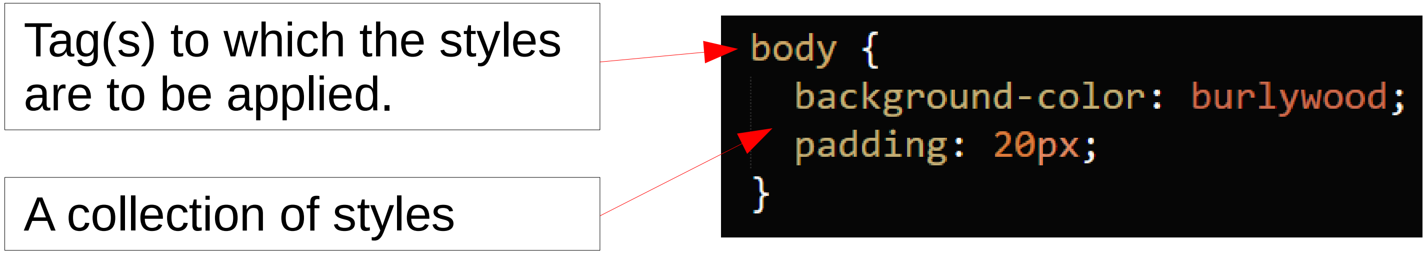
Cascading Style Sheets

Topics

- What is CSS?
- Rendering HTML
- Declarations and @Rules
- CSS Comments and Whitespace
- CSS Selectors
  - Element
  - ID
  - Class
  - Pseudo-class
- CSS Units
- Cascading & Inheritance

What is CSS?

CSS is a set of *properties* – styles that affect how HTML is rendered – and *selectors*, which determine to which tags these styles are applied.



Tag(s) to which the styles are to be applied.

A collection of styles

```css
body {
  background-color: burlywood;
  padding: 20px;
}
```

This combination of a selector and properties is called a CSS rule.

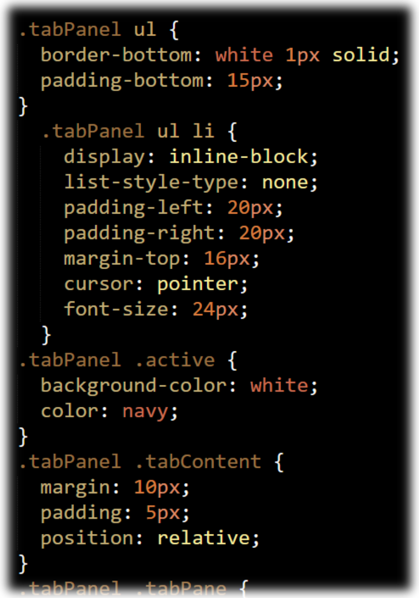
Cascading Style Sheets

Cascading Style Sheets are documents, typically with a .css extension, which can be linked to an HTML document.

```
<link type="text/css" href="scripts/layout/Layout.css" rel="stylesheet" />
```

Within the css document, we can define rules to be applied to tags within the HTML document.

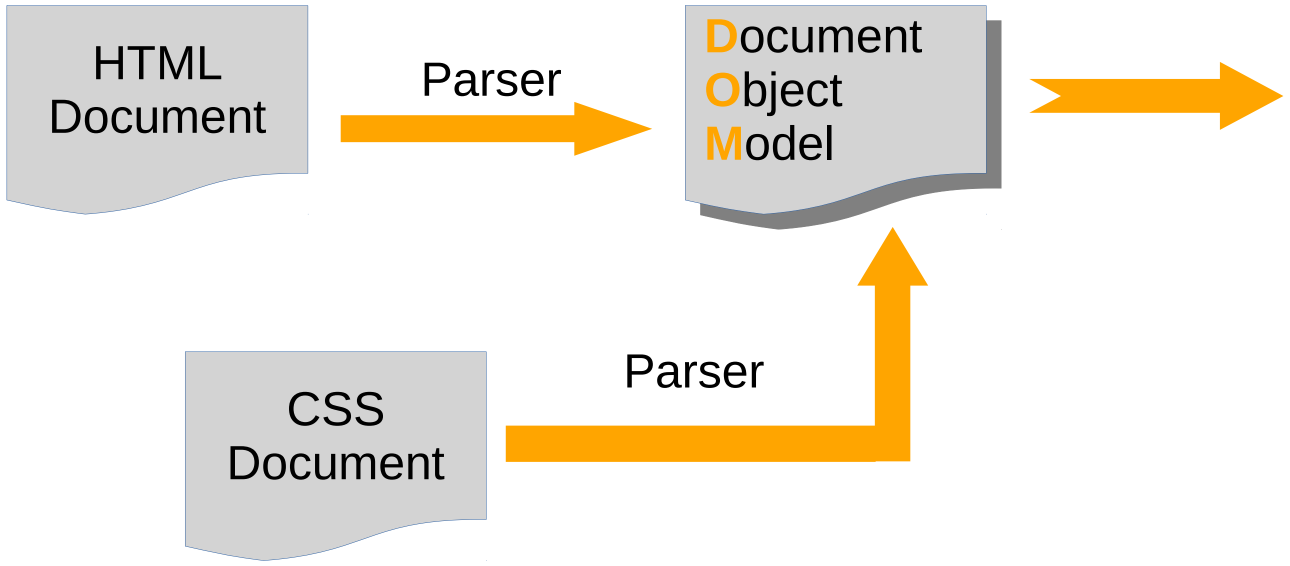
CSS rules can also be defined internally – within a style element of an HTML document.

