

Internet and Web Principal

Week 6

CSS 1

Content

1. Intro to CSS
2. Formatting Text
3. Colors and Backgrounds

Intro to CSS

Cascading Style Sheets (CSS) is a W3C standard for specifying the display of HTML documents, as well as any XML language. Again, presentation relates to how the material is provided to the user, whether on a computer screen, on a cell phone, on paper, or read aloud by a screen reader. HTML may manage the business of defining document structure and meaning, as intended, with style sheets handling the presentation. CSS is its own language, with its own syntax.

Intro to CSS

CSS'S ADVANTAGES

Here is a summary of the advantages of using style sheets.

- Accurate type and layout controls CSS can be used to produce print-like precision. There is even a set of characteristics designed specifically for the printed page.
- Less effort. By altering a single style sheet, you can modify the design of an entire website. This also guarantees that the formatting is consistent throughout the site.
- More accessible locations. When CSS handles all aspects of appearance, you may mark up your material meaningfully, making it more accessible for non-visual or mobile devices.

Intro to CSS

STYLE SHEETS IN ACTION

1. Begin with an HTML-marked-up document.
2. Create style guidelines for how you want specific pieces to appear.
3. Include the style guidelines in the document. When the browser shows the document, it follows your rendering rules.

Intro to CSS

1. Document Marking

You've learned a lot about marking up text in earlier week. For example, you are aware that it is critical to select elements that accurately convey the content's meaning. You may have also heard me mention that markup creates the structure of the document, which is frequently referred to as the structural layer, onto which the presentation layer can be put.

Intro to CSS

2. Formulating the Rules

A style sheet is made up of one or more style instructions (called style rules) that explain how to display an element or collection of elements. The first step in studying CSS is to become acquainted with the components of a rule. As you can see, they're quite simple to follow. Each rule chooses an element and specifies how it should appear.

The example below contains two rules. The first turns all of the document's h1 elements green, while the second specifies that the paragraphs be in a big, sans-serif font. Sans-serif fonts lack a little slab (a serif) at the end of each stroke and appear more sleek and modern.

Intro to CSS

```
h1 { color: green; }
```

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

The **selector**, which identifies the element or items to be impacted, and the **declaration**, which provides the rendering instructions, are the two primary portions of a CSS rule. In turn, the **declaration** consists of a **property** (such as color) and its **value** (green), separated by a colon and a space.

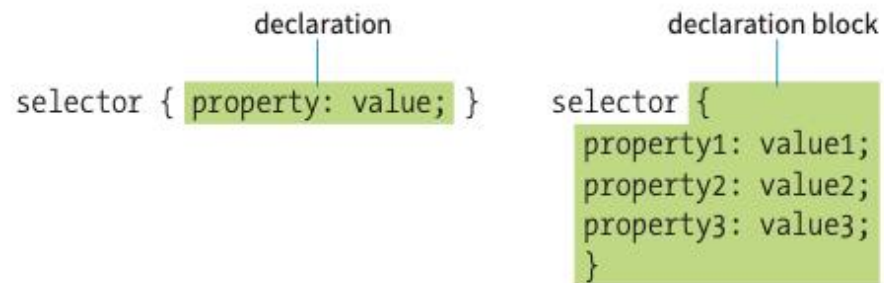


FIGURE 11-3. The parts of a style rule.

Intro to CSS

3. Adding Styles to the Document

A separate, text-only document containing a set of style rules is known as an external style sheet. It must be given the .css suffix. The .css file is then linked to (through the link element) or imported (using a style sheet's @import rule) into one or more HTML documents. As a result, all of the files on a website may use the same style sheet. This is the most effective and popular approach for adding style sheets to content.

Intro to CSS

Embedded style sheets

This style sheet is inserted into a document using the style element, and its rules are exclusive to that document. The style element must be placed in the document's head.

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

Intro to CSS

Inline styles

You can apply properties and values to a single element by using the style attribute in the element itself, as shown here:

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

To add multiple properties, just separate them with semicolons, like this:

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

Inline styles only apply to the element in which they occur. Inline styles should be avoided unless overriding styles from an embedded or external style sheet is absolutely necessary. Inline styles are problematic because they mix presentation information with structural markup. They also make adjustments more complicated because each style attribute must be found in the source.

