!!Contenido Importante!!
03/02/2020 ,18:28:55

Completado
05/02/2020 ,20:13:07

CSS Grid Layout

C	S	S	
G	R	I	D

CODE VIEW

RESPONSIVE LAYOUT WITHOUT MEDIA QUERIES

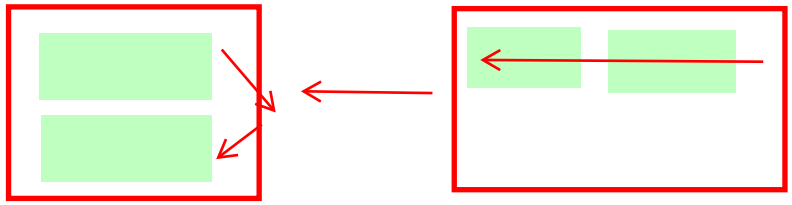
One of the most powerful features in the CSS Grid specification is the ability to create responsive layouts without using media queries. This is done by using the repeat function (covered earlier) along with auto-placement keywords auto-fit or auto-fill. These keywords allow you to place as many grid items of a particular size on a row that will fit within the width of the viewport.

Using the example below, make your web browser narrower and wider to see how using both auto-fit and auto-fill make the page behave.

EXAMPLE grid-layout.com/no-queries

NOTE: *Up to a certain width, both auto-fit and auto-fill will have the same result. The difference between auto-fit and auto-fill can only be seen when there are fewer grid items than could fit on a row. In this example, each grid item is a minimum of 200 pixels and a maximum of 1fr.*

200px - 1fr | 200px - 1fr



En esta viewport particular (1.215 píxeles), un total de seis elementos de cuadrícula de 200 píxeles de ancho podrían caber en la fila.

auto-fit - columas 5 y 6 son creadas actualmente pero ellos tienes
0 pixeles y estan apiladas una encima de la otra
Puedes fijarte que los numeros de las columnas 5 , 6 y 7 arriba a la derecha

RESUMEN : Las filas son creadas pero si no se rellena se pliegan sobre si mismas y no se ven y el resto de elementos se estiran para llenar el vacio

- SI EL ESPACIO ES MAYOR QUE LA ANCHURA DE LAS COLUMNAS SE EXPANDE PARA COGER TODO EL ANCHO
- SI SE REDUCE EL ANCHO DEL NAVEGADOR SE REDUCE HASTA DONDE ESTA DEFINIDO Y SALTA LA FILA SIGUIENTE PARA AJUSTARSE

fijarte/notar/observar

You'll notice several things.

- When making the viewport narrower, both auto-fit and auto-fill automatically push grid items down to another row once each of the grid items would become narrower than 200 pixels wide.
  mas estrecho — empujan — abajo — otra fila cada vez — los elementos de la cuadrícula se convertirían mas estrechos
- When making the viewport wider, auto-fit makes the grid-items stretch to fill out the row.
  ampliado — estiren para rellenar la fila — cuando hacemos el ancho viewport con auto-fit hace grid-item estire hasta llenar la fila
- When making the viewport wider, auto-fill puts in empty columns on the row.
  pone dentro la columna vacia sobre la fila — cuando hacemos el ancho viewport auto-fill pone dentro de las columnas vacias sobre la filas

Fijate/Observa

Notice the column line numbers in the illustration below. At this particular viewport width (1,215 pixels), a total of six 200 pixel wide grid items could fit on the row. Because there are only four 200 pixel wide grid items on the row, CSS Grid creates two more columns. However, the way auto-fit and auto-fill handle those extra columns differs.

ancho — podrian caber — fila — hay solo unos — amplio

Si se crean 2 grid-items mas se añadiran debajo dentro de otra fila

With auto-fit, columns 5 and 6 are actually created, but they are 0 pixels wide and are stacked one on top of the other. You can see this by noting the column line numbers 5, 6, and 7 at the

estan apiladas — uno encima del otro — por fijarte

auto-fill - las columnas 5 y 6 son creadas pero son creadas con un minimo de 200px dentro y se muestra la celda blanca y vacia hasta el final de la fila

hay actualmente un pequeño codigo que necesitas hacer todo esto pase y ello hacer con el grid-template-columns propiedad con grid-container

