

Client-side Technology

HTML and CSS



Web Technology

Asst. Prof. Manop Phankokkruad, Ph.D.

Faculty of Information Technology

King Mongkut's Institute of Technology Ladkrabang



Outline

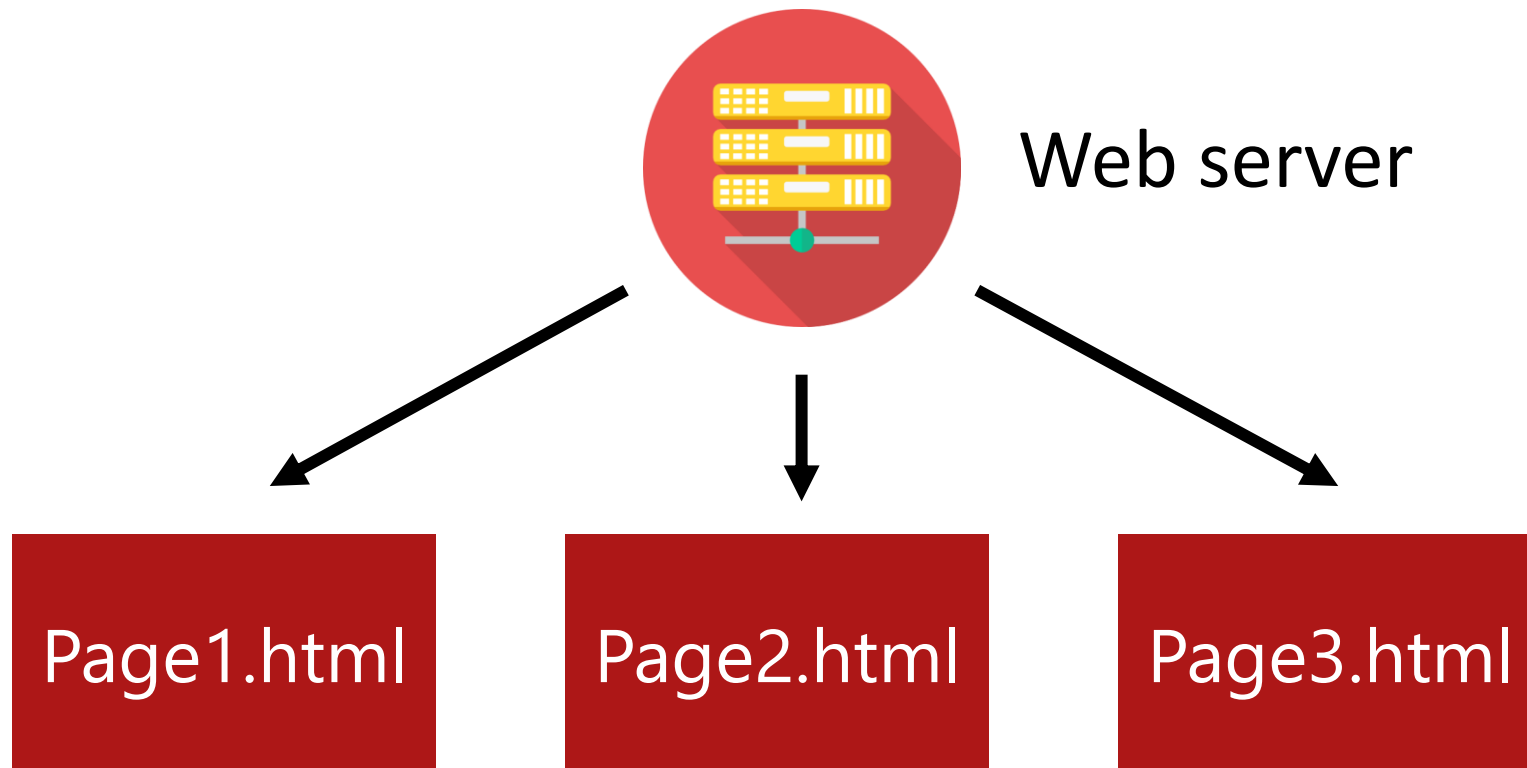
- ❑ Website Development Overview
- ❑ Introduction to HTML
- ❑ Cascading Style Sheets



