

5 Ways to Create Responsive CSS Tables (Simple Examples)

By W.S. Toh / Tips & Tutorials - HTML & CSS / February 26, 2022

Welcome to a tutorial on how to create mobile-friendly responsive CSS tables. When it comes to creating responsive web pages, tables are one of those “unfortunate” things in HTML. Even though there are a ton of responsive solutions for images, videos, text, and layout – Tables are the only tricky ones to deal with.

There are a number of ways to create mobile-friendly responsive tables:

1. Simply set `width: 100%` on the table.
2. Wrap the table in a `<div style="overflow: auto">` container.
3. Hide some columns on the smaller screen sizes.
4. Wrap the cells into new rows on the smaller screen sizes.
5. Use a grid container as a table.

That covers the basics, but let us walk through more examples in this guide – Read on!

① I have included a zip file with all the example source code at the start of this tutorial, so you don't have to copy-paste everything... Or if you just want to dive straight in.

QUICK SLIDES

TABLE OF CONTENTS



Download & Notes



Responsive Tables



Useful Bits & Links



The End

SEARCH

Search ...

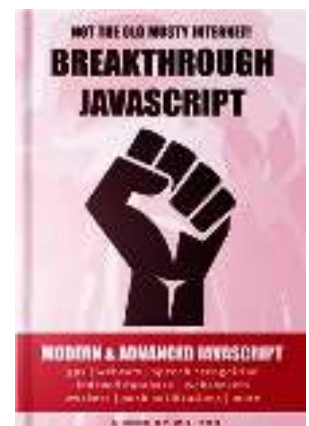


e
a
r
c
h



report this ad

BREAKTHROUGH JAVASCRIPT



Take pictures with the webcam?

Voice commands? Video calls?

Yes, it is possible with Javascript

- Check out Breakthrough

Javascript!



report this ad

FOLLOW US

[Pinterest](#)

[YouTube](#)

DOWNLOAD & NOTES



Firstly, here is the download link to the example code as promised.

QUICK NOTES

If you spot a bug, feel free to comment below. I try to answer short questions too, but it is one person versus the entire world... If you need answers urgently, please check out [my list of websites to get help with programming](#).

EXAMPLE CODE DOWNLOAD

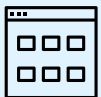
[Click here to download all the example source code](#), I have released it under the MIT license, so feel free to build on top of it or use it in your own project.

ABOUT ME



W.S. Toh is a senior web developer and SEO practitioner with over 20 years of experience. Graduated from the University of London. When not secretly being an evil tech ninja, he enjoys photography and working on DIY projects.

HOW TO CREATE RESPONSIVE TABLES



All right, let us now get into the various ways and examples of creating responsive tables.

METHOD 1) SET 100% WIDTH

1-full-width.html

```
<style>
.full { width: 100%; }
</style>

<table class="full">
  <tr>
    <th>Column 1</th>
    <th>Column 2</th>
    <th>Column 3</th>
  </tr>
  <tr>
    <td>Column 1</td>
    <td>Column 2</td>
    <td>Column 3</td>
  </tr>
```

</table>

Column 1	Column 2	Column 3
Column 1	Column 2	Column 3

This is the easiest way to create a “responsive table”. Setting `width: 100%` will make the table stretch horizontally across the container, but it suffers from a possible problem – On portrait orientation of small devices, the table is going to look extremely squished.

METHOD 2) WRAP TABLE IN OVERFLOW CONTAINER

2-overflow-wrapper.html

```
<style>
.twrap {
  width: 100%;
  overflow-x: auto;
}
</style>

<div class="twrap">
  <table>
    <tr>
      <th>Column 1</th>
      <th>Column 2</th>
      <th>Column 3</th>
      <th>Column 4</th>
      <th>Column 5</th>
    </tr>
    <tr>
      <td>Column 1</td>
      <td>Column 2</td>
      <td>Column 3</td>
      <td>Column 4</td>
      <td>Column 5</td>
    </tr>
  </table>
</div>
```

Column 1	Column 2	Column 3
Column 1	Column 2	Column 3



To prevent a squished table, we can wrap it in a `<div style="overflow: auto">` container. But of course, it comes with the price of an “ugly scrollbar” and some table columns may not be immediately visible.