Intro to CSS

Document structure

Understanding the structure of your document is essential at this point. HTML documents, as previously stated, have an inherent structure, or hierarchy. The sample article we've been playing with, for example, has an html root element that has a head and a body, and the body contains heading and paragraph components. Several paragraphs, in turn, contain inline elements such as images (img) and highlighted text (em). The structure can be visualized as an upside-down tree with branches extending out from the root.

Intro to CSS

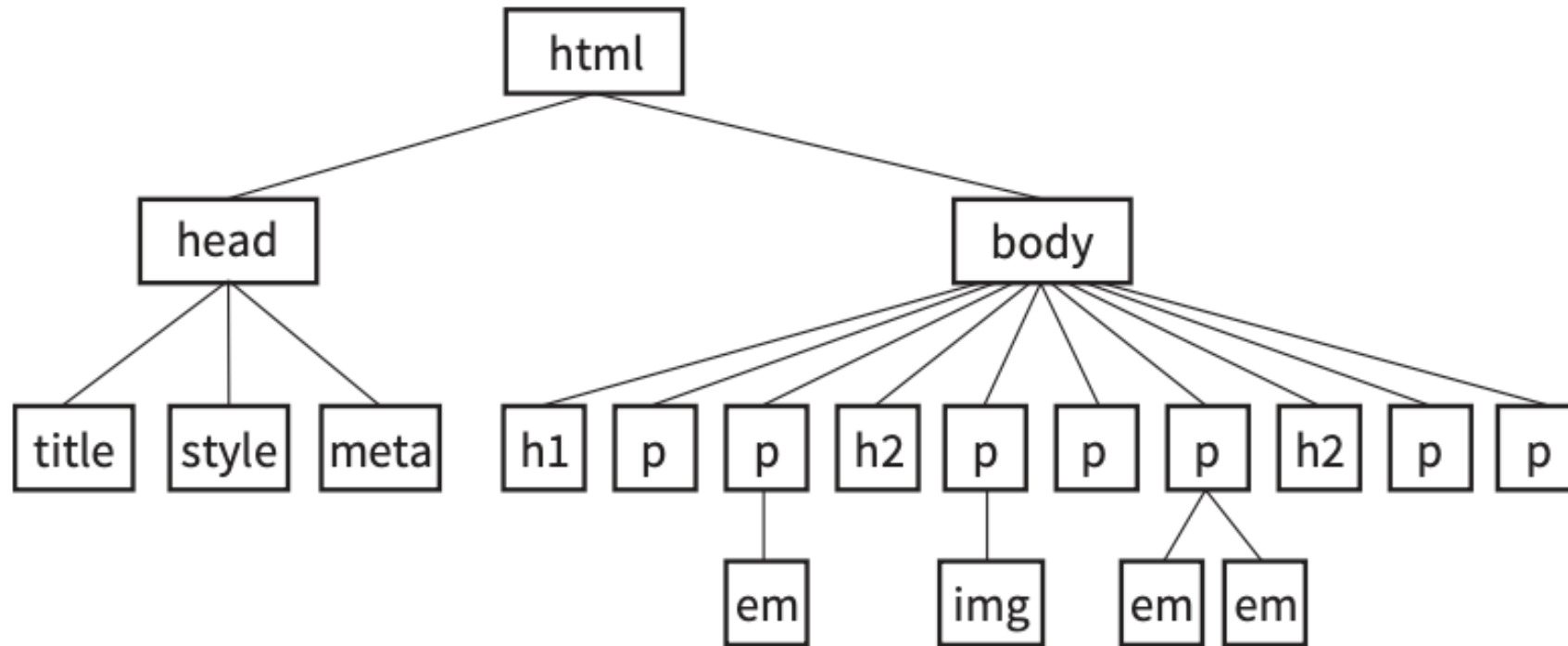


FIGURE 11-6. The document tree structure of the sample document, *cooking.html*.

Intro to CSS

Parents and their children

When it comes to referring to the relationship between items, the document tree transforms into a family tree.

