

# Lecture 4

## CSS 基础



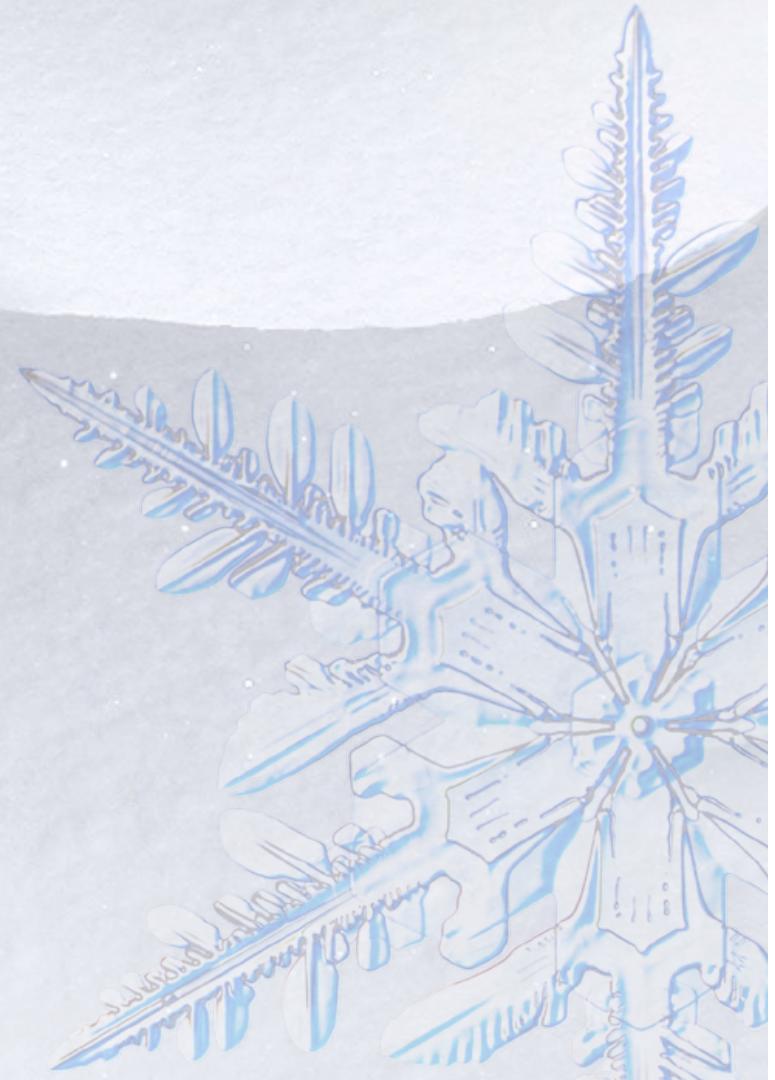
# 目录

什么是CSS?

语法

层叠排序

CSS3



# CSS

❄ Cascading Style Sheets, 层叠样式表

❄ CSS 用于设计风格和布局。

\* 比如，可以使用 CSS 来更改内容的字体、颜色、大小、间距，将内容分为多列，或者添加动画及其他装饰效果。

❄ Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. ( CSS - Wikipedia)

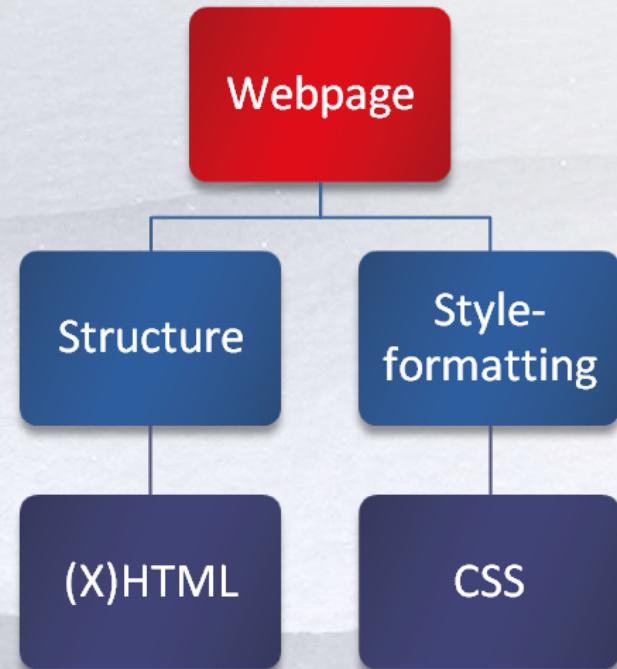
# CSS简史

- ❄ 1994年提出了CSS的最初建议
- ❄ 1996年12月W3C终于推出了CSS规范的第一版本
- ❄ 1998, W3C发布了CSS2
  - \* CSS 2.1
- ❄ CSS3

