



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣"

第一章 HTML5+CSS3课程介绍

第1节 新一代HTML5+CSS3课程介绍及学后水平分析



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣"

第二章 轻松玩转HTML5

第1节 什么是HTML5？HTML4与HTML5有什么区别？

- 什么是HTML5？

HTML5 是HTML的最新版本，它是W3C与WHATWG合作的结果。HTML5继承了原先HTML的部分特征，

添加了许多新的语法特征，比如语义特性、本地存储特性、设备兼容特性、连接特性、网页多媒体特性等。

- HTML4与HTML5有什么区别？

- HTML4中元素不能把文档结构表示清楚，在HTML5中增加了与结构相关的元素，使得文档结构更加清晰，结构更加语义化。
- HTML4在Web应用程序的功能受到限制，比如：不允许同时上传多个文件，HTML5提供了web应用程序使用的API。
- DOCTYPE声明变化：

HTML4中需要指明是HTML的哪个版本，HTML5不需要，只需要<!DOCTYPE html>即可。

1. HTML4 Strict

严格定义类型：该 DTD 不允许使用展示性的和弃用的元素（比如 font）和框架集（Framesets）。

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

2. HTML4 Transitional

过渡定义类型：该 DTD 允许使用展示性的和弃用的元素（比如 font）。不允许框架集（Framesets）。

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

3. HTML4 Frameset

框架定义类型：该 DTD 等同于 HTML Transitional，但允许框架集内容。

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

- 指定字符编码变化
 - HTML4: < meta http-equiv="content-type" content="text/html; charset=UTF-8">
 - HTML5: < meta charset="UTF-8">
 - 新增的元素和废除的元素
 - 新增结构元素：section、header、footer等
 - 新增其他元素：video、audio、canvas等
 - 新增input元素的类型：email、url、number、range等
 - 废除的元素：basefont、font、center、big等
 - 不再使用frame框架：如frameset、frame等
 - 新增一些元素的属性，如contentEditable、designMode、hidden、tabindex等
- HTML5的优势
 - 实时更新
 - 离线缓存
 - 跨平台
 - 视频和音频的支持

第2节 HTML5常用的语义化和结构元素

结构元素

- **header**标签：通常被放置在页面或者页面中某个区块元素的顶部,用来介绍内容或者当做导航链接栏，
在一个文档中，您可以定义多个< header>元素。
- **nav**标签：通常表示页面的导航
- **main**标签：文档主要内容

- **article**标签：使用在相对比较独立、完整的内容模块，一般使用在博客、论坛帖子、新闻报道、用户评论
- **section**标签：一般用来做内容的分组，比如文章的章节
- **aside**标签：包含的内容不是页面的主要内容、具有独立性，是对页面的补充。

< aside>标签一般使用在页面、文章的侧边栏、广告、友情链接等区域。

- **footer**标签：一般放在页面或页面中某个区块的底部，包含版权信息、联系方式等。

其他元素

- **ruby**标签：定义ruby注释(中文或字符)
 - 将< ruby>标签与< rt>和< rp>标签一起使用。
 - < rp>标签可选，该标签定义的是当浏览器不支持"ruby元素"时显示的内容
- **progress**标签：定义进度条
- **mark**标签：用来标记文本
- **embed**标签：用来嵌入指定的内容，比如插件

第3节 HTML5图形绘画标签canvas

- **canvas**定义

canvas标签用来定义图形，是一个图形容器，常常用来进行绘图操作，要与js搭配使用。

- **创建一个画布容器**

width, height定义画布容器大小

```
<canvas id="myCanvas" width="500" height="400" style="border:1px solid #000">
</canvas>
```

- **canvas**本身没有绘图能力，绘图操作要在js里进行

1. 通过js获取canvas元素对象

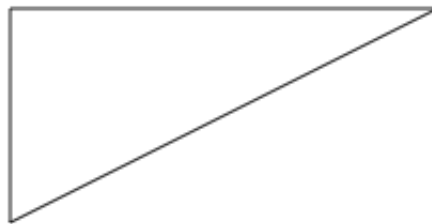
```
var canvas = document.getElementById("myCanvas")
```

2. 创建context对象，context对象拥有多种绘制方法，一般通过调用获取的canvas元素对象下的方法getContext来创建。

```
var cxt = canvas.getContext("2d")
```

3. 实例线条

```
//开始一条路径
ctx.beginPath();
//将画笔移动到指定起始坐标点上
ctx.moveTo(100, 100);
//添加一个点，创建从上一个点到该点的线条
ctx.lineTo(300, 100);
ctx.lineTo(100, 200);
//创建起始坐标点到最终坐标点的路径
ctx.closePath();
//执行绘制
ctx.stroke();
```

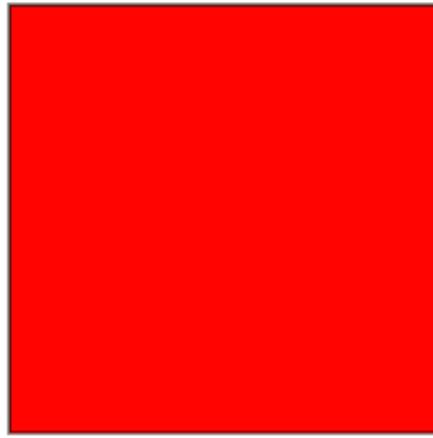


4. 实例矩形

rect(x,y,width,height)

- x, y分别为矩形左上角横纵坐标点
- width, height为矩形的宽高

```
//开始一条路径
ctx.beginPath();
//创建矩形
ctx.rect(100,100,200,200);
//填充颜色
ctx.fillStyle="red";
ctx.fill();
//执行绘制
ctx.stroke();
```

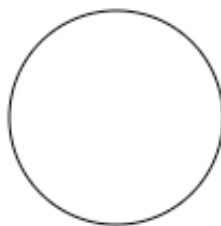


5. 实例圆形

`arc(x,y,r,sAngle,eAngle,counterclockwise)`方法创建弧/曲线

- `x`, `y`分别为圆的中心的横纵坐标
- `r`为圆的半径
- `sAngle` 起始角，以弧度计。(弧的圆形的三点钟位置是0度)
- `eAngle` 结束角，以弧度计。
- `counterclockwise` 可选。规定顺时针还是逆时针绘图。`false`为顺时针，`true`为逆时针

```
//开始一条路径
ctx.beginPath();
//创建圆形
ctx.arc(100, 100, 50, 0, 2 * Math.PI);
//执行绘制
ctx.stroke();
```



