# 1 CSS3简介

## CSS3的现状

CSS3在CSS2基础上，增强或新增了许多特性， 使得Web开发变得更为高效和便捷。

1、浏览器支持程度差，需要添加私有前缀

2、移动端支持优于PC端

3、应用相对广泛

## 手册使用

{} 表示范围 [] 表示全部可选项

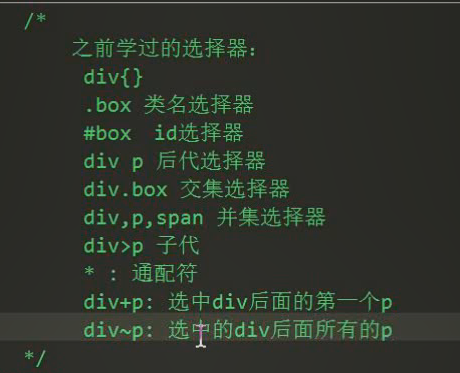
|| 表示或者 | 表示多选一

？ 表示0个或者1个 \* 表示0个或者多个

# 2 选择器

## 常见选择器

CSS3选择器与jQuery中所提供的绝大部分选择器兼容。



## 2.2 属性选择器

其特点是通过属性来选择元素，具体有以下5种形式：（E表示一个标签）

div[mydemo] class名 = mydemo 的div，没有确定值

div[attr=”value”] class值 = value的div

div[attr~=”value”] class值包含value字符

div[attr\*=”value”] class值包含value字符并且在“任意”位置

div[attr^=”value”] class值包含value字符并且在“开始”位置

div[attr$=”value”] class值包含value字符并且在“结束”位置

div[attr|="value"] 指定了属性名，并且属性值是value或者以“value-”开 头的值（比如说zh-cn）

## 2.3 伪类选择器

除了:link、:active、:visited、:hover，CSS3又新增了其它的伪类选择器。

### 2.3.1 结构伪类

以某元素相对于其父元素或兄弟元素的位置来获取元素

E:first-child 第一个子元素

E:last-child 最后一个子元素

E:nth-child(n) 第n个子元素，计算方法是E元素的全部兄弟元素；

E:nth-last-child(n) 同E:nth-child(n) 相似，只是倒着计算；

E:nth-of-type(n) 表示E父元素中的第n个字节点，且类型为E

E:nth-last-of-type(n)表示E父元素中的第n个字节点，且类型为E,从后向前计算

E:empty 没有任何子节点（包括空格）的E元素,(子节点包含文本节点)

n遵循线性变化，其取值0、1、2、3、4、... 但是当n<=0时，选取无效。

n可是多种形式：nth-child(2n)、nth-child(2n+1)、nth-child(-1n+5)等；

代码示例：

|  |
| --- |
| li:nth-child(2n-1){ 选中所有的奇数的li  color: red; }  li:nth-child(7n){ 选中所有的7 的倍数的li   color: red; }  li:nth-child(-1n+5){ 选中前面五个  color: red; }  li:nth-last-child(-1n+5){ 选中后面五个  color: red; }  li:nth-child(even){ 所有的偶数  color:red }  li:nth-child(odd){ 所有的奇数  color:blue;  } |

### 2.3.2 input相关伪类

E:target 结合锚点进行使用，处于当前锚点的元素会被选中；

E:enabled E:disabled

E:checked

示例：E:target

div {  
 width: 200px;  
 height: 200px;  
 background-color: yellowgreen;  
 display: none;  
 }  
 div:target {  
 display: block;  
 }  
  
  
<a href="#div1">显示div1</a>  
<a href="#div2">显示div2</a>  
<div id="div1">1</div>  
<div id="div2">2</div>

示例：E:enabled E:disabled

input:enabled {  
 color: greenyellow;  
}  
input:disabled {  
 color: red;  
}

<input type="" value="请输入" disabled>

示例：

<style>  
 label {  
 float: left;  
 margin: 0 5px;  
 overflow: hidden;  
 position: relative;  
 }  
 label input {  
 position: absolute;  
 left: -50px;  
 top: -50px;  
 }  
 span {  
 float: left;  
 width: 50px;  
 height: 50px;  
 border: 1px solid #1F89bf;  
 }  
 input:checked~span {  
 background-color: yellowgreen;  
 }  
 </style>  
  