<https://www.w3.org/Style/LieBos2e/history/>

# CSS优势

- ✿ 丰富的样式定义
- ✿ 灵活：易于使用和修改
- ✿ 简洁、清晰
- ✿ 多页面应用
- ✿ 多设备兼容性：样式表允许针对多种不同类型的设备进行优化内容。
- ✿ 减少开发成本与维护成本
- ✿ 提高页面性能



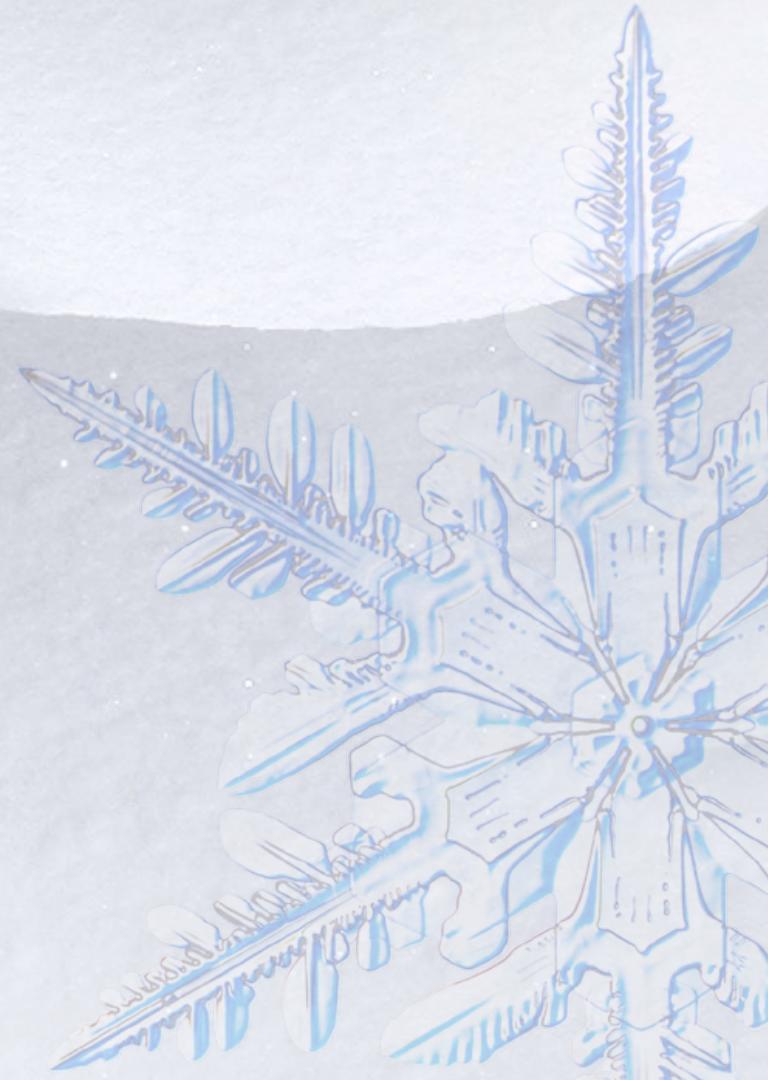
# 目录

什么是CSS?

语法

层叠排序

CSS3



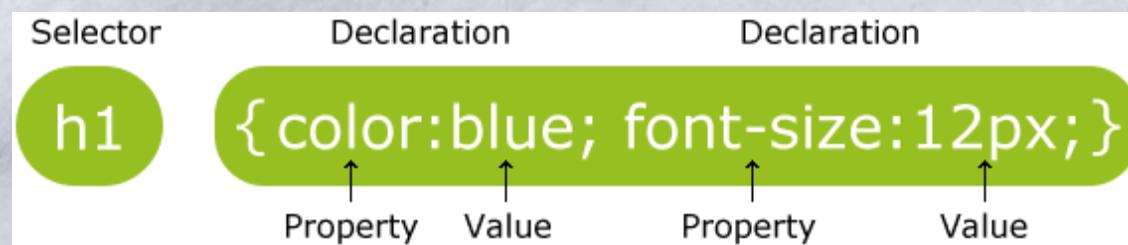
# CSS语法

## \_CSS语法组成

- \* 选择器

- \* 声明

- \* 属性/值



# 选择器

- ❄ id选择器
- ❄ class选择器
- ❄ 标签选择器
- ❄ 属性选择器
- ❄ 伪类选择器
- ❄ 伪元素选择器
- ❄ 组合选择器

# id选择器

❄ id选择器可以为标有特定 id 的 HTML 元素指定特定的样式。

- \* HTML元素以id属性来设置id选择器, CSS 中 id 选择器以 "#" 来定义。
- \* 下面的样式规则将应用于有属性id="para1"的元素:
  - \* `#para1 {text-align:center;color:red;}`

