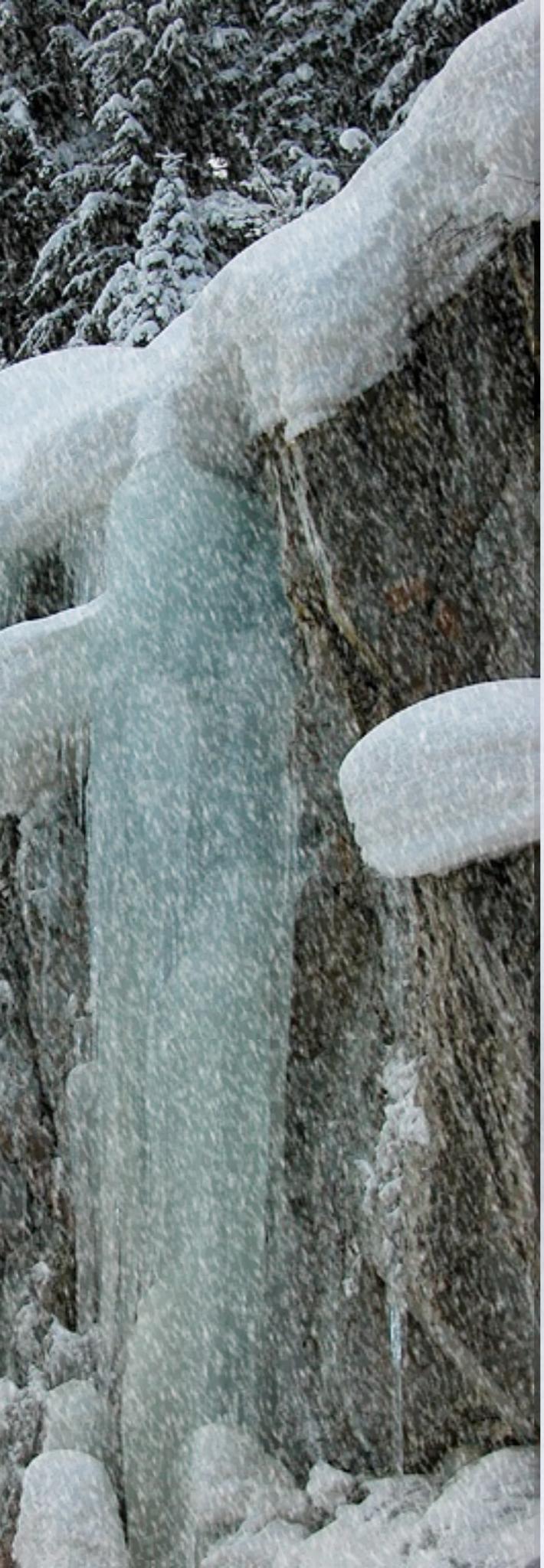


Lecture 4

CSS Basic



Play



Outline

- ❖ **What is CSS? Why CSS?**
 - ❖ **Syntax**
 - ❖ **Cascading Order**
 - ❖ **CSS3**
- 

What is CSS?

* **CSS stands for Cascading Style Sheets**

- * Styles define how to display (X)HTML elements
- * Styles are normally stored in Style Sheets
- * Multiple style definitions will cascade into one

* **Definition:** Cascading Style Sheets (CSS) is a rule based language that applies styling to your HTML elements. You write CSS rules in elements, and modify properties of those elements such as color, background color, width, border thickness, font size, etc.