The descendants of a particular element are all the elements included within it. The h1, h2, p, em, and img elements in FIGURE 11-6, for example, are all descendants of the body element.

A child of another element is one that is directly contained within another element (with no intervening hierarchical levels). The contained element, on the other hand, is the parent. The em element, for example, is the child of the p element, and the p element is its parent.

Intro to CSS

Pass it on

When you use the `p` element as a selector in a font-related style rule, the rule applies to all paragraphs in the page as well as the inline text elements they contain. Take note that I said "some" traits are inherited. It's worth noting that some style sheet properties inherit, while others do not. Text stylistic attributes, such as font size, color, and style, are generally passed down. Borders, margins, backgrounds, and other properties that impact the boxed area around the element are not typically handed down. When you think about it, this makes logic.

Intro to CSS

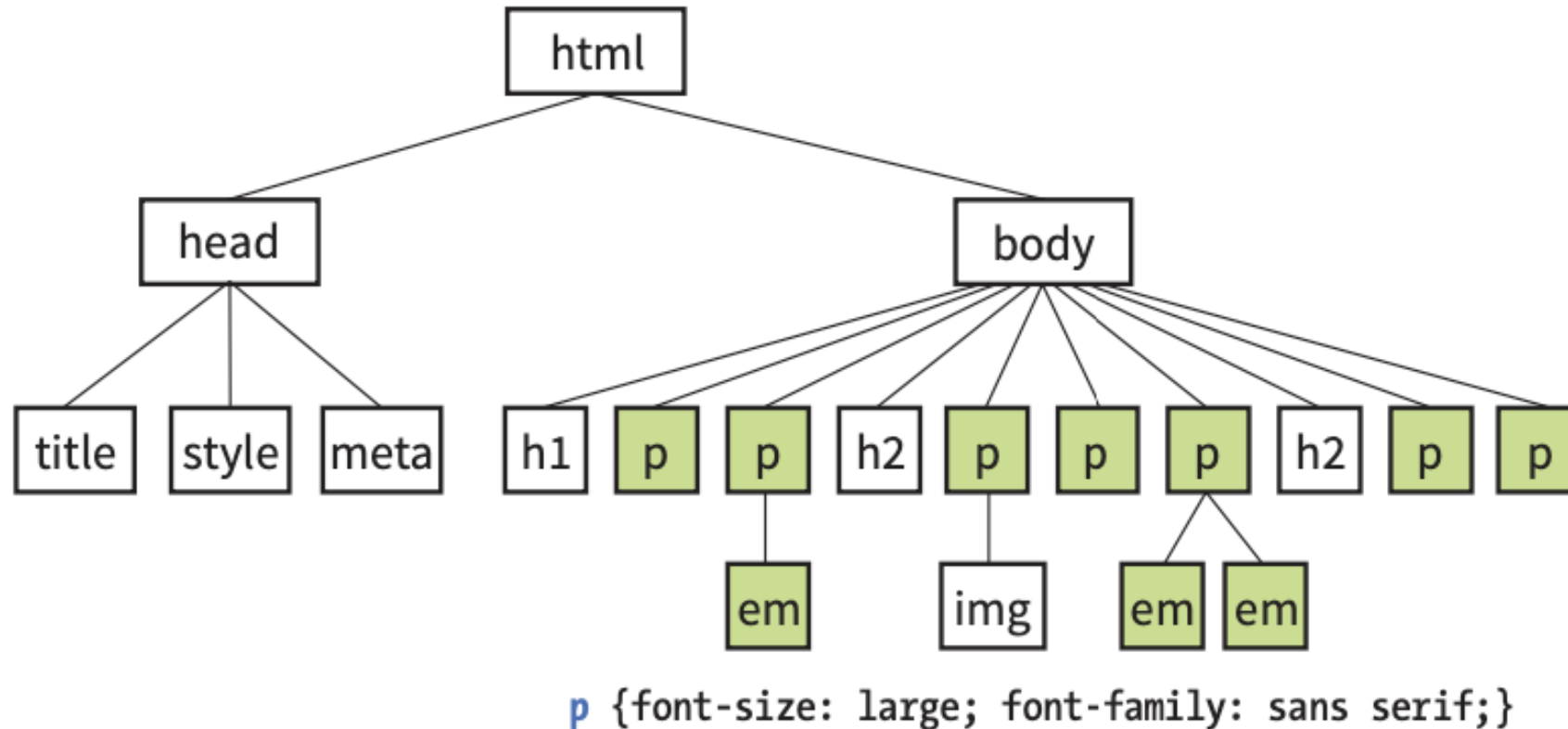


FIGURE 11-7. Certain properties applied to the `p` element are inherited by their children.

