



2.2

## 基础选择器



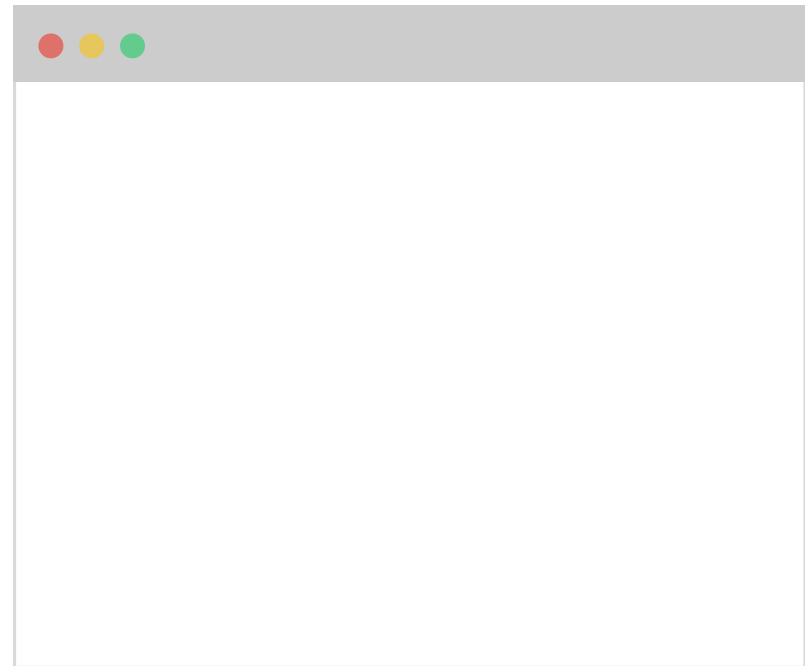
# 选择器 Selector

- 找出页面中的元素，以便给他们设置样式
- 使用多种方式选择元素
  - 按照标签名、类名或 id
  - 按照属性
  - 按照 DOM 树中的位置

# 通配选择器 \*

```
<h1>This is heading</h1>  
<p>this is some paragraph.</p>
```

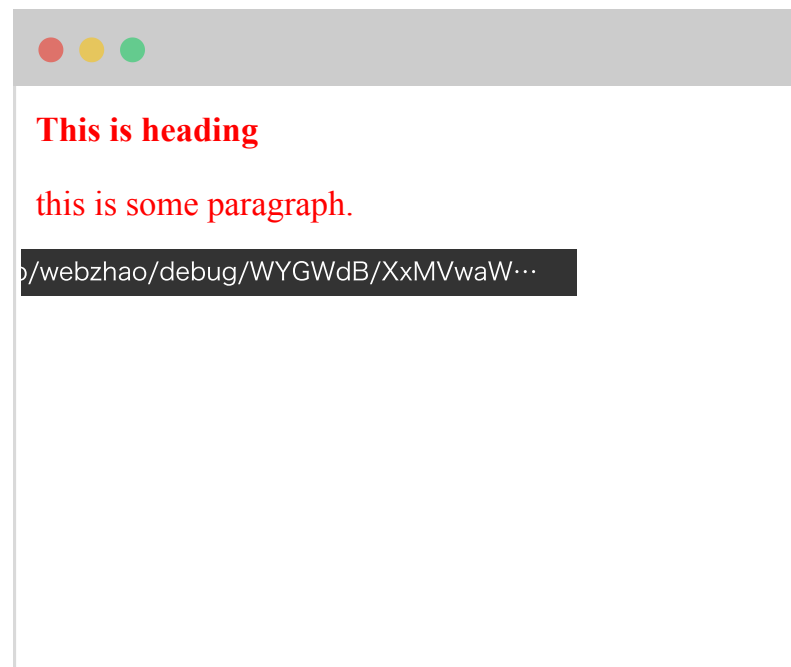
```
<style>  
* {  
  color: red;  
  font-size: 20px;  
}  
</style>
```