Overview

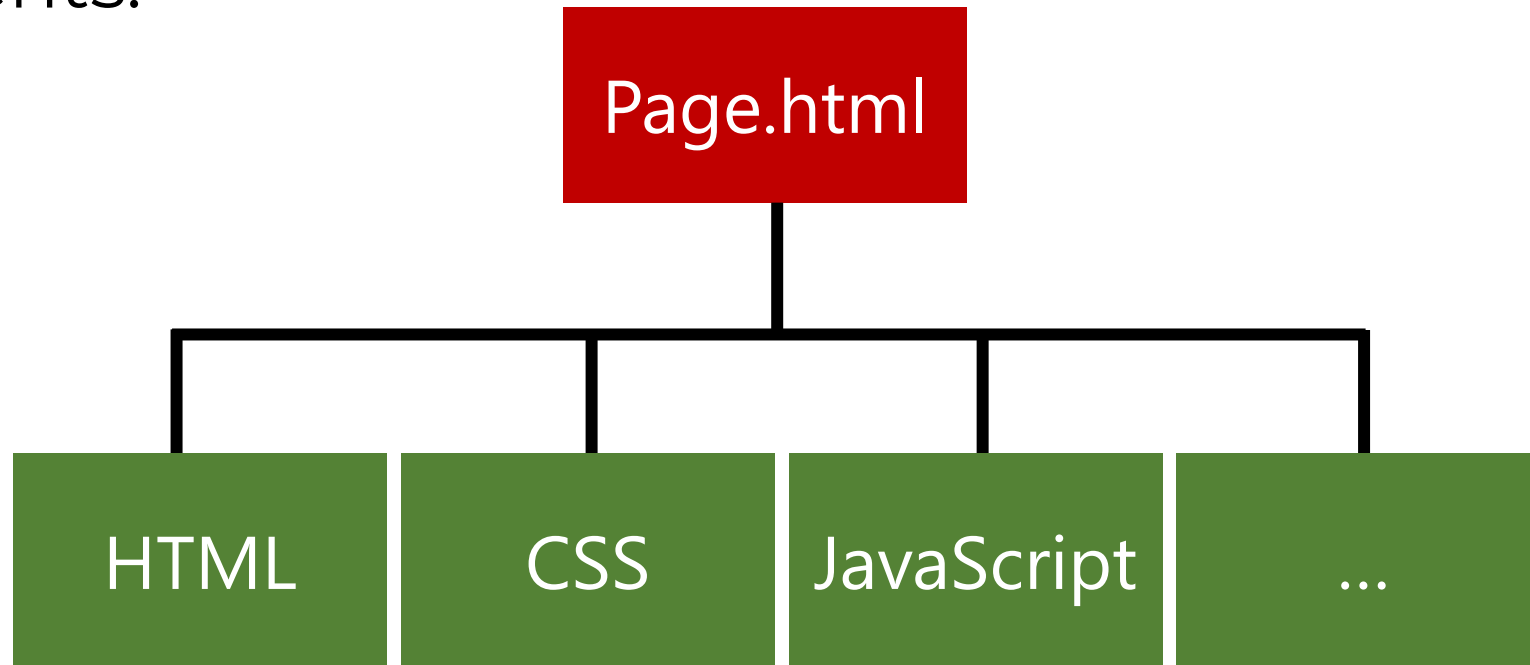


- A website is a group of self-contained, individual pages, sent to the browser from the server one-page at a time.



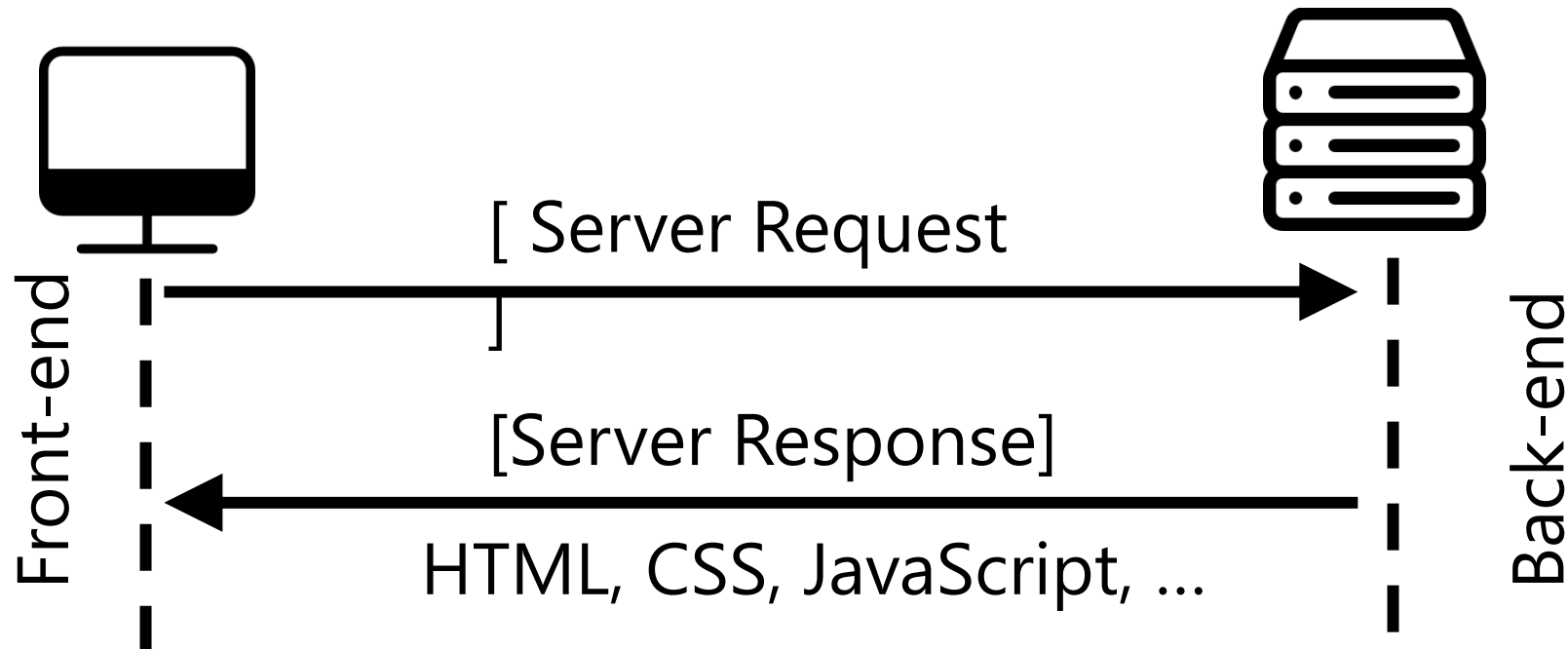
Overview

- ❑ A dynamic web content is built when it is requested, by the user directly, or programmatically while a user is on a page.
- ❑ Most websites contain both static and dynamic elements.



Overview

- ❑ Client-side (front-end) coding includes HTML, CSS and JavaScript. This just means that code will be downloaded from the server and then compiled entirely in the browser.



Overview

Common client-side web technologies:

- ❑ HTML
- ❑ CSS
- ❑ JavaScript

Three layers of web design:

- ❑ Structure - HTML markup, Site planning
- ❑ Style - CSS, Imagery
- ❑ Behavior - JavaScript

- ❑ **HTML** (HyperText Markup Language) is the standard markup language used to create **web pages** and **web applications**. Its elements form the building blocks of pages, representing formatted text, images, form inputs, and other structures.
- ❑ When a browser makes a request to a URL, whether fetching a page or an application, **the first thing that is returned is an HTML document**. This HTML document **may include additional information about its look and layout in the form of CSS, or behavior in the form of JavaScript**.