Intro to CSS

You can use inheritance to your advantage when writing style sheets. For example, if you want all text elements to be blue, you could write separate style rules for every element in the document and set the color to “blue”. A better way would be to write a single style rule that applies the color property to the body element, and let all the elements contained in the body inherit that style

Any property applied to a specific element overrides the inherited values for that property. Going back to the article example, if we specify that the em element should be orange, that would override the inherited blue setting.

Intro to CSS

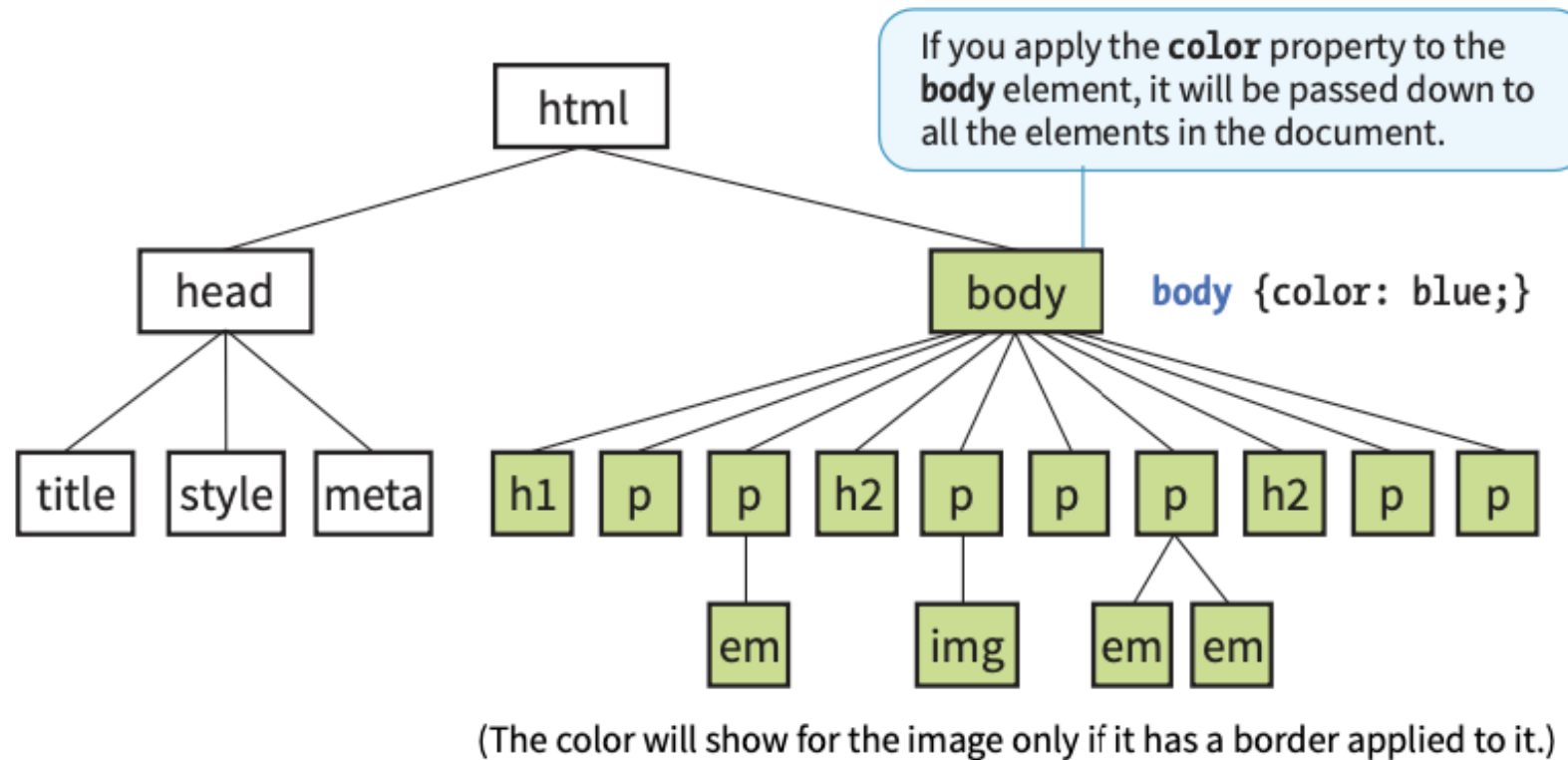


FIGURE 11-8. All the elements in the document inherit certain properties applied to the **body** element.

Intro to CSS

CSS Units

CSS3 provides a variety of units of measurement. They fall into two broad categories: **absolute** and **relative**.

Absolute 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.

px	pixel, defined as equal to 1/96 of an inch in CSS3.
in	inches.
mm	millimeters.
cm	centimeters.
q	¼ millimeter.
pt	points (1/72 inch). Points are a unit commonly used in print design.
pc	picas (1 pica = 12 points or 1/6 inch). Points are a unit commonly used in print design.

Relative units

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

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).

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.
vmin	viewport minimum unit, equal to the value of vw or vh , whichever is smaller.
vmax	viewport maximum unit, equal to the value of vw or vh , whichever is larger.

NOTES

- Although not a “unit,” 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. The spec always says what a percentage value for a property is calculated on.

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

- Child elements do not inherit the relative values of their parent, but rather the resulting *calculated* value.
- IE9 supports **vm** instead of **vmin**. IE and Edge (all versions as of 2017) do not support **vmax**.

Formatting Text

Basic Font Propertis

One of the first things I do when designing a text document (for print or the web) is choose a typeface. Fonts are provided in CSS through a set of font-related attributes that include typeface, size, weight, font style, and special characters. Shortcut properties allow you to declare several font attributes in a single rule.

Formatting Text