# 通配选择器 \*

```
<h1>This is heading</h1>  
<p>this is some paragraph.</p>
```

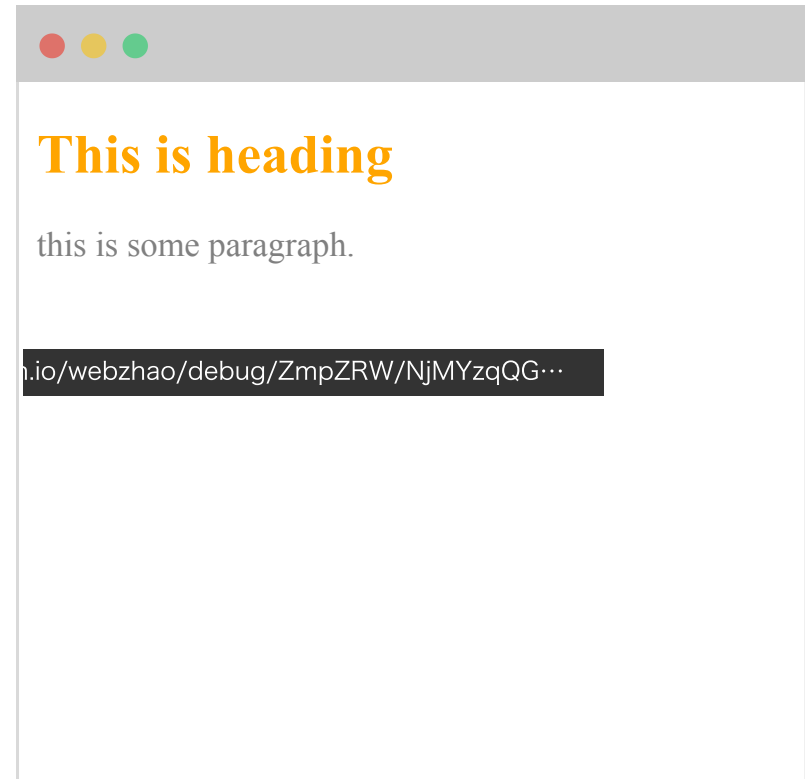
```
<style>  
* {  
  color: red;  
  font-size: 20px;  
}  
</style>
```



# 标签选择器

```
<h1>This is heading</h1>  
<p>this is some paragraph.</p>
```

```
<style>  
h1 {  
  color: orange;  
}  
  
p {  
  color: gray;  
  font-size: 20px;  
}  
</style>
```



# id 选择器

```
<header>
  <h1 id="logo">
    
      HTML5 文档
  </h1>
</header>

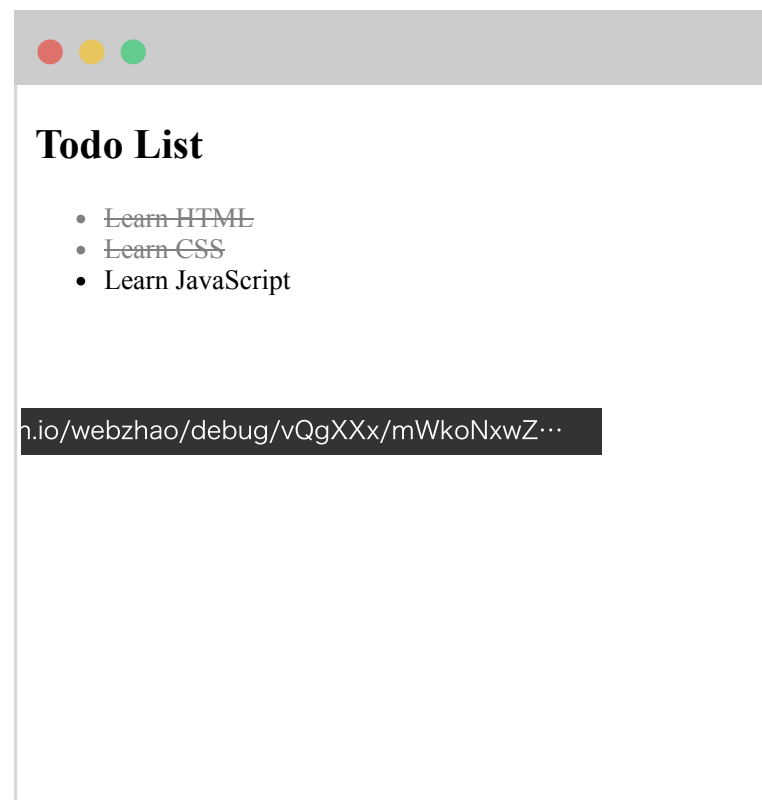
<style>
  #logo {
    font-size: 60px;
    font-weight: 200;
  }
</style>
```



