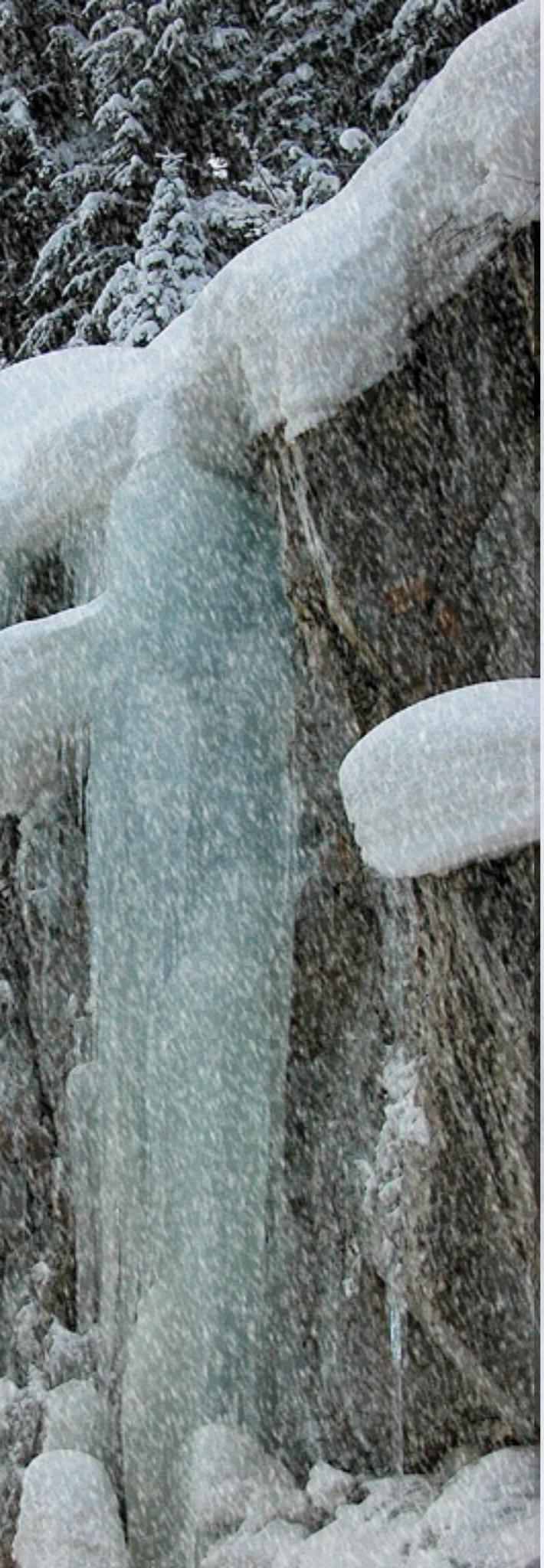


# Lecture 3

## CSS





# Outline

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

# 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

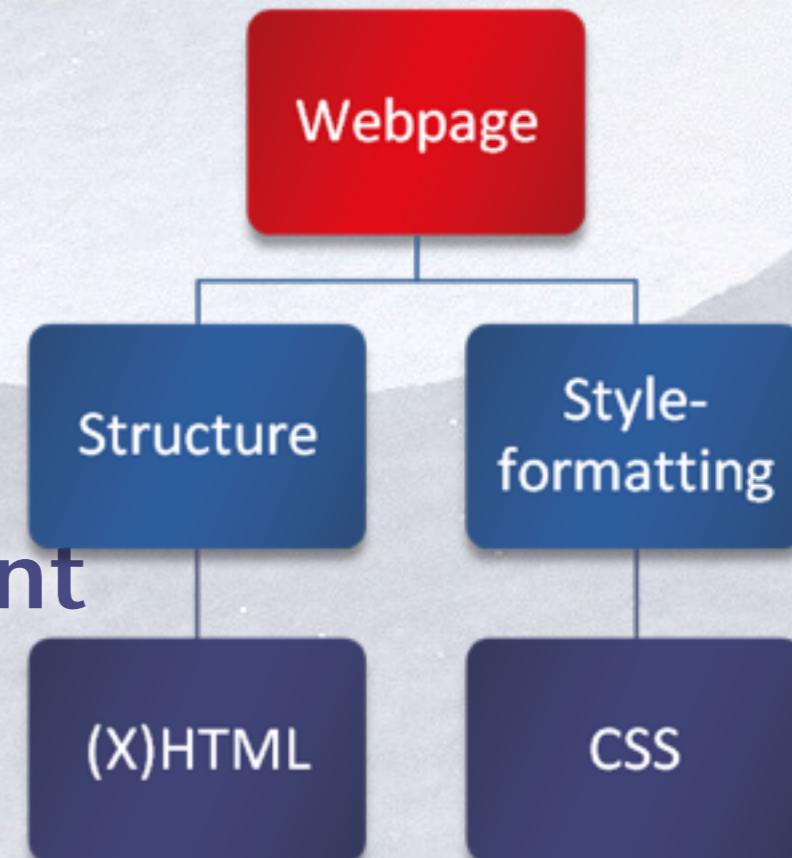
- \* more flexible
- \* short
- \* clear

