

M1C1 CSS Grid Assignment

During the first section of Module 1, you will have reviewed the very helpful CSS Grid tool. CSS Grid is a tool that developers can use to control the layout and behavior of their HTML elements. You were first exposed to CSS Grid during the Coding Foundations course. Now that you have reviewed and learned more about CSS Grid during this first Full Stack section, it is time to practice! You will use HTML and CSS to complete this assignment. It is expected that you will create and edit files using a text editor and that you will use an external stylesheet. Please complete the following assignment and reach out on the Support App to have a mentor review your work. If you have any questions or need any help, please reach out so we can help you! This assignment must be completed to pass this section of the coursework.

*Concepts Included in this Assignment: HTML, CSS, CSS Grid, HTML Classes, HTML IDs, HTML Div, Img, and Paragraph Tags, CSS Background-Color, CSS Text Color, CSS Grid Rows, Columns, and Fractional Units, CSS External Stylesheets and Inline Styling, Google Fonts, and Font Awesome.

Grid 1: Create a div tag with the class of “gridOne”. Create two paragraph tags that respectively say “Hello World” and “My Fantastic Content”. This grid should have two equally sized columns. Use inline styling to turn the text of the second paragraph tag blue.

Example:

Hello World

My Fantastic Content

Grid 2: Create another div tag in the same HTML file. Make sure that none of the grids are nested inside each other. Give this div the class of “gridTwo”. Create three divs inside of gridTwo, this means that they will be child divs in relation to gridTwo. Assign them each a unique class. Add a paragraph tag to each of the divs with whatever text you would like. Each child div should have its own different background color. Try using red, blue, and yellow. This grid should have three columns. The third column should be twice as large as the first two columns. The background colors will help you identify if the sizes created by your grid-template-columns are correct.

Example:

whatever text I want

Grid 3: Create another div tag in the same HTML file. Give this div the class of “gridThree”. Create two child divs inside of gridThree and give them both a unique class. The first child div should contain a paragraph tag and the second should contain an image of your choice. This grid should have two rows. The second row should be three times as large as the first. Use

background colors again to help make the sizes clear. Give gridThree a width of 500 pixels and a height of 300 pixels.

**Make sure that you give the image a width of 500 pixels and a height of 225 pixels. You will also need to make sure that object-fit equals cover so that the image will not be distorted or too large for the row.*

Example:

The text I chose is Awesome



Grid 4: Create another div tag in the same HTML file. Give this div the class of “gridFour”. Create two child divs and give the first a unique class name and the second an ID. The first div should contain a paragraph tag. The second should contain two paragraph tags. GridFour should have two equally sized rows. Now we will deal with a nested grid. Select the second child div by its ID. Use CSS Grid to create another grid in that div. There should be two grid columns of equal size. Give gridFour a width of 500 pixels and a background color of green.

Example:

My Even More Awesome Text

My left text

My right text

Grid 5: Create another grid following the same instructions as in Grid 4, with the exception of the second child div. Give this div the class of “gridFive”. The second child div should have two images instead of two paragraph tags. The first picture should be twice as wide as the second. You will also have to resize these images so that they fit, try making each 200 pixels tall and wide. Make sure to use object-fit: cover; again.

Example:

My Even More Awesome Text



Grid 6: Create another div tag in the same HTML file. Give this div the class of “gridSix”. There should be two elements inside of gridSix. The first should be a font-awesome icon of your choice. The second element should be a paragraph tag. Choose a google font and apply it to the paragraph tag. GridSix should have two grid-template-columns. The second column, with the paragraph, should be three times the size of the first column with the icon. Give gridSix a width of 500px and make the background color light gray.

Example:



MY AWESOME GOOGLE FONT IS APPLIED HERE

*Images used come from Unsplash and are free to use.

