

Programming Web

Dr. Jamil Alagha

*College of Engineering and Artificial Intelligence
University of Palestine*

CSS

Cascading Style Sheet

Agenda

- Introduction
- Colors
- Text
- Boxes
- Lists
- Layout

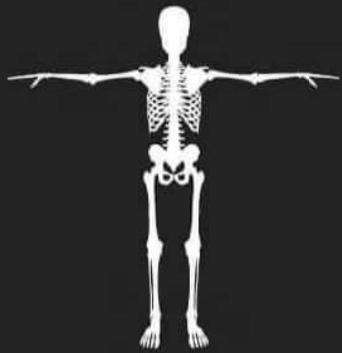
Introduction to CSS

Introduction to CSS

- In this section, we will look at how to make your web pages more attractive, controlling the design of them using CSS.
- CSS allows you to create rules that specify how the content of an element should appear.



HTML



Structure

CSS



Presentation

JavaScript



Behavior

Understanding CSS

- The key to understanding how CSS works is to imagine that there is an invisible box around every HTML element.

The Cottage Garden

The *cottage garden* is a distinct style of garden that uses an informal design, dense plantings, and a mixture of ornamental and edible plants.

The Cottage Garden originated in [England](#) and its history can be traced back for centuries, although they were re-invented in 1870's England, when stylized versions were formed as a reaction to the more structured and rigorously maintained [English estate gardens](#).

The earliest cottage gardens were more practical than their modern descendants, with an emphasis on vegetables and herbs, along with some fruit trees.

The Cottage Garden

The [cottage garden](#) is a distinct style of garden that uses an informal design, dense plantings, and a mixture of ornamental and edible plants.

The Cottage Garden originated in [England](#) and its history can be traced back for centuries, although they were re-invented in 1870's England, when stylized versions were formed as a reaction to the more structured and rigorously maintained [English estate gardens](#).

The earliest cottage gardens were more practical than their modern descendants, with an emphasis on vegetables and herbs, along with some fruit trees.

Understanding CSS

- CSS allows you to create rules that control the way that each individual box (and the contents of that box) is presented.
- In the last example, block level elements are shown with red borders, and inline elements have green borders.
- The `<body>` element creates the first box, then the `<h1>`, `<h2>`, `<p>`, `<i>`, and `<a>` elements each create their own boxes within it.
- Using CSS, you could add a border around any of the boxes, specify its width and height, or add a background color.
- You could also control text inside a box — for example, its color, size, and the typeface used.

Understanding CSS

- CSS works by associating rules with HTML elements.
- These rules govern how the content of specified elements should be displayed.
- A CSS rule contains two parts: a **selector** and a **declaration**.
- **Selectors** indicate which element the rule applies to.
- **Declarations** indicate how the elements referred to in the selector should be styled.



Understanding CSS

- The same rule can apply to more than one element if you separate the element names with **commas**.
- Declarations are split into two parts (a **property** and a **value**), and are separated by a **colon :**.
- You can specify several properties in one declaration, each separated by a **semi-colon ;**

```
h1, h2, h3 {  
    font-family: Arial;  
    color: yellow; }  
  
PROPERTY      VALUE
```

Adding Styles to the Document

There are three ways to attach a style sheet to a document:

- **External style sheets**

A separate, text-only `.css` file associated with the document with the `link` element or `@import` rule

- **Embedded style sheets**

Styles are listed in the `head` of the HTML document in the `style` element.

- **Inline styles**

Properties and values are added to an individual element with the `style` attribute.

Using External CSS

- The `<link>` element can be used in an HTML document to tell the browser where to find the CSS file used to style the page.
- It is an **empty element** (meaning it does not need a closing tag), and it lives **inside the `<head>` element**.
- It should use three attributes:
 - **href**: This specifies the path to the CSS file.
 - **type**: This attribute specifies the type of document being linked to. The value should be **text/css**.
 - **rel**: This specifies the relationship between the HTML page and the file it is linked to. The value should be **stylesheet** when linking to a CSS file.

Using External CSS

```
<!DOCTYPE html>
<html>
  <head>
    <title>Using External CSS</title>
    <link href="css/styles.css" type="text/css"
          rel="stylesheet" />
  </head>
  <body>
    <h1>Potatoes</h1>
    <p>There are dozens of different potato
       varieties. They are usually described as
       early, second early and maincrop.</p>
  </body>
</html>
```

```
body {
  font-family: arial;
  background-color: rgb(185,179,175);}
h1 {
  color: rgb(255,255,255);}
```

Potatoes

There are dozens of different potato varieties. They are usually described as early, second early and maincrop potatoes.

Using External CSS

An **@import** rule imports the contents of an external style sheet into another style sheet (either a *.css* document or embedded with **style**):

Import should be inside **<style> </style>** element

```
<head>
  <style>
    @import url("/path/stylesheet.css");
    p { font-face: Verdana; }
  </style>
  <title>Titles are required.</title>
</head>
```

Using Internal CSS

- You can also include CSS rules within an HTML page by placing them inside a `<style> </style>` element, which usually sits inside the `<head>` element of the page.
- The `<style>` element should use the **type** attribute to indicate that the styles are specified in CSS. The value should be **text/css**.

```
<head>
    <title>Using Internal CSS</title>
    <style type="text/css">
        body {
            font-family: arial;
            background-color: rgb(185,179,175);}
        h1 {
            color: rgb(255,255,255);}
    </style>
</head>
```

External CSS vs. Internal CSS

- When building a site with more than one page, you should use an external CSS style sheet.
- This:
 - Allows all pages to use the same style rules (rather than repeating them in each page).
 - Keeps the content separate from how the page looks.
 - Means you can change the styles used across all pages by altering just one file (rather than each individual page).

Inline CSS

- In HTML 4 and Transitional XHTML, you could also use a **style** attribute on most of the elements that appear in the body of a page.
- The CSS rules that appeared within the value of the attribute would only apply to that one element.
- You should avoid using this attribute in any new site.
- Here is an example that changes the color of the text in a single paragraph red:

```
<p style="color:red;">  
  
<p style="font-size: large;">Paragraph text...</p>  
  
<h3 style="color: red; margin-top: 30px;">Intro</h3>
```

CSS Selectors

- There are many different types of CSS selector that allow you to target rules to specific elements in an HTML document.
- **Universal Selector:** Applies to all elements in the document.
`* { }` targets all elements on the page
- **Type Selector:** Matches element names.
`h1, h2, h3 { }` Targets the `<h1>`, `<h2>` and `<h3>` elements
- **Class Selector:** Matches an element whose class attribute has a value that matches the one specified after the period (or full stop) symbol.
`.note { }` Targets any element whose `class` attribute has a value of note
`p.note { }` Targets only `<p>` elements whose `class` attribute has a value of note

CSS Selectors

- **ID Selector:** Matches an element whose id attribute has a value that matches the one specified after the pound or hash symbol.

`#introduction { }` Targets the element whose id attribute value is introduction

- **Child Selector:** Matches an element that is a direct child of another.

`li>a { }` Targets any `<a>` elements that are direct children of an `` element

- **Descendent Selector:** Matches an element that is a descendent of another specified element (not just a direct child of that element).

`p a { }` Targets any `<a>` elements that sit inside a `<p>` element, even if there are other elements nested between them

CSS Selectors

- **adjacent Sibling Selector:** Matches an element that is the next sibling of another
`h1+p {}` Targets the first direct `<p>` element after (not inside) any `<h1>` element (but not other `<p>` elements)
- **general Sibling Selector:** Matches an element that is a sibling of another, although it does not have to be the directly preceding element
`h1~p {}` If you had two `<p>` elements that are siblings of an `<h1>` element, this rule would apply to both

Example

- ```
h1>Potatoes</h1>
<p id="intro">There are
<i>dozens</i> of different
potato varieties.
</p>
```
- ```
<p>They are usually
described as early, second
early and maincrop
potatoes.</p>
```
- ```
* {font-family: Arial, Verdana,
sans-serif;}
h1 {font-family: "Courier New",
monospace;}
i {color: green;}
i {color: red;}
b {color: pink;}
p b {color: blue !important;}
p b {color: violet;}
p#intro {font-size: 100%;}
p { font-size: 75%;}
```

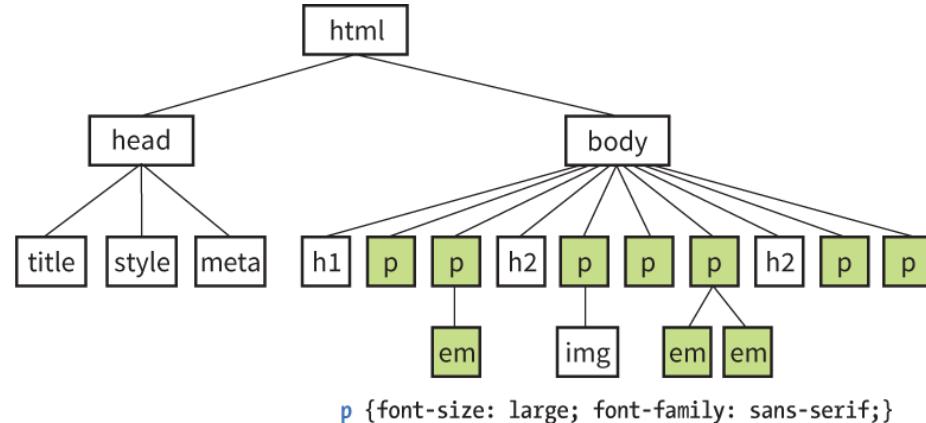
# CSS Comments

```
/* comment goes here */ css comment
!-- comment --> html comment
```

- Content between /\* and \*/ will be ignored by the browser.
- Useful for leaving notes or section labels in the style sheet.
- Can be used within rules to temporarily hide style declarations in the design process.