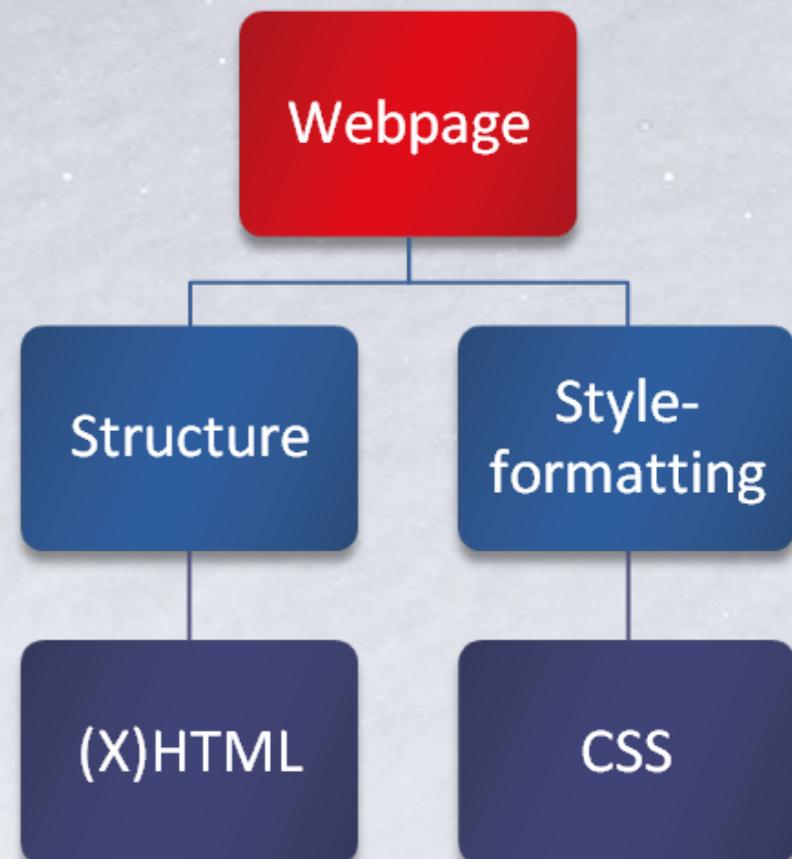
A Brief History of CSS

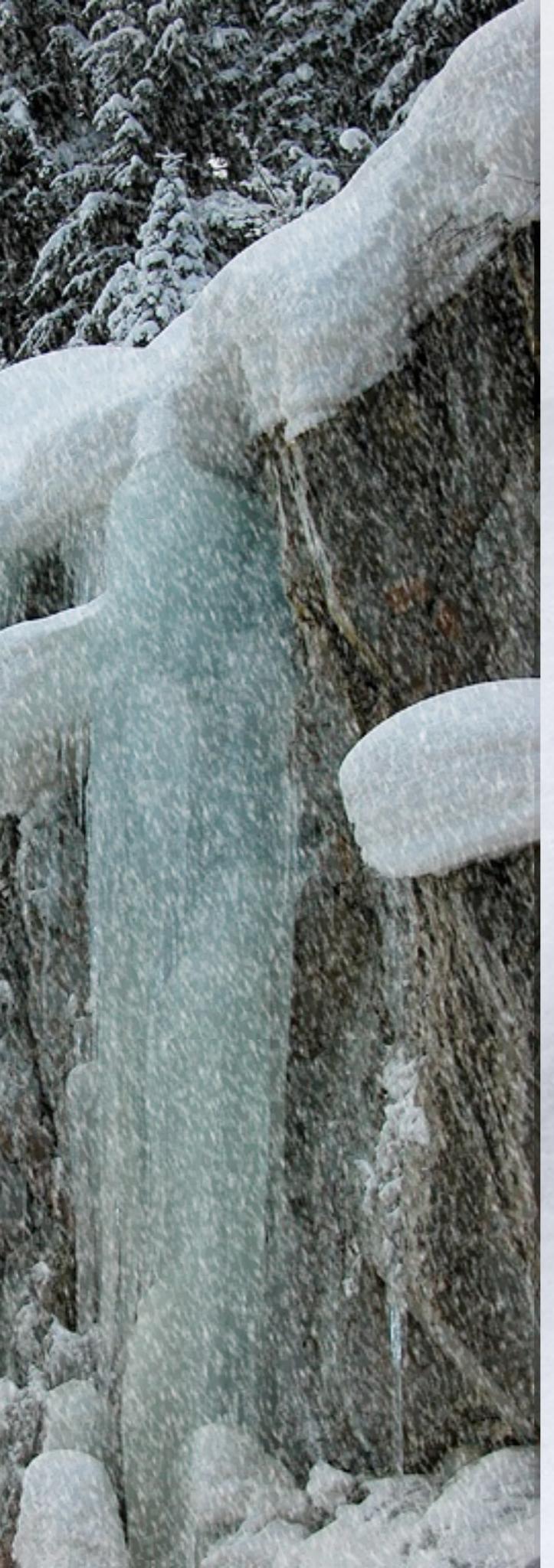
- ❖ CSS was proposed in 1994 as a web styling language. To helps solve some of the problems HTML 4.
- ❖ The first CSS specification to become an official W3C Recommendation is CSS level 1, published in December 1996. The W3C maintains the CSS1 Recommendation.
- ❖ CSS2 was published as a Recommendation in May 1998. The W3C maintains the CSS2 Recommendation.
 - * includes a number of new capabilities
 - * absolute, relative, and fixed positioning of elements
 - * the concept of media types
 - * new font properties such as shadows
- ❖ CSS level 2 revision 1 or CSS 2.1 fixes errors in CSS2
 - * returned to Candidate Recommendation status on 19 July 2007
- ❖ CSS3 was started in 1998 but it has never been completed. Some parts are still being developed and some components work on some browsers.