# 类选择器

- ✿ class 选择器用于描述一组元素的样式，class 选择器有别于id选择器，class可以在多个元素中使用
- ✿ class 选择器在 HTML 中以 class 属性表示, 在 CSS 中，类选择器以一个点 . 号显示
- ✿ 在以下的例子中，所有拥有 center 类的 HTML 元素均为居中。
  - \* .center {text-align:center;}

# 标签选择器

```
p {  
    color:red;  
    text-align:center;  
}
```

# 属性 选择器

```
[title]
{
    color:blue;
}

<h1 title="Hello world">Hello world</h1>
```

# 伪类选择器

 CSS伪类是用来添加一些选择器的特殊效果。

```
selector:pseudo-class {property:value;}  
selector.class:pseudo-class {property:value;}  
  
a:link {color:#000000;} /* 未访问链接 */  
a:visited {color:#00FF00;} /* 已访问链接 */  
a:hover {color:#FF00FF;} /* 鼠标移动到链接上 */  
a:active {color:#0000FF;} /* 鼠标点击时 */
```

# 伪元素选择器

```
selector:pseudo-element {property:value;}  
selector.class:pseudo-element {property:value;}
```

```
p:first-line  
{  
color:#ff0000;  
font-variant:small-caps;  
}
```

你可以使用 "FIRST-LINE" 伪元素向文本的首行设置特殊样式。

# CSS 组合选择符

- ❄ 后代选择器(以空格分隔)
- ❄ 子元素选择器(以大于号分隔)
- ❄ 相邻兄弟选择器 (以加号分隔)
- ❄ 普通兄弟选择器 (以破折号分隔)

# 后代选择符

```
div p  
{  
    background-color:yellow;  
}
```

```
<body>  
  
<div>  
<p>段落 1。在 div 中。</p>  
<p>段落 2。在 div 中。</p>  
</div>  
  
<p>段落 3。不在 div 中。</p>  
<p>段落 4。不在 div 中。</p>  
  
</body>
```

```
* { background-color: rgba(255,0,0,.2); }  
* * { background-color: rgba(0,255,0,.2); }  
* * * { background-color: rgba(0,0,255,.2); }  
* * * * { background-color: rgba(255,0,255,.2); }  
* * * * * { background-color: rgba(0,255,255,.2); }  
* * * * * * { background-color: rgba(255,255,0,.2); }
```

段落 1。在 div 中。

段落 2。在 div 中。

段落 3。不在 div 中。

段落 4。不在 div 中。

# 子元素选择器

```
div>p  
{  
    background-color:yellow;  
}
```

```
<body>  
<h1>Welcome to My Homepage</h1>  
<div>  
<h2>My name is leo</h2>  
<p>I live in Nanjing.</p>  
</div>  
  
<div>  
<span><p>I will not be styled.</p></span>  
</div>  
  
<p>My best friend is Tom.</p>  
</body>
```

Welcome to My Homepage

My name is leo

I live in Nanjing.

I will not be styled.

My best friend is Tom.

# 相邻兄弟选择器

```
div+p  
{  
    background-color:yellow;  
}
```

```
<body>  
<h1>Welcome to My Homepage</h1>  
  
<div>  
<h2>My name is Leo</h2>  
<p>I live in Nanjing.</p>  
</div>  
  
<p>My best friend is Tom.</p>  
<p>I will not be styled.</p>  
  
</body>
```

**Welcome to My Homepage**

**My name is Leo**

I live in Nanjing.

My best friend is Tom.

I will not be styled.