# Inheritance

- Many properties applied to elements are passed down to the elements they contain. This is called **inheritance**.
- For example, applying a sans-serif font to a **p** element causes the **em** element it contains to be sans-serif as well:



# Inheritance (cont'd)

- Some properties inherit; others do not.

Properties related to text usually inherit; properties related to layout generally don't.

- Styles explicitly applied to specific elements override inherited styles.
- You'll learn to use inheritance strategically to keep your style rules simple.

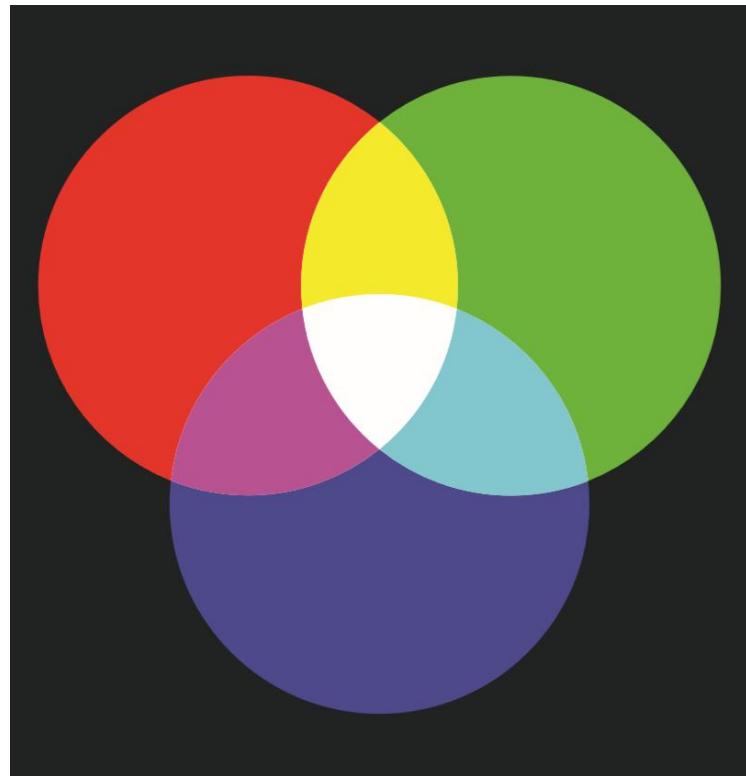
# Colors

# Foreground Color

- The **color** property allows you to specify the color of text inside an element.
- You can specify any **color** in CSS in one of three ways:
  - **RGB values**  
These express colors in terms of how much red, green and blue are used to make it up.
  - **HEX codes**  
These are six-digit codes that represent the amount of red, green and blue in a color, preceded by a pound or hash # sign.
  - **Color names**  
There are 147 predefined color names that are recognized by browsers.
    - [https://www.w3schools.com/colors/colors\\_names.asp](https://www.w3schools.com/colors/colors_names.asp)

# Understanding Colors

- Every color on a computer screen is created by mixing amounts of red, green, and blue.
- **RGB**  
Values for red, green, and blue are expressed as numbers between 0 and 255.
  - `rgb(200, 178, 230)`
  - `rgb(78%, 70%, 90%)`
- **HEX**  
Hex values represent values for red, green, and blue in hexadecimal code. (00 - FF)
  - `#C8B2E6`
  - `#F06` is the same as `#FF0066`
- **Names**  
Colors are represented by predefined names.



# Foreground Color

```
/* color name */
h1 {
 color: DarkCyan;}

/* hex code */
h2 {
 color: #ee3e80;}

/* rgb value */
p {
 color: rgb(100,100,90);}
```

## Marine Biology

### The Composition of Seawater

Almost anything can be found in seawater. This includes materials from the Earth's crust as well as materials released from organisms. The chemical components of seawater that influence life forms are dissolved gases (mostly oxygen and carbon dioxide), and elements vary in their composition as well as in their

# Background Colors

- CSS treats each HTML element as if it appears in a box, and the **background-color** property sets the color of the background for that box.
- You can specify your choice of background color in the same three ways you can specify foreground colors: RGB values, hex codes, and color names.
- If you do not specify a background color, then the background is **transparent**.
- By default, most browser windows have a white background, but browser users can set a background color for their windows, so if you want to be sure that the background is white you can use the background-color property on the <body> element. <body bgcolor="black" >

# Background Colors

```
body {
 background-color: rgb(200,200,200);}

h1 {
 background-color: DarkCyan;}

h2 {
 background-color: #ee3e80;}

p {
 background-color: white;}
```

Marine Biology

The Composition of Seawater

Almost anything can be found in seawater. This includes dissolved materials from Earth's crust as well as materials released from organisms. The most important components of seawater that influence life forms are salinity, temperature, dissolved gases (mostly oxygen and carbon dioxide), nutrients, and pH. These elements vary in their composition as well as in their influence on marine life.

# Opacity

- CSS3 introduces the **opacity** property which allows you to specify the opacity of an element and any of its child elements.
- The value is a **number** between **0.0** and **1.0**.
- The CSS3 **rgba** allows you to specify a color, just like you would with an RGB value, but adds a fourth value (alpha) to indicate opacity.
- The rgba value will only affect the element on which it is applied
- Because some browsers will not recognize RGBA colors, you can offer a fallback so that they display a solid color.
- If there are two rules that apply to the same element, the latter of the two will take priority.

# RGBa Color

- **RGB + an alpha channel** for transparency
- The first three values are RGB. The fourth is the transparency level from 0 (transparent) to 1 (opaque).

Playing with RGBa

Playing with RGBa

Playing with RGBa

```
color: rgba(0, 0, 0, .1);
```

```
color: rgba(0, 0, 0, .5);
```

```
color: rgba(0, 0, 0, 1);
```

# Opacity

```
p.one {
 background-color: rgb(0,0,0);
 opacity: 0.5;}

p.two {
 background-color: rgb(0,0,0);
 background-color: rgba(0,0,0,0.5);}
```

# Opacity

RESULT



RESULT IN OLDER BROWSER



**Text**

# Text

- The properties that allow you to control the appearance of text can be split into two groups:
  - Those that directly affect the font and its appearance ( typeface, regular, bold, and size).
  - Those that would have the same effect on text no matter what font you were using (color of text and the spacing between words and letters).
- The formatting of your text can have a significant effect on how readable your pages are.

# Specifying Typefaces

- The **font-family** property allows you to specify the typeface that should be used for any text inside the element(s) to which a CSS rule applies.
- The value of this property is the name of the typeface you want to use.
- The people who are visiting your site need the typeface you have specified installed on their computer in order for it to be displayed.
- You can specify a **list of fonts** separated by **commas**, so that, if the user does not have your first choice of typeface installed, the browser can try to use an alternative font from the list.
- If a font name is made up of **more than one word**, it should be put in double quotes.

# Specifying Typefaces

```
body {
 font-family: Georgia, Times, serif;}
h1, h2 {
 font-family: Arial, Verdana, sans-serif;}
.credits {
 font-family: "Courier New", Courier,
 monospace;}
```

# Size of Type

- The **font-size** property enables you to specify a size for the font. There are several ways to specify the size of a font. The most common are:
  - **Pixels (px)**  
Pixels are commonly used because they allow web designers very precise control over how much space their text takes up. The number of pixels is followed by the letters px.
  - **Points (pt)**  
A point corresponds to 1/72 of an inch.
  - **1 pt = 0.0352778 cm**
  - **Percentage**  
The default size of text in browsers is 16px. So a size of 75% would be the equivalent of 12px, and 200% would be 32px.
  - **EMS**  
An em is equivalent to the width of a letter m.
  - The default font size is 16 pixels ( used for <p>). **By default, 1em = 16 pixels.**

# Size of Type

```
body {
 font-family: Arial, Verdana, sans-serif;
 font-size: 12px;}

h1 {
 font-size: 200%;}

h2 {
 font-size: 1.3em;}
```

## Briards

by Ivy Duckett

The [briard](#), or berger de brie, is a large breed of dog traditionally used as a herder and guardian of sheep.

### Breed History

The briard, which is believed to have originated in France, has been bred for centuries to herd and to protect sheep. The breed was used by the French Army as sentries, messengers and to search for wounded soldiers because of its fine sense of hearing. Briards were used in the First World War almost to the point of extinction. Currently the population of briards is slowly recovering. Charlemagne, Napoleon, Thomas Jefferson and Lafayette all owned briards.

# Use a Custom Font

- **@font-face** allows you to use a font, even if it is not installed on the computer of the person browsing, by allowing you to specify a path to a copy of the font, which will be downloaded if it is not on the user's machine.
- There are two properties that used with **@font-face**:
  - **font-family**:  
This specifies the name of the font.
  - **src**:  
This specifies the path to the font.

