



CDS 2020: New CSS properties for performance

There are two new CSS properties that may help with layout shift core vitals issue.

The `content-visibility` property allows us to tell the browser when to render an element. The “magic” value here is `auto`, which tells the browser not to render an element until it’s visible in the viewport.

This makes page loads significantly faster, if the browser doesn’t need to render above-the-fold content initially.

The `content-visibility` property works hand in hand with the `contain-intrinsic-size` property to prevent potential issues with Cumulative Layout Shift.

Because the browser isn’t rendering the element fully on load, there may be some shift when it renders the element. In order to minimise shifts happening on the page, we can use the `contain-intrinsic-size` property to specify dimensions for the element even without the content being rendered.

I work a lot with Youtube iframe embeds wrapped around a div with class `video`. If you have many of these in the page it may cause layout shifts. One way

to minimize the CLS for these videos may be to use these new attributes in addition to other attributes.

```
.video {  
  /* other attributes go here */  
  
  content-visibility: auto;  
  /* set height to 560px x 315px */  contain-intrinsic-size: 560px 315px;  
}
```

Improving performance with content-visibility

Depending on the layout of your site, you may want to study where in your layout you want to use some of these new attributes on their own to improve your site's rendering.

Take the following layout, which may look familiar if you've worked in WordPress before, it's modeled after a WordPress site's blog index page:

```
<html>  
<head><head>tle>My demo</title>  
</head>  
<b</title>v>  
  <!-- navigation goes here -->  
</nav>  
  
<h1>Site Title</h1>  
  
<p>introductory content</p>  
  
<article class="posts post1">  
  <h2>post title</h2>  
  
  <p>Post content</p>  
</article>
```

```
<article class="posts post2">
  <h2>post title</h2>

  <p>Post content</p>
</article>

<article class="posts post3">
  <h2>post title</h2>

  <p>Post content</p>
</article>
</body>
</html>
```

We could use [content-visibility](#) to let the browser handle containment. When displayed in the same page, we don't know how long each article will be and how much will it take to render the content. Using `content-visibility` we're telling the browser that it can skip rendering the content of the matching elements until it comes into the viewport.

The CSS looks like this:

```
article {
  /* other attributes go here */

  content-visibility: auto;
}
```