列表

- 使用ol和li来创建一个有序列表。

```
<ol>
    <li>列表项1</li>
    <li>列表项2</li>
    <li>列表项3</li>
</ol>
```

1. 列表项1
2. 列表项2
3. 列表项3

无序列表

- 使用ul和li来创建一个无序列表。

```
<ul>
    <li>列表项1</li>
    <li>列表项2</li>
    <li>列表项3</li>
</ul>
```

- 列表项1
- 列表项2
- 列表项3

定义列表

- 使用dl、dd、dt来创建一个定义列表。

```
<dl>
    <dt>定义项1</dt>
    <dd>定义描述1</dd>
    <dt>定义项2</dt>
    <dd>定义描述2</dd>
    <dt>定义项3</dt>
    <dd>定义描述3</dd>
</dl>
```

定义项1
    定义描述1
定义项2
    定义描述2
定义项3
    定义描述3

HTML这么厉害，但是多写几个空格就不行了呢！

实体（转义字符）

实体

- 在HTML中预留了一些字符。
- 这些预留字符是不能在网页中直接使用的。
- 比如<和>,我们不能直接在页面中使用<和>号，因为浏览器会将它解析为html标签。
- 为了可以使用这些预留字符，我们必须在html中使用字符实体。
- 语法: &实体名;

字符实体

- 小于号<
  - <
- 大于号>
  - >
- 空格
  -
- 和符号&
  - &
- 版权©
  - ©
- 引号”
  - "
- 注册商标®
  - ®
- 商标™
  - ™

工欲善其事必先利其器！

开发工具

文本编辑器

- 在windows中我们只需要使用最简单的记事本就可以完成所有的网页的开发。
- 但是一般我们会使用一些具有提示功能的纯文本编辑器：
  - Notepad++(免费)
  - Sublime(收费)
- 当然还有很多其他的工具。

IDE

- IDE（集成开发工具）
- IDE拥有比纯文本编辑器更加强大的提示功能，也是我们开发中用的比较多的工具。
  - DreamWeaver（收费）
  - WebStorm（收费）
  - Hbuilder（免费）
- 当然也有其他的IDE。

工具的选择

- 上边说了那么多工具我们要使用哪个呢？
- 其实使用哪个工具都不重要，我们也不用费劲心机去讨论工具的好坏，找一个自己喜欢用的即可。
- 而且我们也要做到不依赖于某一个工具，我们要做到，即使只使用最简单的记事本，我们也可以照常开发。

HTML页面实在是太丑了，怎么破？

CSS

CSS

- 层叠样式表 (Cascading Style Sheets)
- css可以用来为网页创建样式表，通过样式表可以对网页进行装饰。
- 所谓层叠，可以将整个网页想象成是一层一层的结构，层次高的将会覆盖层次低的。
- 而css就可以分别为网页的各个层次设置样式。

基本语法

- CSS的样式表由一个一个的样式构成，一个样式又由选择器和声明块构成。
- 语法：
  - 选择器 {样式名:样式值；样式名:样式值；}
  - p {color:red ; font-size:12px;}

行内样式

- 可以直接将样式写到标签内部的style属性中，这种样式不用填写选择器，直接编写声明即可。

```
<p style="color: red;font-size: 30px"></p>
```

- 这种方式编写简单，定位准确。但是由于直接将css代码写到了html标签的内部，导致结构与表现耦合，同时导致样式不能够复用，所以这种方式我们不使用。

内部样式表

- 可以直接将样式写到<style>标签中。

```
<style>
        p{color:red; font-size: 30px;}
</style>
```

- 这样使css独立于html代码，而且可以同时为多个元素设置样式，这是我们使用的比较多的一种方式。
- 但是这种方式，样式只能在一个页面中使用，不能在多个页面中重复使用。

外部样式表

- 可以将所有的样式保存到一个外部的css文件中，然后通过<link>标签将样式表引入到文件中。

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

- 这种方式将样式表放入到了页面的外部，可以在多个页面中引入，同时浏览器加载文件时可以使用缓存，这是我们开发中使用的最多的方式。

选择器

- 选择器（selector），会告诉浏览器：网页上的哪些元素需要设置什么样的样式。
- 比如：p这个选择器就表示选择页面中的所有的p元素，在选择器之后所设置的样式会应用到所有的p元素上。

