



CSS选择器



内容

CSS选择器类型



选择器分组 p,h1

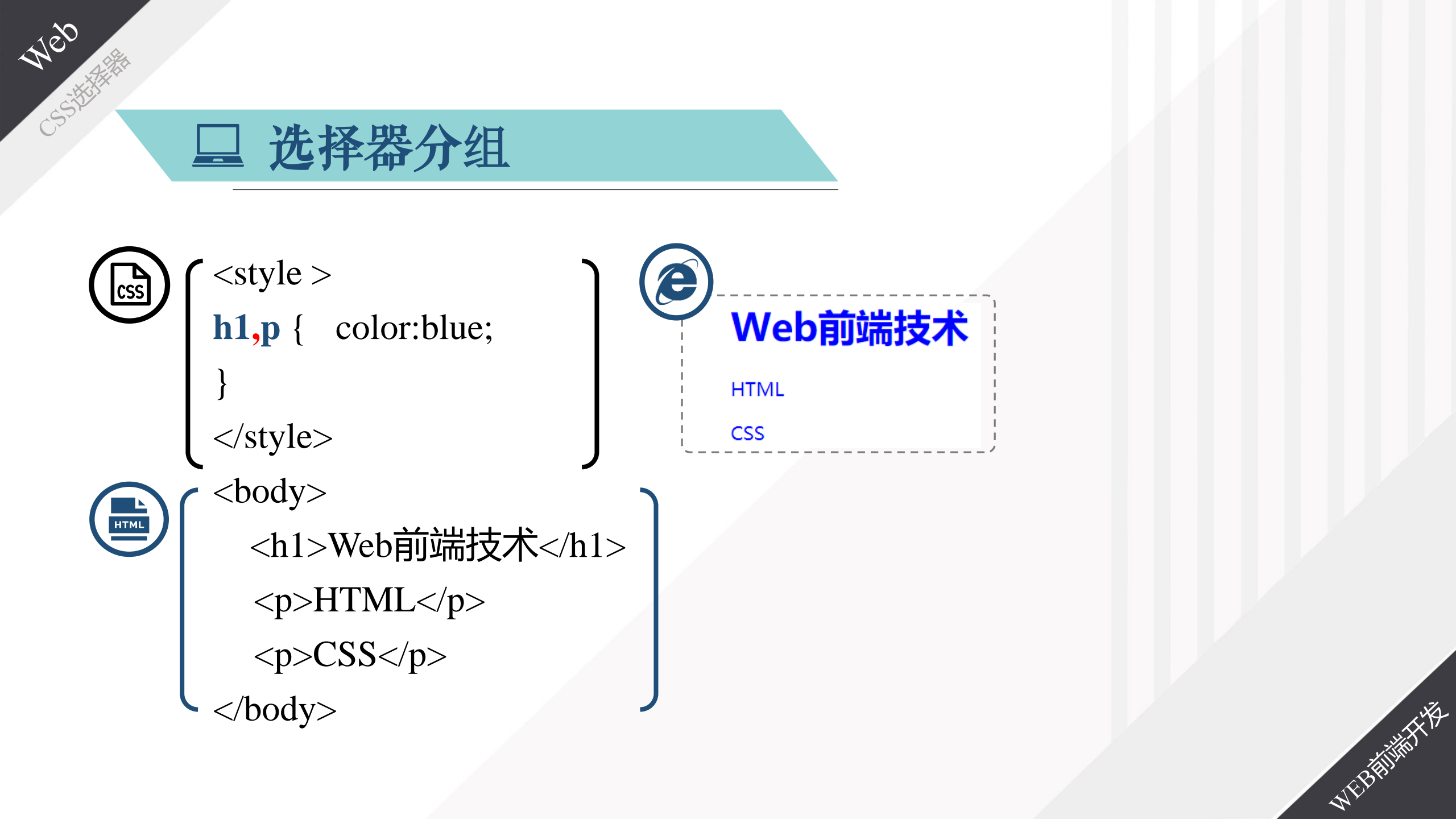
后代选择器 p strong

子元素选择器 p > strong

相邻兄弟选择器 h2+p

属性选择器 img[src]

```
p{  
  font-size:16px;  
  color:blue;  
}
```



选择器分组



```
<style >
h1,p { color:blue;
}
