Properties

- CSS1 includes 60 different properties in 7 categories:
  - Fonts
  - Lists
  - Alignment of text
  - Margins
  - Colors
  - Backgrounds
  - Borders
Property Values

- Keywords - left, small, ...
  - Not case sensitive
- Length - numbers, possibly with decimal points
- Units:
  - px - pixels
  - in - inches, cm - centimeters, mm - millimeters
  - pt - points
  - pc - picas (12 points)
  - em - height of the letter ‘m’
  - ex-height - height of the letter ‘x’
  - No space is allowed between the number and the unit specification e.g., 1.5 in is illegal
Property Values (continued)

• Percentage - just a number followed immediately by a percent sign
• URL values
  – url(protocol://server/pathname)
• Colors
  – Color name
  – rgb(n1, n2, n3)
    • Numbers can be decimal or percentages
  – Hex form: #XXXXXX
• Property values are inherited by all nested tags, unless overridden
Font Properties

<table>
<thead>
<tr>
<th>Generic Name</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>serif</td>
<td>Times New Roman, Garamond</td>
</tr>
<tr>
<td>sans-serif</td>
<td>MS Arial, Helvetica</td>
</tr>
<tr>
<td>cursive</td>
<td>Caflisch Script, Zapf-Chancery</td>
</tr>
<tr>
<td>fantasy</td>
<td>Critter, Cottonwood</td>
</tr>
<tr>
<td>monospace</td>
<td>Courier, Prestige</td>
</tr>
</tbody>
</table>
Font Properties (continued)

- **font-size**
  - Possible values: a length number or a name, such as `smaller`, `xx-large`, etc.
- **font-style**
  - `italic`, `oblique (useless)`, `normal`
- **font-weight - degrees of boldness**
  - `bolder`, `lighter`, `bold`, `normal`
    - Could specify as a multiple of 100 (100 – 900)
- **font**
  - For specifying a list of font properties
  - `font: bolder 14pt Arial Helvetica`
  - Order must be: style, weight, size, name(s)
- **Examples:** [fonts.html](http://example.com/fonts.html), [fonts2.html](http://example.com/fonts2.html)
- **The text-decoration property**
  - `line-through`, `overline`, `underline`, `none`
  - `letter-spacing` – value is any length property value
List properties

- list-style-type
- *Unordered lists*
  - Bullet can be a disc (default), a square, or a circle
  - Set it on either the `<ul>` or `<li>` tag
    - On `<ul>`, it applies to list items...

```html
<h3>Some Common Single-Engine Aircraft</h3>
<ul style="list-style-type: square">
  <li>Cessna Skyhawk</li>
  <li>Beechcraft Bonanza</li>
  <li>Piper Cherokee</li>
</ul>
```
List properties

- On `<li>`, `list-style-type` applies to just that item

<h3>Some Common Single-Engine Aircraft</h3>

<ul>
  <li style="list-style-type: disc">
    Cessna Skyhawk
  </li>
  <li style="list-style-type: square">
    Beechcraft Bonanza
  </li>
  <li style="list-style-type: circle">
    Piper Cherokee
  </li>
</ul>
List properties (continued)

- Can use an image for the bullets in an unordered list
  - Example:
    <li style = "list-style-image: url(bird.jpg)"

- On ordered lists - list-style-type can be used to change the sequence values

<table>
<thead>
<tr>
<th>Property Values</th>
<th>Sequence Type</th>
<th>First Four Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>decimal</td>
<td>Arabic numerals</td>
<td>1, 2, 3, 4</td>
</tr>
<tr>
<td>upper-alpha</td>
<td>Uppercase letters</td>
<td>A, B, C, D</td>
</tr>
<tr>
<td>lower-alpha</td>
<td>Lowercase letters</td>
<td>a, b, c, d</td>
</tr>
<tr>
<td>upper-roman</td>
<td>Uppercase Roman numerals</td>
<td>i, II, III, IV</td>
</tr>
<tr>
<td>lower-roman</td>
<td>Lowercase Roman numerals</td>
<td>i, ii, iii, iv</td>
</tr>
</tbody>
</table>

→ Example: sequence_types.html

- CSS2 has more, like lower-greek and hebrew
Colors

• Color is a problem for the Web for two reasons:
  1. Monitors vary widely
  2. Browsers vary widely

• The color property specifies the foreground colour of elements

```html
<style type = "text/css">
  th.red {color: red}
  th.orange {color: orange}
</style>
```

```html
...<table border = "5">
  <tr>
    <th class = "red"> Apple </th>
    <th class = "orange"> Orange </th>
    <th class = "orange"> Screwdriver </th>
  </tr>
</table>
```

