

**Grid-column-start**: establece el punto de partida de un elemento en nuestra grid.

**Grid-column-end**: establece el punto final de nuestro elemento en nuestra grid, esta propiedad puede utilizar valores negativos si así se requiere

En lugar de definir un elemento en la cuadrícula basado en la posición inicial y final, puedes definirlo basado en la longitud de columnas deseada usando la palabra clave **span**. Ten presente que **span** solo funciona con valores positivos., ósea se mueve a partir de ancho de la columna y no de las líneas de las columnas

Estas dos propiedades se agrupan en un sola de la siguiente forma

**Grid-column: start/ end;**

Al igual que las propiedades anteriores, donde start indica el inicio de la posición y end el final, esta acepta los valores numéricos con la palabra reservada span, y la abreviación de los dos valores en una sola propiedad, con la única diferencia es que en vez de llevar column en su nombre llevan row. Por ejemplo

**grid-row: start /end;**

Ahora bien, estas dos propiedades se pueden abreviar en una sola separando sus valores por barras oblicuas. por ejemplo

**Grid-area: row-start / column-start / row-end / column-end;**

puedes cambiar la posición de un elemento con la propiedad Order, esta propiedad acepta valores positivos como negativo. Por defecto el valor order de todos los elementos es de 0.

Grid: posicionamiento horizontal y vertical, order

Las dos primeras propiedades nos ayudan a extender un elemento dentro de nuestra grid.

Cuando grid-column-start se usa solo, la expansión por defecto del elemento en la cuadrícula será de exactamente una columna

Estar es la mejor manera de posicionar un elemento dentro de una cuadricula, además debemos tener muy en cuenta la palabra reservada span ya que ven ves de contar línea para posicionar contamos columnas o filas enteras. estos valores van dentro de los elementos hijos de nuestra grid



Construyendo el layout de una columna

Con grid-template-columns podemos establecer el tamaño de nuestras columnas de nuestra grid

Si tenemos mas de una columna en nuestro diseño y poseen el mismo ancho podemos utilizar la palabra reserva repeat para establecer cuantas columnas deseamos y su tamaño en nuestro grid, Por ejemplo:

**grid-template-columns: repeat( #columns, ancho de las columnas);**

CSS Grid también introduce una nueva medida, la fracción fr. Cada unidad fr asigna una porción del espacio disponible. Por ejemplo, si dos elementos están establecidos a 1fr y 3fr respectivamente el espacio se divide en 4 porciones iguales; el primer elemento ocupa 1/4 del espacio y el segundo elemento los 3/4 restantes.

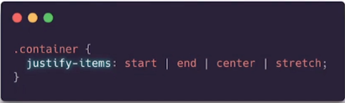
Los valores como pixeles, em, rem son valores fijos, la nueva unidad fr es relativa al tamaño sobrante de la grid

Para crear las filas debemos utilizar la propiedad grid-template-rows , esta propiedad acepta los mismo valores que tenemos en la anterior propiedad.  
  
  
si deseamos abreviar estas dos propiedades podemos utilizar la siguiente propiedad  
  
**grid-template: template-rows / template-columns;**

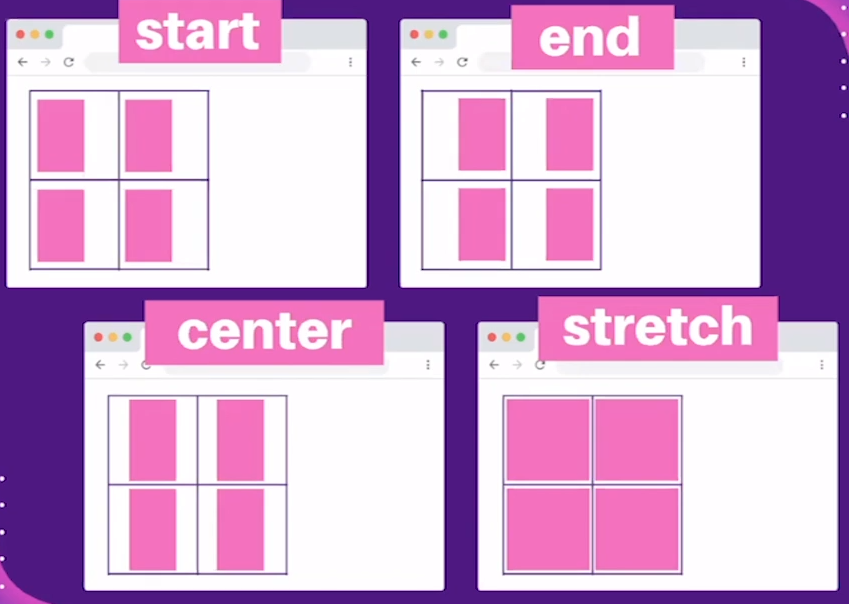


Como alinear un elemento dentro de la grid

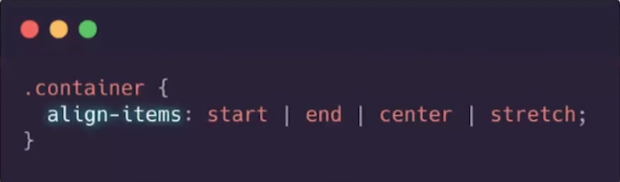
Justify-items tiene los siguientes valores, estos valores utilizan el eje row axis.   
Valores:

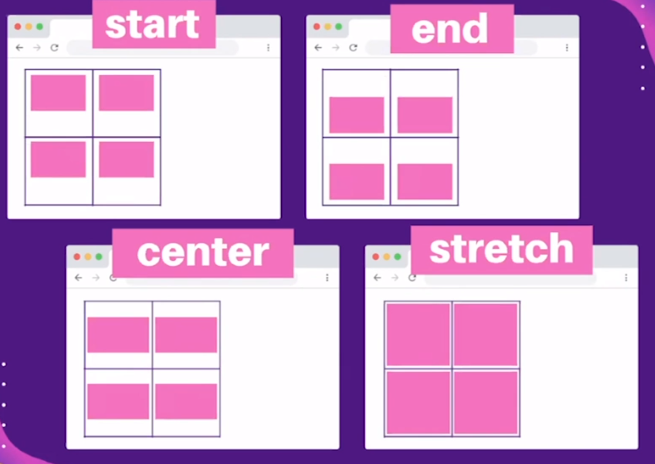


Su ejecución es la siguiente:



La propiedad de aling-items y sus valores ven efectos en el eje column axis, sus valores es el siguiente:

su ejecución es el siguiente



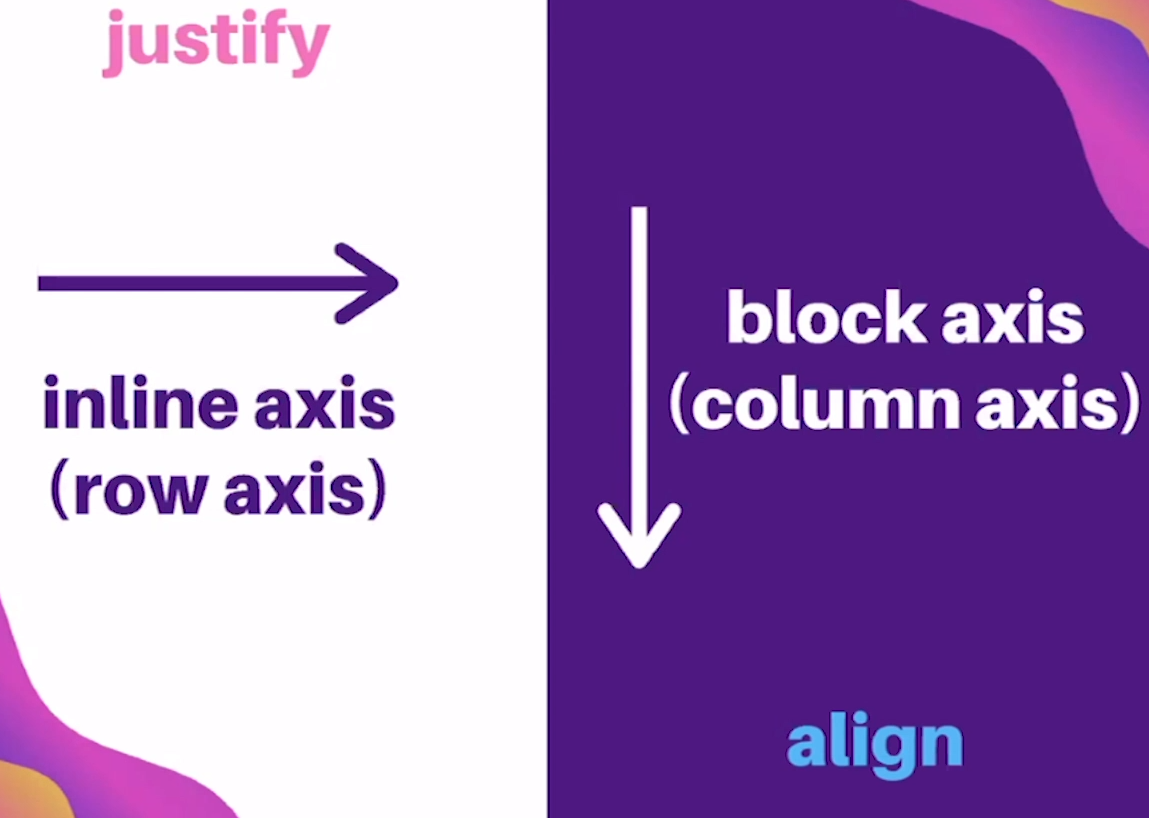
Su propiedad abreviada es **place-items** y el orden de valores son el siguiente:

**Place-items: aling-items / justify-items; p**

Para poder alinear los elementos hijos inmediatos de una grid utilizamos las siguientes propiedades

**Justify-items**

**Aling-items**La primera propiedad alinea de manera vertical todos nuestros hijos dentro de la grid,   
  
  
este alineamiento depende de la forma de escritura que tengamos en nuestra región, pues estos siguen el modelo de caja logia que se está implementando.



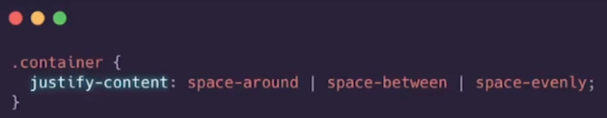
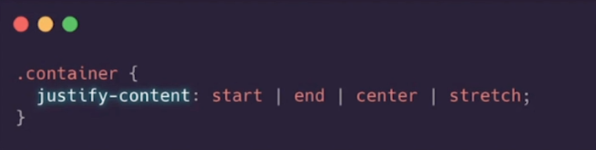
Los más importante la teoría y implantación de caja lógica para su alineación.



Alinear la grid dentro del contenedor.

la propiedad justify-content y sus valores se ejecutan dentro del eje row-axis.

Algunos valores son:



Como los anteriores valores de **justify-items y aling-items,** utiliza para su alineamiento el modelo caja lógica, esto es muy importante al tener al momento de la implementación de esta técnica.

Es muy importante al momento de tener en cuenta que el momento de aplicar esta propiedad se va a hacer dentro de la grid y que las divisiones deben ser valores absolutos y no relativos



