CSS 3 Guide – Grids

With CSS3 you can create grid-based layouts.

• Property	Values	Notes
Grid Container Properties		
display:	grid	Turns an HTML element into a grid container, which automatically turns its child elements into grid items.
	inline-grid	Used when you want two or more grids to share the same horizontal plane. If there is not enough space on the horizontal plane to fit all of the inline-grids, the ones that don't fit will wrap to the next line.

	Syntax
	display: grid;
	display: inline-grid;	

grid-template-columns:	Lengths Percentages Relative auto fr repeat(value,value) (see values defined at the end of this guide)	The **grid-template-columns** property defines the number of columns in your grid. If there are three space-separated values, then your grid will have three columns. The values assigned to this property will also define the width of each column. Be aware that if you create more columns than your grid container needs, you will end up with an empty column.

	Syntax
	grid-template-columns: 100px auto 25% auto;
	grid-template-columns: 100px 1fr 1fr 1fr;
	grid-template-columns: 100px repeat(3, 1fr);	

grid-template-rows:	Same as the **grid-template-columns** values (see values defined at the end of this guide)	The **grid-template-rows** property defines the number of rows in your grid. If there are three space-separated values, then your grid will have three rows. The values assigned to this property will also define the height of each row. Be aware that if you create more rows than your grid container needs, you will end up with an empty row.
	Syntax grid-template-rows: auto 80px 20% auto;	
place-content:	Is a shorthand property for the **align-content** and **justify-content** properties. If the place-content property is assigned one value: the value applies to both align-content and justify-content properties. If the place-content property is assigned to two values: the first value applies to the align-content property, and the second value applies to the justify-content property.	

• Property	Values	Notes
justify-content:	This property is used to horizontally align the grid items inside the grid container. The total width of the grid items must be less than the grid container's width for the **justify-content** property to have any effect. For example, if your grid container has a width of 400px and the amount assigned to all the grid columns totals 400px, then this property will have no effect.	
	space-evenly	The value "space-evenly" will give the columns an equal amount of space between and around them.
	space-around	The value "space-around" will give the columns an equal amount of space between them, and a half space on either end.
	space-between	The value "space-between" will give the columns an equal amount of space between them, and no space around them.
	center	The value "center" will align the grid items toward the horizontal middle of the grid container.
	start	The value "start" will align the grid items toward the left side of the grid container. This is the default setting of a grid container.
	end	The value "end" will align the grid items toward the right side of the grid container.
	Syntax `justify-content: space-evenly;`	
align-content:	This property is used to vertically align the grid items inside the grid container. The total height of all grid items must be less than the container's height for the **align-content** property to have any effect. For example, if your grid container has a height of 500px and the amount assigned to all the grid rows totals 500px, then this property will have no effect.	
	space-evenly	The value "space-evenly" will give the rows an equal amount of space between and around them.
	space-around	The value "space-around" will give the rows an equal amount of space between them, and a half space on either end.
	space-between	The value "space-between" will give the rows an equal amount of space between them, and no space around them.
	center	The value "center" will align the grid items toward the vertical middle of the grid container.
	start	The value "start" will align the grid items toward the top of the grid container. This is the default setting of a grid container.
	end	The value "end" will align the grid items toward the bottom of the grid container.
	Syntax `align-content: space-around;`	

