

Learn CSS By Use Cases

ebook by Joe Harrison

**Outro
Thank You
For Reading**



Joe Harrison
@frontendjoe

Outro

You've made it this far already, great work.

Now go break stuff!

I focus my game around knowing this core set of CSS properties and features inside out – I can only advise you to do the same. Once you scope the language down to the things you actually need, it becomes a lot more manageable to learn.

Practice, patience and persistence.

Always,

Joe



Joe Harrison
@frontendjoe

Learn CSS By Use Cases

ebook by Joe Harrison



Outro Extras



Joe Harrison
@frontendjoe

Outro

CSS not in the ebook but in source code

inherit

The inherit keyword can be added as a value for any property. It means that this value will be inherited from any parent (ancestor) that has the same property set. I find it very useful when styling child elements that share styles with the parent – if the value ever changes we only have to change it in one place, handy.

auto

The auto keyword can only be used in certain properties, most notably: grid, flex, margin, width and height. Each context will be different which makes auto styling tricky to learn (just practice).

Outro

CSS not in the ebook but in source code

none

The `none` keyword can be added to many properties including border, background, outline and `user-select`. In each context the behaviour will be similar, usually disabling the property.

none vs 0

When used with composite styles like border and background the values will correspond to different properties. If we add “`border: 0`” it will set “`border-width: 0`” while “`border: none`” will set “`border-style: none`”. Both will achieve the same effect but it’s important to understand the difference.

Outro

CSS not in the ebook but in source code

Properties

visibility - Used to define whether an element is visible or not. When transitioned along with opacity we can create smooth effects, you'll see plenty examples of this property combo in my source code.

backface-visibility - Used to define whether an element is still visible from behind, when it is 3D rotated using the transform property. In my rotating badges use case (in extras – Instagram Content section), I set backface-visibility to hidden as I'm manually adding the “backface” myself. In my split button 2 use case I don't set the property here as I want the “backface” visible.

Outro

CSS not in the ebook but in source code

Properties

perspective - Used to define the depth of a 3D transform element. It should be added to an ancestor of the transformed element and my go to value is always 1000px (this seems to work nice in most 3D use cases). As well as setting the perspective property inside an ancestor, we can alternatively use the perspective() property value function inside transform. This works well when it's not possible to add the property to the parent, just remember that the perspective function must appear first inside a transform property value. In my square spinner use case (in extras – Animation section) I demonstrate the property value function technique.

Outro

CSS not in the ebook but in source code

Properties

appearance - Used to disable the native appearance of HTML form elements, most notably on iOS mobile devices. In my use cases that feature the HTML input range element, I use the appearance property to hide the ugly default slider styles and then add my own. If you find all the occurrences of “appearance: none” in my source code (there are not many), then try toggling the property on and off – you should be able to see what's going on. Hiding the appearance on HTML select elements is also useful when adding custom style dropdowns – you will find my example of using this property in the Single Use section.

Outro

CSS not in the ebook but in source code

@ Rules

These include @media (to create responsive media queries) and @keyframes (to create animations). I don't really use many others.

Transitions

Its difficult to show transition use cases in a static ebook, but you'll find loads of these guys in my source code.

Variables Custom Properties

We touch on variables in my text-shadow use case (to create long shadows by reusing the same color). Without variables we would have to edit many lines of code if we decided to change the color.

Outro

CSS not in the ebook but in source code

Vendor Prefixes

```
input::-moz-placeholder { color: gray; }
```

Above, the example ensures the placeholder pseudo element will work in Mozilla Firefox.

The original use case of vendor prefixes was to increase browser compatibility. Nowadays, modern frameworks will add browser compatibility prefixes automatically - usually via a library called autoprefixer.

<https://www.npmjs.com/package/autoprefixer>

For this reason I just don't worry about them.

Outro

CSS not in the ebook but in source code

Vendor Prefixes Continued

I mostly use vendor prefixes to take advantage of cutting edge CSS styling features, such as input range pseudo element selectors.

```
.slider::-webkit-slider-thumb {  
    -webkit-appearance: none;  
    /* custom thumb styles*/  
}
```

In the above example I'll use vendor prefixes to target the thumb handle part of the HTML input range slider

Styling these types of elements is not possible without vendor prefixes. However, as they're largely experimental – use them with caution.

Outro

CSS not in the ebook but in source code

Column Properties

You will find many use cases for these properties in the source code – in the Single Use section. They are essentially used to apply advanced formatting to any block of text, whether it be a list or a paragraph.

The main properties are as follows:

columns - Displays text as columns.

column-span - Allows headers to span across the top/bottom of text columns.

column-rule - Add borders between columns.

column-gap - Add gaps between columns.

Outro

CSS things I don't use

Functions

calc - I find this is unnecessary and confusing.

clamp - Still very new to me, I need to try using it in production before adding use cases to this ebook – watch this space.

