Grid Templates

```css
.grid-contai
  height: 140px;
  display: grid;
  grid-template-columns: [col-1-start] 150px [col-1-end] 180px auto;
  grid-template-rows: auto [row-2-start] 100px [row-2-end];
}
```

Turn <div> into a grid

Assign name to line

Assign width/height

150px   180px   auto

<unnamed line>

100px

row-2-start

auto

row-2-end

col-1-start   col-1-end   <unnamed line>   <unnamed line>

From a Grid Cell Perspective

Shorthand: `grid-column: col-1-start / col-1-end`

```css
.cell-1 {
  grid-column-start: col-1-start;
  grid-column-end: col-1-end;
  grid-row-start: 1;
  grid-row-end: 2;
}
```

Refer to line names (if set) or line numbers

1

2

col-1-start

col-1-end

From a Grid Cell Perspective

```css
.cell-2 {
  grid-column-start: col-1-end;
  grid-column-end: 4;
  grid-row-start: 1;
  grid-row-end: 2;
}
```

1

2
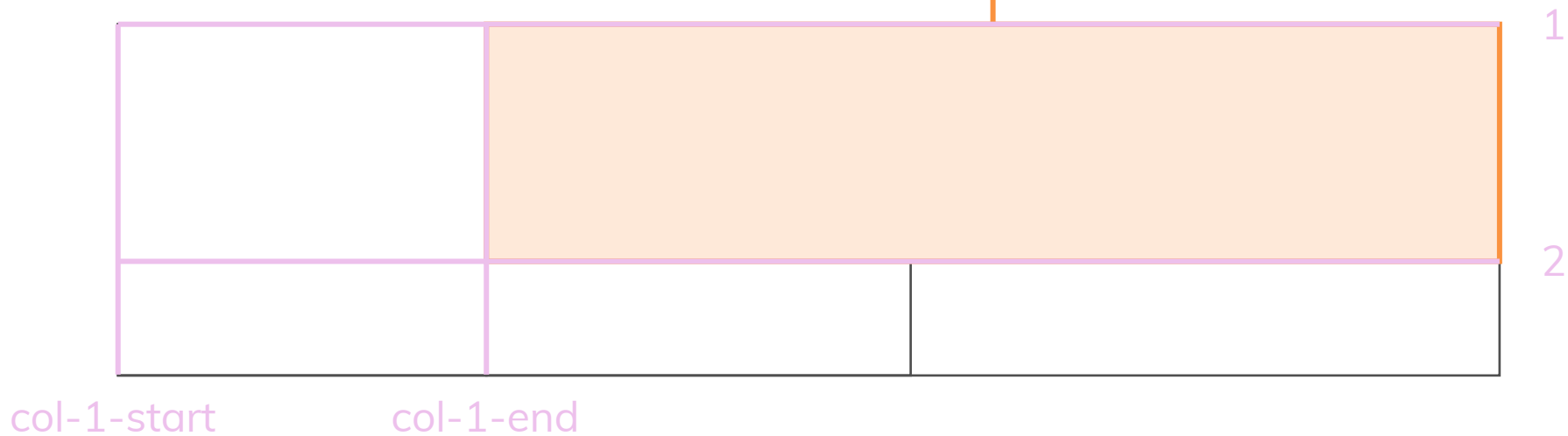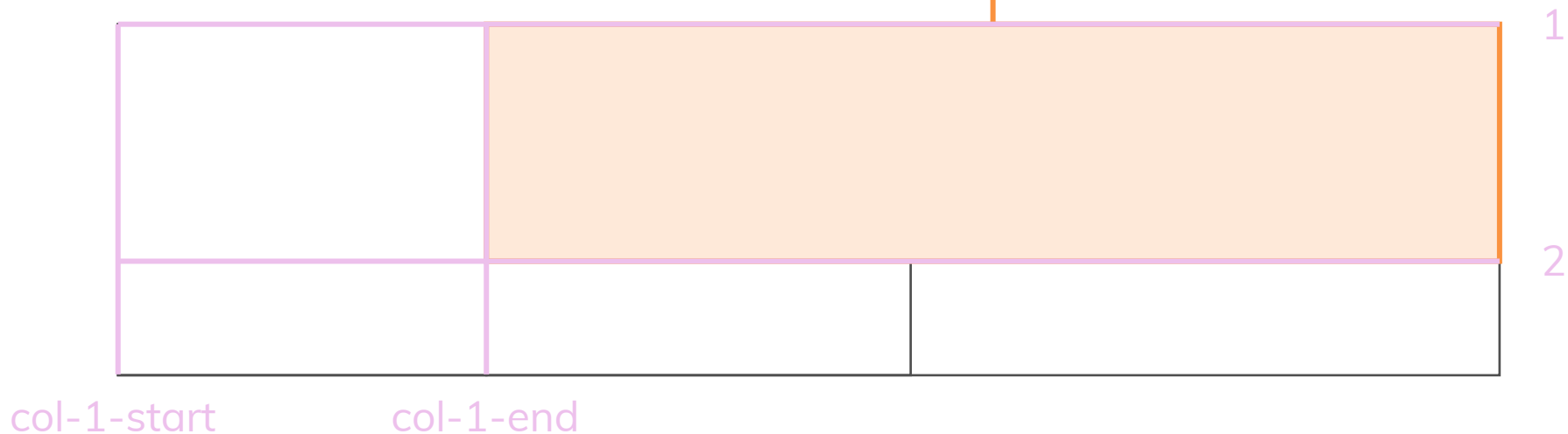
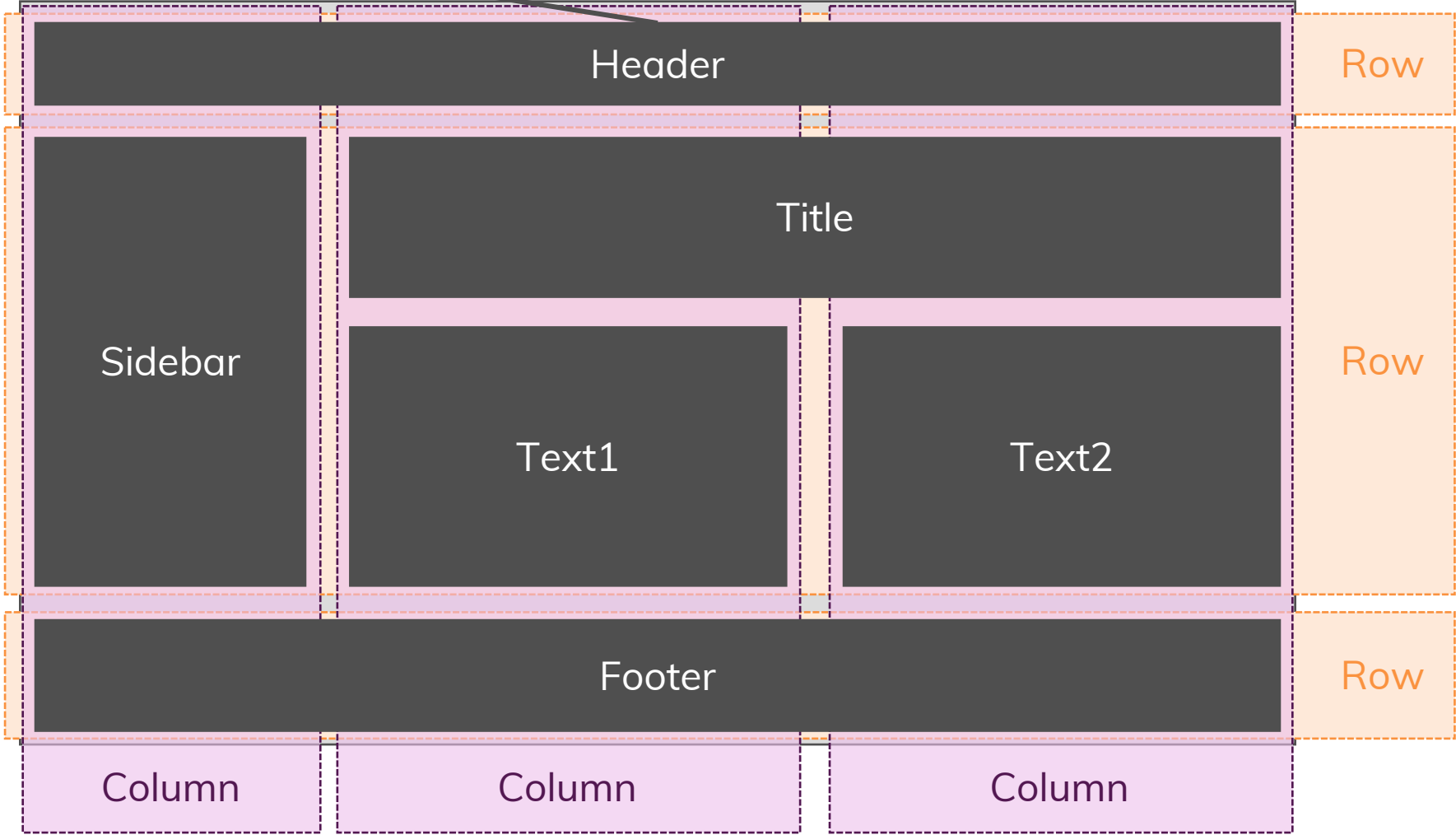col-1-start   col-1-end