# Use a Custom Font

```
@font-face {
 font-family: 'ChunkFiveRegular';
 src: url('fonts/chunkfive.eot');}

h1, h2 {
 font-family: ChunkFiveRegular, Georgia, serif;}
```

# Bold

- The **font-weight** property allows you to create bold text. There are two values that this property commonly takes.
  - **normal**: This causes text to appear at a normal weight.
  - **bold**: This causes text to appear bold.

**Values:** normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900

```
.credits {
 font-weight: bold;}
```

# Italic

- If you want to create italic text, you can use the **font-style** property.
- There are three values this property can take:
  - **normal**: This causes text to appear in a normal style (as opposed to italic or oblique).
  - **italic**: This causes text to appear italic.
  - **oblique**: This causes text to appear oblique (generally the same as italics).

```
.credits {
 font-style: italic;}
```

# Uppercase & Lowercase

- The **text-transform** property is used to change the case of text giving it one of the following values:
  - uppercase**: This causes the text to appear uppercase.
  - lowercase**: This causes the text to appear lowercase.
  - capitalize**: This causes the first letter of each word to appear capitalized.

```
h1 {
 text-transform: uppercase;}

h2 {
 text-transform: lowercase;}

.credits {
 text-transform: capitalize;}
```

## BRIARDS

By Ivy Duckett

The [briard](#), or berger de brie, is a large breed of dog traditionally used as a herder of sheep.

### breed history

# Underline & Strike

- The **text-decoration** property allows you to specify the following values:
  - **none**: This removes any decoration already applied to the text.
  - **underline**: This adds a line underneath the text.
  - **overline**: This adds a line over the top of the text.
  - **line-through**: This adds a line through words.

```
.credits {
 text-decoration: underline;}

a {
 text-decoration: none;}
```

by Ivy Duckett

The briard, or berger de brie,  
of sheep.

# Line Height

- In CSS, the **line-height** property sets the height of an entire line of text.
- Increasing the line-height makes the vertical gap between lines of text larger.

```
p {
 line-height: 1.4em;}
```

# Letter & Word Spacing

- You can control the space between each letter with the **letter-spacing** property.
- It is particularly helpful to increase the kerning when your heading or sentence is all in uppercase. If your text is in sentence (or normal) case, increasing or decreasing the kerning can make it harder to read.
- You can also control the gap between words using the **word-spacing** property.

# Letter & Word Spacing

```
h1, h2 {
 text-transform: uppercase;
 letter-spacing: 0.2em;}
.credits {
 font-weight: bold;
 word-spacing: 1em;}
```

## BRIARDS

by Ivy Duckett

The [briard](#), or berger de brie, is a large breed of dog traditionally used to protect sheep.

## BREED HISTORY

The briard, which is believed to have originated in France, was used to protect sheep. The breed was used by the French to search for wounded soldiers because of its fine sense of hearing. World War almost to the point of extinction. Currently the breed is recovering. Charlemagne, Napoleon, Thomas Jefferson and

# Alignment

- The **text-align** property allows you to control the alignment of text. The property can take one of four values:
  - **left**: This indicates that the text should be left-aligned.
  - **right**: This indicates that the text should be right-aligned.
  - **center**: This allows you to center text.
  - **justify**: This indicates that every line in a paragraph, except the last line, should be set to take up the full width of the containing box.

# Alignment

```
h1 {
 text-align: left;}

p {
 text-align: justify;}

.credits {
 text-align: right;}
```

## Briards

RESULT

by Ivy Duckett

The [briard](#), or berger de brie, is a large breed of dog traditionally used as a herder and guardian of sheep.

### Breed History

The briard, which is believed to have originated in France, has been bred for centuries to herd and to protect sheep. The breed was used by the French Army as sentries, messengers and to search for wounded soldiers because of its fine sense of hearing. Briards were used in the First World War almost to the point of extinction. Currently the population of briards is slowly recovering. Charlemagne, Napoleon, Thomas Jefferson and Lafayette all owned briards.

# Horizontal Text Alignment

## text-align

**Values:** left, right,  
center, justify, start,  
end

### Examples:

```
text-align: left;
```

Paragraph 1. The text-align property controls the horizontal alignment of the text within an element. It does not affect the alignment of the element on the page. The resulting text behavior of the various values should be fairly intuitive.

```
text-align: right;
```

Paragraph 2. The text-align property controls the horizontal alignment of the text within an element. It does not affect the alignment of the element on the page. The resulting text behavior of the various values should be fairly intuitive.

```
text-align: center;
```

Paragraph 3. The text-align property controls the horizontal alignment of the text within an element. It does not affect the alignment of the element on the page. The resulting text behavior of the various values should be fairly intuitive.

```
text-align: justify;
```

Paragraph 4. The text-align property controls the horizontal alignment of the text within an element. It does not affect the alignment of the element on the page. The resulting text behavior of the various values should be fairly intuitive.

# Indenting Text

- The **text-indent** property allows you to indent the **first line** of text within an element.
- The amount you want the line indented by can be specified in a number of ways but is usually given in pixels or ems.
- It can take a negative value, which means it can be used to push text off the browser window.

```
h1 {
 background-image: url("images/logo.gif");
 background-repeat: no-repeat;
 text-indent: -9999px;}

.credits {
 text-indent: 20px;}
```

# Indents

## text-indent

**Values:** *Length, percentage*

### Examples:

```
p {text-indent: 2em;}
```

```
p {text-indent: 25%;}
```

```
p {text-indent: -35px;}
```

Paragraph 1. The text-indent property indents only the first line of text by a specified amount. You can specify a length measurement or a percentage value.

Paragraph 2. The text-indent property indents only the first line of text by a specified amount. You can specify a length measurement or a percentage value.

Paragraph 3. The text-indent property indents only the first line of text by a specified amount. You can specify a length measurement or a percentage value.

# Text Shadow

- The **text-shadow** property has become commonly used despite lacking support in all browsers.
- The value of this property is quite complicated because it can take three lengths and a color for the drop shadow.
- The first length indicates how far to the left or right the shadow should fall.
- The second value indicates the distance to the top or bottom that the shadow should fall.
- The third value is optional and specifies the amount of blur that should be applied to the drop shadow.
- The fourth value is the color of the drop shadow.

# Text Shadow

```
p.one {
background-color: #eeeeee;
color: #666666;
text-shadow: 1px 1px 0px #000000;}

p.two {
background-color: #dddddd;
color: #666666;
text-shadow: 1px 1px 3px #666666;}

p.three {
background-color: #cccccc;
color: #ffffff;
text-shadow: 2px 2px 7px #111111;}

p.four {
background-color: #bbbbbb;
color: #cccccc;
text-shadow: -1px -2px #666666;}

p.five {
background-color: #aaaaaa;
color: #ffffff;
text-shadow: -1px -1px #666666;}
```

# Text Shadow

The briard is known as a heart wrapped in fur.

The briard is known as a heart wrapped in fur.

The briard is known as a heart wrapped in fur.

The briard is known as a heart wrapped in fur.

The briard is known as a heart wrapped in fur.

# Text Shadow

## text-shadow

**Values:** 'horizontal-offset' 'vertical-offset' 'blur-radius' 'color', none

The value is two offset measurements, an optional blur radius, and a color value (with no commas between).

**Example:**

**The Jenville Show**

```
text-shadow: .2em .2em .1em silver;
```

**The Jenville Show**

```
text-shadow: .2em .2em .3em silver;
```

# First Letter or Line

- You can specify different values for the first letter or first line of text inside an element using **:first-letter** and **:first-line**.
- Technically these are not properties. They are known as **pseudo-elements**.
- You specify the pseudo-element at the end of the selector, and then specify the declarations as you would normally for any other element.

```
p.intro:first-letter {
 font-size: 200%;}

p.intro:first-line {
 font-weight: bold;}
```

The briard, or berger de brie, is a large breed of dog traditionally used as a herder and guardian of sheep.

# Links

- Browsers tend to show links in blue with an underline by default, and they will change the color of links that have been visited to help users know which pages they have been to.
- In CSS, there are two **pseudo-classes** that allow you to set different styles for links that have and have not yet been visited.
- **:link** This allows you to set styles for links that have not yet been visited.
- **:visited** This allows you to set styles for links that have been clicked on.

# Links

```
a:link {
 color: deeppink;
 text-decoration: none;}

a:visited {
 color: black;}
```

## Dog Breeds: B

- Basset Hound
- Beagle
- Bearded Collie
- Beauceron
- Bedlington Terrier
- Belgian Shepherd
- Bergamasco
- Bichon Frise
- Bloodhound
- Bolognese
- Border Collie
- ~~B~~uder Terrier
- Borzoi
- Bouvier des Flandres
- Briard
- Bull Terrier
- Bulldog

# Responding to User

- There are three pseudo-classes that allow you to change the appearance of elements when a user is interacting with them.
- **:hover** This is applied when a user hovers over an element with a pointing device such as a mouse. This has commonly been used to change the appearance of links and buttons.
- **:active** This is applied when an element is being activated by a user. when a button is being pressed or a link being clicked.
- **:focus** This is applied when an element has focus ( ready for tying or u focus using tab .

