
Cascading Style Sheets

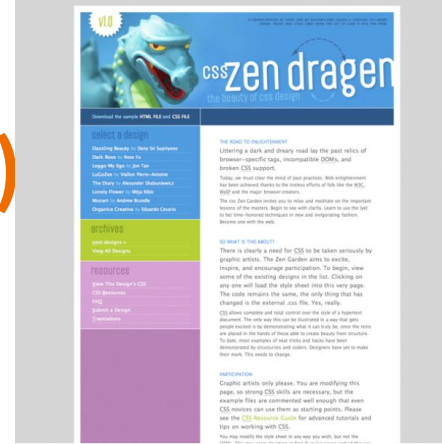
Learning Outcomes

After completing this lesson, you should be able to

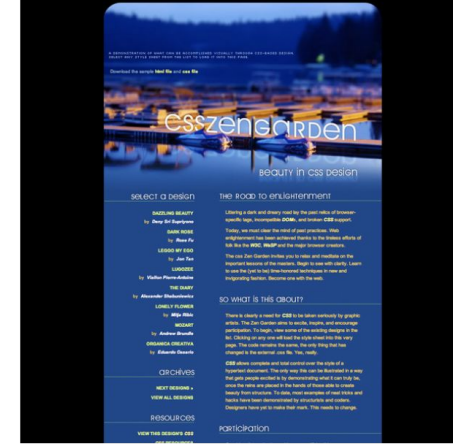
- Explain what Cascading Style Sheet (CSS) is
- Add CSS to HTML document in different ways
- Explain Basic CSS concepts
- Add Text related styles such as font and color

Cascading Style Sheets (CSS)

is the W3C standard for defining the presentation of documents written in HTML



CSS Zen Garden
by Matthew Buchanan



By the Pier
by Peter Ong Kelmscott



Organica Creativa
by Eduardo Cesario



Shaolin Yokobue
by Javier Cabrera

How Stylesheets Work

Start with a document that has been marked up in HTML

Write style rules for how you 'd like certain elements to look

Attach the style rules to the document

Create the page shown in the right

I had the pleasure of spending a crisp, Spring day in Portsmouth, NH cooking and chatting with Daniel Lorca of the band Nada Surf as he prepared a gourmet, sit-down dinner for 28 pals.

When I first invited Nada Surf to be on the show, I was told that Daniel Lorca was the guy I wanted to talk to. Then Daniel emailed his response: "i'm way into it, but i don't want to talk about it, i wanna do it." After years of only having access to touring bands between their sound check and set, I've been doing a lot of *talking* about cooking with rockstars. To actually cook with a band was a dream come true.

A photograph of a table set with various dishes. In the background, there is a large bowl of red sauce, a bowl of white rice, and a bowl of green vegetables. In the foreground, there are several plates of green vegetables and red sauce. The table is covered with a brown cloth.

Daniel prepared a salad of arugula, smoked tomatoes, tomato jam, and grilled avocado (it's as good as it sounds!). I jokingly called it "6-hour Salad" because that's how long he worked on it. The fresh tomatoes were slowly smoked over woodchips in the grill, and when they were softened, Daniel separated out the seeds which he reduced into a smoky jam. The tomatoes were cut into strips to put on the salads. As the day meandered, the avocados finally went on the grill after dark. I was on flashlight duty while Daniel checked for the perfect grill marks.

I wrote up a streamlined adaptation of his recipe that requires *much* less time and serves 6 people instead of *fivetimes* that amount.

In addition to the smoky grilled salad, Daniel served tarragon cornish hens with a cognac cream sauce loaded with chanterelles and grapes, and wild rice with grilled ramps (wild garlicky leeks). Dinner was served close to midnight, but it was a party so nobody cared.

We left that night (technically, early the next morning) with full bellies, new cooking tips, and nearly 5 hours of footage. I'm considering renaming the show "Cooking with Nada Surf".

Writing CSS Rules

A **stylesheet** is made up of one or more style instructions (called style rules) that describe how an element or group of elements should be displayed

Example Rules

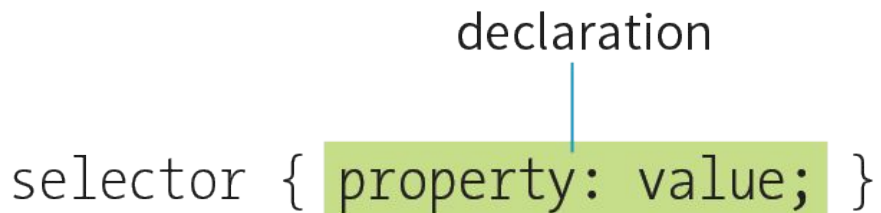
```
h1 {  
    color: green;  
}  
  
p {  
    font-size: large;  
    font-family: sans-serif;  
}
```