第4节 HTML5矢量图形SVG

- 什么是SVG?
 - SVG指可伸缩矢量图形
 - SVG用于定义用于网络的基于矢量的图形
 - SVG是一种基于XML语法的图像格式
 - SVG图像在放大的情况下图形不会失真
- SVG的优点
 - SVG是可伸缩的
 - 文件体积较小
 - 可无限放大且不会失真

svg代码直接插入网页，示例如下：

```
<svg xmlns="http://www.w3.org/2000/svg">
  <circle id="mycircle" cx="400" cy="300" r=50/>
</svg>
```

SVG 代码也可以写在一个独立文件中，然后用 ``、`<object>`、`<embed>`、`<iframe>` 等标签插入网页。

```

```

CSS 也可以使用 SVG 文件

```
.logo {
  background: url(icon.svg);
}
```

第5节 HTML5媒体播放常用标签

- video标签-视频播放

示例如下：

```
<video width="500" height="400" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  您的浏览器不支持 HTML5 video 标签。
</video>
```

- video标签可选属性:

- **autoplay**: 自动播放, 视频就绪后马上播放。
- **controls**: 显示控件
- **width**: 视频播放器的宽度
- **height**: 视频播放器的高度
- **loop**: 循环播放
- **muted**: 视频静音
- **poster**: 视频下载时显示的封面图, 用户点击播放时隐藏
- **preload**: 三个值
 - auto 页面加载, 开始加载音频/视频
 - metadata 页面加载后仅加载音频/视频的元数据
 - none 页面加载后不加载音频/视频
- **src**: 视频的路径

- video支持三种视频格式: mp4、webm、ogg

- video元素支持多个< source >元素, < source >元素可以链接不同的视频文件, 浏览器会使用第一个可识别的格式

- **audio标签-音频播放**

示例如下:

