

# DAY12 C3选择器、伪类、阴影、圆角、渐变和过渡

2023年4月6日 18:01

## CSS3的介绍

### 渐进增强

基于低版本浏览器开发，逐步添加新功能 向前看 政府银行类项目  
优雅降级

基于高版本开发，逐步兼容低版本浏览器 向后看 新兴项目类型

主流的浏览器	内核	兼容性前缀
chrome	webkit(旧)/blink(新)	-webkit
safari	webkit	-webkit
欧鹏	presto	-o
ie	trident	-ms
火狐	gecko	-moz

以border-radius为例

- webkit-border-radius
- o-border-radius
- ms-border-radius
- moz-border-radius

## 选择器

### 相邻兄弟选择器 css2

E + F 选中紧贴在E后面的兄弟元素F

```
<style>
  p {
    height: 40px;
    line-height: 40px;
    border: 1px solid #000;
  }
  /* 选中紧贴在E后面的兄弟元素F */
  p:hover+p {
    background-color: yellow;
  }
</style>

<body>
```

```
<p>1</p>
<hr>
<p>2</p>
<p>3</p>
<p>4</p>
<p>5</p>
</body>
```

## 普通兄弟选择器 css3

E ~ F 选中E后面的兄弟元素F

```
<style>
  p {
    height: 40px;
    line-height: 40px;
    border: 1px solid #000;
  }
  /* 选中E后面的兄弟元素F */
  p:hover~p {
    background-color: yellow;
  }
</style>

<body>
  <p>1</p>
  <hr>
  <p>2</p>
  <p>3</p>
  <p>4</p>
  <p>5</p>
</body>
```

## 属性选择器

css2

E[attr] 选中具有attr属性的元素E

E[attr="value"] 选中具有attr属性,并且属性值是value的元素E

css3

E[attr^="value"] 选中具有attr属性,并且属性值是value开头的元素E

E[attr\$="value"] 选中具有attr属性,并且属性值是value结尾的元素E

E[attr\*="value"] 选中具有attr属性,并且属性值中有value的元素E

```
<style>
```

```

/* 组合选择器 */
/* 选中title属性的元素 */
[title] {
    border: 5px solid yellow;
}
/* 选中title属性的元素img */
img[title] {
    border: 5px solid red;
}
img[title="dog"] {
    border: 5px solid green;
}
/* 开头 */
img[title^="dog"] {
    border: 5px solid purple;
}
/* 结尾 */
img[title$="cat"] {
    border: 5px solid orange;
}
/* 含有 */
img[title*="dog"] {
    border: 5px solid blue
}
</style>

```

```

<body>
  <a href="" title="">文字文字文字</a>
  
  
  <!-- css3 -->
  
  
  
</body>

```

## 伪类

### 结构伪类

清一色

E:nth-child(n) 同级中第n个元素

E:nth-last-child(n) 同级中倒数第n个元素

E:first-child 同级中第1个元素

**E:last-child** 同级中最后个元素

混搭

**E:nth-of-type(n)** 同级中同类型的第n个元素

**E:nth-last-of-type(n)** 同级中同类型的倒数第n个元素

**E:first-of-type** 同级中同类型的第1个元素

**E:last-of-type** 同级中同类型的最后个元素

**E:only-of-type** 同级中同类型的唯一一个元素

其他类型

**:root** 选中根元素 html

**:empty** 选中空元素

**E:only-child** 选中同级中唯一一个元素

```
<style>
  :root {
    /* background-color: red; */
  }
  html {
    /* background-color: red; */
  }
  div {
    width: 100px;
    height: 100px;
    background-color: pink;
  }
  div:empty {
    border: 5px solid #000;
  }
  div:only-child {
    border: 5px solid red;
  }
</style>
```

```
<body>
  <!-- 丁克 -->
  <div></div><br>
  <!-- 独生子 -->
  <div>
    <div>div1</div>
  </div><br>
  <!-- 多子女 -->
  <div>
    <div>div2</div>
    <div>div3</div>
```

```
</div>
</body>
```

奇数偶数行

**E:nth-child(n)**

n

数字 1 2 3 4...

