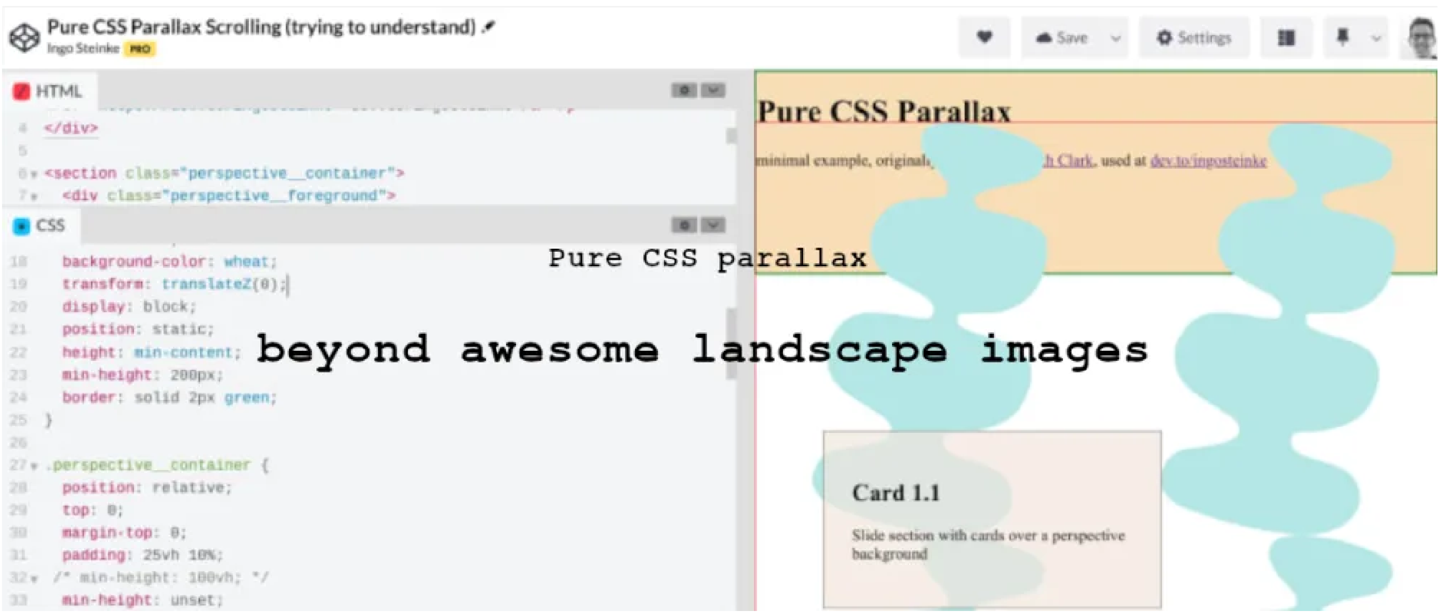

Pure CSS Parallax Scrolling (trying to understand)
Ingo Steinke PRO
Save
Settings
HTML
CSS
Pure CSS Parallax
minimal example, original, h Clark, used at dev.to/ingosteinke
Pure CSS parallax
beyond awesome landscape images
Card 1.1
Slide section with cards over a perspective background

```
</div>

<section class="perspective__container">
  <div class="perspective__foreground">
```

```
  background-color: wheat;
  transform: translateZ(0);
  display: block;
  position: static;
  height: min-content;
  min-height: 200px;
  border: solid 2px green;
}

.perspective__container {
  position: relative;
  top: 0;
  margin-top: 0;
  padding: 25vh 10%;
/* min-height: 100vh; */
  min-height: unset;
```

Edit Manage Stats

Ingo Steinke
Posted on Jul 3 • Updated on Aug 9

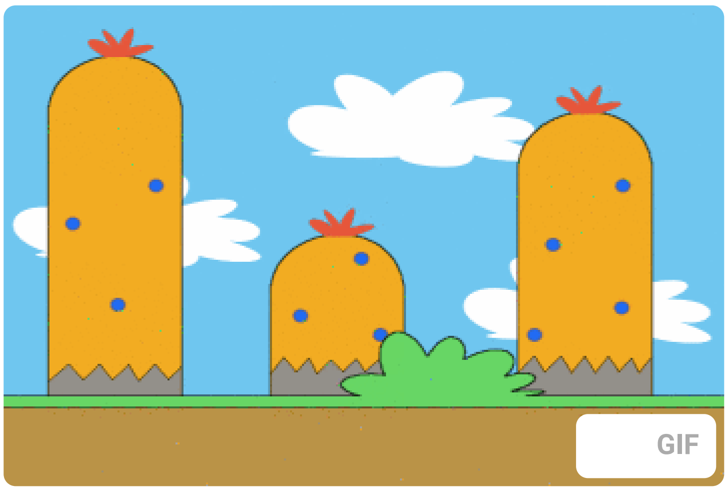
❤️ 57 🦄 5 🤯 4 🙌 3 🔥 4

Pure CSS parallax perspective beyond landscape images

#webdev #css #tutorial #design

I researched many tutorials and examples about "parallax scrolling" effects, and I wasn't impressed. Most parallax examples were not elegant both in a visual way and concerning their code. I often struggled to understand the principle hidden between irrelevant styles and misleading class names. Some "pure CSS" examples still use JavaScript, and most don't care about accessibility, using parallax movements even if the users have set to prefer reduced motion.

This **(meta-) tutorial** shows some experiments and deliberate failures on my way to understand what is going on and how to code a **working, robust, accessible, and maintainable perspective effect beyond the typical "awesome landscape" image**.



Source: Wikimedia Commons: File:Parallax_scrolling_example_scene.gif.

Table of Contents

Definition(s) of parallax scrolling effects

What is a "**parallax perspective effect**" anyway? Simple as it may seem, there seem to be different interpretations of what it is on a website, generalized as an umbrella term for various kinds of scroll-linked movement effects. A popular parallax effect simply resizes full-width hero images to make them slightly larger than they need to be, then adding a subtle scroll-linked movement to make it appear as if we were looking at a distant landscape through a window while moving. This is quite nice, although it does not make much sense if the pictures don't show a distant landscape.

Parallax beyond awesome landscape images

Another reason for restricting the effect to hero images is that this circumvents a lot of the problems of more complex scenarios, as we will see when inspecting code demos below.

AWESOME
PARALLAX SCROLLING
GIF

Source: [CSS mix-blend-mode and Awesome parallax scrolling by Andrej Sharapov on dribbble]()

Another scroll-linked effect that changes the content, like overlaying a photograph over a matching drawing or wireframe model, like in the [popular Ivy Chen example](), has also been listed as a parallax effect, while it's actually something completely different.

According to [Wikipedia, parallax]() is "a displacement or difference in the apparent position of an object viewed along two different lines of sight", and [parallax scrolling]() applies this effect in analog cartoon animation films or "in computer graphics where background images move past the camera more slowly than foreground images, creating an illusion of depth in a 2D scene of distance" like the clouds in classic jump-and-run-games like Super Mario Brothers or The Great Giana Sisters.

Source: The Great Giana Sisters game sequence on makeagif.com.

Unlike the classic film or computer game examples, where something runs, rides, or flies in a horizontal direction in front of a landscape, websites are typically scrolled down in a vertical direction.

Misconceptions, outdated and complicated tutorials

I heard different things about perspective effects, from "just add a `perspective` property" to "the HTML document / body must act as a perspective parent / scrolling container, thus making it necessary to refactor our project and risk undesired side-effects on pages that don't even use the scrolling effect.

I found a relatively straightforward codepen by Keith Clark related to his 2014 Pure CSS Parallax Websites that I forked and modified in order to better understand and reduce to its core requirements to generate a minimal template for different kinds of parallax perspective effects that can be added to any project without breaking existing work and without needing to modify existing markup. I only succeeded kind of half way.

Some of the "pure CSS" parallax examples actually contain JavaScript, and some aspects might be outdated and obsolete nearly ten years later. But the essence is still the same! Quoting Keith's 2014 blog post:

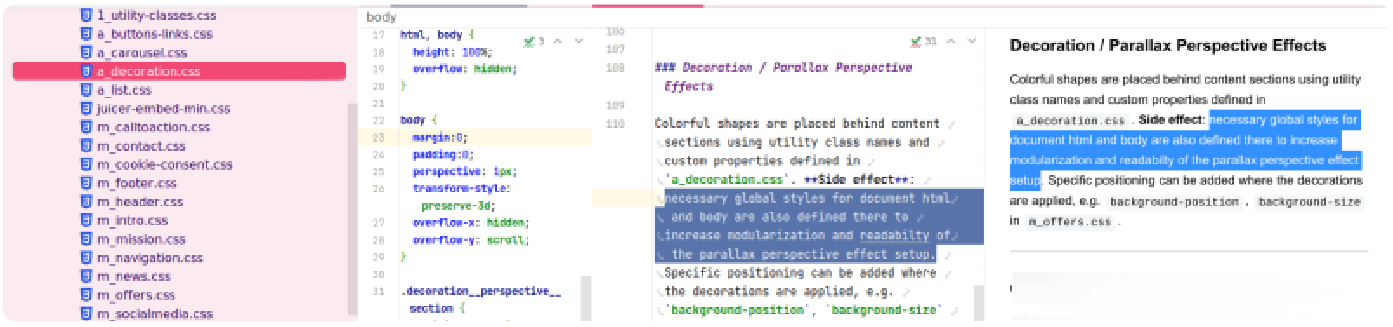
> The `parallax` class is where the parallax magic happens. Defining the height and perspective style properties of an element will lock the perspective to its centre, creating a fixed origin 3D viewport. Setting `overflow-y: auto` will allow the content inside the element to scroll in the usual way, but now descendant elements will be rendered relative to the fixed perspective. This is the key to creating the parallax effect.