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  您的浏览器不支持 audio 元素。
</audio>
```

- 支持三种音频格式: mp3、ogg、wav

- audio标签可选属性: controls、loop、muted、preload、src(这些参数同上)

第6节 HTML5新增input类型

- 新增的输入类型
 - color类型：选色器
 - date类型：日期选择器
 - datetime-local类型：选择一个日期和时间
 - email类型输入格式为email模式，自动验证输入值是否为email
 - month类型选择一个月份
 - number类型规定输入值为数字
 - max规定允许的最大值
 - min规定允许的最小值
 - step规定每一段数字间隔
 - range类型包含一定范围内数字值的输入域
 - max规定允许的最大值
 - min规定语序的最小值
 - step规定每一段数字间隔
 - search类型用于搜索域
 - time类型选择一个时间
 - url类型规定输入值为url格式
 - week选择周和年

第7节 HTML5常用的表单属性

- autocomplete属性：该属性规定form或input域是否有自动完成功能，当用户在输入域输入时提示之前输入的内容选项。
 - 注意：该属性适用于**form**标签，以及以下**input**标签:text, search, url, tel, email, password, datepickers, range以及color。
- autofocus属性：该属性规定在页面加载时，该input框自动获得焦点
- width和height属性：设置宽高
 - 注意：该属性只适用于image类型的input标签。
- multiple属性：规定**input**元素可选择多个值
 - 注意：该属性适用于email和file类型，提交多个邮箱时用逗号，隔开。
- pattern属性：使用一个正则表达式用于验证input框的值
 - 注意：该属性适用于以下类型的**input**标签：text, search, url, tel, email和password。

- placeholder属性：该属性给输入框提供输入提示
 - 注意：该属性适用于以下类型的标签：text, search, url, tel, email以及password。
- required属性：表示该输入框必须填写，不能为空
 - 注意：该属性适用于以下类型的标签：text, search, url, tel, email, password, datepickers, number, checkbox, radio以及file。

第8节 HTML5必备知识点Web本地储存

- 什么是HTML5web存储？
 - HTML5web存储可以在本地储存用户的浏览数据。web存储相对cookie更加的安全与快速，它的数据不会保存在服务器上。他也可以存储大量的数据，而不影响网站的性能。
- 客户端储存数据之localStorage
 - 用于长久保存网站的数据，保存的数据没有过期时间，可手动删除
 - 常用API如下：
 - 保存数据：localStorage.setItem(key,value);
 - 读取数据：localStorage.getItem(key);
 - 删除单个数据：localStorage.removeItem(key);
 - 删除所有数据：localStorage.clear();
 - 得到某个索引的key：localStorage.key(index);
 - 提示：键/值对通常以字符串储存
- 客户端储存数据之sessionStorage
 - sessionStorage储存的数据在用户关闭浏览器窗口后，数据会被删除
 - API如localStorage相同
- cookie, localStorage以及sessionStorage的区别
 - 共同点：都是保存在浏览器端，且是同源共享的。
 - 区别：
 - cookie数据会在浏览器和服务器之间进行传递，每次http请求都会携带cookie，而sessionStorage和localStorage仅在客户端保存，不参与服务器通信。
 - 存储大小限制也不同，cookie数据不能超过4k，sessionStorage和localStorage的大小一般是5M。
 - 数据的周期时间不同，cookie数据在设置的过期时间内一直有效，sessionStorage数据会在当前浏览器窗口时清除，localStorage数据则始终存在，即使关闭浏览器或窗口，直到手动清除数据。
 - cookie相比其他两个安全性不高，容易被拦截获取



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣"

第三章 CSS3新特性及核心知识

第1节 CSS3新增选择器

- `el1~el2`匹配element1元素之后的每一个element2元素，必须具有相同的父元素
- `[attribute^=value]`匹配元素属性值以指定value值开头的元素
- `[attribute$=value]`匹配元素属性值以指定value值结尾的元素
- `[attribute*=value]`匹配元素属性值包含指定value值的元素
- `el:first-of-type`匹配的el元素是其父元素的第一个子元素
- `el:last-of-type`匹配的el元素是其父元素的最后一个子元素
- `el:only-of-type`匹配的el元素是其父元素唯一的子el元素
- `el:only-child`匹配属于父元素中唯一子元素的el元素
- `el:nth-child(n)`匹配父元素中的第n个子元素，子元素不是el元素则不生效
 - n可以是数字，关键字或者公式
- `el:nth-last-child(n)`匹配父元素中的倒数第n个子元素，子元素不是el元素则不生效
 - n可以是数字，关键字或者公式
- `el:nth-last-of-type(n)`匹配父元素中同类型的倒数第n个子元素
 - n可以是数字，关键字或者公式
- `el:last-child`匹配父元素中最后一个子元素
- `:root`匹配文档的根元素，该选择器等同于html(在html文档中根元素就是html)
- `el:empty`匹配每个没有任何子级的元素(包过文本节点)
- `:target`匹配当前活动的target元素的样式
- `el:enabled`匹配每个启用的元素(主要用于表单元素)
- `el:disabled`匹配每个禁用的元素(主要用于表单元素)
- `el:checked`匹配每个选中的输入元素(仅用于单选按钮或复选框)
- `:not(selector)`匹配除了指定selector元素的其他元素
- `::selection`匹配被用户选中或处于高亮状态的元素
 - 该选择器只可以应用于少数的css属性：color，background，cursor和outline
- `el:out-of-range`匹配值在指定区间外的input元素

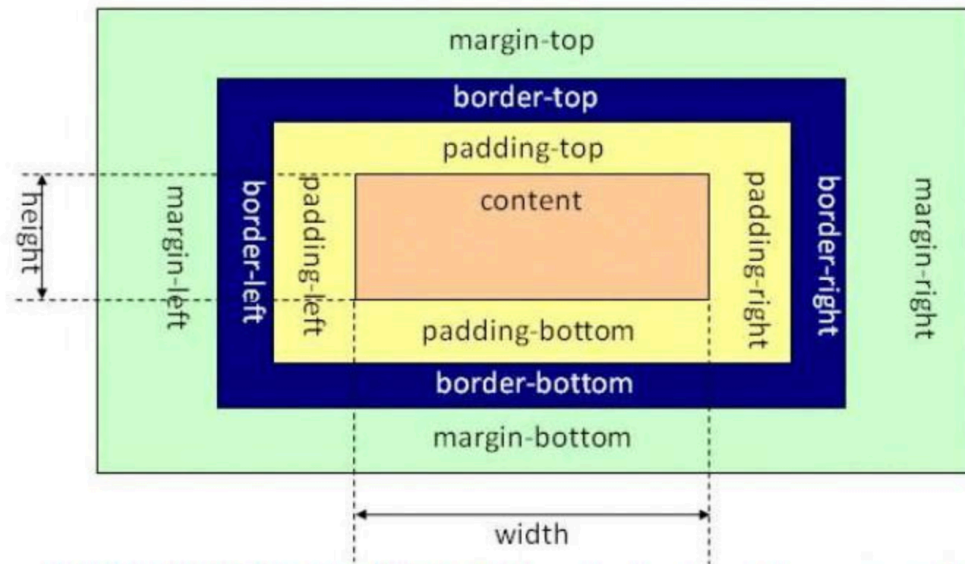
- **注意：** 该选择器只作用于能指定区间之外值得元素，例如input元素的min和max属性
- el:in-range匹配值在指定区间内的input元素
- el:read-write匹配可读及可写的元素
- el:read-only匹配设置了readonly属性的元素
- el:optional匹配可选的输入元素
 - optional选择器只适用于表单元素：input、select和textarea
- el:required匹配设置了required属性的元素
 - required选择器只适用于表单元素：input、select和textarea
- el:valid匹配输入值为合法的元素
 - **注意：** :valid 选择器只作用于能指定区间值的元素，例如 input 元素中的 min 和 max 属性，及正确的 email 字段, 合法的数字字段等。
- el:invalid匹配输入值为非法的元素

第2节 CSS3新增的用户界面属性

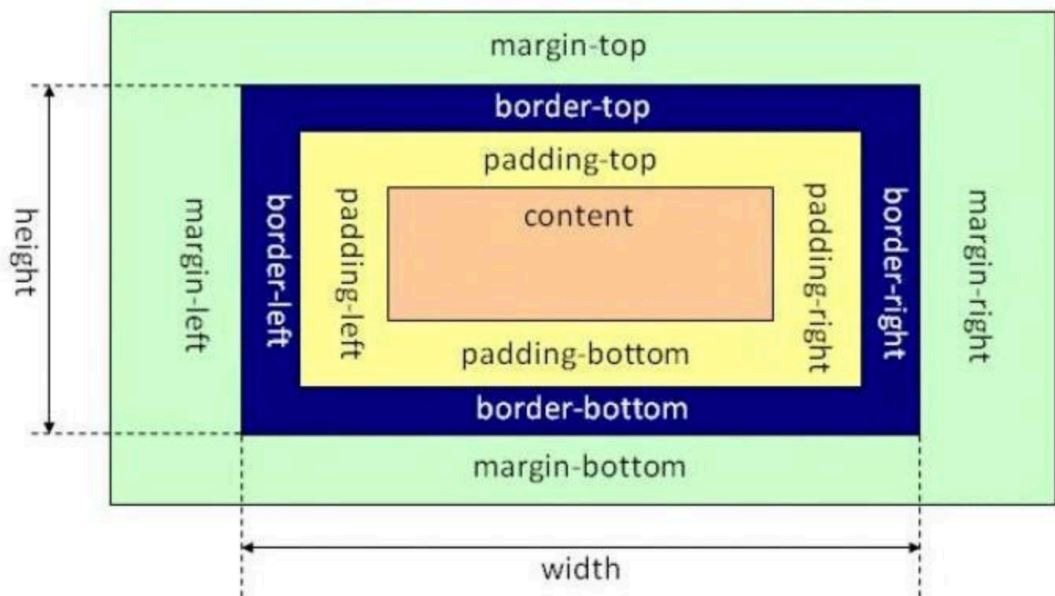
- resize属性规定是否可以调整元素尺寸
 - none **默认值** 无法调整元素的尺寸
 - both 可调整元素的高度和宽度
 - horizontal 可调整元素的宽度
 - vertical 可调整元素的高度
- box-sizing属性改变盒子模型的组成方式

以下是该属性的三个值：

- content-box **默认值** padding和border不计入width和height内



- border-box padding和border计入width和height内



- inherit 继承父元素的box-sizing属性的值
- outline-offset属性设置轮廓框架在border边缘外的偏移

outline与border有两点不同：

- outline不占用空间
- outline可能是非矩形

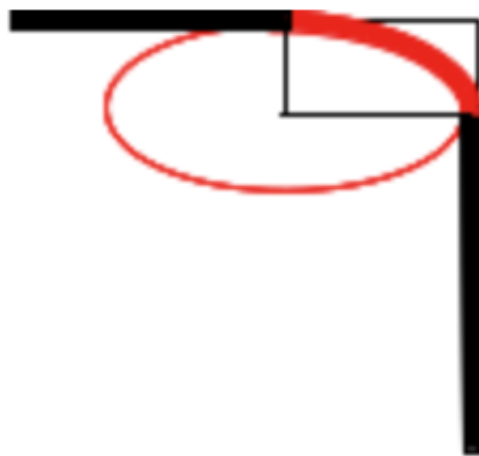
第3节 CSS3常用的样式属性边框与盒阴影

- **CSS3圆角属性border-radius：** 被用于创建圆角

- border-radius属性每个半径四个值的顺序分别是左上角，右上角，右下角，左下角
- 可传的值如下
 - length：定义圆形半径或椭圆的半长轴，半短轴。负值无效。
 - percentage：使用百分数定义圆形半径或椭圆的半长轴，半短轴。水平半轴相对于盒模型的宽度；垂直半轴相对于盒模型的高度。负值无效。



- /斜杆分隔写法，斜杆左边的值为水平半轴，右边值垂直半轴