# Responding to User

```
input {
 padding: 6px 12px 6px 12px;
 border: 1px solid #665544;
 color: #ffffff;}

input.submit:hover {
 background-color: #665544;}

input.submit:active {
 background-color: chocolate;}

input.text {
 color: #cccccc;}

input.text:focus {
 color: #665544;}
```



# Section Summary

- There are properties to control the choice of font, size, weight, style, and spacing.
- There is a limited choice of fonts that you can assume most people will have installed.
- If you want to use a wider range of typefaces there are several options, but you need to have the right license to use them.
- You can control the space between lines of text, individual letters, and words. Text can also be aligned to the left, right, center, or justified. It can also be indented.
- You can use pseudo-classes to change the style of an element when a user hovers over or clicks on text, or when they have visited a link.

```
<!DOCTYPE html>
<html> <head> <title>Text</title>
<style type="text/css">
body {padding: 20px;}
h1, h2, h3, a {font-weight: normal;
color: #0088dd; margin: 0px;}
h1 {font-family: Georgia, Times, serif;
font-size: 250%; text-shadow: 2px 2px 3px
#666666; padding-bottom: 10px;}
h2 {font-family: "Gill Sans", Arial,
sans-serif; font-size: 90%;
text-transform: uppercase; letter-
spacing: 0.2em;}
h3 {font-size: 150%;}
p {font-family: Arial, Verdana, sans-
serif; line-height: 1.4em;
color: #665544;}
p.intro:first-line {font-weight: bold;}
.credits {font-style: italic; text-align:
right;}
a {text-decoration: none;}
a:hover {text-decoration: underline;}
</style> </head>
```

## Example

```
<body>
<h1>Briards</h1>
<h2>A Heart wrapped in fur</h2>
<p class="intro">The <a class="breed"
href="http://en.wikipedia.org/wiki/Briard">
briard, or berger de brie, is a large breed of dog
traditionally used as a
herder and guardian of sheep.</p>
<h3>Breed History</h3>
<p>The briard, which is believed to have originated in
France, has been bred for centuries to herd and to
protect sheep. The breed was used by the French Army as
sentries, messengers and to search for wounded soldiers
because of its fine sense of hearing. Briards were used
in the First World War almost to the point of
extinction. Currently the population of briards is
slowly recovering. Charlemagne, Napoleon, Thomas
Jefferson and Lafayette all owned briards.</p>
<p class="credits">by Ivy Duckett</p>
</body>
</html>
```

# Boxes

# Introduction

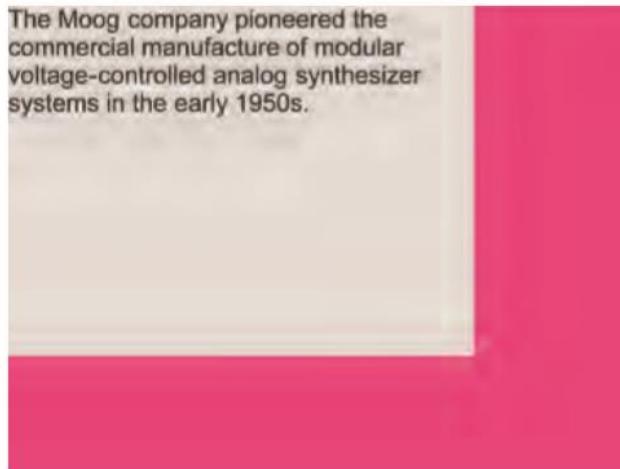
- You saw how CSS treats each HTML element as if it lives in its own box.
- You can set several properties that affect the appearance of these boxes.
  - Control the dimensions of your boxes.
  - Create borders around boxes.
  - Set margins and padding for boxes.
  - Show and hide boxes.

# Box Dimensions

- By default a box is sized just big enough to hold its contents.
- To set your own dimensions for a box you can use the **height** and **width** properties.
- The most popular ways to specify the size of a box are to use **pixels**, **percentages**, or **ems**.
- Traditionally, pixels have been the most popular method because they allow designers to accurately control their size.
- **When you use percentages, the size of the box is relative to the size of the browser window or, if the box is encased within another box, it is a percentage of the size of the containing box.**

# Box Dimensions

```
<div>
 <p>The Moog company pioneered the commercial
 manufacture of modular voltage-controlled
 analog synthesizer systems in the early
 1950s.</p>
</div>
```



```
div.box {
 height: 300px;
 width: 300px;
 background-color: #bbbbbaa;}
p {
 height: 75%;
 width: 75%;
 background-color: #0088dd;}
```

# Limiting Width

- Some page designs expand and shrink to fit the size of the user's screen.
- In such designs, the **min-width** property specifies the smallest size a box can be displayed at when the browser window is narrow, and the **max-width** property indicates the maximum width a box can stretch to when the browser window is wide.

```
td.description {
 min-width: 450px;
 max-width: 650px;
 text-align: left;
 padding: 5px;
 margin: 0px;}
```



The Rhodes piano is an electro-mechanical piano, invented by Harold Rhodes during the fifties and later manufactured in a number of models, first in collaboration with Fender and after 1965 by CBS. It employs a piano-like keyboard with hammers that hit small metal pins, amplified by electromagnetic pickups. \$1400



The Wurlitzer electric piano is an electro-mechanical piano, created by the Rudolph Wurlitzer Company of Mississippi. The Wurlitzer company itself never called the instrument an "electric piano", instead inventing the phrase "Electronic Piano" and using this as a trademark throughout the production of the instrument. It employs a piano-like keyboard with hammers that hit small metal pins, amplified by electromagnetic pickups. \$1600



A Clavinet is an electronically amplified clavichord manufactured by the Hohner company. Each key uses a rubber tip to perform a hammer on a string. Its distinctive bright staccato sound is often compared to that of an electric guitar. Various models were produced over the years, including the models I, II, L, C, D6, and E7. \$1200

# Limiting Height

- In the same way that you might want to limit the width of a box on a page, you may also want to limit the height of it.
- This is achieved using the **min-height** and **max-height** properties.

```
h2, p {
 width: 400px;
 font-size: 90%;
 line-height: 1.2em;}

h2 {
 color: #0088dd;
 border-bottom: 1px solid #0088dd;}

p {
 min-height: 10px;
 max-height: 30px;}
```

## Fender Mustang

The Fender Mustang was introduced in 1964 as the basis of a major redesign of Fender's student models then consisting of the Musicmaster and Duo-Sonic. It was originally popular in sixties surf music and attained cult status in the 1990s largely as a result of its use by a number of alternative rock bands. The Fender Stratocaster or "Strat" is one of the most popular electric guitars of all time, and its design has been copied by many guitar makers. It was designed by Leo Fender, George Fullerton and Freddie Tavares in 1954.

The Gibson Les Paul is a solid body electric guitar that was first sold in 1952. The Les Paul was designed by Ted McCarty in collaboration with popular guitarist Les Paul, whom Gibson enlisted to endorse the new model. It is one of the most well-known electric guitar types in the world.

# Content Overflow

- The **overflow** property tells the browser what to do if the content contained within a box is larger than the box itself.
- It can have one of two values:
  - **hidden** This property simply hides any extra content that does not fit in the box.
  - **scroll** This property adds a scrollbar to the box so that users can scroll to see the missing content.

```
p.one {
 overflow: hidden;}

p.two {
 overflow: scroll;}
```

## Fender Stratocaster

The Fender Stratocaster or "Strat" is one of the most popular electric guitars of all time, and its design has been copied by many guitar makers.

## Gibson Les Paul

The Gibson Les Paul is a solid body electric guitar that was first sold in 1952. The Les Paul was designed by



# Border, Margin, and Padding

- Every box has three available properties that can be adjusted to control its appearance.
  1. **Border**: The border separates the edge of one box from another.
  2. **Margin**: Margins sit outside the edge of the border.
  3. **Padding**: Padding is the space between the border of a box and any content contained within it.



# Border Width

- The **border-width** property is used to control the width of a border.
- The value of this property can either be given in **pixels** or using one of the following values: **thin**, **medium**, **Thick**
- You can control the individual size of borders using four separate properties:
  - **border-top-width**
  - **border-right-width**
  - **border-bottom-width**
  - **border-left-width**
- You can also specify different widths for the four border values in one property:
  - **border-width: 2px 1px 2px 1px;**

# Border Width

```
p.one {
 border-width: 2px;}

p.two {
 border-width: thick;}

p.three {
 border-width: 1px 4px 12px 4px;}
```

Hohner's "Clavinet" is  
essentially an electric  
clavichord.

Hohner's "Clavinet" is  
essentially an electric  
clavichord.