• The background-color property specifies the background color of elements
Colors

• There are three color collections

  1. There is a set of 16 colors that are guaranteed to be displayable by all graphical browsers on all color monitors


<table>
<thead>
<tr>
<th>Name</th>
<th>Hexadecimal Code</th>
<th>Name</th>
<th>Hexadecimal Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>black</td>
<td>000000</td>
<td>green</td>
<td>008000</td>
</tr>
<tr>
<td>silver</td>
<td>C0C0C0</td>
<td>lime</td>
<td>00FF00</td>
</tr>
<tr>
<td>gray</td>
<td>808080</td>
<td>olive</td>
<td>808080</td>
</tr>
<tr>
<td>white</td>
<td>FFFFFF</td>
<td>yellow</td>
<td>FFFFFF</td>
</tr>
<tr>
<td>maroon</td>
<td>800000</td>
<td>navy</td>
<td>000080</td>
</tr>
<tr>
<td>red</td>
<td>FF0000</td>
<td>blue</td>
<td>0000FF</td>
</tr>
<tr>
<td>purple</td>
<td>800080</td>
<td>teal</td>
<td>008080</td>
</tr>
<tr>
<td>fuchsia</td>
<td>FF00FF</td>
<td>aqua</td>
<td>00FFFF</td>
</tr>
</tbody>
</table>

  2. There is a much larger set, the Web Palette

• 216 named colors http://www.w3schools.com/html/htmlcolornames.asp

  3. Any one of 16 million different colors

• #000000, #000001, #000002, ..., FFFFFF, FFFFFF

Some material Copyright © 2013 Pearson Education, Inc. Publishing as Pearson Addison-Wesley.
Alignment of Text

• The `text-indent` property allows indentation
  – Takes either a length or a % value

• The `text-align` property has the possible values, `left` (the default), `center`, `right`, or `justify`

• Sometimes we want text to flow around another element - the `float` property
  – The `float` property has the possible values, `left`, `right`, and `none` (the default)
  – If we have an element we want on the right, with text flowing on its left, we use the default `text-align` value (`left`) for the text and the `right` value for `float` on the element we want on the right
Alignment of Text

Some text with the default alignment - left

This is a picture of a Cessna 210. The 210 is the flagship single-engine Cessna aircraft. Although the 210 began as a four-place aircraft, it soon acquired a third row of seats, stretching it to a six-place plane. The 210 is classified as a high performance airplane, which means its landing gear is retractable and its engine has more than 200 horsepower. In its first model year, which was 1960, the 210 was powered by a 260 horsepower fuel-injected six-cylinder engine that displaced 471 cubic inches. The 210 is the fastest single-engine airplane ever built by Cessna.
Working with the Box Model

- The **box model** is an element composed of four sections:
  - Margin
  - Border
  - Padding
  - content
The Box Model

• Borders – every element has a `border-style` property
  – Controls whether the element has a border and if so, the style of the border
• `border-style` values: none, dotted, dashed, and double
• `border-width`: thin, medium (default), thick, or a length value in pixels
  – Border width can be specified for any of the four borders (e.g., `border-top-width`)
• `border-color`: any color
  – Border color can be specified for any of the four borders (e.g., `border-top-color`)
  – Example: `borders.html`
## Border Styles

<table>
<thead>
<tr>
<th>Border Style</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>border-top-width: value</td>
<td>Width of the top border</td>
<td>Where <code>value</code> is the width of the border in absolute or relative units, or defined with the keyword &quot;thin&quot;, &quot;medium&quot;, or &quot;thick&quot;</td>
</tr>
<tr>
<td>border-right-width: value</td>
<td>Width of the right border</td>
<td></td>
</tr>
<tr>
<td>border-bottom-width: value</td>
<td>Width of the bottom border</td>
<td></td>
</tr>
<tr>
<td>border-left-width: value</td>
<td>Width of the left border</td>
<td></td>
</tr>
<tr>
<td>border-width: top right bottom left</td>
<td>Width of any or all of the borders</td>
<td></td>
</tr>
<tr>
<td>border-top-color: color</td>
<td>Color of the top border</td>
<td>Where <code>color</code> is a color name or color value</td>
</tr>
<tr>
<td>border-right-color: color</td>
<td>Color of the right border</td>
<td></td>
</tr>
<tr>
<td>border-bottom-color: color</td>
<td>Color of the bottom border</td>
<td></td>
</tr>
<tr>
<td>border-left-color: color</td>
<td>Color of the left border</td>
<td></td>
</tr>
<tr>
<td>border-color: top right bottom left</td>
<td>Color of any or all of the borders</td>
<td></td>
</tr>
<tr>
<td>border-top-style: type</td>
<td>Style of top border</td>
<td>Where <code>type</code> is one of the nine border styles: solid, dashed, dotted, double, outset, inset, groove, ridge, or none</td>
</tr>
<tr>
<td>border-right-style: type</td>
<td>Style of right border</td>
<td></td>
</tr>
<tr>
<td>border-bottom-style: type</td>
<td>Style of bottom border</td>
<td></td>
</tr>
<tr>
<td>border-left-style: type</td>
<td>Style of left border</td>
<td></td>
</tr>
<tr>
<td>border-style: top right bottom left</td>
<td>Style of any or all of the borders</td>
<td></td>
</tr>
</tbody>
</table>