```
border-radius: 100px/50px;
```

```
/* 等价于: */
```

```
border-top-left-radius: 100px 50px;
```

```
border-top-right-radius: 100px 50px;
```

```
border-bottom-right-radius: 100px 50px;
```

```
border-bottom-left-radius: 100px 50px;
```

- **CSS3盒阴影属性box-shadow：** 被用来添加阴影

```
box-shadow: h-shadow v-shadow blur spread color inset;
```

- *h-shadow*: 必需的。水平阴影的位置。允许负值
- *v-shadow*: 必需的。垂直阴影的位置。允许负值
- *blur*: 可选。模糊距离
- *spread*: 可选。阴影的大小
- *color*: 可选。阴影的颜色。
- *inset*: 可选。阴影向内扩散

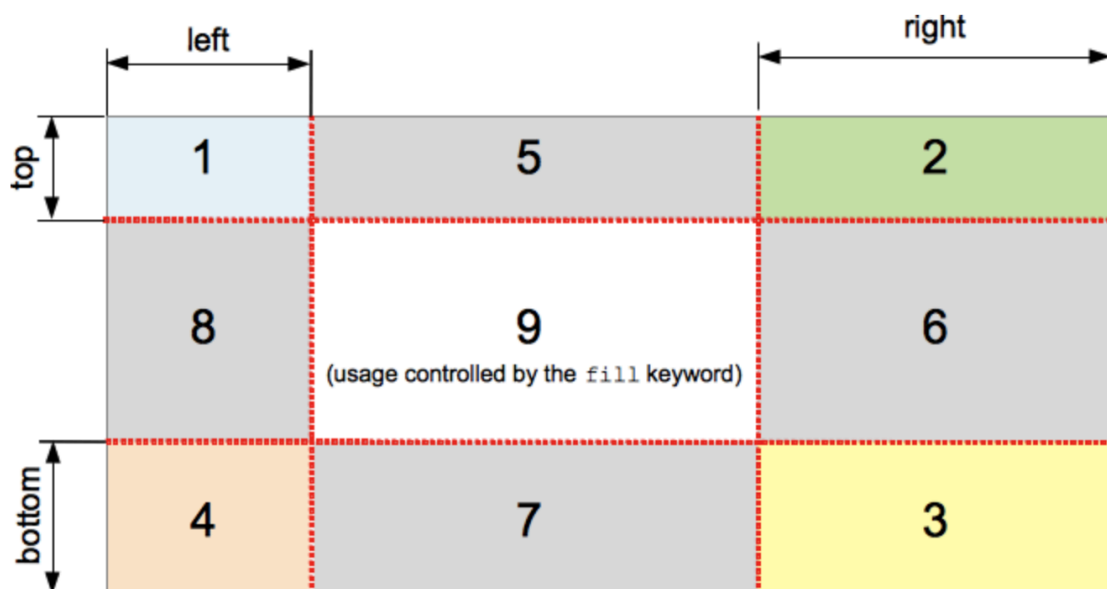
- **CSS3图片边框属性border-image**: 指定一个图片作为边框

```
border-image: border-image-source border-image-slice / border-image-width  
border-image-outset border-image-repeat
```

- *border-image-source*: 边框图片的资源路径
- *border-image-slice*: 切割图片,[< number> | < percentage>]
- *border-image-width*: 图像边界的宽度 [< length> | < percentage> | < number> | auto]

length 带 px, em, in ... 单位的尺寸值
percentage 百分比
number 不带单位的数字; 它表示 border-width 的倍数
auto 使用 auto, border-image-width 将会使用 border-image-slice 的值

- *border-image-outset*: 定义边框图像向外偏移的宽度[< length> | < number>]
- *border-image-repeat*: 用于设置图像边界是否应重复 (repeat)、拉伸 (stretch) 或铺满 (round)



第4节 CSS3背景新特性及精灵图片的使用

- background-image: 添加背景图片
 - 可添加多个背景，每张图片用，逗号隔开，前面的图片会绘制在后面的图片上
- background-size: 设置背景图片的大小
 - 可传的值如下

<length> 值，指定背景图片大小，不能为负值。
<percentage> 值，指定背景图片相对背景区 (background positioning area) 的百分比。
<auto> 值，以背景图片的比例缩放背景图片。
<cover> 值，缩放背景图片以完全覆盖背景区。
<contain> 值，按原有宽高比例缩放背景图片以完全装入背景区。

- background-origin: 指定背景图片的位置
 - 可传的值如下
 - border-box: 背景图片的摆放以border区域为参考
 - padding-box: **默认值** 背景图片的摆放以padding区域为参考
 - content-box: 背景图片的摆放以content区域为参考
 - 注意：当使用 `background-attachment` 为fixed时，该属性将被忽略不起作用。
- background-clip: 指定从哪个位置开始绘制
 - 可传的值如下

border-box: 默认值，背景延伸至边框外沿（但是在边框下层）。
padding-box: 背景延伸至内边距（padding）外沿。不会绘制到边框处。
content-box: 背景被裁剪至内容区（content box）外沿。

第5节 详细讲解CSS3渐变背景

- 线性渐变 (linear-gradient)

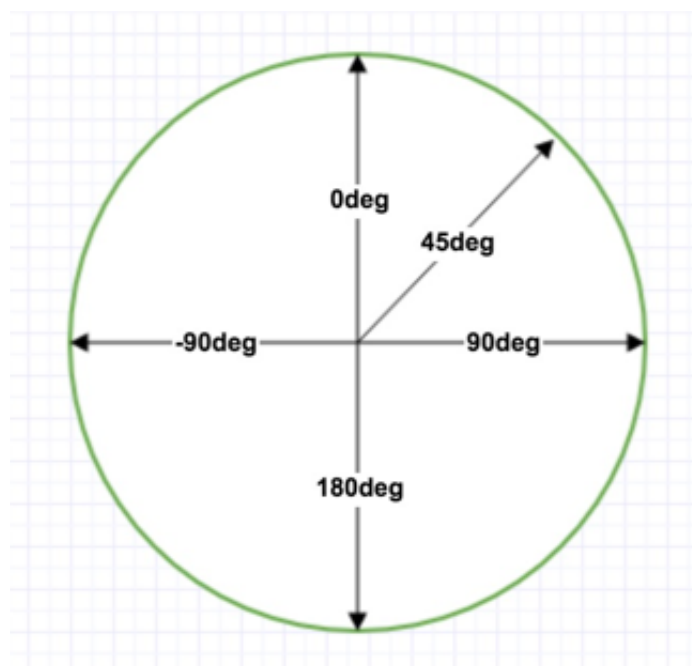
1. 标准语法:

