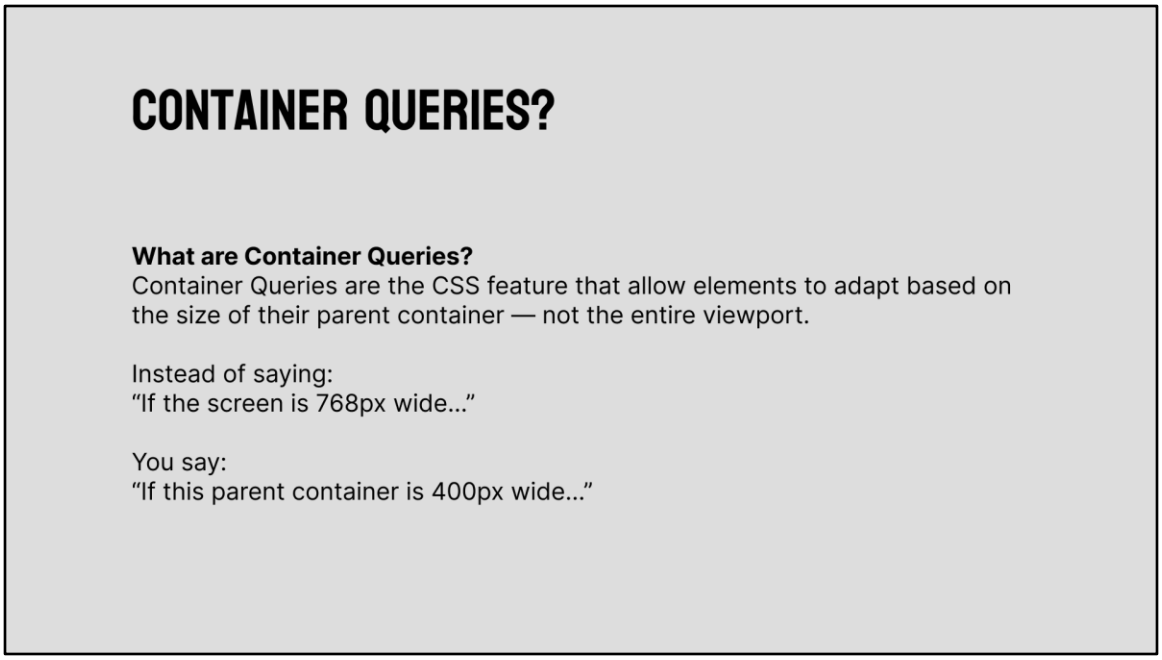
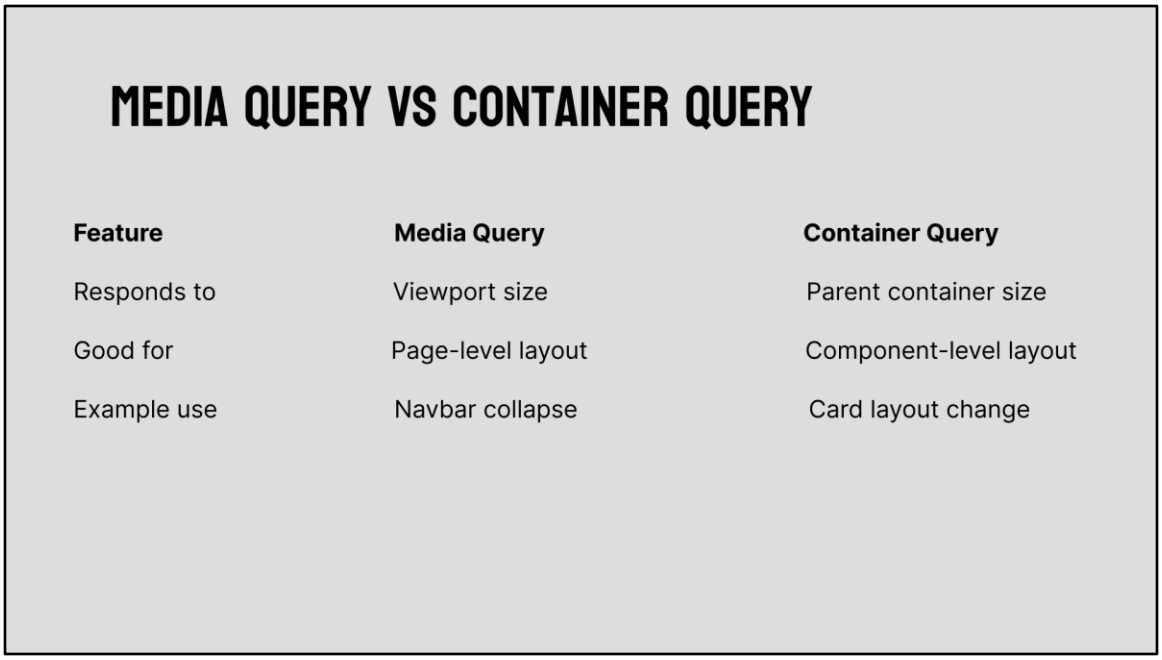
CONTAINER QUERIES?