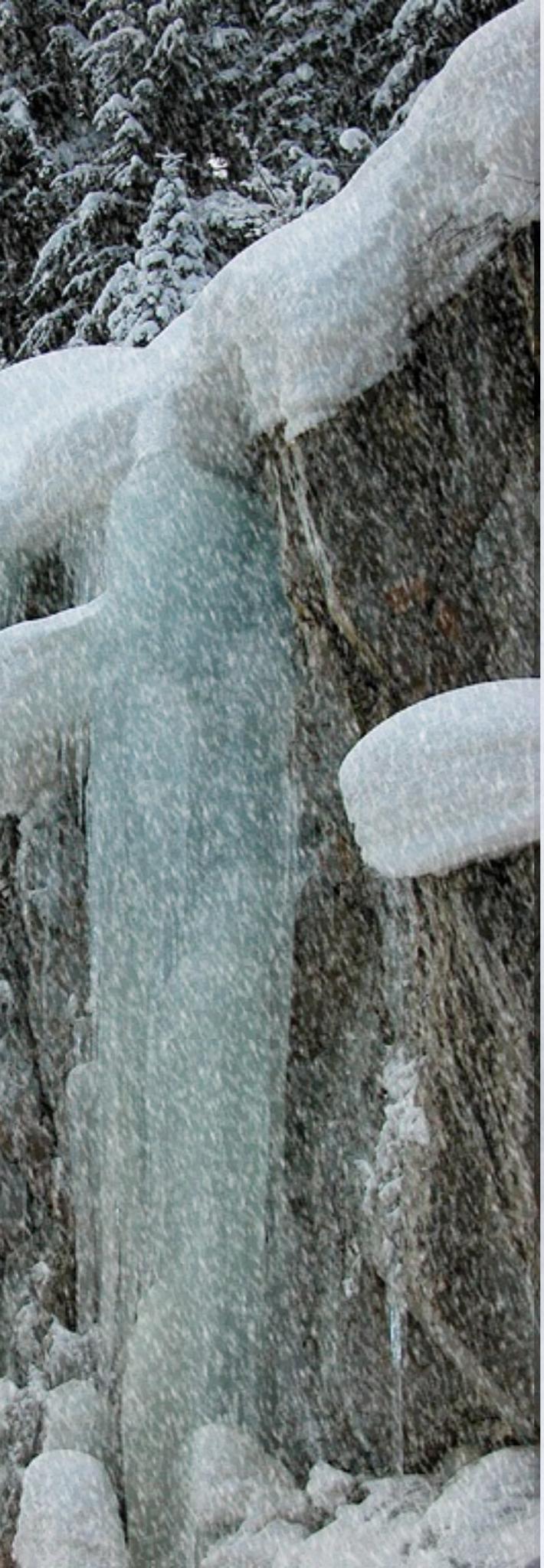
❖ Basic formatting tool

❖ Easy multiple document management

❖ Save time by using selector classes

❖ New opportunities in formatting





# Outline

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

# Syntax of CSS

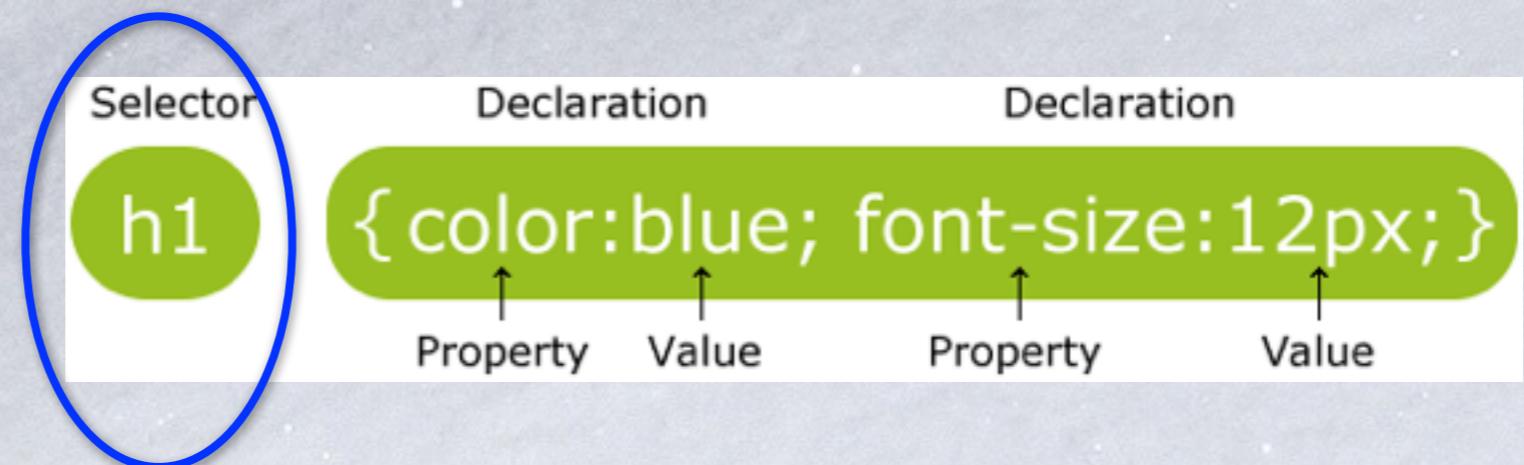
\* The CSS syntax is made up of 5 parts:

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

# Selector

\* **Definition:** identifies the HTML elements that the rule will be applied to, identified by the actual element name, e.g. <body>, or by other means such as class attribute values.