```
background: linear-gradient(direction, color-stop1, color-stop2, ...);
```

- direction:渐变的方向，默认情况是从上到下
- 标准语法方向是渐变的末尾位置，其他浏览器前缀语法是渐变的起始位置
- 注意：标准语法要放在其他浏览器语法后面

2. 角度渐变：标准语法0度指向北，其他浏览器前缀语法指向东

换算公式：90-标准语法角度=其他浏览器前缀语法角度



3. 重复的线性渐变：repeating-linear-gradient()

- 径向渐变 (radial-gradient)

```
background: radial-gradient(shape size at position, start-color, ..., last-color);
```


值	描述
<i>shape</i>	<p>确定圆的类型:</p> <ul style="list-style-type: none"> ● ellipse (默认): 指定椭圆形的径向渐变。 ● circle : 指定圆形的径向渐变
<i>size</i>	<p>定义渐变的大小, 可能值:</p> <ul style="list-style-type: none"> ● farthest-corner (默认) : 指定径向渐变的半径长度为从圆心到离圆心最远的角 ● closest-side : 指定径向渐变的半径长度为从圆心到离圆心最近的边 ● closest-corner : 指定径向渐变的半径长度为从圆心到离圆心最近的角 ● farthest-side : 指定径向渐变的半径长度为从圆心到离圆心最远的边
<i>position</i>	<p>定义渐变的位置。可能值:</p> <ul style="list-style-type: none"> ● center (默认) : 设置中间为径向渐变圆心的纵坐标值。 ● top: 设置顶部为径向渐变圆心的纵坐标值。 ● bottom: 设置底部为径向渐变圆心的纵坐标值。
<i>start-color, ..., last-color</i>	用于指定渐变的起止颜色。

- 重复径向渐变: **repeating-radial-gradient()**

第6节 CSS3文本效果及特殊字体引入

- **text-shadow**: 给文本添加阴影效果

```
text-shadow: h-shadow v-shadow blur color;
```

对应值描述如下:

值	描述
<i>h-shadow</i>	必需。水平阴影的位置。允许负值。
<i>v-shadow</i>	必需。垂直阴影的位置。允许负值。
<i>blur</i>	可选。模糊的距离。
<i>color</i>	可选。阴影的颜色。

- text-overflow: 指定当文本溢出包含它的元素的形式

```
text-overflow: clip|ellipsis|string;
```

对应值描述如下：

值	描述
clip	修剪文本。
ellipsis	显示省略符号来代表被修剪的文本。
<i>string</i>	使用给定的字符串来代表被修剪的文本。(只在firefox浏览器有效)

- word-wrap: 属性允许内容在过长情况下可以自动换行。

```
word-wrap: normal|break-word;
```

对应值描述如下：

值	描述
normal	只在允许的断字点换行（浏览器保持默认处理）。
break-word	在长单词或 URL 地址内部进行换行。

- word-break: 指定内容的断行规则。

```
word-break: normal|break-all|keep-all;
```

对应值如下：

值	描述
normal	使用浏览器默认的换行规则。
break-all	允许在单词内换行。
keep-all	只能在半角空格或连字符处换行。

- @font-face 规则：定义自己的文本字体

首先定义字体的名称（比如 myFirstFont），然后指向该字体文件。

- 提示：URL请使用小写字母的字体，大写字母在IE中会产生意外的结果

```
<style>
  @font-face {
    font-family: myFont;
    src: url('Dexsar.otf'),
         url('Dexsar.ttf');
  }

  .textShadow {
    font-family: myFont;
  }
</style>
```

第7节 讲解CSS3的多列布局

- column-count 指定了需要分割的列数。

```
.colCount {
  -webkit-column-count: 3; /* Chrome, Safari, Opera */
  -moz-column-count: 3; /* Firefox */
  column-count: 3;
}
```

- column-gap 指定了列与列间的间隙

```
{
  -moz-column-gap:40px; /* Firefox */
  -webkit-column-gap:40px; /* Safari and Chrome */
  column-gap:40px;
}
```

- `column-rule-style` 指定了列与列间的边框样式

```
{
  -webkit-column-rule-style: solid; /* Chrome, Safari, Opera */
  -moz-column-rule-style: solid; /* Firefox */
  column-rule-style: solid;
}
```

- `column-rule-width` 指定了两列的边框厚度

```
{
  -webkit-column-rule-width: 1px; /* Chrome, Safari, Opera */
  -moz-column-rule-width: 1px; /* Firefox */
  column-rule-width: 1px;
}
```

- `column-rule-color` 指定了两列的边框颜色

```
{
  -webkit-column-rule-color: #ff0000; /* Chrome, Safari, Opera */
  -moz-column-rule-color: #ff0000; /* Firefox */
  column-rule-color: #ff0000;
}
```

- `column-rule` 是 `column-rule-*` 所有属性的简写