Ice <https://unsplash.com/photos/hSPVuakrJqs>

Lantern <https://unsplash.com/photos/qC117ahi-mE>

Street https://unsplash.com/photos/_Tul8tZHlk4

Preguntas Teóricas:

1.- ¿Cuáles son algunas de las cosas que hacen que SCSS sea diferente de CSS? (ponme ejemplos)

El desarrollo del lenguaje SCSS hace que la programación en CSS sea más fácil y sencilla. Se me ocurren dos ejemplos que Jordan ha explicado en los videotutoriales: la adaptación a las pantallas (responsive) y la ejecución de comandos en escala (permite manejar los datos de forma escalable, es decir, desde un código sencillo se puede desarrollar una aplicación que sea capaz de gestionar datos a mayor escala).

2.- ¿Qué es una variable SCSS? (porque crees que debes utilizarla pon un ejemplo de una variable, escribe una variable y como se pondría para utilizarla)

Una variable es un código que se prepara para poder ser utilizado de forma continuada a lo largo del trabajo de programación que se está desarrollando. Las variables SCSS comienzan con el símbolo de dólar (\$) . Un ejemplo de variable SCSS puede ser el siguiente: `&color-text: #999999` . A la hora de utilizarlo, con colocar en la opción "color:" "color-text", de manera automática se aplicará la opción asignada, o sea, `#999999` .

3.- ¿Qué es un SCSS Mixin? (porque crees que debes utilizarla pon un ejemplo de una variable, escribe una variable y como se pondría para utilizarla)

Mixin en SCSS es un código agrupado que puede ser utilizado de manera fácil y recurrente para realizar los trabajos de programación más fácil y rápidamente, agrupando dos o más variables, según la necesidad. Por ejemplo, siguiendo el ejemplo de la variable anterior, si además de un color de texto determinado, queremos que el tipo de fuente sea uno determinado, y de un tamaño concreto, podremos escribir lo siguiente: `@mixin text-select { color: #999999 font: Ubuntu size: 1em`

Esto nos servirá más adelante en la programación para que colocando el código "`@include text-select`" en alguna de las opciones de texto que estemos diseñando, serán aplicadas esas condiciones de forma directa, siguiendo esas tres condiciones prefijadas.

4.- ¿Qué significa Unidad fraccionaria (fr) con CSS Grid?

La unidad fraccionario en CSS Grid nos permite distribuir el espacio longitudinal entre los elementos que hayamos dispuesto en esa área.

Asignación de cuadrícula CSS M1C1

Durante la primera sección del Módulo 1, habrás revisado la muy útil herramienta CSS Grid. CSS Grid es una herramienta que los desarrolladores pueden utilizar para controlar el diseño y el comportamiento de sus elementos HTML. Estuvo expuesto por primera vez a CSS Grid durante el curso Fundamentos de codificación. Ahora que has revisado y aprendido más sobre CSS Grid durante esta primera sección Full Stack, ¡es hora de practicar! Utilizará HTML y CSS para completar esta tarea. Se espera que cree y edite archivos usando un editor de texto y que use una hoja de estilo externa. Complete la siguiente tarea y comuníquese con la aplicación de soporte para que un mentor revise su trabajo. Si tiene alguna pregunta o necesita ayuda, comuníquese con nosotros para que podamos ayudarlo. Esta tarea debe completarse para aprobar esta sección del trabajo del curso.

*Conceptos incluidos en esta tarea: HTML, CSS, cuadrícula CSS, clases HTML, ID de HTML, etiquetas HTML Div, Img y párrafo, color de fondo CSS, color de texto CSS, filas, columnas y unidades fraccionarias de cuadrícula CSS, CSS externo Hojas de estilo y estilos en línea, Google Fonts y Font Awesome.

Cuadrícula 1: crea una etiqueta div con la clase "gridOne". Cree dos etiquetas de párrafo que digan respectivamente "Hola mundo" y "Mi contenido fantástico". Esta cuadrícula debe tener dos columnas del mismo tamaño. Utilice un estilo en línea para convertir el texto de la etiqueta del segundo párrafo en azul.

