Asignación de caja flexible M1C2

*Durante la segunda sección del Módulo 1, habrá revisado la muy útil herramienta Flexbox. Flexbox es otra herramienta útil que los desarrolladores pueden utilizar para controlar el comportamiento y el diseño de sus elementos HTML. La primera vez que usaste Flexbox fue en el curso Fundamentos de codificación. Ahora que ha revisado y aprendido más sobre Flexbox, ¡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.*

*El siguiente es un enlace a un repositorio de git que contiene el HTML y CSS de inicio para esta tarea. Copie y pegue el código inicial para que pueda seguir la tarea.*[*https://github.com/bottegaMentor/m1c2-flexbox--assignment/tree/main*](https://github.com/bottegaMentor/m1c2-flexbox--assignment/tree/main)

\*Conceptos incluidos en esta tarea: HTML, CSS, Flexbox, clases HTML, ID HTML, HTML Div, Img,h1 y etiquetas de párrafo, color de fondo CSS, color de texto CSS, hojas de estilo externas CSS y estilo en línea, dirección flexible, ajuste , Base, Crecer y Reducir, Justificar contenido, Alinear contenido, Alinear uno mismo y Alinear elementos.

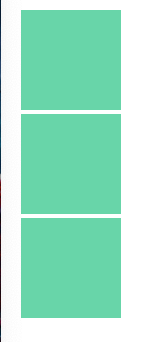
**Caja flexible 1:** Seleccione la clase de flexOne. Utilice flexbox para colocar los elementos en el div flexOne uno al lado del otro. Agregue 5 píxeles de margen a la clase cuadrada solo cuando estén dentro de flexOne.

**Ejemplo:**

****

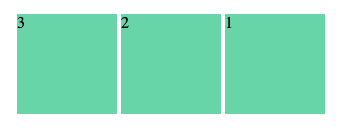
**Caja flexible 2:** Seleccione la clase de flexTwo. Utilice caja flexible. Establezca la dirección flexible en la columna. El valor predeterminado es fila, que coloca los elementos uno al lado del otro. La columna los vuelve a colocar uno encima del otro. Dale a los cuadrados dentro de flexTwo un margen de 2 píxeles.

**Ejemplo:**

****

**Caja flexible 3:** Seleccione la clase de flexThree. Repita los pasos de flexTwo, excepto que esta vez configurará flex-direction en fila-reversa. Agregue un 1, 2 y 3 dentro de los respectivos divs cuadrados dentro de flexThree para que pueda ver cómo el orden ahora se ha invertido.

**Ejemplo:**



**Caja flexible 4:** Seleccione la clase de flexFour. Utilice caja flexible. Establezca el ancho del div en 350 px. Dale un margen de 5px a los cuadrados dentro de flexFour. Configure el flujo flexible para "envolver". También cambie el color de fondo a #3d8b6d. Verá que el cuarto cuadrado automáticamente se ajusta hacia abajo y comienza una nueva fila porque no puede caber en los 450 píxeles con los otros cuadrados. Esto sería muy útil para un sitio web como Instagram, cuando sabes el formato que deseas para tus imágenes o publicaciones pero no sabes cuántas tendrás. Al usar flex-wrap, su código maneja automáticamente el formato de sus elementos.

**Ejemplo:**



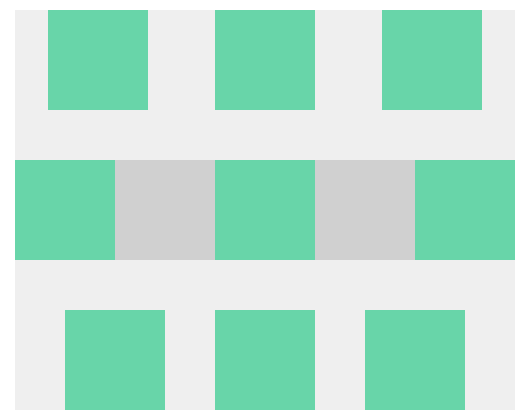
**Caja flexible 5:** Seleccione la clase de flexFive. Agregue un ancho de 350 px y use flexbox. Seleccione la clase de elemento para los divs dentro de flexFive. Establezca el color de fondo en #0b6643. Agregue un ancho y alto de 100 px. Agrega un margen de 2px para que podamos ver los cuadrados individuales. Ahora, apunte a la identificación de "crecer". En la identificación, establezca flex-grow en 2. Esto hará que el espacio sobrante, que normalmente se dividiría en partes iguales entre los elementos o no se usaría mediante flex, se aplicará al elemento del medio y hará que ocupe el doble de espacio adicional. que los otros elementos.