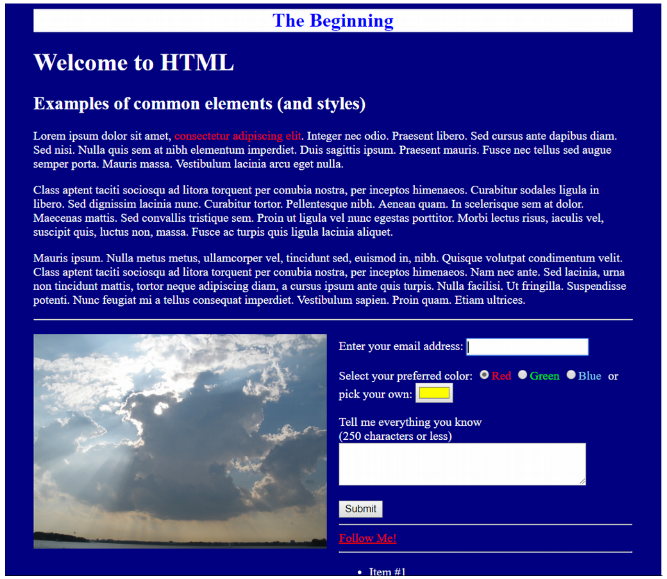


```
.tabPanel ul {
  border-bottom: white 1px solid;
  padding-bottom: 15px;
}
  .tabPanel ul li {
    display: inline-block;
    list-style-type: none;
    padding-left: 20px;
    padding-right: 20px;
    margin-top: 16px;
    cursor: pointer;
    font-size: 24px;
  }
.tabPanel .active {
  background-color: white;
  color: navy;
}
.tabPanel .tabContent {
  margin: 10px;
  padding: 5px;
  position: relative;
}
.tabPanel .tabPane {
```

platform
By Per Scholas

The HTML Rendering Process





The Beginning

Welcome to HTML

Examples of common elements (and styles)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero. Sed cursus ante dapibus diam. Sed nisi. Nulla quis sem at nibh elementum imperdiet. Duis sagittis ipsum. Praesent mauris. Fusce nec tellus sed augue semper porta. Mauris massa. Vestibulum lacinia arcu eget nulla.

Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Curabitur sodales ligula in libero. Sed dignissim lacinia nunc. Curabitur tortor. Pellentesque nibh. Aenean quam. In scelerisque sem at dolor. Maecenas mattis. Sed convallis tristique sem. Proin ut ligula vel nunc egestas porttitor. Morbi lectus risus, iaculis vel, suscipit quis, luctus non, massa. Fusce ac turpis quis ligula lacinia aliquet.

Mauris ipsum. Nulla metus metus, ullamcorper vel, tincidunt sed, euismod in, nibh. Quisque volutpat condimentum velit. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nam nec ante. Sed lacinia, urna non tincidunt mattis, tortor neque adipiscing diam, a cursus ipsum ante quis turpis. Nulla facilisi. Ut fringilla. Suspendisse potenti. Nunc feugiat mi a tellus consequat imperdiet. Vestibulum sapien. Proin quam. Etiam ultrices.

