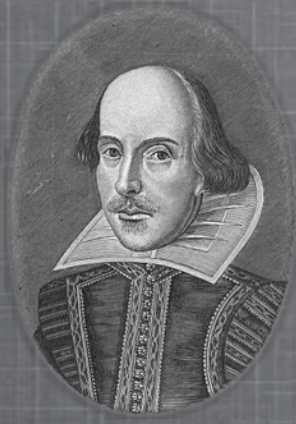


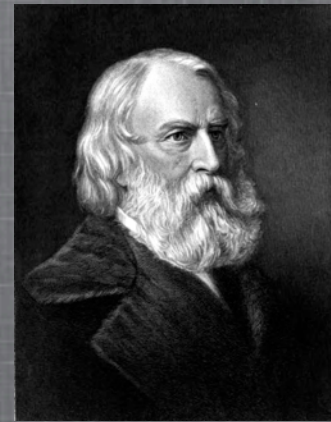
CSS

# HTML: Looking Back

- HTML dictates order, structure, and function
- Does very little to specify layout or visual rendering



Shakespeare



Longfellow

CSS

Let's Write Some Poetry!



YOU

C  
S tyle  
S

# The Purpose of CSS

- If HTML is the content and meaning
  - ➔ CSS helps to convey that meaning
- Allows developers to separate the content from layout and design
  - Content and design inherently different in nature
    - ➔ Change in content does not require change in design

# CSS Zen Garden

- Site using consistent HTML content
- Differing external CSS files create dramatically different layout
- Support for multiple browsers

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

hint: change the styles on the page

# An Example

Consider the **boldface** font in the following examples:

**Chunky** bacon is delicious.

- Meaning is conveyed by the styling
- Remove the style and meaning is lost

Today I will go outside.  
**Monday** I will run 2 miles.

- Attention is given to the information
- No additional meaning is lost when removed

# A Bit of Reasoning

`<strong>`

➔ **Hola.**

`<em>`

➔ *Hola!*

- Explicitly conveys meaning

`<b>`

➔ **Hola.**

`<i>`

➔ *Hola!*

- Typographical artifacts: any meaning is implied



# In Detail

Today I will go outside.  
Monday I will run 2 miles.

- Special attention is given to the date
- Assists the user in resolving, parsing the information
  - ➔ Key goal of Design

# CSS: How to Use It

1. Select the elements you want to modify

➡ **Selectors**

2. Indicate which aspect(s) of the element you want to modify

➡ **Assign values to properties**

# CSS Anatomy

body {  
background-color: #FFFFFF;  
}

Selector

Property

Value

This also works:

```
body { background-color: #FFFFFF }
```

Example of hex color codes:

[ColorSchemer Online Color Generator](#)

C  
S tyle  
S heet

# Stylesheet: A Sheet of Styles

- Place each style block together to make a stylesheet!

# Selectors

- Select an **element**

  - ➔ `body {} p {} strong {} div {}`

- Select a **nested element**

  - ➔ `p strong {} div p {} ul li ul li span {}`

- Select **multiple elements**

  - ➔ `p, div {} strong, em {} p, span {}`

# More Selectors

- Select an element with a **class**

➔ `p.indent { }`   `span.blue { }`   `div.page { }`

- Select an element with an **id**

➔ `div#home { }`   `div p#tools { }`

- Select an element with an **id** *and* a **class**

➔ `ul.square#top { }`   `div.page table.math#pset { }`

# About Properties

- Properties can be specific to a group of elements
  - ex: border-spacing is useless for <div> elements
- *Some* property values can be inherited by children
  - ex: font-size set on <body> will be inherited by a child <div> element
- Properties can have default values
  - ex: background-color's default value is transparent



# How do you know?

... Or find out?



- Luckily, the W3C produces a specification for CSS, too
  - Lists all properties, defaults, possible values
  - Also suggests how property values affect rendering
- Many commercial sites also provide similar references
  - ex: SitePoint, w3schools (both have HTML & CSS)

# Key Properties

- background-color
- background-image
- color
- width
- height
- font-family
- font-size
- font-weight
- text-decoration
- text-align

# Values & Units

- Values are typically keywords
  - ex: colors: red, blue
  - ex: text alignment: left, right, center
- Values, especially for layout
  - Pixels: 15px
  - Points: 12pt
  - Percentages- relative to size of parent: 50%

# An Example

Consider the **boldface** font in the following examples:

Chunky bacon is delicious.

- Use the `<strong>` tag

Today I will go outside.  
Monday I will run 2 miles.

- use `font-weight: bold;`

# Firebug: Your New Best Friend



**Firebug**

[Get Firebug](#)

- Firefox extension for “inspecting” page elements
  - Useful tool when “something doesn’t look right”
- Provides interactive view of HTML source
- Allows for real-time changes in HTML and CSS
- ➔ The Web is inherently Open Source!