Writing CSS Rules

CSS Rule Syntax

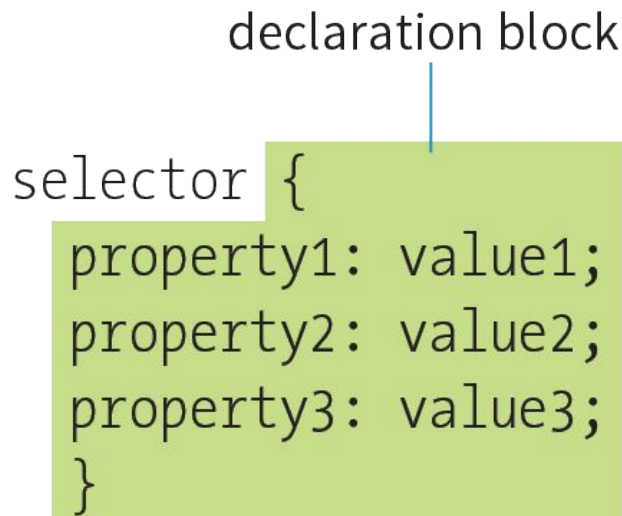
declaration

selector { property: value; }

A diagram showing a CSS rule. The text "selector { property: value; }" is displayed. A blue vertical line points from the word "declaration" above to the "property: value;" part, which is highlighted with a light green background.

declaration block

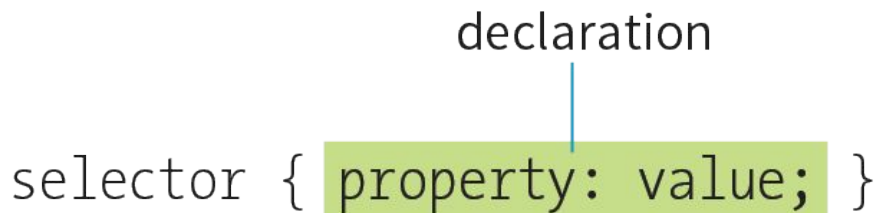
selector {
property1: value1;
property2: value2;
property3: value3;
}

A diagram showing a CSS rule with a multi-line declaration block. The text "selector {" is on the left, and a light green box contains the following lines: "property1: value1;", "property2: value2;", "property3: value3;", and "}". A blue vertical line points from the word "declaration block" above to the start of the green box.

Writing CSS Rules

The **selector** identifies the element or elements to be affected

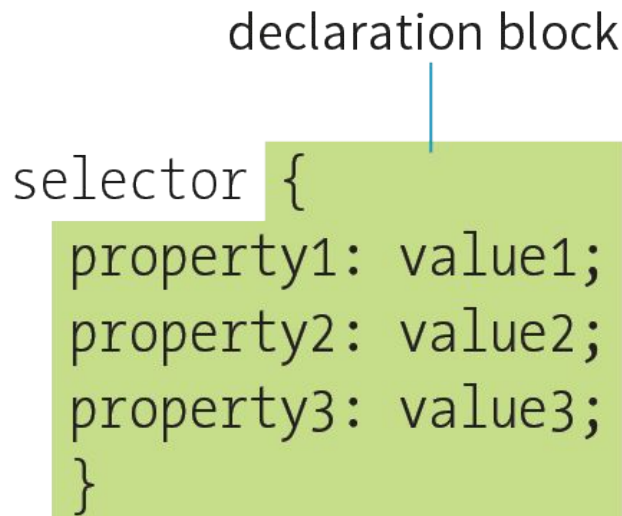
declaration



A diagram showing a CSS rule. The text "selector { property: value; }" is displayed. A blue vertical line points from the word "declaration" above to the "property: value;" part, which is highlighted with a light green background.

```
selector { property: value; }
```

declaration block



A diagram showing a CSS rule. The text "selector { property1: value1; property2: value2; property3: value3; }" is displayed. A blue vertical line points from the words "declaration block" above to the opening curly brace of the block, which is highlighted with a light green background.

```
selector {  
  property1: value1;  
  property2: value2;  
  property3: value3;  
}
```


Writing CSS Rules

The **declaration** provides the rendering instructions

declaration

selector { property: value; }

A diagram illustrating a single CSS declaration. The text "selector { property: value; }" is shown. A blue vertical line points from the word "declaration" above to the "property: value;" part, which is highlighted with a light green background.

declaration block

selector {
property1: value1;
property2: value2;
property3: value3;
}

A diagram illustrating a CSS declaration block. The text "selector {" is followed by a light green highlighted block containing three lines of declarations: "property1: value1;", "property2: value2;", and "property3: value3;". A blue vertical line points from the word "declaration block" above to the start of the green block.

Selectors

Element type selector	Universal selector (*)
Grouped selectors	Pseudo-class selectors
Descendent selectors	Pseudo-element selectors
ID and class selectors	Attribute selectors
Child, next-sibling, and following-sibling selectors	

Selectors

h1 and **p** are element selectors

```
h1 {  
    color: green;  
}  
  
p {  
    font-size: large;  
    font-family: sans-serif;  
}
```

Declarations

The declaration is made up of a **property/value** pair

Each declaration must end with a semicolon to keep it separate from the following declaration

```
h1 {  
    color: green;  
}  
  
p {  
    font-size: large;  
    font-family: sans-serif;  
}
```

Exercise

In the html document you have created, insert the following code in the `<head></head>` element

```
<style>
  h1 {
    color: green;
  }

  p {
    font-size: large;
    font-family: sans-serif;
  }
</style>
```

Grouped selectors

You can apply same style rule to multiple elements

```
p, ul, td, th { color: navy; }
```

