## Applied Visual Design: Center an Element Horizontally Using the margin Property

Another positioning technique is to center a block element horizontally. One way to do this is to set its margin to a value of auto.

## 

## Applied Accessibility: Add an Accessible Date Picker

<input type="date" id="pickdate" name="date">

Adding type="date" creates a date picker itself.

## Applied Visual Design: Create Visual Balance Using the text-align Property

This section of the curriculum focuses on Applied Visual Design. The first group of challenges build on the given card layout to show a number of core principles.

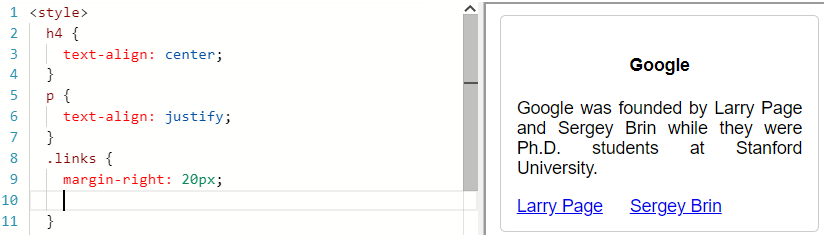
Text is often a large part of web content. CSS has several options for how to align it with the text-alignproperty.

text-align: justify;causes all lines of text except the last line to meet the left and right edges of the line box.

text-align: center;centers the text

text-align: right;right-aligns the text

And text-align: left;(the default) left-aligns the text.



## Applied Visual Design: Use the strong Tag to Make Text Bold

To make text bold, you can use the strong tag. This is often used to draw attention to text and symbolize that it is important. With the strong tag, the browser applies the CSS of font-weight: bold;to the element.

<p>Google was founded by Larry Page and Sergey Brin while they were Ph.D. students at <strong>Stanford University.</strong></p>

|  |  |
| --- | --- |
| HTML tag | Transformation |
| strong | Bold |
| u | Underline |
| em | Italicise |
| s | strikethrough |
| hr | Add horizontal line cross width of containing element |

|  |  |
| --- | --- |
| CSS Property | Transformation |
| text-transform | **lowercase, uppercase, capitalize,** inherit, initial, none |
| font-size | adjust the size of the text in an element |
| font-weight |  |
| transform | transform: skewX(20deg) |
|  | transform: skewY(20deg) |
|  | transform: scale(2.5) |

## Applied Visual Design: Adjust the background-color Property of Text

Instead of adjusting your overall background or the color of the text to make the foreground easily readable, you can add a background-color to the element holding the text you want to emphasize. This challenge uses rgba()instead of hex codes or normal rgb().

rgba stands for:  
  r = red  
  g = green  
  b = blue  
  a = alpha/level of opacity



## Applied Visual Design: Adjust the Size of a Header Versus a Paragraph Tag

The font size of header tags (h1through h6) should generally be larger than the font size of paragraph tags.

## Applied Visual Design: Add a box-shadow to a Card-like Element

/\* Keyword values \*/

box-shadow: none;

/\* offset-x | offset-y | color \*/

box-shadow: 60px -16px teal;

/\* offset-x | offset-y | blur-radius | color \*/

box-shadow: 10px 5px 5px black;

/\* offset-x | offset-y | blur-radius | spread-radius | color \*/

box-shadow: 2px 2px 2px 1px rgba(0, 0, 0, 0.2);

/\* inset | offset-x | offset-y | color \*/

box-shadow: inset 5em 1em gold;

/\* Any number of shadows, separated by commas \*/

box-shadow: 3px 3px red, -1em 0 0.4em olive;

/\* Global keywords \*/

box-shadow: inherit;

box-shadow: initial;

box-shadow: unset;

## Applied Visual Design: Decrease the Opacity of an Element

The opacityproperty in CSS is used to adjust the opacity, or conversely, the transparency for an item.

A value of 1 is opaque, which isn't transparent at all.  
A value of 0.5 is half see-through.  
A value of 0 is completely transparent.

opacity: 0.7;

## Applied Visual Design: Adjust the Hover State of an Anchor Tag

For example, the styling of an anchor tag can be changed for its hover state using the :hoverpseudo-class selector. Here's the CSS to change the colorof the anchor tag to red during its hover state:

a:hover{

color: red;

}

## Applied Visual Design: Change an Element's Position

Block-level items automatically start on a new line (think headings, paragraphs, and divs) while inline items sit within surrounding content (like images or spans). The default layout of elements in this way is called the normal flow of a document, but CSS offers the position property to override it.