![Border Styles Diagram](image-url)
The Box Model

• Margin – the space between the border of an element and its neighbor element
• The margins around an element can be set with margin-left, etc. - just assign them a length value

```
• margin-left: 0.35in;
• margin-bottom: 0.35in
```

This is a picture of a Cessna 210. The 210 is the flagship single-engine Cessna aircraft. Although the 210 began as a four-place aircraft, it soon acquired a third row of seats, stretching it to a six-place plane. The 210 is classified as a high performance airplane, which means its landing gear is retractable and its engine has more than 200 horsepower. In its first model year, which was 1960, the 210 was powered by a 260 horsepower fuel-injected six-cylinder engine that displaced 471 cubic inches. The 210 is the fastest single-engine airplane ever built by Cessna.
The Box Model

• Padding – the distance between the content of an element and its border
  – Controlled by padding, padding-left, etc.
  → Example: marpads.html
• Background Images
• The background-image property
  → Can also specify background-image
  – Repetition can be controlled
    • background-repeat property
    • Possible values: repeat (default), no-repeat, repeat-x, or repeat-y
    • background-position property
      – Possible values: top, center, bottom, left, or right
The `<span>` and `<div>` tags

- One problem with the font properties is that they apply to whole elements, which are often too large
  - Solution: a new tag to define an element within a larger element - `<span>`

- Use `<span>` to apply a document style sheet to its content
  ```html
  <style type = "text/css"> 
  bigred {font-size: 24pt; font-family: Arial; color: red}
  </style>
  <p>
  Now is the <span class = "bigred"> best time </span> ever!
  </p>
  ```
Positioning

• CSS2.1 has additional properties - one of the most important is *positioning*
  – *Normal Flow* - block formatting of *block* boxes, inline formatting of *inline* boxes, relative positioning of block or inline boxes
  – *Floats* - laid out according to normal flow, then shifted
  – *Absolute positioning* - box is removed entirely from normal flow
  – Values
    • static, relative, absolute, fixed
  – Offsets
    • top, right, left, bottom

• Each CSS box is laid out on the screen (or page) in one of the three ways: in its *normal* position, *relative* position or at an *absolute* position.

• This *position* property – value: *static* (i.e. “normal”), *relative*, *absolute*, or *fixed*; and the *top*, *right*, *bottom*, *left* offset properties – value: a length or percentage.
Absolute Positioning

- **Absolute Positioning**
  - The element is positioned relative to its first positioned (not static) ancestor

```html
<p style = "position: absolute; left: 50px; top: 100px;">

- If an element is nested inside another element and is absolutely positioned, the top and left properties are relative to the enclosing element

```
Relative Positioning

- *Relative Positioning*
  - If no `top` and `left` properties are specified, the element is placed exactly where it would have been placed if no `position` property were given
  - But it can be moved later using JavaScript

<p>This is some text.</p>
<span style="position:relative; top: -1em">Up we go!</span> Here is some more text.

</p>
Stacking Elements

- Specify stacking order with:
  - `z-index: value`
CSS2 Properties: 3D Layering

- In CSS2, each box has a position in three dimensions. In addition to their horizontal and vertical positions, boxes lie along a “z-axis” and are formatted one on top of the other.
- Positioning on the z-axis is controlled by the `z-index` property (0,1,2,3 ...). The higher `z-index`ed objects are stacked above objects with lower z-index.
- Given:
  ```css
  .pile {position:absolute; top: 1in; left:1in; width:1in; height:1in}
  ```

How would this HTML snippet to be rendered?

```html
<img class="pile" style="z-index:1" src="aibo.jpg" alt="AIBO" />
<div class="pile" style="z-index:3">Top text</div>
<div class="pile">Bottom text</div>
<div class="pile" style="z-index:2">Middle text</div>
```
Resulting page with v.s. without 3D layering
CSS2 Properties: Visibility, Overflow and Clipping

• The `visibility` property specifies whether the boxes generated by an element are rendered. Invisible boxes still affect layout.
• By default `visible`, a box is made invisible by setting its visibility to hidden:
  
  
  – `<p style="visibility: hidden"> ... </p>`

• Generally, the content of a box is confined within its edges. Sometimes, a box may overflow so that its content lies partly or entirely outside of the box. The `overflow` property specifies whether the content of a block is clipped (made invisible) when it overflows.