```
{
  -webkit-column-rule: 1px solid #ff0000; /* Chrome, Safari, Opera */
  -moz-column-rule: 1px solid #ff0000; /* Firefox */
  column-rule: 1px solid #ff0000;
}
```

- `column-width` 指定了列的宽度

```
{
  column-width:100px;
  -moz-column-width:100px; /* Firefox */
  -webkit-column-width:100px; /* Safari and Chrome */
}
```

第8节 设备兼容必备知识多媒体查询

- 使用多媒体查询可以在指定的设备上使用对应的样式替代原有的样式。

```
//大于480px
@media screen and (min-width: 480px) {
    body {
        background-color: red;
    }
}
//大于960px/小于1200px
@media screen and (min-width:960px) and (max-width:1200px){
    body{
        background:yellow;
    }
}
```



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣"

第四章 体验CSS3特效的乐趣

第1节 深度讲解2D转换动画效果

transform属性下的函数

- translate(x,y)平移

x为x轴上的平移，y为y轴上的平移

```
{
  transform: translate(50px,100px);
  -ms-transform: translate(50px,100px); /* IE 9 */
  -webkit-transform: translate(50px,100px); /* Safari and Chrome */
}
```

另外两种方法：

1. translateX(x)元素仅在水平方向位移
2. translateY(y)元素仅在垂直方向位移

- rotate(angle)旋转

正度数表示顺时针旋转，负值则相反

```
{
  transform: rotate(30deg);
  -ms-transform: rotate(30deg); /* IE 9 */
  -webkit-transform: rotate(30deg); /* Safari and Chrome */
}
```

- scale()缩放

scale()方法可以将元素根据中心原点进行缩放

```
{
  -ms-transform:scale(2,3); /* IE 9 */
  -webkit-transform: scale(2,3); /* Safari */
  transform: scale(2,3); /* 标准语法 */
}
```

另外两种方法：

1. scaleX(x)元素仅在水平方向上缩放(x轴缩放)
2. scaleY(y)元素仅在垂直方向上缩放(y轴缩放)

- skew()倾斜

语法：

```
transform:skew(<angle> [,<angle>]);
```

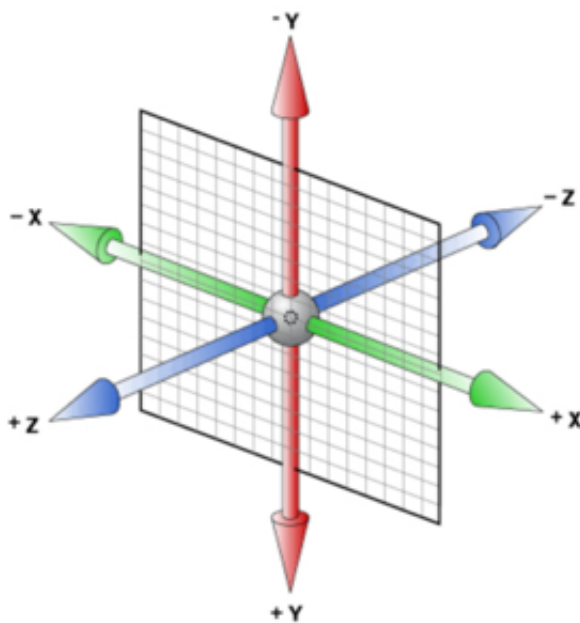
- 包含两个参数值，分别表示X轴和Y轴倾斜的角度，如果第二个参数为空，则默认为0，参数为负表示向相反方向倾斜。

```
{
  transform: skew(30deg,20deg);
  -ms-transform: skew(30deg,20deg); /* IE 9 */
  -webkit-transform: skew(30deg,20deg); /* Safari and Chrome */
}
```

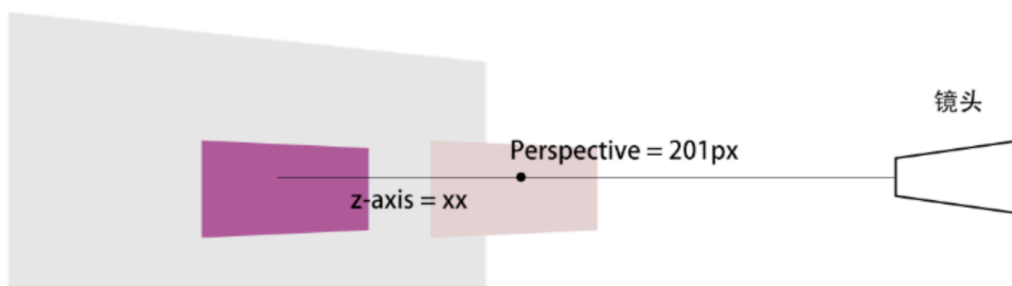
- skewX(< angle>);表示只在X轴(水平方向)倾斜。
- skewY(< angle>);表示只在Y轴(垂直方向)倾斜。

第2节 深度讲解3D转换动画效果

- translate3d(x,y,z)定义3d转换



- translateZ(z)在z轴上的平移，z轴为面向屏幕的这个方向
- perspective()方法设置3d透视视图
- 在父元素上加上transform-style: preserve-3d可修改元素层级



- rotateX定义沿着X轴的3d旋转

```
{
  transform: rotateX(180deg);
  -webkit-transform: rotateX(180deg); /* Safari 与 Chrome */
}
```

- rotateY定义沿着Y轴的3d旋转

```
{
  transform: rotateY(180deg);
  -webkit-transform: rotateY(180deg); /* Safari 与 Chrome */
}
```

第3节 讲解CSS3过渡实用技巧

CSS3 过渡是元素从一种样式逐渐改变为另一种的效果。

语法：

```
transition: property duration timing-function delay;
```

值	描述
<i>transition-property</i>	指定CSS属性的name, transition效果
<i>transition-duration</i>	transition效果需要指定多少秒或毫秒才能完成
<i>transition-timing-function</i>	指定transition效果的转速曲线
<i>transition-delay</i>	定义transition效果开始的时候

案例


