Native lazy loading in Chrome

Addy Osmani posted a note on twitter about native lazy loading support that will, hopefully, appear in Chrome 75 (canary builds as I write this). This is awesome and I hope that other browsers will implement this as a native feature but it introduces complexities that we need to evaluate before we implement this in development.

Before we start

Native lazy loading works for both images and iframes but it's behind two flags. Surprisingly neither of them is `Experimental Web Platform features`.

Go into `chrome://flags` and enable the following flags if you want both elements to work with lazy loading:

- `enable-lazy-image-loading`
- `enable-lazy-frame-loading`

Restart your browser and you're good to go.

The basics

- `loading="lazy"` Lazy loads an offscreen image when the user scrolls near it
- `loading="eager"` Loads an image right away instead of lazy-loading. This is the same as not using the attribute at all
- `loading="auto"` lets the browser decides whether or not to lazy load the element

```
<img src="building1.jpg" loading="lazy" alt=".."/>
<img src="building1.jpg" loading="eager" alt=".."/>
<img src="building1.jpg" loading="auto" alt=".."/>
```

The lazy loading feature will also work with `picture` elements as long as you add the `loading` attribute to the fallback image element. If I understand it correctly

the `img` element drives the display of any image inside the `picture` element so, if you add `loading` to it, whatever image loads will be lazy loaded.

```
<picture>
  <source media="(min-width: 40em)" srcset="big.jpg 1x, big-hd.jpg 2x">
  <source srcset="small.jpg 1x, small-hd.jpg 2x">
  <img src="fallback.jpg" loading="lazy">
</picture>
```

Same thing happens if the image has `srcset` attributes. As long as the image has the `loading` attribute set to `lazy` then the image will be lazy loaded.

```
<!-- Lazy-load an image that has srcset specified -->
<img src="small.jpg"
     srcset="large.jpg 1024w, medium.jpg 640w, small.jpg 320w"
     sizes="(min-width: 36em) 33.3vw, 100vw"
     alt="A rad wolf" loading="lazy">
```

The one example I haven't seen elsewhere is for iframes. The example below shows a Youtube iframe embed set up for lazy loading. The same values apply here as they apply for images.

```
<iframe   loading="lazy"
          width="560" height="315"
          src="https://www.youtube.com/embed/yY1FSsUV-8c"
          frameborder="0"
          allow="accelerometer; autoplay; encrypted-media; gyroscope; pict
          allowfullscreen></iframe>
```

Feature Detection and Lazy Loading Crossbrowser

I'm using the yall lazy loading library in this example and initialize it when we determine the browser doesn't support native lazy loading. We need to load the library using something like the following command

```
<script defer src="scripts/yall.js"></script>
```

This will load the library every time we load the page. I am willing to take the extra 2Kb (and potentially the additional weight of an intersection observer polyfill) if it means that page will load faster because not all images are loaded when the page loads.

Once we've loaded yall.js, we check if the browser supports native lazyloading (`'loading' in HTMLImageElement.prototype`).

If it does, we store all the images we want to work with (the ones with a `lazyload` class) in a variable and then, for each of those images, we copy the `data-src`attribute to the source (`src`) attribute. This will lazy load the images.

If the browser doesn't support native lazy loading then we load the lazysizes script and initialize it. The script will take the `data-src` attribute and use it to lazy load the images instead.

```
(async () => {
  if ('loading' in HTMLImageElement.prototype) {
    const images = document.querySelectorAll('img.lazy');
    images.forEac'img.lazy'
        img.src = img.dataset.src;
    });
  } else {
    // Make sure the library is already loaded
    // Initialize yall
    document.addEventListener("DOMContentLoaded", yall);
  }
})();
"DOMContentLoaded"
```

Using the script above we can use the following to load an image above the fold immediately.

```
<!-- Let's load this in-viewport image normally -->
<img src="hero.jpg" alt=""/>
```

And this is the code we use to lazy load an image. Note how it doesn't have a `src` attribute because we'll change it programmatically.

```
<!-- Let's lazy-load the rest of these images -->
<img  data-src="image1.jpg"
      loading="lazy" alt=""
      class="lazy"/>
```

I've got a working example in this pen.

References

- Blink intent to ship
- HTML specification PR
- Explainer
- Demo large image list