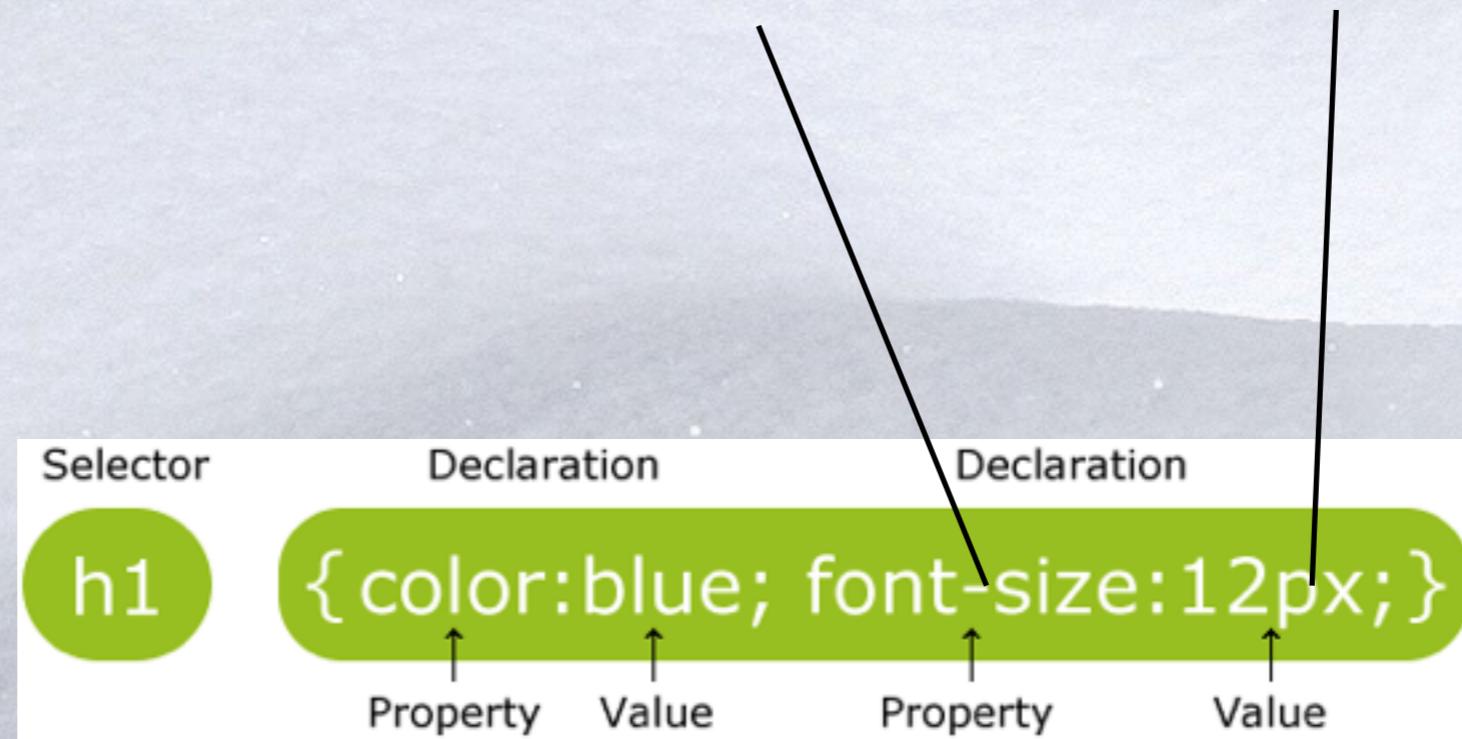
\* **Example:**



The selector is normally the HTML element you want to style

# Property & Value

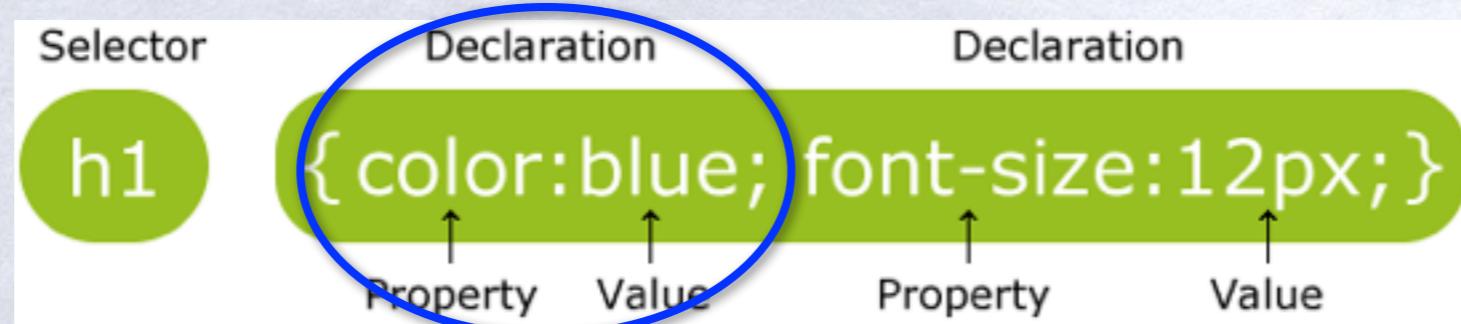
\* **Definition:** The property is the style attribute you want to change. Each **property** has a **value**.



- \* Properties are separated from their respective values by colons :
- \* Pairs are separated from each other by semicolons ;

# Declaration

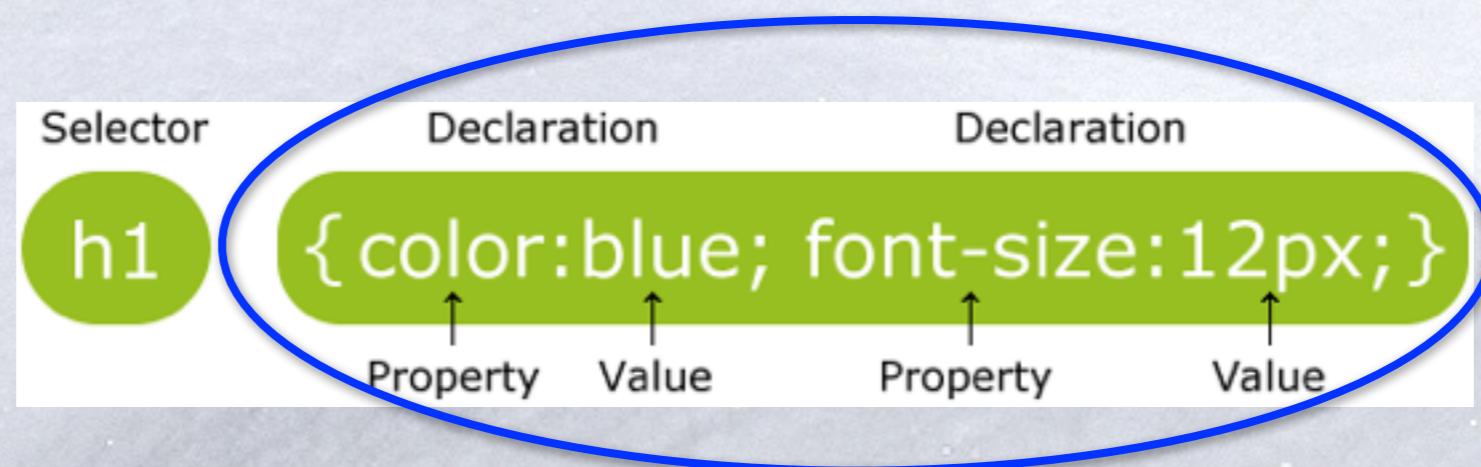
\* **Definition:** Each CSS line that includes property and value



\*Each declaration consists of a property and a value.

# Declaration Block

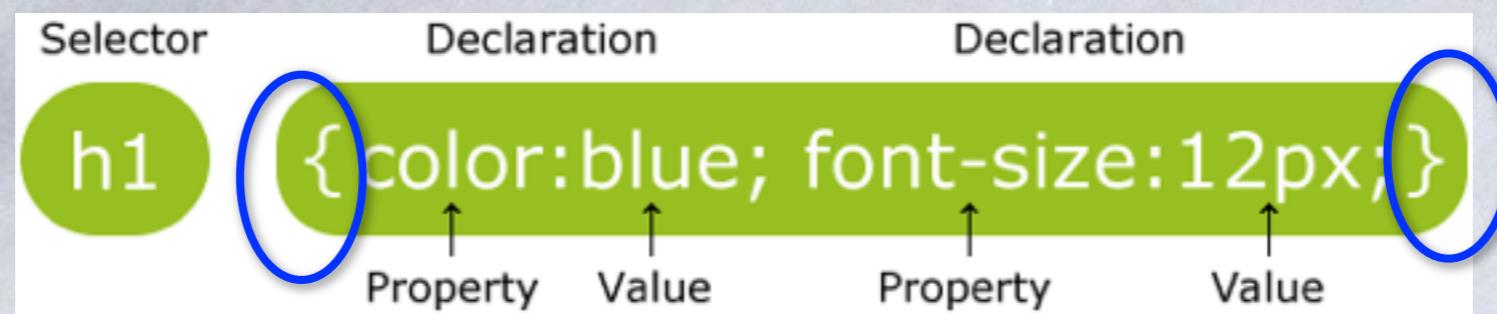
- \* Definition: multiple declaration lines including the curly braces



# Curly Braces

❖ **Definition:** the curly braces contain the properties of the element you want to manipulate, and the values that you want to change them to. The curly braces plus their content is called a declaration block.