Descendant Selectors

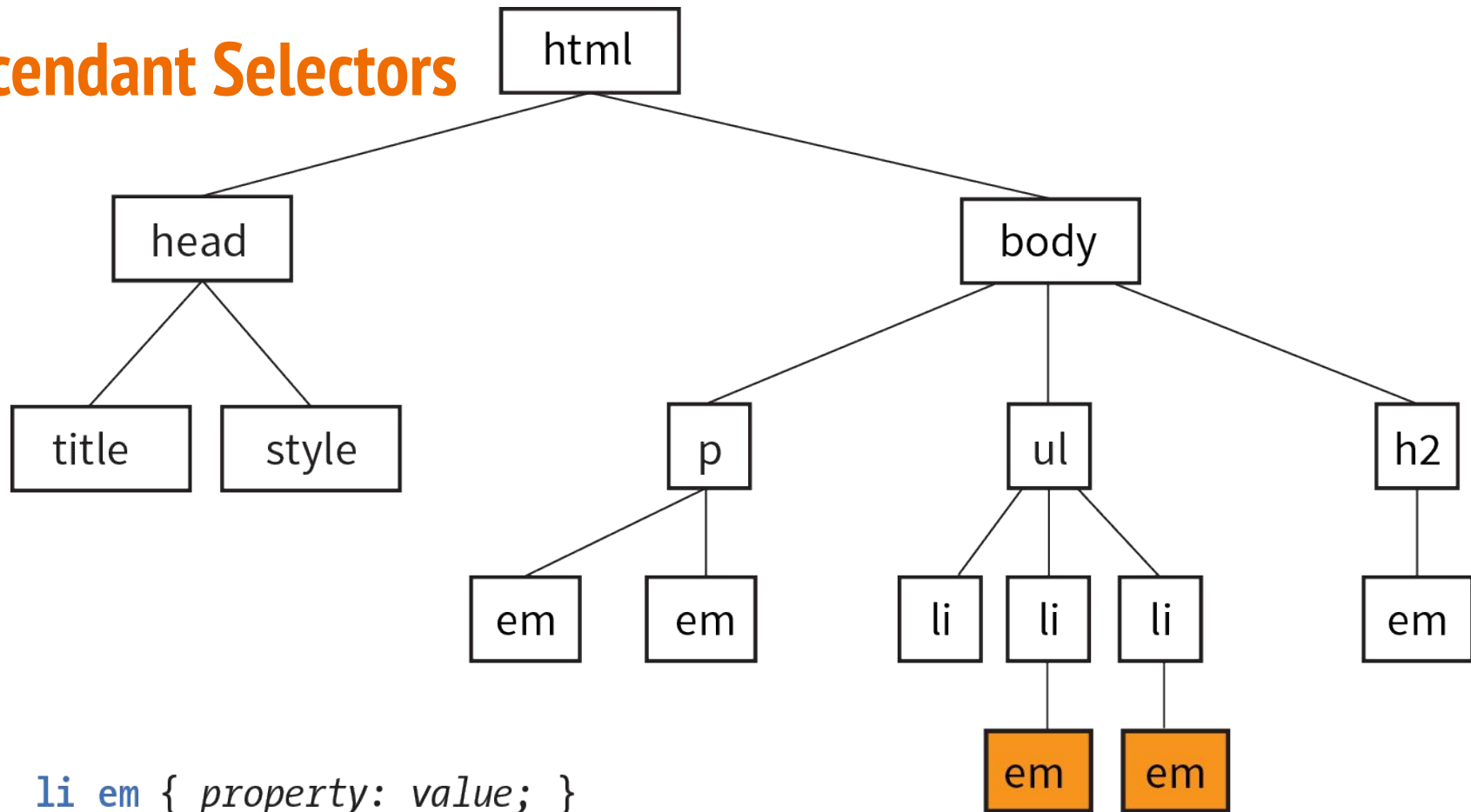
A descendant selector targets elements that are contained within (and therefore are descendants of) another element

Descendant selectors are indicated in a list separated by a character **space**

The following example targets emphasized text (**em**) elements, but only when they appear in list items (**li**)

```
li em {  
  color: olive;  
}
```

Descendant Selectors



`li em { property: value; }`

Child selector

A child selector is similar to a descendant selector, but it targets only the direct children of a given element

They are indicated with the greater-than symbol (>)

The following rule affects emphasized text, but only when it is directly contained in a **p** element

An em element inside a link (**a**) within the paragraph would not be affected

```
p > em {font-weight: bold;}
```

Next-sibling selector

Targets an element that comes directly after another element with the same parent

It is indicated with a plus (+) sign

The following rule gives special treatment to paragraphs that follow an **h1**

```
h1 + p {  
    font-style: italic;  
}
```

ID Selectors

id attribute gives an element a unique identifying name (its id reference)

ID selectors allow you to target elements by their **id** values

The symbol that identifies ID selectors is the octothorpe (**#**), also known as a **hash** or pound symbol

```
<li id="sleestak">Sleestak T-shirt</li>
```

```
li#sleestak { color: olive; }
```

```
#sleestak { color: olive; }
```

