



CSS3 元素选择器

讲师：许井龙

ngsteel@qq.com

一. :target 目标伪类选择器

1. 巩固练习：文章目录
2. 巩固练习：图片切换

二. 结构化伪类选择器

1. 巩固练习：尺子

三. :checked伪类选择器

1. 扩展知识：appearance 样式

四. 属性选择器

五. 伪元素选择器

1. ::before
2. ::after
3. ::first-letter
4. ::first-line
5. ::selection
6. 扩展知识：counter(), counters()

六. 多媒体选择器

1. @media
2. 扩展知识：attr()

七、兄弟元素选择器

当用户点击a元素，该元素锚链接指向的元素会显示在视口范围

`` 一个锚链接 ``

`<p id= "e1" >`

当用户点击a元素，如果元素的href= "#e1" ，此时我具有:target状态

`</p>`

`p:target{`

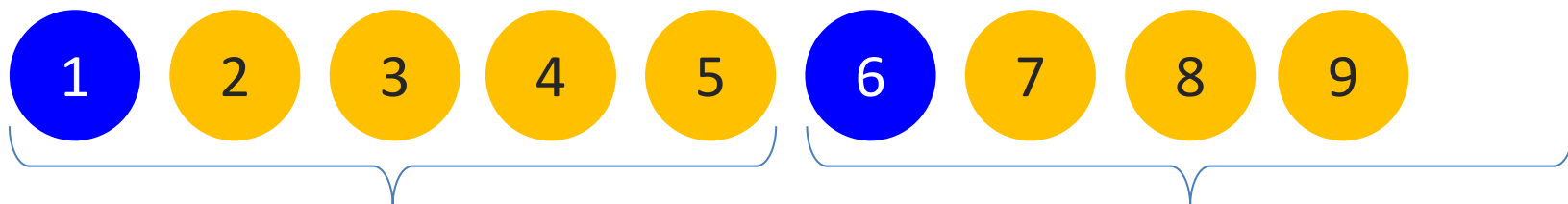
//编写你的CSS样式

`}`

- 文章列表
- 图片切换

- E:nth-of-type(参数),
 - E 表示CSS选择器。E选择器选中的元素 **(必须为相同类型, 所有被该选择器选择的层级)**, 根据后面的参数, 再次“筛选”那个位置上的元素
 - 参数可选择值
 1. number 选择number指定位置上的元素
 2. odd 选择奇数位置上的元素
 3. Even 选择奇数位置上的元素
 4. $an+b$ 从第b个元素开始, 每a个一组, 选取第第一个元素

$5n+1$



分组方向

$5n+4$



分组方向

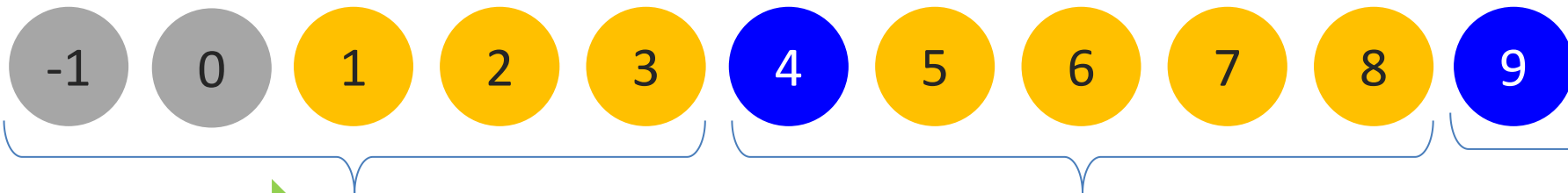
$-5n+8$



分组方向

编号-1, 0的两个元素不是真正的元素, 占位用

$5n-1$



分组方向

- E:nth-child(参数),
 - E 表示CSS选择器。E选择器选中的元素，根据后面的参数，再次“筛选”那个位置上的元素
 - 参数可选择值
 1. **number** 选择number指定位置上的元素
 2. **odd** 选择奇数位置上的元素
 3. **even** 选择偶数位置上的元素
 4. **an + b** 从第b个元素开始，每a个一组，选取第a个元素

```
span:nth-of-type(even) {  
    color: red;  
}
```

```
<div class="box">
```

```
    <div>1111</div>
```

1

```
    <span>2222</span>
```

```
    <p>3333</p>
```

2

```
    <span>4444</span>
```

```
    <p>5555</p>
```

```
    <p>666</p>
```

3

```
    <span>7777</span>
```

```
</div>
```



1111

2222

3333

4444

5555

66666

77777

- 1.选中同一层级的所有span元素。
- 2.在选中后的span, 重新计算位置, 重新设置位置中位于偶数位置。

```
span:nth-child(even) {  
    color: red;  
}
```

```
<div class="box">
```

1 `<div>1111</div>`

2 `2222`

3 `<p>3333</p>`

4 `4444`

5 `<p>5555</p>`

6 `<p>666</p>`

7 `7777`

```
</div>
```



1111

2222

3333

4444

5555

77777

66666

选中同一层级的所有span元素。

在同一层级中位置必须在偶数位。

	E:nth-of-type	E:nth-child
1	E选择器选中的所有元素，且被选中的元素都在同一DOM层级。	
2	E选择器选中的所有元素， 重新设置位置序号 。(可以理解为，忽略其他不同类型元素的存在)。	E选择器选中的所有元素，被选中的元素必须在指定的位置且类型也必须相同

- : checked 当input type= 'checkbox' 被选中时。


```
input[type=checkbox] {  
    -webkit-appearance: none;  
    -moz-appearance: none;  
    appearance: none;  
}
```