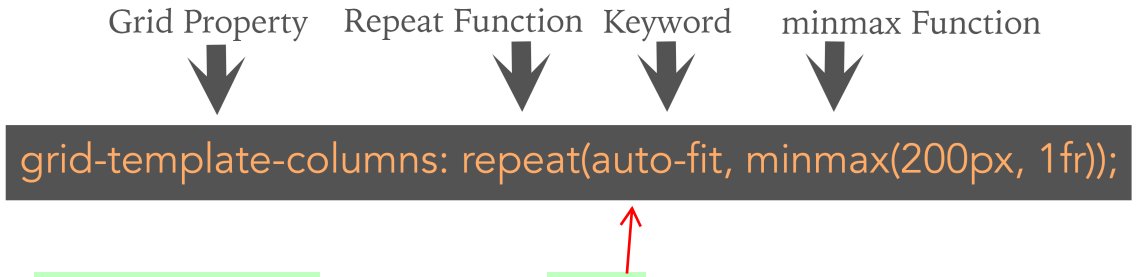
aunque sin embargo

top right. So, even though the cells are actually created, ellas no se ven they don't show up.

With auto-fill, columns 5 and 6 are also created, but now they are are created with a 200 pixel minimum width and show up as blank cells at the end of the row.

mostraran como una tabla de celdas hasta el final de la fila

There is actually very little code needed to make all this happen and it's all done with the grid-template-columns property in the grid container.



This grid-template-columns property uses the auto-fit keyword and sets the minimum width for each grid item to 200 pixels and a maximum width of 1fr (or, one fractional unit).

```
<style>
.container1 {
    border: 1px solid black;
    display: grid;
    grid-gap: 5px;
    grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
    grid-template-rows: repeat(2, 200px);
    }
.container2 {
    border: 1px solid black;
    display: grid;
    grid-gap: 5px;
    grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));
    grid-template-rows: repeat(2, 200px);
    }
```

CSS

Add the CSS seen here to create two grids. The first one is a grid container that uses the auto-fit keyword. The second one is a grid container that uses the auto-fill keyword. The "GENERAL STYLES" section will be used simply to make pretty the individual grid items.

HTML

The HTML for the first grid is contained inside a grid container with a class called container1. We're using this area to show how auto-fit works.

The HTML for the second grid is contained inside a grid container with a class called container2. We're using this area to show how auto-fill works.

ADVANCED EXAMPLE

The link below shows you a page where two areas change layout without media queries. Pay attention to the text in the header and the "cards" showing the different types of grapes. This

uvas

```
/* GENERAL STYLES */
.container1 > div {
    background-color: coral;
    display: flex;
    justify-content: center;
    align-items: center;
    font-size: 48px;
    font-family: arial,helvetica,sans-serif;
    }
.container2 > div {
    background-color: aqua;
    display: flex;
    justify-content: center;
    align-items: center;
    font-size: 48px;
    font-family: arial,helvetica,sans-serif;
    }
</style>
<body>
<h2>auto-fit</h2>
<div class="container1">
    <div>1</div>
    <div>2</div>
    <div>3</div>
    <div>4</div>
</div>
<h2>auto-fill</h2>
<div class="container2">
    <div>1</div>
    <div>2</div>
```

Todos los elementos Grid-Item se van a estirar hasta rellenar el ancho de la fila ya que solo hay definido 4 Grid-item y el Grid-Container es mucho mayor para las dimensiones de los Grid-items que estan definidos (200px , 1fr) PERO SI SE ENCOGEN LOS ELEMENTO como maximo minimo sera de 200px

estira todo hasta rellenar toda la fila

illustration uses the auto-fit keyword for both areas. It also uses "auto" instead of "1fr" in the grid-template-columns property. This way, the layout will center and extra space will be equal on both sides of the layout. Also, the cards don't expand because they are set to a width of 300 pixels.

en lugar de

el espacio extra va estar equilibrado ambos lados del layout

Check out the HTML source of the page to learn how it was done. The main element is the grid container for the top of the page. The section element is the grid container for the "cards."

grid-layout.com/no-queries-example.html

```
    <div>3</div>
    <div>4</div>
</div>
</body>
```


Steven D. Anderson, Ph.D.
James Madison University
School of Media Arts & Design
anderssd@jmu.edu