• Property	Values	Notes
justify-items:	This property is used to horizontally align grid items within their column(s) track.	
	stretch	This default setting for grid items causes a grid item to stretch horizontally to fill the column(s) it is spanning.
	start	Aligns the grid item to the left of the column(s) it is spanning.
	center	Aligns the grid item to the center of the column(s) it is spanning.
	end	Aligns the grid item to the right of the column(s) it is spanning.
	Syntax justify-items: center;	
align-items:	This property is used to vertically align grid items within their row(s) track.	
	stretch	This default setting for grid items causes a grid item to stretch vertically to fill the row(s) it is spanning.
	start	Aligns the grid item to the top of the row(s) it is spanning.
	center	Aligns the grid item to the vertical center of the row(s) it is spanning.
	end	Aligns the grid item to the right of the row(s) it is spanning.
	Syntax justify-items: center;	
place-items:	Is a shorthand property for the **align-items** and **justify-items** properties. If the place-items property is assigned one value: the value applies to both align-items and justify-items properties. If the place-item property is assigned to two values: the first value applies to the align-items property, and the second value applies to the justify-items property.	
column-gap:	Lengths or relative measurements	This property sets the gap (gutter space) between the columns.
	Syntax grid-column-gap: 20px;	
row-gap:	Lengths or relative measurements	This property sets the gap (gutter space) between the rows.
	Syntax grid-row-gap: 20px;	
gap:	Lengths or relative measurements	This **grid-gap** property is a shorthand property that allows you to set both the column and row gaps. One value applies equally to both the column and row gaps. With two space-separated values, the first value applies to the row gap, and the second to the column gap.
	Syntax grid-gap: 20px 30px;	
grid-column-gap:	Deprecated use the column-gap property	
grid-row-gap:	Deprecated use the row-gap property	
grid-gap:	Deprecated use the gap property	

<table>
<tr><th>Property</th><th>Values</th><th>Notes</th></tr>
<tr><td colspan="3"><b>Grid Item Properties</b></td></tr>
<tr><td rowspan="4">grid-column:</td><td colspan="2">By default a grid container has one grid item for each cell, but you can style the grid items so that they will span into cells in adjacent columns and/or rows. The <b>grid-column</b> property defines which column(s) a grid item will occupy. You define where the item will start, and where the item will end.
Note: The <b>grid-column</b> property is a shorthand property for the <b>grid-column-start</b> and the <b>grid-column-end</b> properties.</td></tr>
<tr><td>number / number
1 / 4</td><td>The first number defines the column line where the grid item will start, and the 2nd number defines the column line where it will end. The <b>1 / 4</b> value means to start at column line 1 and span through 3 columns up to column line 4.</td></tr>
<tr><td>2 / span 3</td><td>start in column 2 and span to occupy a total of 3 columns</td></tr>
<tr><td colspan="2"><b>Syntax</b>
grid-column: 1 / 4;
grid-column: 2 / span 3;</td></tr>
<tr><td rowspan="4">grid-row:</td><td colspan="2">The <b>grid-row</b> property works similarly to <b>grid-column</b> except that <b>grid-row</b> applies to rows.
Note: The <b>grid-row</b> property is a shorthand property for the <b>grid-row-start</b> and the <b>grid-row-end</b> properties.</td></tr>
<tr><td>1 / 5</td><td>The first number defines the row line where the grid item will start, and the 2nd number defines the row line where it will end. The <b>1 / 5</b> value means to start at row line 1 and span through 3 rows up to row line 5.</td></tr>
<tr><td>2 / span 4</td><td>start in row 2 and span to occupy a total of 4 rows</td></tr>
<tr><td colspan="2"><b>Syntax</b>
grid-row: 1 / 5;
grid-row: 2 / span 4;</td></tr>
<tr><td rowspan="6">align-self:</td><td colspan="2">In a grid the align-self property aligns a grid item within the row or rows that it spans.</td></tr>
<tr><td>stretch</td><td>This default setting for grid items causes a grid item to stretch vertically to fill the row(s) it is spanning.</td></tr>
<tr><td>start</td><td>Aligns the grid item to the top of the row(s) it is spanning.</td></tr>
<tr><td>center</td><td>Aligns the grid item to the center of the row(s) it is spanning.</td></tr>
<tr><td>end</td><td>Aligns the grid item to the bottom of the row(s) it is spanning.</td></tr>
<tr><td colspan="2"><b>Syntax</b>
align-self: center;</td></tr>
<tr><td rowspan="6">justify-self:</td><td colspan="2">In a grid the justify-self property aligns a grid item within the column or columns that it spans.</td></tr>
<tr><td>stretch</td><td>This default setting for grid items causes a grid item to stretch horizontally to fill the column(s) it is spanning.</td></tr>
<tr><td>start</td><td>Aligns the grid item to the left of the column(s) it is spanning.</td></tr>
<tr><td>center</td><td>Aligns the grid item to the horizontal center of the column(s) it is spanning.</td></tr>
<tr><td>end</td><td>Aligns the grid item to the right of the column(s) it is spanning.</td></tr>
<tr><td colspan="2"><b>Syntax</b>
align-self: end;</td></tr>
</table>