❖ **Example:**



# Class and id Selectors

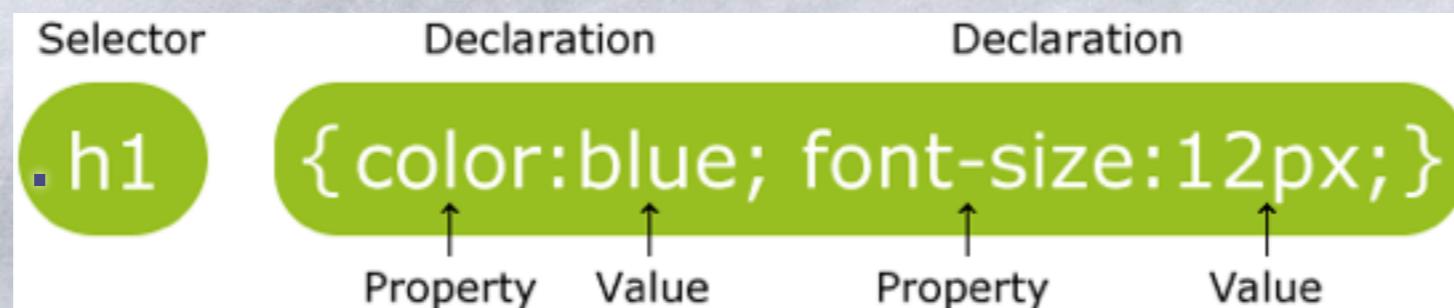
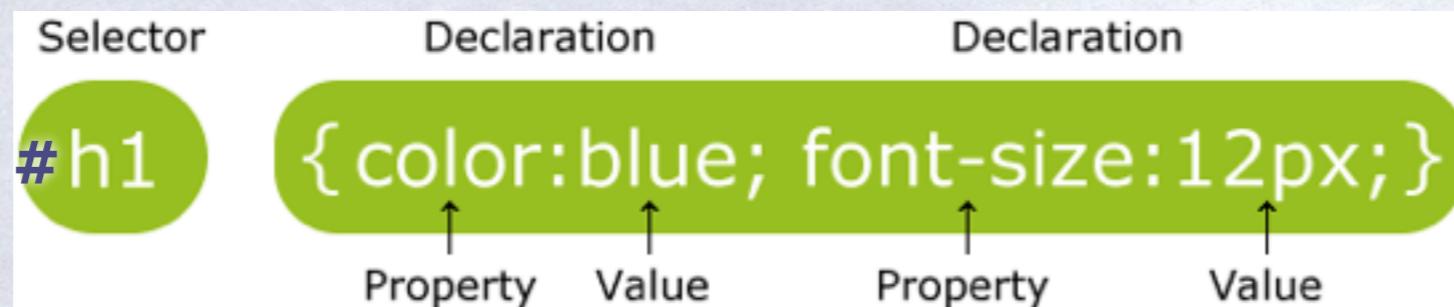
- ❖ In addition to setting a style for a HTML element, CSS allows you to specify your own selectors called "id" and "class".
- ❖ id - The id selector is used to specify a style for a single, unique element.
  - \* The id selector uses the id attribute of the HTML element, and is defined with a "#".
  - \* The style rule below will be applied to the element with id="para1":
    - \* `#para1 {text-align:center;color:red;}`

# Class and id 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;}`

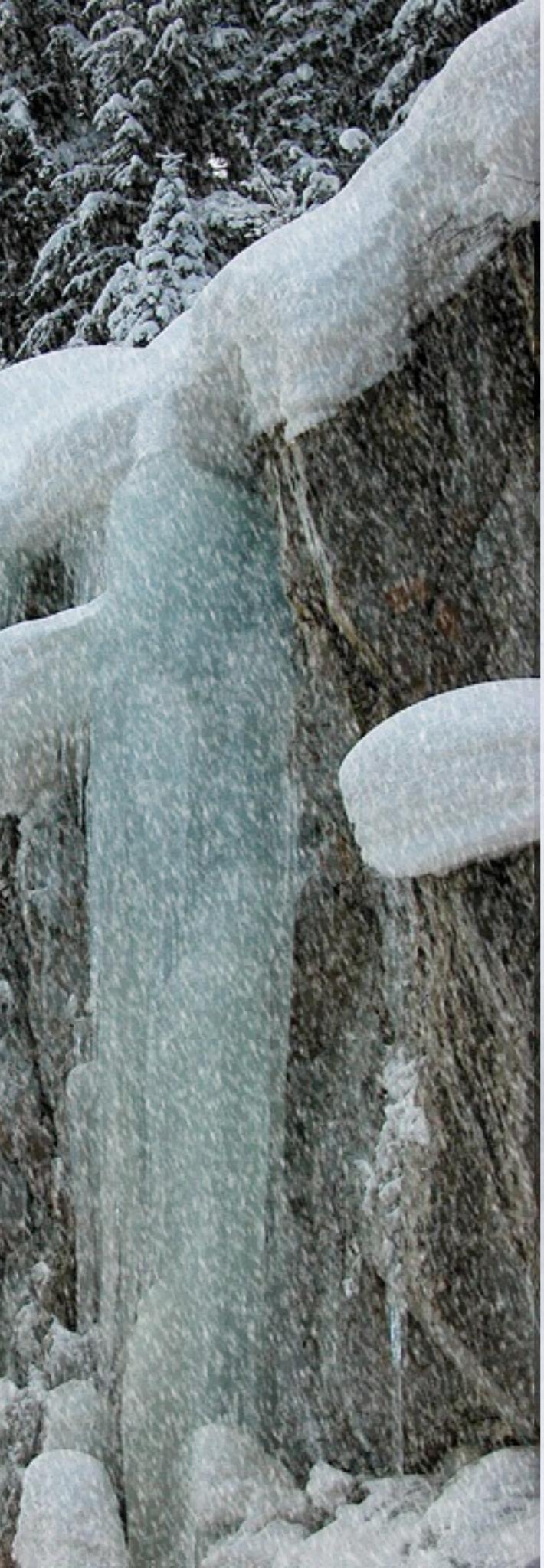
# Class and id Selectors

\* In the image below what is the h1 selector an ID or a Class?



# 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
  - ❖ Positioning
- 

# 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
  - \* inside external \*.css file
  - \* <link> tag &@import
- \* 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.

❖ 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: < em>Inheriting Styles</ em> Demo</ p>      Html