METHOD 3) HIDE COLUMNS

```
3-hide-column.html

<style>
.hidder { width: 100%; }
@media screen and (max-width:768px) {
  .tohide { display: none; }
}
</style>

<table class="hidder">
  <tr>
    <th>Column 1</th>
    <th>Column 2</th>
    <th class="tohide">Column 3</th>
    <th class="tohide">Column 4</th>
    <th>Column 5</th>
  </tr>
  <tr>
    <td>Column 1</td>
    <td>Column 2</td>
    <td class="tohide">Column 3</td>
    <td class="tohide">Column 4</td>
    <td>Column 5</td>
  </tr>
</table>
```

Column 1	Column 2	Column 3	Column 4	Column 5
Column 1	Column 2	Column 3	Column 4	Column 5

Personally, I think this is a more elegant solution. When it comes to smaller screens, we use the media query `@media` to hide the columns that are not critical. The table still fits nicely on the screen and looks less cluttered. But the pain of this method is having to define which columns are important, and it is literally useless if “all columns are important and has to be shown”.

METHOD 4) WRAP CELLS INTO NEW ROWS

```
4-wrap-column.html
```

```

<style>
.gridtable { width: 100%; }
@media screen and (max-width:768px) {
  .gridtable, .gridtable thead, .gridtable tbody, .gridtable tr {
    display: grid;
    width: 100%;
  }
  .gridtable tr { grid-template-columns: auto auto auto; }
}
</style>

<table class="gridtable">
  <tr>
    <th>Column 1</th>
    <th>Column 2</th>
    <th>Column 3</th>
    <th>Column 4</th>
    <th>Column 5</th>
  </tr>
  <tr>
    <td>Column 1</td>
    <td>Column 2</td>
    <td>Column 3</td>
    <td>Column 4</td>
    <td>Column 5</td>
  </tr>
</table>

```

Column 1	Column 2	Column 3	Column 4	Column 5
Column 1	Column 2	Column 3	Column 4	Column 5

Some people may not fully agree with this method, but this is a rather genius solution. On the small screen sizes, we use the media query `@media` to turn the rows into a grid layout `.gridtable tr { display: grid }`. Thereafter, set the max number of cells per row `grid-template-columns: auto auto auto`.

The cells will automatically wrap into new rows, making things a little more legible. But this is not a perfect solution either – Tables with a lot of columns are going to look extremely confusing with a lot of extra rows.

METHOD 5) USE A GRID CONTAINER AS A TABLE

5-grid.html

```

<style>
/* (A) ON BIG SCREEN */
.thegrid {
  display: grid;
  grid-template-columns: auto auto auto auto;
  grid-gap: 5px;
}

```

```

/* (B) ON SMOL SCREEN */
@media screen and (max-width:768px) {
  .thegrid { grid-template-columns: auto auto; }
}
</style>

<div class="thegrid">
  <!-- HEAD -->
  <div class="head">Column 1</div>
  <div class="head">Column 2</div>
  <div class="head">Column 3</div>
  <div class="head">Column 4</div>

  <!-- ROWS -->
  <div class="cell">Column 1</div>
  <div class="cell">Column 2</div>
  <div class="cell">Column 3</div>
  <div class="cell">Column 4</div>
</div>

```

Column 1

Column 2

Column 3

Column 4

Column 1

Column 2

Column 3

Column 4

The modern-day `display: grid` – Put in a little bit of creative juice, and we can use it to create lists and tables. Yes, this final example is not using the “traditional” `<table>`, but a `<div style="display: grid">`.

USEFUL BITS & LINKS



That’s all for the tutorial, and here is a small section on some extras and links that may be useful to you.

LINKS & REFERENCES

- [Using media queries](#) – MDN
- [5 Practical Solutions to Make Responsive Data Tables](#) – Michał Jarosz

INFOGRAPHIC CHEAT SHEET

Responsive CSS Table

(Click To Enlarge)

THE END



Thank you for reading, and we have come to the end of this chapter on responsive tables. I hope that it has helped you to improve your projects, and if you have anything to add to this guide, please feel free to comment below. Good luck and happy coding!

[← Previous Post](#)

[Next Post →](#)

Leave a Comment

Your email address will not be published. Required fields are marked *

Type here..

Name* Email*

[Post Comment »](#)



[report this ad](#)

Copyright © 2022 Code Boxx. All rights reserved.

Code Boxx participates in the eBay Partner Network, an affiliate program designed for sites to earn commission fees by linking to ebay.com. We also participate in affiliate programs with Bluehost, ShareASale, Clickbank, and other sites. We are compensated for referring traffic.

[About Us](#) [Contact Us](#) [Affiliate Disclosure](#) [Terms of Use](#) [Privacy Policy](#) [Credits](#)

Tracker

DoubleClick

Google Analytics

Google Tag Manager