• Property	Values	Notes
`place-self:`	Is a shorthand property for the **align-self** and **justify-self** properties. If the place-self property is assigned one value: the value applies to both align-self and justify-self properties. If the place-self property is assigned to two values: the first value applies to the align-self property, and the second value applies to the justify-self property.	

Grid Specific Measurement Systems

`absolute lengths`	inches, centimeters, points, etc. (must be non-negative)
`relative lengths`	pixels, em, vh, vw, etc. (must be non-negative)
`percentages`	10%, 25%, 42%, etc. (must be non-negative)
`flex`	**fr** is a fractional unit, and **1fr** is one part of the available space. **2fr** equals two parts of the available space. The available space is the space remaining after allotting the specified amount to those rows and/or columns that have a defined width. **fr** is a relative measurement system. Sometimes also called the flex factor **Example** `grid-template-columns: 100px 1fr 2fr 1fr;`
Keywords	

`auto`	If multiple rows/columns are assigned the auto value, then browser will determine the width/height of each grid track by distributing size until each auto row/column has reached its max-content size. Once a row/column has reached its max-content size it will not receive more sizing until all rows/columns assigned the auto value have reached their max-content size. From that point each auto row/column will have any remaining size distributed to them evenly.
	Examples
	`grid-template-rows: 80px auto 20%;`
	`grid-template-rows: auto 80px 20% auto;`

`max-content`	Sets the size of a row/column depending on the grid item with the largest content in that grid track. **Example** `grid-template-columns: 100px auto max-content`
`min-content`	Sets the size of a row/column depending on the largest unwrappable item in that grid track. This will be the smallest possible size that does not lead to an overflow. **Example** `grid-template-columns: 100px auto max-content`

Functions	

`repeat(recurrences, measurement)`	The repeat function can be applied to rows or columns and prevents repetition.
	Examples
	The code below repeats the '1fr' value 3 times.
	`grid-template-columns: 1fr 1fr 1fr;`
	The below repeat function can eliminate the above repetition.
	`grid-template-columns: repeat(3, 1fr);`