Position properties (relative, fixed, static, sticky, absolute) are paired with the CSS offset properties of left or right, and top or bottom. These say how many pixels, percentages, or ems to move the item *away* from where it is normally positioned.

|  |  |
| --- | --- |
| position: relative; | It allows you to specify how CSS should move it relative to its current position in the normal flow of the page. |
| example moves the paragraph 10 pixels away from the bottom: p {  position: relative;  bottom: 10px;  } |
| position: absolute; | It locks the element in place relative to its closest parent container. If its parent item does not have position property set to relative then the browser keep looking up the chain and ultimately default to body tag. |
| Unlike the relative position, this removes the element from the normal flow of the document, so surrounding items ignore it. |
| Absolute positioning is based on co-ordiantes of the display while Relative position is relative to where the element is placed: |
| #searchbar {  position: absolute;  top: 50px;  right: 50px;  }  /\* Container element \*/  section {  position: relative;  } |
| position: fixed; | It a type of absolute positioning that locks an element relative to the **browser window.** |
| One key difference between the fixed and absolute positions is that an **element with a fixed position won't move** when the user scrolls. |
| #navbar {  position: fixed;  top: 0px;  left: 0px;   } |

## Relative position vs Absolute position

**Absolute Position**

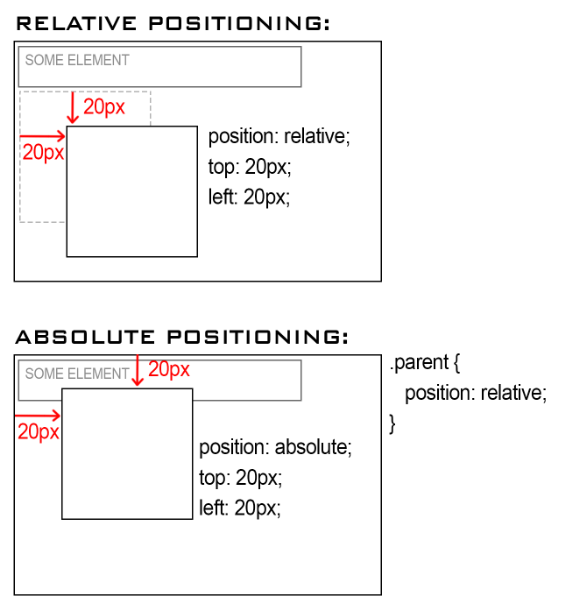
An absolute position element is positioned relative to the first parent element that has a position = relative.

**Relative Position**

A relative positioned element is positioned relative to its normal position.

To position an element relatively, the property position is set as relative. The difference between absolute and relative positioning is how the position is being calculated.

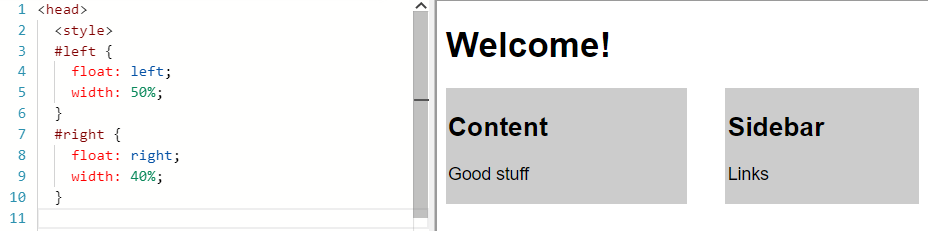
 Basically **relative** position is relative to previous element or window, while **absolute** don't care about the other elements unless it has a parent if you using top and left...



## Applied Visual Design: Push Elements Left or Right with the float Property

Floating elements are removed from the normal flow of a document and pushed to either the leftor rightof their containing parent element.

It's commonly used with the widthproperty to specify how much horizontal space the floated element requires.



## Applied Visual Design: Add Background Image

body {

background: url(https://i.imgur.com/MJAkxbh.png);

}

## Applied Visual Design: Adjust the Tone of a Color

Hue, Saturation, Lightness

header {

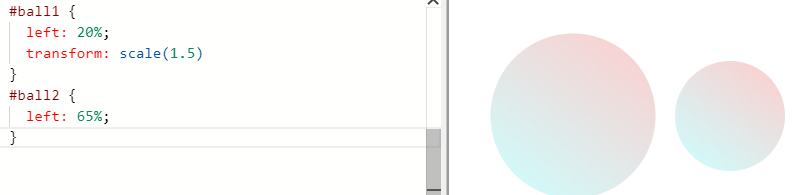
background-color: hsl(180, 90%, 35%);

color: #FFFFFF;

}

## Applied Visual Design: Use the CSS Transform scale Property to Change the Size of an Element

To change the scale of an element, CSS has the transform property, along with its scale()function.



