

COMP6080

JS Order + DOM

Presented by
Christian Scott

Canva

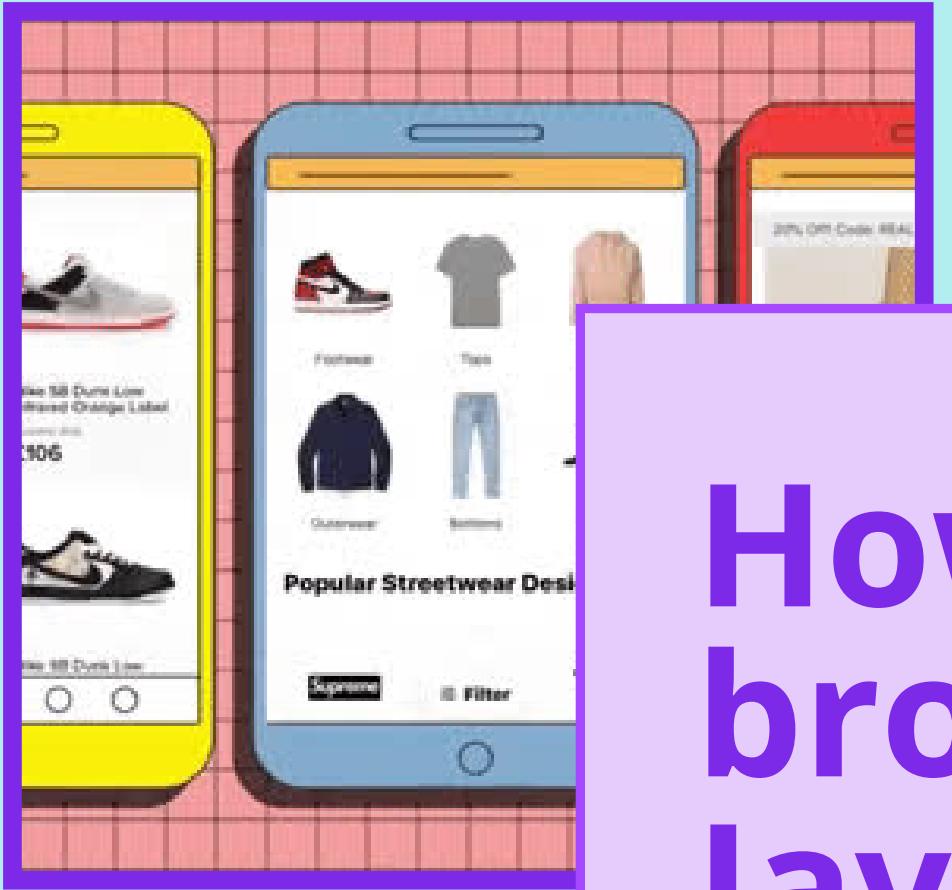


Inline

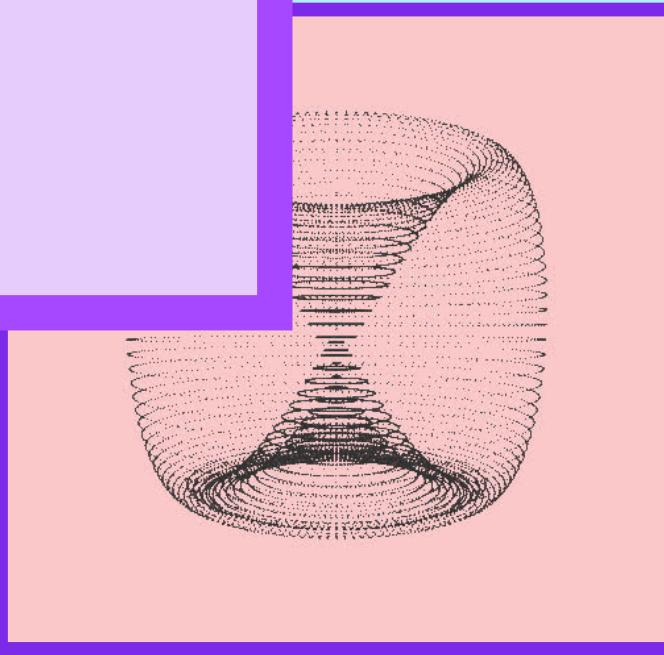
```
<script>
  console.log("Hello!");
</script>
```

External

```
<script src="hello.js">
</script>
```



How does the browser run JavaScript?