Hypertext + Markup language

- ❑ **Hypertext** is text which contains links to other texts.



- ❑ A **markup language** is a set of markup tags.
- ❑ The purpose of the tags is to group and describe page content.

Element Tags

- ❑ The fundamental building block in every HTML document is the element tag, which marks an element in the document. A **starting tag** indicates the beginning of that element, while an **ending tag** indicates the ending. The general syntax of a two-sided element tag is

```
<element>Content</element>
```

where ***element*** is the name of the element, content is the element's content, **<element>** is the starting tag, and **</element>** is the ending tag.

HTML: Elements

□ The essential element tags

Primary Structure

- html
- head
- body

Head Elements

- title
- meta
- link

Structural Elements

- p
- br
- h1 – h6
- ul
- ol
- a
- img
- div

HTML: Elements

□ The essential element tags (cont.)

Formatting Elements

- em
- i
- strong
- b
- q
- blockquote
- span

HTML: Basic Elements

- The basic elements of an HTML page are:
- A text header, denoted using the `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>` tags.
- A paragraph, denoted using the `<p>` tag.
- A horizontal ruler, denoted using the `<hr>` tag.
- A link, denoted using the `<a>` (anchor) tag.
- A list, denoted using the `` (unordered list), `` (ordered list) and `` (list element) tags.
- An image, denoted using the `` tag
- A divider, denoted using the `<div>` tag
- A text span, denoted using the `` tag.

Element Attributes

- ❑ Elements will often contain one or more element attributes. Each attribute provides additional information to the browser about the purpose of the element or how the element should be handled by the browser. The general syntax of an element attribute within a two-sided tag is

```
<element attr1="value1" attr2="value2" >  
    Content  
</element>
```

For example

```
<p id="intro">This is my first paragraph.</p>
```

Element Attributes

□ Commonly used core HTML attributes

Attribute	Description
<code>class="text"</code>	Defines the general classification of the element
<code>dir="ltr rtl auto"</code>	Defines the text direction of the element content as left-to-right, right-to-left, or determined by the browser
<code>hidden</code>	Indicates that the element should be hidden or is no longer relevant [HTML5]
<code>id="text"</code>	Provides a unique identifier for the element
<code>lang="text"</code>	Specifies the language of the element content
<code>style="definition"</code>	Defines the style or appearance of the element content
<code>tabindex="integer"</code>	Specifies the tab order of the element (when the tab button is used to navigate the page)
<code>title="text"</code>	Assigns a title to the element content

Element Attributes

The essential attributes

- **href** : `This is a link`
- **src** : ``
- **width** and **height** : ``
- **alt** : ``

HTML Basic Structure

```
<!DOCTYPE html>
```

The doctype is not actually a tag, but a declaration, telling the browser what kind of html you are using. The doctype above declares HTML 5.

```
<html> </html>
```

<html> element defines the whole HTML document.

HTML Basic Structure

<head></head>

<head> element contains special elements that instruct the browser where to find stylesheets, provide meta info, and more.

<body></body>

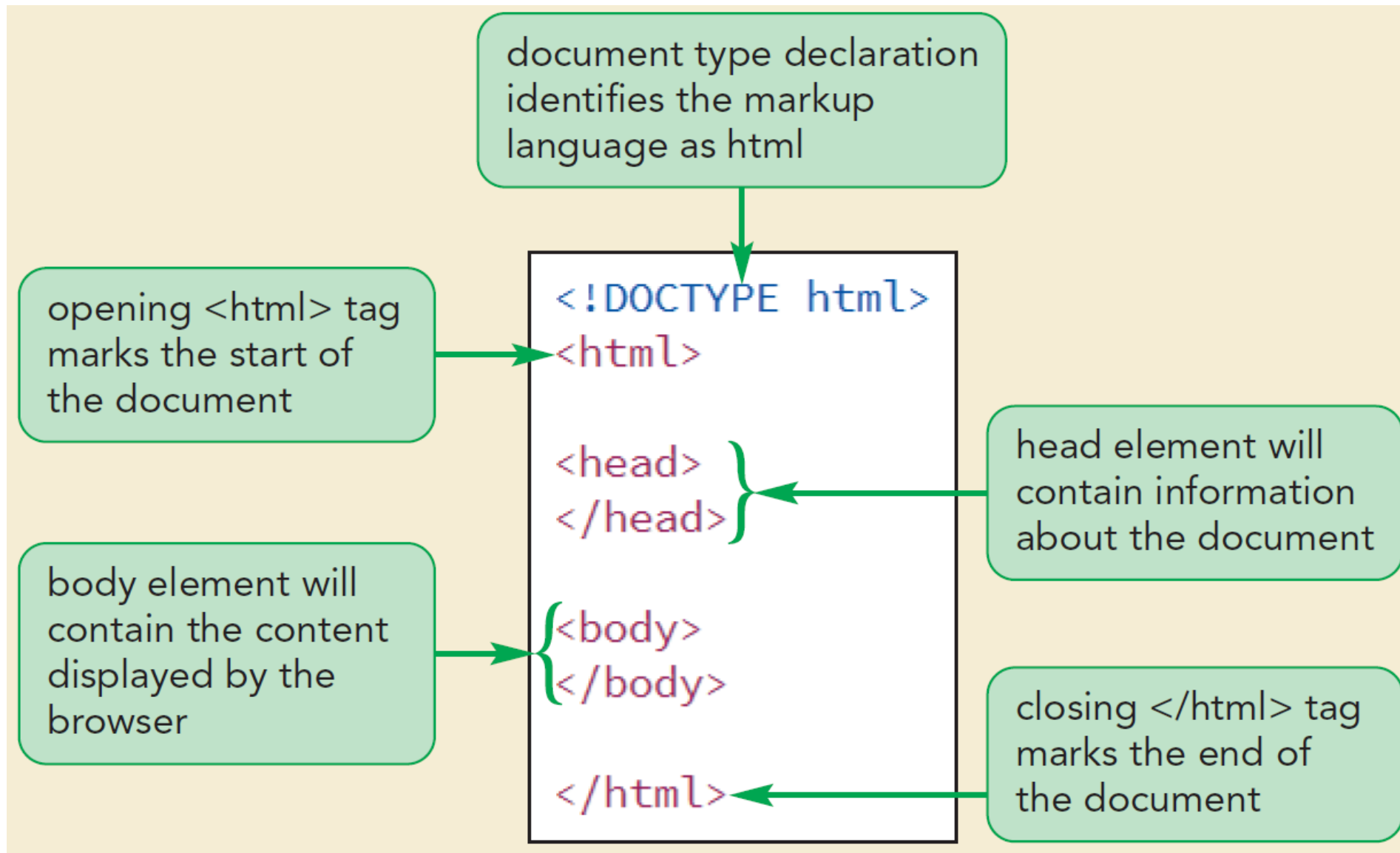
<body> element contains the document content what is shown inside the browser window.