Pravin Ale
Content Management



Container queries are a modern CSS feature that allow elements to respond to the size of their parent container instead of the entire screen. Traditionally, we used media queries that check the viewport width. But with container queries, a component can adapt based on the space it actually has available. This makes layouts more flexible and component-focused.



This table shows the key difference between media queries and container queries. Media queries respond to the viewport size, which makes them ideal for page-level layout changes, such as collapsing a navigation bar. On the other hand, container queries respond to the size of a parent container, making them better suited for component-level adjustments, like changing the layout of a card inside a grid. In simple terms, media queries control the overall page, while container queries control individual components.

Example

Media Query

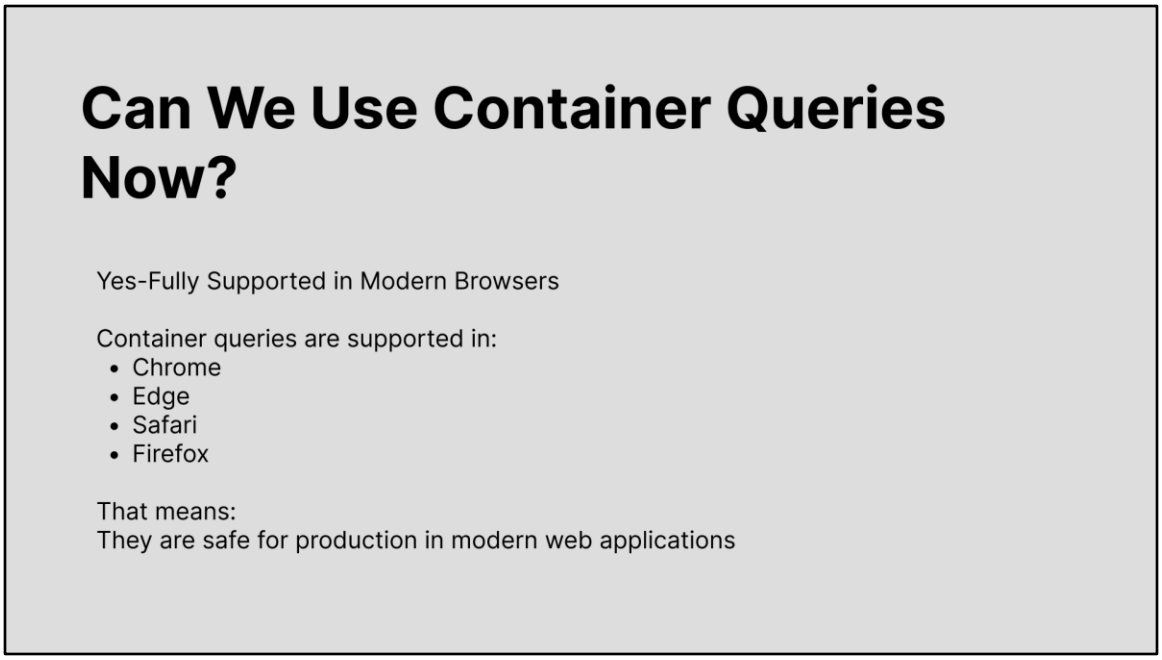
```
@media (max-width: 768px) {
  .card {
    flex-direction: column;
  }
}
```