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

css

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

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

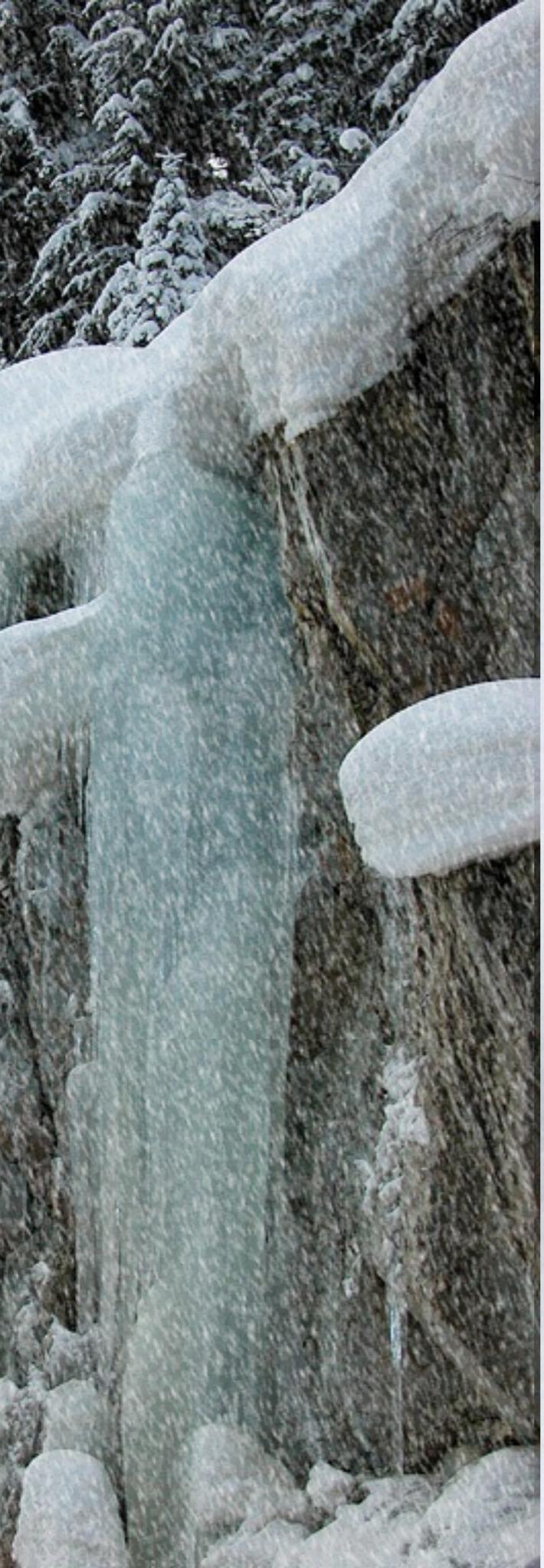
```
p.message {color: green;}
```

```
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;  
}
```



