

1. text-shadow (针对文字)

```
text-shadow: color x y blur, [color x y blur]...
```

- `color` : 阴影颜色, `blur` : 模糊半径
- 阴影默认颜色和文字颜色相同
- 多阴影操作作用逗号隔开
- 阴影 `x` 轴设置正值, 就是下方阴影, 设置负值就是上方阴影, `y` 同理

2. box-shadow (针对容器)

```
box-shadow: X Y blur spread color inset;  
box-shadow: 10px 10px 2px 5px red inset;
```

- `spread` : 阴影范围扩大
- `inset` : 外阴影(默认), `outset` 外阴影
- 盒子阴影的默认颜色是黑色。

3. mask 遮罩

- 遮罩用的图片: 透明区域遮罩在图片上, 图片上的内容也会边透明

mask
url
repeat
X
Y
w
h
多遮罩

```
-webkit-mask: url('./image/自制/mask4.png') no-repeat right center/200px 200px;
```

- `mask` 目前还没有标准化, 要添加浏览器前缀。
- 默认是x、y都平铺
- `mask` 可以设置 位置、大小、`-webkit-mask-size`; 也可以设置多遮罩, 用两个 `URL()`, 用逗号隔开

4. box-reflect 倒影

```
box-reflect: <direction> <offset> <mask-box-image>;
```

- `direction` : `above` 指定倒影在对象的上边, `below` 下边, `left` 左边, `right` 右边
- `offset` : 倒影与对象之间的间隔 (px %)
- `mask-box-image` :
 - `url` 使用绝对或相对地址指定遮罩图像

- `linear-gradient` 使用线性渐变创建遮罩图像
- `radial-gradient` 使用径向(放射性)渐变创建遮罩图像
- `box-reflect` 目前还没有标准化，要添加浏览器前缀。
- 扩展：`scaleX(-1)` x 轴水平翻转，`scaleY(-1)` y 轴垂直翻转，`scale(-1)` x, y 都翻转。

5. blur模糊

```
filter : blur(2px);
```

6. calc计算

- 四则运算

```
width:calc(100% - 100px); /* 符号两边要用空格。 */
```

7. column 分栏布局

属性	描述
<code>column-count</code>	分栏的个数
<code>column-width</code>	分栏的宽度
<code>column-gap</code>	分栏的间距
<code>column-rule</code>	分栏的边线
<code>column-span</code>	合并分栏

- 注：`column-count` 个数和 `column-width` 宽度不能一起设置

8. 伪元素概念

- 伪元素本质上是创建了一个有内容的虚拟容器。这个容器不包含任何 `DOM` 元素，但是可以包含内容，另外开发者还可以为伪元素定制样式。
- 伪类(`:`) 和 伪元素(`::`) 的区别
 - 数量：
 - 伪类可以同时存在多个拼接：`input:first-child:focus{}`
 - 伪元素只能存在一个：`input::after{}`
 - 位置：
 - 伪元素只能在最后面：`input:checked::after{}`
- 常见的伪元素选择器

伪元素	描述
<code>::first-letter</code>	选择元素文本的第一个字

伪元素	描述
<code>::first-line</code>	选择元素文本的第一行
<code>::before</code>	在元素内容的最前面添加新内容
<code>::after</code>	在元素内容的最后面添加新内容
<code>::selection</code>	匹配用户被用户选中或者处于高亮状态的部分
<code>::placeholder</code>	匹配占位符的文本，只有元素设置了 <code>placeholder</code> 属性时，该伪元素才能生效

9. css Hack

- 用服务器环境测试
- Hack 分类
 1. CSS 属性前缀法
 - 属性前缀法是在 CSS 样式属性名前加上一些只有特定浏览器才能识别 hack 前缀，以达到预期的页面展现效果。

前缀标识	兼容浏览器	写法
<code>_</code>	IE6	<code>_background:blue;</code>
<code>+</code> 、 <code>*</code>	IE6/7	<code>+background:blue;</code>
<code>\9</code>	IE6/7/8/9	<code>background:blue\9;</code>
<code>\0</code>	IE8/9/10/11	<code>background:blue\0;</code>