Choosing a typeface, or font family as it is known in CSS, is a nice place to start. Let's start with the font-family and its values.

font-family

Values: one or more font or generic font family names, separated by commas

Default: depends on the browser

Applies to: all elements

Inherits: yes

Use the font-family property to specify a font or list of fonts (known as a font stack) by name, as shown in these examples:

```
body { font-family: Arial; }  
var { font-family: Courier, monospace; }  
p { font-family: "Duru Sans", Verdana, sans-serif; }
```

Formatting Text

Here are some important syntax requirements:

- Except for generic font families, all font names must be capitalized. For example, instead of arial, use Arial.
- As illustrated in the second and third instances, use commas to separate multiple font names.
- Keep in mind that font names featuring a character space (such as Duru Sans in the third example) must be enclosed in quotation marks.

Formatting Text

Specifying Font Size

Use the aptly named font-size property to specify the size of the text. font-size

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

Default: medium

Applies to: all

Inherits: yes

Formatting Text

Using one of the CSS length units, as shown here:

```
h1 { font-size: 1.5em; }
```

When specifying a number of units, be sure the unit abbreviation immediately follows the number, with no extra character space in between

As a percentage value, sized up or down from the element's inherited font size:

```
h1 { font-size: 150%; }
```

Using one of the absolute keywords (xx-small, x-small, small, medium, large, x-large, xx-large). On most current browsers, medium corresponds to the default font size.

```
h1 { font-size: x-large; }
```

Using a relative keyword (larger or smaller) to nudge the text larger or smaller than the surrounding text:

```
strong { font-size: larger; }
```

Formatting Text

Font Weight (Boldness)

After font families and size, the remaining font properties are straightforward. For example, if you want a text element to appear in bold, use the font-weight property to adjust the boldness of type.

font-weight

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

Default: normal

Applies to: all elements

Inherits: yes

As you can see, the font-weight property has many predefined values, including descriptive terms (normal, bold, bolder, and lighter) and nine numeric values (100 to 900) for targeting various weights of a font if they are available.