# 类选择器

```
<h2>Todo List</h2>
<ul>
  <li class="done">Learn HTML</li>
  <li class="done">Learn CSS</li>
  <li>Learn JavaScript</li>
</ul>

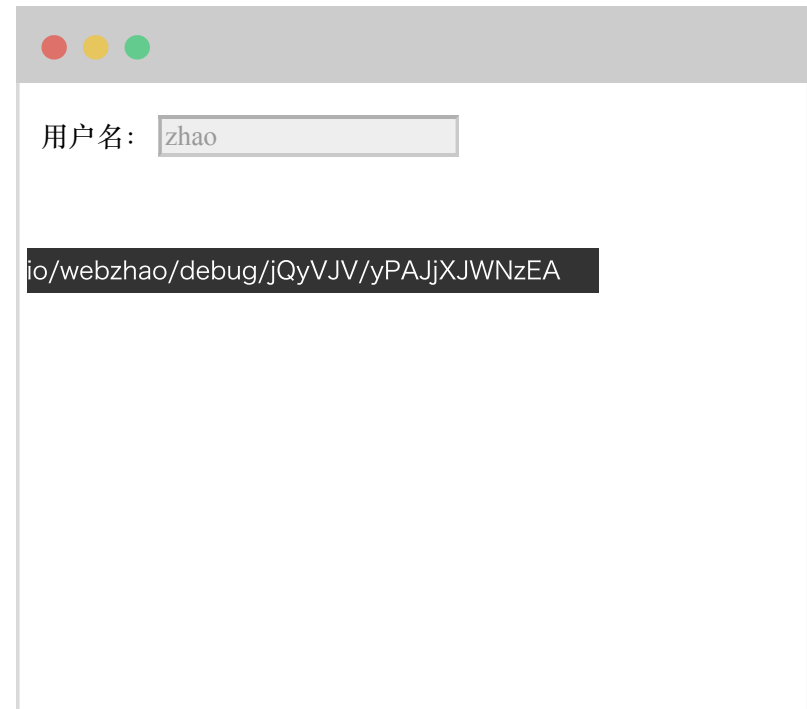
<style>
.done {
  color: gray;
  text-decoration: line-through;
}
</style>
```



# 属性选择器

```
<p>
  <label>用户名: </label>
  <input value="zhao" disabled>
</p>

<style>
  input[disabled] {
    background: #eee;
    color: #999;
  }
</style>
```





```
<p>
  <label>密码: </label>
  <input type="password" required>
</p>
```

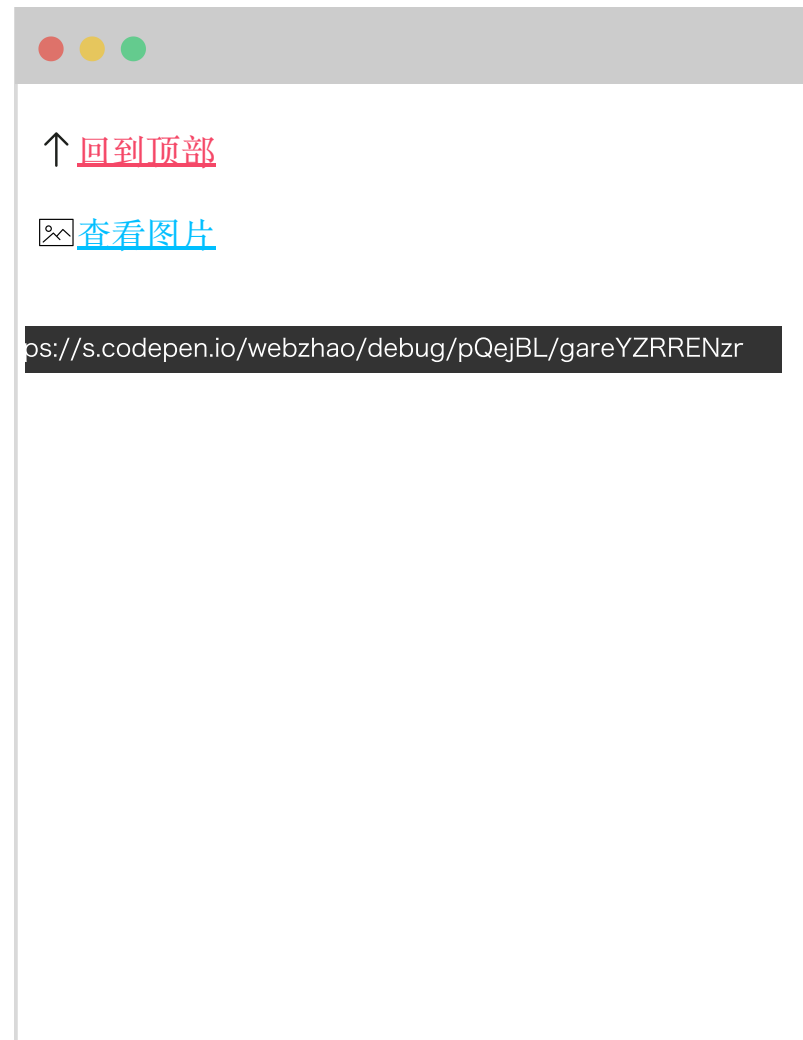
```
<style>
  input[type="password"] {
    border-color: red;
    color: red;
  }
</style>
```



```
<p><a href="#top">回到顶部</a></p>
<p><a href="a.jpg">查看图片</a></p>
```

```
<style>
  a[href^="#"] {
    color: #f54767;
    background: 0 center/1em
      url(top.svg) no-repeat;
    padding-left: 1.1em;
  }

  a[href$=".jpg"] {
    color: deepskyblue;
    background: 0 center/1em
      url(image.svg) no-repeat;
    padding-left: 1.1em;
  }
</style>
```



