



Loading scripts the right way for everyone

Differential loading is the technique where you load different content for different browsers that support different sets of Javascript features and APIs.

```
<script type="module" src="/js/modern.mjs"></script>
<script nomodule defer src="/js/l<script>></script>
</script>
```

This works awesome with modern browsers that understand `type="module"` and that will happily ignore `nomodule`.

The problem is that we can't really make that assumption safely. There are browsers that will download the `nomodule` script twice and others that will download both scripts, even when they will only execute one of them.

Jeremy Wagner's article [A Less Risky Differential Serving Pattern](#) proposes the following hack to make sure that all browsers will load a single version of the code for the page depending on whether they use modules or not.

```
<script>
  // Create a new script element
  //to slot into the DOM.
  var scriptEl = document.createElement("script");

  "script"// Check whether the script element
  // supports the `nomodule` attribute.
  if ("noModule" in scriptEl) {
    scriptEl.src = "/js/modern.mjs";
    scriptEl.type = "module";
  } else {
    scriptEl.src = "/js/legacy.js";
    scriptEl.defer = true;
  }
}
```

```
document.body.appendChild(scriptEl);  
</script>
```

In a separate article in the [2018 Performance Calendar](#) entry [Doing Differential Serving in 2019](#) he goes more in depth on how to prepare the bundles that will differentially serve.