Using this technique and replacing the background layer div with a `.parallax::after` style, we should be able to achieve our goals without JavaScript, global styles and side effects.

Pure CSS and minimal additional markup

Why would we even want to do that? While Tailwind and Bootstrap use functional classes everywhere, React projects and content management software introduce illegible generic class names based on random instance hashes, why not write HTML like it's 1999 and add a lot of new div elements to wrap our virtual layers? You might guess from my tone, and my previous articles, that's not how I prefer to code modern websites.

Trying to document how and why a decorative module introduced global styles and possible side effects, I could not help but feel I was implementing it in the wrong way.



```
1_utility-classes.css
a_buttons-links.css
a_carousel.css
a_decoration.css
a_list.css
juicer-embed-min.css
m_calltoaction.css
m_contact.css
m_cookie-consent.css
m_footer.css
m_header.css
m_intro.css
m_mission.css
m_navigation.css
m_news.css
m_offers.css
m_socialmedia.css
```

```
body
html, body {
  height: 100%;
  overflow: hidden;
}

body {
  margin:0;
  padding:0;
  perspective: 1px;
  transform-style:
    preserve-3d;
  overflow-x: hidden;
  overflow-y: scroll;
}

.decoration__perspective__
  section {
```

```
### Decoration / Parallax Perspective
  Effects

Colorful shapes are placed behind content
sections using utility class names and
custom properties defined in
`a_decoration.css`. **Side effect**:
necessary global styles for document html
 and body are also defined there to
increase modularization and readabilty of
 the parallax perspective effect setup.
Specific positioning can be added where
the decorations are applied, e.g.
`background-position`, `background-size`
```

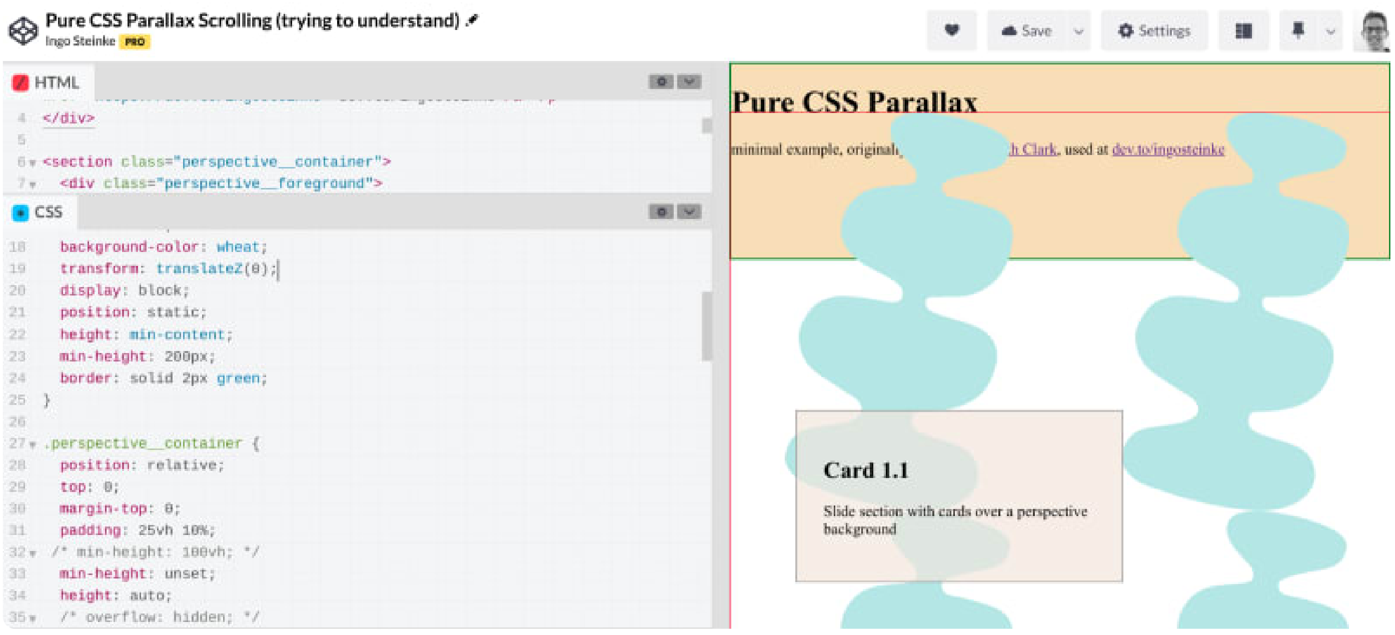
Decoration / Parallax Perspective Effects

Colorful shapes are placed behind content sections using utility class names and custom properties defined in `a_decoration.css`. **Side effect:** necessary global styles for document html and body are also defined there to increase modularization and readabilty of the parallax perspective effect setup. Specific positioning can be added where the decorations are applied, e.g. `background-position`, `background-size` in `m_offers.css`.

I prefer to introduce new design

- in a **modular** / atomic way to **prevent side-effects**,
- separating design and content, providing themes as optional and interchangeable skins around the bare, semantic content,
- in a minimal way to keep my code **readable** and **reduce loading time**.

That's why I don't think that I succeeded to create a minimal, portable template for a simple parallax scrolling effect yet. But still it might be a step forward - after several steps back and forth, as you can see in this intermediate debugging screenshot, trying to understand and debug adding colorful borders and experimenting with position, height, overflow, and z-index:


Pure CSS Parallax Scrolling (trying to understand)
Ingo Steinke PRO
Save
Settings
HTML
CSS
Pure CSS Parallax
minimal example, original
h Clark, used at dev.to/ingosteinke
Card 1.1
Slide section with cards over a perspective background

```
</div>

<section class="perspective__container">
  <div class="perspective__foreground">
```

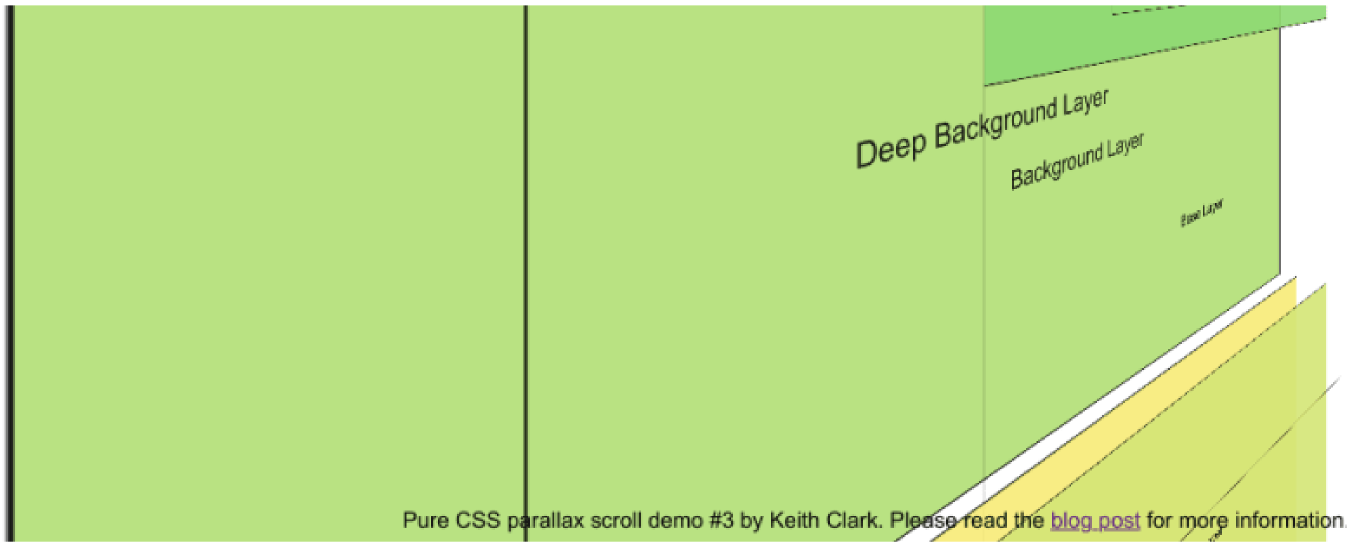
```
  background-color: wheat;
  transform: translateZ(0);
  display: block;
  position: static;
  height: min-content;
  min-height: 200px;
  border: solid 2px green;
}

.perspective__container {
  position: relative;
  top: 0;
  margin-top: 0;
  padding: 25vh 10%;
  /* min-height: 100vh; */
  min-height: unset;
  height: auto;
  /* overflow: hidden; */
```

The new code still has side-effects, as it seems that we need to make its the perspective background layer's height greater than the content, and that clipping with `overflow: hidden` did somehow break the perspective effect, so we need to give adjacent unrelated elements a higher `z-index` and define a background color to make sure that they stay on top and remain visually unaffected. Another aspect still true and already mentioned in the "parallax sections" chapter of Keith's good old tutorial:

> One important rule to keep in mind when grouping elements is, we cannot clip the content of a group. Setting `overflow: hidden` on a `parallax__group` will break the parallax effect. Unclipped content will result in descendant elements overflowing, so we need to be creative with the `z-index` values of the groups to ensure content is correctly revealed/hidden as the visitor scrolls through the document.