• Lastly, the `clip` property changes which part of a box is visible.
Working with overflow and clipping

• The overflow property syntax:
  – `overflow: type`

- `overflow: visible`
- `overflow: hidden`
- `overflow: scroll`
- `overflow: auto`
## Setting the Display Style

Values of the display style

<table>
<thead>
<tr>
<th>Display</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>block</td>
<td>Display as a block-level element</td>
</tr>
<tr>
<td>inline</td>
<td>Display as an inline element</td>
</tr>
<tr>
<td>inline-block</td>
<td>Display as an inline element with some of the properties of a block (much like an inline image or frame)</td>
</tr>
<tr>
<td>inherit</td>
<td>Inherit the display property of the element’s parent</td>
</tr>
<tr>
<td>list-item</td>
<td>Display as a list item</td>
</tr>
<tr>
<td>none</td>
<td>Do not display the element</td>
</tr>
<tr>
<td>run-in</td>
<td>Display as either an inline or block-level element depending on the context (CSS2)</td>
</tr>
<tr>
<td>table</td>
<td>Display as a block-level table (CSS2)</td>
</tr>
</tbody>
</table>
# Setting the Display Style

Values of the display style

<table>
<thead>
<tr>
<th>Display</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inline-table</td>
<td>Display as an inline table (CSS2)</td>
</tr>
<tr>
<td>table-caption</td>
<td>Treat as a table caption (CSS2)</td>
</tr>
<tr>
<td>table-cell</td>
<td>Treat as a table cell (CSS2)</td>
</tr>
<tr>
<td>table-column</td>
<td>Treat as a table column (CSS2)</td>
</tr>
<tr>
<td>table-column-group</td>
<td>Treat as a group of table columns (CSS2)</td>
</tr>
<tr>
<td>table-footer-group</td>
<td>Treat as a group of table footer rows (CSS2)</td>
</tr>
<tr>
<td>table-header-group</td>
<td>Treat as a group of table header rows (CSS2)</td>
</tr>
<tr>
<td>table-row</td>
<td>Treat as a table row (CSS2)</td>
</tr>
<tr>
<td>table-row-group</td>
<td>Treat as a group of table rows (CSS2)</td>
</tr>
</tbody>
</table>
Hiding Elements

- Two different styles that allow you to hide elements:
  - Display style
  - Visibility style

Visibility hidden
Object is hidden but still is part of the page flow

Display: none
Object is hidden and is removed from the page flow
CSS3 Online Resources

- CSS3 Information and Preview
  - [http://www.css3.info](http://www.css3.info)
- CSS3 online rule generator
  - [http://css3please.com/](http://css3please.com/)

**W3C Standards CSS3 Relevant Modules**
- CSS Transitions ([http://www.w3.org/TR/CSS3-transitions/](http://www.w3.org/TR/CSS3-transitions/))
- CSS 2D Transforms ([http://www.w3.org/TR/CSS3-2d-transforms/](http://www.w3.org/TR/CSS3-2d-transforms/))
- CSS 3D Transforms ([http://www.w3.org/TR/CSS3-3d-transforms/](http://www.w3.org/TR/CSS3-3d-transforms/))
- Media Queries ([http://www.w3.org/TR/CSS3-mediaqueries/](http://www.w3.org/TR/CSS3-mediaqueries/))
- Multi-Column Layout ([http://www.w3.org/TR/CSS3-multicol/](http://www.w3.org/TR/CSS3-multicol/))
- Web Fonts ([http://www.w3.org/TR/CSS3-webfonts/](http://www.w3.org/TR/CSS3-webfonts/))
Commonly Supported Properties at a Glance

Some material Copyright © 2010 by Dan Cederholm. Publishing as A BOOK APART. Edited by Wei Liu.
Vendor Prefix

A positive catalyst for the evolution to exciting technologies “… force the vendors and the Working Group to work together to devise the tests necessary to determine interoperability. Those tests can then guide those who follow, helping them to achieve interoperable status much faster. They could literally ship the prefixed implementation in one public beta and drop the prefix in the next.”

```
.foo {
  -webkit-border-radius: 10px;
  -moz-border-radius: 10px;
  border-radius: 10px;
}
```
CSS3 Backgrounds and Borders Module