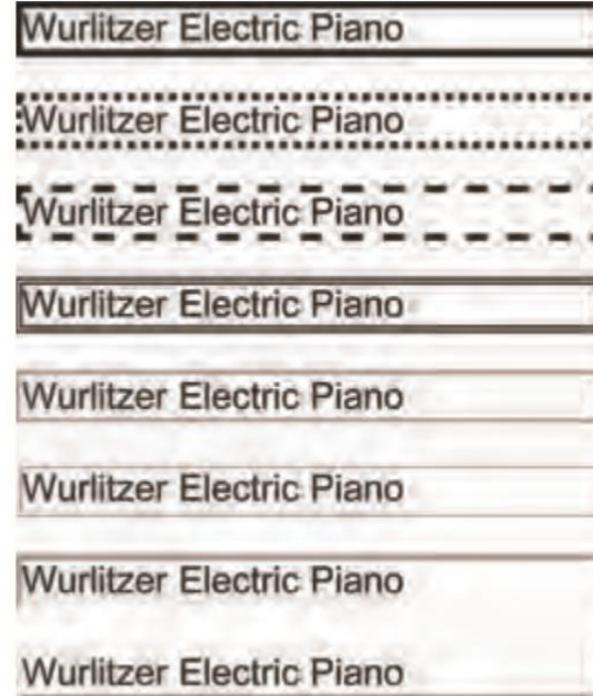
Hohner's "Clavinet" is  
essentially an electric  
clavichord.

# Border Style

- You can control the style of a border using the **border-style** property.
- This property can take the following values:
  - **solid** a single solid line
  - **dotted** a series of square dots
  - **dashed** a series of short lines
  - **double** two solid lines
  - **groove** appears to be carved into the page
  - **ridge** appears to stick out from the page
  - **inset** appears embedded into the page
  - **outset** looks like it is coming out of the screen
  - **hidden / none** no border is shown

# Border Style

```
p.one {border-style: solid;}
p.two {border-style: dotted;}
p.three {border-style: dashed;}
p.four {border-style: double;}
p.five {border-style: groove;}
p.six {border-style: ridge;}
p.seven {border-style: inset;}
p.eight {border-style: outset;}
```



# Border Color

- You can specify the color of a border using either **RGB** values, **hex codes** or CSS color **names**.
- It is possible to individually control the colors of the borders on different sides of a box using.
  - **border-top-color**
  - **border-right-color**
  - **border-bottom-color**
  - **border-left-color**
- It is also possible to use a shorthand to control all four border colors in the one property: **border-color: darkcyan deeppink darkcyan deeppink;** or **border-color: darkcyan;**

# Border Color

```
p.one {
 border-color: #0088dd;}
p.two {
 border-color: #bbbbbaa #111111 #ee3e80 #0088dd;}
```

The ARP Odyssey was introduced in 1972.

The ARP Odyssey was introduced in 1972.

# Shorthand Border

- The **border** property allows you to specify the **width**, **style** and **color** of a border in one property (and the values should be coded in that specific order).

```
p {
 width: 250px;
 border: 3px dotted #0088dd;}
```

Here is a simple chord sequence  
played on a Hammond organ  
through a Leslie speaker.

# Padding

- The **padding** property allows you to specify how much space should appear between the content of an element and its border.
- The value of this property is most often specified in pixels.
- If a width is specified for a box, padding is added onto the width of the box.
- You can specify different values for each side of a box using:
  - **padding-top**
  - **padding-right**
  - **padding-bottom**
  - **padding-left**
- Or you can use a shorthand : **padding: 10px 5px 3px 1px;**

# Padding

```
p {
 width: 275px;
 border: 2px solid #0088dd;}

p.example {
 padding: 10px;}
```

Analog synths produce a wave sound, whereas the sounds stored on a digital synth have been sampled and then turned into numbers.

**Analog synths produce a wave sound, whereas the sounds stored on a digital synth have been sampled and then ...**

Analog synths produce a wave sound, whereas the sounds stored on a digital synth have been sampled and then turned into numbers.

Analog synths produce a wave sound, whereas the sounds stored on a digital synth have been sampled and then turned into numbers.

# Margin

- The **margin** property controls the gap between boxes. Its value is commonly given in pixels.
- You can specify values for each side of a box using:
  - **margin-top**
  - **margin-right**
  - **margin-bottom**
  - **margin-left**
- You can also use the shorthand: **margin: 1px 2px 3px 4px;** or **margin: 10px 20px;** (10 for left and right; 20 for top and bottom )

# Margin

```
p {
 width: 200px;
 border: 2px solid #0088dd;
 padding: 10px;}

p.example {
 margin: 20px;}
```

Analog synthesizers are often said to have a "warmer" sound than their digital counterparts.

Analog synthesizers are often said to have a "warmer" sound than their digital counterparts.

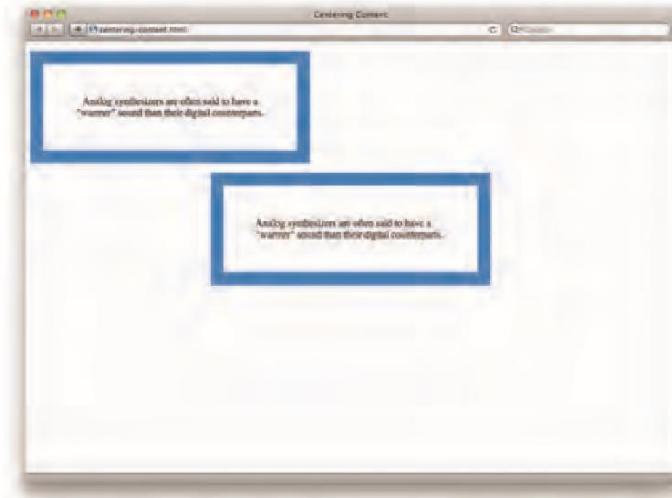
# Centering Content

- If you want to center a box on the page (or center it inside the element that it sits in), you can set the **left-margin** and **right-margin** to **auto**.
- Once you have specified the width of the box, setting the left and right margins to auto will make the browser put an equal gap on each side of the box.

```
body {
 text-align: center;}

p {
 width: 300px;
 padding: 50px;
 border: 20px solid #0088dd;}

p.example {
 margin: 10px auto 10px auto;
 text-align: left;}
```



# Change Inline/Block

- The **display** property allows you to turn an inline element into a block-level element or vice versa, and can also be used to hide an element from the page.
- The values this property can take are:
  - **inline** This causes a block-level element to act like an inline element.
  - **block** This causes an inline element to act like a block-level element.
  - **inline-block** This causes a block-level element to flow like an inline element, while retaining other features of a block-level element. ( can contain block elements )
  - **none** This **hides** an element from the page. In this case, the element acts as though it is not on the page at all (although a user could still see the content of the box if they used the view source option in their browser).

# Change Inline/Block

```
li {
 display: inline;
 margin-right: 10px;}

li.coming-soon {
 display: none;}
```

```

 Home
 Products
 <li class="coming-soon">Services
 About
 Contact

```

Home Products About Contact

# Hide Boxes

- The **visibility property** allows you to hide boxes from users but It leaves a space where the element would have been.
- This property can take two values:
  - **hidden** This hides the element.
  - **visible** This shows the element.
- If the visibility of an element is set to hidden, a blank space will appear in its place.

```
li {
 display: inline;
 margin-right: 10px;}

li.coming-soon {
 visibility: hidden;}
```

Home Products

About Contact

# Box Shadow

- In CSS3, the **box-shadow** property allows you to add a drop shadow around a box.
- It works just like the **text-shadow** property. The value of this property take three lengths and a color for the drop shadow.
- The first length indicates how far to the left or right the shadow should fall.
- The second value indicates the distance to the top or bottom that the shadow should fall.
- The third value is optional and specifies the amount of blur that should be applied to the drop shadow. ( if fourth value exist : it the spread )
- The last value is the color of the drop shadow.

```
box-shadow: -5px -5px 5px #777777;
```

# Box Shadow

```
p.one {
 -moz-box-shadow: -5px -5px #777777;
 -webkit-box-shadow: -5px -5px #777777;
 box-shadow: -5px -5px #777777;}

p.two {
 -moz-box-shadow: 5px 5px 5px #777777;
 -webkit-box-shadow: 5px 5px 5px #777777;
 box-shadow: 5px 5px 5px #777777;}

p.three {
 -moz-box-shadow: 5px 5px 5px 5px #777777;
 -webkit-box-shadow: 5px 5px 5px 5px #777777;
 box-shadow: 5px 5px 5px 5px #777777;}
```



# Rounded Corners

- CSS3 introduces the ability to create rounded corners on any box, using a property called **border-radius**. The value indicates the size of the radius in **pixels**.
- You can specify individual values for each corner of a box using:
  - border-top-right-radius
  - border-bottom-right-radius
  - border-bottom-left-radius
  - border-top-left-radius
- You can also use a shorthand of these four properties (in clockwise order: top, right, bottom, left). For example: border-radius: 5px, 10px, 5px, 10px;

# Rounded Corners

```
p {
border: 5px solid #cccccc;
padding: 20px;
width: 275px;
text-align: center;
border-radius: 10px;
-moz-border-radius: 10px;
-webkit-border-radius: 10px;}
```

Pet Sounds featured a number of unconventional instruments such as bicycle bells, buzzing organs, harpsichords, flutes, Electro-Theremin, dog whistles, trains, Hawaiian-sounding string instruments, Coca-Cola cans and barking dogs.