使用CSS3的appearance属性改变元素的外观

- [attribute] 选择器用于选取带有指定属性的元素。

```
<input type="checkbox" name="city" id="city">
```

```
[type] {  
    选取所有含有type属性的元素  
}
```

- [attribute=value] 选择器用于选取带有指定属性和值的元素。

```
<input type="checkbox" name="city" id="city">
```

```
input[type=checkbox] {  
    选取input元素，但其type属性值必须实checkbox  
}
```

或者

```
input[name=city] {  
    选取input元素，但其name属性值必须实city  
}
```

- `[attribute^=value]` 选择器匹配属性值以指定值开头的每个元素。

```
<input type="checkbox" name="city" id="city" lang="en" >
```

```
input[type^=che] {  
    选取input元素，但其type属性值必须以che开头  
}
```

- [attribute\$=value] 选择器匹配属性值以指定值结尾的每个元素。

```
<input type="checkbox" name="city" id="city" lang="en" >
```

```
input[type$=box] {  
    选取input元素，但其type属性值必须以box结尾  
}
```


- [attribute~=value] 选择器用于选取属性值中包含指定词汇的元素。

```
<div class="box box1">.box1</div>
```

```
div[class~=box] {
```

 选取div元素，但其class属性值必须包含box单词（一个单词前后必须有空格）

```
}
```

- [attribute|=value] 选择器用于选取带有以指定值开头的属性值的元素。
- 注意：该值必须是整个单词，比如 lang="en"，或者后面跟着连字符，比如 lang="en-us"。

```
<div class="box box3" lang="en">.box3</div>  
<div class="box box4" lang="en-us">.box4</div>  
<div class="box box5" lang="us en">.box5</div>
```

```
div [lang|=en]{  
    此时只有box3 和 box4 被选中  
}
```

```
<div class=box1></div>
```

```
.box1::before{  
    content: "Hello";  
}
```

Hello

`World`

```
<div class=box1></div>
```

```
.box1::after{  
    content: "Hello";  
}
```

```
<span>World</span>
```

Hello

1. :: 是规范要求，实际开发中使用：，目的是得到IE8的支持
2. 必须配合content属性使用。
3. ::before ::after 相当于在元素内“动态”创建一个虚拟的inline元素，但该元素的文本内容无法被选中，该元素也无法通过JS获取其对应的DOM.

1. 解决子元素浮动造成的父元素高度塌陷。
2. 页面中“重复出现”的文本内容，且不需要JS操作。
3. 客制化目录前的序号或者符号[配合CSS3 新增的`counter()`, `counters()` 一起使用]

```
<ul>
  <li>item1</li>
  <li>item2</li>
  <li>item3</li>
  <li>item4</li>
</ul>
```



```
ul>li{
  counter-increment: num;
  list-style: none;
}
ul>li:before{
  content: counter(num) "、";
}
```

```
1、item1
2、item2
3、item3
4、item4
5、item5
6、item6
7、item7
8、item8
```

```
<ul>
```

```
<li>item1</li>
```

```
<li>item2
```

```
<ul>
```

```
<li>item-sub1</li>
```

```
<li>item-sub2</li>
```

```
<li>item-sub3</li>
```

```
</ul>
```

```
</li>
```

```
<li>item3</li>
```

```
<li>item4</li>
```

```
</ul>
```

```
ul{
```

```
  counter-reset: num;
```

```
}
```

```
ul>li{
```

```
  counter-increment: num;
```

```
  list-style: none;
```

```
}
```

```
ul>li:before{
```

```
  content: counters(num, "-") "、";
```

```
}
```



```
1、item1
2、item2
  2-1、item-sub1
  2-2、item-sub2
  2-3、item-sub3
3、item3
4、item4
```

- 选中元素第一行文本的第一个字母或者汉字

```
p:first-line{  
    font-size: 36px;  
    color: red;  
    font-family: "Microsoft YaHei UI";  
}
```

- 选中元素第一行文本

默认选中文本样式

富、年岁已高的导演们远离了舆论的漩

```
p::selection{  
    color: red;  
    background-color: yellow;  
}
```

注意：只能设置在块元素中

年岁已高的导演们远离了舆论的漩涡，而取

设置了::selection后

- 控制打印机页面样式

```
@media print {  
    .box{  
        你的样式  
    }  
}
```

- attr() 获取元素属性值

```
@media print {  
  a:before{  
    content: attr(href);  
  }  
}
```



网页

[百度](#)

打印模式

`百度`

<http://www.baidu.com>百度

.box1

.box2

.box3

.box4

.box5

.box3

是.box2的紧邻兄弟元素

.box2,.box3,.box4

是.box2的兄弟元素

```
.box{  
  width: 100px;  
  height: 100px;  
  background-color: #f0f0f0;  
}
```

```
.box2+div{  
  background-color: #00f;  
}
```

```
<div class="box box1">.box1</div>  
<div class="box box2">.box2</div>  
<div class="box box3">.box3</div>  
<div class="box box4">.box4</div>  
<div class="box box5">.box5</div>
```

.box1

.box2

.box3

.box4

.box5

.box2是参照物


```
.box{  
  width: 100px;  
  height: 100px;  
  background-color: #f0f0f0;  
}
```

```
.box2~div{  
  background-color: #f0f;  
}
```

```
<div class="box box1">.box1</div>  
<div class="box box2">.box2</div>  
<div class="box box3">.box3</div>  
<div class="box box4">.box4</div>  
<div class="box box5">.box5</div>
```



.box1

.box2

.box2是参照物

.box3

.box4

.box5

