

Bootstrap kolommen

Hoe rekent Bootstrap met kolommen?

Klasse (Bootstrap 5)	Schermbreedte (Pixels)	Mensentaal: Welke apparaten horen hier doorgaans bij?
		Alle normale telefoons (rechtop / portrait) [index.html].
<code>.col-</code> (<i>geen letters</i>)	Minder dan 576px [index.html]	• <i>Voorbeelden:</i> iPhone (alle reguliere modellen), Samsung Galaxy, Google Pixel. De content staat hier vrijwel altijd in 1 strakke kolom onder elkaar [index.html].
<code>.col-sm-</code>	Vanaf 576px tot 768px [index.html]	Grote telefoons (liggend / landscape) en kleine mini-tablets [index.html]. • <i>Voorbeelden:</i> iPhone Pro Max (liggend), iPad Mini (rechtop), oudere kleine e-readers en budget-tablets.
<code>.col-md-</code>	Vanaf 768px tot 992px [index.html]	Normale tablets (rechtop / portrait) [index.html]. • <i>Voorbeelden:</i> De standaard iPad (10.2 / 10.9 inch rechtop), Samsung Galaxy Tab (rechtop). Dit is het klassieke kantelpunt waarop een site vaak van mobiel menu naar desktop-menu switcht.
<code>.col-lg-</code>	Vanaf 992px tot 1200px [index.html]	Grote tablets (liggend) en kleine laptops / netbooks [index.html]. • <i>Voorbeelden:</i> iPad Air & Pro (liggend), de klassieke iPad (liggend), kleinere MacBooks (12/13 inch), Microsoft Surface.
<code>.col-xl-</code>	Vanaf 1200px tot 1400px [index.html]	Normale laptops en standaard computerschermen [index.html]. • <i>Voorbeelden:</i> MacBook Pro 14/16 inch, standaard kantoormonitoren (Full HD resoluties in browser-venster), grotere desktops. Hier wil je vaak je 3- of 4-koloms grids voluit laten glanzen.

Klasse (Bootstrap 5)	Schermbreedte (Pixels)	Mensentaal: Welke apparaten horen hier doorgaans bij?
.col-xxl-	Vanaf 1400px en groter [index.html]	Grote desktops, iMacs en brede computermontoren [index.html]. • <i>Voorbeelden:</i> 24/27 inch iMacs, breedbeeld (Ultrawide) montoren. Hier grijpen de Bootstrap-containers in om te zorgen dat je tekstregels niet pijnlijk lang worden uitgerekt [index.html].

hoe werkt het?

Elk beeldscherm wordt verdeeld in twaalf kolommen: telefoon, tablet, laptop etc. Als je niet specificeert voor welk apparaat je codeert en zegt dat de classname .col-12 is, dan zal elk scherm proberen de inhoud van het element over de 12 kolommen te verdelen, ook de telefoon.

Als je geen apparaat en geen breedte specificeert en je hebt één kolom en je schrijft col (dus: 'deze div is een col zonder specificatie'), dan wordt dit gelezen als col-12, dus verdeling over de hele breedte op elk apparaat. Die ene div met col komt in alle schermen over de hele breedte. Zijn er twee elementen met col in een rij dan verdelen die twee cols de rij en dan krijgt ieder element dus de helft (col-6).

Bij één enkel element in een rij met specificatie col-4 wordt de inhoud over 4 van de 12 kolommen verdeeld, ook op een telefoon. De rij begint links. Als je twee div's naast elkaar zet zonder te specificeren voor welk scherm, zal het er op ieder scherm op dezelfde manier verdeeld uitzien. Dus twee div's met de eerst col-4 en de tweede col-8 geven op ieder apparaat een verdeling over de breedte van 4/12 en 8/12.

Als je verder gaat specificeren naar apparaat, bijvoorbeeld één kolom col-sm-4, dan zal de tablet (en alles breder dan dat) de inhoud over de eerste 4 kolommen verdelen. De 8 kolommen ernaast blijven leeg en de volgende row komt eronder. Op een telefoon (die geen sm-apparaat is), zal de col-sm-4 worden gelezen als col en dan zal dit gespecificeerde element worden uitgesmeerd over de 12 kolommen.

Twee kolommen met specificatie naar apparaat

Als er twee kolommen zijn, de eerste col-sm-4 en de tweede col-sm-8, krijg je op een tablet en alles wat breder is een verdeling van 4/12 en 8/12 over de breedte van het scherm. Een telefoon kan de sm-regel niet toepassen en leest geen verdere specificatie voor de telefoon en valt terug op de basis. De telefoon maakt van de twee kolommen ieder col-12 (mobile first) en zet de twee kolommen onder elkaar.

Verder differentiëren per apparaat

Een voorbeeld om op verschillende apparaten een andere weergave te krijgen:

```
<div class="col-sm-3 col-md-6">....</div>
<div class="col-sm-9 col-md-6">....</div>
```

Op een telefoon zullen de div's ieder een volle rij pakken, iets grotere schermen laten 3/12 en 9/12 zien en nog grotere apparaten half-half.

Heb je drie kolommen, de eerste wel gespecificeerd en de twee andere niet: col-sm-4, col en col, dan zal de telefoon de col-sm-4 over de breedte laten zien en de andere twee op de regel eronder en dan ieder voor de helft van de breedte.

Alleen het apparaat specificeren en niet de breedte per kolom

Hieronder een voorbeeld. Als je wel zegt voor welk apparaat maar niet de breedte per kolom dan zullen de div's verdeeld worden over de breedte van de row, maar in een smaller scherm worden de div's op elkaar gestapeld.

```
<div class="row">
  <div class="col-md">1 of 4</div>
  <div class="col-md">2 of 4</div>
  <div class="col-md">3 of 4</div>
  <div class="col-md">4 of 4</div>
</div>
```

Je hebt col wel nodig

```
<div class="container-fluid">
  <div class="row">
    <div class="col">1 uit 4</div>
    <div class="col">2 uit 4</div>
    <div class="col">3 uit 4</div>
    <div class="col">4 uit 4</div>
  </div>
</div>
```

</div>

Als je in bootstrap wel 'row' aangeeft in een div en je zet drie andere div's, als <div class="col"> binnen die div, dan komen die drie binnenste div's in iedere browser naast elkaar door de row, want de row betekent: display: flex !important;. Zie hierboven. Normaal zouden die drie div's onder elkaar komen; zie voorbeeld hieronder.

```
<div class="container-fluid">
  <div class="row">
    <div>1 uit 4</div>
    <div>2 uit 4</div>
    <div>3 uit 4</div>
    <div>4 uit 4</div>
  </div>
</div>
</div>
```

- [Login](#) om te reageren