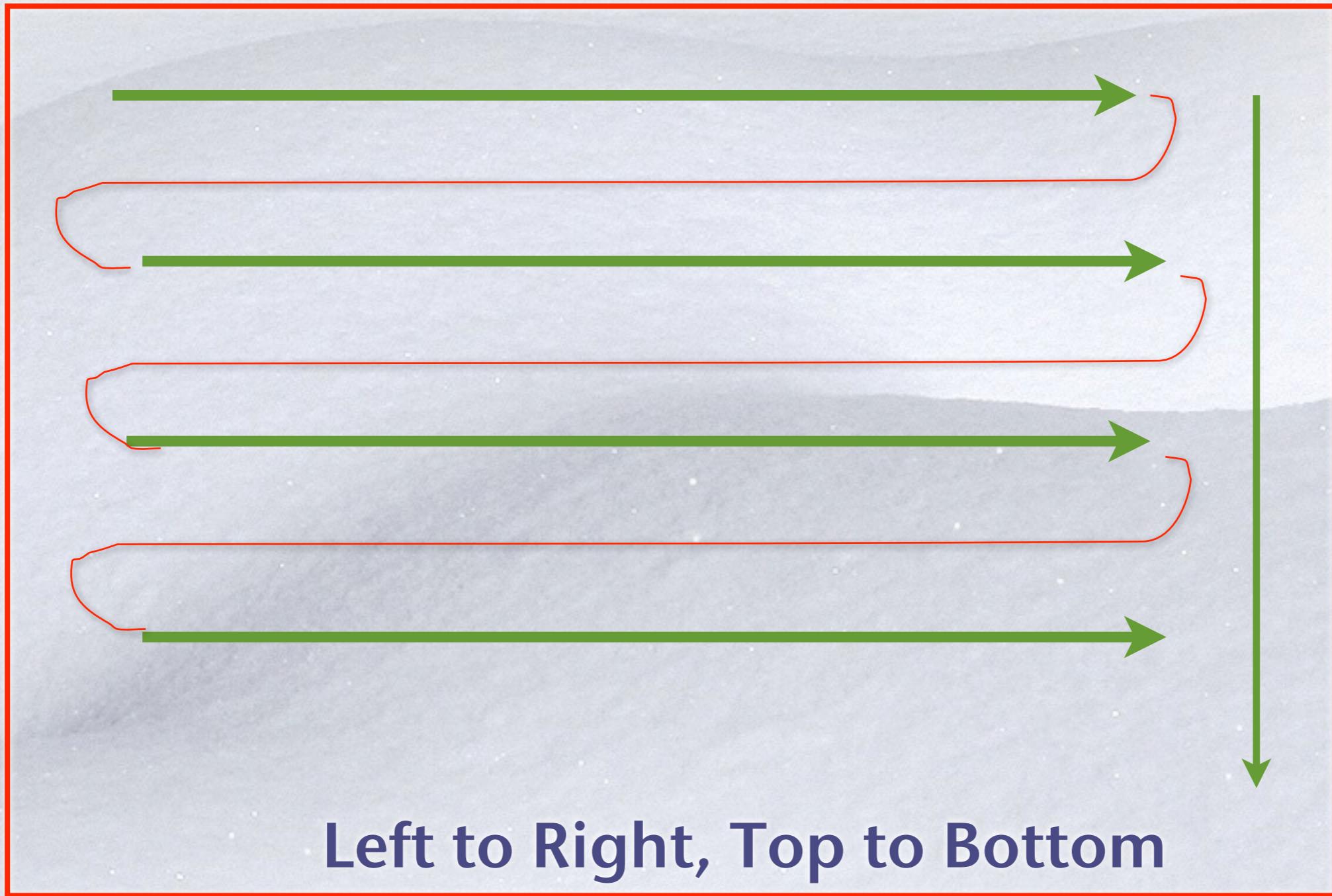
# Outline

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

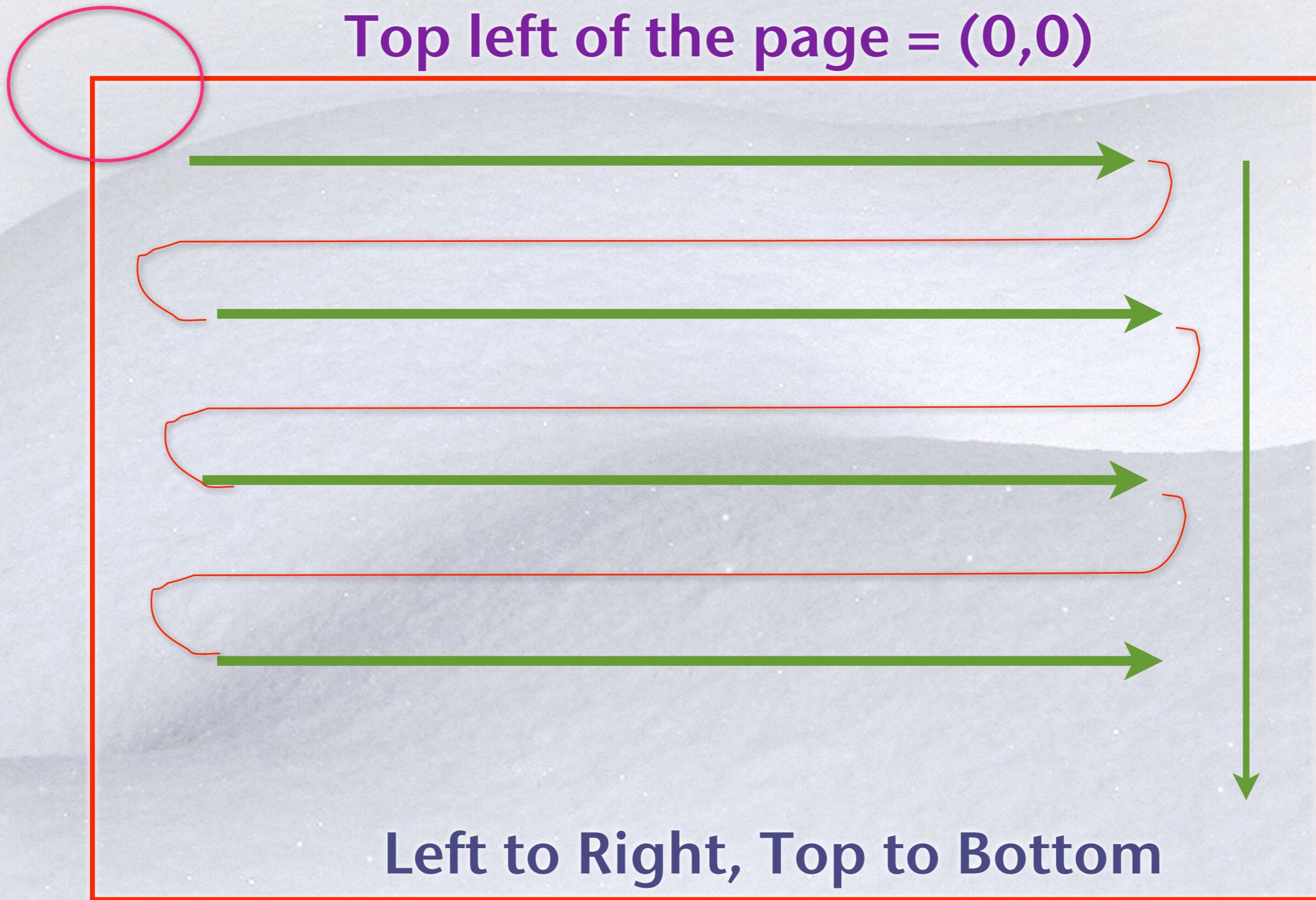
# Positioning – what is it?

- ❄ Positioning refers to the layout of the items on your page.
- ❄ It also refers to the “position” descriptor in CSS rules (more on this in a minute)
- ❄ <http://www.intensivstation.ch/en/templates/>
- ❄ <http://www.solucija.com/home/css-templates/>

# Normal Flow – no “positioning”



# Normal Flow – no “positioning”



# Normal Flow

This is a paragraph to which I have set the width.

If the next paragraph fits next to it on the right, it will line up.

The blue box is the container (more on this)

# Normal Flow

This is a paragraph to which I have set the width.

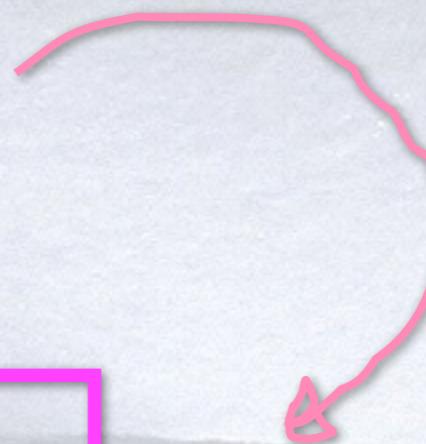
However, if the second paragraph is too wide to fit the container, it will shift down.

The blue box is the container (more on this)

# Normal Flow

This is a paragraph to which I have set the width.

However, if the second paragraph is too wide to fit the container, it will shift down.



This is the basic principle of Normal Flow

# Box Model

All of the items in your webpage generate invisible “boxes” – you have to figure out how all of those boxes will fit into your page, like a puzzle.

Image  
with link

Small  
print text,  
bullet list

Image

Set of links  
(navigation)

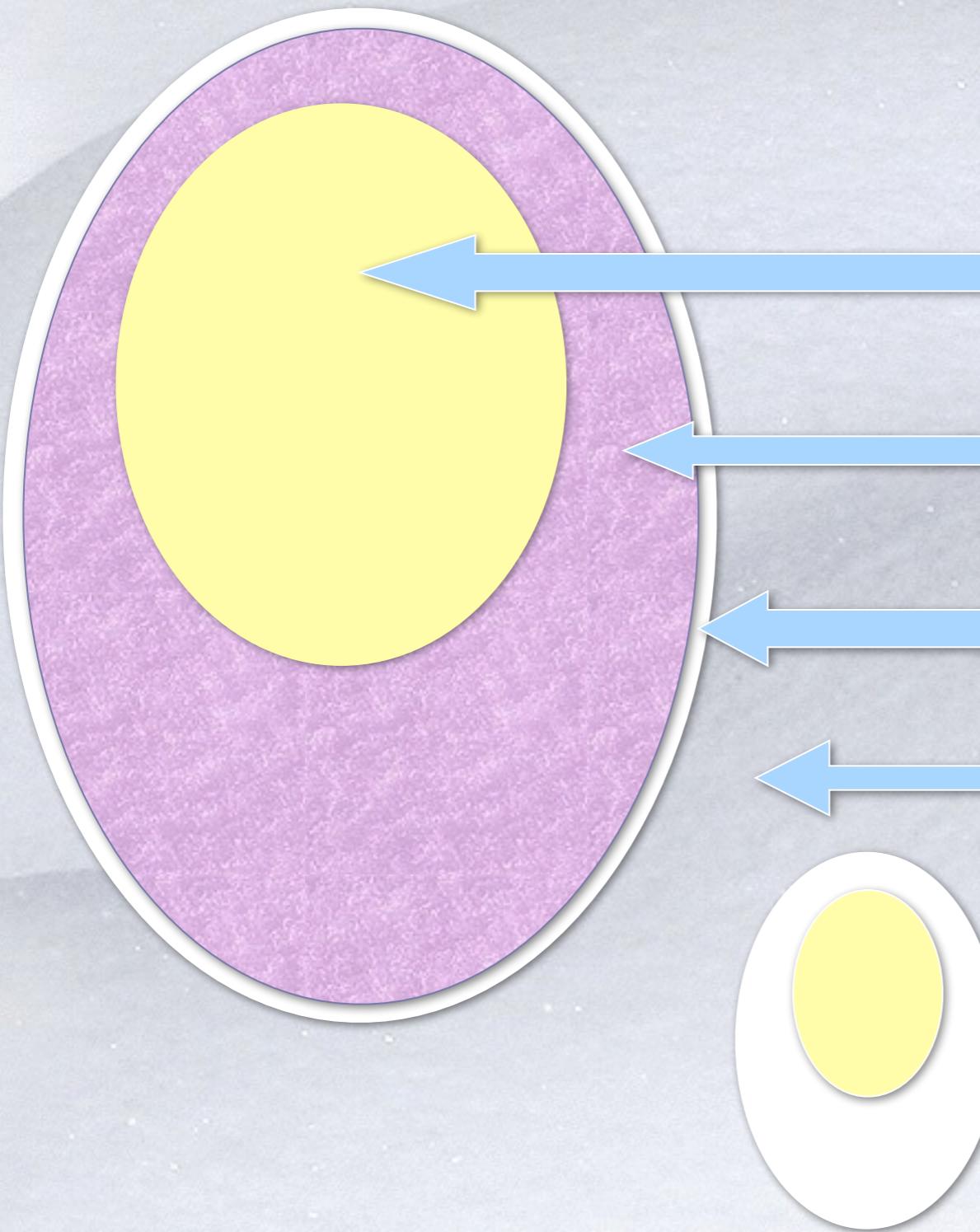
Regular text

# Box Model



# Box Model

Think of it like an egg:



