



CSS Media Query Pro Tips

#css #mediaquery #responsiveness

 **Chinedu Orié** Oct 23, 2019 · 3 min read

Media Query is very important in building web applications that can display well across devices of different screen sizes.

A media query is an HTML/CSS functionality that allows the content of a Web page to adapt to the type of media that the page is being rendered in, such as a computer screen or that of a phone or tablet. - [techopedia](#)

Responsive web design Approaches

There are two major approaches to implementing responsive web designs namely:

- Desktop-first approach
- Mobile-first approach

It's very important to note that each of the approaches has some rules that must be observed for it to give the expected result. Let's dive deep into each of the approaches.

Desktop-first Approach

In a desktop-first approach, we focus on building layouts that display correctly on desktops (large screens) and then scale down to smaller screen sizes (tablets, mobile phones) in descending order.

Rules for a desktop-first approach

- Use (max-width) for breakpoints
- The breakpoints must be consistently in descending order

Let's take an example of a desktop-first approach.

```
<div class="container">
  <div class="box1"></div>
  <div class="box2"></div>
  <div class="box3"></div>
  <div class="box4"></div>
</div>

.container {
  width: 100vw;
  display: flex;
  justify-content: space-between;
  flex-wrap: wrap;
}

.box1, .box2, .box3, .box4 {
  border: 1px solid black;
  height: 100px;
  min-width: 200px;
  flex-basis: 20%;
}

.box1 {
  background-color: red;
}

.box2 {
  background-color: purple;
}

.box3 {
  background-color: blue;
}

.box4 {
  background-color: green;
}

@media (max-width: 768px) {
  .box1, .box2, .box3, .box4 {
    flex-basis: 48%;
  }
}

@media (max-width: 500px) {
  .box1, .box2, .box3, .box4 {
    flex-basis: 100%;
  }
}
```

-  19
-  11
-  35
- 

**Chinedu Orié**

Software Engineer with a great passion for technology.


Follow

LOCATION
Lagos


EDUCATION
BE, Computer Engineering

JOINED
Apr 27, 2018


Trending on DEV Community 🔥



If you've tried Linux and opted against using it, tell us why for the chance to be featured on our podcast!
[#discuss](#) [#podcast](#)