Enter your email address:

Select your preferred color: [✓ Red] Green Blue or pick your own: 🟨

Tell me everything you know
(250 characters or less)

Submit

Follow Me!

- Item #1

Rendered HTML

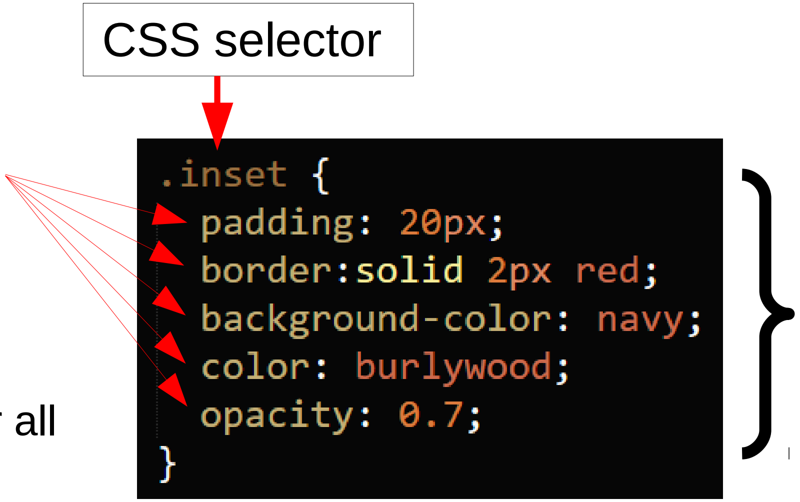
The **Document** **Object** **Model** (DOM) is the same tree structure that you work with using javascript.

CSS Declaration Blocks

CSS Declarations
Each declaration consists of:

Property: Value;

The semicolon is required for all but the last style.



```
.inset {
  padding: 20px;
  border:solid 2px red;
  background-color: navy;
  color: burlywood;
  opacity: 0.7;
}
```

CSS declarations grouped within a pair of curly braces are a **Declaration Block.**

A declaration block paired with one or more CSS selectors is a **CSS Rule**.

@ Rules

CSS defines additional syntax called “@ Rules”. Some of the more common @ rules are:

• @import	imports an additional css document
• @supports	applies rule(s) only if a browser supports a feature
• @media	applies rule(s) only if a device matches a condition

@media is used to provide alternative styling for phone, tablet, and laptop screens.

CSS Comments and Whitespace

CSS comments are enclosed with /* and */.

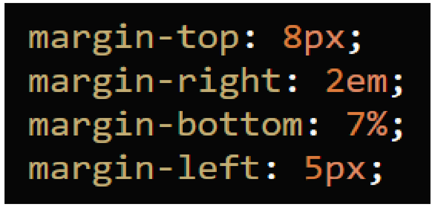
As with any language, CSS comments should be used sparingly to clarify intent.

Within the syntax described above, CSS ignores all whitespace.

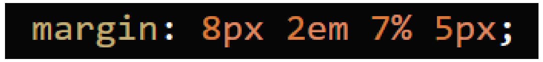
Whitespace should be used to make CSS as readable as possible.

CSS Shorthand Properties

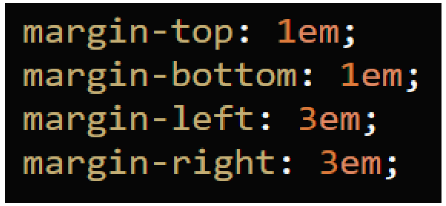
Some of the CSS properties support short-hand. For example, the declarations on the left and the ones on the right are identical:



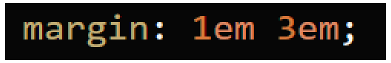
```
margin-top: 8px;
margin-right: 2em;
margin-bottom: 7%;
margin-left: 5px;
```



```
margin: 8px 2em 7% 5px;
```



```
margin-top: 1em;
margin-bottom: 1em;
margin-left: 3em;
margin-right: 3em;
```



```
margin: 1em 3em;
```

The padding property supports similar shorthand tblr syntax.