</style>
```



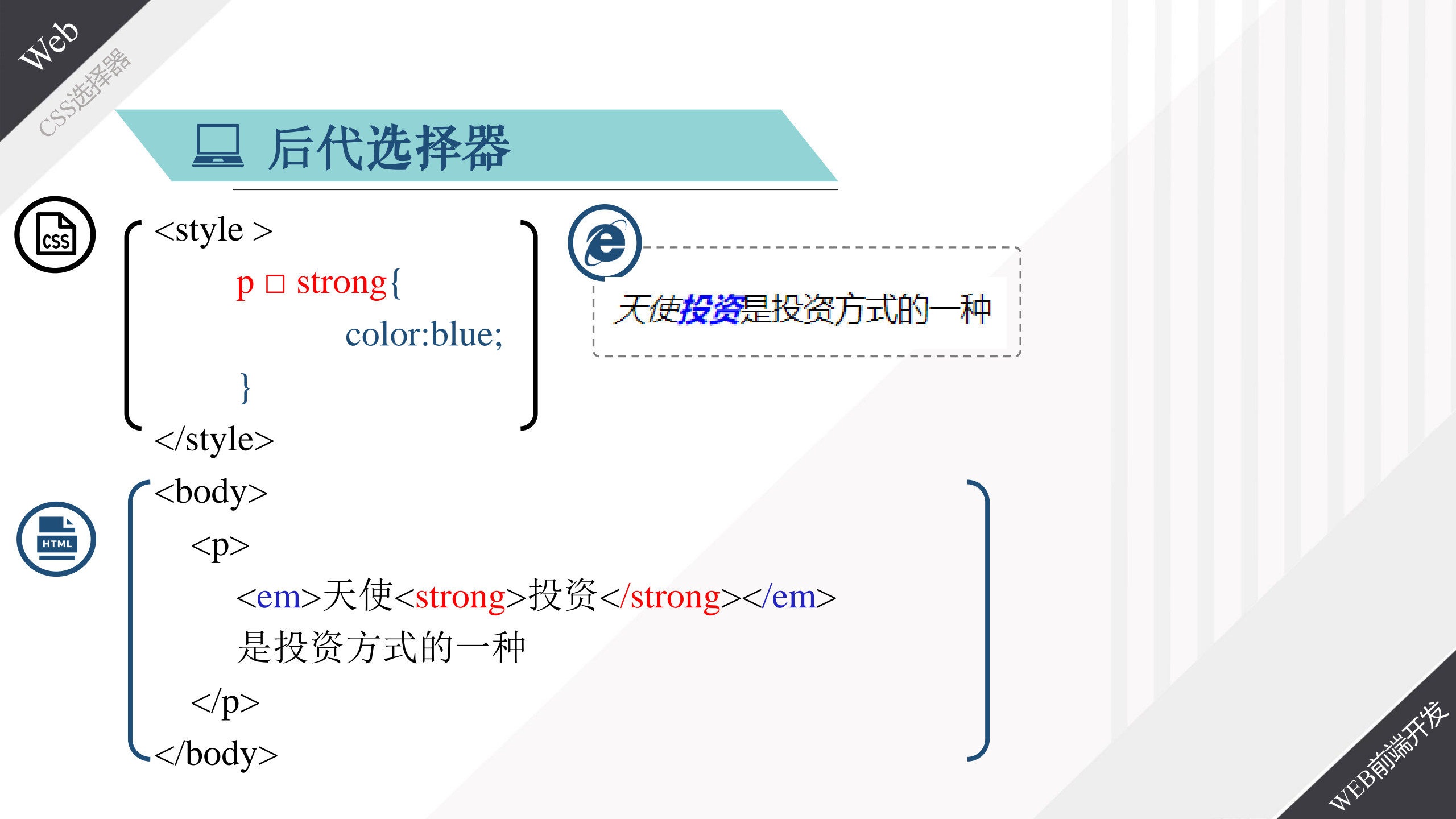
Web前端技术

HTML

CSS



```
<body>
  <h1>Web前端技术</h1>
  <p>HTML</p>
  <p>CSS</p>
</body>
```



后代选择器



```
<style >
```

```
  p □ strong {  
    color:blue;  
  }
```

```
</style>
```