# 后续兄弟选择器

```
div~p  
{  
    background-color:yellow;  
}
```

```
<body>
```

```
<p>之前段落，不会添加背景颜色。</p>
```

```
<div>
```

```
<p>段落 1。在 div 中。</p>
```

```
<p>段落 2。在 div 中。</p>
```

```
</div>
```

```
<p>段落 3。不在 div 中。</p>
```

```
<p>段落 4。不在 div 中。</p>
```

```
</body>
```

之前段落，不会添加背景颜色。

段落 1。在 div 中。

段落 2。在 div 中。

段落 3。不在 div 中。

段落 4。不在 div 中。

# 注释

❄ /\* ..... \*/

❄ 例如：

\* /\* This is a basic comment it will not appear on the page \*/  
p{ text-align:center; color:black; font-family:arial;}

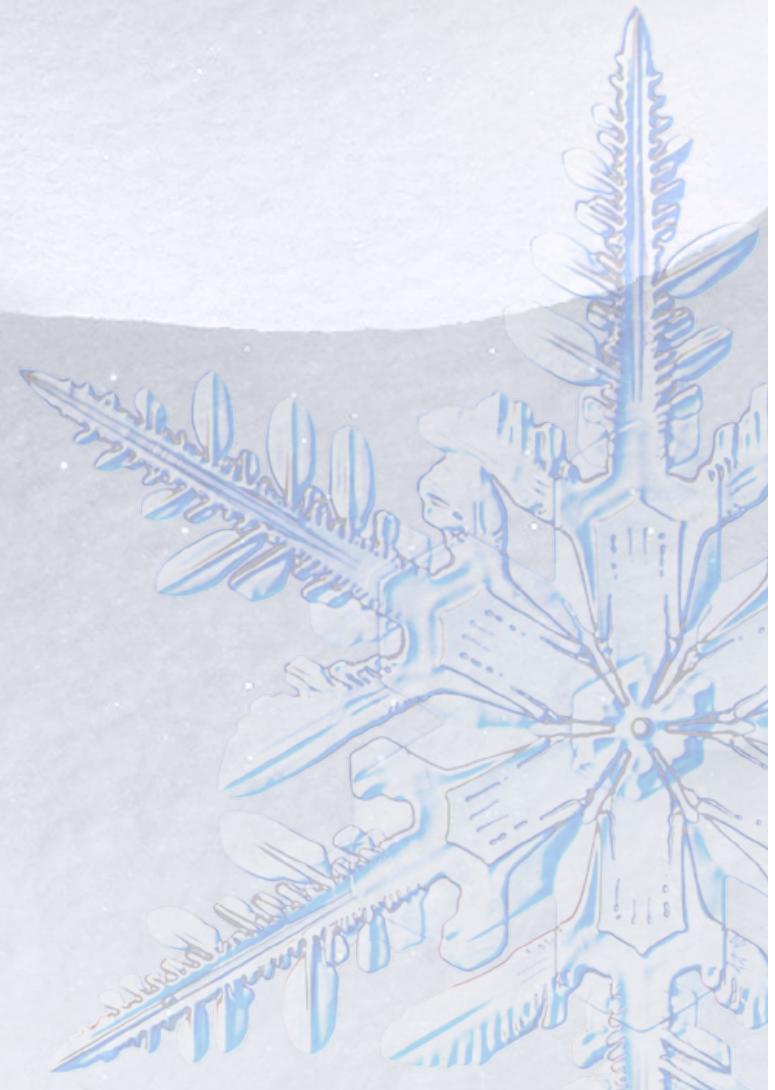
# 目录

什么是CSS?

语法

层叠排序

CSS3



# 层叠样式表

\* 用来解决样式说明冲突的整个排序过程称为层叠。

- \* 浏览器缺省样式
- \* 内联样式
- \* 嵌入样式表
- \* 外部样式表

# 外部样式表

\* 链接方式：指的是使用 HTML 头部的 `<link>` 标签引入外部的 CSS 文件。

- \* `<head>`
- \* `<link rel="stylesheet" type="text/css" href="style.css">`
- \* `</head>`

\* 导入方式：指的是使用 CSS 规则引入外部 CSS 文件。

- \* `<style>`
- \* `@import url(style.css);`
- \* `</style>`

## 比较

- \* `link` 属于 HTML，通过 `<link>` 标签中的 `href` 属性来引入外部文件，而 `@import` 属于 CSS，所以导入语句应写在 CSS 中，要注意的是导入语句应写在样式表的开头，否则无法正确导入外部文件；
- \* `@import` 是 CSS2.1 才出现的概念，所以如果浏览器版本较低，无法正确导入外部样式文件；
- \* 当 HTML 文件被加载时，`link` 引用的文件会同时被加载，而 `@import` 引用的文件则会等页面全部下载完毕再被加载；
- \* 使用 DOM 控制样式时的差别。