HTML Basic Structure

- ❑ Let's briefly look at a small example file to gain a more concrete understanding of HTML syntax and semantics.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <p>This is my first paragraph.</p>
  </body>
</html>
```

HTML Basic Structure



HTML : Nesting

- ❑ The use of first three tags (html, head and body), introduces an important concept: Nesting, which is when tags “wrap” other tags.

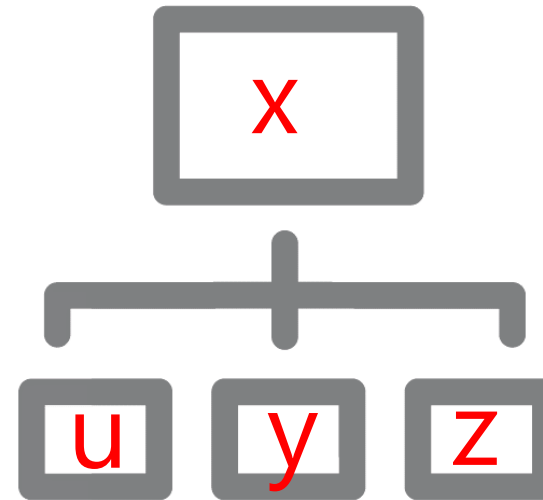
```
<html>
  <head></head>
  <body>
    <h1></h1>
    <div>
      <p></p>
    </div>
    <p></p>
  </body>
</html>
```

When you create markup, you should indicate nesting by indenting the nested tags with 2 spaces (preferred) or a tab.

HTML: Document Hierarchy

- ❑ Document Hierarchy: Parents, children and siblings

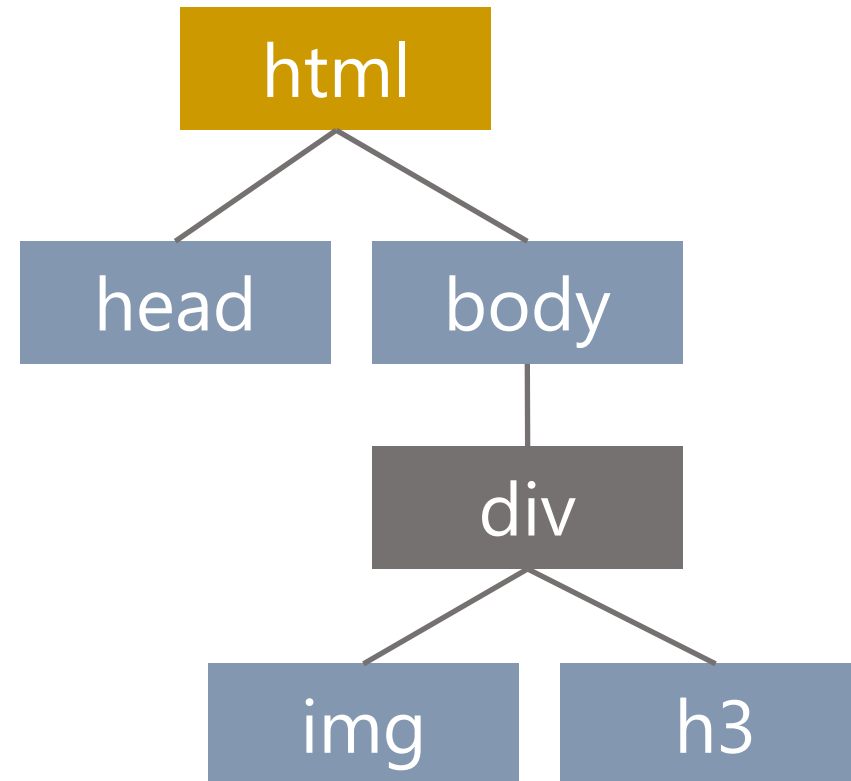
```
<parent x>  
  <child u> </child u>  
  <child y> </child y>  
  <child z> </child z>  
</parent x>
```



HTML: Document Hierarchy

- ❑ Document Hierarchy: Parents, children and siblings

```
<html>
<head></head>
<body>
  <div id="logo">
    
    <h3></h3>
  </div>
</body>
</html>
```



HTML : Tables

- An HTML table is defined with the `<table>` tag.
- Each table row is defined with the `<tr>` tag.
- A table header is defined with the `<th>` tag. By default, table headings are bold and centered.
- A table data/cell is defined with the `<td>` tag.