Container Query

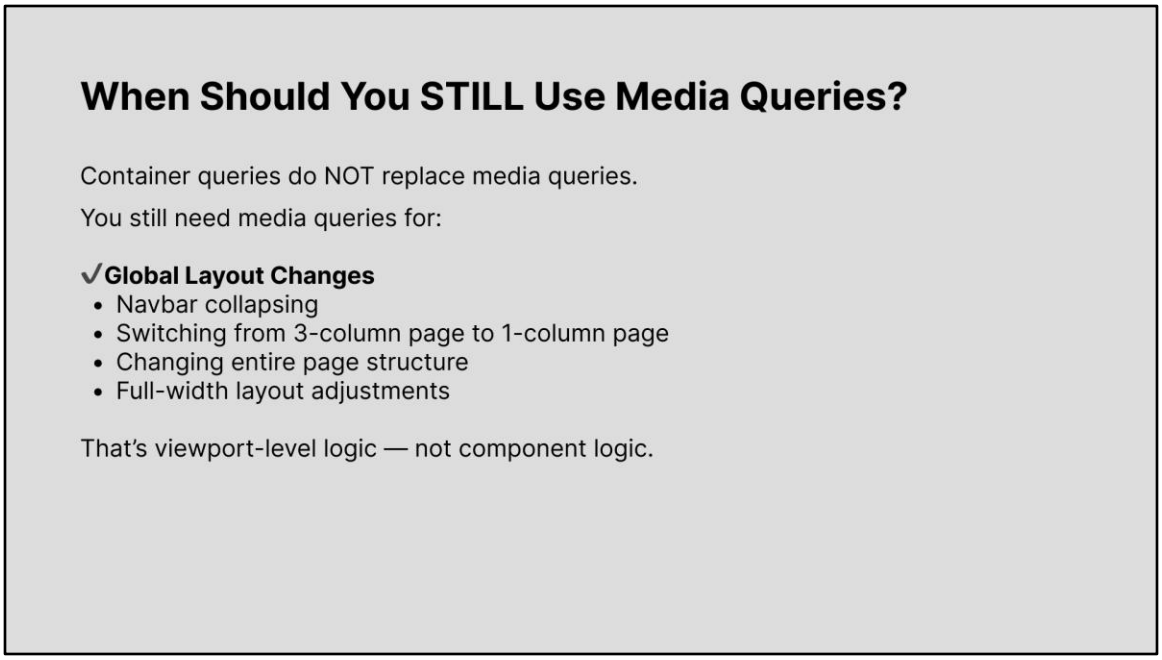
```
.card-container {
  container-type: inline-size;
}

@container (max-width: 400px) {
  .card {
    flex-direction: column;
  }
}
```

On the left, we see a media query. It checks if the screen width is 768 pixels or smaller, and then changes the card layout to a column. This decision is based on the entire viewport size.
On the right, we see a container query. First, we define a container using container-type: inline-size. Then we apply a rule that changes the card layout only if its parent container is 400 pixels or smaller.
The key difference is that the media query reacts to the screen size, while the container query reacts to the size of the parent container.



Yes, we can use container queries now. They are fully supported in all major modern browsers, including Chrome, Edge, Safari, and Firefox. Since these browsers cover the majority of users today, container queries are safe to use in real-world production applications. For older browsers, fallback strategies can still be applied, but in modern environments, they are reliable and ready to use.



Container queries do not replace media queries. We still need media queries for global layout changes that depend on the entire screen size. For example, collapsing a navigation bar, switching from a three-column layout to a single-column layout, or restructuring the whole page are viewport-level decisions. These changes affect the overall page structure, not just a single component. That's why media queries remain essential in responsive design.

