CSS COLOR MARKERS

<meta name="viewport" content="width=device-width, initial-scale=1.0">

* Makes your page same on all devices.

<link rel="stylesheet" href="styles.css">

* To link the style css to the html

margin:auto;

* To center your marker on the page, set its margin property to auto. This sets margin-top, margin-right, margin-bottom, and margin-left all to auto. When the shorthand margin property has two values, it sets margin-top and margin-bottom to the first value, and margin-left and margin-right to the second value.
* Then it becames margin: 10px auto;
* Multiple classes can be added to an element by listing them in the class attribute and separating them with a space.There are two main color models: the additive RGB (red, green, blue) model used in electronic devices, and the subtractive CMYK (cyan, magenta, yellow, black) model used in print
* CSS rule, use the shorthand padding property to add 10px of top and bottom padding, and set the left and right padding to 0. This works similarly to the shorthand margin property you used earlier.
* A color wheel is a circle where similar colors, or hues, are near each other, and different ones are further apart. For example, pure red is between the hues rose and orange.
* Two colors that are opposite from each other on the color wheel are called complementary colors. If two complementary colors are combined, they produce gray. But when they are placed side-by-side, these colors produce strong visual contrast and appear brighter.
* A very common way to apply color to an element with CSS is with hexadecimal or hex values. While hex values sound complicated, they're really just another form of RGB values.
* Hex color values start with a # character and take six characters from 0-9 and A-F. The first pair of characters represent red, the second pair represent green, and the third pair represent blue. For example, #4B5320.
* You may already be familiar with decimal, or base 10 values, which go from 0 - 9. Hexadecimal, or base 16 values, go from 0 - 9, then A - F:
* 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
* With hex colors, 00 is 0% of that color, and FF is 100%. So #00FF00 translates to 0% red, 100% green, and 0% blue, and is the same as rgb(0, 255, 0).
* The HSL color model, or hue, saturation, and lightness, is another way to represent colors.
* The CSS hsl function accepts 3 values: a number from 0 to 360 for hue, a percentage from 0 to 100 for saturation, and a percentage from 0 to 100 for lightness.
* If you imagine a color wheel, the hue red is at 0 degrees, green is at 120 degrees, and blue is at 240 degrees.
* Saturation is the intensity of a color from 0%, or gray, to 100% for pure color.
* Lightness is how bright a color appears, from 0%, or complete black, to 100%, complete white, with 50% being neutral.
* background-color: hsl(240,100%,50%);
* you can also use a color transition, or gradient, on an element.
* A gradient is when one color transitions into another. The CSS linear-gradient function lets you control the direction of the transition along a line, and which colors are used.
* One thing to remember is that the linear-gradient function actually creates an image element, and is usually paired with the background property which can accept an image as a value.
* The linear-gradient function is very flexible -- here is the basic syntax you'll use in this tutorial:
* linear-gradient(gradientDirection, color1, color2, ...);
* gradientDirection is the direction of the line used for the transition. color1 and color2 are color arguments, which are the colors that will be used in the transition itself. These can be any type of color, including color keywords, hex, rgb, or hsl.
* background: linear-gradient(90deg); background: lineargradient(90deg, rgb(255, 0, 0),rgb(0,255,0),rgb(0,0,255);
* Color-stops allow you to fine-tune where colors are placed along the gradient line. They are a length unit like px or percentages that follow a color in the linear-gradient function.
* For example, in this red-black gradient, the transition from red to black takes place at the 90% point along the gradient line, so red takes up most of the available space:
* linear-gradient(90deg, red 90%, black);
* Even without the color-stops, you might have noticed that the colors for the green marker transition at the same points as the red marker. The first color is at the start (0%), the second is in the middle (50%), and the last is at the end (100%) of the gradient line

.

* The linear-gradient function automatically calculates these values for you, and places colors evenly along the gradient line by default
* If no gradientDirection argument is provided to the linear-gradient function, it arranges colors from top to bottom, or along a 180 degree line, by default.
* Opacity describes how opaque, or non-transparent, something is. For example, a solid wall is opaque, and no light can pass through. But a drinking glass is much more transparent, and you can see through the glass to the other side.
* With the CSS opacity property, you can control how opaque or transparent an element is. With the value 0, or 0%, the element will be completely transparent, and at 1.0, or 100%, the element will be completely opaque like it is by default
* Another way to set the opacity for an element is with the alpha channel. Similar to the opacity property, the alpha channel controls how transparent or opaque a color is.
* To add an alpha channel to an rgb color, use the rgba function instead.
* the default display property for div elements is block. So when two block elements are next to each other, they stack like actual blocks.
* .cap,.sleeve{
* display:inline-block;
* rule to target both the cap and sleeve classes together.
* All HTML elements have borders, though they're usually set to none by default. With CSS, you can control all aspects of an element's border, and set the border on all sides, or just one side at a time. For a border to be visible, you need to set its width and style.
* This can be replaced by:
* border-left-width: 10px;
* border-left-style: solid;
* border-left-color: black;
* border-left: 10px double rgba(0, 0, 0, 0.75);
* The box-shadow property lets you apply one or more shadows around an element. Here is basic syntax:
* box-shadow: offsetX offsetY color;
* box-shadow: offsetX offsetY blurRadius color;
* If a blurRadius value isn't included, it defaults to 0 and produces sharp edges. The higher the value of blurRadius, the greater the blurring effect is.
* box-shadow: offsetX offsetY blurRadius spreadRadius color; it is used to spread the shadow bigger.