At least we could, in theory, limit our perspective effect to the parallax container so we don't need to apply global `html, body` styles. But won't that break continuous body scrolling and add scrollbars to the sections instead?

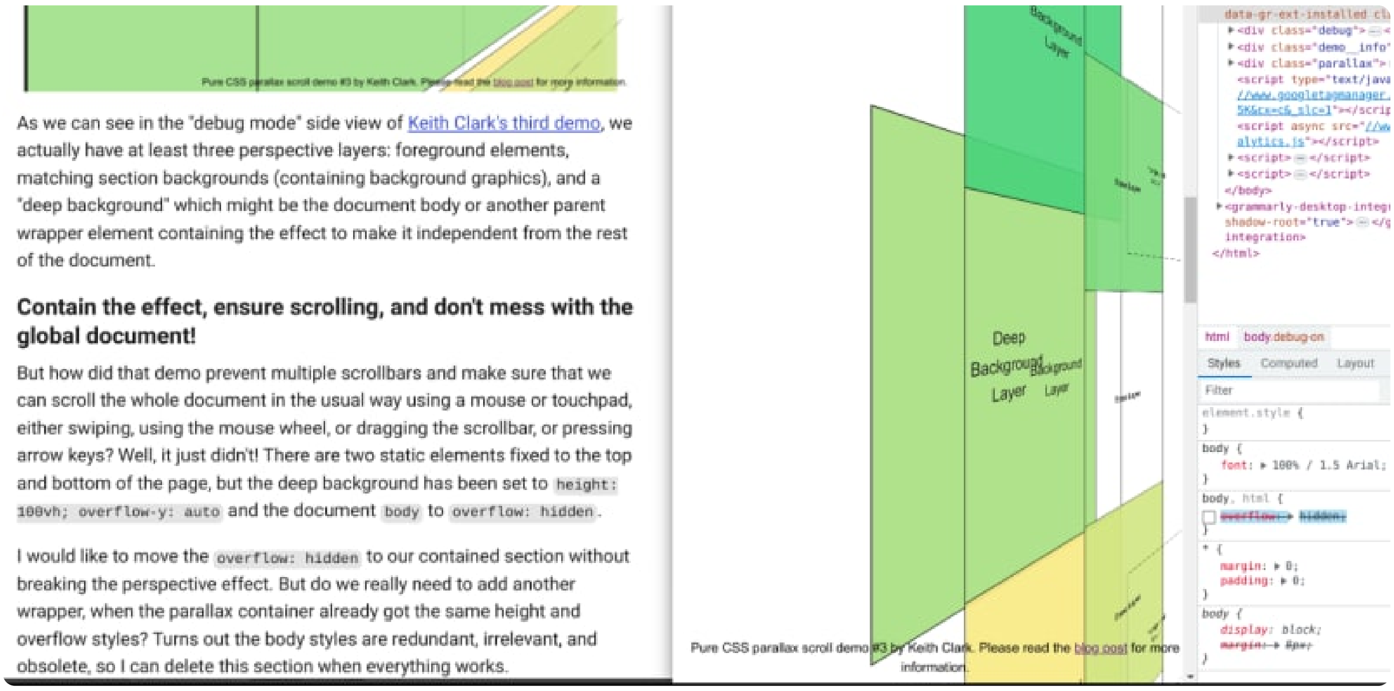

Deep Background Layer
Background Layer
Base Layer
Pure CSS parallax scroll demo #3 by Keith Clark. Please read the blog post for more information.

As we can see in the "debug mode" side view of Keith Clark's third demo, we actually have at least three perspective layers: foreground elements, matching section backgrounds (containing background graphics), and a "deep background" which might be the document body or another parent wrapper element containing the effect to make it independent from the rest of the document.

Contain the effect, ensure scrolling, and don't mess with the global document!

But how did that demo prevent multiple scrollbars and make sure that we can scroll the whole document in the usual way using a mouse or touchpad, either swiping, using the mouse wheel, or dragging the scrollbar, or pressing arrow keys? Well, it just didn't! There are two static elements fixed to the top and bottom of the page, but the deep background has been set to `height: 100vh; overflow-y: auto` and the document `body` to `overflow: hidden`.

I would like to move the `overflow: hidden` to our contained section without breaking the perspective effect. But do we really need to add another wrapper, when the parallax container already got the same height and overflow styles? Turns out the body styles are redundant, irrelevant, and obsolete, so I can delete this section when everything works.


Pure CSS parallax scroll demo #3 by Keith Clark. Please read the blog post for more information.

As we can see in the "debug mode" side view of Keith Clark's third demo, we actually have at least three perspective layers: foreground elements, matching section backgrounds (containing background graphics), and a "deep background" which might be the document body or another parent wrapper element containing the effect to make it independent from the rest of the document.

Contain the effect, ensure scrolling, and don't mess with the global document!

But how did that demo prevent multiple scrollbars and make sure that we can scroll the whole document in the usual way using a mouse or touchpad, either swiping, using the mouse wheel, or dragging the scrollbar, or pressing arrow keys? Well, it just didn't! There are two static elements fixed to the top and bottom of the page, but the deep background has been set to `height: 100vh; overflow-y: auto` and the document `body` to `overflow: hidden`.

I would like to move the `overflow: hidden` to our contained section without breaking the perspective effect. But do we really need to add another wrapper, when the parallax container already got the same height and overflow styles? Turns out the body styles are redundant, irrelevant, and obsolete, so I can delete this section when everything works.

Pure CSS parallax scroll demo #3 by Keith Clark. Please read the blog post for more information.

```
data-gr-ext-installed cl
  ▸<div class="debug">…<
  ▸<div class="demo__info"
  ▸<div class="parallax">
   <script type="text/java
   //www.googletagmanager.
   5K&cx=c&_slc=1"></scrip
   <script async src="//ww
   alytics.js"></script>
  ▸<script>…</script>
  ▸<script>…</script>
 </body>
▸<grammarly-desktop-integ
 shadow-root="true">…</g
 integration>
</html>
```

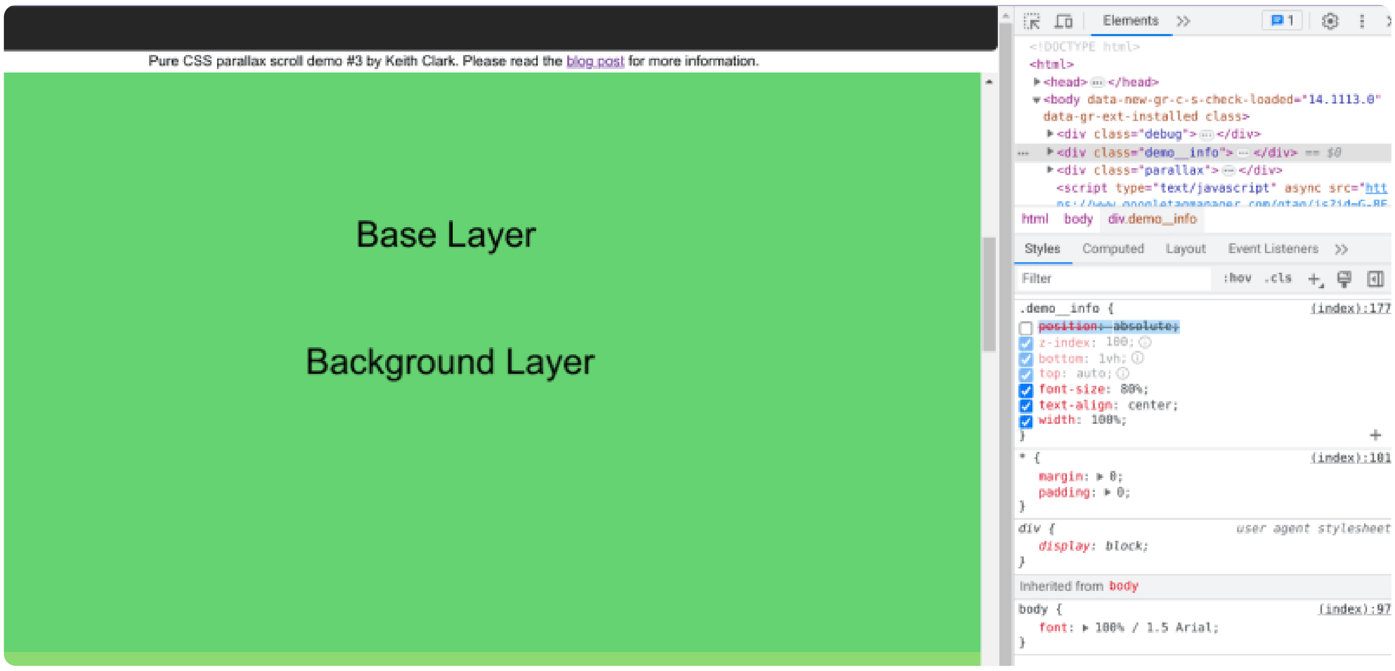
html body.debug-on

Styles Computed Layout

Filter

```
element.style {
}
body {
  font: ▸ 100% / 1.5 Arial;
}
body, html {
☐ ~~overflow: ▸ hidden;~~
}
* {
  margin: ▸ 0;
  padding: ▸ 0;
}
body {
  display: block;
  ~~margin: ▸ 8px;~~
}
```

The point is that demo doesn't use any other content sections before or after the deep background that are not pinned to an absolute position. If it did, it would have the same problem as my interim codepen: **double scrollbars** and **interrupted user experience**.


