

CSS Grid Layout excels at dividing a page into major regions or defining the relationship in terms of size, position, and layer, between parts of a control built from HTML primitives.

Like tables, grid layout enables an author to align elements into columns and rows. However, many more layouts are either possible or easier with CSS grid than they were with tables. For example, a grid container's child elements could position themselves so they actually overlap and layer, similar to CSS positioned elements.

Basic example

The example below shows a three-column track grid with new rows created at a minimum of 100 pixels and a maximum of auto. Items have been placed onto the grid using line-based placement.

HTML

```
<div class="wrapper">
  <div class="one">One</div>
  <div class="two">Two</div>
  <div class="three">Three</div>
  <div class="four">Four</div>
  <div class="five">Five</div>
  <div class="six">Six</div>
</div>
```

CSS

```
.wrapper {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-gap: 10px;
  grid-auto-rows: minmax(100px, auto);
}
.one {
  grid-column: 1 / 3;
  grid-row: 1;
```

```
}  
.two {  
  grid-column: 2 / 4;  
  grid-row: 1 / 3;  
}  
.three {  
  grid-column: 1;  
  grid-row: 2 / 5;  
}  
.four {  
  grid-column: 3;  
  grid-row: 3;  
}  
.five {  
  grid-column: 2;  
  grid-row: 4;  
}  
.six {  
  grid-column: 3;  
  grid-row: 4;  
}
```