**The yolk is the content**

**The albumen is the padding**  
**The shell is the border**

**The margin is how far the egg is from anything else  
(another tiny egg, perhaps)**

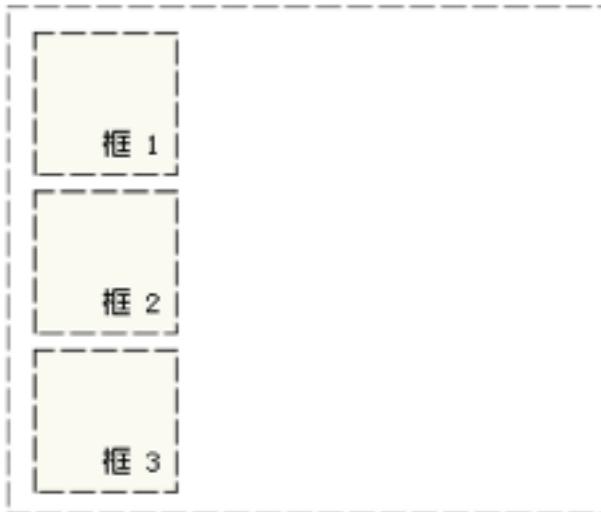
# Interrupt the Flow

- ❖ Absolute
- ❖ Relative
- ❖ Float

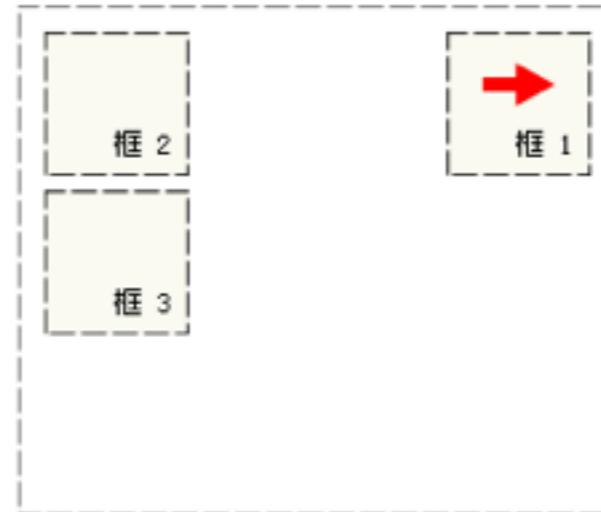
When you want to do fancier layout, you can **position** “boxes” or “containers.” By doing this, you interrupt the normal (top to bottom, left to right) flow. You can do this in three ways; **Float**, **Absolute**, and **Relative**.

# Float

不浮动的框



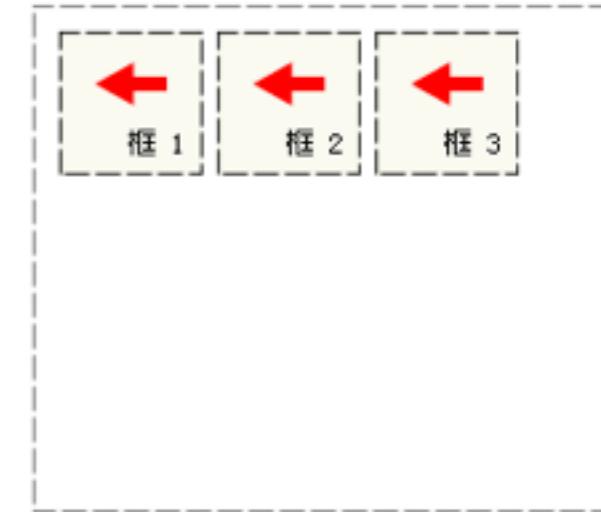
框 1 向右浮动



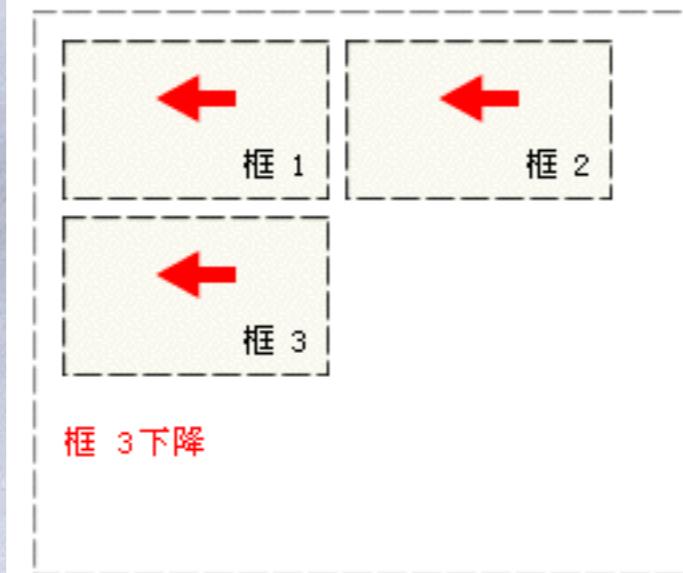
框 1 向左浮动



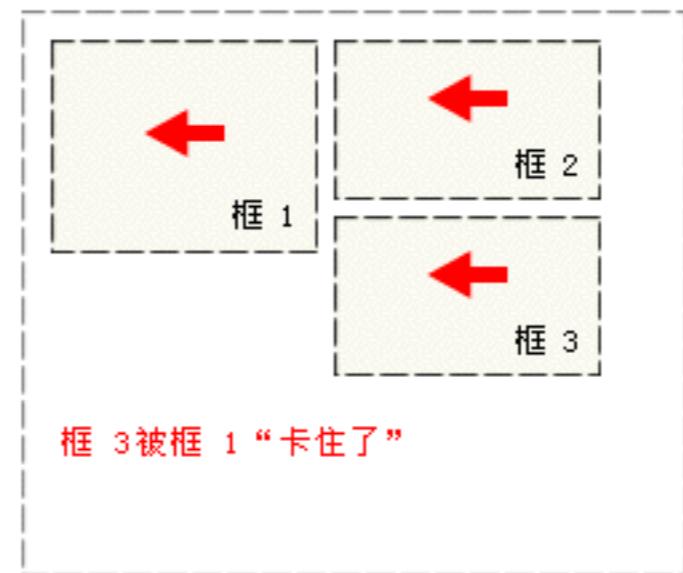
所有三个框向左浮动



框 1 向左浮动



所有三个框向左浮动



# Absolute

```
<html>
<head>
<style type="text/css">
h2.pos_abs
{
position:absolute;
left:100px;
top:150px
}
</style>
</head>
```

```
<body>
<h2 class="pos_abs">这是带有绝对定位的标题</h2>
<p>通过绝对定位，元素可以放置到页面上的任何位置。下面的标
题距离页面左侧 100px，距离页面顶部 150px。</p>
</body>

</html>
```