Will the heading change?

```
<body>
  <script>
    const heading = document.querySelector('h1');
    if (heading !== null) {
      heading.textContent = 'I changed!';
    }
  </script>
  <h1>I did not change</h1>
</body>
```

Will the heading change?

No.

```
<body>
  <script>
    const heading = document.querySelector('h1');
    if (heading !== null) {
      heading.textContent = 'I changed!';
    }
  </script>
  <h1>I did not change</h1>
</body>
```

Will the heading change?

```
<body>
  <h1>I did not change</h1>
  <script>
    const heading = document.querySelector('h1');
    if (heading !== null) {
      heading.textContent = 'I changed!';
    }
  </script>
</body>
```

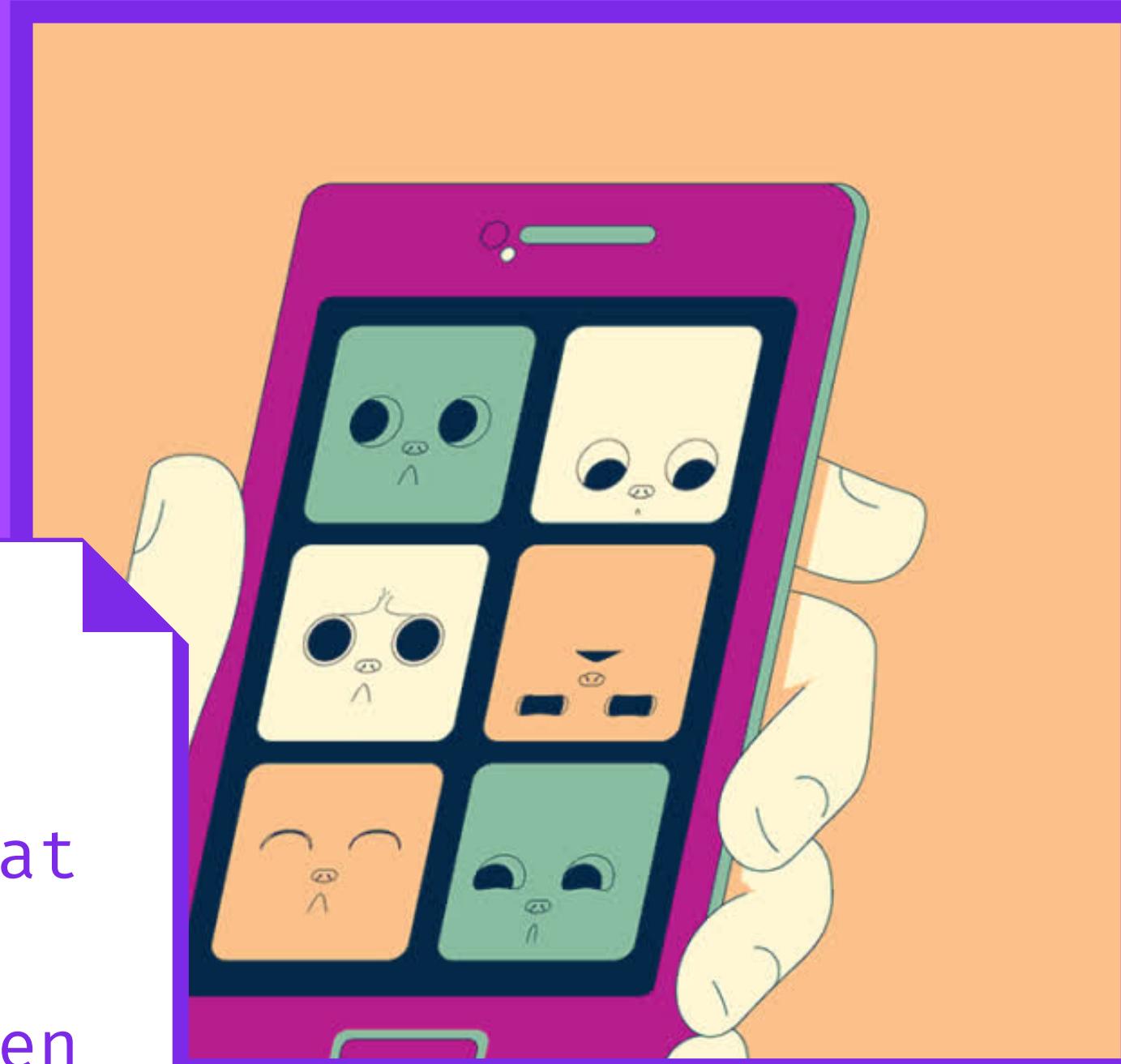
Will the heading change?

