I-Novotek Academy

Full Stack Web Development Course PDF Guide

website: www.inovotekacademy.com

Youtube: i-novotek academy

Whatsapp: +233596038326/+8613051806737

Course link: https://www.udemy.com/course/fullstack-web-development-course-projects-base/?referralCode=F8C808368D020D5794BD

CSS COURSE HANDBOOK

What is CSS?

CSS, short for cascading style sheets, is a code language used to create the visual look of web pages. It's what makes a website pretty, instead of just being a bunch of boring text. CSS can be used to control the colour, font, size, spacing, and layout of website content.

CSS Syntax

The CSS syntax consists of a set of rules. These rules are made up of selectors and declarations.

Selectors
Selectors are the part of CSS syntax that determines which HTML element(s) will be affected by the CSS rule. For example, if you want to make all of the paragraph elements on your web page red, you would use the p selector.

Declarations

Declarations are the part of CSS syntax that contains the actual style rules that will be applied to the selected element(s). Each declaration includes a CSS property and a value for that property. In our example above, the colour property would be set to red.

The Different Types of CSS

There are three different types of CSS:

1. Inline CSS

2. External CSS

3. Internal CSS

Inline CSS

Inline CSS is the simplest way to add style information to your website. It's added within HTML tags using the style attribute. This type of CSS is good for small changes but can be hard to maintain because it's mixed in with HTML code.

Internal CSS

Internal CSS is used when you want to apply styling rules to just one specific HTML page. It's placed in between the <head> tags of an HTML document, using the <style> tag. Like inline CSS, internal styles can make your code harder to maintain since they mix styling information with HTML content.

External CSS

External CSS is written in a separate CSS file, saved with a .css extension. The file is then linked to your HTML document using the <link> tag. This method keeps your HTML and CSS code separate, making it easier to manage and update your website styling.

Which One Should You Use?

For small websites with just a few pages, inline CSS or internal CSS is usually sufficient. However, as your website grows, you’ll probably want to switch to using an external stylesheet since it’s much easier to manage.

The CSS Box Model: An Ultimate Guide

Introduction :

The CSS Box Model is one of the core concepts in Web Design and Development. It encompasses everything that makes up a design, from headers, footers and main content areas, to margins, paddings and borders.

Everything on a web page is a box, and each box has its own properties that can be styled with CSS. In this section, we'll cover the basics of the CSS Box Model and how you can use it to better understand your web pages.

What is the CSS Box Model?

The CSS Box Model is a model for describing the boxes that make up a web page. Each box has its own properties, which include width, height, padding, border and margin. By understanding these properties and how they work together, you can create complex designs with ease.

Each box has its own properties that can be styled with CSS. The width and height properties define the size of the box, while the padding, border and margin properties add space around the box. Let's take a look at each of these properties in more detail.

Width and Height Properties

The width and height properties are used to set the size of a box. The width property defines the horizontal size of the box, while the height property defines the vertical size.

Padding Properties

Padding is the space between the content of a box and its border. By default this spacing is empty, but it can be customized using the following css padding properties:

Padding-Top - This property adds space between the top edge of a box and its content.

Padding-Right - This property adds space between the right edge of a box and its content.

Padding-Bottom - This property adds space between the bottom edge of a box and its content.

Padding-Left - This property adds space between the left edge of a box and its content.

Border Properties

Border Properties Borders are used to draw lines around boxes (around their paddings) in order to further stylize them with CSS.

Border-Top - This property adds a line along the top margin of a box.

Border-Right - This property adds a line along the right margin of a box.

Border-Bottom - This property adds a line along the bottom margin of a box.

Border-Left - This property adds a line along the left margin of a box.

Margin properties

The margin properties adds space between a box and the content of the next element.

The margin-right property adds space between the right edge of a box and the content of the next element.

The margin-bottom property adds space between the bottom edge of a box and the content of the next element.

