How does inline-block work, and how does it differ from inline and block elements?

When working with CSS, you often encounter three different types of display behaviors for elements: inline, block, and inline-block.

Each of these properties affects how elements are positioned and how they interact with other elements on the page.

In this lecture, we will focus on how the inline-block property works and how it differs from both inline and block-level elements.

Block-level elements behave like large containers or "blocks" that take up the full width of their parent container. They always start on a new line, and their height and width can be adjusted.

Inline elements only take up as much space as they need. They flow within the surrounding content and do not break onto a new line.

The inline-block property is a hybrid of these two behaviors. Like inline elements, inline-block elements remain in the text flow without starting on a new line.

However, unlike inline elements, you can adjust the width and height of an inline-block element, just as you would with block-level elements.

In short, the key difference between inline and inline-block is that inline elements cannot have their size controlled, whereas inline-block elements allow for full control over dimensions while still staying inline with other content.

Let's take a look at an example.

<div class="container">

<span class="inline-block-element element1">Inline-Block Element 1</span>

<span class="inline-block-element element2">Inline-Block Element 2</span>

</div>

In the above example, we have a div with a class of container. Inside that div element, we have two span elements.

Here is the accompanying CSS:

.inline-block-element {

display: inline-block;

width: 150px;

height: 100px;

}

.element1 {

background-color: lightblue;

}

.element2 {

background-color: lightgreen;

}

Each of the span elements is set to display:inline-block and has a width and height set as well.

The inline-block elements will appear side by side because they behave like inline elements, but they also have a specified width and height, which gives them block-like properties.

But, if you remove the display: inline-block property, neither the height nor the width will be applied even though you defined it clearly.

Here is the revised CSS:

.inline-block-element {

width: 150px;

height: 100px;

}

.element1 {

background-color: lightblue;

}

.element2 {

background-color: lightgreen;

}

In this code, we removed the display: inline-block; property but kept everything else intact. Here, the span elements revert to their default behavior as inline elements.

As a result, the specified width and height are ignored, and the elements only take up the space needed for their content.

Understanding how inline-block works is useful because you can use it for creating layouts that require both alignment and dimension control within a continuous text flow.