<label>  
 <input type="radio" name="tab">  
 <span class="div1"></span>  
</label>  
<label>  
 <input type="radio" name="tab">  
 <span class="div1"></span>  
</label>  
<label>  
 <input type="radio" name="tab">  
 <span class="div1"></span>  
</label>

### 2.3.3 :not

p:not(.p0) {  
 background-color: #2aabd2;  
 }

<p>1111</p>  
<p>1111</p>  
<p class="p0">1111</p>

## 2.4 伪元素选择器

CSS2中 E:before或者E:after，是属于伪类的，并且没有伪元素的概念，CSS3中 提出伪元素的概念，如 E::before和E::after，并且归属到了伪元素当中，伪类里就不再存在E:before或者 E:after伪类，这样做的目的是用来做兼容处理。

E::before、E::after是一个行内元素，需要转换成块元素，当然这里只写一个冒号浏览器仍然能识别。

E::first-letter文本的第一个字母或字（如中文、日文、韩文等）；

E::first-line 文本第一行； 文本第一行高亮；

E::selection 可改变选中文本的样式；

":" 与 "::" 区别在于区分伪类和伪元素

|  |
| --- |
| */\* 选中第一个字母 \*/* **li**::**first-letter**{  **color**: **red**; }  */\* p内容最后加入content文本 \*/*  **p**::**after**{  **content**: ‘haha’; } |

重要案例：

p::after {  
 content: attr(hiatt);  
 }

<p hiatt="haha">1111</p>

# 4盒模型

## 4.1 简介

CSS3中可以通过box-sizing 属性来指定盒模型，根据不同的盒模型，采用不同的盒子大小计算方式，优点是兼容性好。

box-sizing 指定盒模型类型

content-box： 盒子宽度 = 设置的width+border宽+padding值

border-box： 盒子宽度 = 设置的width 怪异模式

同时CSS3引入了私有化前缀机制，即针对不同内核采用不同的属性前缀。

-webkit- webkit内核

-moz- 火狐

-ms- IE

-o- 欧朋

在后续的属性值中，大多数属性在书写时，都要加上前缀，如：

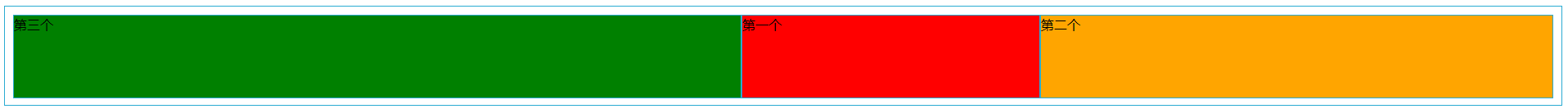
-webkit-box-reflect:below

## 4.2 弹性盒模型

<style>  
 .box {  
 height: 100px;  
 border: solid 1px #2aabd2;  
 padding: 10px;  
 display: -webkit-box;  
 }  
 .box div {  
 width: 100px;  
 height: 100px;  
 background-color: greenyellow;  
 border: 1px solid #2aabd2;  
 }  
 .box div:nth-of-type(1) {  
 -webkit-box-ordinal-group: 2; /\* 设置具体顺序 \*/  
 -webkit-box-flex: 1; /\* 设置伸缩比例 \*/  
 }  
 .box div:nth-of-type(2) {  
 -webkit-box-ordinal-group: 3; /\* 设置具体顺序 \*/  
 -webkit-box-flex: 2;  
 }  
 .box div:nth-of-type(3) {  
 -webkit-box-ordinal-group: 1; /\* 设置具体顺序 \*/  
 -webkit-box-flex: 3;  
 }  
</style>

<div class="box">  
 <div style="background-color: red ">第一个</div>  
 <div style="background-color: orange ">第二个</div>  
 <div style="background-color: green ">第三个</div>  
</div>

效果：



使用弹性盒模型的时候，盒子元素必须要加display:box 或 display:inline-box，且属性名前要加上浏览器内核前缀：

display: -webkit-box;