2. 选择器前缀法
 - 选择器前缀法是针对一些页面表现不一致，或需要特殊对待的浏览器，在 CSS 选择器前加上一些只有某些特定浏览器才能识别的 hack 前缀。

前缀标识	兼容浏览器	写法
<code>*html</code>	IE6	<code>*html .box{ width:100px;height:100px;}</code>
<code>*+html</code>	IE7	<code>*+html .box{ width:100px;height:100px;}</code>
<code>:root</code>	IE9及现代浏览器	<code>:root .box{ width:100px;height:100px;}</code>

3. IE条件注释法(了解)
 - IE10以上已经不支持注释法

前缀标识	兼容浏览器	写法(是写在 <code>body</code> 内容区域)
<code><--[if IE]>...<![endif]></code>	IE	<code><--[if IE]> <div></div> <![endif]></code>
<code><--[if IE 7]>...<![endif]></code>	IE7	

前缀标识	兼容浏览器	写法(是写在 <code>body</code> 内容区域)
<code><--[if lte IE 7]>...<![endif]></code>	IE7及以下	
<code><--[if gte IE 7]>...<![endif]></code>	IE7及以上	
<code><--[if ! IE 7]>...<![endif]></code>	非IE7	

- IE低版本常见 bug
 - `opacity` 透明度 IE8 及以下版本不识别
 - 解决方法: `filter:alpha(opacity=50);`
 - IE6 下的双边距 bug (同时拥有浮动和外边距的块元素会有双倍外边距)
 - 解决方法: `_display:inline;`
 - IE6 下的最小高度 bug (IE6 最小高度为 19px)
 - 解决方法: `overflow:hidden;`
 - IE9 及以下版本 图片链接有边框问题(`<a> `)
 - 解决方法: `border:none;`

10. 渐进增强 与 优雅降级

- 渐进增强: 针对低版本的浏览器进行构建页面, 保证最基本的功能, 然后再针对高级浏览器进行效果、交互等改进和追加功能, 以达到更好的用户体验。
 - 先兼容IE6/7... 然后进行界面优化做更高级的界面
- 优雅降级: 一开始就构建完整的功能, 然后再针对低版本的浏览器进行兼容。
 - 先做更高级的界面, 然后想办法吧IE6/7...也做成这种界面 (可以用图片遮罩)

11. 网页布局扩展

- `margin` 扩展: (内联元素支持 `margin` 左右 和 `padding` 上下左右(`padding` 会覆盖))
 - 没有固定宽时,负值的 `margin-left` 和 `margin-right` 都是可以增加宽度, 正值的相反(减少宽度);
 - 在有固定宽时, `margin-left` 和 `margin-right` 不会增加宽度, 只会产生位移。
 - 注意:
 - `margin-top` 为负值不会增加高度, 只会产生向上位移
 - `margin-bottom` 为负值不会产生位移, 会减少自身的供 `css` 读取的高度
 - 在有固定宽度和高度的时候, `margin-left\right\top` 都会产生位移, 此位移不脱离文档流, 但是会让所有跟随的元素都会被位移
 - `margin-bottom` 位移: 一个父级元素内有两个内联块子元素, 两个子元素设置宽高, 第一个子元素设置 `margin-bottom` 负值, 第二个子元素不设置, 就会看到第一个子元素向下位移。
- 版心, 通栏布局是最基础
- 等高布局
 - 利用 `margin-bottom` 负值和 `padding-bottom` 正值配合
- 三列布局, 左右固定, 中间自适应

1. BFC 方式：左边的左浮动，右边的右浮动，中间内容加 `overflow:hidden`;
2. 定位
3. 浮动（双飞翼布局、圣杯布局）
 - 双飞翼布局：先写中间部分（宽度要 `100%`，不然浮动后的宽度是根据内容决定的），三个容器都浮动，设置左侧右侧 `margin` 负值，设置中间左右 `margin` 的值
 - 圣杯布局：把父容器设置 `margin` 左右值，然后同理，最后用 `transform: translate()` 或者 `position: relative`; 偏移
4. flex 弹性