表达式  $2n$   $2n+1$   $3n$   $4n...$

关键词

even 偶数

odd 奇数

## 目标伪类

**E:target** 选中锚点元素,并且这个锚点处于激活状态下

```
<style>
  div {
    width: 200px;
    height: 200px;
    border: 3px solid #000;
    float: left;
    font-size: 30px;
    line-height: 200px;
    text-align: center;
  }
  div:target {
    background-color: pink;
  }
</style>
```

```
<body>
  <a href="#box1">box1</a>
  <a href="#box2">box2</a>
  <a href="#box3">box3</a>
  <div id="box1">1</div>
  <div id="box2">2</div>
  <div id="box3">3</div>
</body>
```

## UI状态伪类

**E:enabled** 元素处于可用状态 ----表单控件可用

**E:disabled** 元素处于禁用状态 ----表单控件可用, 元素需要设置disabled属性

**E:checked** 元素处于选中状态 ---单选 多选

```
<style>
```

```

    input[type="text"]:enabled {
        background-color: pink;
    }
    input[type="password"]:disabled {
        background-color: yellow;
    }
    input[type="checkbox"]:checked {
        width: 200px;
        height: 200px;
    }
</style>

```

```

<body>
  <!-- 表单控件 -->
  <p>
    <input type="text">
    <input type="password" disabled>
  </p>
  <!-- 多选 -->
  <p>
    <input type="checkbox">西瓜
    <input type="checkbox">恐龙蛋
  </p>
</body>

```

## 否定伪类

**E:not(val)** 选中不是val选择器的元素E

```

<style>
  li {
    width: 400px;
    height: 40px;
    line-height: 40px;
  }
  li:not(:last-of-type) {
    border-bottom: 1px solid #000;
  }
</style>

```

```

<body>
  <!-- ul>li*5{文本文本文本文本} -->
  <ul>
    <li>文本文本文本文本</li>
    <li>文本文本文本文本</li>
    <li>文本文本文本文本</li>

```

```
<li>文本文本文本文本</li>
<li>文本文本文本文本</li>
</ul>
</body>
```

## 阴影

### 文字阴影

**text-shadow:**水平偏移 垂直偏移 模糊程度 颜色

- 水平偏移 垂直偏移必须写（默认值 0）
- 模糊程度不能写负数
- 颜色默认为文本颜色
- 多阴影用逗号隔开

```
/* 多阴影 */
text-shadow: 50px 50px 10px red, -50px -50px 30px gold;
```

### 盒子阴影

**box-shadow:**水平偏移 垂直偏移 模糊程度 阴影大小 颜色 (内阴影 inset)

- 水平偏移 垂直偏移必须写（默认值 0）
- 模糊程度不能写负数
- 阴影大小默认为盒子大小
- 颜色默认为文本颜色
- 多阴影用逗号隔开

```
/* (内阴影 inset) */
box-shadow: 0px 0px 10px 10px red inset;
/* 多阴影 */
box-shadow: 30px 30px 10px -10px red, 60px 60px 10px -10px gold;
```

## 圆角

**border-radius:**

数值

百分比 参考元素的宽高

### 圆角的简写

一个值 四个角

两个值 1-左上右下 2-右上左下

三个值 1-左上 2-右上 3-右下

四个值 1234-顺时针-左上 右上 右下 左下

## 取消圆角

`border-radius:0`

## 单边的圆角 (top/bottom与left/right不能调换位置)

`border-top/bottom-left/right-radius`

## 常见的圆角

圆形、椭圆形、胶囊形

**圆形**: 1.元素的宽高一致

2.圆角设置为宽高的一半或者50%

`border-radius: 50%;`

**椭圆形**: 1.元素的宽高不一致

2.圆角设置为50%

**胶囊形**: 1.元素的宽高不一致

2.圆角设置为高度的一半

## 渐变

### 线性渐变

`background-image: linear-gradient(方向,颜色1,颜色2....)`

颜色可以写多个

方向有多种写法

#### 01.默认值

从上到下

#### 02.上下左右

`to left` 去到左边 从右到左

`to right`

`to top`

`to bottom` 默认值

#### 03.对角线

`to right top` 去到右上角 从左下角到右上角

`to right bottom`

`to left bottom`

`to left top`

#### 04.角度

`0deg` 从下到上

`180deg` 默认值 从上到下



`background-image: linear-gradient(to top right, red, green);`

## 颜色突变

两段颜色在同一个位置

```
/* 渐变突变 */
/* 70%到75%之间只有5%的渐变区域 */
background-image: linear-gradient(red 70%, green 75%);
/* 70%到70%之间只在一根线上渐变，只要渐变区域够小，可以忽略不计，得到纯色色块，也叫突变*/
background-image: linear-gradient(red 70%, green 70%);
```