**Ejemplo:**

****

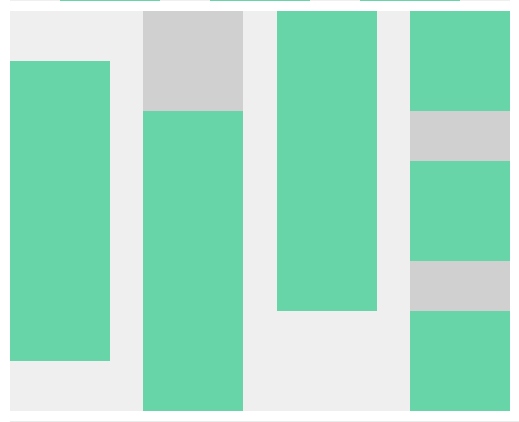
**Caja flexible 6:** Seleccione la clase de flexSix. Agregue una altura de 400 px y un ancho de 500 px a flexSix. Establezca el color de fondo en #f1f1f1. Establezca la dirección flexible en columna para que los divs secundarios vuelvan a estar uno encima del otro. En cada div secundario, con las clases de uno, dos y tres, use flexbox. En uno, establezca justificar contenido en espacio alrededor. En dos, establezca justificar contenido en espacio entre, establezca el margen en "50px 0" y establezca el color de fondo en #d6d6d6. El color nos ayudará a ver las filas que estamos creando. En tres, establezca justificar contenido en espacio uniformemente. Ahora puedes ver las diferentes formas en que se comportarán esas configuraciones de justificación de contenido.

**Ejemplo:**

****

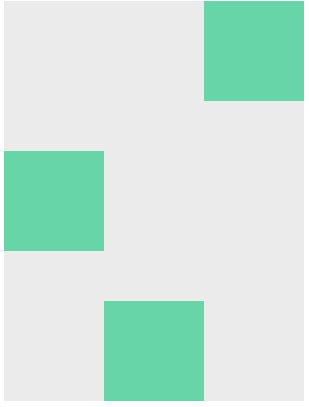
**Caja flexible 7:** Seleccione la clase de flexSeven. Dale una altura de 400 px y un ancho de 500 px. Utilice flexbox y establezca justify-content en el espacio intermedio. Esto nos dará 4 columnas. Dejaremos la dirección flexible en la fila predeterminada y usaremos justify-content para mover los elementos de una manera diferente. En los divs del uno al cuatro, use flexbox y establezca la dirección flexible en columna para que los elementos permanezcan uno encima del otro. En el div uno, podemos configurar justificar contenido en el centro. Dos para flexionar el extremo, tres para flexionar el inicio y cuatro para el espacio intermedio. Ahora puede ver cómo también podemos organizar elementos verticalmente usando justify-content si también configuramos flex-direction en columna.

**Ejemplo:**

****

**Caja flexible 8:** Seleccione la clase de flexEight. Utilice caja flexible. Dale al div una altura de 400 px y establece el color de fondo en #eeeeee. Dale un ancho de 300px. Ahora puedes seleccionar la clase de uno dentro de flexEight. Utilice flexbox y configure align-items al centro. Haga lo mismo con dos y configure align-items en flex-end. Configure tres para las mismas cosas y use flex-start para alinear elementos. Ahora puedes ver que alinear elementos también nos permite mover elementos verticalmente.

**Ejemplo:**



**Caja flexible 9:** Seleccione la clase de flexNine. Utilice caja flexible. Establezca el ancho en 450 px y el alto en 250 px. Establece el color de fondo en #eeeeee. Utilice elementos de alineación para que los elementos comiencen en la parte superior. Utilice justify-content para espaciar los elementos de manera uniforme. Ahora puede seleccionar la identificación especial. Utilice la propiedad align-self y configúrela en flex-end. De esta manera podemos anular el tercer cuadrado y hacer que comience en la parte inferior en lugar de en la parte superior.

**Ejemplo:**

