CSS is the language we use to style a Web page.

# What is CSS?

* CSS stands for Cascading Style Sheets
* CSS describes how HTML elements are to be displayed on screen, paper, or in other media
* CSS saves a lot of work. It can control the layout of multiple web pages all at once
* External stylesheets are stored in CSS files

# Why Use CSS?

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

## CSS Example

body {

background-color: lightblue;

}

h1 {

color: white;

text-align: center;

}

p {

font-family: verdana; font-size: 20px;

}

# CSS Solved a Big Problem

HTML was NEVER intended to contain tags for formatting a web page! HTML was created to describe the content of a web page, like:

<h1>This is a heading</h1>

<p>This is a paragraph.</p>

When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS. CSS removed the style formatting from the HTML page!

# Three Ways to Insert CSS

There are three ways of inserting a style sheet:

* External CSS
* Internal CSS
* Inline CSS

# External CSS

With an external style sheet, you can change the look of an entire website by changing just one file!

Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.

<link rel="stylesheet" href="mystyle.css">

An external style sheet can be written in any text editor, and must be saved with a .css extension.

The external .css file should not contain any HTML tags. Here is how the "mystyle.css" file looks:

**Mystyle.css**

body {

background-color: lightblue;

}

h1 {

color: navy; margin-left: 20px;

}

# Internal CSS

An internal style sheet may be used if one single HTML page has a unique style. The internal style is defined inside the <style> element, inside the head section.

## Example

Internal styles are defined within the <style> element, inside the <head> section of an HTML page:

<head>

<style> body {

background-color: linen;

}

h1 {

color: maroon; margin-left: 40px;

}

</style>

# Inline CSS

An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

<h1 style="color:blue;text-align:center;">This is a heading</h1>

<p style="color:red;">This is a paragraph.</p>

# Cascading Order

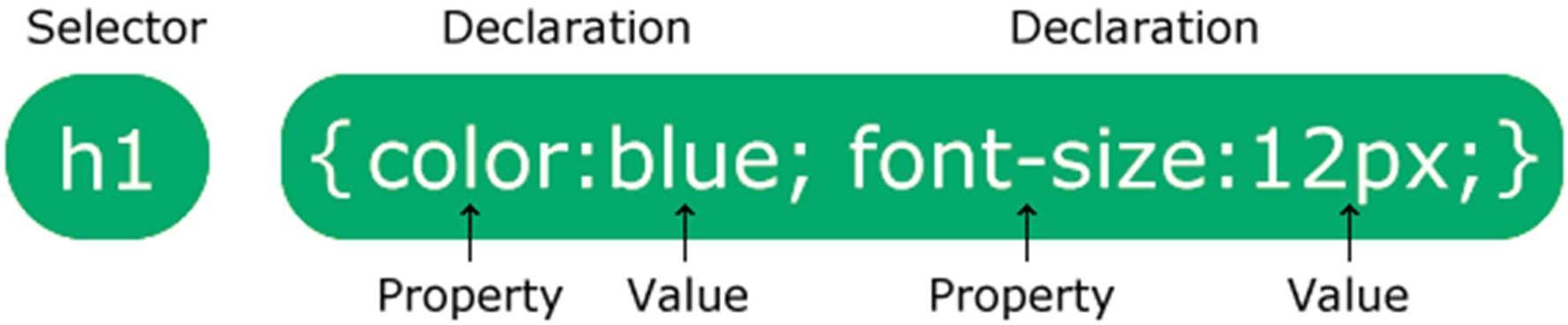
What style will be used when there is more than one style specified for an HTML element?

All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
3. Browser default

So, an inline style has the highest priority, and will override external and internal styles and browser defaults.

# CSS Syntax



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

## Example

In this example all <p> elements will be center-aligned, with a red text color:

p {

color: red;

text-align: center;

}

**Example Explained**

* p is a selector in CSS (it points to the HTML element you want to style:

<p>).

* color is a property, and red is the property value
* text-align is a property, and center is the property value

