**20. Box model, inline and block elements**

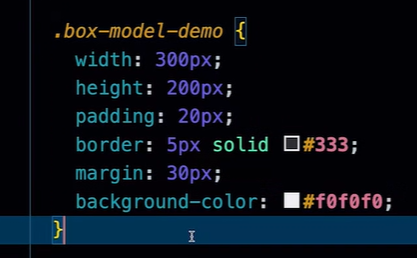
1. Box-model:

* The amount of space an element consumes on the webpage/website is called as box model. That space any particular element is going top consume is governed by Box model.
* There are 2 types of box model:  
  1. Content-box.

2. Border-box.

Tip from Hitesh Sir:

* “Sometimes it’s very important to understand the code than write the code.”
* See the Code > Analyze the Code > Understand it



* The above styling can be applied to any kind of box shaped element for eg a div, footer, header etc.

A close-up of a box

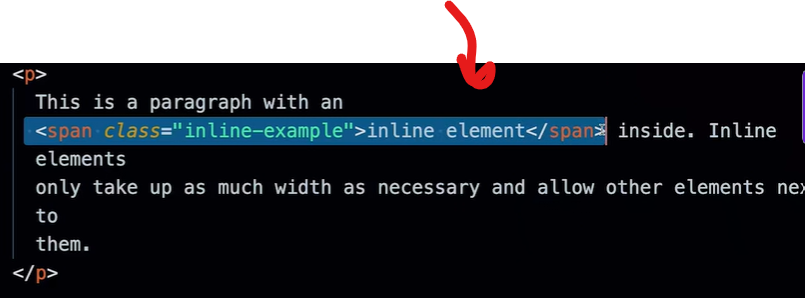
AI-generated content may be incorrect.

* Both the above boxes, have the same “.box-model-demo” styling. But still we can say that the size of the above 2 boxes seem to differ. The reason is that their sizes are being overrided by 2 other classes. They are:  
  

A close-up of a text

AI-generated content may be incorrect.

* Inline elements: These elements don’t consume any special space. These elements consume the space inline, just alongside the element of which inline element is a part.
* Block elements: These elements consume the entire space from left to right.



**𝘈𝘯𝘴𝘸𝘦𝘳𝘴 𝘱𝘰𝘸𝘦𝘳𝘦𝘥 𝘣𝘺 𝘊𝘰𝘱𝘪𝘭𝘰𝘵**

Q. in HTML, what is a span element? Does by-default span element take inline space?

A. The <span> element in HTML is like a transparent container—it doesn't *do* much visually, but it's incredibly useful for grouping bits of text or content so you can apply styles or JavaScript to them.

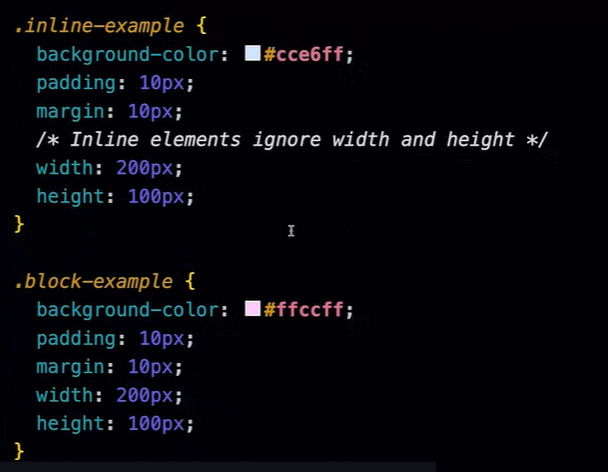
By default, yes, a <span> is an **inline** element. That means:

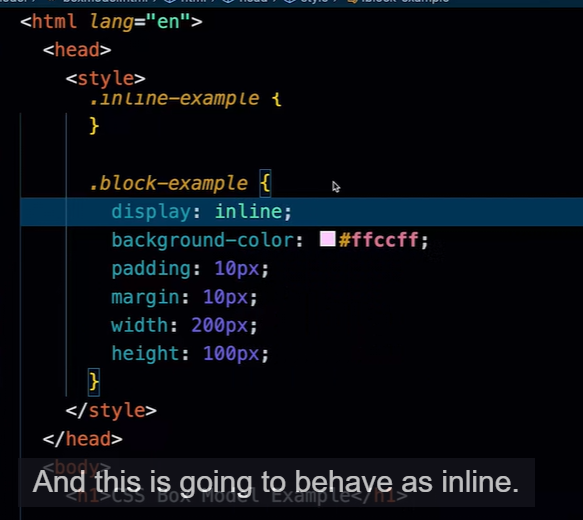
* It flows *within* a line of text rather than starting on a new line.
* It only takes up as much horizontal space as its content needs.
* You can’t set width and height on it—unless you change its display to something like inline-block or block.

Here’s a mini example:

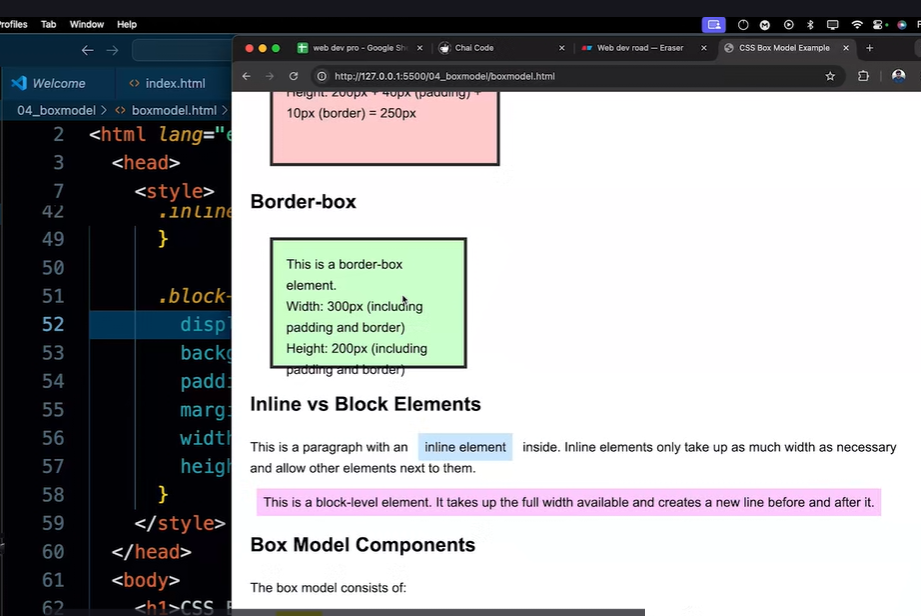
<p>This is a <span style="color: red;">highlighted</span> word in a sentence.</p>

The word "highlighted" gets special treatment, but it still flows naturally in the sentence. Want to experiment with making it behave like a block element? I can show that too.





* Explicitly turning a block level element into an inline element by using display property. Just entering “display: inline;”

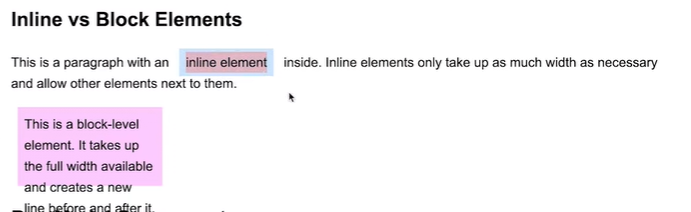
O/P:  


**Conclusion:**

* In content-box, when we specify a particular width x for an element, the total amount of space it will be consuming finally is x(which is the element width) + padding width + border width.
* In border-box, when we specify a particular width y for element, the total amount of space it will be consuming finally is y = element width + padding width + border width.

A close-up of a box

AI-generated content may be incorrect.



* Inline element takes as much as space, as it is required by the content in that element and just works with that. It doesn’t break out of the line.
* Block level element consumes the entire space, from left to right. It stays in the space we mentioned it to that element, but occupies the entire space from left to right and doesn’t allow a\