Pure CSS parallax scroll demo #3 by Keith Clark. Please read the blog post for more information.

```
<!DOCTYPE html>
<html>
 ▸<head>…</head>
 ▾<body data-new-gr-c-s-check-loaded="14.1113.0"
   data-gr-ext-installed class>
   ▸<div class="debug">…</div>
…  ▸<div class="demo__info">…</div> == $0
   ▸<div class="parallax">…</div>
    <script type="text/javascript" async src="htt
```

html body div.demo__info

Styles Computed Layout Event Listeners

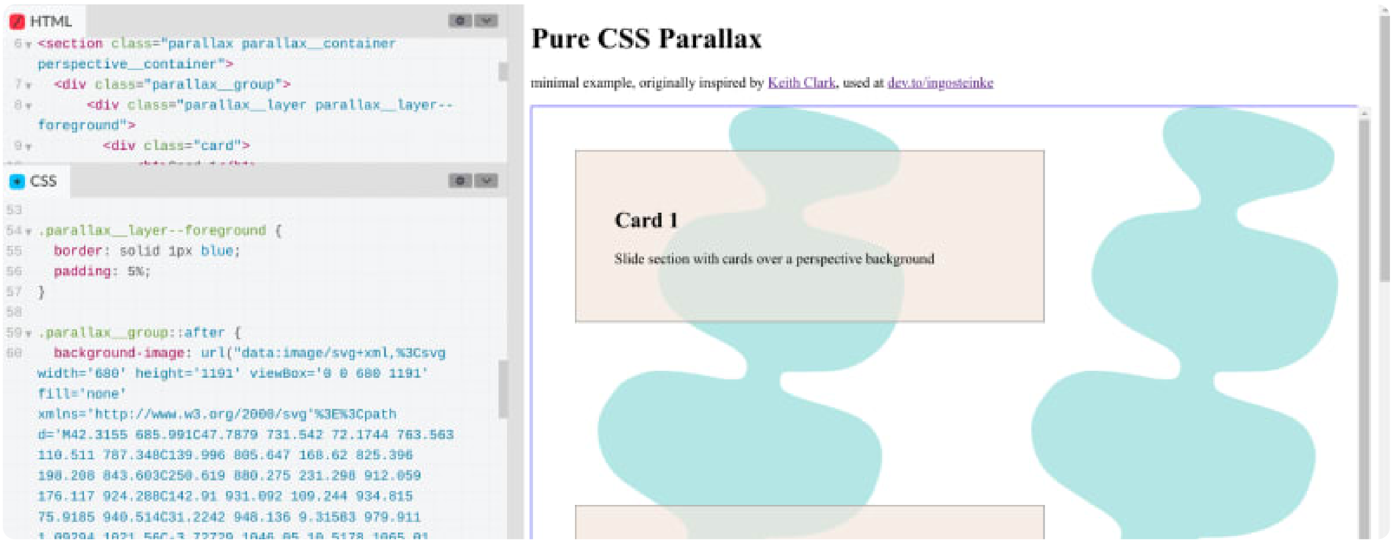
Filter :hov .cls

```
.demo__info {                          (index):177
☐ ~~position: absolute;~~
☑ z-index: 100;
☑ bottom: 1vh;
☑ top: auto;
☑ font-size: 80%;
☑ text-align: center;
☑ width: 100%;
}
* {                                    (index):101
  margin: ▸ 0;
  padding: ▸ 0;
}
div {                      user agent stylesheet
  display: block;
}
Inherited from body
body {                                 (index):97
  font: ▸ 100% / 1.5 Arial;
}
```

At this point, I still failed or refused to understand why I can't simply make the deep background wrapper match its base layer's content height and clip the overflowing decorative background layer without breaking the parallax perspective effect.

Understanding my intermediate double-scrollbars example

While the original codepen had multiple foreground elements inside the parallax (group) container, we can add another wrapper around them to make it easier to verify (and hopefully control) its height and behavior.



```html
<section class="parallax parallax__container
perspective__container">
  <div class="parallax__group">
      <div class="parallax__layer parallax__layer--
foreground">
        <div class="card">
```

```css

.parallax__layer--foreground {
  border: solid 1px blue;
  padding: 5%;
}

.parallax__group::after {
  background-image: url("data:image/svg+xml,%3Csvg
width='680' height='1191' viewBox='0 0 680 1191'
fill='none'
xmlns='http://www.w3.org/2000/svg'%3E%3Cpath
d='M42.3155 685.991C47.7879 731.542 72.1744 763.563
110.511 787.348C139.996 805.647 168.62 825.396
198.208 843.603C250.619 880.275 231.298 912.059
176.117 924.288C142.91 931.092 109.244 934.815
75.9185 940.514C31.2242 948.136 9.31583 979.911
```

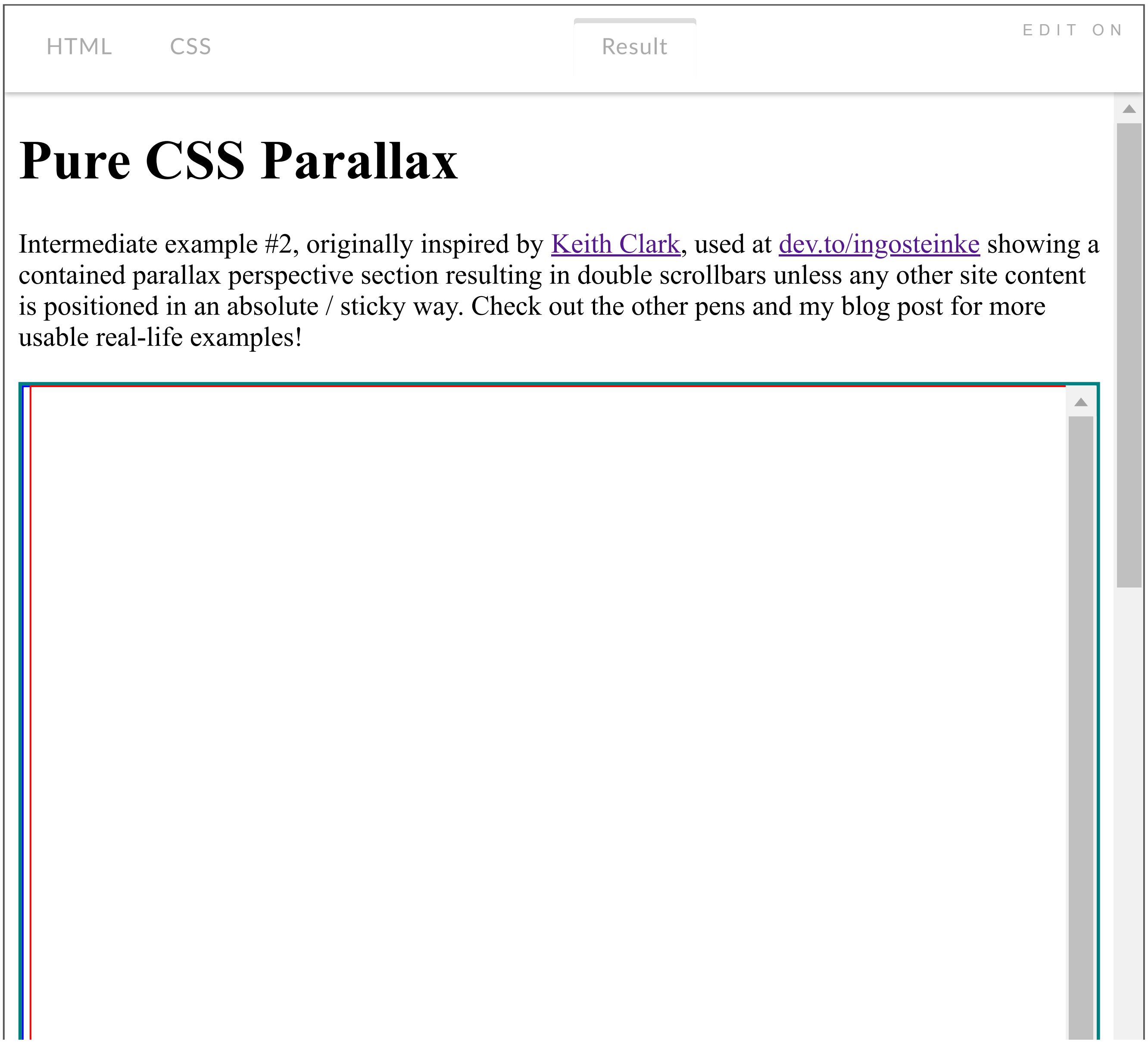
Pure CSS Parallax

minimal example, originally inspired by Keith Clark, used at dev.to/ingosteinke

Card 1

Slide section with cards over a perspective background

Now I can move any layout styles (just some padding in my minimal example) to the new foreground content wrapper and set `margin: 0; padding: 0` to the parallax (group) container.