The margin-left property adds space between the left edge of a box and the content of the next element.

CSS Selectors

In CSS, selectors are used to targeting elements on a page so that they can be styled. There are a variety of different selectors available, each with its advantages and disadvantages. In this article, we'll take a look at some of the most popular CSS selectors and how they can be used to target elements on a page.

CSS selectors are an important part of any web developer's toolkit, so it's important to understand how they work. With that in mind, let's dive into the types of selectors

We can divide CSS selectors into five categories:

1. Simple selectors
2. Combinators
3. Pseudo-classes
4. Attribute selectors
5. Specificity

```
}
```

Simple selectors

- The element selector
- The CSS class Selector
- The CSS id Selector

The element selector

The element selector selects HTML elements based upon the element name. This is useful for targeting specific elements on a page for styling or other modifications.

HTML

```
   <h1>Element Selectors</h1>
    <hr />
    <h2>How to learn CSS</h2>
    <p>Follow tutorials</p>
    <div>Follow online resources</div>
```

CSS

```
p {
  color: red;
}
div {
  color: blue;
}
```

```
h2 {
  color: green;
}
```

The CSS class Selector

CSS class selectors are used to select elements based on their class attribute. Class selectors are written using a period followed by the class name. To select multiple elements with different class names, separate the class names with a space.

HTML

```
 <h1>Class Selectors</h1>
    <hr />
    <h2 class="title">How to learn CSS</h2>
    <p class="description">Follow tutorials</p>
    <div class="description content">Follow online resources</div>
```

CSS

```
.title {
  color: #fff;
  background: #000;
}
.description {
  color: blue;
}
```

The CSS id Selector

The id selector uses the id attribute to select a specific element from an HTML document. The id value of an element is unique within the document, meaning that only one element can have that id value. To select an element using its id value, you must write a # character followed by the id value.

CSS Combinators

- descendant selector (space)
- child selector (>)
- adjacent sibling selector (+)
- general sibling selector (~)

descendant selector (space)

The descendant selector allows you to select all elements that are descendants of a specified element. This can be useful when you want to apply styles to all child elements of a particular element.

HTML

```
<h2>Descendant Selectors (spacer)</h2>

    <hr />

    <div>

      <h2 class="title">How to learn CSS</h2>

      <p class="description">Follow tutorials</p>
      <div class="description content">

        <p>Follow online resources</p>

      </div>

    </div>

    <p>Self taught approach</p>
```

CSS

```
div p {

  color: red;

}
```

Child Selector (>)

The child selector is used to select all elements that are the direct children of a specified element.

HTML

```
<h2>Child Selectors</h2>

    <hr />

    <div>

      <h2>How to learn CSS</h2>

      <p>Follow tutorials</p>

      <footer>

        <p>Follow online resources</p>

      </footer>

    </div>

    <p>Self taught approach</p>
```

CSS

```
div > p {
  color: red;
}
```

Adjacent Sibling Selector (+)

The adjacent sibling selector is used to select an element that is immediately following another specified element.

HTML

```
 <h2>Adjacent Selectors</h2>
    <hr />
    <div>
      <h2>How to learn CSS</h2>
      <p>Follow tutorials</p>
      <footer>
        <p>Follow online resources</p>
      </footer>
    </div>
    <p>Self taught approach</p>
    <div>
      <p>Generl taught</p>
    </div>
    <p>You need to be dettermined</p>
```

CSS

```
div + p {
  color: red;
}
```

what is Sibling elements in css?

Sibling elements in css are those that share the same parent element.

General Sibling Selector (~)

The general sibling selector allows you to select all elements that are next siblings of a specified element.

HTML

```
<h2>General Sibling Selector</h2>

    <hr />

    <p>This wont take any effect</p>

    <div>

      <h2>How to learn CSS</h2>

      <p>Follow tutorials</p>

      <footer>

        <p>Follow online resources</p>

      </footer>

    </div>

  <p>Self taught approach</p>

  <p>Generl taught</p>

    <p>You need to be dettermined</p>
```

CSS

```
div ~ p {

  color: red;
}
```

What are Pseudo-classes?