通过绝对定位，元素可以放置到页面上的任何位置。下面的标题距离页面左侧 100px，距离页面顶部 150px。

这是带有绝对定位的标题

# Relative

```
<html>
<head>
<style type="text/css">
h2.pos_left
{
position:relative;
left:-20px
}
h2.pos_right
{
position:relative;
left:20px
}
</style>
</head>
<body>
<h2>这是位于正常位置的标题</h2>
<h2 class="pos_left">这个标题相对于其正常位置向左移动</h2>
<h2 class="pos_right">这个标题相对于其正常位置向右移动</h2>
<p>相对定位会按照元素的原始位置对该元素进行移动。</p>
<p>样式 "left:-20px" 从元素的原始左侧位置减去 20 像素。</p>
<p>样式 "left:20px" 向元素的原始左侧位置增加 20 像素。</p>
</body>
```

这是位于正常位置的标题

这个标题相对于其正常位置向左移动

这个标题相对于其正常位置向右移动

相对定位会按照元素的原始位置对该元素进行移动。

样式 "left:-20px" 从元素的原始左侧位置减去 20 像素。

样式 "left:20px" 向元素的原始左侧位置增加 20 像素。

```
<html>
  <head>
    <title>My Resume</title>
    <link href="home.css" rel="stylesheet"
      type="text/css">
  </head>
  <body>
    <div id="container">
      Content goes here
    </div>
  </body>
</html>
```

HTML

**#container**

**#banner**

**#nav**

**#content**

**#footer**

```
<body>
<div id="container">
    <div id="banner"></div>
    <div id="nav">
        <span><a href="home.htm">home</a></span>
        <span><a href="contact.htm">contact</a></span>
    </div>
    <div id="content">
        <h1>My Resume</h1>
        <p>Resume text</p>
    </div>
    <div id="footer">Copyright info here</div>
</div>
</body>
```

HTML

```
body {  
    font-size: 1em; }  
  
#container {  
    width: 920px; }  
  
#banner {  
    width: 920px; height: 120px; }  
  
#nav {  
    float: left; width: 200px; }  
  
#content {  
    width: 720px; }  
  
#footer {  
    font-size: .8em; }
```

CSS

# 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

<http://www.csszengarden.com/?cssfile=/209/209.css>

Download the sample [html file](#) and [css file](#)

# CSS\*Zen Garden

A demonstration of what can be accomplished visually through CSS-based design. Select any stylesheet from the list to load it into this page.

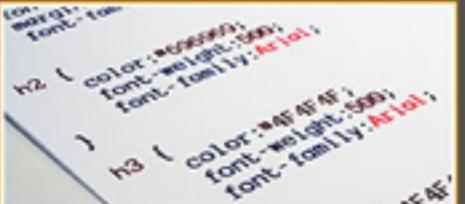
## \* Select a Design

- [Under the Sea!](#)  
by [Eric Stoltz](#)
- [Make 'em Proud](#)  
by [Michael McAghan and Scotty Reifsnnyder](#)
- [Orchid Beauty](#)  
by [Kevin Addison](#)
- [Oceanscape](#)  
by [Justin Gray](#)
- [CSS Co., Ltd.](#)  
by [Benjamin Klemm](#)
- [Sakura](#)  
by [Tatsuya Uchida](#)
- [Kyoto Forest](#)  
by [John Politowski](#)

## \* The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible [DOMs](#), and broken [CSS](#) support. Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the [W3C](#), [WaSP](#) and the major browser creators. The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

## \* So What is This About



```
h2 { color:#000000; font-weight:bold; font-family:Arial; }  
h3 { color:#4F4F4F; font-weight:bold; font-family:Arial; }
```

## \* Participation



<http://www.csszengarden.com/?cssfile=207/207.css>



“ A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page. ”

## The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

## So What is This About?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks

**CSS ZEN GARDEN**

Select a design

<http://www.csszengarden.com/?cssfile=192/192.css>



*CSS Zen Garden*

## The Beauty of CSS Design

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample [html file](#) and [css file](#)

## THE ROAD TO ENLIGHTENMENT

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support. Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators. The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

## SO WHAT IS THIS ABOUT?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets

BOSTON GLOBE—TUESDAY, APRIL 30, 1866  
FAMOUS 225-YEAR-OLD LANDMARK  
THAT BOSTON SOON MAY KNOW NO MORE  
Home of Paul Revere in Danger of Being Razèd—  
Movement Afoot to Preserve It.

Select a design

- Under the Seal by Eric Stoltz
- Make 'em Proud by Michael McAghon and Scotty Reifsnyder
- Orchid Beauty by Kevin Addison
- Oceanscape by Justin Gray
- CSS Co., Ltd. by Benjamin Klemm
- Sakura by Tatsuya Uchida
- Kyoto Forest by John Politowski
- A Walk in the Garden by Simon Van Hauwermeiren

Archives

- [next designs »](#)
- [View All Designs](#)

Resources



# W3C CSS Validator



\* [jigsaw.w3.org/css-validator/](http://jigsaw.w3.org/css-validator/)

- \* checks your CSS to make sure it meets the official CSS specifications
- \* more picky than the web browser, which may render malformed CSS correctly

Romanian Magyar Ελληνικά 简体中文

通过指定URI 通过文件上传 通过直接输入

验证指定URI的CSS内容  
输入你想验证的文档(带CSS的HTML文档或者CSS文档)的URI:

URI地址:

更多选项

Check

Try now the W3C Validator Suite™ premium service that checks your entire website and evaluates its conformance with W3C open standards to quickly identify those portions of your website that need your attention.

The W3C validators are hosted on server technology donated by HP, and supported by community donations. [Donate and help us build better tools for a better web.](#)

4382

Flattr

注意: 如果你想验证一段嵌入在HTML或者XHTML文档中的CSS, 你首先要[验证你的HTML或者XHTML文档是正确的。](#)

# Reading materials

- ❄ [http://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets)
- ❄ 精通CSS：高级Web标准解决方案
- ❄ W3 List of all CSS properties: <http://www.w3.org/TR/CSS21/propidx.html>
- ❄ W3 CSS 2.1 Specifications: <http://www.w3.org/TR/CSS21/>
- ❄ <http://www.w3.org/TR/CSS21/cascade.html#specificity>

# Thanks!!!