天使**投资**是投资方式的一种



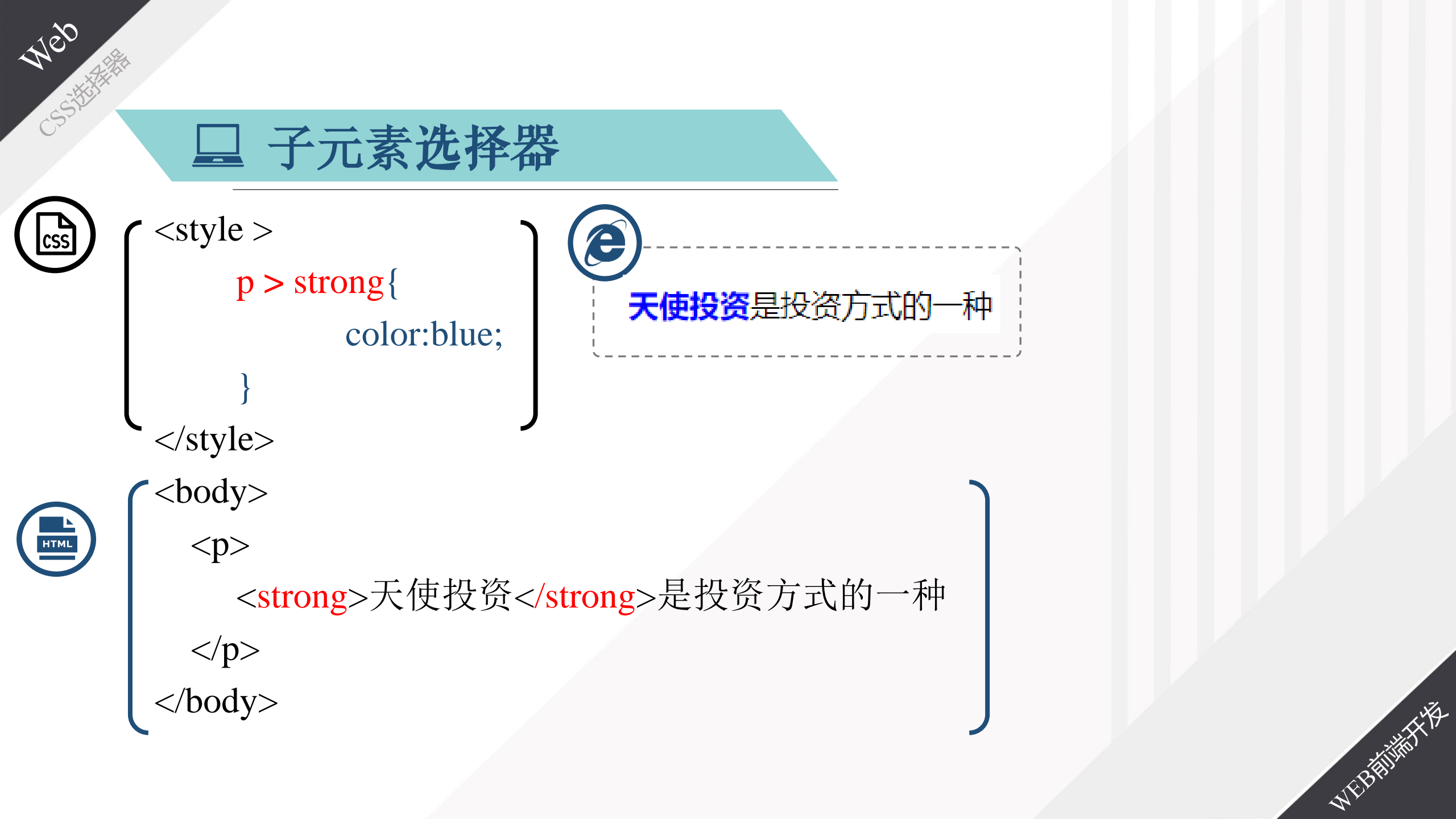
```
<body>
```

```
  <p>
```

```
    <em>天使<strong>投资</strong></em>  
    是投资方式的一种
```

```
  </p>
```

```
</body>
```



子元素选择器



```
<style >
```

```
  p > strong {  
    color:blue;  
  }
```

```
</style>
```