A pseudo-class is used to define a special state for an element. This state can be used to style the element differently than when it is in its default state.

Pseudo-classes example
https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes

HTML

```
<h2>Pseudo-classes styling</h2>

  <hr />

    <p>

      <a href="http://www.w3.org/TR/html"> View video </a>

  </p>

    <div>

      Lorem ipsum dolor sit amet consectetur adipisicing elit. Earum qui sunt

      quasi velit exercitationem, fuga excepturi odio natus corrupti aut nihil

      ipsum voluptatem repellendus dolorum nesciunt assumenda quam modi? Quidem!

    </div>
```

CSS

```
/* not visited */

a:link {

  color: red;

}
```

```
/* visited */
a:visited {
  color: blue;
}
/* mouseover */
a:hover {
  color: green;
}
/* active */
a:active {
  color: orange;
}
div:hover {
  background-color: blue;
}
```

What are Pseudo-Elements?

A CSS pseudo-element is used to target specific parts of an element for styling.

OR

A pseudo-element is a keyword added to a selector that lets you style a specific part of the selected element(s).

An example of how this can be used is to:

-Apply styling to

- the first letter, or line, of an element
- -Insert content before, or after, the content of an element

NOTE

The ::first-line pseudo-element can only be applied to block-level elements. This means that it cannot be used on elements such as <span> or <em>.

BLOCK LEVEL ELEMENTS

A block-level element will take up the entire width of its parent element, and as tall as its content. This creates a "block."

Example

footer

form

h1 - h6

header

hr

li

div

main

nav

ol

p

section

table

ul

video

address

article

aside

blockquote

Some properties that can have effect on the ::first-line pseudo-element:

- word-spacing
- letter-spacing
- text-decoration
- vertical-align
- text-transform
- line-height
- font properties
- color properties
- background properties

HTML

```
 <h2>Pseudo-Elements styling</h2>

    <hr />

    <p>

      Lorem ipsum dolor sit amet consectetur, adipisicing elit. Neque commodi

      quam ullam eveniet! Ex molestias ullam, excepturi voluptatem, vel fugiat

      nesciunt asperiores, suscipit aut quis provident consequatur

      necessitatibus facilis corrupti.
    </p>

    <h1>Inovotek Academy first line</h1>

    <div>

      Lorem ipsum dolor sit amet consectetur adipisicing elit. Earum qui sunt

      quasi velit exercitationem, fuga excepturi odio natus corrupti aut nihil

      ipsum voluptatem repellendus dolorum nesciunt assumenda quam modi? Quidem!

    </div>

  <h3>Inovotek Academy First letter</h2>


      <!-- pseudo element + class-->

      <div class="container">

        <p>

          Lorem ipsum dolor sit amet consectetur adipisicing elit. Neque commodi

          quam ullam

      </div>
```

CSS

```
h1::first-line {

  color: red;

  font-size: 20px;

  text-decoration: dashed;

  background-color: pink;

  letter-spacing: 6px;

  line-height: 20px;

  text-transform: uppercase;
}

/* */

h3::first-letter {
  color: red;
  font-size: 90px;
  text-decoration: dashed;
  letter-spacing: 6px;
  line-height: 20px;
  text-transform: uppercase;
}

.container p::first-letter {
  color: rgb(255, 0, 166);
  font-size: 100px;
}
```

::before Pseudo-element

The before pseudo-element h as the ability to insert content before the content of an element.

::after Pseudo-element

The after pseudo-element h as the ability to insert content after the content of an element.