Property	Description
color	Specifies the foreground color (text and borders) for an element
direction	Indicates whether the text reads left-to-right or right-to-left
font	A shorthand property that combines font properties
font-family	Specifies a typeface or generic font family
font-feature-settings	Allows access to lesser-used OpenType features
font-kerning	Controls how browsers implement kerning data (space between characters)
font-language-override	Controls use of language-specific glyphs
font-size	Specifies the size of the font
font-size-adjust	Matches the x-height of a fallback font with the specified font
font-stretch	Selects a condensed, normal, or extended font
font-style	Specifies italic or oblique fonts
font-synthesis	Controls whether a browser may simulate bold or italic fonts
font-variant	Specifies a small-caps font
font-variant-alternates	Selects alternate versions of character glyphs
font-variant-caps	Selects small caps and similar alternates when available
font-variant-east-asian	Selects alternate glyphs in Chinese, Japanese, and Korean
font-variant-ligatures	Selects ligatures for certain letter pairs when available
font-variant-numeric	Selects alternate number glyphs
font-variant-position	Selects subscript or superscript character glyphs
font-weight	Specifies the boldness of the font
hanging-punctuation	Indicates whether the punctuation may hang outside the content box
hyphens	Controls how text is hyphenated
letter-spacing	Inserts space between letters
line-break	Describes rules for breaking lines
line-height	Indicates the distance between baselines of neighboring text lines

Property	Description
list-style-image	Specifies an image to be used as a list marker
list-style-position	Puts a list marker inside or outside the content area
list-style-type	Selects the marker type for list items
overflow-wrap	Specifies whether the browser can break lines within words to prevent overflow
tab-size	Specifies the length of a tab character
text-align	Indicates the horizontal alignment of text
text-align-last	Specifies how the last line in justified text is aligned
text-decoration	Specifies underlines, overlines, and lines through
text-indent	Specifies the amount of indentation of the first line in a block
text-justify	Denotes how space is distributed in justified text
text-shadow	Adds a drop shadow under the text
text-transform	Changes the capitalization of text when it displays
unicode-bidi	Works with Unicode bidirectional algorithms
vertical-align	Adjusts the vertical position of inline elements relative to the baseline
white-space	Specifies how whitespace in the source is displayed
word-break	Specifies whether to break lines within words
word-spacing	Inserts space between words
word-wrap	Indicates whether the browser can break lines within words to prevent overflow (same as overflow-wrap)

Color and Background

Color Values in RGB

The RGB value is by far the most popular way to specify a color. It also provides you with millions of color options.

For those who are unfamiliar with how computers handle color, I'll start with the fundamentals before diving into CSS syntax.

Color and Background

A note about the RGB color space

Computers generate the colors you see on a monitor by blending three different light hues: red, green, and blue. This is referred to as the RGB color model. You can supply (kind of) color recipes by instructing the computer how much of each hue to mix in. The quantity of light in each color "channel" is normally stated on a scale ranging from 0 (none) to 255 (full blast), but it can alternatively be given as a percentage. The closer the three numbers are to 255 (100%), the more white the final color is.

Color and Background

Choosing a Color

There are several methods for selecting a color and determining its RGB color values. One quick and easy method is to go to Google.com and search "color picker," and voila—a fully functional color picker appears. If you keep an image-editing program open and nearby, such as Adobe Photoshop, you can use its built-in color picker.

Color and Background

FOREGROUND COLOR

The foreground of an element consists of its text and border (if one is specified). You specify a foreground color with the color property, as we saw in the last chapter when we rolled it out to give text a little pizzazz. Here are the details for the color property one more time.

color

Values: color value (name or numeric)

Default: depends on the browser and user's preferences

Applies to: all

Inherits: yes

In the following example, the foreground of a blockquote element is set to green with a color name. You can see that applying the color property to the blockquote element means the color is inherited by the p and em elements it contains (FIGURE 13-9). The thick dashed border around the whole block- quote is green as well; however, if we were to apply a border-color property to this same element, that color would override the green foreground setting.