元素选择器

- 元素选择器（标签选择器），可以根据标签的名字来从页面中选取指定的元素。
- 语法：

  标签名 { }

- 比如p则会选中页面中的所有p标签，h1会选中页面中的所有h1标签。

类选择器

- 类选择器，可以根据元素的class属性值选取元素。
- 语法：

  .className { }

- 比如.hello会选中页面所有class属性为hello的元素。

ID选择器

- ID选择器，可以根据元素的id属性值选取元素。
- 语法：

  #id { }

- 比如#box会选中页面中id属性值为box的元素，和class属性不同，id属性是不能重复的。

复合选择器

- 复合选择器，可以同时使用多个选择器，这样可以选择同时满足多个选择器的元素。
- 语法：
  - 选择器1选择器2{}
- 例如div.box1会选中页面中具有box1这个class的div元素。

群组选择器

- 群组选择器，可以同时使用多个选择器，多个选择器将被同时应用指定的样式。
- 语法：

  选择器1,选择器2,选择器3 { }

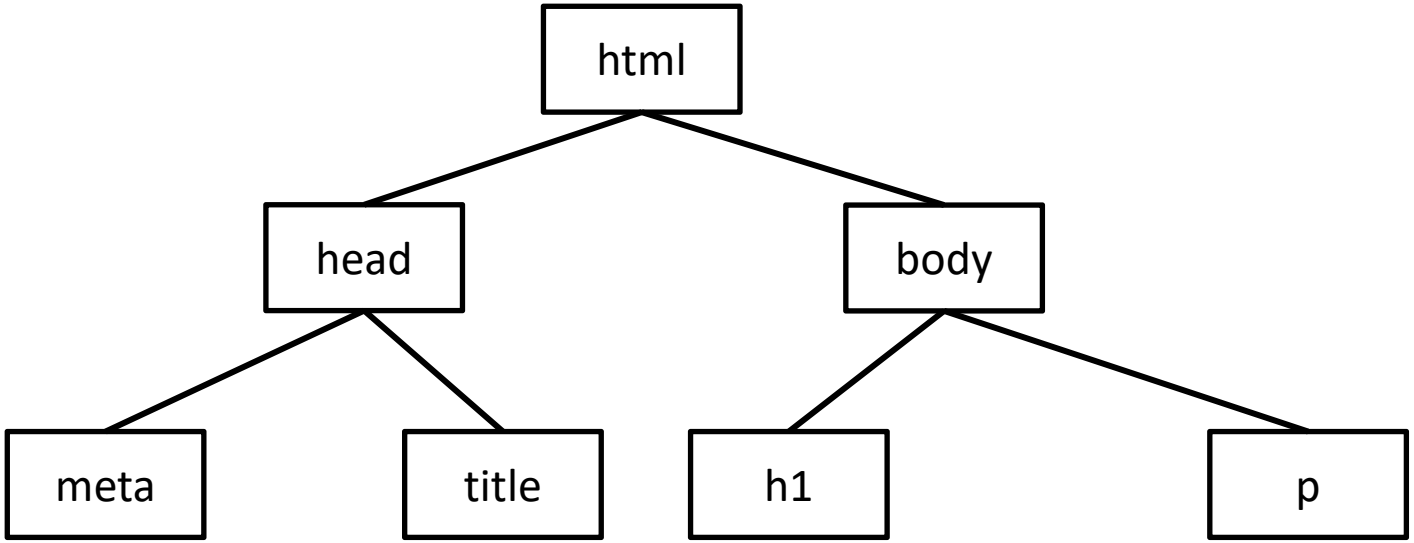
- 比如p,.hello,#box会同时选中页面中p元素，class为hello的元素，id为box的元素。

通用选择器

- 通用选择器，可以同时选中页面中的所有元素。
- 语法：

```
*{ }
```

HTML族谱



标签之间的关系

- 祖先元素
  - 直接或间接包含后代元素的元素。
- 后代元素
  - 直接或间接被祖先元素包含的元素。
- 父元素
  - 直接包含子元素的元素。
- 子元素
  - 直接被父元素包含的元素。
- 兄弟元素
  - 拥有相同父元素的元素。

后代选择器

- 后代选择器可以根据标签的关系，为处在元素内部的代元素设置样式。
- 语法：

  祖先元素 后代元素 后代元素 { }
- 比如p strong 会选中页面中所有的p元素内的strong元素。

子元素选择器

- 子元素选择器可以给另一个元素的子元素设置样式。
- 语法：

  父元素 > 子元素{}

- 比如body > h1将选择body子标签中的所有h1标签。

兄弟选择器

- 除了根据祖先父子关系，还可以根据兄弟关系查找元素。
- 语法:
  - 查找后边一个兄弟元素
    - 兄弟元素 + 兄弟元素{}
  - 查找后边所有的兄弟元素
    - 兄弟元素 ~ 兄弟元素{}