Why to use Styles?

★ Documents written with CSS are

- * more flexible
- * short
- * clear
- * Basic formatting tool
- * Easy multiple document management
- * Save time by using selector classes
- * New opportunities in formatting





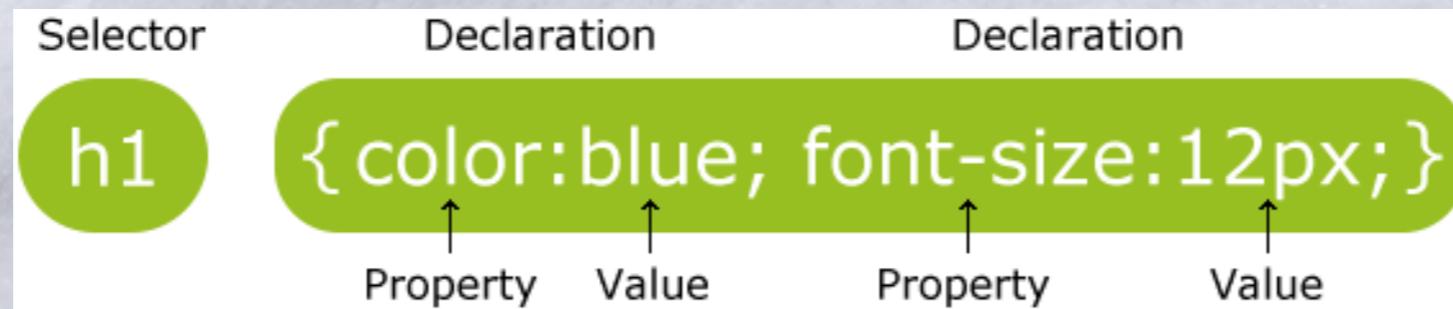
Outline

- * What is CSS? Why CSS?
 - * Syntax
 - * Cascading Order
 - * CSS3
- 

Syntax of CSS

* The CSS syntax is made up of 5 parts:

- * selector
- * property/value
- * declaration
- * declaration block
- * curly braces



Selectors

- ❄ 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 Selectors

- ❖ **id** - The id selector is used to specify a style for a single, unique element.
 - * The id selector uses the id attribute of the HTML element, and is defined with a "#".
 - * The style rule below will be applied to the element with id="para1":
 - * **#para1 {text-align:center;color:red;}**

Class Selectors

 **Class** - The class selector is used to specify a style for a group of elements. Unlike the id selector, the class selector is most often used on several elements.