```
<table>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
</table>
```

HTML : Lists

❑ Ordered Lists

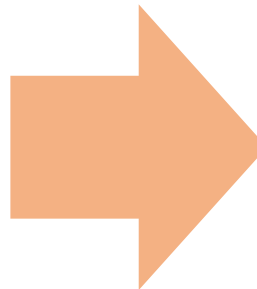
```
<ol>
  <li>item1</li>
  <li>item2</li>
  ...
</ol>
```



1. Coffee
 2. Tea
 3. Coca Cola
-
- I. Coffee
 - II. Tea
 - III. Coca Cola

❑ Unordered Lists

```
<ul>
  <li>item1</li>
  <li>item2</li>
  ...
</ul>
```



- Coffee
 - Tea
 - Coca Cola
-
- Coffee
 - Tea
 - Coca Cola

HTML : Lists

□ Description Lists

```
<dl>
  <dt>term1</dt>
  <dd>description1</dd>
  <dt>term2</dt>
  <dd>description2a</dd>
  <dd>description2b</dd>
  ...
</dl>
```

```
<dl>
  <dt>Coffee</dt>
  <dd>Black hot drink</dd>
  <dt>Milk</dt>
  <dd>White cold drink</dd>
</dl>
```

HTML : Comments

Adding Comments

- ❑ A comment is descriptive text that is added to the HTML file but that does not appear in the browser window when the page is displayed. Comments are added with the following markup:

```
<!-- comment -->
```

- ❑ **CSS (Cascading Style Sheets)** is used to control the look and layout of HTML elements.
- ❑ CSS styles can be applied directly to an HTML element, defined separately on the same page, or defined in a separate file and referenced by the page.
- ❑ Styles cascade based on how they are used to select a given HTML element.

CSS : Cascading + Style Sheet

- ❑ A **style sheet** is a set of rules defining how an HTML element will be presented in the browser. These rules are targeted to specific elements in the HTML document.
- ❑ The **cascade** part is a set of rules for resolving conflicts with multiple CSS rules applied to the same elements.

Link an HTML document to a Style Sheet

- ❑ **Browser Style Sheet** – the browsers put default styling on some elements.
- ❑ **External Style Sheet**

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

link elements link the web page to a style sheet file

filename of the CSS style sheet

rel attribute indicates the type of link relationship

```
<meta charset="utf-8" />
<meta name="keywords" content="Thai, restaurant, Charlotte, food" />
<title>About Curbside Thai</title>
<link href="ct_base.css" rel="stylesheet" />
<link href="ct_layout2.css" rel="stylesheet" />
</head>
```

Link an HTML document to a Style Sheet

❑ Internal Style Sheet

```
<html>
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
```

❑ Inline Styles

```
<h1 style="color:blue;">This is a Blue Heading</h1>
```

CSS : Inheritance

- ❑ Most elements will inherit many style properties from their parent elements by default.

HTML

relationship

<code><body></code>	→	parent of site
<code><div></code>	→	parent of <code>ul</code> and <code>li</code> , child of
<code></code>	→	<code>body</code>
<code></code>	→	parent of <code>li</code> , child of <code>div</code> and
<code></code>		<code>body</code>
<code></div></code>		child of <code>ul</code> , <code>div</code> , and <code>body</code>
<code></body></code>		

CSS : Specificity

Specificity refers to how specific your selector is in naming an element.

body

make the paragraph 16px, Verdana, red



p

make the paragraph blue



p.pink

make the paragraph pink

16px, Verdana, pink

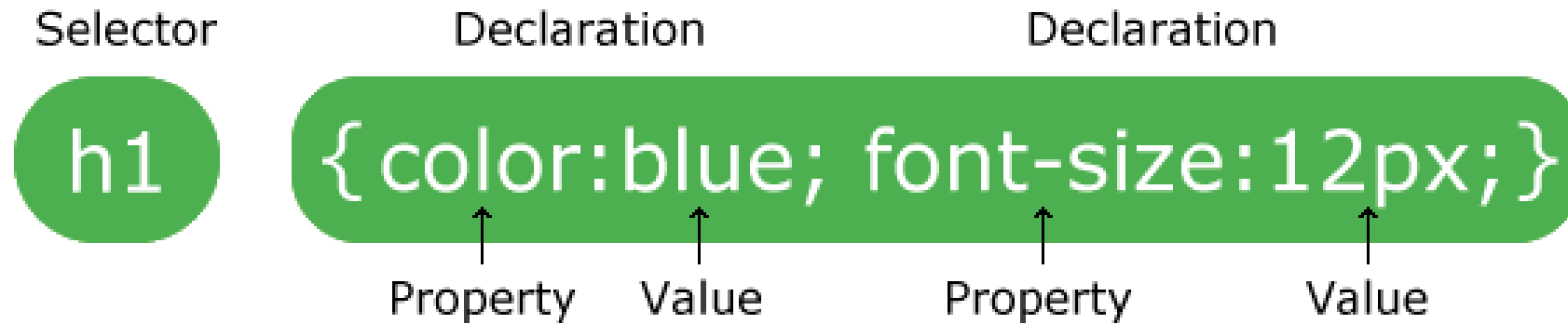
CSS : Syntax

Syntax is the rules for how to write the language.
Three terms for describing the styles:

- CSS rules
- CSS selector
- CSS declaration

CSS : Rules

- ❑ A CSS rule-set consists of a ***selector*** and a ***declaration*** block.



The ***selector*** points to the HTML element you want to style. The declaration block contains one or more declarations separated by semicolons.

CSS : Selector

- ❑ The selector associates the CSS rules with HTML elements.