Class Selectors

The **class** identifier is used to classify elements into a conceptual group

Unlike the id attribute, **multiple elements may share a class name**

An element may belong to more than one class

You can target elements belonging to the same class with a **class** selector

Class names are indicated with a period (**.**) at the beginning of the selector

Class Selectors

For example, to select all paragraphs with `class="special"`, use the following selector

```
p.special {  
    color: orange;  
}
```

Class Selectors

To apply a property to all elements of the same class, omit the element name in the selector

```
.special {  
  color: orange;  
}
```

The Universal Selector

The universal element selector (*) matches any element

The following style rule puts a 1-pixel gray border around every element in the document

```
* {  
  border: 1px solid gray;  
}
```

Exercise

Change the color of **h1**

Add a new rule that makes the **h2** elements orange

Add a **100-pixel** left margin to paragraph (**p**) elements by using this declaration

```
margin-left: 100px;
```

Add a **100-pixel** left margin to **h2** elements

Exercise

Add an orange, **1-pixel** border to the bottom of the **h1** element by using this declaration

```
border-bottom: 1px solid orange;
```

Move the image to the right margin, and allow text to flow around it with the **float** property

```
img {  
    float: right;  
    margin: 0 10px;  
}
```

Attaching the Styles to the Document

There are three ways that style information can be applied to an HTML document

- Embedded** stylesheets

- Inline** styles

- External** stylesheets

Embedded style sheets

The stylesheet is placed in a document via the **style** element

The rules apply only to the document where the style is embedded

```
<head>
  <title>Required document title here</title>
  <style>
    /* style rules go here */
  </style>
</head>
```

Inline styles

You can apply properties and values to a single element by using the style attribute in the element itself

```
<h1 style="color: red">Introduction</h1>
```

To add multiple properties, just separate them with semicolons

```
<h1 style="color: red; margin-top: 2em">Introduction</h1>
```

Inline styles

Inline styles should be avoided, unless it is absolutely necessary to override styles from an embedded or external style sheet

Inline styles are problematic in that they intersperse presentation information into the structural markup

External style sheets

A separate text-only document that contain style rules

It must be named with the `.css` suffix

This is the most preferred method for attaching stylesheets to content

Exercise

Write an **inline style** that makes the second **h2** gray in the previous html document

```
<h2 style="color: gray">The Main Course</h2>
```

Exercise

Create a file named **styles.css** and take all the style rules from the html document to the **styles.css** file

Attach the styles to the html by including the following element in the **<head></head>** element

```
<link rel="stylesheet" href="styles.css">
```


Main Concepts

Inheritance

Parents and children

Priority

Document structure

The Box Model

Cascade

Specificity

Rule order

CSS Units

Inheritance

Styled HTML elements pass down **certain style properties** to the elements they contain

Inheritance provides a mechanism for styling elements that don't have any explicit styles rules of their own

Exercise

In the previous html document add emphasis to the first word of the second paragraph using `` element

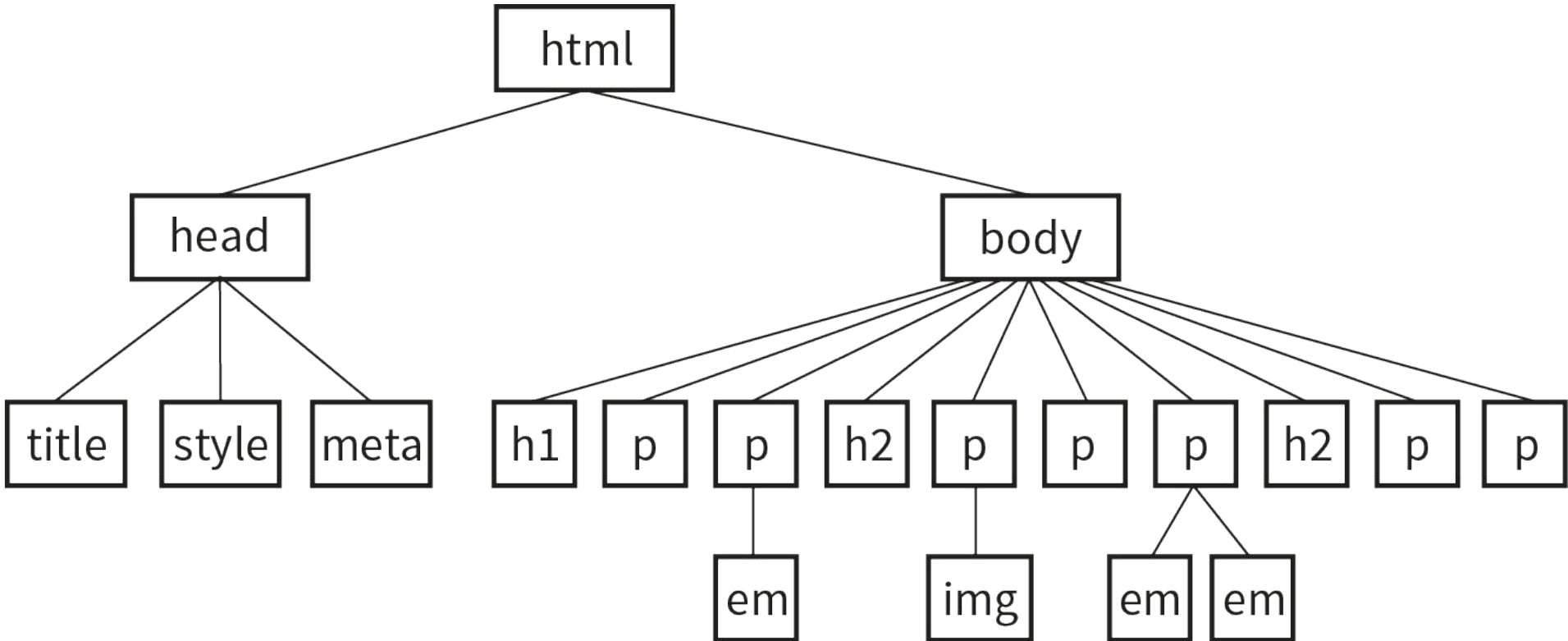
Change the font size of the paragraph and check if the font size of the content inside `` element also changes or not