- * This allows you to set a particular style for any HTML elements with the same class.
- * The class selector uses the HTML class attribute, and is defined with a "."
- * In the example below, all HTML elements with class="center" will be center-aligned:
 - * `.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

```
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。

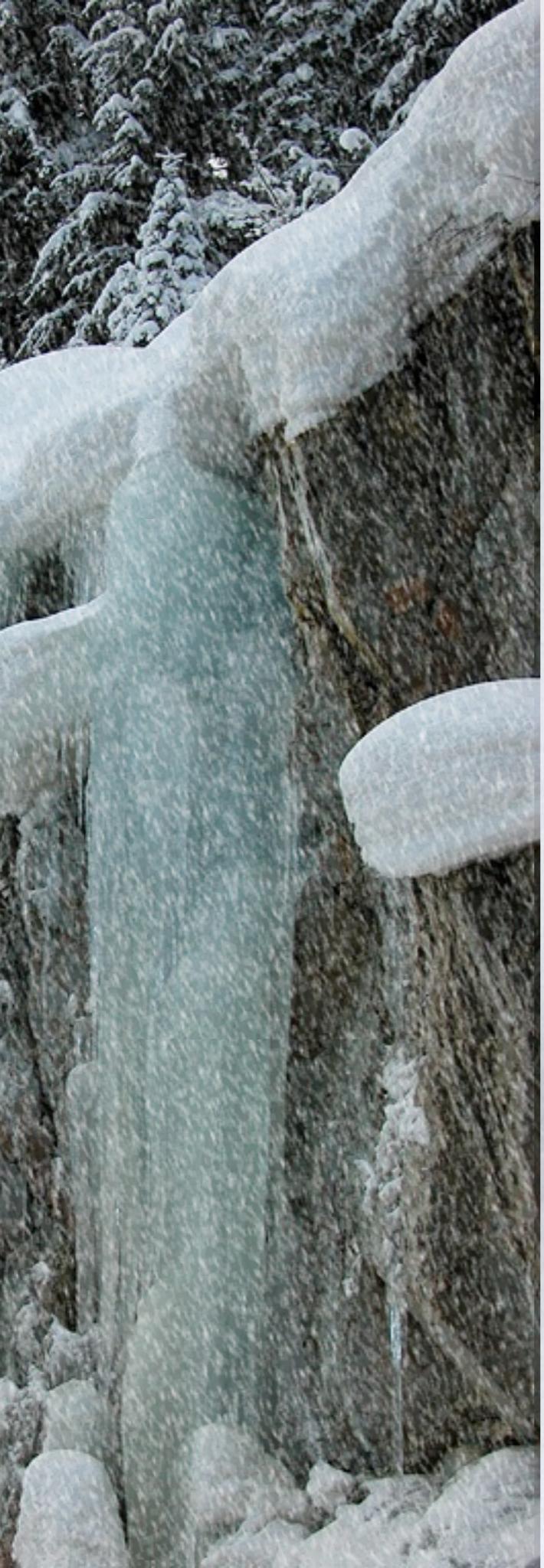
段落 2。在 div 中。

段落 3。不在 div 中。

段落 4。不在 div 中。

Comments

- ❖ Comments are used to explain your code, and may help you when you edit the source code at a later date. Comments are ignored by browsers.
 - ❖ You add comments by enclosing them in /* and */
 - ❖ Comments can span several lines, and the browser will ignore these lines.
- ❖ Example:
- * /* This is a basic comment it will not appear on the page*/
p{ text-align:center; color:black; font-family:arial;}



Outline

- ❖ What is CSS? Why CSS?
 - ❖ Syntax
 - ❖ Cascading Order
 - ❖ CSS3
- 

Cascading style sheets

* it's called Cascading Style Sheets because the properties of an element **cascade** together in this order:

- * browser's default styles
- * external style sheet files
- * internal style sheets (inside a `<style>` tag in the page's header)
- * inline style (the `style` attribute of the HTML element)

External Style Sheet

 Applies styles as a separate file with a .css extension. The file is then referenced from inside the <head> element by a link to the file.

- * <link> tag &@import

 For Example:

```
<head>
<title>my external style sheet page</title>
<link rel="style sheet" type="text/css" href="my-external-stylesheet.css">
</head>
<body>
<p>this is a simple paragraph</p>
</body>
```

 You can create an external style sheet in your text editor.

Internal Style Sheet

❖ Applies styles to HTML by placing the CSS rules inside the tag `<style>` inside the document tag `<head>`.

❖ For Example:

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

Inline CSS

* Applies styles directly to the elements by adding declarations into the style

* For Example:

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

How CSS is Applied to A Web Page

- ✿ Web developers rarely use inline CSS. Since they prefer to not mix content with presentation. And it is not efficient since you have to declare the style individually for every component.
- ✿ Internal and External style sheets are more popular because you can style multiple elements with one rule.
- ✿ External style sheets are best because they allow you to save all the style information on a separate file from the content. You can then modify a style for a site and it will update all of the pages in a site.

Multiple Style Sheets

- ❖ An internal style sheet has following properties for the h3 selector:

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

► What will be the format of <h3> tag?

- ❖ External style sheet has these:

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

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

- ❖ Your Web Browser has default formatting:

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

Inheriting styles

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

Html

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

css

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

css

CSS: Inheriting Styles Demo

CSS: Inheriting Styles Demo

- * when multiple styles apply to an element, they are inherited
- * a more tightly matching rule can override a more general inherited rule
- * not all properties are inherited (notice link's color above)

Cascading order

- * Find all declarations that apply to the element and property in question, for the target media type. Declarations apply if the associated selector matches the element in question and the target medium matches the media list on all @media rules containing the declaration and on all links on the path through which the style sheet was reached.
- * Sort according to importance (normal or important) and origin (author, user, or user agent). In ascending order of precedence:
 - * user agent declarations
 - * user normal declarations
 - * author normal declarations
 - * author important declarations
 - * user important declarations
- * Sort rules with the same importance and origin by specificity of selector: more specific selectors will override more general ones. Pseudo-elements and pseudo-classes are counted as normal elements and classes, respectively.
- * Finally, sort by order specified: if two declarations have the same weight, origin and specificity, the latter specified wins. Declarations in imported style sheets are considered to be before any declarations in the style sheet itself.

Calculating a selector's specificity

- ❖ count 1 if the declaration is from a 'style' attribute rather than a rule with a selector, 0 otherwise (= a) (In HTML, values of an element's "style" attribute are style sheet rules. These rules have no selectors, so $a=1$, $b=0$, $c=0$, and $d=0$.)
- ❖ count the number of ID attributes in the selector (= b)
- ❖ count the number of other attributes and pseudo-classes in the selector (= c)
- ❖ count the number of element names and pseudo-elements in the selector (= d)

example

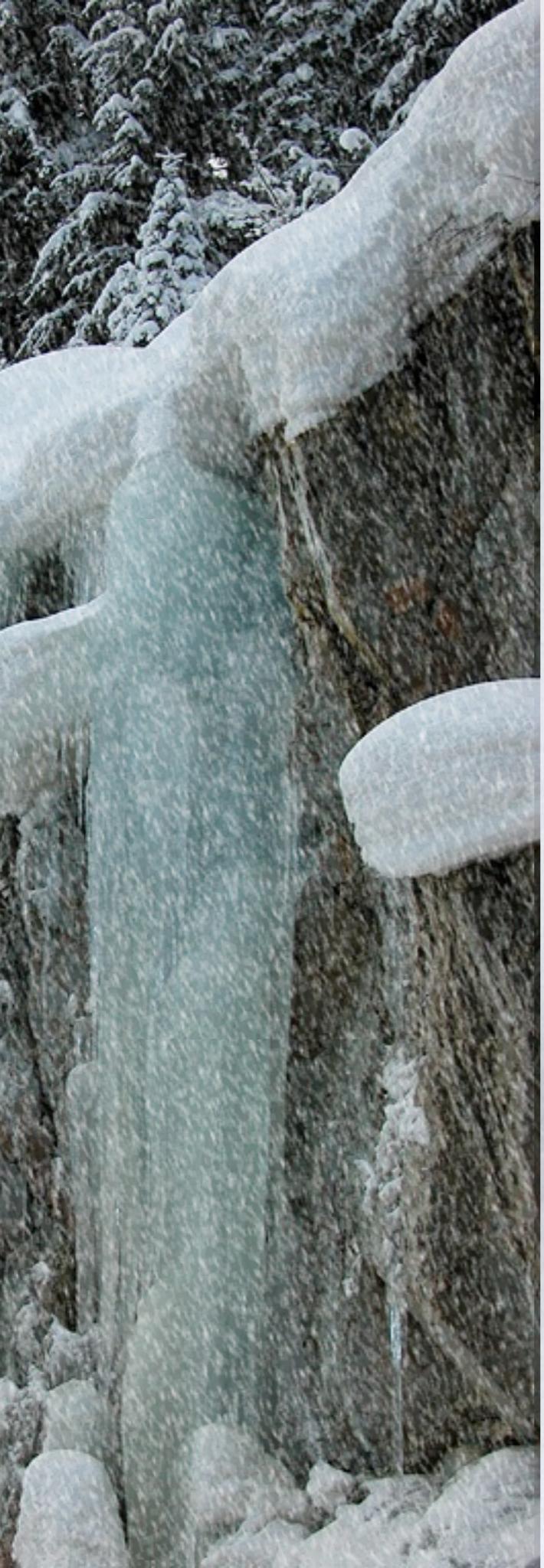
```
p.message {color: green;}
```

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

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

example 2