# 嵌入的样式表

❄ 嵌入方式指的是在 HTML 头部中的 `<style>` 标签下书写 CSS 代码。

❄ 例如：

```
<head>
  <title>my page</title>
  <style type="text/css">
    p{color:red}</style>
  </head>
<body>
  <p>this is a simple paragraph
  </p>
</body>
```

# 内联样式

※ 内联方式指的是直接在 HTML 标签中的 style 属性中添加 CSS

※ 例如：

- \* `<p style="color: red;"> This is a simple paragraph and the inline style makes it red.</p>`

# CSS如何用于网页

- ✿ Web开发人员很少使用内联CSS。因为不喜欢把内容和呈现混合在一起，而且它效率不高，因为必须为每个组件单独声明样式。
- ✿ 嵌入和外部样式表更受欢迎，因为可以用一条规则设置多个元素的样式。
- ✿ 外部样式表是最好的，因为允许将所有样式信息保存在一个单独的文件中。然后，可以修改网站的样式，它将更新网站中的所有页面。

# 样式冲突

✿ 嵌入的样式表对于h3选择器具有以下属性：

```
h3 { text-align: right;  
     font-size: 20pt }
```

✿ 外部样式表有如下属性：

```
h3 { color: red;  
     text-align: left;  
     font-size: 8pt }
```

✿ 浏览器缺省设置为：

```
h3 { color: black;  
     font size: 10pt }
```

► <h3> 的最终样式将如何？

- color: red;
- text-align: right;
- font-size: 20pt

# 继承

<p>CSS: <em>Inheriting Styles</em> Demo</p>

Html

```
<style>
p { color:red; }
</style>
```

css

*CSS: Inheriting Styles Demo*

```
<style>
p { color:red; border: 1px solid red; }
</style>
```

css

*CSS: Inheriting Styles Demo*

✿当元素的一个继承属性 (inherited property) 没有指定值时，则取父元素的同属性的计算值

✿不是所有的属性都被继承(注意上面link的颜色)

# CSS 声明的源

## ❄ CSS 声明可以有不同的来源：

- \* 用户代理：浏览器会有一个基本的样式表来给任何网页设置默认样式。这些样式统称用户代理样式。一些浏览器通过使用真正的样式表，而其他则通过代码模拟，但无论是哪种情形都应是不可被检测的。而且部分浏览器允许用户修改用户代理样式。尽管 HTML 标准对用户代理样式做了诸多限制，浏览器仍大有可为，具体表现在不同浏览器间会存在重大的差异。为了减轻开发成本以及降低样式表运行所需的基本环境，网页开发者通常会使用一个 CSS reset 样式表，强制将常见的属性值转为确定状态。
- \* 网页作者，可以定义文档的样式，这是最常见的样式表。大多数情况下此类型样式表会定义多个，它们构成网站的视觉和体验，即主题。
- \* 用户，作为浏览器的用户，可以使用自定义样式表定制使用体验。
- \*

# 层叠顺序

 层叠算法决定如何找出要应用到每个文档元素的每个属性上的值。

- \* 首先过滤来自不同源的全部规则，并保留要应用到指定元素上的那些规则。这意味着这些规则的选择器匹配指定元素，同时也是一个合适的 @规则 (at-rule) 的一部分。
- \* 其次，它依据重要性对这些规则进行排序。即是指，规则后面是否跟随着!important 以及规则的来源。层叠是按升序排列的，这意味着来自用户自定义样式表的 !important 值比用户代理样式表的普通值优先级高：
- \* 假如层叠顺序相等，则使用哪个值取决于优先级。

# 根据来源和重要程度排序

\* 相互冲突的声明将按以下顺序应用，后一种声明将覆盖前一种声明：

- \* 用户代理样式表中的声明（例如，浏览器的默认样式，在没有设置其他样式时使用）。
- \* 用户样式表中的常规声明（由用户设置的自定义样式）。
- \* 作者样式表中的常规声明（这些是我们 web 开发人员设置的样式）。
- \* 作者样式表中的 !important 声明
- \* 用户样式表中的 !important 声明
- \* 用户代理样式表中的 !important 声明

# 优先级 (specificity)

---

- ✿ 优先级就是分配给指定的 CSS 声明的一个权重，它由匹配的选择器中的 每一种选择器类型的 数值 决定。
- ✿ 而当优先级与多个 CSS 声明中任意一个声明的优先级相等的时候，CSS 中最后的那个声明将会被应用到元素上。
- ✿ 当同一个元素有多个声明的时候，优先级才会有意义
- ✿ 给元素添加的内联样式（例如，`style="font-weight:bold"`）总会覆盖外部样式表的任何样式，因此可看作是具有最高的优先级。