```
.test
{
  width:100px;
  height:100px;
  background:red;
  transition:width 2s;
  -webkit-transition:width 2s; /* Safari */
}

test:hover
{
  width:200px;
}
```

- **transition-property**属性

值	描述
none	没有属性会获得过渡效果。
all	所有属性都将获得过渡效果。
<i>property</i>	定义应用过渡效果的 CSS 属性名称列表，列表以逗号分隔。

- **transition-timing-function**属性

值	描述
linear	规定以相同速度开始至结束的过渡效果。
ease	默认值。规定慢速开始，然后变快，然后慢速结束的过渡效果。
ease-in	规定以慢速开始的过渡效果。
ease-out	规定以慢速结束的过渡效果。
ease-in-out	规定以慢速开始和结束的过渡效果。

添加多个样式变换效果，属性之间使用逗号分隔

```
.test {
  width: 100px;
  height: 100px;
  background: red;
```

```
-webkit-transition: width 2s, height 2s, -webkit-transform 2s; /* For
Safari 3.1 to 6.0 */
transition: width 2s, height 2s, transform 2s;
}

test:hover {
width: 200px;
height: 200px;
-webkit-transform: rotate(360deg); /* Chrome, Safari, Opera */
transform: rotate(360deg);
}
```

第4节 解锁CSS3动画特效技能

- CSS3 @keyframes规则

@keyframes规则内指定一个CSS样式和动画将逐步从目前的样式更改为新的样式。

语法：

```
animation: name duration timing-function delay iteration-count direction
fill-mode play-state;
```

单个属性设置如下：

值	说明
<i>animation-name</i>	指定动画的名称
<i>animation-duration</i>	动画需要多少秒或毫秒完成
<i>animation-timing-function</i>	设置动画将如何完成一个周期
<i>animation-delay</i>	设置动画在启动前的延迟间隔。
<i>animation-iteration-count</i>	定义动画的播放次数。
<i>animation-direction</i>	指定是否应该轮流反向播放动画。
animation-fill-mode	规定当动画不播放时（当动画完成时，或当动画有一个延迟未开始播放时），要应用到元素的样式。
<i>animation-play-state</i>	指定动画是否正在运行或已暂停。

案例：

```
div
{
  width:100px;
  height:100px;
  background:red;
  animation:myAnimation 5s;
  -webkit-animation:myAnimation 5s; /* Safari and Chrome */
}

@keyframes myAnimation
{
  from {background:red;}
  to {background:yellow;}
}

@-webkit-keyframes myAnimation /* Safari and Chrome */
{
  from {background:red;}
  to {background:yellow;}
}
```

- 过渡的速度曲线如下： (**animation-timing-function**)

值	描述
linear	动画从头到尾的速度是相同的。
ease	默认。动画以低速开始，然后加快，在结束前变慢。
ease-in	动画以低速开始。
ease-out	动画以低速结束。
ease-in-out	动画以低速开始和结束。

- **animation-delay** 属性允许负值，例如-2s 使动画马上开始，但跳过 2 秒进入动画。
- **animation-iteration-count** 传数字（表示播放多少次）或者infinite(无限播放)
- **animation-direction** 属性定义是否循环交替反向播放动画。

注意：如果动画被设置为只播放一次，该属性将不起作用。

值	描述
normal	默认值。动画按正常播放。
reverse	动画反向播放。
alternate	动画在奇数次（1、3、5...）正向播放，在偶数次（2、4、6...）反向播放。
alternate-reverse	动画在奇数次（1、3、5...）反向播放，在偶数次（2、4、6...）正向播放。

- **animation-fill-mode**属性

值	描述
none	默认值。动画在动画执行之前和之后不会应用任何样式到目标元素。
forwards	在动画结束后（由 animation-iteration-count 决定），动画将应用该属性值。
backwards	动画将应用在 animation-delay 定义期间启动动画的第一次迭代的关键帧中定义的属性值。这些都是 from 关键帧中的值（当 animation-direction 为 "normal" 或 "alternate" 时）或 to 关键帧中的值（当 animation-direction 为 "reverse" 或 "alternate-reverse" 时）。
both	动画遵循 forwards 和 backwards 的规则。也就是说，动画会在两个方向上扩展动画属性。

- **animation-play-state**属性

值	描述
paused	指定暂停动画
running	指定正在运行的动画



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣"

第五章 css3 flex弹性布局-容器属性

内容参考下面两篇文章：<http://www.ruanyifeng.com/blog/2015/07/flex-grammar.html>和 [A Visual Guide to CSS3 Flexbox Properties](#)。

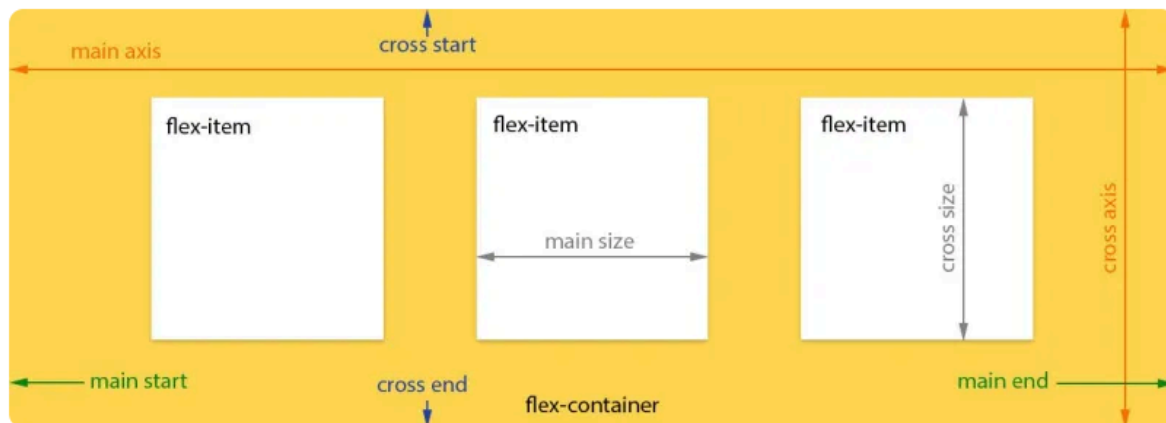
第1节 了解前端核心知识点flex弹性布局

- flex布局是什么？

Flex意思是弹性布局，是css3中的新布局模块，用来为盒模型提供最大的灵活性。可改进容器中的项目对齐，方向和顺序，即使他们具有动态甚至是未知大小。Flex容器能够调节子节点的宽度或高度，以便适应不同的屏幕尺寸。

- 基本概念

flex布局可以将一个元素看成一个容器，父元素下的子元素会自动成为子容器，这样就构成一个flex项目。



容器存在两根轴：水平的主轴(main axis)和垂直的交叉轴(cross axis)。主轴开始的位置叫做main start，结束位置叫做main end；交叉轴的开始位置叫做cross start，结束位置叫做cross end。单个项目占据的主轴空间叫做main size，占据的交叉轴空间叫做cross size。

- 用法

使用flex布局，需在父元素上设置display属性