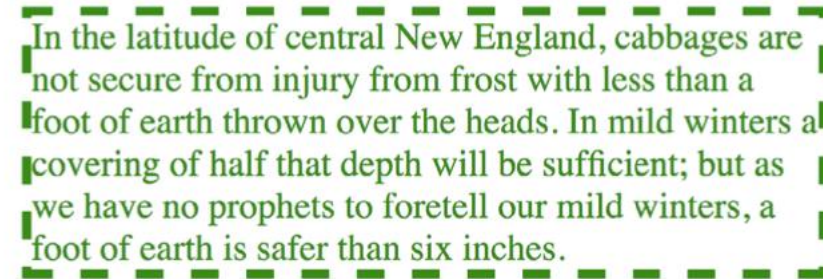
Color and Background

THE STYLE RULE

```
blockquote {  
border: 4px dashed;  
  
color: green;  
  
}
```

THE MARKUP

```
<blockquote>  
In the latitude of central New England, cabbages are not secure ... </blockquote>
```



In the latitude of central New England, cabbages are not secure from injury from frost with less than a foot of earth thrown over the heads. In mild winters a covering of half that depth will be sufficient; but as we have no prophets to foretell our mild winters, a foot of earth is safer than six inches.

FIGURE 13-9. Applying a color to the foreground of an element.

Color and Background

BACKGROUND COLOR

Use background-color to apply a background color to any element. background-color

Values: color value (name or numeric) | transparent

Default: transparent

Applies to: all elements

Inherits: no

A background color fills the canvas behind the element that includes the content area, and any padding (extra space) added around the content, extending behind the border out to its outer edge. Let's see what happens when we use the background-color property to make the background of the same sample blockquote light green (FIGURE 13-10):

```
blockquote {  
border: 4px dashed;  
color: green; background-color: #c6de89;  
}
```

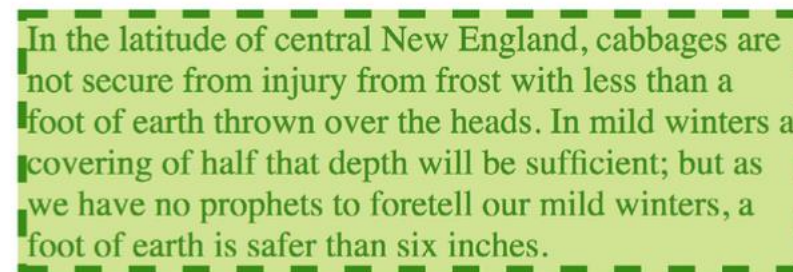


FIGURE 13-10. Adding a light green background color to the sample blockquote.

Color and Background

Adding a Background Image

The background-image property adds a background image to any element. Its primary job is to provide the location of the image file.

background-image

Values: `url(location of image) | none`

Default: `none`

Applies to: all elements

Inherits: no

The value of background-image is a sort of URL holder that contains the location of the image (see Note).

The URL is relative to wherever the CSS rule is at the time. If the rule is in an embedded style sheet (a style element in the HTML document), then the pathname in the URL should be relative to the location of the HTML file. If the CSS rule is in an external style sheet, then the pathname to the image should be relative to the location of the .css file.

As an alternative, providing site root relative URLs for images ensures that the background image can be found regardless of the location of the style rules.

Color and Background

- **background-repeat**
- **Values:** repeat | no-repeat | repeat-x | repeat-y | space | round
- **Default:** repeat
- **Applies to:** all
- **Inherits:** no
- If you want a background image to appear just once, use the **no-repeat** key- word value:
- `body {background-image: url(star.png); background-repeat: no-repeat; }`
- You can also restrict the image to tiling only horizontally (**repeat-x**) or vertically (**repeat-y**), as shown in these examples:
- `body {background-image: url(star.png); background-repeat: repeat-x; }`
- `body { background-image: url(star.png); background-repeat: repeat-y; }`

Color and Background

The remaining keyword values, space and round, attempt to fill the available background painting area an even number of times.

When background-repeat is set to space, the browser calculates how many background images can fit across the width and height of the background area, then adds equal amounts of space between each image.

