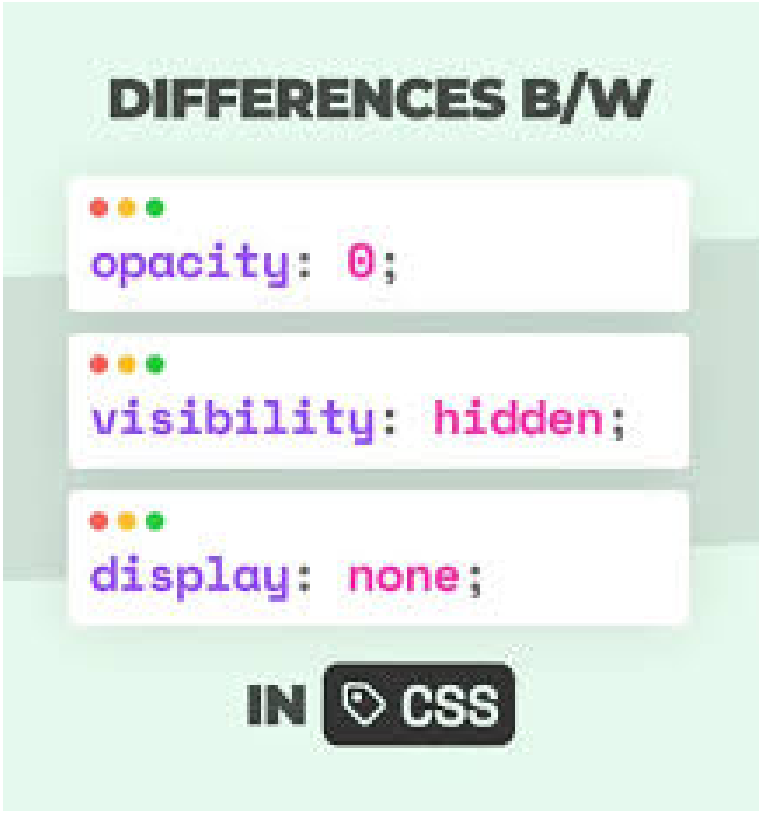
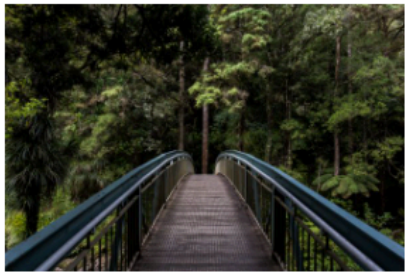
Opacity vs visibility



The opacity property can take a value from (0.0 - 1.0). The lower the value, the more transparent:

opacity 0.2

opacity 0.5



opacity 1
(default)

How to Set Opacity in CSS

To set the opacity of a background, image, text, or other element, you can use the CSS opacity property. Values for this property range from 0 to 1. If you set the property to 0, the styled element will be completely transparent (ie. invisible). If you set the property to 1, the element will be completely opaque.

Text Opacity CSS

Setting the opacity of text in CSS is nearly identical to setting the opacity of the background of an element. You can set the opacity of an entire element — the background, text within the element, the border, and everything else — using the opacity property.

To set the opacity of text and only text, then you need to use the CSS color property and RGBA color values.

Border Opacity CSS

Setting the opacity of a border in CSS is like setting the opacity of text. If you'd like to specify the opacity of the border of an element and nothing else, then you need to use the CSS shorthand border property and RGBA color values.

Image Opacity in CSS

You can set the opacity of an image in CSS as well. The opacity property is frequently used with the :hover selector to style an image. That way, the opacity of the image will change only when a user hovers over it. You have two options.

You can have the image become transparent when the user hovers over it and then become opaque when their mouse moves away. That's called a transparent hover effect.

CSS Opacity Gradient

In CSS, you can create a color gradient that shows one color gradually changing into another in a certain direction like top to bottom, left to right, or diagonally. Rather than changing from one color to a different color (think: red to blue), a gradient could show one color gradually changing from fully opaque to fully transparent.

The **opacity** in CSS is the property of an element that describes the transparency of the element. It is the opposite of transparency & represents the degree to which the content will be hidden behind an element.