伪类和伪元素

- 有时候，你需要选择本身没有标签，但是仍然易于识别的网页部位，比如段落首行或鼠标滑过的连接。CSS为他们提供一些选择器：伪类和伪元素。

给链接定义样式

- 有四个伪类可以让你根据访问者与该链接的交互方式，将链接设置成4种不同的状态。
- 正常链接
  - a:link
- 访问过的链接
  - a:visited（只能定义字体颜色）
- 鼠标滑过的链接
  - a:hover
- 正在点击的链接
  - a:active

继承

- 就像父亲的财产会遗传给儿子一样，在CSS中祖先元素的样式同样也会被子元素继承。
- 继承是指应用在一个标签上的那些CSS样式会同时被应用到其内嵌标签上。
- 比如为父元素设置了字体颜色，子元素也会应用上相同的颜色。
- 当然并不是所有的样式都会被继承，这一点我们讲到具体样式时，再去讨论。

如果一个元素同时满足了多个选择器，哪个样式生效？

选择器的权重

选择器的权重

- 在页面中使用CSS选择器选中元素时，经常都是一个元素同时被多个选择器选中。
- 比如：
  - body h1
  - h1
- 上边的两个选择器都会选择h1元素，如果两个选择器设置的样式不一致那还好不会产生冲突，但是如果两个选择器设置的是同一个样式，这样h1到底要应用那个样式呢？CSS中会默认使用权重较大的样式，权重又是如何计算的呢？

权重的计算

- 不同的选择器有不同的权重值：
  - 内联样式：权重是 1000
  - id选择器：权重是 100
  - 类、属性、伪类选择器：权重是 10
  - 元素选择器：权重是 1
  - 通配符：权重是 0
- 计算权重需要将一个样式的全部选择器相加，比如上边的body h1的权重是20，h1的权重是10，所以第一个选择器设置的样式会优先显示。

单位

- px
  - 如果我们将一个图片放大的话，我们会发现一个图片是有一个一个的小色块构成的，这一个小色块就是一个像素，也就是1px，对于不同的显示器来说一个像素的大小是不同的。
- 百分比
  - 也可以使用一个百分数来表示一个大小，百分比是相对于父元素来说的，如果父元素使用的大小是16px，则100%就是16px，200%就是32px。

颜色

- 在CSS中可以直接使用颜色的关键字来代表一种颜色。
- 17中颜色
  - aqua、black、blue、fuchsia、gray、green、lime、maroon、navy、olive、orange、purple、red、silver、teal、white、yellow。
- 还有147种svg颜色，这里就不一一列举了，但是明显即使这些颜色变成double，也不足以描绘我们世界中所有的颜色。

十六进制颜色

- 用的最多的颜色是十六进制符号。一个颜色值，比如：#6600FF实际上包含了三组十六进制的数字。
- 上边的例子中66代表红色的浓度，00代表绿色的浓度，FF代表了蓝色的浓度。最后的颜色是由这些指定浓度的红绿蓝混合而成的。
- 如果每一组数中的两个数字都相同，就可以把十六进制的数字缩短为只有3个字符，如将#6600FF缩短为#60F。

RGB值

- 也可以使用计算机中常用的RGB值来表示颜色。可以使用0~255的数值，也可以使用0%~100%的百分比数。
  - RGB(100%,0%,0%)
  - RGB(0,255,0)
- 第一个数表示红色的浓度，第二个数表示绿色浓度，第三个数表示蓝色的浓度。


尚硅谷
www.atguigu.com

表单的基本设置

讲师：李立超

表单

- 现实生活中的表单是用来提交信息的，比如：办理银行卡填写的申请表、找工作填写的简历、入学时填写的个人信息表。这些都是表单的一种
- 网页中的表单是用来向服务器提交信息的，我们最常用到的表单就是baidu的搜索框

在搜索框填入关键字后，点击搜索按钮，关键字会提交到baidu的服务器，服务器根据用户输入的关键字进行检索，返回相应信息

表单

- 表单可以将用户填写的信息提交的服务器
- 例子：

```
<form action="1.html" method="get">
    <input type="text" name="name"><br />
    <input type="password" name="pwd"><br />
    <input type="submit" value="提交">
</form>
```