Check the inherited style rules using **Chrome DevTools**

Document structure

HTML documents have an implicit structure, or hierarchy

Document structure



Parent/Child Relationship

All the elements contained within a given element are said to be its descendants

For example, the **h1**, **h2**, **p**, **em**, and **img** elements in the document tree shown in the previous slide are all **descendants** of the **body** element

Parent/Child Relationship

An element that is directly contained within another element (with no intervening hierarchical levels) is said to be the **child** of that element

Conversely, the containing element is the **parent**

For example,

the **em** element is the **child** of the **p** element, and the **p** element is its **parent**

Parent/Child Relationship

All of the elements higher than a particular element in the hierarchy are its **ancestors**

Two elements with the same parent are **siblings**

Property Inheritance

In general, properties related to the styling of text—**font size, color, style,** and the like—are inherited by child elements

Properties such as **borders, margins, backgrounds,** and so on that affect the boxed area around the element tend not to be passed down

Exercise

On **Chrome DevTools** check that margin property of a paragraph is now inherited by its children

Exercise

If you want to apply similar color to all texts in your document to which element should you add the color style rule in your stylesheet?

Check your answer by adding the style rule to make all your document color to blue

Conflicting Styles: The Cascade

CSS allows you to apply several style sheets to the same document, which means there are bound to be conflicts

The “**cascade**” refers to what happens when several sources of style information compete for control of the elements on a page

CSS uses a hierarchical system that assigns different **weights** to the various sources of style information to handle conflicting rules

Weight is considered based on the **priority** of the style rule source, the **specificity** of the selector, and **rule order**

Style Rule Hierarchy

Any style rule marked **!important** by the **reader** (user)

Individual users can apply their own styles

Any style rule marked **!important** by the **author**

Stylesheets written by the author

Stylesheets created by the reader (user)

Browser's default style rules ("**user agent stylesheet**")

Assigning Importance

If you want a rule not to be overridden by a subsequent conflicting rule, include the `!important` indicator just after the property value and before the semicolon for that rule

```
p {  
    color: blue !important;  
}
```

Specificity

It is possible for conflicts to arise in which an element is getting style instructions from more than one rule

When two rules in a style sheet conflict, the **type of selector** is used to determine the winner

Specificity refers to the fact that more specific selectors have more weight when it comes to handling style rule conflicts

Rule order

Within a style sheet, if there are conflicts within style rules of identical weight, whichever one comes last in the list “wins.”

```
p {  
    color: red;  
}  
  
p {  
    color: blue;  
}  
  
p {  
    color: green;  
}
```


The Box Model

Browsers see every element on the page (both block and inline) as being contained in a little rectangular box

You can apply properties such as borders, margins, padding, and backgrounds to these boxes

Exercise

Use **Chrome DevTools** to check the rectangular box around each element

CSS Units

CSS3 provides a variety of units of measurement

They fall into two broad categories: **absolute** and **relative**

CSS Units

Absolute units have predefined meanings or real-world equivalents

With the exception of pixels, they are not appropriate for web pages that appear on screens

Relative units are **based on the size of something else**, such as the default text size or the size of the parent element

Child elements do not inherit the relative values of their parent, but rather the resulting calculated value

ABsolute Units

px

pixel, defined as equal to 1/96 of an inch in CSS3

in

inches

mm

millimeters

Relative units

em

a unit of measurement equal to the current font size

ex

x-height, approximately the height of a lowercase “x” in the font

rem

root em, equal to the em size of the root element (html)

Relative units

ch

zero width, equal to the width of a zero (0) in the current font and size

vw

viewport width unit, equal to 1/100 of the current viewport (browser window) width

vh

viewport height unit, equal to 1/100 of the current viewport height

Relative units

vmin

viewport **min**imum unit, equal to the value of **vw** or **vh**, whichever is smaller

vmax

viewport **max**imum unit, equal to the value of **vw** or **vh**, whichever is larger

CSS Units: Percentages

Percentages are another common measurement value for web page elements

Percentages are calculated relative to another value, such as the value of a property applied to the current element or its parent or ancestor