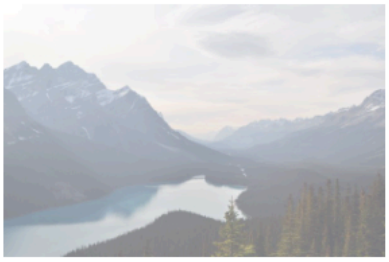
Image Hover Opacity

The hover opacity property is applied to the image when the mouse puts it over the image otherwise opacity

property changes. The value of opacity can easily reverse the process by setting the opacity as a higher value at first and then lowering it when hovering over it

Transparent Hover Effect

The opacity property is often used together with the :hover selector to change the opacity on mouse-over:


visibility in css

Use the invisible utility to hide an element, but still maintain its place in the DOM, affecting the layout of other elements

CSS Visibility Parameters

What is CSS Visibility?

Visibility is a property in CSS that designates whether or not a specific element is — you guessed it — visible. It's also important to remember that hidden properties still take up space on your website pages — and there's a different property to use if you want to hide *and* remove it from the document.

If you use the CSS visibility property and set your element to hidden, it is no longer visible, but there is still room allocated for it. Therefore, you're hiding the element from your view, but it's still accessible in the page flow. That means the layout of the document won't be altered.

visible

Just like it sounds, visible makes things visible. Nothing is hidden by default, so this value does nothing unless you have set hidden on this or a parent of this element.

hidden

The hidden value hides things. This is different than using display: none, because hidden only *visually* hides elements. The element is still there, and still takes up space on the page, but you can't see it anymore (kind of like turning the opacity to 0). Interestingly, this property does not inherit by default. That means that, unlike the display or opacity properties, you can make an element hidden, and still have one of its children as visible

inherit

The default value. This simply causes the element to inherit the value of its parent.

CSS Visibility Property Values

As mentioned, you can use this property to adjust an element's visibility. Here are the different property values you should know.

1. Hidden: You have successfully hidden the element. (Remember: There's still space on the page for it!)
2. Visible: This is the default setting. The element is visible.
3. Initial: This property is set to its default value.
4. Collapse: You only use collapse for table rows, columns, row groups, and column groups. If you try to use this on a different element, it will display as hidden.
5. Inherit: It inherits the property from a parent element.

Accessibility

Using a `visibility` value of `hidden` on an element will remove it from the accessibility tree. This will cause the element and all its descendant elements to no longer be announced by screen reading technology.

Interpolation

When animated, visibility values are interpolated between *visible* and *not-visible*. One of the start or ending values must therefore be `visible` or no interpolation can happen. The value is interpolated as a discrete step, where values of the easing function between `0` and `1` map to `visible` and other values of the easing function (which occur only at the start/end of the transition or as a result of `cubic-bezier()` functions with y values outside of [0, 1]) map to the closer endpoint.

The difference between Opacity and visibility and display: none;