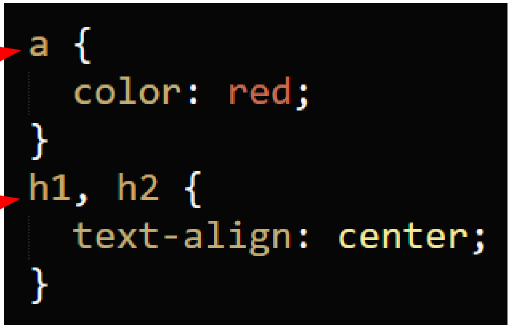
The border, font and background properties also provide shorthand, though their syntax is quite different from that of margin and padding.

CSS Element Selectors

The simplest selectors are the element (or "type") selectors. These apply properties to all elements in the document that are of a given type:

This declaration block will apply to all <a> tags in the document.

This declaration block will apply to all <h1> and <h2> tags in the document.

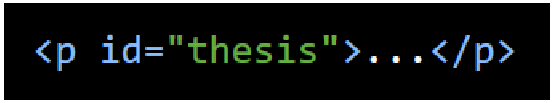


```
a {
  color: red;
}
h1, h2 {
  text-align: center;
}
```

Notice that multiple elements may be listed separated by commas.

ID Selectors

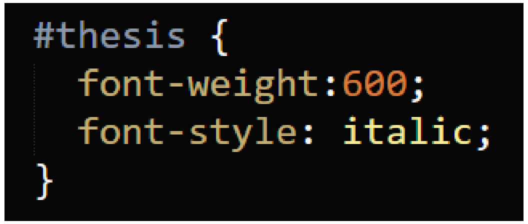
These selectors pertain to elements that have been given an *id* attribute:



```html
<p id="thesis">...</p>
```

The *id* attribute is special in HTML – its value must be unique within the document. For example, the value “thesis” used above must be assigned to only one element.

Having used this *id*, we can define a style to apply specifically to that element using the syntax “#idname”:



```css
#thesis {
  font-weight:600;
  font-style: italic;
}
```

Class Selectors

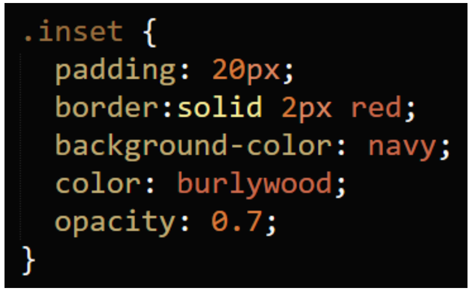
We often need to define css declarations that we can apply to elements *a la carte*. We can do this using class selectors and the class attribute. A class selector is defined using '.' followed by the class name.

In this example, we define a class selector ".inset" and apply it to two different elements.

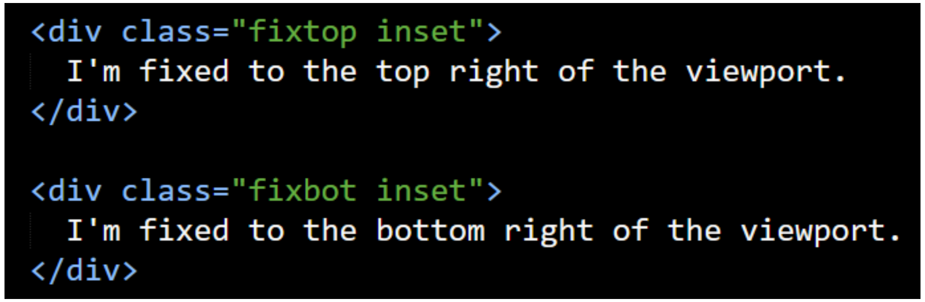
Notice that the class attribute can take multiple values, each separated by a space.

Notice also that the "." is not used within the class attribute.

This allows us to mix styles to elements *a la carte*.



```css
.inset {
  padding: 20px;
  border:solid 2px red;
  background-color: navy;
  color: burlywood;
  opacity: 0.7;
}
```



```html
<div class="fixtop inset">
  I'm fixed to the top right of the viewport.
</div>

<div class="fixbot inset">
  I'm fixed to the bottom right of the viewport.
</div>
```

Attribute Selectors

Attribute selectors select elements based on their attributes and attribute values.

These are less common than the previously described selectors, but can be used to great affect and can be used alongside custom “data-” attributes.