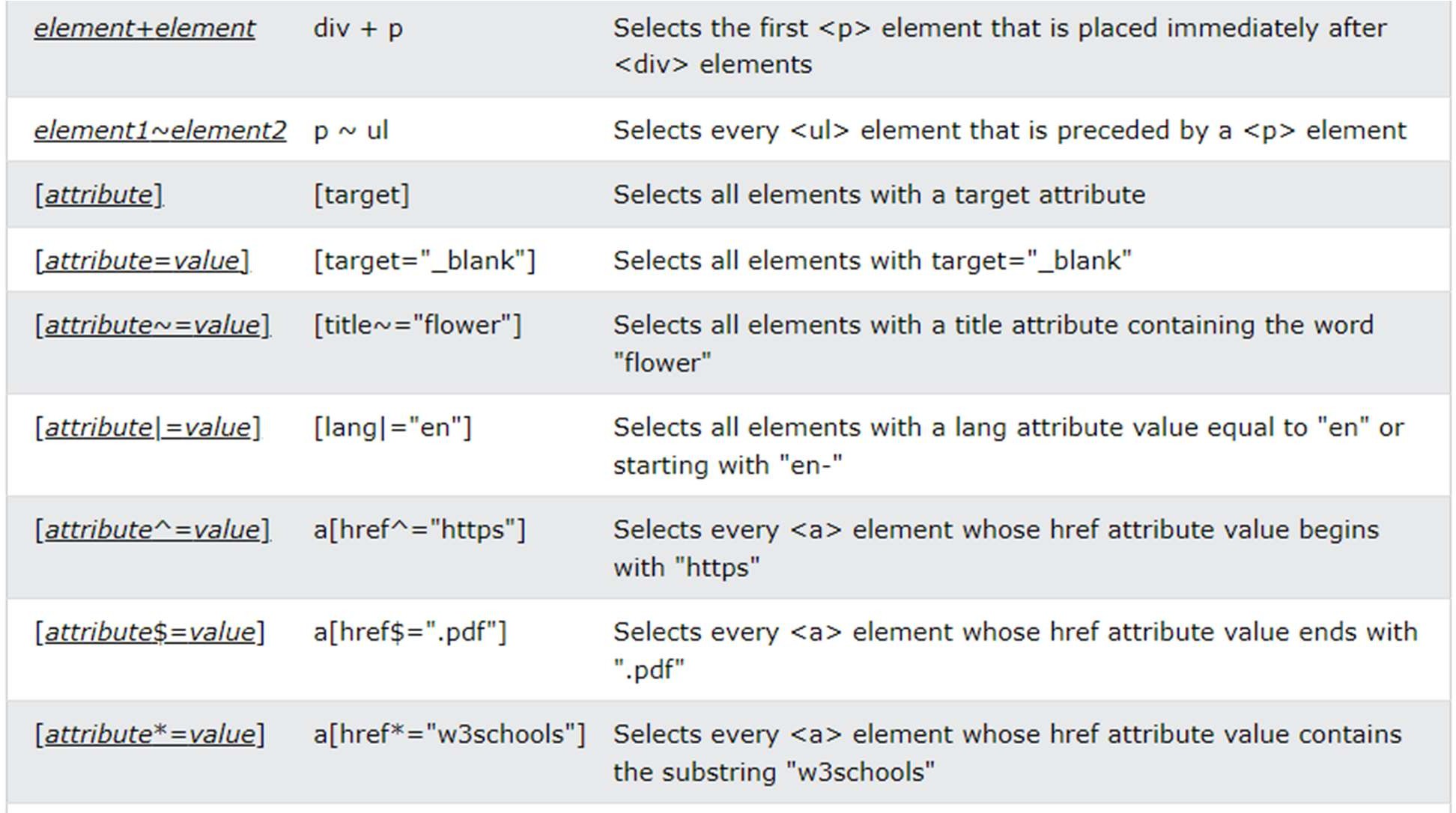
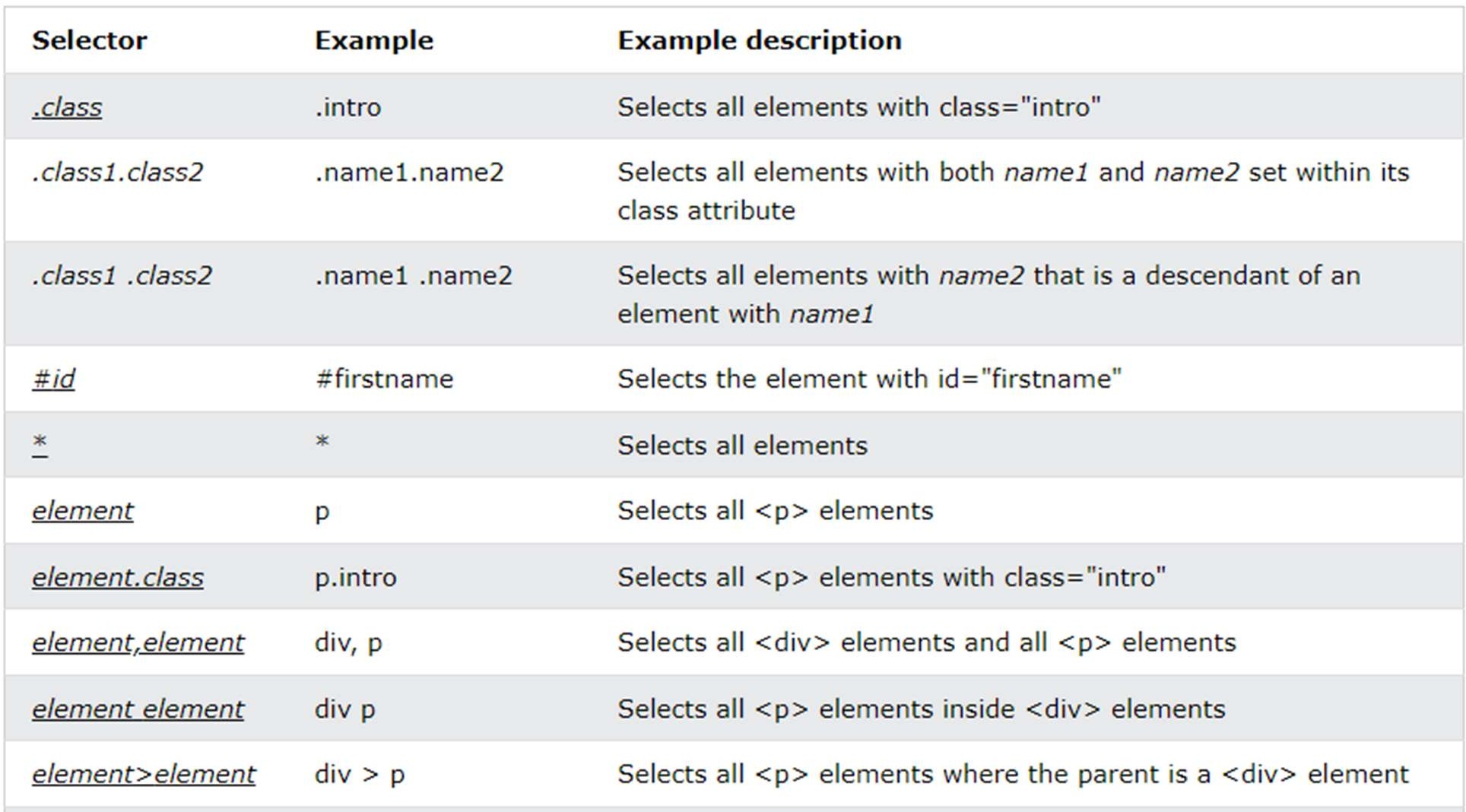
CSS selectors