天使投资是投资方式的一种



```
<body>
```

```
  <p>
```

```
    <strong>天使投资</strong>是投资方式的一种
```

```
  </p>
```

```
</body>
```



相邻兄弟选择器



```
<style >  
  h2+p {  
    font-weight: bold;  
  }  
</style>
```



HTML

超文本标记语言

用于构建网页结构，添加页面内容。

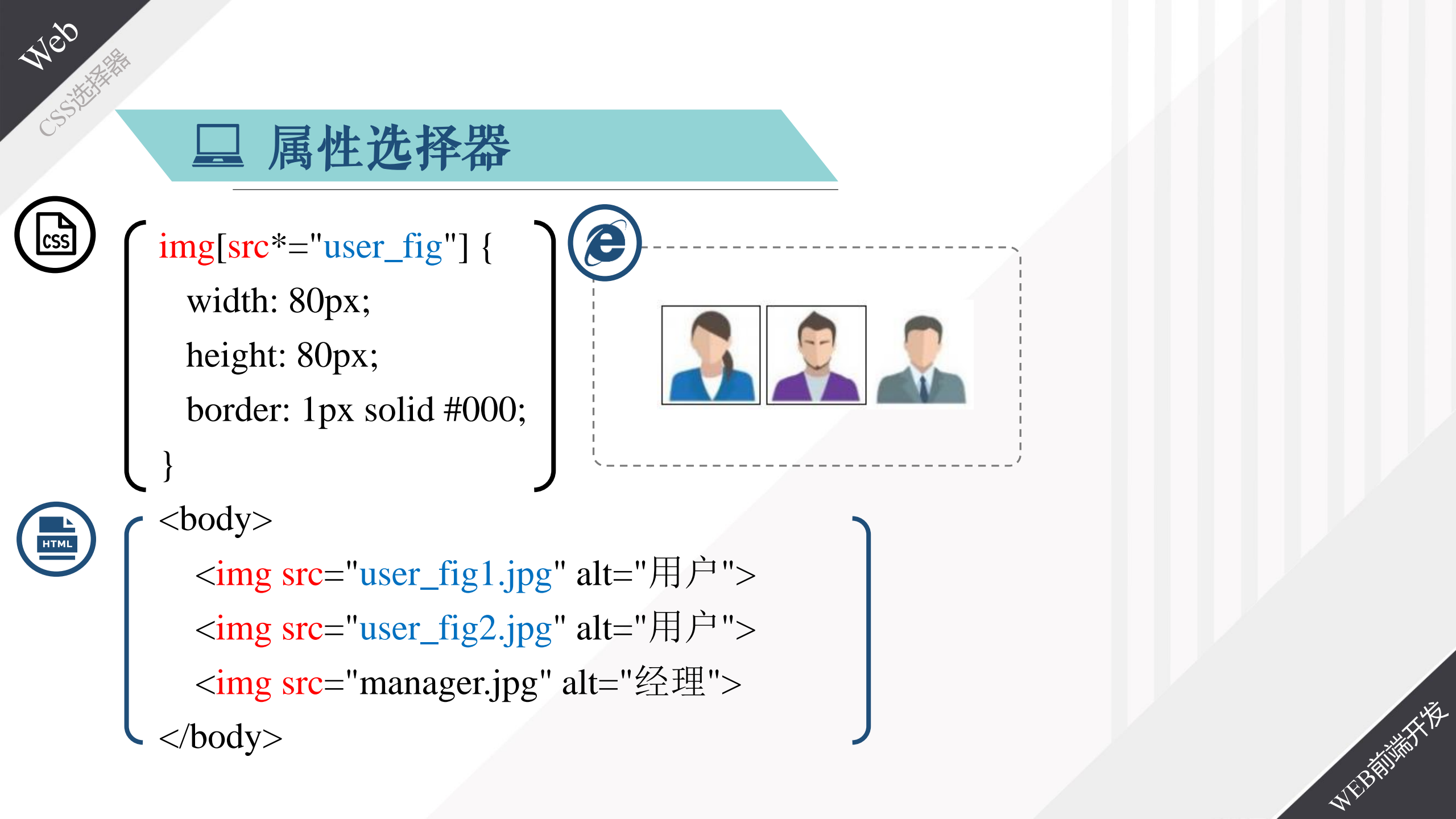


```
<body>  
  <h2>HTML</h2>  
  <p>超文本标记语言</p>  
  <p>用于构建网页结构，添加页面内容。</p>  
</body>
```