# 伪类 (pseudo-classes)

- 不基于标签和属性定位元素
- 几种伪类
  - 动态伪类
  - 结构性伪类

# 动态伪类

```
<p>
  <a href="http://example.com" target="_blank">
    example.com
  </a>
</p>

<label>
  用户名:
  <input type="text">
</label>

<style>
  a:link {
    color: black;
  }

  a:visited {
    color: gray;
  }
</style>
```

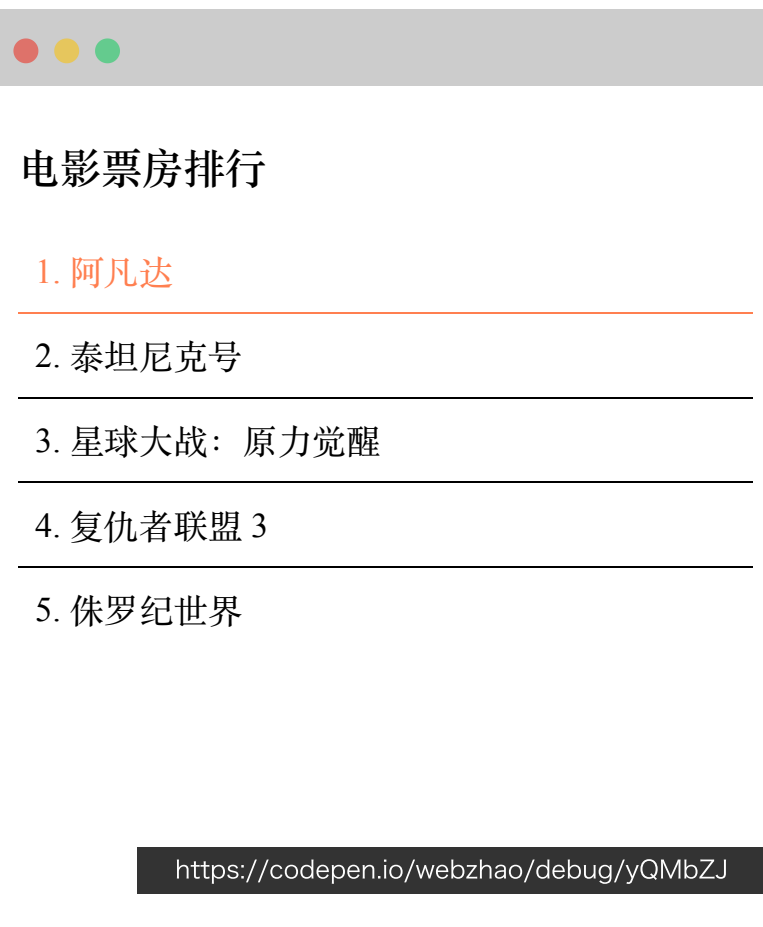


# 结构性伪类

```
<h2>电影票房排行</h2>
<ol>
  <li>阿凡达</li>
  <li>泰坦尼克号</li>
  <li>星球大战：原力觉醒</li>
  <li>复仇者联盟 3</li>
  <li>侏罗纪世界</li>
</ol>

<style>
  li {
    list-style-position: inside;
    border-bottom: 1px solid;
    padding: 0.5em
  }

  li:first-child {
    color: coral
  }
</style>
```