# 优先级的计算

- ✿ 如果声明来自一个'style'属性而不是一条选择器样式规则， 算1， 否则就是0 (= a) (HTML中， 一个元素的 "style" 属性值是样式表规则， 这些属性没有选择器， 所以a=1, b=0, c=0, d=0)
- ✿ 计算选择器中ID属性的数量 (= b)
- ✿ 计算选择器中其它属性和伪类的数量 (= c)
- ✿ 计算选择器中元素名和伪元素的数量 (= d)
- ✿ 4个数连起来a-b-c-d表示优先级

# 例子1

```
p.message {color: green;}
```

```
body#home div#warning p.message {color: red;}
```

	a	b	c	d
1:	0	0	1	1
2:	0	2	1	3

# !important例子

```
<div class="foo" style="color: red;">What color am I?</div>
```

```
.foo[style*="color: red"] {  
    color: firebrick !important;  
}
```

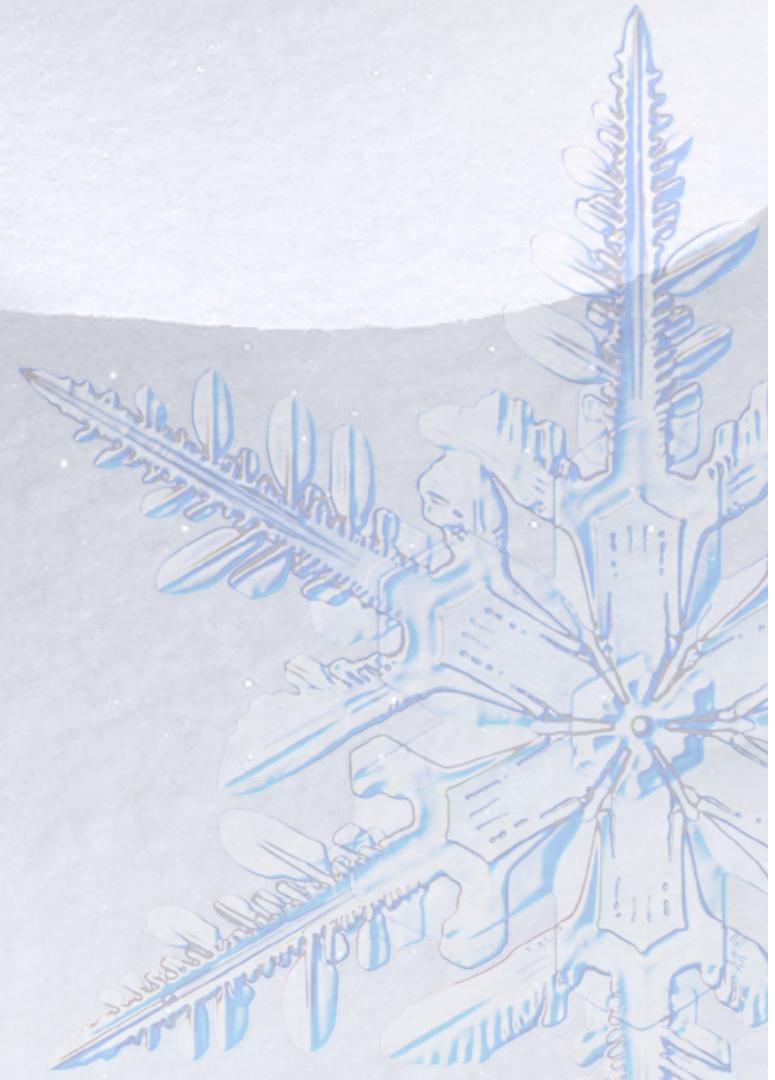
# 目录

什么是CSS?