```
.flexTest {  
  display: -webkit-flex; /* Safari */  
  display: flex;  
}
```

或者将其显示为行内元素

```
.flexTest {  
  display: -webkit-inline-flex; /* Safari */  
  display: inline-flex;  
}
```

注意，设置flex布局后，子元素的float、clear和vertical-align属性将失效。

第2节 讲解flex布局容器六大属性之flex-direction

- flex-direction属性决定主轴的方向

值	描述
row	默认值。主轴为水平方向，起点在左边。
row-reverse	主轴为水平方向，起点在右边。
column	主轴为垂直方向，起点在上面。
column-reverse	主轴为垂直方向，起点在下面。

第3节 讲解flex布局容器六大属性之flex-wrap和flex-flow

- flex-wrap属性设置子项目的换行方式

值	描述
nowrap	默认值。不换行
wrap	换行，从上往下排列，第一行在上方。
wrap-reverse	换行，从下往上排列，第一行在下方。

- flex-flow属性是flex-direction属性和flex-wrap属性的简写方式，默认值是row nowrap。

语法：

```
flex-flow: <flex-direction> || <flex-wrap>;
```

第4节 讲解flex布局容器六大属性之justify-content

- justify-content属性设置子项目在主轴上的对齐方式。

值	描述
flex-start	默认值。左对齐
flex-end	右对齐
center	居中
space-between	两端对齐，子项目之间间隔一样
space-around	每个子项目两侧的间隔一样。

第5节 讲解flex布局容器六大属性之align-items

- align-items属性规定子项目在交叉轴上的对齐方式

值	描述
flex-start	交叉轴的起点对齐
flex-end	交叉轴的终点对齐
center	交叉轴的中点对齐
baseline	以子项目的第一行文字为基线对齐
stretch	默认值。如果项目未设置固定高度，将占满整个容器的高度。

第6节 讲解flex布局容器六大属性之align-content

- align-content多行情况下的对齐方式，类似justify-content的对齐方式

值	描述
flex-start	与交叉轴的起点对齐
flex-end	与交叉轴的终点对齐
center	与交叉轴的中点对齐
space-between	与交叉轴的两端对齐，每行之间间隔一样
space-around	每行上下的间隔一样
stretch	默认值。未设置高度将占满整个空间，如设置的高度不够则每行下方填充空白填满容器



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣"

第六章 css3 flex弹性布局-项目属性

内容参考下面两篇文章：<http://www.ruanyifeng.com/blog/2015/07/flex-grammar.html>和 [A Visual Guide to CSS3 Flexbox Properties](#)。

第1节 讲解flex布局项目六大属性之order和flex-grow

- order属性规定子容器（项目）的排列顺序。数值越小越靠前，默认为0,支持负数

```
.item{
  order:<integer>
}
```

- flex-grow属性规定子容器的放大比例，默认为0，剩余空间不足则不会放大。

```
.item {
  flex-grow:<number>
}
```

如果子容器的flex-grow属性都为1，则等分剩余空间。

第2节 讲解flex布局项目六大属性之flex-shrink

- flex-shrink属性规定了子容器的缩小比例，默认为1，如果空间不足，该子容器将缩小。

```
.item{
  flex-shrink:<number>
}
```

如果子容器的flex-shrink属性都为1，当空间不足时，都将等比例缩小。

第3节 讲解flex布局项目六大属性之flex-basis和flex

- flex-basis属性可修改子容器占据的主轴空间的大小。默认值为auto，即子容器的本来大小

```
.item{
  flex-basis:<length> | auto;
}
```

- flex属性是flex-grow, flex-shrink和flex-basis的简写，默认值是0 1 auto，后两个属性可选。

```
.item {
  flex: none | [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'> ]
}
```

该属性设置auto等同于（1 1 auto），none等同于（0 0 auto）

提示：建议优先使用这个属性，而不是单独写三个分离的属性，因为浏览器会推算相关值。

第4节 讲解flex布局项目六大属性之align-self

- align-self属性允许单个子容器与其他子容器有不一样的对齐方式，可覆盖align-items属性。默认值为auto，表示继承父元素的align-items属性。

```
.item {
  align-self: auto | flex-start | flex-end | center | baseline | stretch;
}
```



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣"

第七章 综合实战案例及课程总结

第1节 HTML5+CSS3项目实战(一)

第2节 HTML5+CSS3项目实战(二)

第3节 HTML5+CSS3项目实战(三)

第4节 HTML5+CSS3项目实战(四)

第5节 课程总结及常见面试题

- 常见HTML5和CSS3面试题
 - CSS实现垂直水平居中

1. 水平居中

- 1.内联元素 `text-align: center`
- 2.块级元素 `margin-left`和`margin-right`设置`auto` `margin: 0 auto`
- 3.利用flex布局的`justify-content: center`在主轴上水平居中

2. 垂直居中

- 将`height`和`line-height`设为相同值使元素垂直居中
- 利用`vertical-align: middle`实现子元素的垂直居中
- 利用flex布局的`flex-direction: column`;和`justify-content: center`;实现垂直居中
- 绝对定位`position: absolute`

已知元素高度：

```
.parent {
  position: relative;
}
.child {
  position: absolute;
  top: 50%;
  height: 100px;
  margin-top: -50px;
}
```

未知元素高度:

```
.parent {
  position: relative;
}
.child {
  position: absolute;
  top: 50%;
  transform: translateY(-50%);
}
```

- 解释css sprites, 如何使用

Css精灵把一堆小的图片合并成一张图片, 减轻服务器对图片的请求数量, 应用background-position对图片进行截取

- CSS3有哪些新特性?

CSS3实现圆角 (border-radius) , 阴影 (box-shadow) , 对文字加特效 (text-shadow、) , 线性渐变 (gradient) , transform (平移, 旋转, 缩放, 倾斜) , 增加了更多的CSS选择器, 多背景, 媒体查询, border-image, flex布局等

- 左右布局, 一侧定宽, 另一侧自适应撑满

采用flex布局, 一侧设置固定宽度, 另一侧设置flex: 1让元素分配剩下的空间