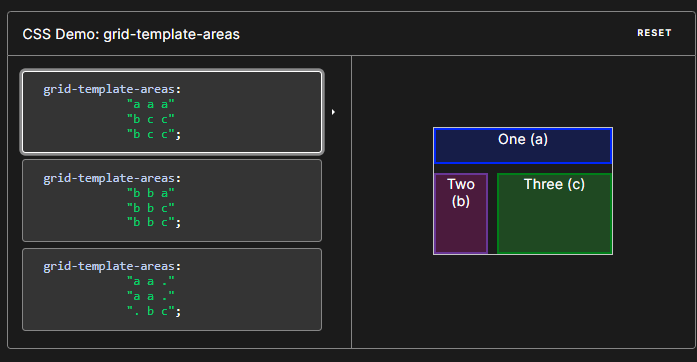
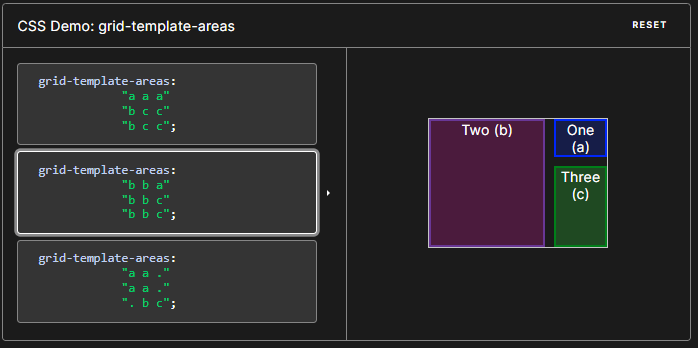
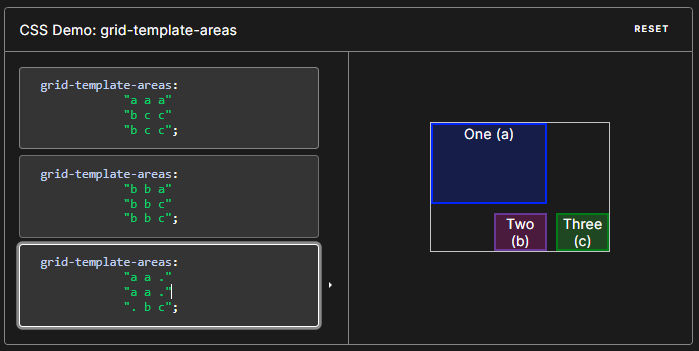
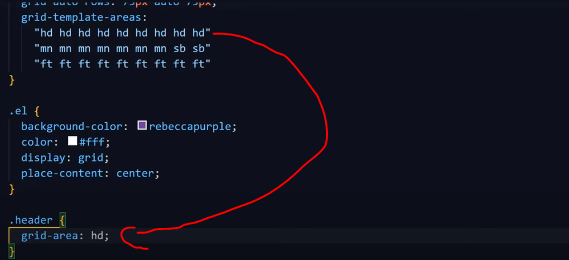
**CSS Grid-template-area:**

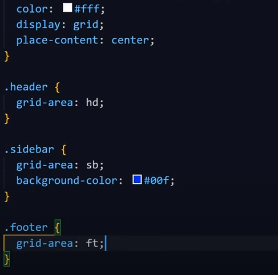
The grid-template-areas CSS property specifies named grid areas, establishing the cells in the grid and assigning them names.

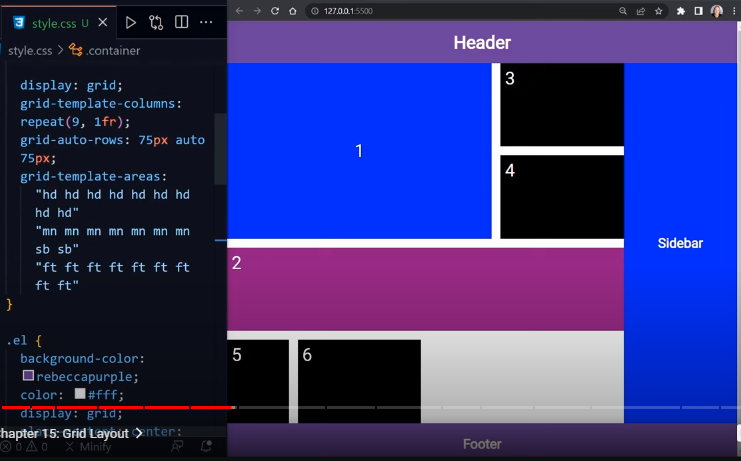












Those areas are not associated with any particular grid item, but can be referenced from the grid-placement properties grid-row-start, grid-row-end, grid-column-start, grid-column-end, and their shorthands grid-row, grid-column, and grid-area.

Syntax

/\* Keyword value \*/

grid-template-areas: none;

/\* <string> values \*/

grid-template-areas: "a b";

grid-template-areas:

"a b b"

"a c d";

/\* Global values \*/

grid-template-areas: inherit;

grid-template-areas: initial;

grid-template-areas: revert;

grid-template-areas: revert-layer;

grid-template-areas: unset;

Values

1. none

The grid container doesn't define any named grid areas.

1. <string>+

A row is created for every separate string listed, and a column is created for each cell in the string. Multiple cell tokens with the same name within and between rows create a single named grid area that spans the corresponding grid cells. Unless those cells form a rectangle, the declaration is invalid.

**Example**:

HTML

<section id="page">

<header>Header</header>

<nav>Navigation</nav>

<main>Main area</main>

<footer>Footer</footer>

</section>

CSS

#page {

display: grid;

width: 100%;

height: 250px;

grid-template-areas:

"head head"

"nav main"

"nav foot";

grid-template-rows: 50px 1fr 30px;

grid-template-columns: 150px 1fr;

}

#page > header {

grid-area: head;

background-color: #8ca0ff;

}

#page > nav {

grid-area: nav;

background-color: #ffa08c;

}

#page > main {

grid-area: main;

background-color: #ffff64;

}

#page > footer {

grid-area: foot;

background-color: #8cffa0;

}