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0
    <title>Document</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>

    <div class="page-layout">
        <aside class="sidebar">Sidebar</aside>
        <main class="content">Main Content</main>
        <aside class="ads">Ads</aside>
    </div>

</body>
</html>
```

Switching from 3-column page to 1-column page

```css
aside{
    border: 5px solid;
    padding: 5px;
    background-color: blue;
}

main{
    border: 5px solid;
    padding: 5px;
    background-color: red;
}

.page-layout {
  display: grid;
  grid-template-columns: 250px 1fr 250px;
  gap: 20px;
}

@media (max-width: 768px) {
  .page-layout {
    grid-template-columns: 1fr;
  }
}
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>

    <nav class="navbar">
        <div class="logo">Logo</div>
        <ul class="menu">
            <li>Home</li>
            <li>About</li>
            <li>Contact</li>
        </ul>
        <div class="hamburger">
            Menu
        </div>
    </nav>

</body>
</html>
```

Navbar collapsing

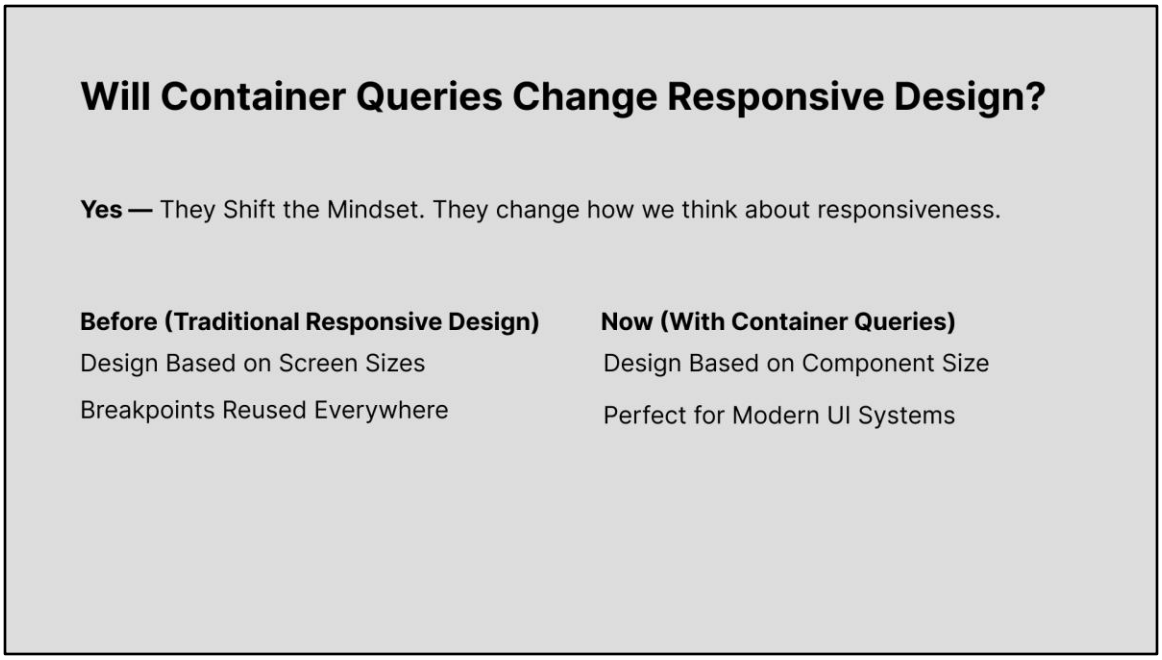
```css
.navbar {
  display: flex;
  justify-content: space-between;
  align-items: center;
}

ul li{
    list-style: none;
}

.menu {
  display: flex;
  gap: 20px;
}

.hamburger{
    display: none;
}


@media (max-width: 768px) {
  .menu {
    display: none;
  }
  .hamburger{
    display: block;
  }
}
```



Yes, container queries change the way we think about responsive design. Traditionally, we designed layouts based on screen sizes and reused the same breakpoints across the entire page. Everything depended on the viewport width. With container queries, the focus shifts to the component itself. Instead of asking how wide the screen is, we ask how much space this component has. This makes design more modular and better suited for modern UI systems built around reusable components.

THANKYOU