语法

层叠排序

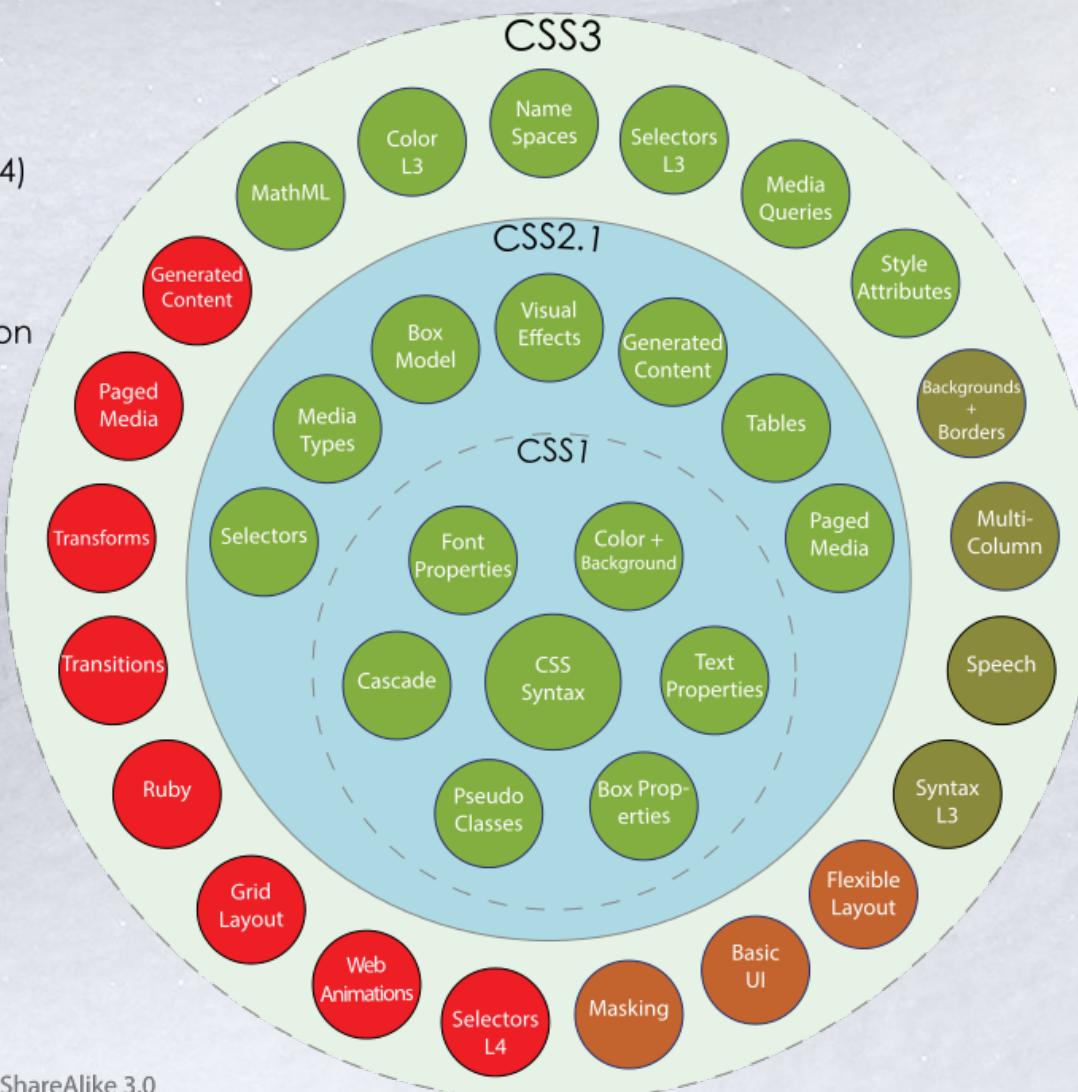
CSS3



# CSS3

Taxonomy & Status (October 2014)

- W3C Recommendation
- Candidate Recommendation
- Last Call
- Working Draft
- Obsolete or inactive



# CSS3



CSS3规范的一个新特点是被分为若干个相互独立的模块。

- \* 一方面分成若干较小的模块较利于规范及时更新和发布，及时调整模块的内容，这些模块独立实现和发布，也为日后CSS的扩展奠定了基础。
- \* 另外一方面，由于受支持设备和浏览器厂商的限制，设备或者厂商可以有选择的支持一部分模块，支持CSS3的一个子集，这样有利于CSS3的推广。