```
#example {  
    font-size: 14px !important;  
}  
  
#container #example {  
    font-size: 10px;  
}
```



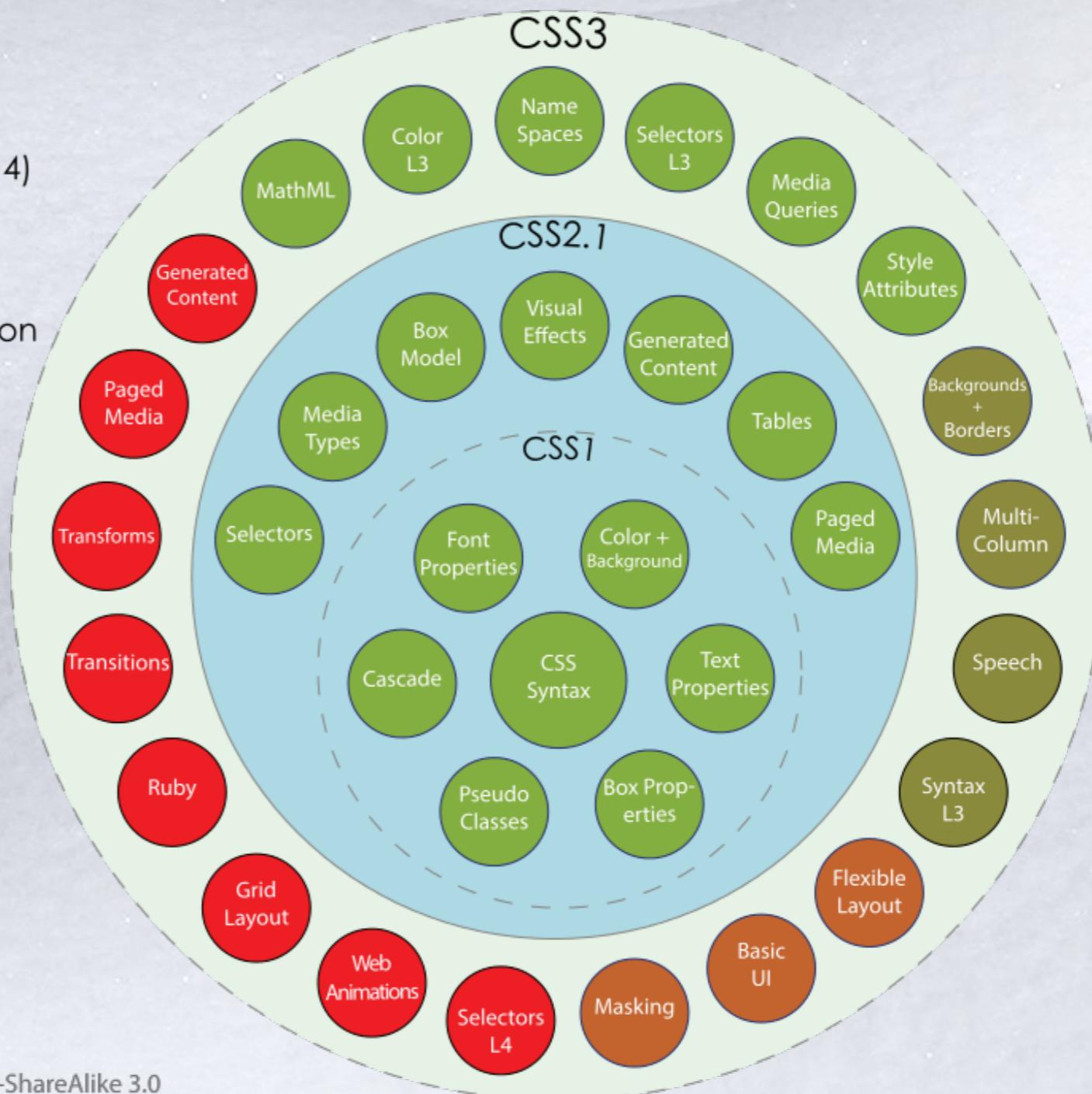
Outline

- ❖ What is CSS? Why CSS?
 - ❖ Syntax
 - ❖ Cascading Order
 - ❖ **CSS3**
- 

CSS3

Taxonomy & Status (October 2014)

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



CSS Level 3

❄ Modularized for easier browser uptake

❄ Almost 50 modules (readiness varies)

❄ Use browser-specific prefixes until finalized

❄ Dramatically improves performance

❄ Examples:

- * Border radius (rounded corners) without images

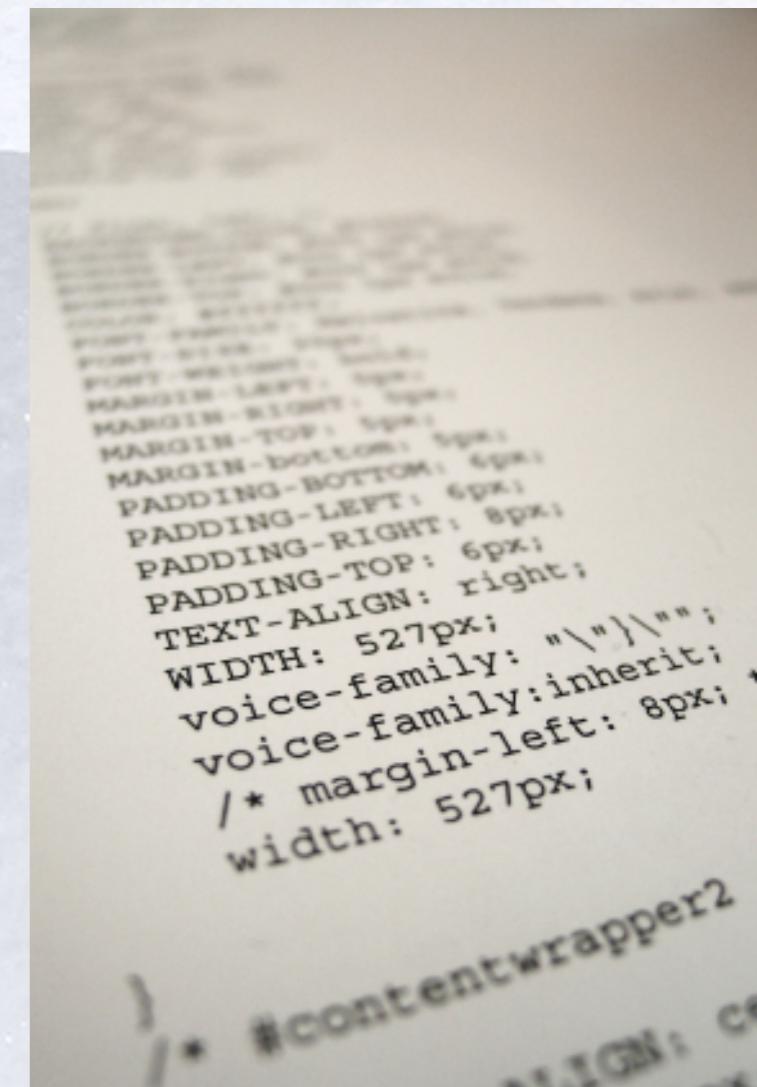
- * Gradients