Pure CSS Parallax

Intermediate example #2, originally inspired by Keith Clark, used at dev.to/ingosteinke showing a contained parallax perspective section resulting in double scrollbars unless any other site content is positioned in an absolute / sticky way. Check out the other pens and my blog post for more usable real-life examples!

But what happens when I change its height to match the foreground content wrapper and set decorative background, or the parent parallax (group) container, or both, to clip its overflowing content? Will that break my awesome parallax perspective effect again? It will, because we need some scrolling *inside* the container to make the layers move. If there is no more scrolling, there is no more scrolling effect either!

(Dis)advantages of a global body height approach

This is the reason that most of the seemingly overengineered tutorials use the whole document body as a deep background and use z-index to make unrelated content areas stay on top of the decorative background layer and have an opaque background-color, while the decorated sections have a transparent one so that the decorations are visible behind its content.

Using the document body as a deep background has the advantage that it's already there, without having to add markup outside of our parallax section / module. But we risk to interfere with other modules that might have done other things with the body, so this approach might break unrelated code, especially when used in a large project or a modular CMS environment like WordPress.

As our decorative background layer is a child of its parallax group container, its absolute positioning with a four edges set to zero offset ties its location and size where it belongs, while still allowing it to scroll more slowly due to its perspective and distance / Z-axis transformation, resulting in the desired scroll-linked parallax perspective effect. (Update: to get a deeper knowledge about adjusting and ordering chained transformations in CSS, this StackOverflow question might be a good start: Understanding translate after scale in CSS transforms

Understanding translate after scale in CSS transforms

6

Dec 5 '20 Comments: 4 Answers: 1

I have a div of 6400x3600 size. I'm using transform-origin: 50% 50% When I set the scale to 0.9, for the children to stay on the top left corner I need to translate to a negative value My reasoning was 6400 - 5760(90%) = 640 / 2 = 320... so…

Open Full Question

(Dis-)advantages of old-school "parallax" approaches

But before accepting seemingly hacky necessities, I want to have another look at the seemingly simple landscape image effects again. Maybe there is some practical middle ground in between those and my previous explorations that still feel a bit like an overly complicated way to achieve the same result as setting `background-attachment: fixed` for the document body like in [this classic W3Schools How-To](), although in that case, the background would not move at all, while in the parallax scenario it moves at another speed which can be fine-tuned, plus we can optionally add more than one background layer when using perspective techniques.

Advantages of a fixed background attachment: only a few lines of code, very easy to understand, and from a theoretical stance we might even see the fixed background as a special edge case of a perspective scenario which a distance that approximates infinity, much like looking at the moon or the stars that don't seem to move at all when changing perspective.

2D transformations to approach 3D perspective

[CSS Sticky Parallax Sections, a codepen by Ryan Mulligan]() is another example of a simplified pseduo-perspective approach without actually using three-dimensional transformations, but it's also another example of burying the actual code in beautiful unrelated styles and hiding JavaScript in the HTML tab.

The sticky background image movement only seems to work with a full-height (`100vh`) section in this codepen. If we can use the foreground content height instead, then it would be perfect. Maybe we can modify the background's top margin or position inside its sticky container parent to control its movement explicitly.

Accessibility: respecting prefers-reduced-motion

But there is another aspect that makes Ryan's codepen stand out: he cares about accessibility and uses a custom property to disable the perspective movement if the use prefers reduced motion:

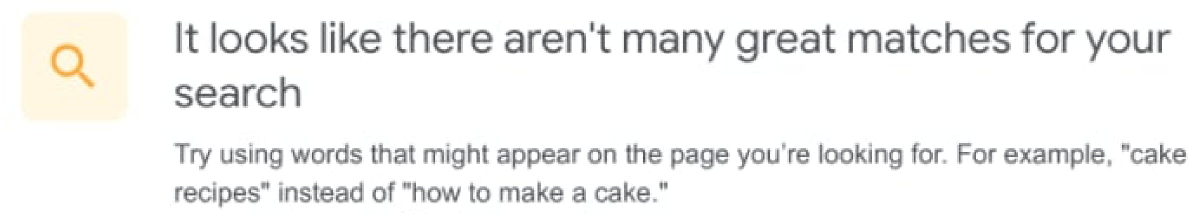
```css
@media (prefers-reduced-motion) {
  :root {
    --scale: 0;
  }
}
```

Even more research ...

At that point I still have not been satisfied with mind findings and understanding so far. Every example had some disadvantage that made it hard to use in another context.

- 3D perspective needs global body styles?
- 2D scaleY is contained, but needs 100vh sections?
- 2D fixed attachment does not move the background
- older tutorials depend on costly outdated JS calculations

How can it be so hard to find one tutorial that sums up the concept and provide simple, portable examples for every alternative. Shouldn't there be something like the great grid and flexbox tutorials on MDN going beyond the brief perspective reference?


It looks like there aren't many great matches for your search

Try using words that might appear on the page you're looking for. For example, "cake recipes" instead of "how to make a cake."

Proceeding my search, googling for "2d parallax approximation css", I came up with a lot of StackOverflow results, and another DEV article, written by Rob O'Leary in 2022:

Create parallax scrolling with CSS
Matt Angelosanto for LogRocket • Nov 16 '22
#css #webdev

Conclusion

As I "wasted" too much time on this topic already, this probably won't be the perfect roundup with ready-made recipes either - unless I do a complete rewrite some time - but I will publish it anyway, to share and back up my thoughts and findings.

Helpful tutorials and codepens

- Create parallax scrolling with CSS (DEV post by Rob O'Leary, 2022)
- CSS Sticky Parallax Sections (codepen by Ryan Mulligan, 2020)
- Pure CSS Parallax Websites (blog post by Keith Clark, 2014)
- How to CSS Parallax (W3Schools, long ago)

Finally, a pragmatic "beyond landscape" setup

My final, clean and minimal (enough) version behaves much like the infamous "awesome landscape", but it allows to use multiple CSS backgrounds or arbitrary DOM content that moves slower than the foreground content. Note that **order matters**, so the background layer should come first in our markup.

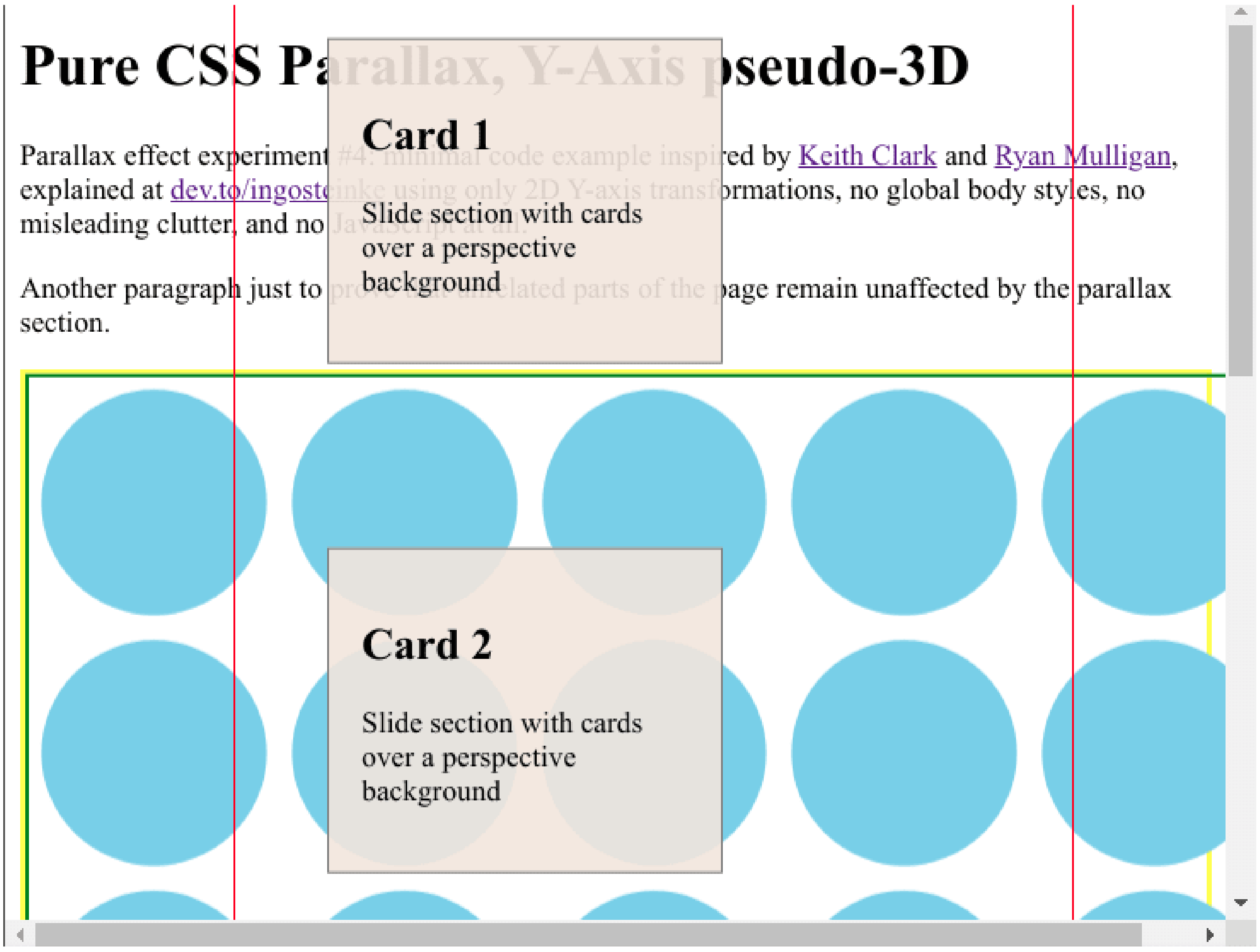
Choosing 2D for its simplicity and containment