- http://www.w3.org/TR/css3-background/
- Border Radius
  - http://www.css3.info/preview/rounded-border/
- Border Images
  - Section 6, Example 26, 27 of http://www.w3.org/TR/css3-background/
Transition

- transition-property: The property to be transitioned (in this case, the `background` property, the value can be *all*)
- transition-duration: How long the transition should last (0.3 seconds)
- transition-timing-function: How fast the transition happens over time (ease, linear, ease-in, ease-out, ease-in-out, and cubic-bezier)

```css
a.foo {
  padding: 5px 10px;
  background: #9c3;
  -webkit-transition-property: background;
  -webkit-transition-duration: 0.3s;
  -webkit-transition-timing-function: ease;
}

a.foo:hover {
  background: #690;
}
```
Transition

- transition-delay: the transition can be delayed after the user action (e.g. 0.5 seconds)
- shorthand – all together using the transition property:

```css
a.foo {
  padding: 5px 10px;
  background: #9c3;
  transition: background 0.3s ease 0.5s;
}

a.foo:hover {
  background: #690;
}
```
2D Transform – Scale Transform

• Scale Transform
  – scale(2.0) – twice as larger of the original
  – scale(0.5) – half of the original size

• Notice how the transform doesn’t disturb the rest of the elements in the document, and zooms the photo out from the center, without affecting the layout around it.

• You can also set a transform-origin that will dictate where the scaling will expand from: top, bottom, center, or a percentage.

ul.gallery li a:hover img {
  -webkit-transform: scale(1.5);
  -moz-transform: scale(1.5);
  -o-transform: scale(1.5);
  transform: scale(1.5);
}
ul.gallery li a:hover img {
  -webkit-transform: scale(1.5);
  -moz-transform: scale(1.5);
  -o-transform: scale(1.5);
  transform: scale(1.5);
  -webkit-box-shadow: 4px 4px 10px rgba(0, 0, 0, 0.5);
  -moz-box-shadow: 4px 4px 10px rgba(0, 0, 0, 0.5);
  box-shadow: 4px 4px 10px rgba(0, 0, 0, 0.5);
}

Some material Copyright © 2010 by Dan Cederholm. Publishing as A BOOK APART. Edited by Wei Liu.
2D Transform - ROTATE, SKEW, AND TRANSLATE

- rotate(degrees)
  - +1deg, ... +360deg rotate clockwise
  - -1deg, ..., -360deg rotate counter-clockwise
- skew(x, y)
  - x degrees horizontally
  - y degrees vertically
- translate(x,y)
  - Move from the original location x pixel to the right, and y pixel from the top
ul.gallery li a:hover img {
    -webkit-transform: scale(1.5) rotate(-10deg);
    -moz-transform: scale(1.5) rotate(-10deg);
    -o-transform: scale(1.5) rotate(-10deg);
    transform: scale(1.5) rotate(-10deg);
    -webkit-box-shadow: 4px 4px 10px rgba(0, 0, 0, 0.5);
    -moz-box-shadow: 4px 4px 10px rgba(0, 0, 0, 0.5);
    box-shadow: 4px 4px 10px rgba(0, 0, 0, 0.5);
}
ul.gallery li a:hover img {
    -webkit-transform: scale(1.5) skew(-5deg, 30deg);
    -moz-transform: scale(1.5) skew(-5deg, 30deg);
    -o-transform: scale(1.5) skew(-5deg, 30deg);
    transform: scale(1.5) skew(-5deg, 30deg);
}
Translate Example

ul.gallery li a:hover img {
  -webkit-transform: scale(1.5) translate(20px, 40px);
  -moz-transform: scale(1.5) translate(20px, 40px);
  -o-transform: scale(1.5) translate(20px, 40px);
  transform: scale(1.5) translate(20px, 40px);
}

Some material Copyright © 2010 by Dan Cederholm. Publishing as A BOOK APART. Edited by Wei Liu.
CSS 3 Gradients Example

• CSS gradients can be assigned anywhere that an image can be declared in the stylesheet. In other words, background-image, list-style-image, border-image, and generated content.

• Parameters
  – The type of gradient:
    • linear (in the example below) or radial gradient
  – The direction of gradient
    • The starting and end point, or
    • The starting point and angle in degrees
  – The starting colour and terminating colour

• Browsers that don’t yet support CSS gradients will ignore those background-image rules and just be flat white. Specify a flat background colour if so desired before

```css
#foo {
  background-image: -webkit-gradient(linear, 0% 0%, 0% 100%, from(#fff), to(#999));
  background-image: -moz-linear-gradient(0% 100% 90deg, #fff, #999);
}
```