```
body {  
    font-family: Arial, Helvetica;  
}  
p {color: #666666;}  
a {color: blue;}  
h1, h2, h3 { font-weight: bold; }
```

CSS : Selectors

essential selector types :

- ❑ type/element selector
- ❑ id selector
- ❑ class selector

element p { property: value }

id #id { property: value }

class .class { property: value }

body, p, h1, ul are the simplest type selector, which targets an html element by name.

CSS : Selectors

CSS

```
#logo {declaration}  
.ingredients {declaration}
```

HTML

```
  
<ul class = "ingredients">
```

The most important difference between IDs and classes is that there can be only one ID on a page, but multiple classes.

- An **ID** is more specific than a class.
- An **element** can have both an ID and multiple classes.

Creating a Style Sheet

❑ Writing Style Comments

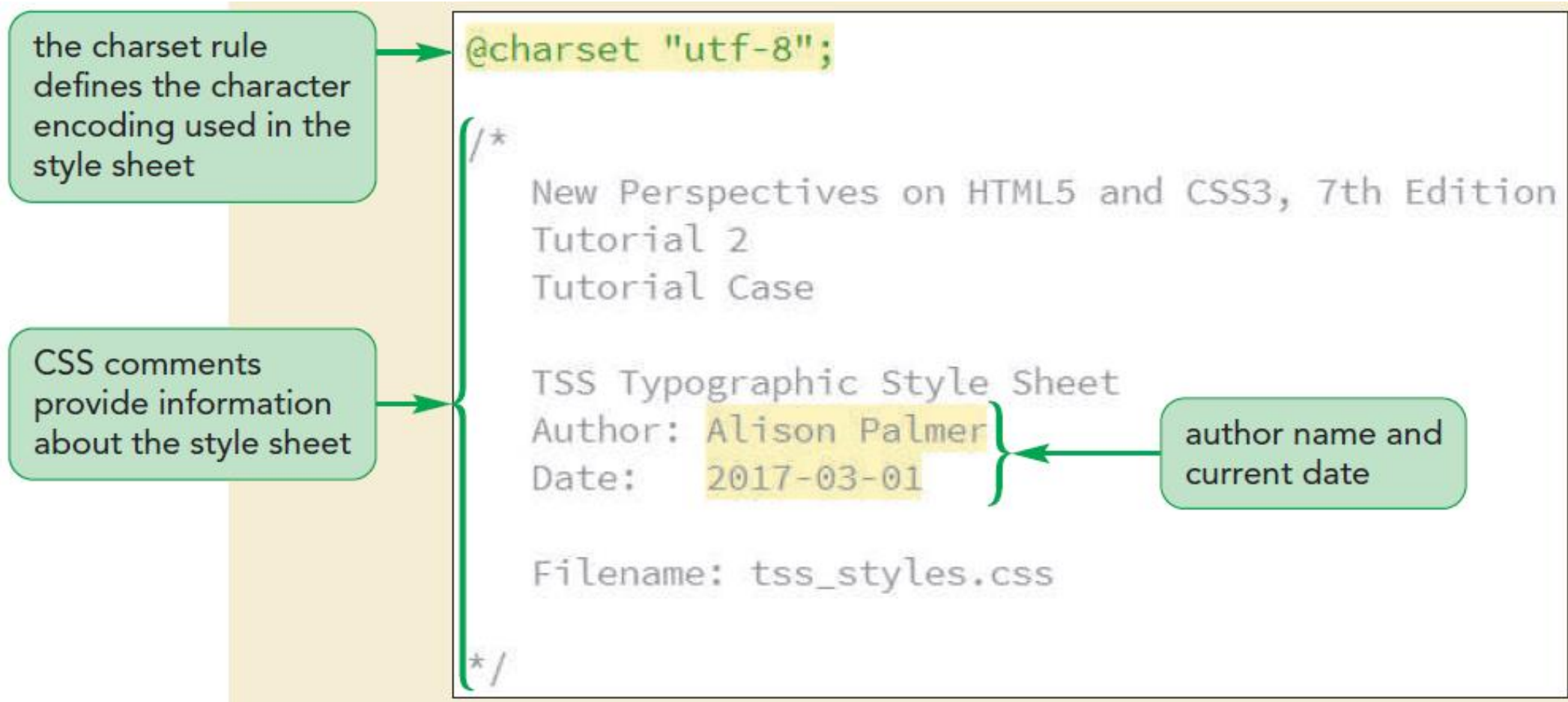
```
/*  
    Comment  
*/
```

❑ Defining the Character Encoding