常用属性列表：

box-orient 定义盒模型的布局方向，默认

horizontal 水平显示（默认），

vertical 垂直显示

box-direction 元素排列顺序，

normal 正序（默认），

reverse 反序

box-ordinal-group 设置元素的具体位置

box-flex 定义盒子的弹性空间

子元素的尺寸 =

盒子尺寸\*子元素box-flex值 / 所有子元素box-flex属性值之和

box-pack 对盒子富裕的空间进行管理

start 所有子元素在盒子左侧显示，富裕空间在右侧

end 所有子元素在盒子右侧显示，富裕空间在左侧

center 所有子元素居中

justify 富余空间在子元素之间平均分布

box-align 在垂直方向上对元素的位置进行管理

star 所有子元素在据顶

end 所有子元素在据底

center 所有子元素居中

## 4.3 阴影

box-shadow:[inset] x y blur [spread] color

参数一 新增投影方式

inset：内投影

不给：外投影

x、y： 阴影偏移

blur： 模糊半径

spread： 扩展阴影半径，先扩展原有形状，再开始画阴影

color： 阴影颜色

## 4.4 倒影

倒影：box-reflect:below 10px（只支持-webkit-）

参数一 倒影方式

above|below|left|right;

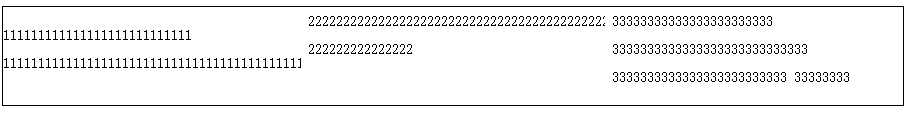
参数二 倒影距离

# 5 布局

## 5.1 分栏布局

<style>  
 .wrap {  
 width: 900px;  
 border: 1px solid #000;  
 font: 14px/28px "宋体";  
 -webkit-column-width: 250px; /\* 设置一栏每栏宽度 \*/  
 }  
 </style>  
<div class="wrap">  
 <p>111111111111111111111111111  
 11111111111111111111111111111111111111111111111111</p>  
 <p>2222222222222222222222222222222222222222222  
 222222222222222</p>  
 <p>33333333333333333333333  
 3333333333333333333333333333  
 3333333333333333333333333  
 33333333</p>  
</div>

运行效果：



column-width: 250px; 以250像素自动设置宽度

column-count: 3 自动以3栏布局

column-gap: 12px 栏目距离

column-rule:1px solid #fff 栏目间隔线

注意：绝大多数新增的属性都是需要添加浏览器私有化前缀，但是布局这里只有webkit内核支持。

## 5.2 响应式

我们引入多个样式表，依据媒体查询，对不同的设备采用不同的样式表（不兼容IE678）：

|  |
| --- |
| <link rel="stylesheet" type="text/css" href="styleA.css" media="screen and (min-width: 800px)">  <link rel="stylesheet" type="text/css" href="styleB.css" media="screen and (min-width: 600px) and (max-width: 800px)">  <link rel="stylesheet" type="text/css" href="styleC.css" media="screen and (max-width: 600px)"> |

横竖屏切换CSS：（:landscape横屏）

|  |
| --- |
| <link rel=”stylesheet” media=”all and (orientation:landscape)”href=”landscape.css”>  <link rel=”stylesheet” media=”all and (orientation:portrait)” href=”portrait.css”> |

单独对元素设置：

<style>  
 /\*整体 黄绿\*/  
 .wrap {  
 width: 800px;  
 height: 800px;  
 background-color: greenyellow;  
 }  
 /\*<400 变红\*/  
 @media screen and (max-width:400px) {  
 .wrap { background-color: orange}  
 }  
 /\*400-800 变红\*/  
 @media screen and (min-width:400px) and (max-width:600px) {  
 .wrap { background-color: red}  
 }

/\*600以上 更改样式--这时采用引入CSS方式 \*/  
 @import url("index.css") screen and (min-width: 600px);

</style>  
  
 <div class="wrap">  
 wrap  
 </div>

媒体类型

all 所有媒体

braille 盲文触觉设备

embossed 盲文打印机

print 手持设备

projection 打印预览

screen 彩屏设备

speech '听觉'类似的媒体类型

tty 不适用像素的设备

tv 电视

关键字

and

not not关键字是用来排除某种制定的媒体类型

only only用来定某种特定的媒体类型

移动端meta

<meta name="viewport" content="" />

width [pixel\_value | device-height]

height [pixel\_value | device-height]

user-scalable 是否允许缩放 （no||yes）

initial-scale 初始比例

minimum-scale 允许缩放的最小比例

maximum-scale 允许缩放的最大比例

target-densitydpi [dpi\_value | device-dpi | high-dpi | medium-dpi | low-dpi]