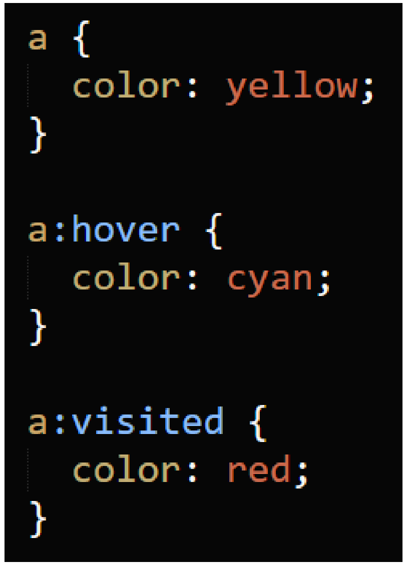
For further information, readers are referred to the MDN online tutorial.

Pseudo-class Selectors

Elements often exist in more than one state. For example, a link may have been recently visited or never visited, or it might have a mouse hovering over it or not. These element states are differentiated in CSS using pseudo-classes:

In this example, a single <a> element has a different color style applied depending on whether it is:

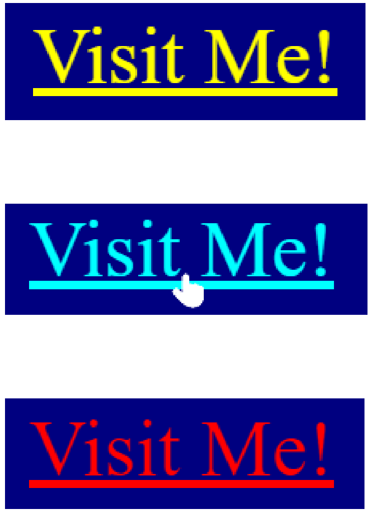
- In its default state (unvisited);
- Has a mouse over it;
- Previously clicked ("visited")



```
a {
  color: yellow;
}

a:hover {
  color: cyan;
}

a:visited {
  color: red;
}
```



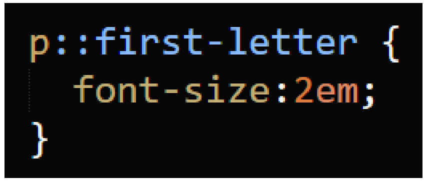
Common Pseudo-classes

The list of pseudo classes is numerous. Here are a few of the more common ones:

Pseudo-class	Selects element(s) when...
:active	Element is being activated (clicked) by the user
:checked	Checkbox or radiobutton is ‘on’.
:disabled / :enabled	Element is unable / able to accept focus or interact
:first-child / :last-child	Element is the first / last among siblings
:focus	Element has focus
:nth-child(n)	Element is an nth child among siblings (includes even/odd).

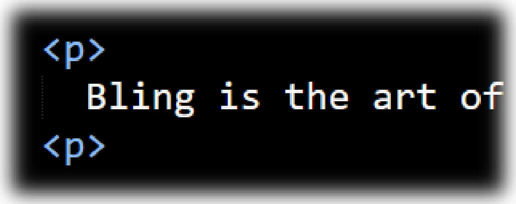
Pseudo-elements

Pseudo-elements are keywords applied to the end of selectors using a double-colon (::). They select specific parts of an element. One common use is to enhance the first letter of paragraphs:



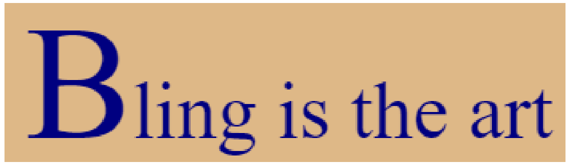
```
p::first-letter {
  font-size:2em;
}
```

+



```
<p>
  Bling is the art of
<p>
```

=



Bling is the art

Pseudo-elements can be used to insert new content around the selected elements – thus the name “pseudo-element”. Some interesting examples of this technique are available online.

Combining Selectors - Combinators

Name	Syntax	Selects
Group of selectors	A, B	Any element matching A and/or B (see Group of selectors on one rule, below - Group of Selectors is not considered to be a combinator).
Descendant selector	A B	Any element matching B that is a descendant of an element matching A (that is, a child, or a child of a child, etc.).
Child selector	A > B	Any element matching B that is a direct child of an element matching A.
Adjacent sibling selector	A + B	Any element matching B that is the next sibling of an element matching A (that is, the next child of the same parent).
General sibling selector	A ~ B	Any element matching B that is one of the next siblings of an element matching A (that is, one of the next children of the same parent).

