Cascading Style Sheets (CSS)

CSS (Cascading Style Sheets) is the stylesheet language used alongside HTML to format and set the style of HTML elements. Like HTML, there are several versions of CSS, and each subsequent version introduces a new set of capabilities that can be used for formatting HTML elements. Browsers are updated alongside it to support these new features.

Example

At a fundamental level, CSS is used to define the style of each class or type of HTML elements (i.e., `body` or `h1`), such that any element within that page would be represented as defined in the CSS file. This could include the font family, font size, background color, text color and alignment, and more.

Code: css

```
body {
  background-color: black;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: helvetica;
  font-size: 10px;
}
```

As previously mentioned, this is why we may set unique IDs or class names for certain HTML elements so that we can later refer to them within CSS or JavaScript when needed.

Syntax

CSS defines the style of each HTML element or class between curly brackets `{}`, within which the properties are defined with their values (i.e. `element { property : value; }`).

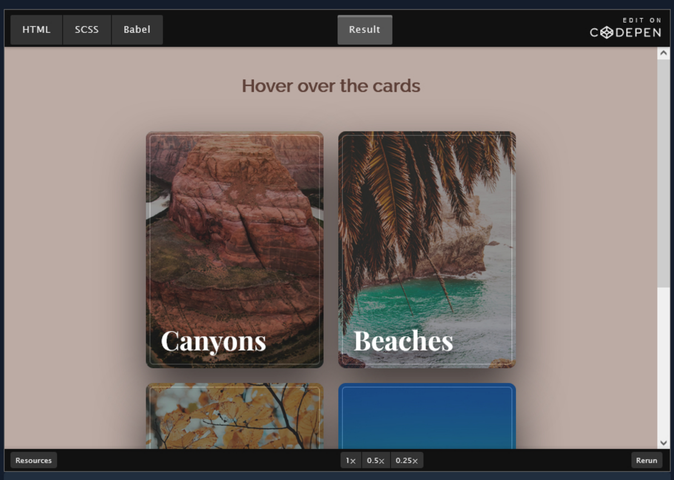
Each HTML element has many properties that can be set through CSS, such as `height`, `position`, `border`, `margin`, `padding`, `color`, `text-align`, `font-size`, and hundreds of other properties. All of these can be combined and used to design visually appealing web pages.

CSS can be used for advanced animations for a wide variety of uses, from moving items all the way to advanced 3D animations. Many CSS properties are available for animations, like `@keyframes`, `animation`, `animation-duration`, `animation-direction`, and many others. You can read about and try out many of these animation properties here.

Usage

CSS is often used alongside JavaScript to make quick calculations, dynamically adjust the style properties of certain HTML elements, or achieve advanced animations based on keystrokes or the mouse cursor location.

The following example beautifully illustrates such capabilities of CSS when used with HTML and JavaScript "Parallax Depth Cards - by Andy Merskin on CodePen":



You can click on the tabs on the top-left to view the source code.

This shows that even though HTML and CSS are among the most basic cornerstones of web development when used properly, they can be used to build visually stunning web pages, which can make interacting with web applications a much easier and more user-friendly experience.

Furthermore, CSS can be used alongside other languages to implement their styles, like `XML` or within `SVG` items, and can also be used in modern mobile development platforms to design entire mobile application User Interfaces (UI).

Frameworks

Many may consider CSS to be difficult to develop. In contrast, others may argue that it is inefficient to manually set the style and design of all HTML elements in each web page. This is why many CSS frameworks have been introduced, which contain a collection of CSS style-sheets and designs, to make it much faster and easier to create beautiful HTML elements.

Furthermore, these frameworks are optimized for web application usage. They are designed to be used with JavaScript and for wide use within a web application and contain elements usually required within modern web applications. Some of the most common CSS frameworks are:

- Bootstrap
- SASS
- Foundation
- Bulma
- Pure

Questions

Answer the question(s) below to complete this Section and earn cubes!

+1 What is the CSS "property: value" used to make an HTML element's text aligned to the left?

Submit your answer here... Submit Hint

Previous Next

Go to Questions

Table of Contents

Introduction to Web Applications
Introduction
Web Application Layout
Front End vs. Back End

Front End Components
HTML
Cascading Style Sheets (CSS)
JavaScript

Front End Vulnerabilities
Sensitive Data Exposure
HTML Injection
Cross-Site Scripting (XSS)
Cross-Site Request Forgery (CSRF)

Back End Components
Back End Servers
Web Servers
Databases
Development Frameworks & APIs

Back End Vulnerabilities
Common Web Vulnerabilities
Public Vulnerabilities

Next Steps
Next Steps