## Applied Visual Design: Use the CSS Transform scale Property to Scale an Element on Hover

div:hover{

transform: scale(1.5)

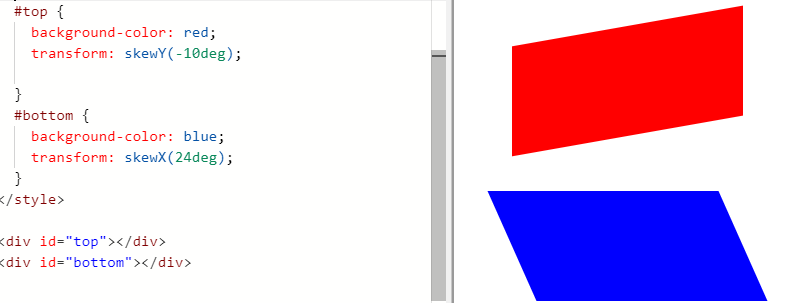
}

## Applied Visual Design: Use the CSS Transform Property skewX to SkewY an Element Along the X-Axis

The next function of the transform property is skewX(), which skews the selected element along its X (horizontal) axis by a given degree.

The following code skews the paragraph element by 24 degrees along the X-axis.

skewY()property skews an element along the Y (vertical) axis.



## Applied Visual Design: Animate Elements at Variable Rates

To illustrate, the animation on the right consists of two "stars" that each decrease in size and opacity at the 20% mark in the @keyframes rule, which creates the twinkle animation.

.star-1 {

margin-top: 15%;

margin-left: 60%;

animation-name: twinkle-1;

animation-duration: 1s;

}

.star-2 {

margin-top: 25%;

margin-left: 25%;

animation-name: twinkle-2;

animation-duration: 1s;

}

@keyframes twinkle-1 {

20% {

transform: scale(0.5);

opacity: 0.5;

}

}

@keyframes twinkle-2 {

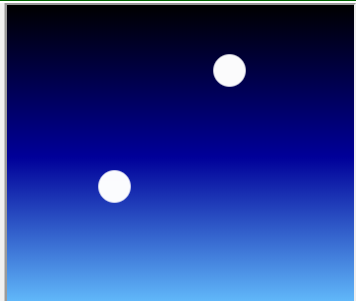
20% {

transform: scale(0.5);

opacity: 0.5;

}

}



## Applied Visual Design: Animate Multiple Elements at Variable Rates

In the animation running in the code editor, there are three "stars" in the sky that twinkle at the different rates on a continuous loop. To make them twinkle at different rates, we use the animation-duration property to different values for each element. Setting animation-duration of the elements with the classes star-1, star-2, and star-3 to 1s, 0.9s, and 1.1s, respectively.

.stars {

background-color: white;

height: 30px;

width: 30px;

border-radius: 50%;

animation-iteration-count: infinite;

}

.star-1 {

margin-top: 15%;

margin-left: 60%;

animation-duration: 1s;

animation-name: twinkle;

}

.star-2 {

margin-top: 25%;

margin-left: 25%;

animation-duration: 0.9s;

animation-name: twinkle;

}

.star-3 {

margin-top: 10%;

margin-left: 50%;

animation-duration: 1.1s;

animation-name: twinkle;

}

@keyframes twinkle {

20% {

transform: scale(0.5);

opacity: 0.5;

}

}

## Applied Visual Design: Change Animation Timing with Keywords

In CSS animations, the animation-timing-function property controls how quickly an animated element changes over the duration of the animation.

If the animation is a car moving from point A to point B in a given time (your animation-duration), the animation-timing-function says how the car accelerates and decelerates over the course of the drive.

animation-timing-function: ease-out | linear;

ease-out: slow

linear: faster

<style>

.balls {

border-radius: 50%;

background: linear-gradient(

35deg,

#ccffff,

#ffcccc

);

position: fixed;

width: 50px;

height: 50px;

margin-top: 50px;

animation-name: bounce;

animation-duration: 1s;

animation-iteration-count: infinite;

}

#ball1 {

left:27%;

animation-timing-function:ease-out;

}

#ball2 {

left:56%;

animation-timing-function: linear;

}

@keyframes bounce {

0% {

top: 0px;

}

100% {

top: 249px;

}

}

</style>