Properties

accent-color - I always build custom UI components so this is not necessary for me.

backdrop-filter - Useful when creating glassmorphism but I don't personally use it.

background-clip - Defines how a background extends within an element, but I don't use it.

Outro

CSS things I don't use

Properties

border-image - Can be used to create gradient borders but I prefer other techniques.

border-spacing - Table property that adds space between table cells – never desired by myself.

caret-color - I always build custom UI components so this is not necessary for me.

clip-path - Allows you to create abstract shapes but I prefer to achieve it with other techniques like using actual SVG's.

direction - Reverse the direction of text, usually to left-to-right for certain regions.

empty-cells - Hide borders on empty cells in a table, why would one ever want this.

Outro

CSS things I don't use

Properties

float - Previously used to build layouts, it's actually to float images inside paragraphs.

grid - Shorthand to set the majority of grid properties, I personally don't like it.

mask-image - I honestly have no idea what any of these mask properties do.

outline - Puts a border around the currently focused element - I just disable it and add my own styles with :focus and box-shadow.

quotes - I always want normal quotes.

tab-size - I always want normal tab size.

Outro

CSS things I don't use

Flow Layout

In CSS, Flow Layout (or Normal Flow) is the way in which block and inline HTML elements are laid out on a web page, before any styling changes have affected their layout. Block elements take up a whole line (like a paragraph) and inline elements appear side by side (like span and input).

Why I don't use it

In short – I just do all of my block/inline styling in either flexbox or grid these days. Only when working with paragraphs of text (like a blog website) would it ever be helpful to me.



Outro

CSS things I don't use

!important

I believe this is used to “hack” specificity and override property values when working with third party libraries. !important has been known to cause a headache when debugging CSS in Chrome dev tools.

Why I don't use it

In reality – I just never really need it. The easiest way to “hack” specificity (if you really need to) is simple, just be more specific. People think they need it in CSS, I’d always argue that they dont.



Outro

CSS things I don't use

Viewport Units

vh and vw

vh stands for viewport height and vw stands for viewport width. These lengths can be used when making elements responsive, as they will inevitably scale when the browser is resized.

Why I don't use them.

I put one example in the source code but I don't have any in production. In the context of position: fixed elements the offset properties are relative to the browser, so I just use % instead.



Outro

CSS things I don't use

Viewport Units

vmin and vmax

These seem really complex when learning but the concept is actually rather simple. vmin is a percentage of the smallest length out of vh and vw, whereas vmax is the highest length – easy.

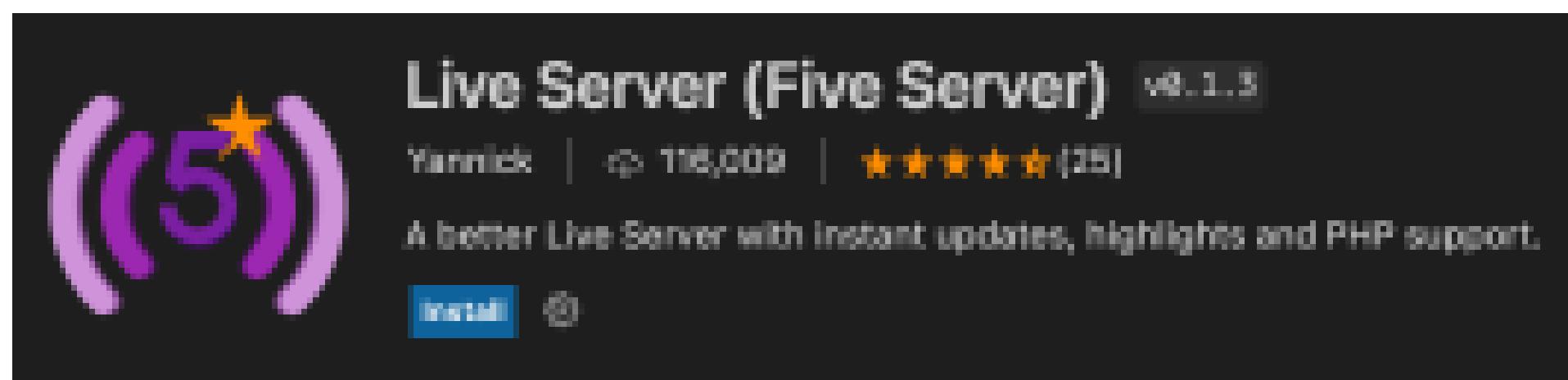
Why I don't use them

Again these are targeted at being useful in responsive design and are often applied to fonts in hero banners. I think I'm put off by varied browser support with often unexpected results.

Outro

Source Code

To run my source code I recommend using a VS Code extension called live-server. The extension will enable you to make live edits to the code, then see the result instantly in your browser (no need to refresh the page).



Once you've installed the extension, simply run the website by clicking the “Go Live” button in the bottom right corner of VS Code.

