

Grids and flexbox cheat sheet
Note: ‘|’ stands for alternatives or OR.
Grid

The syntax for creating a grid:

```
1  selector{
2      display: grid; /* or inline-grid */
3  }
```

Grid shorthand consists of the following properties with default values:

grid

A grid will allow you organize the various elements on your page.

grid-template-rows: none

This feature allows you configure your elements so that they are organized similarly to rows on a table.

grid-template-columns: none

This feature allows you configure your elements but with this setting the elements are organized like columns on a table.

grid-template-areas: none

This feature allows you configure the names of a grid and how they sit in relation to one another.

grid-auto-rows: auto

Default setting for all row sizes that have not been explicitly configured.

grid-auto-columns: auto

Default setting for all column sizes that have not been explicitly configured.

grid-auto-flow: row

Default location for rows that are not explicitly allocated.

column-gap: normal

This sets the gap between the columns

row-gap: normal

This sets the gap between the rows

Grid properties for container

grid-template-columns: measurement units | % units |repeat()

Defines the line names, and maintains a constant size of column items. Can accept a range of different measurement sizes.

grid-template-rows: measurement units | % units |repeat()

Defines the line names, and maintains a constant size of rows. Can accept a range of different measurement sizes.

grid-auto-columns: measurement unit (fixed value for all columns)

Determines the default size for columns that have not been explicitly configured.

grid-auto-rows: measurement unit (fixed value for all rows)

Determines the default size for rows that have not been explicitly configured.

grid-template: "header header" auto

This allows you define and maintain named cells on a grid

"main right" 75vh

This defines two cells named main and right, that have a sizing of 75% of the viewport height.

"footer footer" 20rem

This defines two cells named footer and footer, that have a sizing of 20 root em (rem). This defines the size in relation to the html font size.

Gap

grid-gap: measurement units

Determines the gap between rows and columns

grid-column-gap: measurement units

Determines the gap between columns

grid-row-gap: m-unit-1 m-unit-2

Determines the gap between columns

Alignment

justify-items: start | center | end | stretch

Defines the default space that is allot to each item on the grid

align-items: start | center | end | stretch

Defines the default space related to an item along the grid’s block axis

place-items: start | stretch /* shorthand for two properties above */

This feature allows you align items with the block and inline directions.

Justification

justify-content: start | center | end | stretch | space-between | space-evenly | space-around

Defines browser allocation of space to content items in relation to the main-axis

align-content: start | center | end | stretch | space-between | space-evenly | space-around

Defines browser allocation of space to content items in relation to cross axis and block axis

place-content: center | start

This feature allows you align items with the block and inline directions.

Positioning

grid-auto-flow: row | column | dense

This relates to how the items are placed automatically within the grid

grid-auto-columns: measurement units

This relates to the size for columns created without specific size specifications

grid-auto-rows: measurement units

This relates to the size for rows created without specific size specifications

Grid properties for items (child)

grid-column: column position /* E.g. 1/2 */

Allows for specifying where on the grid the column is to start.

grid-column-start: column start position

This property determines the starting column position an item is placed on a grid.

grid-column-end: column end position

This property determines the end column position an item is placed on a grid.

grid-row: row position /* E.g. 1/2 */

Allows for specifying where on the grid the row is to start.

grid-row-start: row start position

This property determines the starting row position an item is placed on a grid.

grid-row-end: row end position

This property determines the end row position an item is placed on a grid.

Justification and alignment

justify-self: start | center | end | stretch

Determines how an item is positioned inside its aligned container in relation to the appropriate axis.

align-self: start | center | end | stretch

Aligns an item within a grid area.

place-self: start | stretch /* shorthand for two properties above */

This setting lets one align and justify an item within a block.

Flexbox

The syntax for creating a flexbox:

Here the selector can refer to any of the following flex attributes

Attribute selector

Class Selector

ID Selector

Type Selectors

Universal Selectors

The display relates to how you want the selector to be shown. Setting display to flex makes the given selector a flex box. Setting display to **inline-flex** makes the selector a flex box container while will be inline.

Properties for flexbox container

flex-direction: row | row-reverse | column | column-reverse

It is possible to specify the direction your elements will follow. Traditionally text goes from left to right which is flex's default setting however it can be set from right to left or even top to bottom. The four flex-direction are:

row : organized from left to right

row-reverse: organized from right to left

column: organized from top to bottom

column-reverse: organized from bottom to top.

flex-wrap: wrap | nowrap

The standard layout is to plot the elements from left to right in a straight line. The wrap feature allows you customize this to match the size of the window displaying the page.

wrap: Automatically wrap the items with as the window space gets smaller.

Nowrap: Default setting, items remain rigid and don't respond to adjustments made to the window size.

align-items: flex-start | flex-end | center | Stretch

This determines how the flex items are to be positioned on the page. Items can be aligned in a variety of ways

Flex-start: Similar to standard writing, items start at the top left-hand corner and are positioned from left to right

Flex-end: Position begins in the bottom right hand corner.

Center: Item is positioned from the center.

Stretch: item expands to fill the container.

justify-content: flex-start | flex-end | center | space-between | space-evenly

Justify-content determines the alignment of the flex items.

Flex-start: goes from right to left along the main axis.

Flex-end: goes from left to right along the main axis.

Center: Starting at the middle, alignments expands from there.

Space-between: first and last item are flush with the left and right wall respectively, every other item is evenly spaced.

Space-evenly: each item is equidistant from each other and the boundary wall

Properties for flexbox items (child)

flex-grow: factor of flex's main size

This attribute enables the flex container to grow proportionally to the other containers present.

flex-shrink: factor of flex's main size

This allows elements to shrink in relation to items around it.

flex-basis: auto | factor of main's size | measurement unit

The sets the initial main size of an item. It can be overridden if other stylized elements are configured.

order:position in flex /* Set ascending by default */

The standard positioning of items is by source order, however this feature will enable you to configure where the items appear on the page.

align-self: start | center | end | stretch

This determines where on the page the child items will be positioned. Similar to the main flex attributes, start is to the left and end is to the right.