Ejemplo:

Cuadrícula 2: cree otra etiqueta div en el mismo archivo HTML. Asegúrese de que ninguna de las rejillas esté encajada una dentro de la otra. Dale a este div la clase de "gridTwo". Cree tres divs dentro de gridTwo, esto significa que serán divs secundarios en relación con gridTwo. Asígnales a cada uno una clase única. Agrega una etiqueta de párrafo a cada uno de los divs con el texto que desees. Cada div secundario debe tener su propio color de fondo diferente. Intente usar rojo, azul y amarillo. Esta cuadrícula debe tener tres columnas. La tercera columna debe ser dos veces más grande que las dos primeras columnas. Los colores de fondo le ayudarán a identificar si los tamaños creados por las columnas de su plantilla de cuadrícula son correctos.

Ejemplo:

Cuadrícula 3: cree otra etiqueta div en el mismo archivo HTML. Dale a este div la clase de "gridThree". Crea dos divs secundarios dentro de gridThree y dale a ambos una clase única. El primer div secundario debe contener una etiqueta de párrafo y el segundo debe contener una imagen de su elección. Esta cuadrícula debe tener dos filas. La segunda fila debe ser tres

veces más grande que la primera. Utilice colores de fondo nuevamente para ayudar a aclarar los tamaños. Asigne a gridThree un ancho de 500 píxeles y un alto de 300 píxeles.

*Asegúrate de darle a la imagen un ancho de 500 píxeles y un alto de 225 píxeles. También deberá asegurarse de que el ajuste del objeto sea igual a la cobertura para que la imagen no se distorsione ni sea demasiado grande para la fila.

Ejemplo:

Cuadrícula 4: cree otra etiqueta div en el mismo archivo HTML. Dale a este div la clase de "gridFour". Cree dos divs secundarios y asigne al primero un nombre de clase único y al segundo una identificación. El primer div debe contener una etiqueta de párrafo. El segundo debe contener dos etiquetas de párrafo. GridFour debe tener dos filas del mismo tamaño. Ahora nos ocuparemos de una cuadrícula anidada. Seleccione el segundo div secundario por su ID. Utilice CSS Grid para crear otra cuadrícula en ese div. Debe haber dos columnas de cuadrícula del mismo tamaño. Dale a gridFour un ancho de 500 píxeles y un color de fondo verde.

Ejemplo:

Cuadrícula 5: cree otra cuadrícula siguiendo las mismas instrucciones que en la Cuadrícula 4, con la excepción del segundo div secundario. Dale a este div la clase de "gridFive". El segundo div secundario debe tener dos imágenes en lugar de dos etiquetas de párrafo. La primera imagen debe tener el doble de ancho que la segunda. También tendrás que cambiar el tamaño de estas imágenes para que quepan, intenta hacer que cada una tenga 200 píxeles de alto y ancho. Asegúrese de utilizar object-fit: cover; de nuevo.

Ejemplo:

Cuadrícula 6: cree otra etiqueta div en el mismo archivo HTML. Dale a este div la clase de "gridSix". Debería haber dos elementos dentro de gridSix. El primero debe ser un ícono con una fuente increíble de tu elección. El segundo elemento debe ser una etiqueta de párrafo. Elija una fuente de Google y aplíquela a la etiqueta del párrafo. GridSix debería tener dos columnas de plantilla de cuadrícula. La segunda columna, con el párrafo, debe tener tres veces el tamaño de la primera columna con el icono. Dale a gridSix un ancho de 500 px y haz que el color de fondo sea gris claro.

Ejemplo:

*Las imágenes utilizadas provienen de Unsplash y son de uso gratuito.

Hielo <https://unsplash.com/photos/hSPVuakrJqs>

Linterna <https://unsplash.com/photos/qC117ahi-mE>

Calle https://unsplash.com/photos/_Tul8tZHlk4