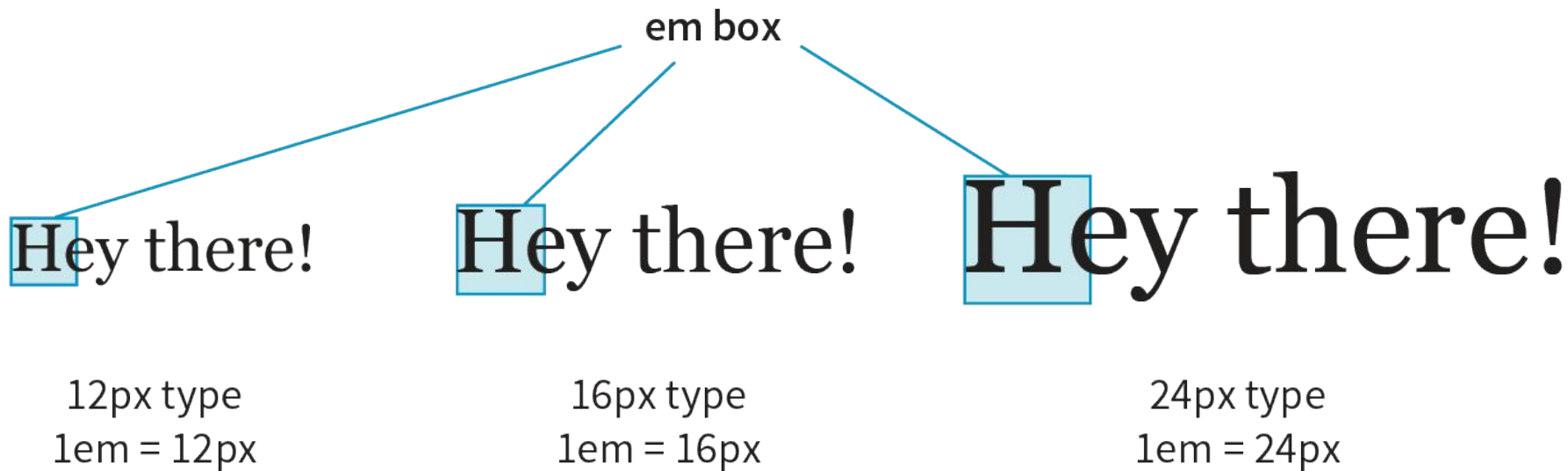
When used for page layouts, percentage values ensure that page elements stay proportional

The **rem** unit

In modern browsers, the default root font size is **16 pixels**; therefore, a rem is equivalent to a 16-pixel unit

The **em** unit

For text with a font size of **16 pixels**, an **em** measures **16 pixels**; for **12-pixel** text, an **em** equals **12 pixels**



The `em` unit

□□ **This is a 24pt Heading**

□□ **A Heading in 20pt**

□□ Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam facilisis imperdiet pretium. Proin fermentum urna sed arcu efficitur tincidunt. Donec id libero euismod, venenatis augue in, vestibulum lectus. Donec ultricies finibus eleifend. Aenean egestas augue sem, vitae ultricies libero fringilla a. Aliquam at tellus purus. Donec accumsan metus sit amet leo volutpat pellentesque.

```
h1, h2, p { margin-left: 2em; }
```

Viewport percentage lengths (vw/vh)

The viewport width (**vw**) and viewport height (**vh**) units are relative to the size of the viewport (browser window)

A **vw** is equal to **1/100** the width of the viewport

A **vh** is equal to **1/100** the height of the viewport

Viewport-based units are useful for making images and text elements stay the full width or height of the viewport

Viewport percentage lengths (vw/vh)

```
header {  
  width: 100vw;  
  height: 100vh;  
}
```

```
img {  
  width: 50vw;  
  height: 50vh;  
}
```

Formating Text

The slide features a light blue background with a white central rectangle. The title "Formating Text" is centered in a bold, orange font. The design is accented with horizontal lines: two teal lines at the top, two teal lines at the bottom, and two thin blue lines just above and below the title. Additionally, there are two short, thick, olive-green bars positioned symmetrically on either side of the title, aligned with its vertical center.

Basic Font Properties

In CSS, fonts are specified using a set of font-related properties for **typeface**, **size**, **weight**, **font style**, and **special characters**

There are also **shortcut properties** that let you **specify multiple font attributes in a single rule**

Font Properties

`font-family`

`font-size`

`font-weight`

`font-style`

`font-variant`

`font`

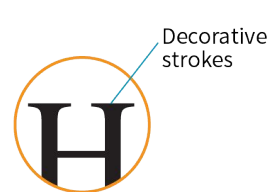
Typeface (Font Family)

font-family

```
p { font-family: "Duru Sans", Verdana, sans-serif; }
```

Generic font families

serif



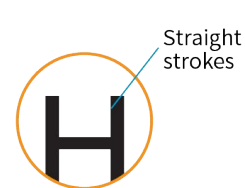
Hello
Times

Hello
Georgia

Hello
Times New Roman

Hello
Lucida

sans-serif



Hello
Verdana

Hello
Trebuchet MS

Hello
Arial

Hello
Arial Black

monospace

W i

Monospace font
(equal widths)

W i

Proportional font
(different widths)

Hello
Courier

Hello
Courier New

Hello
Andale Mono

cursive

Hello
Apple Chancery

Hello
Comic Sans

Hello
Snell

fantasy

Hello
Impact

HELLO
Stencil

HELLO
Mojo

Generic font families

Serif

Examples: Times New Roman, Georgia

Serif typefaces have decorative slab-like appendages (serifs) on the ends of certain letter strokes