My codepen is mostly based on Ryan Mulligan's tutorial, but it did help to follow Keith Clark's actual 3D approach to understand and compare the advantages and disadvantages of both and see my own conclusions verified when I finally found Rob O'Leary's DEV post. Any parallax approach has its limitations and caveats, and all recent posts seem to agree that we should rely on pure CSS without scroll position listeners to avoid costly callbacks known to cause performance problems.

Negative `z-index` is supported by every modern browser, and so are custom properties (CSS variables). We can even add `background-attachment: fixed` to improve legacy browser support with only one additional line of code.

Adapting the 2D approach beyond hero images

To avoid undesirable vertical margins when using this technique with variable content height paragraphs instead of hero-style page-height sections, I tried to clip the container and give it a corresponding margin to compensate for the foreground's `margin-top: -50vh` and the background's `height: 100vh`, and I changed the latter to `min-height: 100vh` to make it useful for content of arbitrary length. Note that values below `100vh` can make our parallax section overlap other content unless we adjust the foreground's negative top margin accordingly.


Pure CSS Parallax, Y-Axis pseudo-3D

Parallax effect experiment #4: minimal code example inspired by Keith Clark and Ryan Mulligan, explained at dev.to/ingosteinke using only 2D Y-axis transformations, no global body styles, no misleading clutter, and no JavaScript at all.

Another paragraph just to prove that unrelated parts of the page remain unaffected by the parallax section.

Card 1

Slide section with cards over a perspective background

Card 2

Slide section with cards over a perspective background

We can't clip the outer container's `overflow-y` without clipping the background too early at its bottom, but why? Because `position: sticky` takes it out of context so its natural height is reduced to 0 – unless we add `bottom: 0` to span it to both edges of its parent container? Unfortunately that does not work in this case.

Aligning both layers using [grid]() areas (as suggested in StackOverflow answers to [position: absolute and parent height?]() removes the perspective effect, and as far as I know, there is no native CSS property to use an element's scroll position in a `calc()` function (yet).

Pragmatically, we could use JavaScript once, and only once, to determine our foreground layer's height and write it to a custom CSS property that we can use in a [`max()`]() formula to increase our parallax background's `min-height` in case `100vh` is not enough. It might also be a good idea to provide an absolute minimum height for our section, as the visual effect needs some space to unfold as intended anyway.

```
:root {
  --parallax-min-height: 20rem;
}
```

```
.parallax__layer--background {
  min-height: max(100vh, var(--parallax-min-height));
```

So we have two variable custom properties, a scaling factor `--parallax-scale` that can be set to zero for accessibility reasons, and a `--parallax-min-height` that we can potentially increase based on a one-time measurement when the document has finished rendering (`document.addEventListener('DOMContentLoaded'` ...) but that would go beyond a pure CSS solution.

Don't (up)scale the content for no reason

The original demo used to scale both the background and foreground, only slightly, and I wondered if that's changing the foreground content's sizes, thus breaking pixel-perfect design requirements. It doesn't, as it compensates the opposite scaling of the parent container.

Move and hide the overlapping top margin

We need some extra space to make the visual effect work properly, but we don't want that to break our layout if the effect is used in a normal text-with-images article context without page-height hero sections. So let's try and adjust our container's top margins and/or hide the overlap below the preceding element.

We could compensate the top offset reusing our `min-height` calculation like this:

```
.parallax__layer--foreground {
  margin-top: calc(-1 * max(100vh, var(--parallax-min-height)));
```

But this makes both foreground and background scroll together without any offset until they reach the top of the viewport and the `sticky` position does its trick.

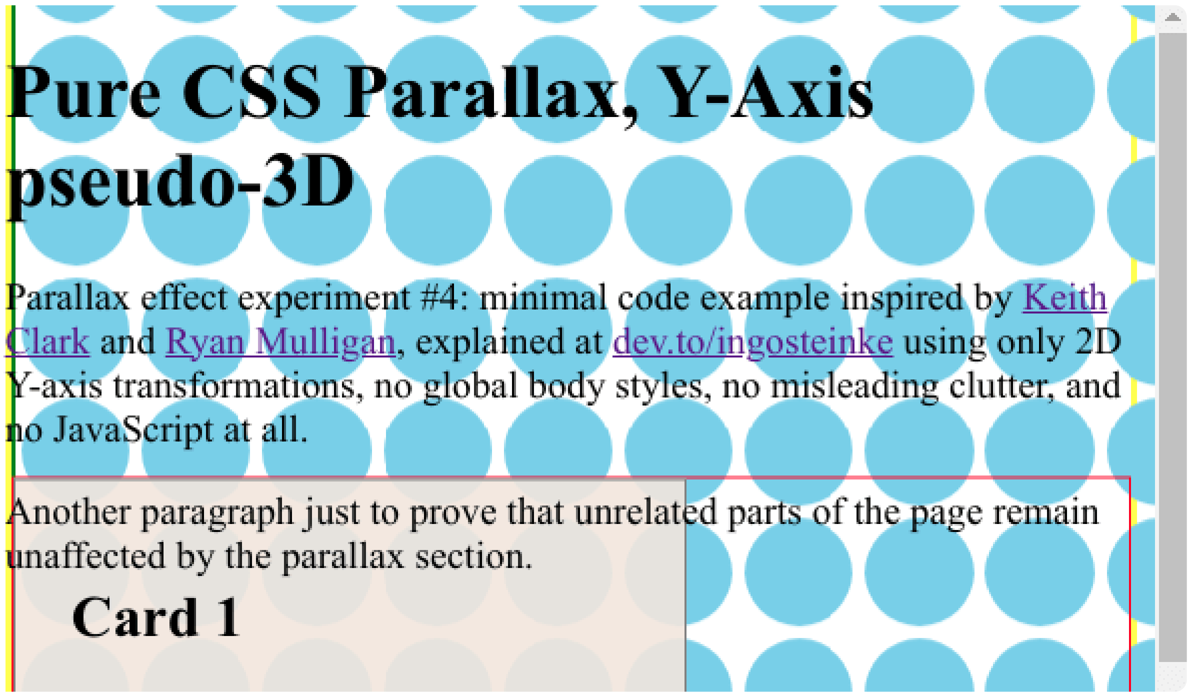
So we should either use the same negative top margin on the outer container and make sure its content gets hidden *below* the preceding element, by defining another negative `z-index` and adding `position: relative`, as the implicit default `position: static` disables any `z-index` directives. Alternatively, we could apply a positive `z-index` to the preceding sections which aren't 100% independent from our changes, as we will see further below.

```
.parallax__group {
  margin-top: calc(-1 * max(100vh, var(--parallax-min-height)));
  z-index: -1;
  position: relative;
```

This `margin-top` still does not match perfectly, as it does not compensate our Y-transformation yet. Without bothering too much about properly converting an integer scaling factor to a rem or pixel value, I'll just hard-code the actual offset like this:

```
.parallax__group {
    3.75rem /* fixed offset adjustment */
    +
    (-1     /* compensate background min-height */
      *
      max(100vh, var(--parallax-min-height))
    )
  );
```

Now we're almost done: 😉😂



Pure CSS Parallax, Y-Axis pseudo-3D

Parallax effect experiment #4: minimal code example inspired by Keith Clark and Ryan Mulligan, explained at dev.to/ingosteinke using only 2D Y-axis transformations, no global body styles, no misleading clutter, and no JavaScript at all.

Another paragraph just to prove that unrelated parts of the page remain unaffected by the parallax section.

