grid-template-areas

Baseline Widely available ✓ ✓ ✓ ✓

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

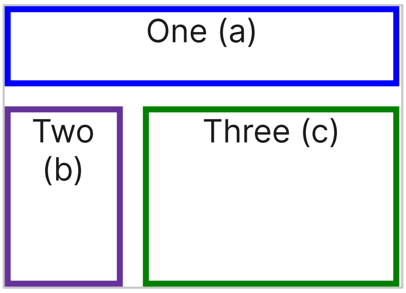
Try it

CSS Demo: grid-template-areas

```
grid-template-areas:
  "a a a"
  "b c c"
  "b c c";
```

```
grid-template-areas:
  "b b a"
  "b b c"
  "b b c";
```

```
grid-template-areas:
  "a a ."
  "a a ."
  ". b c";
```



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

CSS

```
/* Keyword value */
grid-template-areas: none;

/* <string> values */
grid-template-areas: "a b";
grid-template-areas:
  "a 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

`none`

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

`<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.

All the remaining unnamed areas in a grid can be referred using *null cell tokens*. A null cell token is a sequence of one or more `.` (U+002E FULL STOP) characters, e.g., `.`, `...`, or `.....` etc. A null cell token can be used to create empty spaces in the grid.

Formal definition

Initial value	`none`
Applies to	grid containers
Inherited	no
Computed value	as specified
Animation type	discrete

Formal syntax

```
grid-template-areas =
  none       |
  <string>+
```

Sources: CSS Grid Layout Module Level 2

Examples

Specifying named grid areas

HTML

HTML

```html
<div id="page">
  <header>Header</header>
  <nav>Navigation</nav>
  <main>Main area</main>
  <footer>Footer</footer>
</div>
```

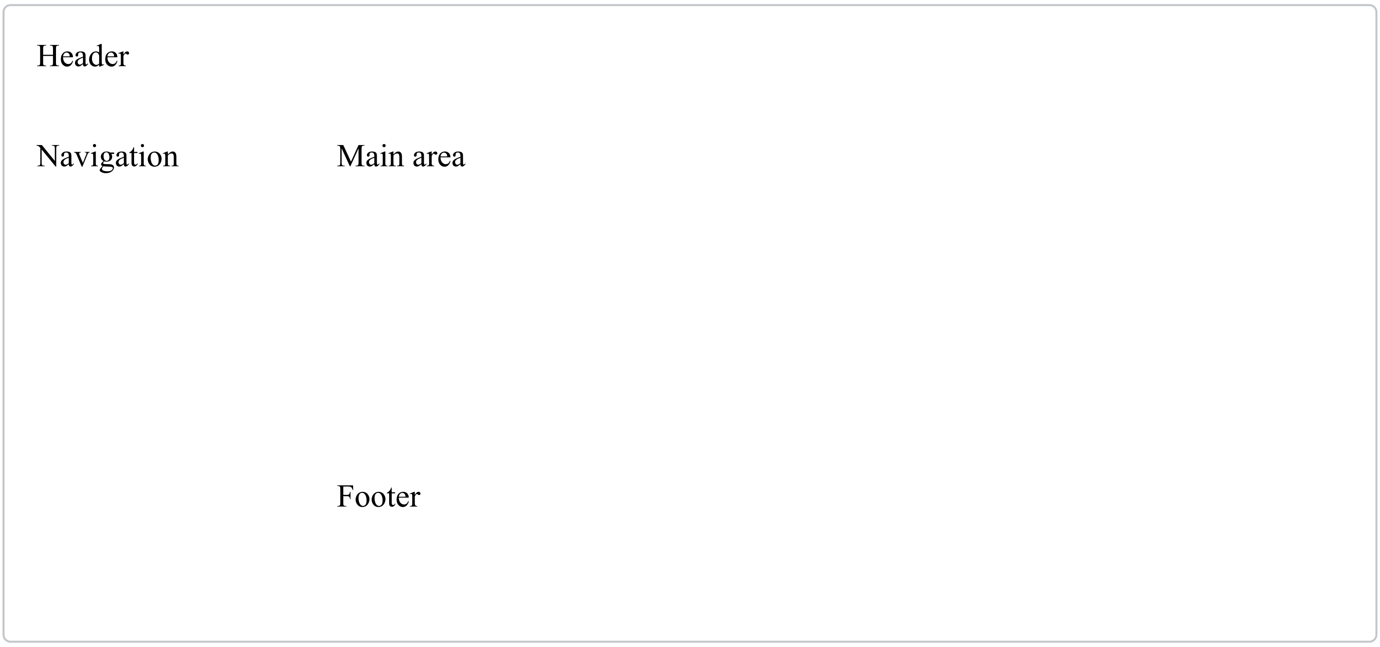
CSS

CSS

```css
#page {
  display: grid;
  width: 100%;
  height: 250px;
  grid-template-areas:
    "head head"
    "nav  main"
    ".  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;
}
```

In the above code, a null token (`.`) was used to create an unnamed area in the grid container, which we used to create an empty space at the bottom left corner of the grid.

RESULT



Header

Navigation

Main area

Footer

Specifications

Specification
[CSS Grid Layout Module Level 2 # grid-template-areas-property](#)

Browser compatibility

Report problems with this compatibility data • View data on GitHub

	🖥					📱						
	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS

`grid-template-areas`	✓ 57	✓ 16	✓ 52	✓ 44	✓ 10.1	✓ 57	✓ 52	✓ 43	✓ 10.3	✓ 6	✓ 57	✓ 10.3
`none`	✓ 57	✓ 79	✓ 52	✓ 44	✓ 10.1	✓ 57	✓ 52	✓ 43	✓ 10.3	✓ 7	✓ 57	✓ 10.3

✓ Full support

See also

- `grid-template-rows`
- `grid-template-columns`
- `grid-template`
- Grid template areas
- Video: Grid template areas

mdn_

Your blueprint for a better internet.