# 组合器 (Combinators)

```
<label>
  用户名:
  <input class="error" value="aa">
</label>
<span class="error">最少3个字符</span>
```

```
<style>
  .error {
    color: red;
  }

  input.error {
    border-color: red;
  }
</style>
```



# 组合器 (Combinators)

名称	语法	说明	示例
直接组合	$AB$	满足A同时满足B	<code>input:focus</code>
后代组合	$A \ B$	选中B, 如果它是A的子孙	<code>nav a</code>
亲子组合	$A > B$	选中B, 如果它是A的子元素	<code>section &gt; p</code>

```

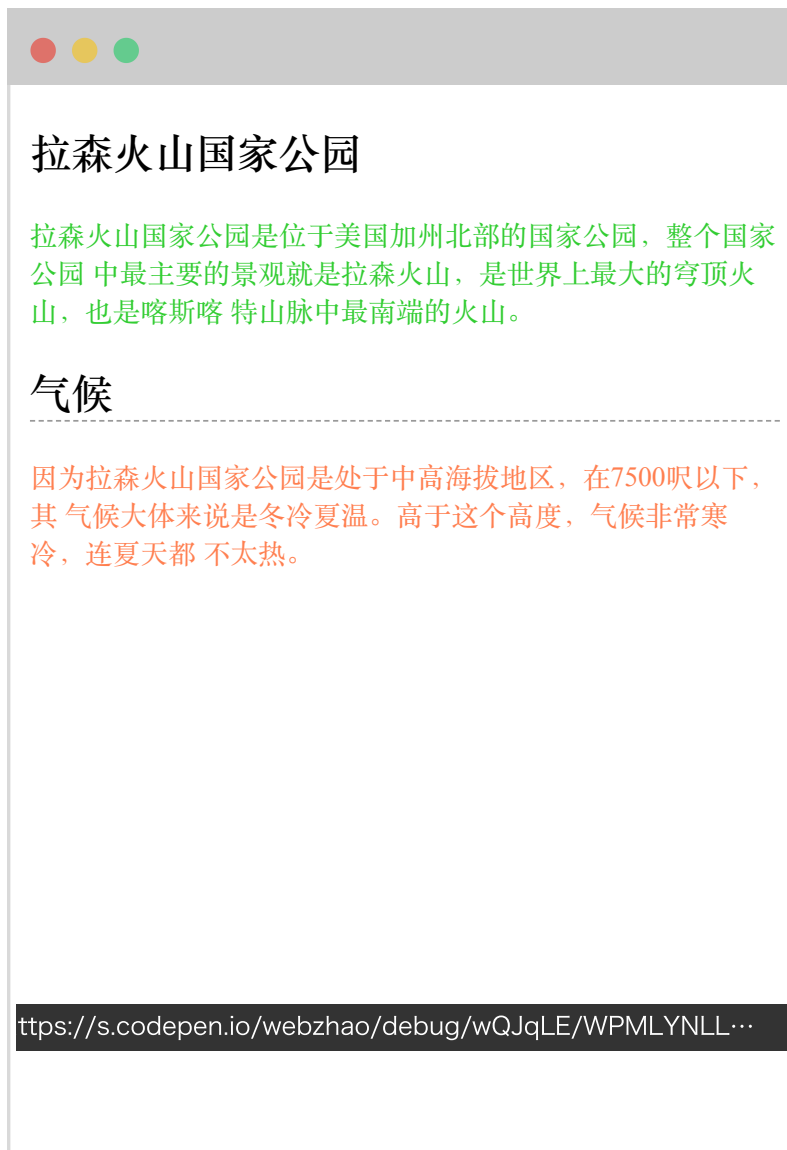
<article>
  <h1>拉森火山国家公园</h1>
  <p>拉森火山国家公园是位于 ... </p>
  <section>
    <h2>气候</h2>
    <p>因为拉森火山国家公园 ... </p>
  </section>
</article>

<style>
  article p {
    color: coral;
  }

  article > p {
    color: limegreen;
  }

  article section h2 {
    border-bottom: 1px dashed #999;
  }

```





# 选择器组

```
body, h1, h2, h3, h4, h5, h6, ul, ol, li {  
    margin: 0;  
    padding: 0;  
}
```

```
[type="checkbox"], [type="radio"] {  
    box-sizing: border-box;  
    padding: 0;  
}
```