## 径向渐变

`background-image: radial-gradient(方向, 颜色1, 颜色2...)`

渐变的形状

默认值 椭圆形

圆形 circle

at position 渐变中心点的位置/方向

水平 垂直

数值

百分比 参考元素的宽高

方位词 （可以混搭）

left right center （默认 center）

top bottom center （默认 center）

```
/* 03.方位词 */
/* left right center */
/* top bottom center */
background-image: radial-gradient(circle at right top, red, green);
/* 只有方位词,顺序可以换 */
background-image: radial-gradient(circle at top right, red, green);
/* 可以混搭 */
background-image: radial-gradient(circle at 100px bottom, red, green);
/* 如果只写一个值，默认另外一个值是center */
background-image: radial-gradient(circle at bottom, red, green);
```

## 重复线性渐变

`background-image: repeating-linear-gradient(方向, 颜色1, 颜色2)`

!颜色必须添加数值或百分比

`background-image: repeating-linear-gradient(red 5%, blue 20%);`

## 重复径向渐变

`background-image: repeating-radial-gradient(方向, 颜色1, 颜色2)`

!颜色必须添加数值或百分比

`background-image: repeating-radial-gradient(red 5%, blue 20%);`

## 过渡

### 过渡 transition

`transition-property`: 要发生过渡的css属性

默认值 all

`transition-duration`: 持续时间

必须写

s ms 1s=1000ms

默认值 0s

`transition-delay`: 延迟时间

s ms 1s=1000ms

默认值 0s

过渡是主动触发的，一般过渡只有在鼠标移入之后才有效

所以过渡属性要写在元素的默认状态中

```
<style>
div {
  width: 500px;
  height: 400px;
  background-color: pink;
  /* transition-property: width, background-color; */
  transition-duration: 4s;
  /* transition-delay: 2s; */
}
div:hover {
  width: 600px;
  background-color: yellow;
}
```

```
</style>
```

```
<body>  
  <div></div>  
</body>
```

## 过渡中间值

慢慢消失 用opacity: 0;

```
<style>  
  .baba {  
    width: 400px;  
    height: 500px;  
    background-color: green;  
    margin: 50px;  
  }  
  .son {  
    width: 200px;  
    height: 200px;  
    background-color: blue;  
    transition-duration: 2s;  
  }  
  .baba:hover .son {  
    /* display: none; */  
    /* visibility: hidden; */  
    /* 只有中间值的属性才可以发生过渡 */  
    opacity: 0;  
  }  
</style>
```

```
<body>  
  <!-- 鼠标放到baba son慢慢消失 -->  
  <div class="baba">  
    <div class="son"></div>  
  </div>  
</body>
```

## 过渡的运动状态

transition-timing-function

ease 默认值 先快后慢

linear 匀速

```
<style>  
  .father {  
    width: 800px;  
    height: 300px;  
    border: 1px solid #000;  
    margin: 30px auto;
```

```

    }
    .father div {
      width: 100px;
      height: 100px;
      margin: 20px;
      background-color: pink;
      transition-duration: 10s;
    }
    .ease {
      transition-timing-function: ease;
    }
    .linear {
      transition-timing-function: linear;
    }
    .father:hover div {
      width: 500px;
    }
  }
</style>

<body>
  <div class="father">
    <div class="ease">先快后慢</div>
    <div class="linear">匀速</div>
  </div>
</body>

```

## 过渡简写

transition:要发生过渡的属性 持续时间 延迟时间 运动状态;

- 顺序不强调
- 持续时间和延迟时间有先后顺序 持续时间在前
- 持续时间必须写
- 最简单的过渡

transition:持续时间

## 回流和重绘

dom 树

回流:元素发生了高度 宽度的变化, 导致元素以及元素后面的所有元素进行重新排列

元素在空间上发生了变化

重绘:仅仅影响了当前元素的变化, 对其他元素没有影响

背景色 文本颜色 2d变换