

CSS Grid cheatsheet

Container

```
.grid-container {
  /* Display properties */
  display: grid;
  display: inline-grid;

  /* Columns and rows */
  grid-template-columns: 1rem 2rem 1rem; /* Measurement units */
  grid-template-columns: 25% 50% 25%; /* Percentage units */
  grid-template-columns: 1em auto 1em 2fr; /* Fill remaining widths with
  grid-template-columns: repeat(12, 1fr); /* Repeat columns without needin
  grid-template-columns: subgrid; /* Use column tracks defined on parent g

  grid-template-rows: 1rem 10% auto repeat(5, 10px); /* Mix any group, sam
  grid-template-rows: subgrid; /* Use row tracks defined on parent grid */

  /* Automatic columns and rows */

  grid-auto-columns: 10px; /* No matter how many columns of content end up
  grid-auto-rows: 1rem; /* No matter how many rows of content end up in th

  /* Areas */
  grid-template-areas:
    "header header"
    "main aside"
    "footer footer"; /* Grid-style */

  grid-template-areas: "header header" "main aside" "footer footer"; /* In

  /* Template shorthand */
  grid-template:
    "header header" auto
    "main aside" 10vh
    "footer footer" 10rem
    / 80% 20%;

  /* The above is the same as below long-hand */
  grid-template-columns: 80% 20%;
  grid-template-rows: auto 100vh 10rem;
  grid-template-areas:
    "header header"
    "main aside"
    "footer footer";

  /* Gaps */
  grid-row-gap: 1rem;
  grid-column-gap: 0.5rem; /* Define values separately */

  grid-gap: 1rem 0.5rem; /* Short-hand for row / column */
  grid-gap: 1rem; /* Gap in both dimensions */

  /* Item justification (horizontal or column alignment) */
  justify-items: start; /* Align items to the left */
  justify-items: center; /* Align items centered within its column */
  justify-items: end; /* Align items to the right */
  justify-items: stretch; /* (default) Fills available area (horizontally)

  /* Item alignment (vertical or row alignment) */
  align-items: start; /* Align items to the top */
  align-items: center; /* Align items centered within its row */
  align-items: end; /* Align items to the bottom */
  align-items: stretch; /* (default) Fills available area (vertically) */

  /* Place item shorthand */
  place-items: start stretch;

  /* The above is the same as below long-hand */
  align-items: start;
  justify-items: stretch;

  /* Content justification (horizontal or column alignment) */
  justify-content: start; /* Align content to the left */
  justify-content: center; /* Align content centered horizontally within t
  justify-content: end; /* Align content to the right */
  justify-content: stretch; /* (default) Fills available area (horizontall

  justify-content: space-around; /* Chooses a space for both sides of the
  justify-content: space-between; /* Chooses a space to go between columns
  justify-content: space-evenly; /* Chooses a space that goes between all
```

Child

```
.grid-child {
  /* Column position */
  grid-column-start: 1;
  grid-column-end: 2;

  grid-column: 1 / 2; /* Short hand */
  grid-column: 1 / span 2; /* Span 2 columns without explicitly defining a
  grid-column: 1; /* Start in and occupy a single column */

  /* Row position */
  grid-row-start: 2;
  grid-row-end: 4;

  grid-row: 2 / 4; /* Short hand */
  grid-row: 2 / span 3; /* Span 3 rows without explicitly defining an endpoint */
  grid-row: 1; /* Start in and occupy a single row */

  /* Area positioning */
  grid-area: header; /* You can use a named grid area from the container */

  grid-area: 2 / 1 / 4 / 2; /* Or you can use positioning. This is equivalent
  grid-row-start: 2;
  grid-column-start: 1;
  grid-row-end: 4;
  grid-column-end: 2;

  /* Self justification (horizontal or column alignment) */
  justify-self: start; /* Align item to the left */
  justify-self: center; /* Align item centered within its column */
  justify-self: end; /* Align item to the right */
  justify-self: stretch; /* (default) Fills available area (horizontally)

  /* Self alignment (vertical or row alignment) */
  align-self: start; /* Align item to the top */
  align-self: center; /* Align item centered within its row */
  align-self: end; /* Align item to the bottom */
  align-self: stretch; /* (default) Fills available area (vertically) */

  /* Placement shorthand */
  place-self: start stretch;

  /* The above is the same as below long-hand */
  align-self: start;
  justify-self: stretch;

}
```

```

/* Content alignment (horizontal or column alignment) */
align-content: start; /* Align content to the top */
align-content: center; /* Align content centered vertically within the grid */
align-content: end; /* Align content to the bottom */
align-content: stretch; /* (default) Fills available area (vertically) */

align-content: space-around; /* Chooses a space for the top and bottom of each row */
align-content: space-between; /* Chooses a space to go between rows, no space at the top or bottom */
align-content: space-evenly; /* Chooses a space that goes between all rows */

/* Place item shorthand */
place-content: center start;

/* The above is the same as below long-hand */
align-content: center;
justify-content: start;

/* Automatic grid positioning */

grid-auto-flow: row; /* Left-to-right rows, then top-to-bottom*/
grid-auto-flow: column; /* Top-to-bottom columns, then left-to-right */
grid-auto-flow: dense; /* Responds with best-guess on left-to-right, top-to-bottom */

/* There is one final shorthand for all container properties in one */

/* Explicit grid columns, rows, and areas */
grid:
  "header header" auto
  "main aside" 100vh
  "footer footer" 10rem
  / 80% 20%; /* You can include a template as the only value, which is equivalent to below */
grid-template: auto auto 100vh 10rem;
grid-template-columns: 80% 20%;
grid-template-rows: auto 100vh 10rem;
grid-template-areas:
  "header header"
  "main aside"
  "footer footer";

/* Automatic grid flows */
grid: 1rem / auto-flow dense 1fr; /* You can include rows, a flow, and a column */
grid-template-rows: 1rem;
grid-auto-flow: dense;
grid-auto-columns: 1fr;

grid: auto-flow dense 1rem / repeat(10, 10%); /* Conversely, you can do the same with columns */
grid-auto-flow: dense;
grid-auto-rows: 1rem;
grid-template-columns: repeat(10, 10%);

}

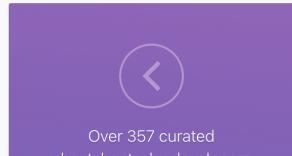
```

References

- [GRID: A simple visual cheatsheet](#)
- [CSS Tricks: A Complete Guide to Grid](#)
- [Browser support](#)

▶ [0 Comments](#) for this cheatsheet. [Write yours!](#)

[devhints.io](#) / Search 357+ cheatsheets



Other CSS cheatsheets

Sass
cheatsheet

CSS flexbox
cheatsheet

Top cheatsheets

Elixir
cheatsheet

ES2015+
cheatsheet

Cheatsheets, by developers
for developers.

[Devhints home](#)

[CSS system fonts
cheatsheet](#) ●

[CSSnext
cheatsheet](#) ●

[React.js
cheatsheet](#) ●

[VIM/VIIT
cheatsheet](#) ●

[Stylus
cheatsheet](#) ●

[Bootstrap
cheatsheet](#) ●

[Vim
cheatsheet](#) ●

[Vim scripting
cheatsheet](#) ●