```
@charset "encoding";
```

Creating a Style Sheet

❑ Adding the @charset rule and style comments



Working with Color

- ❑ **Color Names** - CSS supports 147 color names covering common names such as red, green, and yellow to more exotic colors such as ivory, orange, crimson, khaki, and brown.

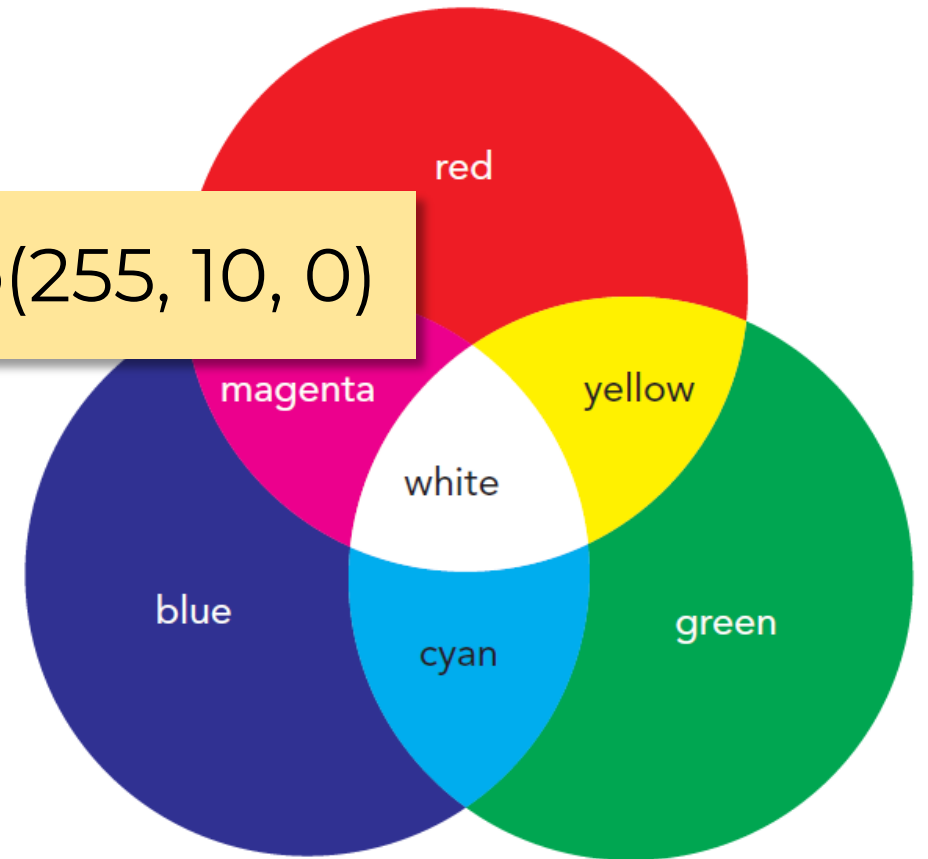
- ❑ RGB Color Values

`rgb(red, green, blue)`



`rgb(255, 10, 0)`

where *red*, *green*, and *blue* are the intensities of the red, green, and blue components of the color. Intensities range from 0 to 255



Working with Color

- ❑ **RGB values** are sometimes expressed as hexadecimal numbers where a hexadecimal number is a number expressed in the base 16 numbering system

#redgreenblue)



#FFFFFF00

Hexadecimal numbering includes these ten numeric characters and six extra characters: A (for 10), B (for 11), C (for 12), D (for 13), E (for 14), and F (for 15). For values above 15.

Working with Color

- ❑ **HSL Color Values** - HSL color values were introduced in CSS3 and are based on a color model in which each color is determined by its hue, saturation, and lightness.

```
hsl(hue, saturation, lightness)
```

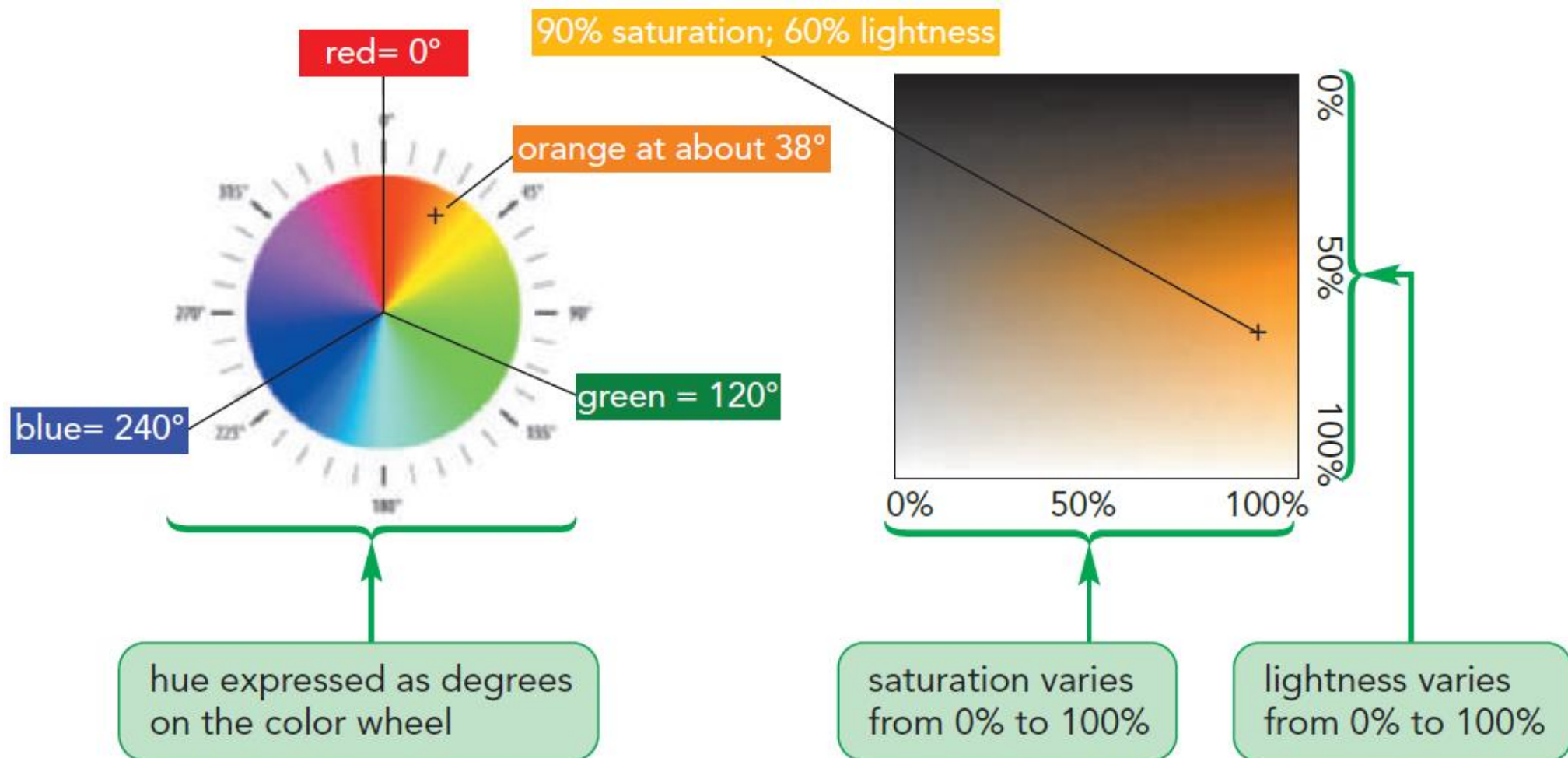


