



Lazy Loading MPEG-DASH

After images the biggest payloads in my pages are Youtube embedded videos and since I already lazy load images I've decided to explore how to best lazy load embedded Youtube iframes; I embed Vimeo videos far less often so I'm not going to lazy load them.

I have the following requirements for any lazy loading solution:

- The video must be responsive
- Any external script, stylesheets and assets must be less than 5kb in size

The original

This is what I normally do when working with video on the web and in my Wordpress blogs. I take the `iframe` element directly from Youtube and wrap it around a `div` with a class of `video` to style it accordingly

```
<div class="video">
<video width="560" height="315"
  src="path/to/manifest.mpd"
  poster="path/to/poster/image.png">
</video>
</div>
```

The CSS is simple and styles both the `iframe` and paragraphs inside the `div`.

```
div.video {
  clear: both;
  display: block;
  margin: 1em auto;
  max-width: inherit;
  text-align: center;
}

div.video p {
```

```
font-style: italic;
font-weight: 700;
margin-top: -.125em
}

video {
  margin: 1.5em 0
}
```

The issue we try to resolve with lazy loading is that each embedded video loads many additional resources whether we are interacting with the video or not. Those additional HTTP connections will slow down overall page load times and gives a lower Page Speed Score.

Furthermore the longer it takes for a page to display content to the user and become interactive the more likely the user it is to leave the page and not come back. Hence this experiment... Using [digital inspiration's lazy loading method](#) as a model we'll explore how to use it, what it does and whether it's actually worth using it or not.

The lazy loaded version

Lazy loading is a design pattern commonly used in computer programming to defer initialization of an object until the point at which it is needed. It can contribute to efficiency in the program's operation if properly and appropriately used.

— [Wikipedia](#)

In the context of a web page/application I use lazy loading to prevent an item from loading until it becomes visible on the screen (comes into the viewport). Once the video comes into view we wait for user's input by either clicking or tapping the video before playback begins.

We use the HTML below to indicate where to place the video and what video (using the video Youtube ID) to place there.

```
<div class="youtube-player" data-id="VIDEO_ID"></div>
```

The Javascript below will build three elements:

- A div element to hold the video iframe
- An img element containing the poster image and a play button
- The iframe element that contains the video and attributes like autoplay and no related videos at the end

```
document.addEventListener("DOMContentLoaded",
function() {
    let div;
    let n;
    let v = document.getElementsByClassName("youtube-player");
    for (n = 0; n < v.length; n++) {
        div = document.createElement("div");
        div.setAttribute("data-id", v[n].dataset.id);
        div.innerHTML = loadThumb(v[n].dataset.id);
        div.onclick = loadIframe;
        v[n].appendChild(div);
    }
});

function loadThumb(id) {
    let thumb = '';
    let play = '<div class="play"></div>';
    return thumb.replace("ID", id) + play;
}

function loadIframe() {
    let iframe = document.createElement("iframe");
    let embed = "https://www.youtube.com/embed/ID?autoplay=1&rel=0";
    iframe.setAttribute("src", embed.replace("ID", this.dataset.id));
    iframe.setAttribute("frameborder", "0");
    iframe.setAttribute("allowfullscreen", "1");
    this.parentNode.replaceChild(iframe, this);
}
```

Finally, we use CSS to style the video and programmatically add the play button. as an overlay.

```
.youtube-player {
  position: relative;
  padding-bottom: 56.23%;
  /* Use 75% for 4:3 videos */
  height: 0;
  overflow: hidden;
  max-width: 100%;
  background: #000;
  margin: 5px;
}

.youtube-player iframe {
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  z-index: 100;
  background: transparent;
}

.youtube-player img {
  bottom: 0;
  display: block;
  left: 0;
  margin: auto;
  max-width: 100%;
  width: 100%;
  position: absolute;
  right: 0;
  top: 0;
  border: none;
  height: auto;
  cursor: pointer;
  -webkit-transition: .4s all;
```

```
-moz-transition: .4s all;
transition: .4s all;
}

.youtube-player img:hover {
  -webkit-filter: brightness(75%);
}

.youtube-player .play {
  height: 72px;
  width: 72px;
  left: 50%;
  top: 50%;
  margin-left: -36px;
  margin-top: -36px;
  position: absolute;
  background: url("../images/play.png") no-repeat;
  cursor: pointer;
}
```

- <https://www.labnol.org/internet/light-youtube-embeds/27941/>
- <https://webdesign.tutsplus.com/tutorials/how-to-lazy-load-embedded-youtube-videos--cms-26743>
- <http://codepen.io/tutsplus/pen/RRVRro>
- <https://www.sitepoint.com/faster-youtube-embeds-javascript/>