使用<form>标签来创建一个表单
表单中必须要有两个属性action和method
action表示提交表单到服务器中的地址
method表示提交表单的方法

一个表单中可以包含多个表单项

表单项

文本框

```
<input type="text" name="name">
```

密码框

```
<input type="password" name="pwd">
```

多选框

```
<input type="checkbox" name="sports">
```

单选框

```
<input type="radio" name="gender">
```

提交按钮

```
<input type="submit" value="提交">
```

下拉列表

```
<select>
    <option>北京</option>
</select>
```

input

- input是我们使用的最多的表单项，它可以根据不同的type属性呈现不同的状态。
- type属性可选值：
  - text：文本框
  - password：密码框
  - submit：提交按钮
  - radio：单选按钮
  - checkbox：多选框
  - reset ：重置按钮

select、option

- select用于创建一个下拉列表。
- option表示下拉列表中的列表项。
- optgroup用于为列表项分组。

textarea

- textarea用来创建一个文本域，实际效果和文本框类似，只是可以输入多行数据。
- 可用属性：
  - cols：文本域的列数
  - rows：文本域的行数

fieldset、legend、label

- fieldset用来为表单项进行分组。
- legend用于指定每组的名字。
- label标签用来为表单项定义描述文字。


尚硅谷
www.atguigu.com

表格的基本设置

讲师：李立超

表格

- 在Web的历史中，HTML的表格发挥了极大的作用。最初创建表格就是为了以表格的形式显示数据，后来表格变成了一个极受欢迎的布局工具。
- 但是有了CSS以后，CSS在布局网页方面实际上会更出色，所以现在我们使用表格的作用只有一个，就是用来表示格式化的数据。
- HTML中的表格可以很复杂，但是通常情况下我们不需要创建过于复杂的表格。

table、tr、th、td

- 使用table标签创建一个表格。
- tr表示表格中的一行。
- tr中可以编写一个或多个th或td。
- th表示表头。
- td表示表格中的一个单元格。

caption、thead、tbody、tfoot

- caption表示表格的标题。
- thead表示表格的头部。
- tbody表示表格的主体。
- tfoot表示表格的底部。

合并单元格

- 合并单元格指将两个或两个以上的单元格合并为一个单元格。
- 合并单元格可以通过在th或td中设置属性来完成。
- 横向合并
  - colspan
- 纵向合并
  - rowspan

表格的样式

- 之前学习的很多属性都可以用来设置表格的样式，比如color可以用来设置文本的颜色。padding可以设置内容和表格边框的距离。
- text-align：设置文本的水平对齐。
- vertical-align：设置文本的垂直对齐。
  - 可选值：top、baseline、middle、bottom
- border-spacing：边框间距
- border-collapse：合并边框
  - collapse：合并边框
  - separate：不合并边框


尚硅谷
www.atguigu.com

盒子模型

讲师：李立超

在网页中“一切皆是盒子”