# Cursor Style

- The **cursor** property allows you to control the type of mouse cursor that should be displayed to users.
- For example, on a form you might set the cursor to be a hand when the user hovers over it.
- Here are the most commonly used values for this property:
  - auto
  - crosshair
  - default
  - pointer
  - move
  - text
  - wait
  - help
  - url("cursor.gif");

# Cursor Style

```

Walt Whitman
```



```
a {
cursor: move;}
```

# Lists

# List Style Type

- The **list-style-type** property allows you to control the shape or style of a bullet point (also known as a marker).
- It can be used on rules that apply to the **<ol>**, **<ul>**, and **<li>** elements.
- For an unordered list you can use the following values:
  - none, disc, circle, square
- For an ordered (numbered) list you can use the following values:
  - decimal, decimal-leading-zero, lower-alpha, upper-alpha, lower-roman, upper-roman

# List Style Type

```
<h1>The Complete Poems</h1>
<h2>Emily Dickinson</h2>

 Life
 Nature
 Love
 Time and Eternity
 The Single Hound

```

```
ol {
 list-style-type: lower-roman;}
```

# The Complete Poems

## Emily Dickinson

- i. Life
- ii. Nature
- iii. Love
- iv. Time and Eternity
- v. The Single Hound

# List Style Image

- You can specify an image to act as a bullet point using the **list-style-image** property.
- The value starts with the letters **url** and is followed by a pair of parentheses.
- Inside the parentheses, the path to the image is given inside double quotes.
- This property can be used on rules that apply to the **<ul>** and **<li>** elements.

# List Style Image

```
<h1>Index of Translated Poems</h1>
<h2>Arthur Rimbaud</h2>

 Ophelia
 To Music
 A Dream for Winter
 Vowels
 The Drunken Boat

```

```
ul {
 list-style-image: url("images/star.png");
}
li {
 margin: 10px 0px 0px 0px;}
```

## Index of Translated Poems

### Arthur Rimbaud

- ★ Ophelia
- ★ To Music
- ★ A Dream for Winter
- ★ Vowels
- ★ The Drunken Boat

# Layout

# Elements Position

- CSS has the following positioning schemes that allow you to control the layout of a page: **normal flow, relative positioning, and absolute positioning.**
- You specify the positioning scheme using the **position** property in CSS.
- You can also float elements using the **float** property.

# Normal Flow

- In normal flow, each block-level element sits on top of the next one.
- Since this is the default way in which browsers treat HTML elements, you do not need a CSS property to indicate that elements should appear in normal flow, but the syntax would be: **position: static;**

```
<body>
 <h1>The Evolution of the Bicycle</h1>
 <p>In 1817 Baron von Drais invented a walking
 machine that would help him get around the
 royal gardens faster...</p>
</body>
```

# Normal Flow

```
h1 {
 background-color: #efefef;
 padding: 10px;}

p {
 width: 450px;}
```

## The Evolution of the Bicycle

In 1817 Baron von Drais invented a walking machine that would help him get around the royal gardens faster: two same-size in-line wheels, the front one steerable, mounted in a frame upon which you straddled. The device was propelled by pushing your feet against the ground, thus rolling yourself and the device forward in a sort of gliding walk.

The machine became known as the Draisienne (or "hobby horse"). It was made entirely of wood. This enjoyed a short lived popularity as a fad, not being practical for transportation in any other place than a well maintained pathway such as in a park or garden.

The next appearance of a two-wheeled riding machine was in 1865, when pedals were applied directly to the front wheel. This machine was known as the velocipede (meaning "fast foot") as well as the "bone shaker," since its wooden structure combined with the cobblestone roads of the day made for an extremely uncomfortable ride. They also became a fad and indoor riding academies, similar to roller rinks, could be found in large cities.

# Relative Positioning

- Relative positioning moves an element in relation to where it would have been in normal flow (original position) .
- For example, you can move it 10 pixels lower than it would have been in normal flow or 20% to the right.
- You can indicate that an element should be relatively positioned using the **position** property with a value of **relative**.
- You then use the offset properties (**top** or **bottom** and **left** or **right**) to indicate how far to move the element from where it would have been in normal flow.

# Relative Positioning

```
p.example {
 position: relative;
 top: 10px;
 left: 100px;}
```

## The Evolution of the Bicycle

In 1817 Baron von Drais invented a walking machine that would help him get around the royal gardens faster: two same-size in-line wheels, the front one steerable, mounted in a frame upon which you straddled. The device was propelled by pushing your feet against the ground, thus rolling yourself and the device forward in a sort of gliding walk.

The machine became known as the Draisienne (or "hobby horse"). It was made entirely of wood. This enjoyed a short lived popularity as a fad, not being practical for transportation in any other place than a well maintained pathway such as in a park or garden.

The next appearance of a two-wheeled riding machine was in 1865, when pedals were applied directly to the front wheel. This machine was known as the velocipede (meaning "fast foot") as well as the "bone shaker," since its wooden structure combined with the cobblestone roads of the day made for an extremely uncomfortable ride. They also became a fad and indoor riding academies, similar to roller rinks, could be found in large cities.

# Absolute Positioning

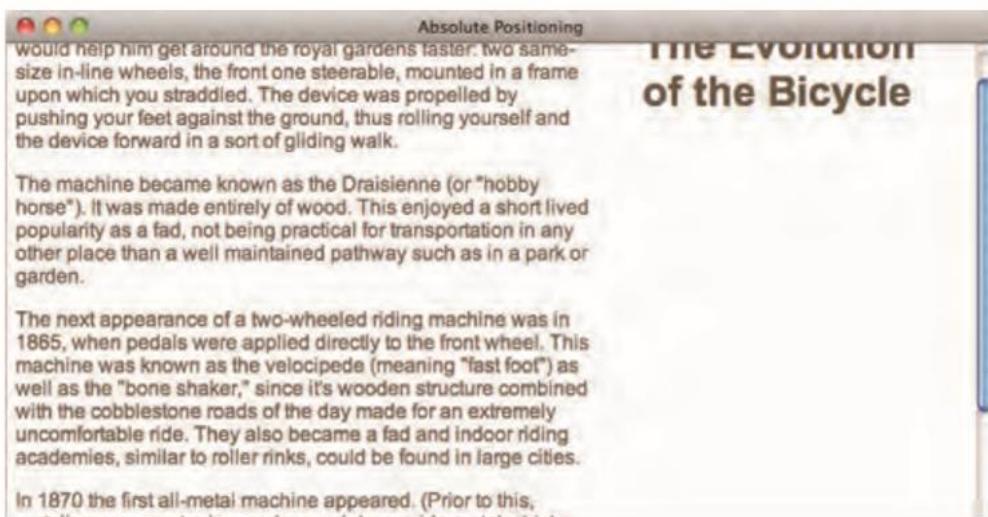
- When the **position** property is given a value of absolute, the box is taken out of normal flow and no longer affects the position of other elements on the page.
- The box offset properties (**top** or **bottom** and **left** or **right**) specify where the element should appear in relation to its **containing element (that has a position)** .

# Absolute Positioning

```
h1 {
 position: absolute;
 top: 0px;
 left: 500px;
 width: 250px;}

p {
 width: 450px;}
```

```
<body>
 <h1>The Evolution of the Bicycle</h1>
 <p>In 1817 Baron von Drais invented a walking
 machine that would help him get around the
 royal gardens faster...</p>
</body>
```



The screenshot shows a web page titled "The Evolution of the Bicycle". The main content area has a heading "Absolute Positioning" and a paragraph about the invention of the Draisienne. Below this, there are two more paragraphs about the evolution of the bicycle. A sidebar on the right contains a blue vertical bar.

Absolute Positioning

would help him get around the royal gardens faster. Two same-size in-line wheels, the front one steerable, mounted in a frame upon which you straddled. The device was propelled by pushing your feet against the ground, thus rolling yourself and the device forward in a sort of gliding walk.

The machine became known as the Draisienne (or "hobby horse"). It was made entirely of wood. This enjoyed a short lived popularity as a fad, not being practical for transportation in any other place than a well maintained pathway such as in a park or garden.

The next appearance of a two-wheeled riding machine was in 1865, when pedals were applied directly to the front wheel. This machine was known as the velocipede (meaning "fast foot") as well as the "bone shaker," since its wooden structure combined with the cobblestone roads of the day made for an extremely uncomfortable ride. They also became a fad and indoor riding academies, similar to roller rinks, could be found in large cities.