属性选择器

选择器	说明
元素[属性]	含有该属性的元素
元素[属性="val"]	属性取值val的元素
元素[属性~="val"]	属性取值列表中包含了val的元素
元素[属性^="val"]	属性值以val开头的元素
元素[属性\$="val"]	属性值以val结尾的元素
元素[属性*="val"]	属性值包含val的元素



属性选择器



```
img[src*="user_fig"] {  
  width: 80px;  
  height: 80px;  
  border: 1px solid #000;  
}
```



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




小结

选择器分组 p,h1

后代选择器 p strong

子元素选择器 p > strong

相邻兄弟选择器 h2+p

属性选择器 img[src]



伪类选择器

用于选择处于特定状态的元素，
效果等同于向文档某个部分
应用了一个类一样



```
.content:first-child {  
  background-color: blue;  
}
```

```
.content:last-child {  
  background-color: blue;  
}
```



```
<p class="content"></p>  
<p class="content"></p>  
<p class="content"></p>  
<p class="content"></p>  
<p class="content"></p>
```



:first-child和:last-child





伪类选择器

用于选择处于特定状态的元素，
效果等同于向文档某个部分
应用了一个类一样



```
.content:nth-child( ) {  
  background-color: blue;  
}
```



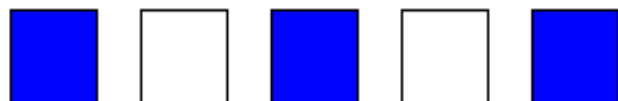
```
<p class="content"></p>  
<p class="content"></p>  
<p class="content"></p>  
<p class="content"></p>  
<p class="content"></p>
```



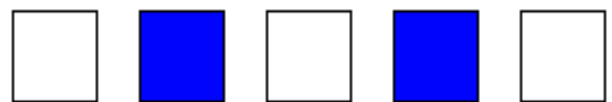
:nth-child



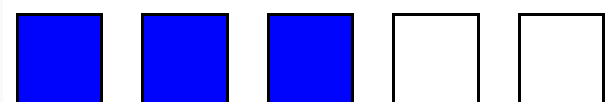
:nth-child(3)



:nth-child(odd)



:nth-child(even)



:nth-child(-n+2)