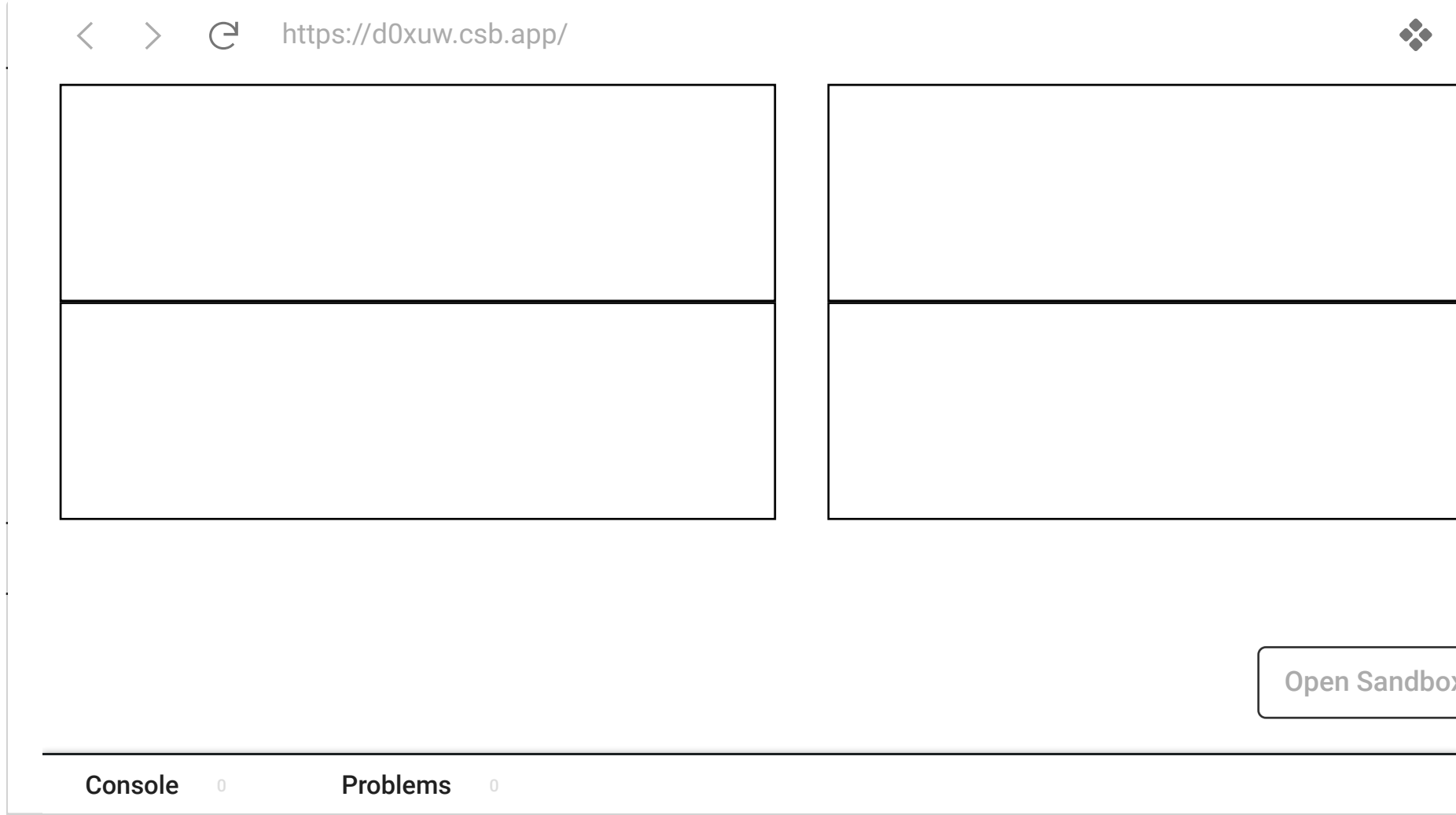


Jan 15: DEV Daily Discovery
[#discuss](#) [#bestofdev](#)



CSS interview questions and answers for Web Developers
[#css](#) [#html](#) [#career](#) [#webdev](#)

[View on codesandbox](#)



In order to simplify the explanation, we used only three screen sizes one for desktop, tablet, and mobile phone. Notice how we observed the rules, using `max-width` for breakpoints and being consistently in descending order of the breakpoint.

If you mistakenly apply mobile phone breakpoint before tablet breakpoints in a desktop-first approach, what happens is that the tablet breakpoint will override the mobile phone breakpoint.

Mobile-first Approach

In a mobile-first approach, we focus on building layouts that display correctly on mobile phones (small screens) and then scale up to larger screen sizes (tablets, desktops) in ascending order.

Rules for a mobile-first approach

- Use `(min-width)` for breakpoints
- The breakpoints must be consistently in ascending order

As an example, let's refactor the styles from the desktop-first approach as follows:

```
.container {
  width: 100vw;
  display: flex;
  flex-direction: column;
}

.box1, .box2, .box3, .box4 {
  border: 1px solid black;
  height: 100px;
  width: 100%;
}

@media (min-width: 768px) {
  .container {
    flex-direction: row;
    flex-wrap: wrap;
    justify-content: space-between;
  }
  .box1, .box2, .box3, .box4 {
    flex-basis: 48%;
  }
}

@media (min-width: 1024px) {
  .container {
    flex-direction: row;
    flex-wrap: wrap;
    justify-content: space-between;
  }
  .box1, .box2, .box3, .box4 {
    flex-basis: 20%;
  }
}
```

[View on codesandbox](#)



Again, note that the order of the breakpoints is very important in order to arrive at the expected result.

Conclusion

In this article, I tried to point out some tips that should be observed to ensure effective media queries.

Although the examples did not cover all screen sizes, the key takeaway is that there must be consistency in the order of the breakpoints for either a desktop-first approach or a mobile-first approach.

Your opinions and contributions are welcome, toss your comments in the discussion section.

Originally published on my [my blog](#)

Discussion

Subscribe



Add to the discussion



Nick Taylor (he/him) • Oct 23 '19



If I'm being honest, I always have to Google the syntax for media queries.
☺ Looking forward to your next post!



3 likes



Reply



Chinedu Orié 🇳🇮 • Oct 23 '19



Yeah, I also was a fan of that until I discovered the formula. Thank you Nick!



2 likes



Reply



Nedy Udombat • Oct 23 '19



Great tips @nedsoft !!!



3 likes



Reply



Chinedu Orié 🇳🇮 • Oct 23 '19



Thank you Nedy !!!



3 likes



Reply



Daniel Worsnup • Oct 24 '19



Do you prefer the mobile first or desktop first? From your experience does one offer significant advantages over the other?



1 like



Reply



Chinedu Orié 🇳🇮 • Oct 24 '19



TBh, I use desktop-first not because it's better but because I am used to it. The reason is that I develop with desktop, so I want the desktop to have a taste of it before other devices (pun intended). So as for me, what works for one is definitely the best provided it does not affect efficiency.



2 likes



Reply

[Code of Conduct](#) • [Report abuse](#)