Differences

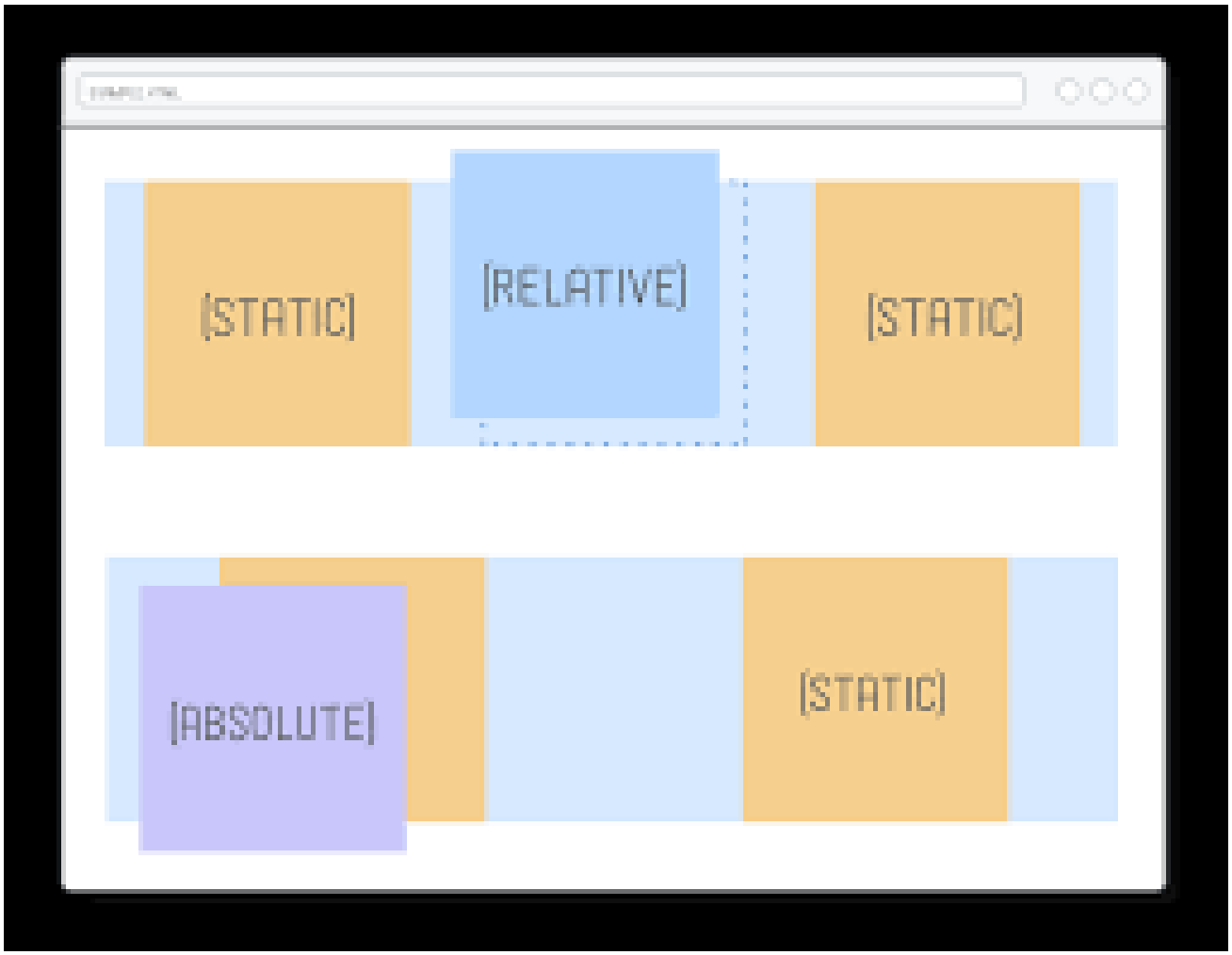
1. `display: none` doesn't take space when the element is rendered. The other ways still take the space normally.
2. The browser will not response to any events of element which uses either `display: none` or `visibility: hidden`. The `visibility: hidden` style behaves like a combination of `opacity: 0` and `pointer-events: none`.
3. Regarding the accessibility, `opacity: 0` is the only property which makes the element accessible in the tab order, and the element's content can be read by screen readers.
4. Applying `display: none` or `opacity: 0` will effect on child elements. `visibility: hidden`, on the other hand, doesn't change the visibility of any children.
5. It's worth noting that if you want to measure the size of element, then you can't use `display: none` at all.

As mentioned in the first difference, an element with `display: none` doesn't take any space on the page. Hence, all properties related to the element size, such as `clientHeight`, `clientWidth`, `height`, `offsetHeight`, `offsetWidth`, `scrollHeight`, `scrollWidth` and `width` are zero.

All properties returned by the `getBoundingClientRect()` method are zero as well.

Similarly, an element with `visibility: hidden` will have empty inner text (equivalent with the `innerText` property).

how to center elements using position



elements have a static position unless specified otherwise as absolute, fixed, relative or sticky.

To center text or links horizontally, just use the text-align property with the value center.

Use the shorthand margin property with the value 0 auto to center block-level elements like a div horizontally

Centering a Fixed-Position Element:

Example: We have created a `<p>` tag with a light green background. By applying `position: fixed;`, the element remains fixed relative to the viewport. Utilizing `top: 50%;` and `left: 50%;` positions the element at the center, while `transform: translate(-50%, -50%);` refines the centering by adjusting the position based on the element's dimensions.

Using position

One method for aligning elements is to use position: absolute;:

Positioning gives us a completely different layout method. The default value is static, and makes the element behave normally.

Using position: relative lets you specify an offset with top, bottom, left, and right.

This can be useful to do a simple offset without using something like margin, but it becomes really powerful when containing a child with position:absolute.

Centering using absolute position

Something that can be very useful but stumps a lot of beginners is centering elements with absolute positioning. If your element has a set size, it's just a matter of offsetting using margins. Let's take the last example but center the icon both horizontally and vertically. To center an element using absolute positioning, just follow these steps:

1. Add left: 50% to the element that you want to center. You will notice that this aligns the left edge of the child element with the 50% line of the parent.
2. Add a negative left margin that is equal to half the width of the element. This moves us back onto the halfway mark.
3. Next, we'll do a similar process for the vertical axis. Add top: 50% to the child
4. And then add a negative top margin equal to half its height.