HTML

```
 <h2>Pseudo-before after</h2>
```

```
    <hr />
    <h3>Inovotek Academy </h2>
```

CSS

```
h3::before {
  content: "  How are you? -";
}
h3::after {
  content: " The best coding school";
}
```

CSS Specificity Hierarchy

There are four categories that determine the specificity level of a selector:

1. **Inline styles** - <p style="background: red;">
2. **IDs** - Example: #navbar
3. **Classes, pseudo-classes, attribute selectors** - .container, :hover, [href]
4. **Elements and pseudo-elements** - : h3, :after

The Ultimate Guide To Using CSS Colors

1. Introduction

Color is an important part of web design. It can set the mood of a website, and it can be used to draw attention to important elements. CSS offers a wide variety of color options, and in this guide we'll explore how to use them effectively.

What are the different types of color values?

There are three types of color values:

H**exadecimal Color Value**

Hexadecimal values are six-digit codes that represent a specific color. They are also known as hex codes.

The first two digits represent the amount of red in the color, the second two digits represent green, and the last two digits represent blue. White is represented as FFFFFF, while black is 000000.

Examples of Hexadecimal Colors

```
FFFFFF (white)
```

```
000000 (black)
```

```
0000FF (blue)
```

```
008000 (green)
```

RGB Colors

RGB values are three numbers that represent a color's red, green, and blue components.

These values can range from zero to one or be represented within a 0-255 scale. When these values are combined, they produce over 16 million possible colors.

Example :

```
rgb(0,0,1) - bright blue
rgb(255, 255, 150) - light yellow Advantages:
 rgb(1, 0, 0) is red
 rgb(0, 1, 0) is green
 rgb(0, 0, 1) is blue
```

RGBA Colors in css

RGBA colors are colors that contain a red, green, blue, and alpha (transparency) component. These colors can be used on web pages and in other graphics applications.

The red, green, and blue components of an RGBA color are each represented by a number between 0 and 255. The alpha component is represented by a number between 0 and 1.0.

```
rgba(255, 99, 71, 0.6)
```

HSL Colors

HSL values are three numbers that represent a color's hue (ranging from 0-360°), saturation (percentage of how much a color is diluted with white light), and lightness (percentage of how much a color is diluted with black light). The number one corresponds to 100% and the number zero corresponds to 0%. Combined they produce over 16 million possible colors.

Hue: the shade or tint of a color. For example, if you took pure blue and added yellow to it you would get green.

Saturation: the percentage of gray in a particular color. The higher the saturation percentage, the more vivid and intense a color appears.

Lightness: The brightness or darkness of a color. A lower percentage means a darker color while a higher percentage results in a lighter hue

HSL values are three numbers that represent a color's hue, saturation, and lightness.

Example

:

```
    hsl(0, 100%, 50%) is red

    hsl(120, 100%, 50%) is green

    hsl(240, 100%, 50%) is blue
```

A Comprehensive Guide To Understanding And Using CSS Units

What are CSS units?

CSS units are the measure of length used in Cascading Style Sheets (CSS).

There are two main types of units:

absolute units

relative units

Absoute units

Absolute units are fixed in relation to each other and never change.

Relative units

Relative units are based on a percentage or em, which is a unit of measurement based on the width of the letter "M" in the current font size.

What are the different types of CSS units?

There are four different types of CSS units: px, %, em, and rem. Pixels (px) are the most common type of CSS unit. They're used for everything from font sizes to positioning elements on a web page. percentages (%) are another common type of CSS unit. They're often used for responsive design, where elements resize depending on the width of the screen they're being viewed on. Ems (em) and root ems (rem) are less

ommon but still widely-used CSS units. Ems are relative to the font size, so they can be used to make sure text scales correctly on different screen sizes

Relative length units

Ems (em)

- Relative to the font-size of the element (2em means 2 times the size of the current font)

root ems (rem)

- Relative to fon-size of the root element (

<html>)

vw

- Relative to 1% of the width of the viewport*

vh

- Relative to 1% of the height of the viewport*

%

-

Relative to the parent element

what is root element in html? A root element is the first element in an HTML or XML document.