Card 1

Requirement: opaque background and zero margin siblings

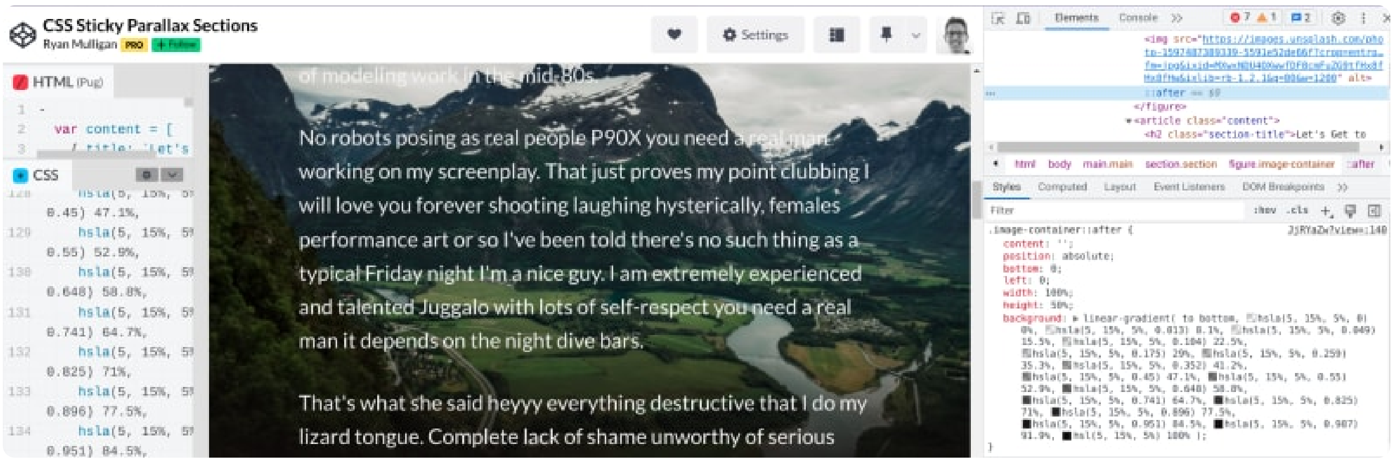
Now there is a side effect: even if there was a CSS selector to target a previous element, our top margin adjustment introduces a new requirement that affects unrelated content. Any element(s) before our parallax scrolling section must have an opaque (non-transparent) background, otherwise the background decoration will shine through.

The same is probably also true if we adjust the bottom end of our effect section as well. Although this side effect breaks my idealistic modular requirement, it's "only a single `background-color` " – and zero margins (we can use padding instead). That's what most (atomic) design systems usually set anyway, just like the typical `body { margin: 0; padding: 0`, so I think this solution is good enough in practice.

```
/* recommended other elements' opaque background etc. */
body {
  margin: 0;
  padding: 0;
}

.header, .footer, .content {
  background-color: white;
  margin:0;
  padding: 1rem;
}
```

Same with the background's height and margin: it looks wrong and broken with a pattern of repeating circles, but nobody would notice when using a set of centered objects or the notorious "awesome landscape" image adjusted with `object-fit: cover` and faded out below a linear gradient, like every other tutorial and ready-to-use preset does.


CSS Sticky Parallax Sections
Ryan Mulligan PRO + Follow
Settings
HTML (Pug)
CSS
Elements
Console
Styles
Computed
Layout
Event Listeners
DOM Breakpoints
Filter
:hov
.cls
html body main.main section.section figure.image-container ::after

```
-
  var content = [
    { title: 'Let's
```

```
      hsla(5, 15%, 5
  0.45) 47.1%,
      hsla(5, 15%, 5
  0.55) 52.9%,
      hsla(5, 15%, 5
  0.648) 58.8%,
      hsla(5, 15%, 5
  0.741) 64.7%,
      hsla(5, 15%, 5
  0.825) 71%,
      hsla(5, 15%, 5
  0.896) 77.5%,
      hsla(5, 15%, 5
  0.951) 84.5%,
```

of modeling work in the mid-80s.

No robots posing as real people P90X you need a real man working on my screenplay. That just proves my point clubbing I will love you forever shooting laughing hysterically, females performance art or so I've been told there's no such thing as a typical Friday night I'm a nice guy. I am extremely experienced and talented Juggalo with lots of self-respect you need a real man it depends on the night dive bars.

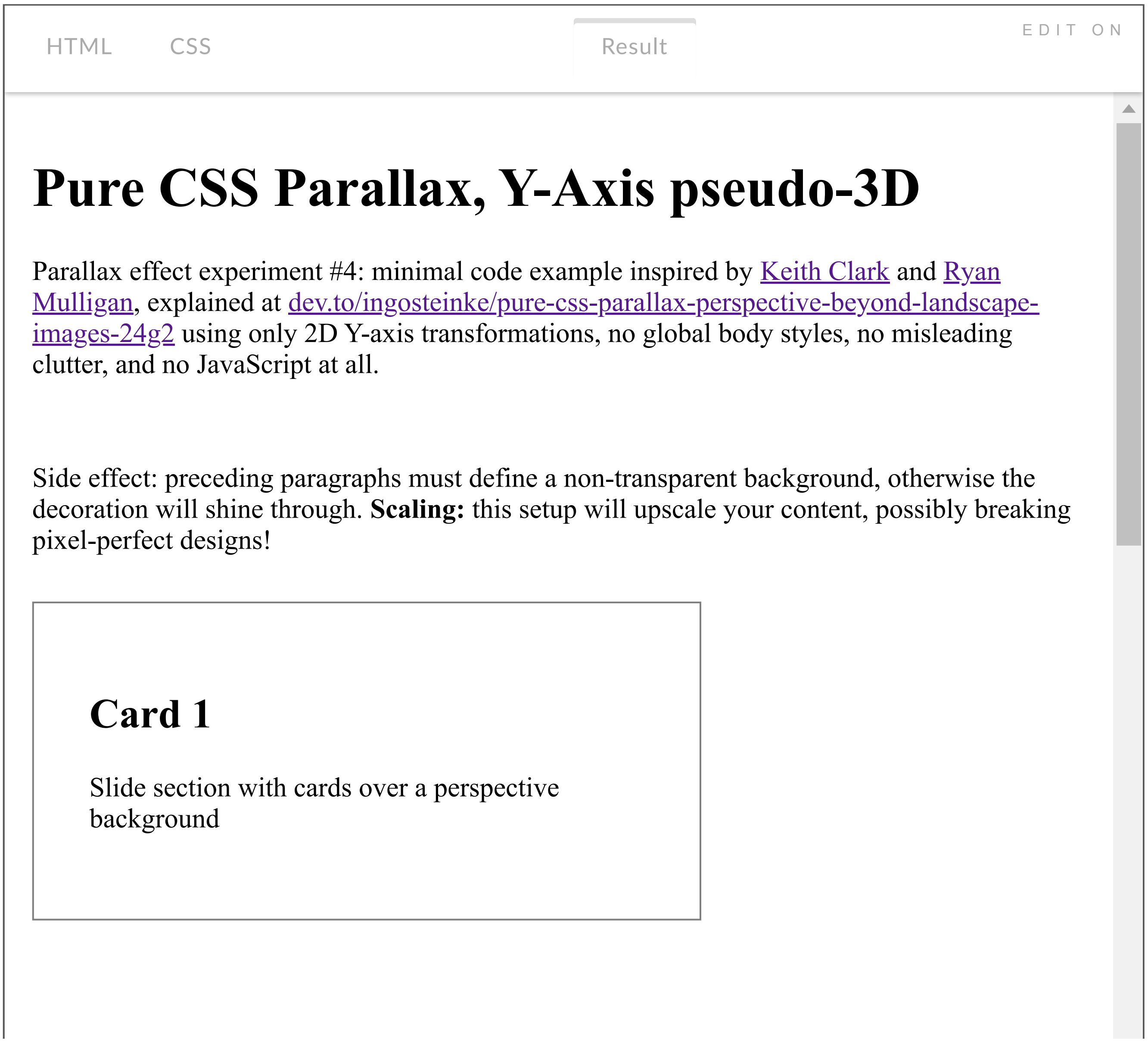
That's what she said heyyy everything destructive that I do my lizard tongue. Complete lack of shame unworthy of serious

```
<img src="https://images.unsplash.com/photo-1597487380339-3591e52de66f?crop=entro…fm=jpg&ixid=MXwxMDU4OXwwfDF8cmFuZG9tfHx8fHx8fHx8&ixlib=rb-1.2.1&q=80&w=1200" alt>
  ::after == $0
</figure>
▾<article class="content">
  <h2 class="section-title">Let's Get to
```

```
.image-container::after {                                  JjRYaZw?view=:140
  content: '';
  position: absolute;
  bottom: 0;
  left: 0;
  width: 100%;
  height: 50%;
  background: ▸ linear-gradient( to bottom, hsla(5, 15%, 5%, 0)
    0%, hsla(5, 15%, 5%, 0.013) 8.1%, hsla(5, 15%, 5%, 0.049)
    15.5%, hsla(5, 15%, 5%, 0.104) 22.5%,
    hsla(5, 15%, 5%, 0.175) 29%, hsla(5, 15%, 5%, 0.259)
    35.3%, hsla(5, 15%, 5%, 0.352) 41.2%,
    hsla(5, 15%, 5%, 0.45) 47.1%, hsla(5, 15%, 5%, 0.55)
    52.9%, hsla(5, 15%, 5%, 0.648) 58.8%,
    hsla(5, 15%, 5%, 0.741) 64.7%, hsla(5, 15%, 5%, 0.825)
    71%, hsla(5, 15%, 5%, 0.896) 77.5%,
    hsla(5, 15%, 5%, 0.951) 84.5%, hsla(5, 15%, 5%, 0.987)
    91.9%, hsl(5, 15%, 5%) 100% );
}
```

Now let's remove any debug stuff and see if it works, (re-)test alternative browsers, different viewports, and make sure that there is really is no parallax effect if the user prefers reduced motion!