The round keyword makes the browser squish the background image horizontally and vertically (not necessarily proportionally) to fit in the back- ground area an even number of times

Color and Background

Property	Description
background	Shorthand property that combines background properties
background-attachment	Specifies whether the background image scrolls or is fixed
background-clip	Specifies how far the background image should extend
background-color	Specifies the background color for an element
background-image	Provides the location of an image to use as a background
background-origin	Determines how the background-position is calculated (from edge of border, padding, or content box)
background-position	Specifies the location of the origin background image
background-repeat	Specifies whether and how a background image repeats (tiles)
background-size	Specifies the size of the background image
color	Specifies the foreground (text and border) color
opacity	Specifies the transparency level of the foreground and background

Intro to CSS

Have you ever wondered why style sheets are referred to as "cascading"? Because CSS allows you to apply many style sheets to the same document, there will inevitably be conflicts. What should the browser do, for example, if a document's imported style sheet specifies that h1 elements should be red but its embedded style sheet specifies that h1s should be purple? Do the two style rules with h1 selectors have the same weight?

The authors of the style sheet specification anticipated this issue and designed a hierarchical structure that allocates varying weights to different sources of style information. When numerous sources of style information compete for control of the components on a page, style information is passed down ("cascades") until it is overridden by a style rule with higher weight. Weight is assigned based on the priority of the style rule source, selector specificity, and rule order.

Intro to CSS

Priority

If you don't apply any style information to a web page, it renders according to the browser's internal style sheet. We've been calling this the default rendering; the W3C calls it the user agent style sheet. Individual users can apply their own styles as well (the user style sheet, also called the reader style sheet), which override the default styles in their browser. However, if the author of the web page has attached a style sheet (the author style sheet), that overrides both the user and the user agent styles. The sidebar "Style Rule Hierarchy" provides an overview of the cascading order from highest to lowest priority.

The only exception is if the user has identified a style as "important," in which case that style will override all competing styles. This permits users to keep settings accommodating a disability such as extra large type for sight impairment.

Intro to CSS

Specificity

It is possible for conflicts to arise in which an element is getting style instructions from more than one rule. For example, there may be a rule that applies to paragraphs and another rule for a paragraph that has the ID “intro.” Which rule should the intro paragraph use?

When two rules in a style sheet conflict, the type of selector is used to determine the winner. The more specific the selector, the more weight it is given to override conflicting declarations. In our example, the selector that includes the ID name (#intro) is more specific than a general element selector (like p), so that rule would apply to the “intro” paragraph, overriding the rules set for all paragraphs.

Intro to CSS

Rule order

After all the style sheet sources have been sorted by priority, and after all the linked and imported style sheets have been shuffled into place, there are likely to be conflicts in rules with equal weights. When that is the case, the order in which the rules appear is important. The cascade follows a “last one wins” rule. Whichever rule appears last has the last word.

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

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

In this scenario, paragraph text will be green because the last rule in the style sheet—that is, the one closest to the content in the document—overrides the earlier ones. Procedurally, the paragraph is assigned a color, then assigned a new one, and finally a third one (green) that gets used.

Intro to CSS

The style sheet should be named with the `.css` suffix. It may also begin with the `@charset` at-rule to declare the character encoding, although you really need to do that only if you are using an encoding other than UTF-8. If you use `@charset`, it must be the first element in the style sheet, with no characters, including comments or style rules, preceding it.

Intro to CSS

Using the link Element

The link element defines a relationship between the current document and an external resource. By far, its most popular use is to link to style sheets. The link element goes in the head of the document, as shown here:

```
<head>  
<title>Titles are required.</title>  
<link rel="stylesheet" href="/path/stylesheet.css">  
</head>
```

You need to include two attributes in the link element:

rel="stylesheet" Defines the linked document's relation to the current document. The value of the rel attribute is always stylesheet when you are linking to a style sheet.

href="url"
Provides the location of the .css file.

You can include multiple link elements to different style sheets, and they'll all apply. If there are conflicts, whichever one is listed last will override previous settings, because of the rule order and the cascade

Thank You

Alfred Tenggono, S.Kom., M.Kom.

alfred.tenggono@jiu.ac

Reference

- Learning Web Design, Jennifer Niederst Robbins, O'Reilly Media, Inc., 2018, ISBN: 978-1-491-96020-2