Generic font families

sans-serif

Examples: Arial, Verdana, Trebuchet MS, Helvetica

Sans-serif typefaces have straight letter strokes that do not end in serifs

Generic font families

monospace

Examples: `Courier New`

Monospace (also called constant width) **typefaces**, all characters take up the same amount of space on a line

Font stack strategies

The best practice for specifying fonts for web pages is to

start with your first choice,

provide some similar alternatives, and then

end with a generic font family

```
body {  
  font-family: Oswald, Univers, Tahoma, Geneva, sans-serif;  
}
```

Exercise

Create HTML
document inside
`menu.html` file for
the page shown

Black Goose Bistro • Summer Menu

Baker's Corner, Seekonk, Massachusetts

Hours: Monday through Thursday: 11 to 9, Friday and Saturday: 11 to midnight

Appetizers

This season, we explore the spicy flavors of the southwest in our appetizer collection.

Black bean purses

Spicy black bean and a blend of mexican cheeses wrapped in sheets of phyllo and baked until golden. \$3.95

Southwestern napoleons with lump crab — **new item!**

Layers of light lump crab meat, bean and corn salsa, and our handmade flour tortillas. \$7.95

Main courses

Big, bold flavors are the name of the game this summer. Allow us to assist you with finding the perfect wine.

Jerk rotisserie chicken with fried plantains — **new item!**

Tender chicken slow-roasted on the rotisserie, flavored with spicy and fragrant jerk sauce and served with fried plantains and fresh mango. **Very spicy.** \$12.95

Shrimp sate kebabs with peanut sauce

Skewers of shrimp marinated in lemongrass, garlic, and fish sauce then grilled to perfection. Served with spicy peanut sauce and jasmine rice. \$12.95

Grilled skirt steak with mushroom fricasee

Flavorful skirt steak marinated in asian flavors grilled as you like it*. Served over a blend of sauteed wild mushrooms with a side of blue cheese mashed potatoes. \$16.95

* We are required to warn you that undercooked food is a health risk.

Exercise

Make main text of the HTML document you have created appear in **Verdana** or some other **sans-serif** font

```
body {  
    font-family: Verdana, sans-serif;  
}
```

Exercise

Use display font called **Marko One** from Google Web Fonts
(www.google.com/webfonts)

```
<head>
  <meta charset="UTF-8">
  <title>CSS Lab</title>
  <link href="http://fonts.googleapis.com/css?family=Marko+One"
rel="stylesheet">
  <link href="css/styles.css" rel="stylesheet" >
</head>
```

Exercise

Write a rule that applies Marko One font to the **h1** element

```
h1 {  
    font-family: "Marko One", Georgia, serif;  
}
```

Specifying Font Size

font-size

Values: length unit | percentage | xx-small | x-small | small | medium | large | x-large | xx-large | smaller | larger

Default: medium

The preferred font-size values are **em**, **rem**, and %

Specifying Font Size

This is an example of the default text size in Verdana.

xx-small | x-small | small | medium | large | x-large | **xx-large**

This is an example of the default text size in Times.

xx-small | x-small | small | medium | large | x-large | **xx-large**

Specifying Font Size

Percentage values

To maintain the browser's default size, set the **font-size** of the root element to **100%**

```
html {  
    font-size: 100%;  
}
```

Specifying Font Size

Percentage values

In the following example, the **h1** inherits the default **16px** size from the **html** element, and applying the **150%** value multiplies that inherited value, resulting in an **h1** that is **24 pixels**:

```
h1 {  
    font-size: 150%; /* 150% of 16 = 24 */  
}
```

Exercise

Set the **font-size** of the body element to **100%**

Set the main heading to be **24 pixels** using **em** or **percentage** units

Make the **h2s** the same size as the **body** text

Font Weight (Boldness)

font-weight

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

Default: normal

An example is shown in the next slide

This is an example of the default text in Verdana.

normal | **bold** | **bolder** | lighter

100 | 200 | 300 | 400 | 500

600 | **700** | **800** | **900**

This is an example of the default text in Times.

normal | **bold** | **bolder** | lighter

100 | 200 | 300 | 400 | 500

600 | **700** | **800** | **900**

Font Style (Italics)

The **font-style** property affects the posture of the text—that is, whether the letter shapes are **vertical (normal)** or **slanted (italic and oblique)**

font-style

Values: `normal` | `italic` | `oblique`

Default: `normal`

Exercise

Make the definition terms `bold` and `italic`

The Shortcut font Property

font

Values: font-style font-weight font-variant font-stretch
font-size/line-height font-family | caption | icon | menu
| message-box | small-caption | status-bar

Default: depends on default value for each property listed

The Shortcut font Property

font

At minimum, the font property must include a font-size value and a font-family value, in that order

```
p {  
  font: 1em sans-serif;  
}
```

The Shortcut font Property

font

At minimum, the font property must include a font-size value and a font-family value, in that order

```
p {  
  font: 1em sans-serif;  
}
```

EXERCISE

Replace all the font properties we've specified for the **h1** element with one declaration with the shorthand font property

```
h1 {  
    font: bold 1.5em "Marko One", Georgia, serif;  
}
```


Changing Text Color

color

Values: color value (name or numeric)

Default: depends on the browser and user's preferences

Changing Text Color

color

```
h1 {  
    color: gray;  
}
```

```
h1 {  
    color: #666666;  
}
```

```
h1 {  
    color: #666;  
}
```

```
h1 {  
    color: rgb(102, 102, 102);  
}
```

Text Line Adjustment

The **line-height** property defines the minimum distance from baseline to baseline in text

line-height

Values: number | length measurement | percentage | normal

Default: normal

Text Line Adjustment

Size of 1em for this text

line-height is set to 2em (twice the text size); the extra space is divided equally above and below the text line, centering it vertically in the line height.