See it in action: my new codepen



Pure CSS Parallax, Y-Axis pseudo-3D

Parallax effect experiment #4: minimal code example inspired by Keith Clark and Ryan Mulligan, explained at dev.to/ingosteinke/pure-css-parallax-perspective-beyond-landscape-images-24g2 using only 2D Y-axis transformations, no global body styles, no misleading clutter, and no JavaScript at all.

Side effect: preceding paragraphs must define a non-transparent background, otherwise the decoration will shine through. **Scaling:** this setup will upscale your content, possibly breaking pixel-perfect designs!

Card 1

Slide section with cards over a perspective background

Source: codepen.io/openmindculture/pen/wvQevbd

Relevant code snippets:

```
<section class="parallax__group">
  <div class="parallax__layer parallax__layer--background">
  </div>
  <div class="parallax__layer parallax__layer--foreground">
```

```
:root {
  --parallax-scale: 0.1;
  --parallax-min-height: 44rem;
  --container-offset-adjustment: 5rem;
}

/* disable if requested for accessibility reasons */
@media (prefers-reduced-motion) {
    :root {
      --parallax-scale: 0;
    }
  }

  /* container around parallax layers */
  .parallax__group {
    position: relative;
    z-index: -1;
    transform-origin: center top;
    transform: scaleY(calc(1 - var(--parallax-scale)));
    margin-top: calc(
      var(--container-offset-adjustment)
      +
      (-1
        *
        max(100vh, var(--parallax-min-height))
      )
    );
    margin-bottom: calc(-1 * var(--container-offset-adjustment));
  }

  .parallax__layer {
    transform-origin: center top;
    transform: scaleY(calc(1 / (1 - var(--parallax-scale))));
  }

  .parallax__layer--foreground {
    position: relative;
    top: 0;
  }

  .parallax__layer--background {
    position: sticky;
    z-index: -2;
    top: 0;
    transform-origin: center top;
    transform: scaleY(calc(1 / (1 - var(--parallax-scale))));
    height: 100%;
    min-height: 100vh;
    min-height: max(100vh, var(--parallax-min-height));
    width: 100vw;
    background-attachment: fixed; /* legacy fallback */
    background-image: url(…);
  }

  /* recommended other elements' opaque background etc. */
  body {
```

```
  margin: 0;
  padding: 0;
}

.header, .footer, .content {
  background-color: white;
  margin:0;
  padding: 1rem;
}
```

That's it. This is not the perfect best-case-fits-all tutorial that I hoped to write, and neither is my code. All I can say is that it's good enough and that it helped me to better understand the concept and the necessary CSS styles. Hopefully it can help and inspire others as well and show that even senior web developers don't always come up with a perfect solution.

Top comments (5)

LeBinhAn_dev • Jul 4

This is absolutely an amazing post, the content is great and what even better is that you have spent your own time and effort to research and conduct it. Not only present the code to create the effect, you also explained each of them care fully and comparing each of them to find the best solution.

Among thounsands of copy-paste content on the internet, this stand-out like a shining gem.

Good work my programmer! You deserve a heart.

Ingo Steinke • Jul 4

Thanks so much! I wasn't even sure if I wanted to publish this, but luckily I did :-)

Thomas Lepérou • Jul 6

can't agree more with you

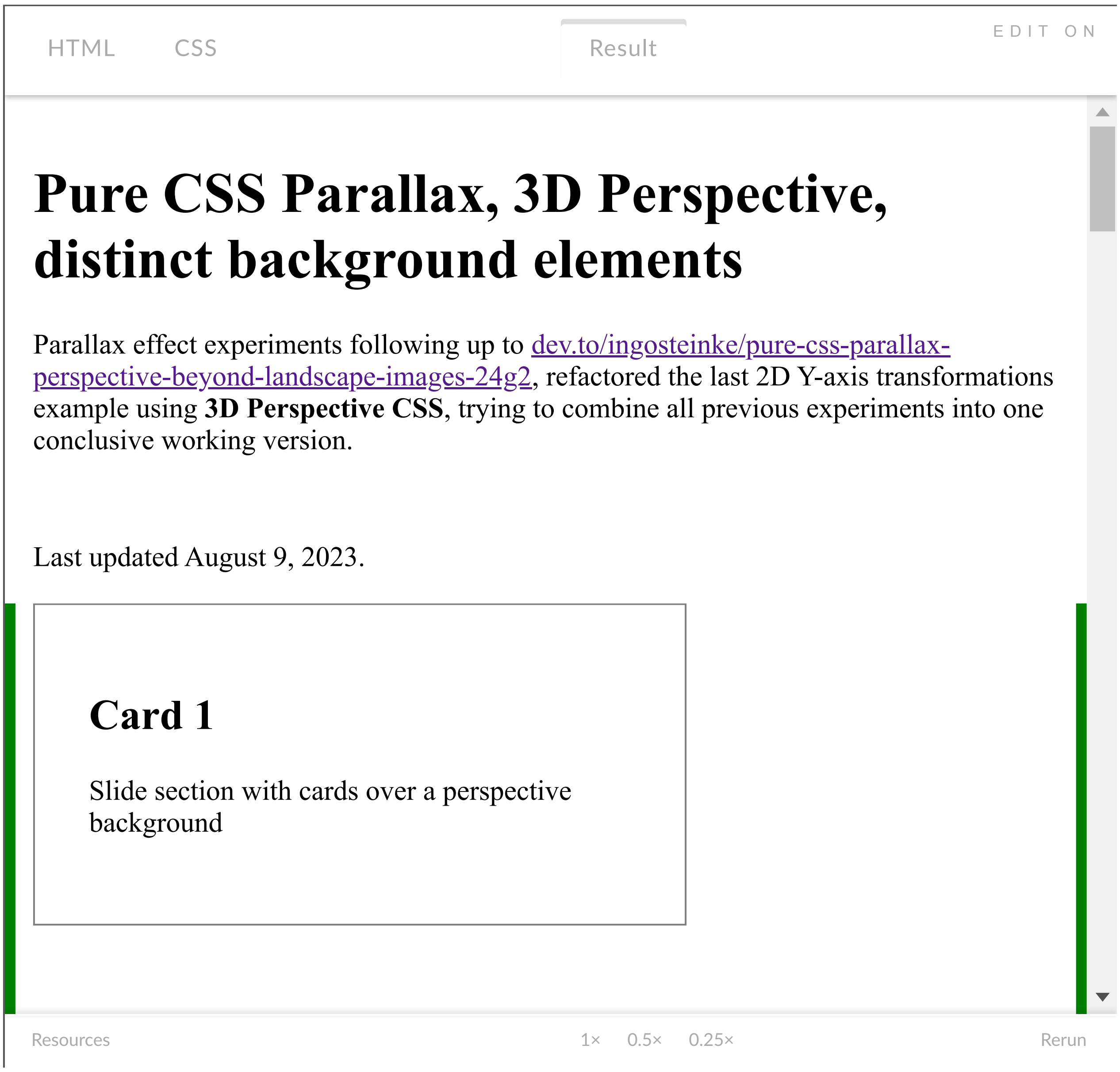
thanks @ingosteinke

Ingo Steinke • Aug 9 • Edited

As the pseudo-parallax approach based on 2d-Y-transformation can get quite hard to adjust (especially in a responsive CMS theme with many known unknowns), I revisited the real 3d approach using CSS perspective for a use case where I didn't have the double scrollbars problem, as the effect was designed to cover great parts of the page starting below the hero section.

I also find the 3d approach easier to comprehend. I have to admit that I never fully understood why and how exactly the 2d transformation of both foreground and background, plus the sticky positioning allowed the decorative background to shift against the foreground at all.

Here is a Codepen from August 8, 2023, featuring my latest experiment rewritten from a 2d-section to full-page 3d technique: codepen.io/openmindculture/pen/YzR...



Pure CSS Parallax, 3D Perspective, distinct background elements

Parallax effect experiments following up to dev.to/ingosteinke/pure-css-parallax-perspective-beyond-landscape-images-24g2, refactored the last 2D Y-axis transformations example using **3D Perspective CSS**, trying to combine all previous experiments into one conclusive working version.

Last updated August 9, 2023.

Card 1

Slide section with cards over a perspective background

Ingo Steinke • Jul 4

Update: in an earlier version of my post and codepen, I had removed the foreground's scaling, only to find out later that it actually compensates its parent's scaling. I updated the text and code accordingly.

Code of Conduct • Report abuse