Yes!

```
<body>
  <h1>I did not change</h1>
  <script>
    const heading = document.querySelector('h1');
    if (heading !== null) {
      heading.textContent = 'I changed!';
    }
  </script>
</body>
```

Why?

JS is executed by the browser when it is encountered. This means that in the first case, the heading element has not even been created yet!



Will the heading change?

```
<body>
  <script src="change-heading.js"></script>
  <h1>I did not change</h1>
</body>
```

Will the heading change?

No.

```
<body>
  <script src="change-heading.js"></script>
  <h1>I did not change</h1>
</body>
```

Will the heading change?

```
<body>
  <h1>I did not change</h1>
  <script src="change-heading.js"></script>
</body>
```

Will the heading change?

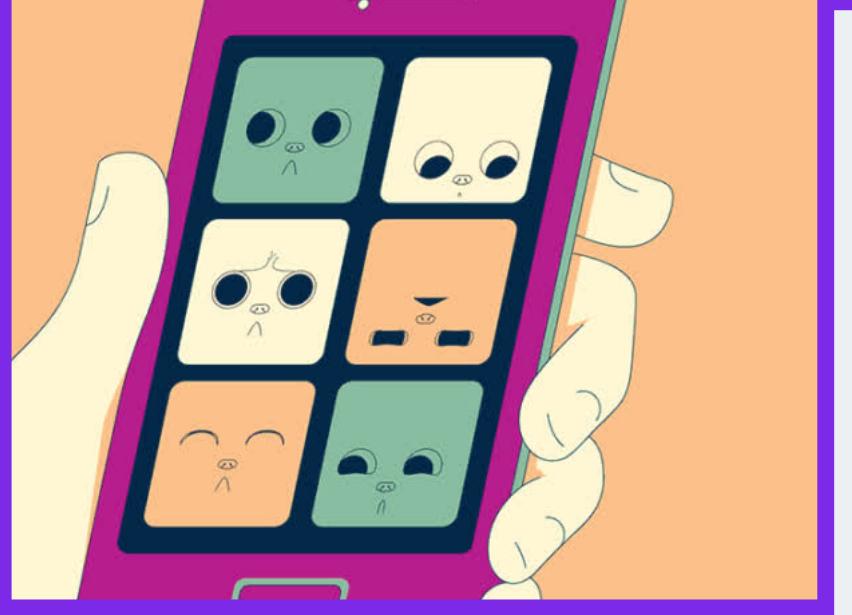
Yes!

```
<body>
  <h1>I did not change</h1>
  <script src="change-heading.js"></script>
</body>
```

Why?

Same as before! By default, the browser will execute JavaScript when it encounters it, even if that means waiting for it to load over the internet





What if the internet is really slow?