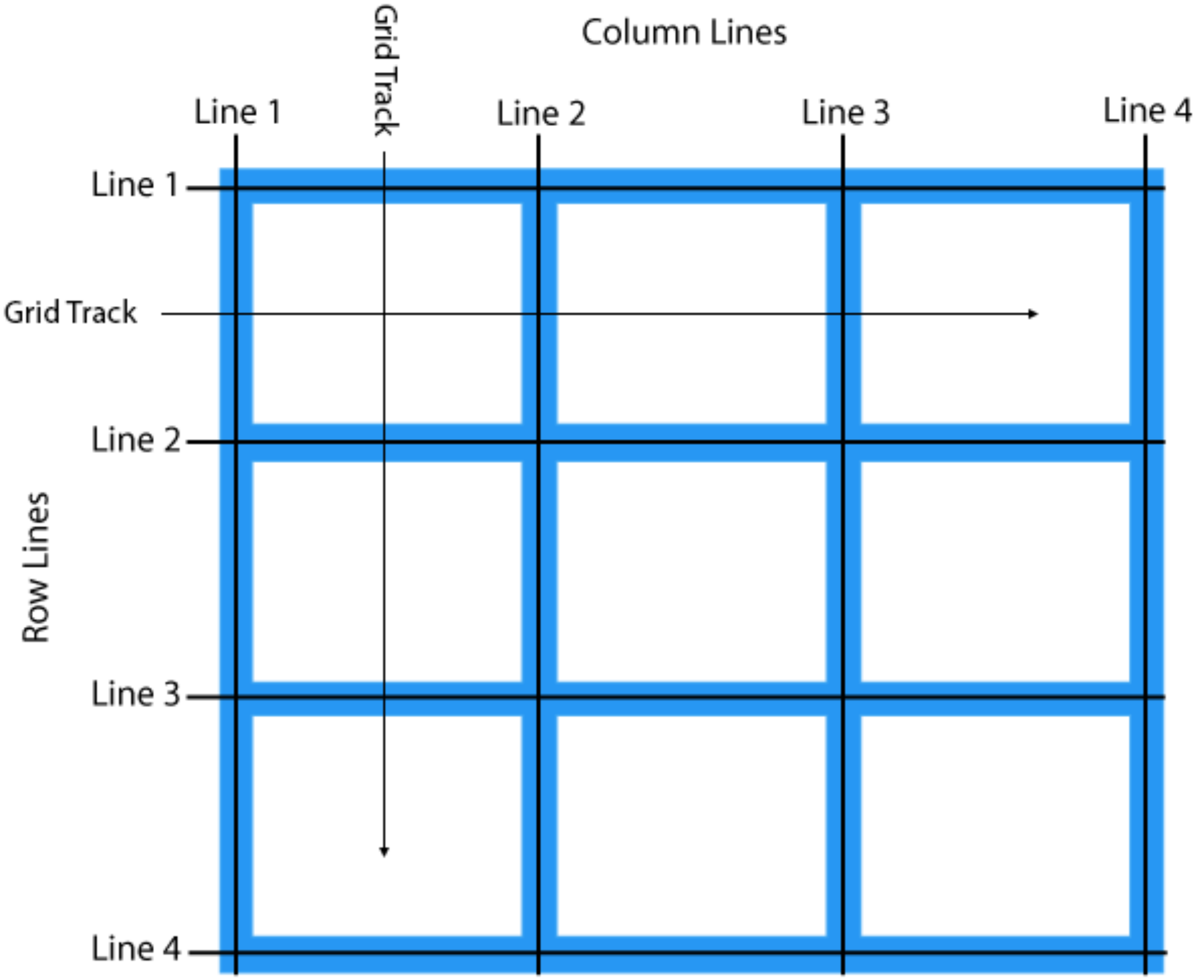
`minmax(min, max)`	Is a function that defines a size range of a grid track which is greater than or equal to min value and less than or equal to max value. If the max value is smaller than min, then max is ignored, and the function is treated as min. Supported values are lengths (non-negative), percentages (non-negative), flex (fr), and keywords (auto, max-content, and min-content). As a maximum, a <flex> value via the 'fr' measurement sets a grid track's flex factor. As a minimum, it is treated as zero (or minimal content, if the grid container is sized under a minimal content constraint). **Example** `grid-template-rows: minmax(10px, auto) 2fr 3fr;`
`fit-content(argument)`	This purpose of this function is to size the row or column to the size needed by its content up to the amount set for its argument. Accepts a length, relative value, or percentage as its argument. The single argument can be an absolute/relative length or a percentage. (A keyword or function is not a valid argument.) **Tech Explanation** This function clamps a given size to an available size according to the formula min(maximum size, max(minimum size, argument)) The function can be used as a track size in CSS Grid properties, where the maximum size is defined by max-content and the minimum size by auto, which is calculated similar to auto (i.e., minmax(auto, max-content)), except that the track size is clamped to the argument if it is greater than the auto minimum. **Example** `grid-template-rows: 30% fit-content(200px) 20%;`

Grid Lines and Tracks

The lines between rows are called *row lines*.

The lines between columns are called *column lines*.

The space between the lines that define a particular row/column is a grid track.


Grid Track
Column Lines
Line 1
Line 2
Line 3
Line 4
Line 1
Grid Track
Line 2
Row Lines
Line 3
Line 4