盒子模型

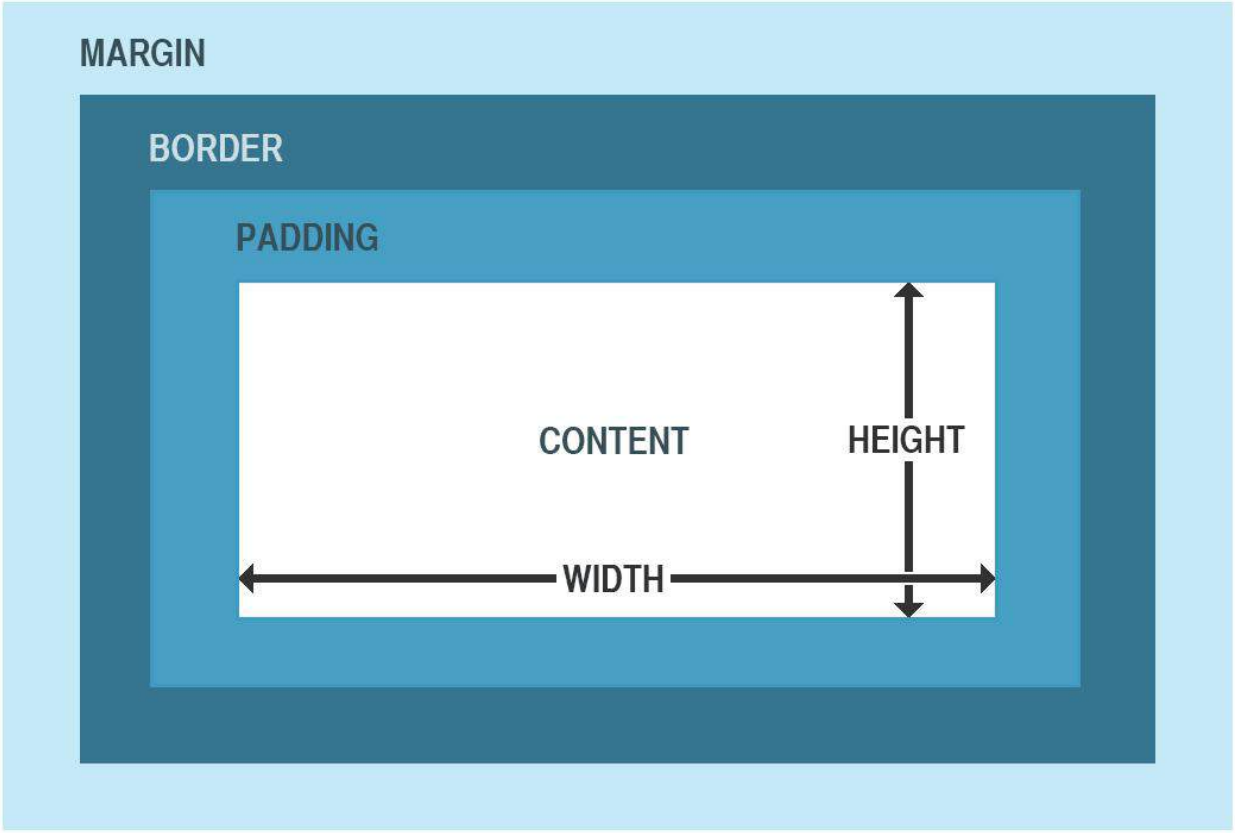
盒子

- CSS处理网页时，它认为每个元素都包含在一个不可见的盒子里。
- 为什么要想象成盒子呢？因为如果把所有的元素都想象成盒子，那么我们对网页的布局就相当于是摆放盒子。
- 我们只需要将相应的盒子摆放到网页中相应的位置即可完成网页的布局。

盒子模型

- 一个盒子我们会分成几个部分：
  - 内容区(content)
  - 内边距(padding)
  - 边框(border)
  - 外边距(margin)

盒子模型



内容区

- 内容区指的是盒子中放置内容的区域，也就是元素中的文本内容，子元素都是存在于内容区中的。
- 如果没有为元素设置内边距和边框，则内容区大小默认和盒子大小是一致的。
- 通过width和height两个属性可以设置内容区的大小。
- width和height属性只适用于块元素。

内边距

- 顾名思义，内边距指的就是元素内容区与边框以内的空间。
- 默认情况下width和height不包含padding的大小。
- 使用padding属性来设置元素的内边距。
- 例如：
  - padding:10px 20px 30px 40px
  - 这样会设置元素的上、右、下、左四个方向的内边距。

内边距

- padding:10px 20px 30px;
  - 分别指定上、左右、下四个方向的内边距
- padding:10px 20px;
  - 分别指定上下、左右四个方向的内边距
- padding:10px;
  - 同时指定上左右下四个方向的内边距
- 同时在css中还提供了padding-top、padding-right、padding-right、padding-bottom分别用来指定四个方向的内边距。

边框

- 可以在元素周围创建边框，边框是元素可见框的最外部。
- 可以使用border属性来设置盒子的边框：
  - border:1px red solid;
  - 上边的样式分别指定了边框的宽度、颜色和样式。
- 也可以使用border-top/left/right/bottom分别指定上右下左四个方向的边框。
- 和padding一样，默认width和height并包括边框的宽度。

边框的样式

- 边框可以设置多种样式：
  - none（没有边框）
  - dotted（点线）
  - dashed（虚线）
  - solid（实线）
  - double（双线）
  - groove（槽线）
  - ridge（脊线）
  - inset（凹边）
  - outset（凸边）

外边距

- 外边距是元素边框与周围元素相距的空间。
- 使用margin属性可以设置外边距。
- 用法和padding类似，同样也提供了四个方向的margin-top/right/bottom/left。
- 当将左右外边距设置为auto时，浏览器会将左右外边距设置为相等，所以这行代码margin:0 auto可以使元素居中。

display

- 我们不能为行内元素设置width、height、margin-top和margin-bottom。
- 我们可以通过修改display来修改元素的性质。
- 可选值：
  - block：设置元素为块元素
  - inline：设置元素为行内元素
  - inline-block：设置元素为行内块元素
  - none：隐藏元素（元素将在页面中完全消失）

visibility

- visibility属性主要用于元素是否可见。
- 和display不同，使用visibility隐藏一个元素，隐藏后其在文档中所占的位置会依然保持，不会被其他元素覆盖。
- 可选值：
  - visible：可见的
  - hidden：隐藏的

overflow

- 当相关标签里面的内容超出了样式的宽度和高度是，就会发生一些奇怪的事情，浏览器会让内容溢出盒子。
- 可以通过overflow来控制内容溢出的情况。
- 可选值：
  - visible：默认值
  - scroll：添加滚动条
  - auto：根据需要添加滚动条
  - hidden：隐藏超出盒子的内容

文档流

- 文档流指的是文档中可现实的对象在排列时所占用的位置。
- 将窗体自上而下分成一行行，并在每行中按从左至右的顺序排放元素，即为文档流。
- 也就是说在文档流中元素默认会紧贴到上一个元素的右边，如果右边不足以放下元素，元素则会另起一行，在新的一行中继续从左至右摆放。
- 这样一来每一个块元素都会另起一行，那么我们如果想在文档流中进行布局就会变得比较麻烦。