The line-height property defines the minimum distance from baseline to baseline in text.

Baseline A baseline is the imaginary line upon which the bottoms of characters sit. Line height in CSS is similar to leading in traditional typesetting.

```
line-height: 2em;
```

Text Line Adjustment

Example

```
p {  
    line-height: 2;  
}  
  
p {  
    line-height: 2em;  
}  
  
p {  
    line-height: 200%;  
}
```

Indents

text-indent

Values: length measurement | percentage

Default: 0

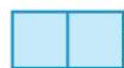
Indents

Example

The effect is shown in the next slide

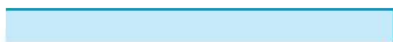
```
p#1 {  
    text-indent: 2em;  
}  
  
p#2 {  
    text-indent: 25%;  
}  
  
p#3 {  
    text-indent: -35px;  
}
```

2em



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.

25%



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.

-35px



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.

Horizontal Text Alignment

text-align

Values: left | right | center | justify | start | end

Default: start

An example is shown in the next slide

`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.

Text Decoration

text-decoration

Values: none | underline | overline | line-through | blink

Default: none

Text Decoration

Examples

I've got laser eyes.

```
text-decoration: underline;
```

I've got laser eyes.

```
text-decoration: overline;
```

~~I've got laser eyes.~~

```
text-decoration: line-through;
```

Text Capitalization

text-transform

Values: none | capitalize | lowercase | uppercase |
full-width

Default: none

Check next slide or example

Text Capitalization

Examples

`text-transform: none;`
(as it was typed in the source)

And I know what you're thinking.

`text-transform: capitalize;`

And I Know What You're Thinking.

`text-transform: lowercase;`

and i know what you're thinking.

`text-transform: uppercase;`

AND I KNOW WHAT YOU'RE THINKING.

Letter Spacing

letter-spacing

Values: length measurement | normal

Default: normal

Letter Spacing

Examples

B l a c k G o o s e B i s t r o S u m m e r M e n u

```
p { letter-spacing: 8px; }
```

Black Goose Bistro Summer Menu

```
p { word-spacing: 1.5em; }
```


List Markers

list-style-type

Values: none | disc | circle | square | decimal |
decimal-leading-zero | lower-alpha | upper-alpha |
lower-latin | upper-latin | lower-roman | upper-roman |
lower-greek

Default: disc

Numbering and Lettering Styles

Keyword	System
<code>decimal</code>	1, 2, 3, 4, 5, ...
<code>decimal-leading-zero</code>	01, 02, 03, 04, 05, ...
<code>lower-alpha</code> or <code>lower-latin</code>	a, b, c, d, ...
<code>upper-alpha</code> or <code>upper-latin</code>	A, B, C, D, ...
<code>lower-roman</code>	i, ii, iii, iv, v, ...
<code>upper-roman</code>	I, II, III, IV, V, ...
<code>lower-greek</code>	α , β , γ , δ , ε , ...

List Item Display Role

Any element can perform like a list item by setting its **display** property to **list-item**

```
p.lettered {  
    display: list-item;  
    list-style-type:  
upper-alpha;  
}
```

List Marker Position

list-style-position

Values: `inside` | `outside` | `hanging`

Default: `outside`

List Marker Position

outside

- **Radish.** Praesent in lacinia risus. Morbi urna ipsum, efficitur id erat pellentesque, tincidunt commodo sem. Phasellus est velit, porttitor vel dignissim vitae, commodo ut urna.
- **Avocado.** Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Curabitur lacinia accumsan est, ut malesuada lorem consectetur eu.
- **Pomegranite.** Nam euismod a ligula ac bibendum. Aenean ac justo eget lorem dapibus aliquet. Vestibulum vitae luctus orci, id tincidunt nunc. In a mauris odio. Duis convallis enim nunc.

inside

- **Radish.** Praesent in lacinia risus. Morbi urna ipsum, efficitur id erat pellentesque, tincidunt commodo sem. Phasellus est velit, porttitor vel dignissim vitae, commodo ut urna.
- **Avocado.** Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Curabitur lacinia accumsan est, ut malesuada lorem consectetur eu.
- **Pomegranite.** Nam euismod a ligula ac bibendum. Aenean ac justo eget lorem dapibus aliquet. Vestibulum vitae luctus orci, id tincidunt nunc. In a mauris odio. Duis convallis enim nunc.

Custom Bullets

`list-style-image`

Values: `url(location) | none`

Default: `none`

Custom Bullet

Example

```
ul {  
  list-style-type: disc;  
  list-style-image: url(/images/custom_bullet.gif);  
  list-style-position: outside;  
}
```