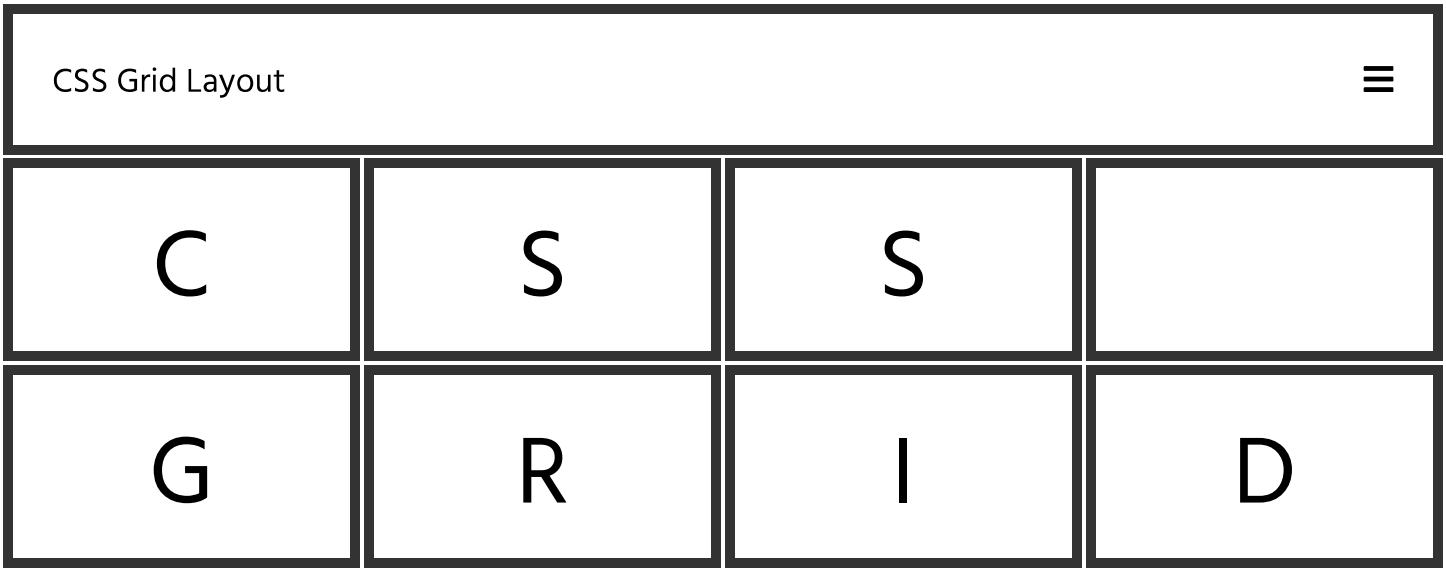
Completado
06/02/2020 ,19:04:51



CODE VIEW

```
<style>
.container > div {
    background: #eee;
    padding: 1em;
    }
.container > div:nth-child(odd) {
    background: #ddd;
    }
.container {
    display: grid;
    grid-template-columns: repeat(2, 1fr 2fr);
    grid-auto-rows: minmax(120px, auto);
    grid-gap: 1em;
    }
</style>
<body>
<div class="container">
    <div>Grid item one. Grid item one. Grid item one. Grid item one. Grid item one. Grid item one. Grid item one.</div>
    <div>Grid item two.</div>
    <div>Grid item three.</div>
    <div>Grid item four.</div>
    <div>Grid item five.</div>
    <div>Grid item six.</div>
    <div>Grid item seven.</div>
    <div>Grid item eight.</div>
</div>
</body>
```

NESTED GRIDS

GETTING STARTED

Starting CSS

← Let's begin by adding the CSS we utilized in the last lesson. We're shading the grid items in alternating shades of gray and creating a grid container that is repeating the pattern 1fr 2fr twice on each row. Use the CSS code to the left to take care of that.

sombreado

entre cada fila

la izquierda para encargarse de eso

Starting HTML

← Next, we will utilize eight boxes (grid items) for our layout. Use the HTML code to the left.

Below is what you get in a browser.

GIRAR/ROTAR

TURN A GRID ITEM INTO A GRID CONTAINER

Cuando anidamos un grid dentro de otro grid

En realidad tu estas solo convirtiendo un grid item en un grid-contenedor

When nesting a grid inside another grid, you're actually just making a **grid item** into a **grid container**. In this case, we'll take the fourth grid item and turn it into a grid container. Then, we'll nest three new grid items inside that container.

Comenzaremos creando una clase llamada anidada que podamos aplicar al cuarto elemento de la cuadrícula. La clase anidada se usará para convertir un elemento de cuadrícula actual en un contenedor de cuadrícula. Agregue el CSS a la izquierda para crear la clase. Observe que grid-template-columnas: repeat (3, 1fr)

```
.nested {
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    grid-auto-rows: minmax(70px, auto);
    grid-gap: 1em;
    }
.nested > div {
    border: #000 1px solid;
    padding: 1em;
    }
<div class="container">
    <div>Grid item one. Grid item one. Grid item one. Grid item one. Grid item one. Grid item one. Grid item one. Grid item one.</div>
    <div>Grid item two.</div>
    <div>Grid item three.</div>
    <div class="nested">
        <div>Nested grid box 1.</div>
        <div>Nested grid box 2.</div>
        <div>Nested grid box 3.</div>
    </div>
    <div>Grid item five.</div>
    <div>Grid item six.</div>
    <div>Grid item seven.</div>
    <div>Grid item eight.</div>
```

We'll begin by making a class called nested that we can apply to the fourth grid item. The nested class will be used to turn a current grid item into a grid container. Add the CSS to the left to create the class. Notice that grid-template-columns: repeat(3, 1fr) indicates that it is a three column nested grid with each column set to use one third of the available space.

indica que es una cuadrícula anidada de tres columnas con cada columna configurada para usar un tercio del espacio disponible.

NOTE: *We'll also add a border and padding to the nested grid items to better see how they are displayed.*

Add the nested class to the fourth grid item of the parent grid, thus turning it into a grid container. Then, put three new elements (divs) inside that container.

convirtiéndolo así en un contenedor de grid

Below is what you get in a browser.

NOTE: *Nested grids are soon expected to give way to a new level of the CSS spec called "subgrid." Here's more on this in Rachel Andrew's article called [CSS Grid Level 2: Here Comes Subgrid](#).*

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