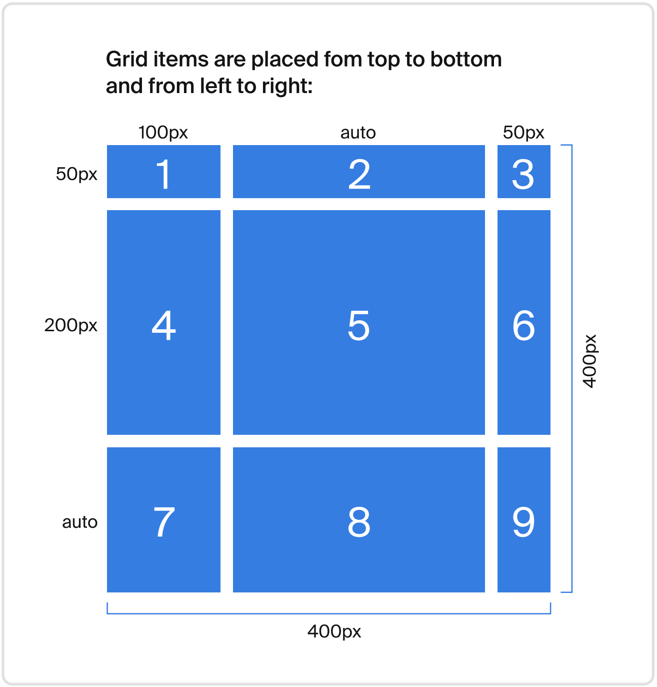
Grid Layout: Part 1

Sprint 3

Basic 3x3 Grid Example

- Set `display: grid;` on parent container
- Use `grid-template-rows`/`grid-template-columns` to set dimensions of rows and columns
- Use `column-gap`, `row-gap` or `column-gap` to control the gutters between rows and columns
- Items are placed from top to bottom, and left to right

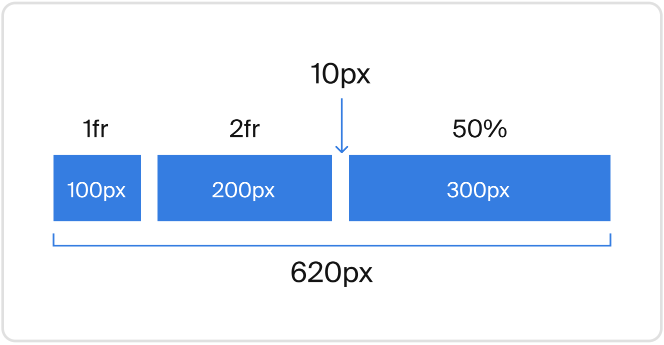
```
.container {
  display: grid;
  grid-template-columns: 100px auto 50px;
  grid-template-rows: 50px 200px auto;
  gap: 10px 20px;
}
```



Grid items are placed fom top to bottom and from left to right:
100px
auto
50px
50px
200px
auto
1
2
3
4
5
6
7
8
9
400px
400px

Relative Grid Units

- All normal CSS units are available to use in your grid templates, such as `px`, `%`, and `vw`
- Rows and columns that are set to `auto` will be assigned dimensions automatically, after all explicitly set dimensions
  are allocated
- The `repeat` function allows you to avoid repetition:
  `grid-template-columns: 20% 20% 20% 20% 20%;` → `grid-template-columns: repeat(5, 20%);`
- The `fr` unit represents a fraction of the entire container



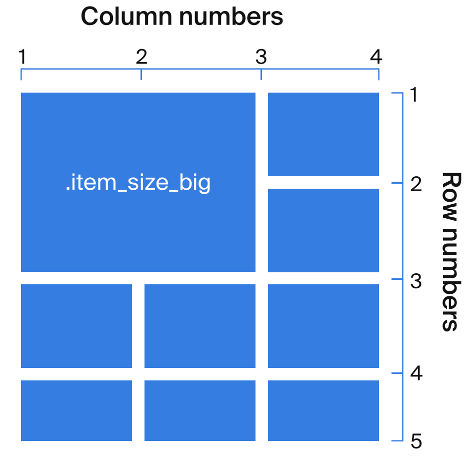
Grid Layout: Part 2

Sprint 3

Position Elements in a Grid

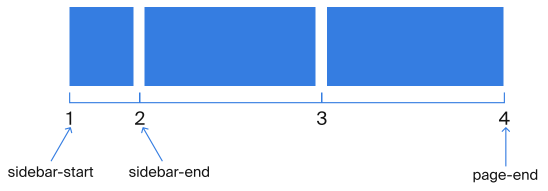
Use negative values to count from the left:

```
.container {
  display: grid;
  grid-template-columns: repeat(3, auto);
  grid-template-rows: repeat(3, auto);
  gap: 10px;
}

.item_size_big {
  grid-column-start: 1;
  grid-column-end: 3;
  grid-row: 1/3; /* shorthand syntax */
}
```



Naming Row and Column Lines

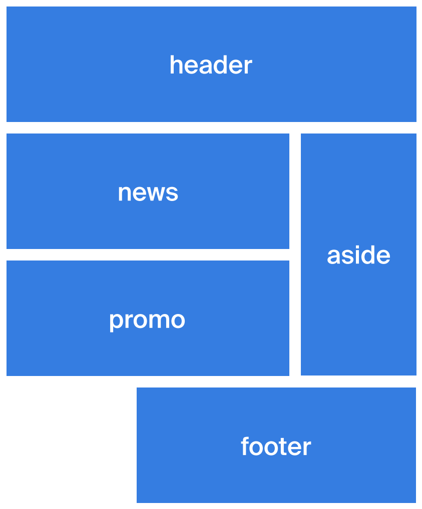
```
/* Three column layout with some named grid lines */

grid-template-rows: [sidebar-start] 100px
                    [sidebar-end] 1fr 1fr
                    [page-end];
```



Grid Areas

- Unnamed areas must be marked with a neutral character, like a period
- Grid areas must be contiguous

```
.container {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-template-rows: repeat(4, 1fr);
  grid-template-areas:
    "header header header"
    "news news aside"
    "promo promo aside"
    ". footer footer";
}

/* assign a grid area to each element */
.header {
  grid-area: header;
}
```



header
news
aside
promo
footer