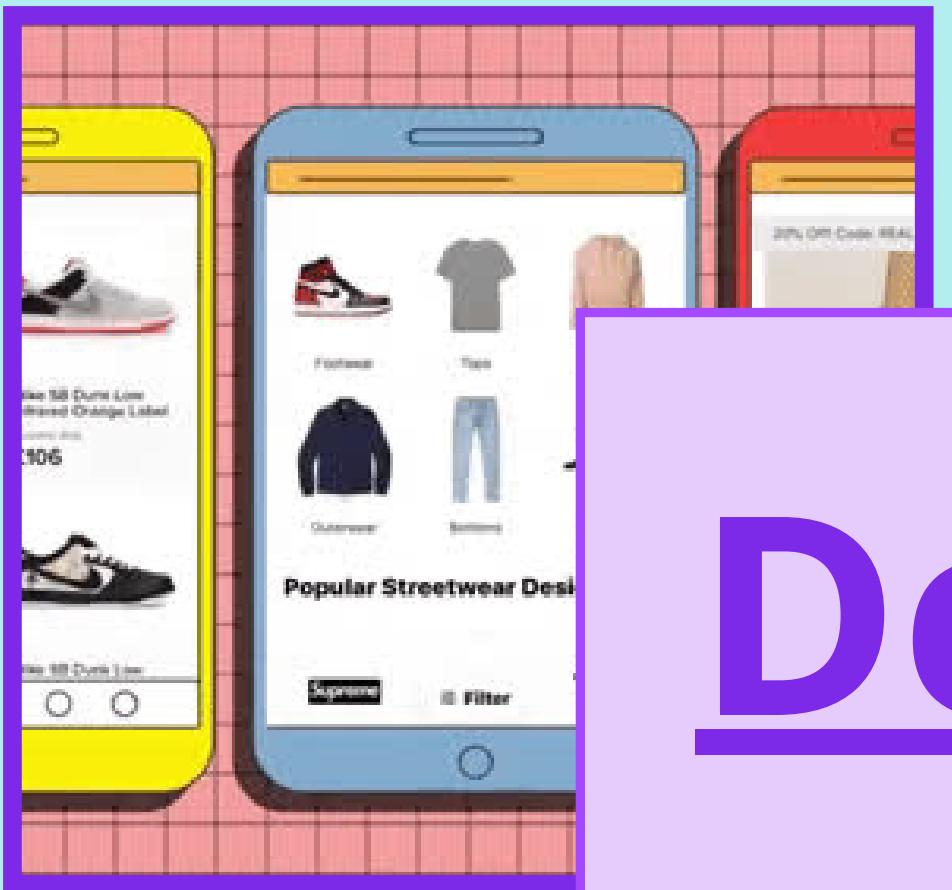
WTF is render blocking?

Before the browser can render the HTML document on your screen, it needs to *parse* the whole document. As mentioned previously, the parser will stop & execute a script whenever it encounters one.

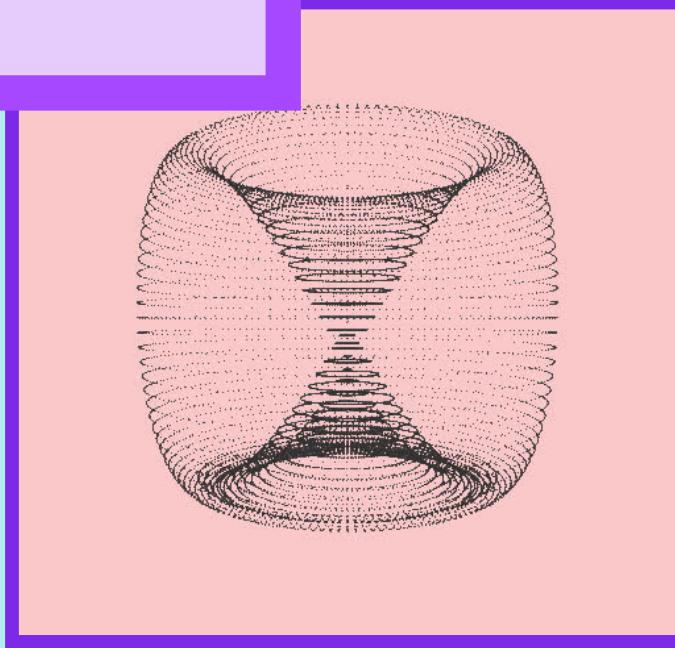
If the script is external (i.e not embedded in the document), this means that the parser will need wait for the request to complete before it can continue.