In 1870 the first all-metal machine appeared. (Prior to this,

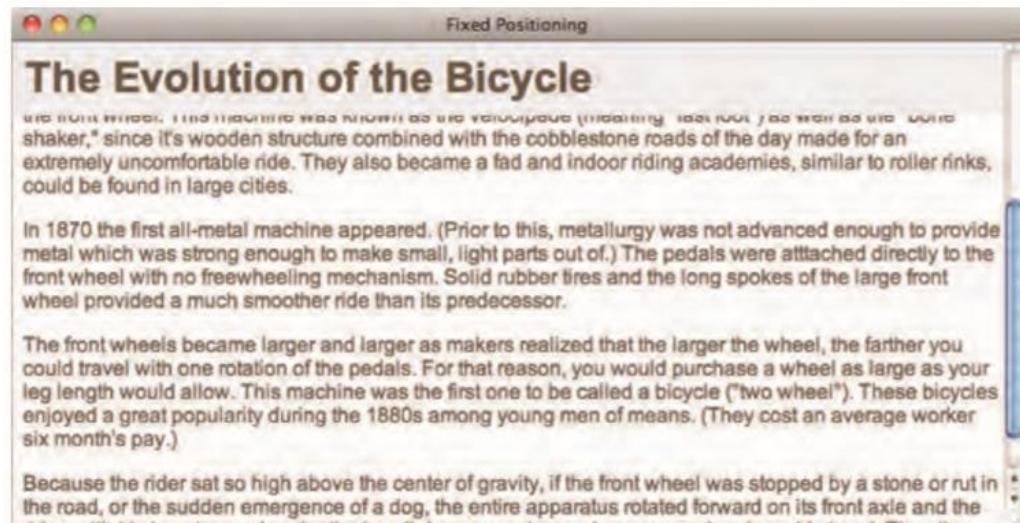
# Fixed Positioning

- Fixed positioning is a type of absolute positioning that requires the position property to have a value of fixed.
- It positions the element in relation to the **browser window**. Therefore, when a user scrolls down the page, it stays in the exact same place. It is a good idea to try this example in your browser to see the effect.
- To control where the fixed position box appears in relation to the browser window, the box offset properties are used.

# Fixed Positioning

```
h1 {
 position: fixed;
 top: 0px;
 left: 50px;
 padding: 10px;
 margin: 0px;
 width: 100%;
 background-color: #efefef;}

p.example {
 margin-top: 100px;}
```



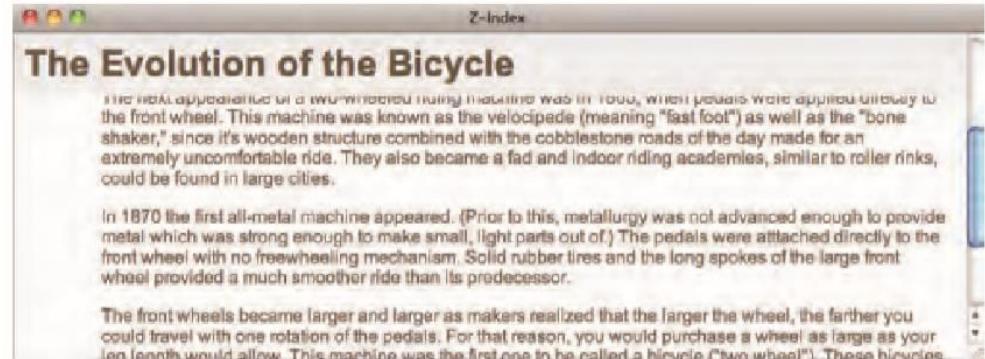
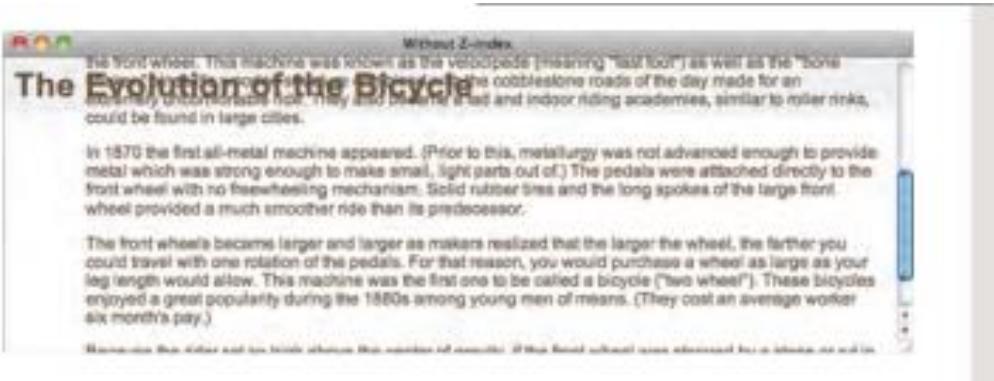
# Overlapping Elements

- When you use relative, fixed, or absolute positioning, boxes can overlap.
- If boxes do overlap, the elements that appear later in the HTML code sit on top of those that are earlier in the page.
- If you want to control which element sits on top, you can use the **z-index** property.
- Its value is a number, and the higher the number the closer that element is to the front.

# Overlapping Elements

```
h1 {
 position: fixed;
 top: 0px;
 left: 0px;
 margin: 0px;
 padding: 10px;
 width: 100%;
 background-color: #efefef;
 z-index: 10;}

p {
 position: relative;
 top: 70px;
 left: 70px;}
```



# Floating Elements

- The float property allows you to take an element in normal flow and place it as far to the left or right of the containing element as possible.
- Anything else that sits inside the containing element will flow around the element that is floated.
- The **float** property is used for positioning and formatting content.
- The float property can have one of the following values:
  - **left** - The element floats to the left of its container.
  - **right** - The element floats to the right of its container.
  - **none** - The element does not float (will be displayed just where it occurs in the text).
  - **inherit** - The element inherits the float value of its parent.

# Floating Elements

<h1>The Evolution of the Bicycle</h1>

<blockquote>"Life is like riding a bicycle.  
To keep your balance you must keep moving." -  
Albert Einstein</blockquote>

<p>In 1817 Baron von Drais invented a walking  
machine that would help him get around the royal  
gardens faster: two same-size in-line wheels, the  
front one steerable, mounted in a frame ... </p>

```
blockquote {
 float: right;
 width: 275px;
 font-size: 130%;
 font-style: italic;
 font-family: Georgia, Times, serif;
 margin: 0px 0px 10px 10px;
 padding: 10px;
 border-top: 1px solid #665544;
 border-bottom: 1px solid #665544;}
```

## The Evolution of the Bicycle

In 1817 Baron von Drais invented a walking machine that would help him get around the royal gardens faster: two same-size in-line wheels, the front one steerable, mounted in a frame upon which you straddled. The device was propelled by pushing your feet against the ground, thus rolling yourself and the device forward in a sort of gliding walk.

The machine became known as the Draisienne (or "hobby horse"). It was made entirely of wood. This enjoyed a short lived popularity as a fad, not being practical for transportation in any other place than a well maintained pathway such as in a park or garden.

The next appearance of a two-wheeled riding machine was in 1865, when pedals were applied directly to the front wheel. This machine was known as the velocipede (meaning "fast foot") as well as the "bone shaker," since its wooden structure combined with the cobblestone roads of the day made for an extremely uncomfortable ride. They also became a fad and indoor riding academies, similar to roller rinks, could be found in large cities.

---

"Life is like riding a bicycle.  
To keep your balance you  
must keep moving." - Albert  
Einstein

---

# Side-By-Side Elements

- A lot of layouts place boxes next to each other.
- The **float** property is commonly used to achieve this.
- When elements are floated, the height of the boxes can affect where the following elements sit.

```
p {
 width: 230px;
 float: left;
 margin: 5px;
 padding: 5px;
 background-color: #efefef;}
```

## The Evolution of the Bicycle

RESU

In 1817 Baron von Drais invented a walking machine that would help him get around the royal gardens faster.

The device known as the Draisienne (or "hobby horse") was made of wood, and propelled by pushing your feet on the ground in a gliding movement.

It was not seen a suitable for any place other than a well maintained pathway.

In 1870 the first all-metal machine appeared. The pedals were attached directly to the front wheel.

Solid rubber tires and the long spokes of the large front wheel provided a much smoother ride than its predecessor.

In 1865, the velocipede (meaning "fast foot") attached pedals to the front wheel, but its wooden structure made it extremely uncomfortable.

# Clearing Floats

- The **clear** property allows you to say that no element (within the same containing element) should touch the left or right hand sides of a box.
- It can take the following values:
  - **none** - Allows floating elements on both sides. This is default
  - **left** - No floating elements allowed on the left side
  - **right** - No floating elements allowed on the right side
  - **both** - No floating elements allowed on either the left or the right side

# Clearing Floats

```
p {
width: 230px;
float: left;
margin: 5px;
padding: 5px;
background-color: #efefef;}
.clear {
clear: left;}
```

## The Evolution of the Bicycle

In 1817 Baron von Drais invented a walking machine that would help him get around the royal gardens faster.

The device known as the Draisienne (or "hobby horse") was made of wood, and propelled by pushing your feet on the ground in a gliding movement.

It was not seen as suitable for any place other than a well maintained pathway.

In 1865, the velocipede (meaning "fast foot") attached pedals to the front wheel, but its wooden structure made it extremely uncomfortable.

In 1870 the first all-metal machine appeared. The pedals were attached directly to the front wheel.

Solid rubber tires and the long spokes of the large front wheel provided a much smoother ride than its predecessor.

# Creating Multi-Column Layout

- Many web pages use multiple columns in their design.
- This is achieved by using a `<div>` element to represent each column.
- The following three CSS properties are used to position the columns next to each other:
  - **width:** This sets the width of the columns.
  - **float:** This positions the columns next to each other.
  - **margin:** This creates a gap between the columns.