The **CSS selectors** module defines the patterns to select elements to which a set of CSS rules are then applied along with their specificity. The CSS selectors module provides us with more than 60 selectors and five combinators. Other modules provide additional pseudo-class selectors and pseudo-elements.

In CSS, selectors are patterns used to match, or select, the elements you want to style. Selectors are also used in JavaScript to enable selecting the DOM nodes to return as a NodeList.

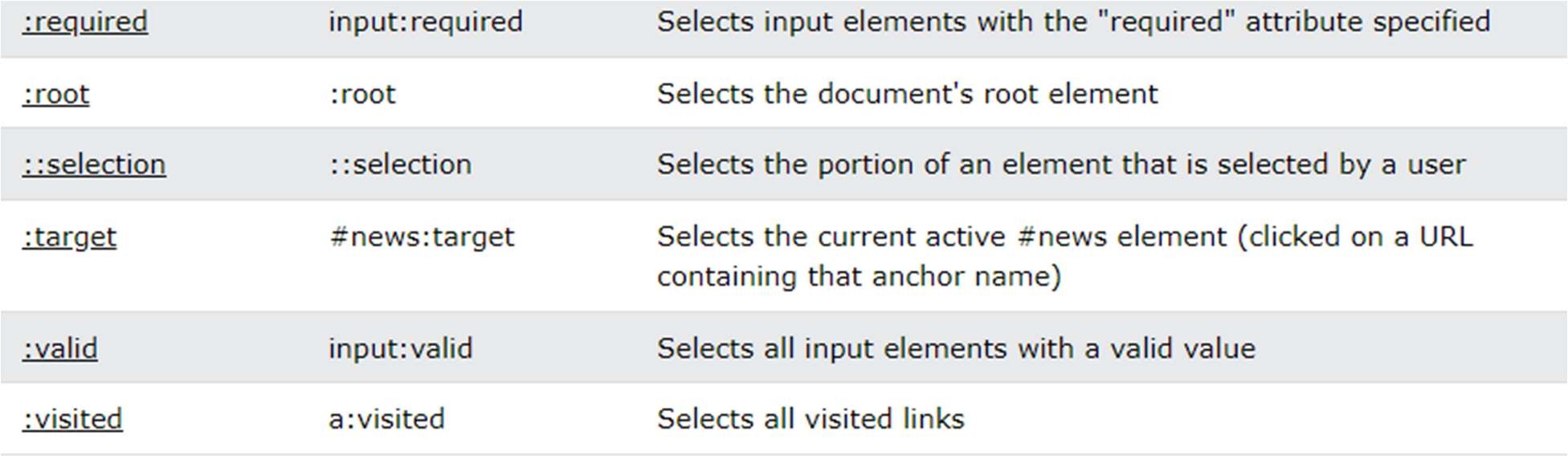
Selectors, whether used in CSS or JavaScript, enable targeting HTML elements based on their type, attributes, current states, and even position in the DOM. Combinators allow you to be more precise when selecting elements by enabling selecting elements based on their relationship to other elements.

CSS Selector









CSS Basic Properties