The effect of this is that sometimes you will need to wait for several seconds to see anything on the page, even once the initial request has completed! This also applies to CSS.

[Google Web Fundamentals - Adding Interactivity With JavaScript](#)



Demo!



server.js — js-in-the-browser

server.js before.html after.html load-items.js

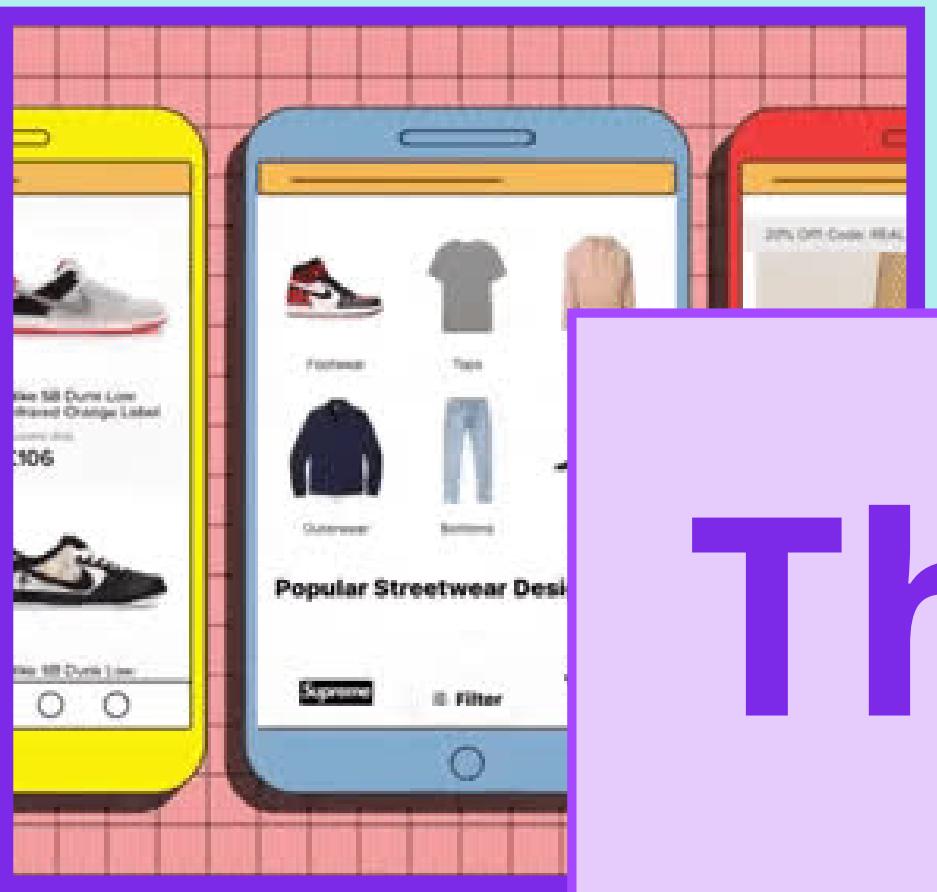
delayed > server.js > ...

```
14 const express = require("express");
13 const app = express();
12
11 app.use(delay(2000));
10
9 app.use(express.static("."));
8
7 app.listen(3000);
6
5 function delay(ms) {
4   return (req, res, next) => {
3     setTimeout(() => next(), ms);
2   };
1 }

15
```

Live Share

Ln 15, Col 1 Spaces: 2 UTF-8 LF JavaScript Prettier: ✓ ⚡ ⌂



Thanks!