- * Multi-column layout

- * Transformations and transitions

- * Web Fonts

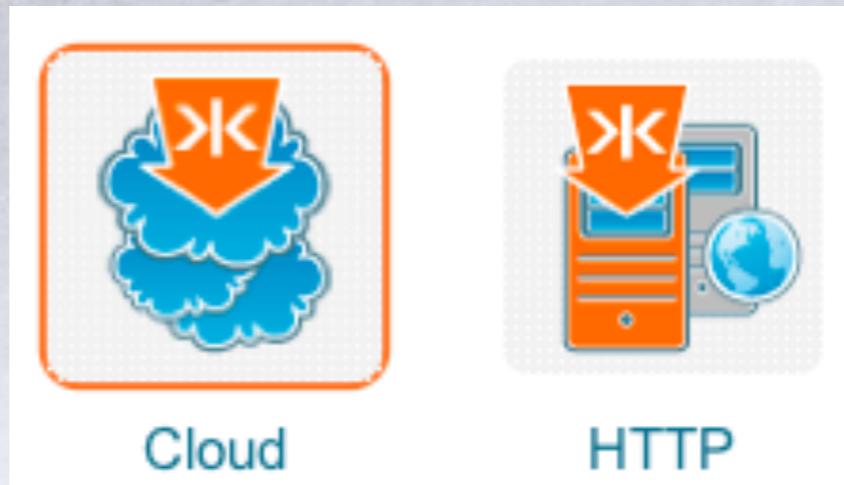
- * Media Queries



Rounded corners

CSS

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



HTML4
css Sprites

Web Fonts

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

Useful links

 <http://www.w3schools.com/css/>

- * Learn CSS

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

- * Check Your CSS syntax

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

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

Reading materials

- ❄ 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!!!