浮动

- 所谓浮动指的是使元素脱离原来的文本流，在父元素中浮动起来。
- 浮动使用float属性。
- 可选值：
  - none：不浮动
  - left：向左浮动
  - right：向右浮动
- 块级元素和行内元素都可以浮动，当一个行内元素浮动以后将会自动变为一个块级元素。
- 当一个块级元素浮动以后，宽度会默认被内容撑开，所以当漂浮一个块级元素时我们都会为其指定一个宽度。

浮动

- 当一个元素浮动以后，其下方的元素会上移。元素中的内容将会围绕在元素的周围。
- 浮动会使元素完全脱离文本流，也就是不再在文档中在占用位置。
- 元素设置浮动以后，会一直向上漂浮直到遇到父元素的边界或者其他浮动元素。
- 元素浮动以后即完全脱离文档流，这时不会再影响父元素的高度。也就是浮动元素不会撑开父元素。
- 浮动元素默认会变为块元素，即使设置display:inline以后其依然是个块元素。

清除浮动

- clear属性可以用于清除元素周围的浮动对元素的影响。
- 也就是元素不会因为上方出现了浮动元素而改变位置。
- 可选值：
  - left：忽略左侧浮动
  - right：忽略右侧浮动
  - both：忽略全部浮动
  - none：不忽略浮动，默认值

定位

- position属性可以控制Web浏览器如何以及在何处显示特定的元素。
- 可以使用position属性把一个元素放置到网页中的任何位置。
- 可选值：
  - static
  - relative
  - absolute
  - fixed

相对定位

- 每个元素在页面的文档流中都有一个自然位置。相对于这个位置对元素进行移动就称为相对定位。周围的元素完全不受此影响。
- 当将position属性设置为relative时，则开启了元素的相对定位。
- 当开启了相对定位以后，可以使用top、right、bottom、left四个属性对元素进行定位。

相对定位的特点

- 如果不设置元素的偏移量，元素位置不会发生改变。
- 相对定位不会使元素脱离文本流。元素在文本流中的位置不会改变。
- 相对定位不会改变元素原来的特性。
- 相对定位会使元素的层级提升，使元素可以覆盖文本流中的元素。

绝对定位

- 绝对定位指使元素相对于html元素或离他最近的祖先定位元素进行定位。
- 当将position属性设置为absolute时，则开启了元素的绝对定位。
- 当开启了绝对定位以后，可以使用top、right、bottom、left四个属性对元素进行定位。

绝对定位的特点

- 绝对定位会使元素完全脱离文本流。
- 绝对定位的块元素的宽度会被其内容撑开。
- 绝对定位会使行内元素变成块元素。
- 一般使用绝对定位时会同时为其父元素指定一个相对定位，以确保元素可以相对于父元素进行定位。

固定定位

- 固定定位的元素会被锁定在屏幕的某个位置上，当访问者滚动网页时，固定元素会在屏幕上保持不动。
- 当将position属性设置为fixed时，则开启了元素的固定定位。
- 当开启了固定定位以后，可以使用top、right、bottom、left四个属性对元素进行定位。
- 固定定位的其他特性和绝对定位类似。

z-index

- 当元素开启定位以后就可以设置z-index这个属性。
- 这个属性可以提升定位元素所在的层级。
- z-index可以指定一个整数作为参数，值越大元素显示的优先级越高，也就是z-index值较大的元素会显示在网页的最上层。


尚硅谷
www.atguigu.com

事件

讲师：李立超

事件

- 关于事件实际上我们已经初步接触过了，指的就是用户与浏览器交互的一瞬间。
- 我们通过为指定事件绑定回调函数的形式来处理事件，当指定事件触发以后我们的回调函数就会被调用，这样我们的页面就可以完成和用户的交互了。
- 这里我们还要更加深入的聊一聊事件的其他内容。

事件处理程序

- 我们可以通过两种方式为一个元素绑定事件处理程序：
  - 通过HTML元素指定事件属性来绑定
  - 通过DOM对象指定的属性来绑定
- 这两种方式都是我们日常用的比较多的，但是更推荐使用第二种方式。
- 还有一种方式比较特殊我们称为设置事件监听器。使用如下方式：
  - 元素对象.addEventListener()

通过HTML标签的属性设置

- 通过HTML属性来绑定事件处理程序是最简单的方式。

```
<button onclick="alert('hello');alert('world')">按钮</button>
```