```
hsl(38, 90%, 60%)
```

Graphic designers consider HSL easier to use because it allows them to set the initial color based on hue and then fine-tune the saturation and lightness values.

Working with Color

- ❑ Defining the color orange under the HSL color model. For example : `hsl(38, 90%, 60%)`



Working with Color

- ❑ Setting Text and Background Colors - CSS supports the following styles to define both the text and background color for each element on your page

```
color: color;  
background-color: color;
```

```
html { background-color: hsl(27, 72%, 72%); }  
body {  
    color: rgb(91, 91, 91);  
    background-color: ivory;  
}
```

Working with Fonts

- ❑ Text characters are based on fonts that define the style and appearance of each character in the alphabet. The default font used by most browsers for displaying text is Times New Roman, but you can specify a different font for any page element using the following **font-family** property

font-family: fonts;



font-family: 'Arial Black';

browser attempts to use the Verdana font first, followed by Geneva, and finally any generic sans-serif font

```
body {  
  color: rgb(91, 91, 91);  
  background-color: ivory;  
  font-family: Verdana, Geneva, sans-serif;  
}
```



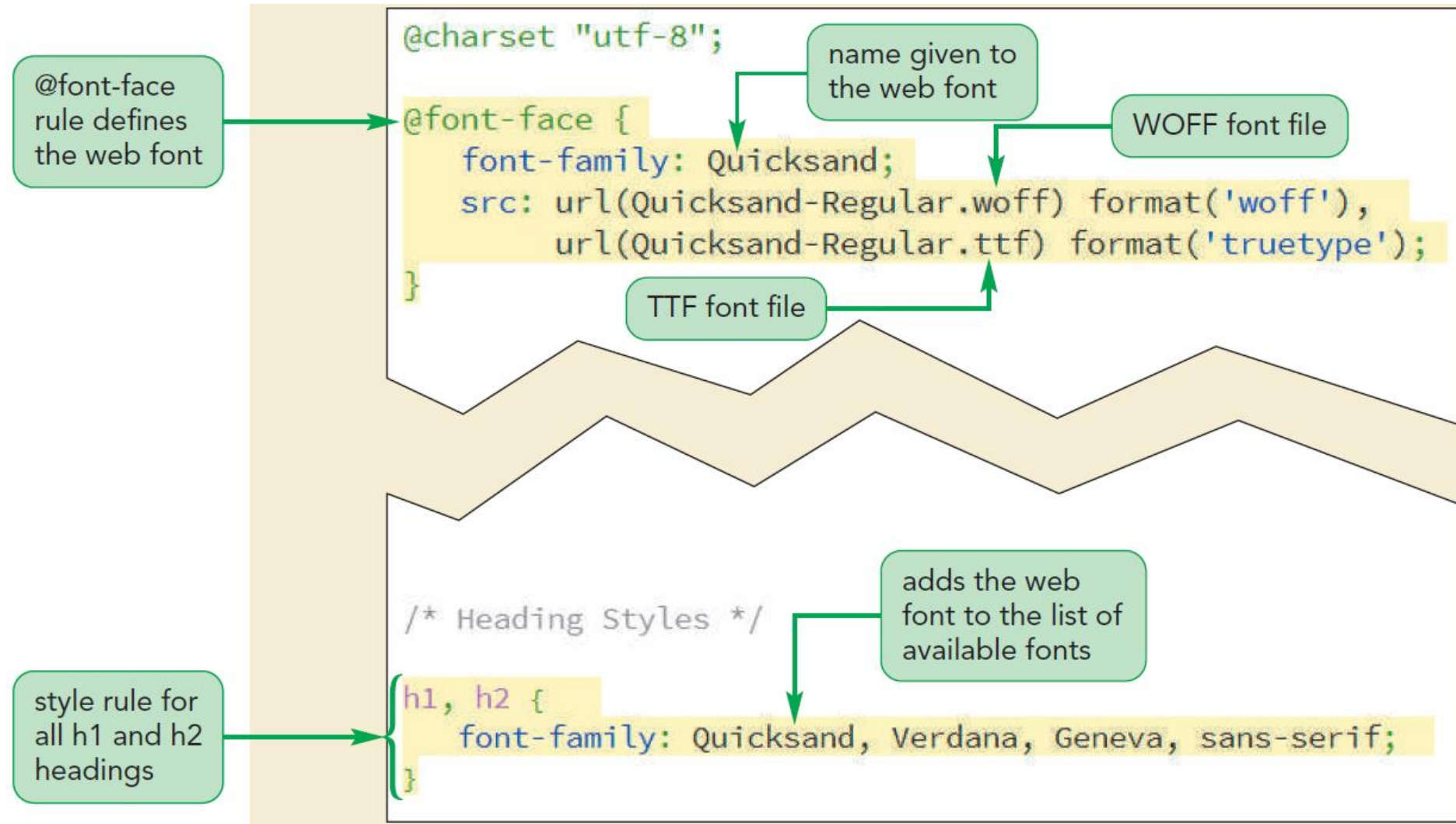
Working with Fonts - Web Fonts

- ❑ The format most universally accepted in almost all current browsers and on almost all devices is the Web Open Font Format (WOFF). Web font files can be downloaded from several sites on the Internet.
- ❑ To access and load a web font, you add the following **@font-face** rule to the style sheet

```
@font-face {  
    font-family: Gentium;  
    src: url('Gentium.woff') format('woff'),  
         url('Gentium.ttf') format('truetype');  
}
```

Working with Fonts - Web Fonts

❑ Accessing a web font



More Information

- ❑ HTML5 Tutorial

<https://www.w3schools.com/html/>

- ❑ CSS Tutorial

<https://www.w3schools.com/css/default.asp>