伪类选择器

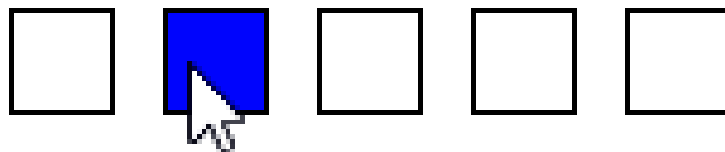
用于选择处于特定状态的元素，
效果等同于向文档某个部分
应用了一个类一样



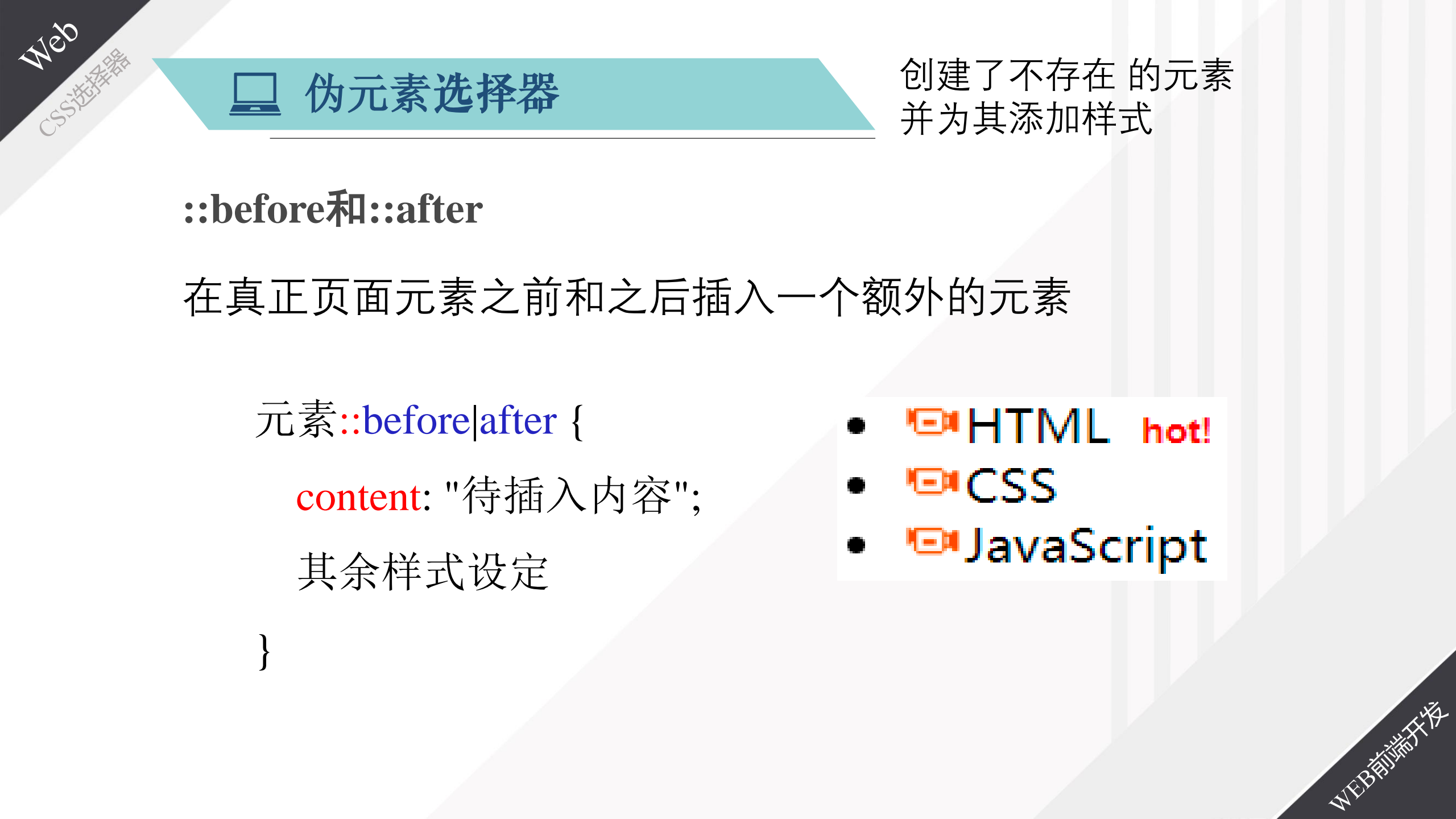
```
.content: hover {  
  background-color: blue;  
}
```



用户行为伪类: hover



```
<p class="content"></p>  
<p class="content"></p>  
<p class="content"></p>  
<p class="content"></p>  
<p class="content"></p>
```



伪元素选择器

创建了不存在的元素
并为其添加样式

::before和::after

在真正页面元素之前和之后插入一个额外的元素

```
元素::before|after {  
    content: "待插入内容";  
    其余样式设定  
}
```

-  HTML hot!
-  CSS
-  JavaScript



伪元素选择器

创建了不存在的元素
并为其添加样式



```
li::before {  
  content: url("video.png");  
}  
li:first-child::after {  
  content: "hot!";  
  color: red;  
  font: bolder 6px Arial;  
}
```



-  HTML hot!
-  CSS
-  JavaScript



```
<ul>  
  <li>HTML</li>  
  <li>CSS</li>  
  <li>JavaScript</li>  
</ul>
```



伪元素选择器

创建了不存在的元素
并为其添加样式

::first-letter和::first-line



```
p::first-letter {  
  font-size: 2em;  
  color: red;  
}  
  
p::first-line {  
  font-weight: bold;  
}
```



Lorem ipsum dolor sit amet consectetur
adipiscing elit. Fuga fugit, id eveniet ipsam
veniam hic placeat, explicabo nihil.

Lorem ipsum dolor sit amet consectetur
adipiscing elit. Fuga fugit, id eveniet ipsam
veniam hic placeat, explicabo nihil.



```
<p>      Lorem ipsum .....  
</p>  
<p>  
      Lorem ipsum.....  
</p>
```