- 这种方式当我们点击按钮以后，onclick属性中对应的JS代码将会执行，也就是点击按钮以后，页面中会弹出两个提示框。
- 这种方式我们直接将代码编写到了onclick属性中，可以编写多行js代码，当然也可以事先在外部定义好函数。
- 这种方式的优点在于，设定步骤非常简单，并且能够确保事件处理程序会在载入时被设定。
- 如果在函数的最后return false则会取消元素的默认行为。

通过DOM对象的属性绑定

- 但是其实上面的写法虽然简单，但却将JS和HTML的代码编写到了一起，并不推荐使用，我们更推荐如下的写法：

```
var btn = document.getElementById('btn');
btn.onclick = function(){
    alert("hello");
};
```

- 这种写法将HTML代码和JS写在不同的位置，维护起来更加容易。

设置事件监听器

- 前边两种方式都可以绑定事件处理程序，但是它们都有一个缺点就是都只能绑定一个程序，而不能为一个事件绑定多个程序。
- 这是我们就可以使用addEventListener()来处理，这个方法需要两个参数：一个是事件字符串，一个是响应函数。

```
btn.addEventListener('click' , function(){alert("hello");});
```

- 但是要注意的是ie8以下的浏览器是不支持上边的方法的，需要使用attachEvent代替。
- 也可以使用removeEventListener()和detachEvent()移除事件。

事件处理中的this

- 在事件处理程序内的 this 所引用的对象即是设定了该事件处理程序的元素。
- 也就是事件是给那个对象绑定的this就是哪个对象。

事件对象

- 在DOM对象上的某个事件被触发时，会产生一个事件对象Event，这个对象中包含着所有事件有关的信息。包括导致事件的元素、事件的类型以及其他与特定事件相关的信息。
- 例如，鼠标操作导致的事件对象中，会包含鼠标位置的信息，而键盘操作导致的事件对象中，会包含与按下的键有关的信息。所有浏览器都支持 event 对象，但支持方式不同。

事件对象

- DOM标准的浏览器会将一个event对象传入到事件的处理程序当中。无论事件处理程序是什么都会传入一个event对象。
- 可以通过这种方式获取：

```
btn.onclick = function(event){
    alert(event.type);
};
```

- Event对象包含与创建它的特定事件有关的属性和方法。触发的事件类型不一样，可用的属性和方法也不一样。

Event对象的通用属性/方法

属性/方法	类型	读/写	说明
bubbles	Boolean	只读	事件是否冒泡
cancelable	Boolean	只读	是否可以取消事件的默认行为
currentTarget	Element	只读	当前正在处理的事件元素
defaultPrevented	Boolean	只读	是否调用了preventDefault()
detail	Number	只读	与事件相关的细节信息
eventPhase	Number	只读	阶段 1:捕获 2:目标 3:冒泡
preventDefault()	Function	只读	取消事件的默认行为
stopImmediatePropagation()	Function	只读	取消事件的进一步捕获或冒泡
stopPropagation()	Function	只读	取消事件的进一步捕获或冒泡
target	Element	只读	事件的目标
trusted	Boolean	只读	是否是浏览器内置事件
type	String	只读	被触发的事件的类型

IE中的事件对象

- 与访问 DOM 中的 event 对象不同，要访问 IE 中的 event 对象有几种不同的方式，取决于指定事件处理程序的方法。
- 在IE中event对象作为window对象的属性存在的，可以使用window.event来获取event对象。
- 在使用attachEvent()的情况下，也会在处理程序中传递一个event对象，也可以按照前边的方式使用。

Event对象的通用属性/方法（IE）

属性/方法	类型	读/写	说明
cancelBubble	Boolean	读/写	是否取消冒泡
returnValue	Boolean	读/写	是否执行默认行为
srcElement	Element	只读	事件的目标
type	String	只读	被触发的事件的类型

事件的触发

- 事件的发生主要是由用户操作引起的。
- 比如mousemove这个事件就是由于用户移动鼠标引起的，在鼠标指针移动的过程中该事件会持续发生。
- 当指定事件被触发时，浏览器就会调用对应的函数去响应事件，一般情况下事件没触发一次，函数就会执行一次。
- 因此设置鼠标移动的事件可能会影响到鼠标的移动速度。所以设置该类事件时一定要谨慎。