From MDN Web Docs

CSS Units

Many CSS properties require entry of properties that have units (lengths), have values in specific domains (colors) or require precise formats (urls). There are excellent tutorials on these subjects available online (e.g. MDN and TutorialsPoint), and this document provides only a brief overview of the most commonly used units.

CSS Lengths

Lengths are used by width, height, margin, padding, border-width, font-size and text-shadow properties. Some properties can be negative (e.g., margin).

Unit	Example	Type	Description
px	width: 200px;	absolute	1px = 1/96th inch. px is device-dependent.
em	margin: 1em;	relative	Represents the inherited font-size of the element.
rem	font-size: 2rem;	relative	Represents the font-size of the root element, which is determined by a user's browser preferences. This length is fully-scalable.
%	margin: 5%;	relative	Defines a length relative to the parent element.

CSS Colors

Colors are used by color, background-color and border-color properties, and colors are used within gradient definitions. CSS also defines a *currentcolor* keyword.

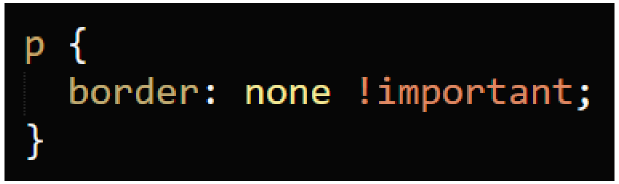
Syntax	Example	Description
name	color: red;	Modern CSS defines over 140 named colors. The color name transparent is also recognized. Names are case-sensitive.
hex	color: #f00; or color: #ff0000; or color: #ff0000ff;	The '#' precedes the hexadecimal rgb(a) value. The 3- and 6-character colors are rgb. The 8-character color includes the alpha channel.
rgb, rgba	color: rgb(255,0,0); or color: rgba(255,0,0,0.5);	Uses the rgb function to convert r, g, and b values (either 0-255 integer values or % values) to color channels.
hsl, hsla	color: hsl(0, 100%, 50%); or color: hsla(0, 100%, 50%, 0.5);	Uses the hsl or hsla function to convert hue, saturation (%) and lightness (%) values to color.

Cascading and Inheritance

The *cascade* in CSS refers to the rules by which styles are applied to elements. These rules account for 3 factors:

- Importance - has a style been marked *!important* ?
- Specificity - how *specific* is the selector for the element?
- Source Order- later-defined styles win

A dirty secret of CSS is that styles can be marked **!important**, meaning that they will override the normal cascading rules and always be applied:



```
p {
  border: none !important;
}
```

!important is effectively a hack, and should not be used except in the rare situation where you cannot edit a core CSS module.

platform
By Per Scholas

Selector Specifity

The second factor CSS considers in applying styles is the selector specificity. Here are the specificities:

Selector Type	**Specificity**	**Details**
Element Type	Lowest	Element selectors apply to all elements of a given type – not very specific.
Class	Medium	Class selectors apply only to elements intentionally given the appropriate class attribute – pretty specific.
ID	High	ID selectors are highly specific because they can apply to only one element in the document.
Inline style attribute	Highest	Styles defined inline on the element override all others (except !important).

platform
By Per Scholas

Source Order

If two styles are tied with respect to importance and specificity, which one “wins”?

Whichever style is defined latest in the source code – the .css document or the .html document – will be applied.

Note that styles are applied individually, not solely as declaration blocks attached to their selectors. This allows styles from lower-specificity selectors to mix and match with other styles applied via higher-specificity selectors.

Inheritance

For some CSS properties, it is natural to inherit that property from a parent element – color and font, for example. For other properties, automatic inheritance would create a mess – margin, padding, background-image, etc.

CSS provides four special property values related to inheritance:

inherit	Sets the property value applied to a selected element to be the same as that of its parent element.
initial	Sets the property value applied to a selected element to be the same as the value set for that element in the browser's default style sheet.
unset	Resets the property to its natural value, which means that if the property is naturally inherited it acts like inherit, otherwise it acts like initial.
revert	Reverts the property to the value it would have had if the current origin had not applied any styles to it.

Topics

- What is CSS?
- Rendering HTML
- Declarations and @Rules
- CSS Comments and Whitespace
- CSS Selectors
  - Element
  - ID
  - Class
  - Pseudo-class
- CSS Units
- Cascading & Inheritance