# Creating Multi-Column Layout

```
<h1>The Evolution of the Bicycle</h1>
<div class="column1of2">
 <h3>The First Bicycle</h3>
 <p>In 1817 Baron von Drais invented a walking
 machine that would help him get around the
 royal gardens faster: two same-size ...</p>
</div>
<div class="column2of2">
 <h3>Bicycle Timeline</h3> ...
</div>
```

```
.column1of2 {
 float: left;
 width: 620px;
 margin: 10px;}
.column2of2 {
 float: left;
 width: 300px;
 margin: 10px;}
```

# Creating Multi-Column Layout

## The Evolution of the Bicycle

### The First Bicycle

In 1817 Baron von Drais invented a walking machine that would help him get around the royal gardens faster: two same-size in-line wheels, the front one steerable, mounted in a frame upon which you straddled. The device was propelled by pushing your feet against the ground, thus rolling yourself and the device forward in a sort of gliding walk.

The machine became known as the Draisienne (or "hobby horse"). It was made entirely of wood. This enjoyed a short lived popularity as a fad, not being practical for transportation in any other place than a well maintained pathway such as in a park or garden.

### Further Innovations

The next appearance of a two-wheeled riding machine was in 1865, when pedals were applied directly to the front wheel. This machine was known as the velocipede (meaning "fast foot") as well as the "bone shaker," since it's wooden structure combined with the cobblestone roads of the day made for an extremely uncomfortable ride. They also became a fad and indoor riding academies, similar to roller rinks, could be found in large cities.

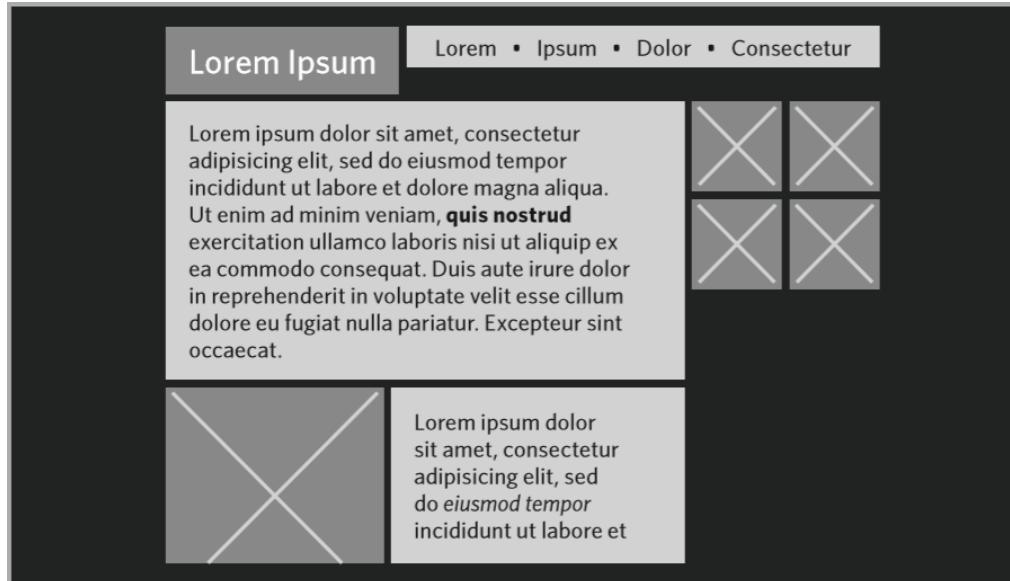
In 1870 the first all-metal machine appeared. (Prior to this, metallurgy was not advanced enough to provide metal which was strong enough to make small, light parts out of.) The pedals were attached directly to the front wheel with no freewheeling mechanism. Solid rubber tires and the long spokes of the large front wheel provided a much smoother ride than its predecessor.

### Bicycle Timeline

- 1817: Draisienne
- 1865: Velocipede
- 1870: High-wheel bicycle
- 1876: High-wheel safety
- 1885: Hard-tired safety
- 1888: Pneumatic safety

# Fixed Width Layouts

- Fixed width layout designs do not change size as the user increases or decreases the size of their browser window.
- Measurements tend to be given in pixels.



# Fixed Width Layouts

## ● Advantages

- Pixel values are accurate at controlling size and positioning of elements.
- The designer has far greater control over the appearance and position of items on the page than with liquid layouts.
- You can control the lengths of lines of text regardless of the size of the user's window.
- The size of an image will always remain the same relative to the rest of the page.

## ● Disadvantages

- You can end up with big gaps around the edge of a page.
- If the user's screen is a much higher resolution than the designer's screen, the page can look smaller and text can be harder to read.
- The design works best on devices that have a site or resolution similar to that of desktop or laptop computers.
- The page will often take up more vertical space than a liquid layout with the same content.

# Fixed Width Layouts

```
<body>
 <div id="header">
 <h1>Logo</h1>
 <div id="nav">

 Home
 Products
 Services
 About
 Contact

 </div>
 </div>
 <div id="content">
 <div id="feature">
 <p>Feature</p>
 </div>
 <div class="article column1">
 <p>Column One</p>
 </div>
 <div class="article column2">
 <p>Column Two</p>
 </div>
 <div class="article column3">
 <p>Column Three</p>
 </div>
 </div>
 <div id="footer">
 <p>© Copyright 2011</p>
 </div>
</body>
```

# Fixed Width Layouts

```
body {
 width: 960px;
 margin: 0 auto;}

#content {
 overflow: auto;
 height: 100%;}

#nav, #feature, #footer {
 background-color: #efefef;
 padding: 10px;
 margin: 10px;}

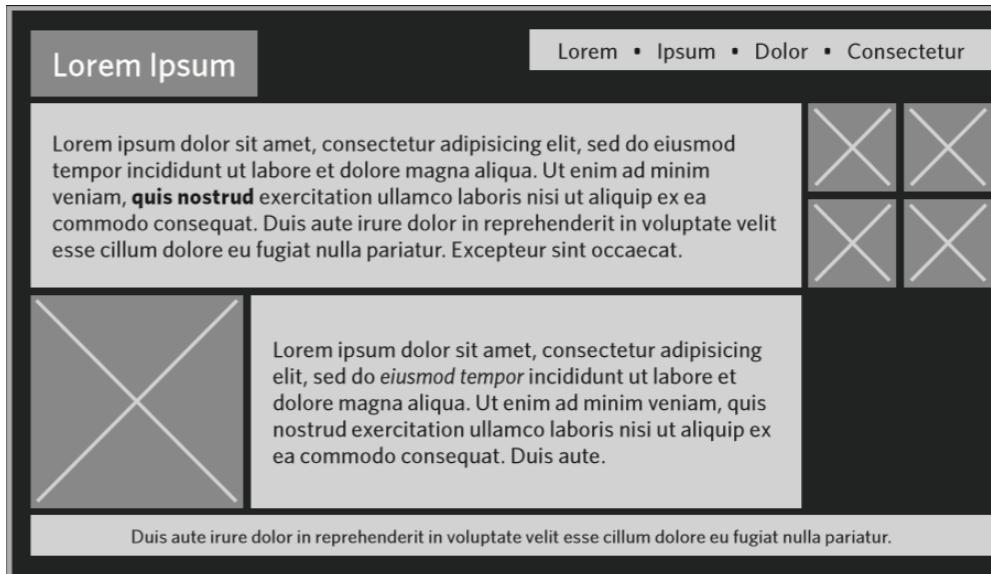
.column1, .column2, .column3 {
 background-color: #efefef;
 width: 300px;
 float: left;
 margin: 10px;}

li {
 display: inline;
 padding: 5px;}
```



# Liquid Layouts

- Liquid layout designs stretch and contract as the user increases or decreases the size of their browser window.
- They tend to use percentages.



# Liquid Layouts

## ● Advantages

- Pages expand to fill the entire browser window so there are no spaces around the page on a large screen.
- If the user has a small window, the page can contract to fit it without the user having to scroll to the side.
- The design is tolerant of users setting font sizes larger than the designer intended (because the page can stretch).

## ● Disadvantages

- If you do not control the width of sections of the page then the design can look very different than you intended, with unexpected gaps around certain elements or items squashed together.
- If the user has a wide window, lines of text can become very long, which makes them harder to read. Also, if the user has a very narrow window, words may be squashed and you can end up with few words on each line.
- If a fixed width item (such as an image) is in a box that is too small to hold it (because the user has made the window smaller) the image can overflow over the text.

# Liquid Layouts

```
<body>
 <div id="header">
 <h1>Logo</h1>
 <div id="nav">

 Home
 Products
 Services
 About
 Contact

 </div>
 </div>
 <div id="content">
 <div id="feature">
 <p>Feature</p>
 </div>
 <div class="article column1">
 <p>Column One</p>
 </div>
 <div class="article column2">
 <p>Column Two</p>
 </div>
 <div class="article column3">
 <p>Column Three</p>
 </div>
 </div>
 <div id="footer">
 <p>© Copyright 2011</p>
 </div>
</body>
```

# Liquid Layouts

```
body {
 width: 90%;
 margin: 0 auto;}
#content {overflow: auto;}
#nav, #feature, #footer {
 margin: 1%;}
.column1, .column2, .column3 {
 width: 31.3%;
 float: left;
 margin: 1%;}
.column3 {margin-right: 0%;}
```

```
li {
 display: inline;
 padding: 0.5em;}
#nav, #footer {
 background-color: #efefef;
 padding: 0.5em 0;}
#feature, .article {
 height: 10em;
 margin-bottom: 1em;
 background-color: #efefef;}
```