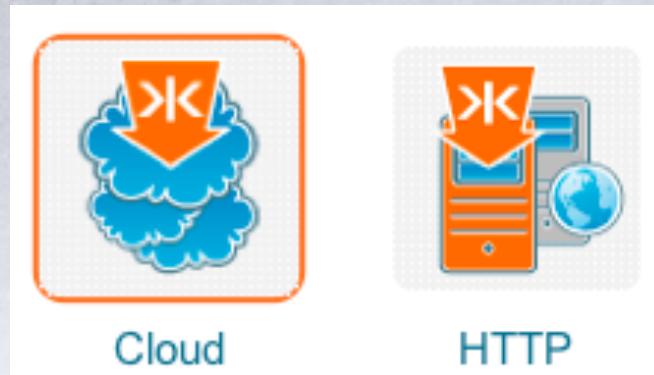
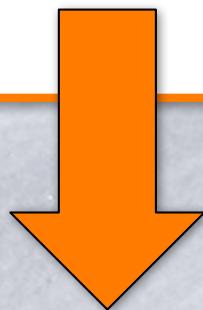
# CSS3的新特征

- ✿ 很多：圆角效果、图形化边界、块阴影与文字阴影、使用RGBA实现透明效果、渐变效果、使用@Font-Face实现定制字体、多背景图、文字或图像的变形处理（旋转、缩放、倾斜、移动）、多栏布局、媒体查询等。

# CSS圆角

CSS

```
a:hover img {  
    border-radius: 10px;  
    border: 2px solid #F47D31;  
    -webkit-transform: scale(1.05);  
}
```



HTML4  
css Sprites

# CSS Sprites图像精灵

✿ 图像精灵是单个图像中包含的图像集合。

✿ 优点：

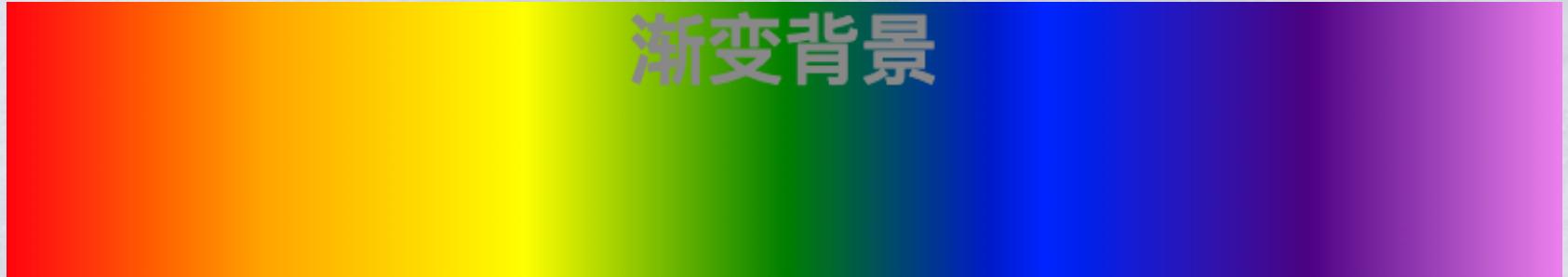
- \* 包含许多图像的网页可能需要很长时间才能加载，同时会生成多个服务器请求。
- \* 使用图像精灵将减少服务器请求的数量并节约带宽。
- \* 解决了网页设计师在图片命名上的困扰
- \* 更换风格方便

✿ 缺点：

- \* 维护麻烦
- \* 图片合并费时费力

# CSS 演变

```
#grad {  
background-image: linear-gradient(to right, red,orange,yellow,green,blue,indigo,violet);  
}
```



渐变背景

# 网络字体

HTML

```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="http://
fonts.googleapis.com/css?family=Tangerine">
  <style>
    h1 {
      font-family: 'Tangerine', serif;
      font-size: 48px;
    }
  </style>
</head>
<body><h1>Making the Web Beautiful!</h1></body>
```



*Making the Web Beautiful!*

[http://code.google.com/apis/webfonts/docs/getting\\_started.html#Quick\\_Start](http://code.google.com/apis/webfonts/docs/getting_started.html#Quick_Start)

# 网上资源

\* <http://www.w3schools.com/css/>, <https://www.w3school.com.cn/>

- \* Learn CSS

\* <http://validator.w3.org/>

- \* Check Your CSS syntax

\* <http://www.csszengarden.com/>

- \* The beauty of CSS Design
- \* One HTML file
- \* 210 CSS

# 阅读资料

- ❄ [http://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets)
- ❄ CSS权威指南
- ❄ 精通CSS：高级Web标准解决方案
- ❄ W3 CSS: <http://www.w3.org/TR/CSS/>
- ❄ <http://www.w3.org/TR/CSS21/cascade.html#specificity>
- ❄ <https://www.w3.org/TR/css3-cascade/>
  
- ❄ <https://codepen.io/pen/>
- ❄ <https://jsfiddle.net/api/mdn/>

# Thanks!!!

