**Element Selectors**

Elements of a document serer as the most basic selector.

**Declarations and Keywords**

The declaration block contains one or more declarations. A declaration is always formatted as a property followed by a colon and then a value followed by a semicolon.

These space-separated words are referred to as keywords because, taken together, they form the value of the property in question.

consider the following fictional rule:

rainbow: red orange yellow green blue indigo violet;

There is no such property as rainbow, but the example is useful for illustrative purposes. The value of rainbow is red orange yellow green blue indigo violet, and the seven keywords add up to a single, unique value.

**What if you want to more than one style to an element or group of elements?**

**Grouping Selectors:**

h2, p {color: gray;}

By placing the h2 and p selectors on the left side of the rule and separating them with a comma, you’ve defined a rule where the style on the right (color: gray;) applies to the elements referenced by both selectors.

**THE UNIVERSAL SELECTOR**

CSS2 introduced a new simple selector called the *universal selector*, displayed as an asterisk (\*). This selector matches any element at all, much like a wildcard. For example, to make every single element in a document red, you would write:

\* {color: red;}

Note that using semicolons at the end of each declaration is crucial when you’re grouping them. Browsers ignore whitespace in stylesheets, so the user agent must rely on correct syntax to parse the stylesheet.

**Class and ID Selector**

Useful when you want to style specific set of elements (certain number of elements of the same type)

.warning {font-weight: *bold*;}

Here elements with class “warning” are applied with font-weight of “bold”. There is a universal selector which is applied implicitly.

Note: The universal selector, represented by \*, is implied when an ID, class, attribute selector, pseudo-class or pseudo-element selector is written without being attached to an element selector.

As you can see, the class selector works by directly referencing a value that will be found in the class attribute of an element. This reference is always preceded by a period (.), which marks it as a class selector.

p.warning {font-weight: *bold*;}

The above select only those ‘p’ elements with class of ‘warning’.

## **Multiple Classes**

Now let’s say you want all elements with a class of warning to be boldfaced, those with a class of urgent to be italic, and those elements with both values to have a silver background. This would be written as follows:

.warning {font-weight: *bold*;}

.urgent {font-style: *italic*;}

.warning.urgent {background: silver;}

By chaining multiple class selectors together, you can select only those elements that have all class names (mentioned in the selector), in any order.

**ID Selector**

**Deciding between class and id**

You may assign classes to any number of elements IDs, on the other hand, should be used once, and only once, within an HTML document. IDs carry more weight than classes when styles are applied.

Also note that class and ID selectors may be case-sensitive, depending on the document language. HTML defines class and ID values to be case-sensitive, so the capitalization of your class and ID values must match that found in your documents.

On a purely syntactical level, the dot-class notation (e.g., .warning) is not guaranteed to work for XML documents. The hash-ID notation (e.g., #lead) will work in any document language that has an attribute that enforces uniqueness within a document. Uniqueness can be enforced with an attribute called id, or indeed anything else, as long as the attribute’s contents are defined to be unique within the document.

**Attribute Selectors:**

Some languages permit while other markup languages may not have ‘class’ or ‘id’ attributes on the element. To address this situation, CSS2 introduced attribute selectors, which can be used to select elements based on their attributes and the values of those attributes. There are four general types of attribute selectors: simple attribute selectors, exact attribute value selectors, partial-match attribute value selectors, and leading-value attribute selectors.

If you want to select elements that have a certain attribute, regardless of that attribute’s value, you can use a simple attribute selector.

h1[class] {color: silver;}

Attribute selectors are highly useful for XML documents