Block and Inline elements

Remember every element is inside its own invisible box. This box will either be a block or an inline element.

Block-level elements by default take up 100% of the space they have, stretching from right to left as far as they can. This means that they will not allow other elements to share the line they are on. So they appear to start on a new line in the browser window. These include <div>, <header>, <main>, <footer>, <nav>, <p>, <ul>, <li>, <section>, <article>, <form>, and <h1> - <h6>.

Inline elements by default will allow other elements to share the same line as long as there's room, then they will only wrap to the next line once it's full. They will only take up as much width as necessary. These include <a>, <img>, <span>, <button>,and <label>.

The <h1> element and all the paragraphs here are block level elements. If I place a border around them, you can see that they take up all the space they can.

However, if we placed some images side by side as long as there is room, they will share the horizontal space available.

You can change the defaults of inline and block elements with the display property. If I wanted to make a block element into an inline element, I would target it in CSS and use the display: inline declaration. And if I wanted to make an inline element into a block-level element I would use the display: block declaration.

There is also one more display property value, inline-block. It's kinds of a mix between the two. It would cause a block-level element to flow like an inline element, but it will retain other features of a block-level elements like allowing you to set a width and height on the element and top and bottom margins would work more predictably. The declaration you would use is display: inline-block.

Here we have 3 spans one inline, one inline-block and one block. Notice the difference between the 3. Even though the widths and heights are all set the same between the 3, the inline widths, heights and margin, padding top and bottom are not always what is expected. Inline-block is sometimes a better choice for a simple way to align a few items side-by-side.
https://www.w3schools.com/css/tryit.asp?filename=trycss_inline-block_span1

Because of these different display values, centering elements on our page works differently with different elements. Two common ways to center elements are text-align: center and margin: 0 auto.

Text-align: center will center the contents of the box. So, If I put text-align: center on the <h1> or the <p> then the content will get centered inside that invisible box or container.

Margin: 0 auto will center the box itself. The first value refers to the top and bottom margin. In our case we will just leave them zero. The second value refers to the right and left it will automatically give an equal gap on each side of the box.

With margin: 0 auto if the width of the element takes the whole space you won't see the centering. There needs to be a width set so there is space on each side to automatically give it an equal measurement on each side. Or, in other words, the box needs a width otherwise it will take up the full width of the page.

So if I now give the <h1> and <p> a width, the contents are still centered inside the box with the text-align: center but I can use margin:0 auto to center the box itself.

If I try text-align: center with an image, which is an inline element, it doesn't appear to center. That's because its border is tight around the image. The content is the image and it is already centered inside its tight border. But we can change the image to display: block, give it a width, then we can center it with margin: 0 auto. Or we could place the image inside a block-level element and then use text-align: center to center the contents of that element which would include that child element.