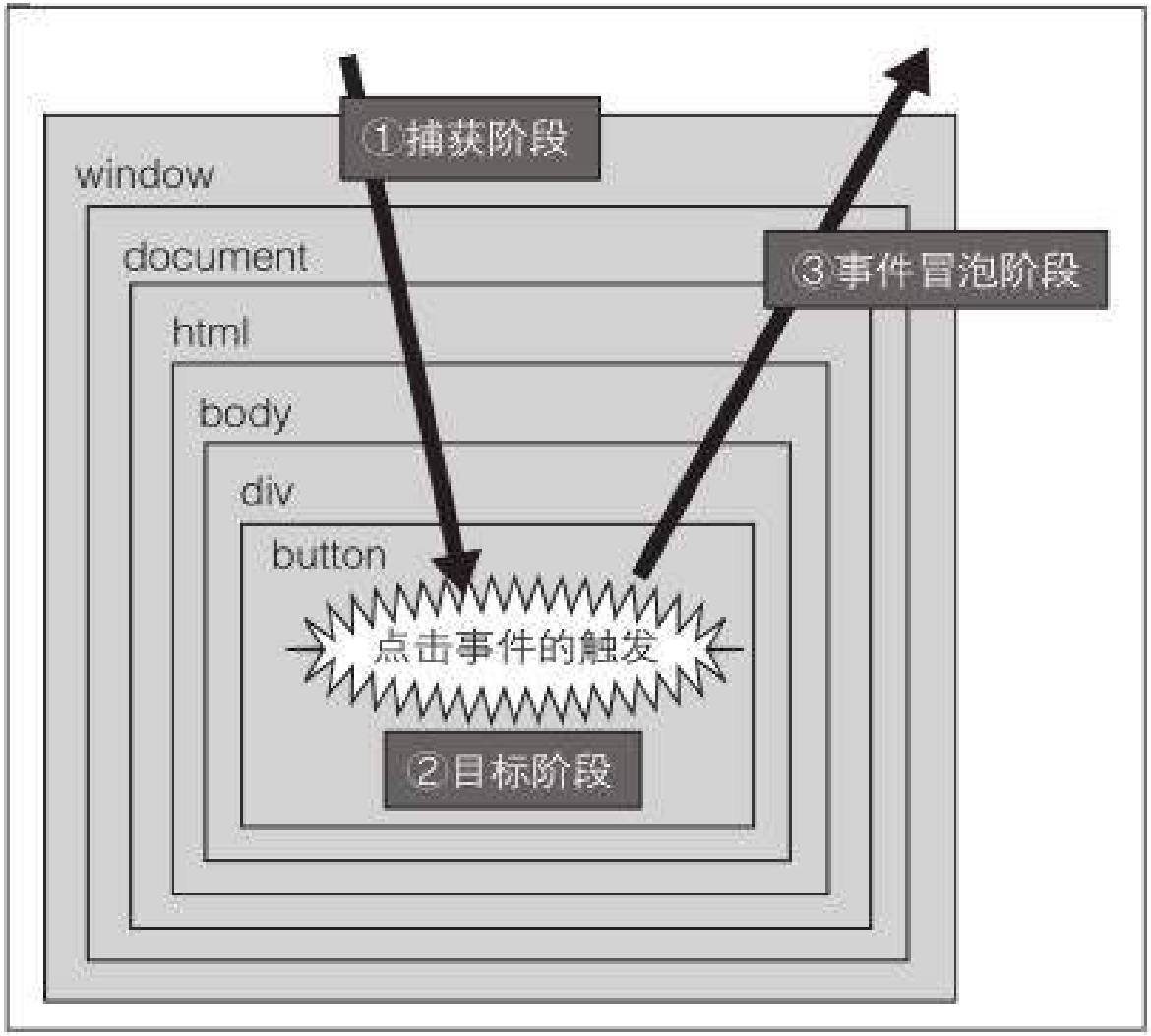
事件的传播

- 在网页中标签与标签之间是有嵌套关系的，比如这样一个页面：

```
<html>
    <body>
        <div id="foo">
                <button id="bar">sample</button>
        </div>
    </body>
</html>
```

- 如果这时用户点击了sample按钮，则会以该按钮作为事件目标触发一次点击事件。
- 这时，事件的处理将会分为捕获阶段、目标阶段、事件冒泡这三个阶段。

事件的传播流程


①捕获阶段
window
document
html
body
div
button
点击事件的触发
②目标阶段
③事件冒泡阶段

事件的传播

- 捕获阶段
  - 这一阶段会从window对象开始向下一直遍历到目标对象，如果发现有对象绑定了响应事件则做相应的处理。
- 目标阶段
  - 这一阶段已经遍历结束，则会执行目标对象上绑定的响应函数。
- 事件冒泡阶段
  - 这一阶段，事件的传播方式和捕获阶段正好相反，会从事件目标一直向上遍历，直至window对象结束，这时对象上绑定的响应函数也会执行。

取消事件传播

- 我们可以使用event对象的两个方法完成：
  - stopPropagation()
  - stopImmediatePropagation()
- 取消默认行为：
  - preventDefault()


尚硅谷
www.atguigu.com