An Alternative Way

```css
.cell-2 {
  grid-column-start: col-1-end;
  grid-column-end: span 2;
  grid-row-start: 1;
  grid-row-end: 2;
}
```

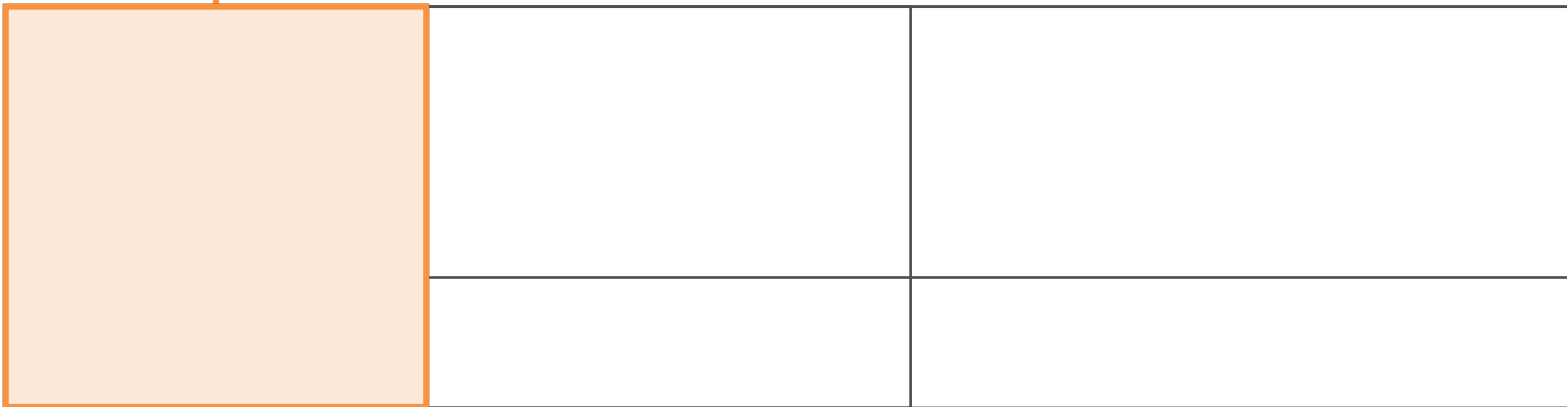By the Way: Overlapping is allowed, control stacking via `z-index`

1

2

col-1-start   col-1-end

Grid Areas

Header Area

Header

Title

Sidebar

Text1

Text2

Footer
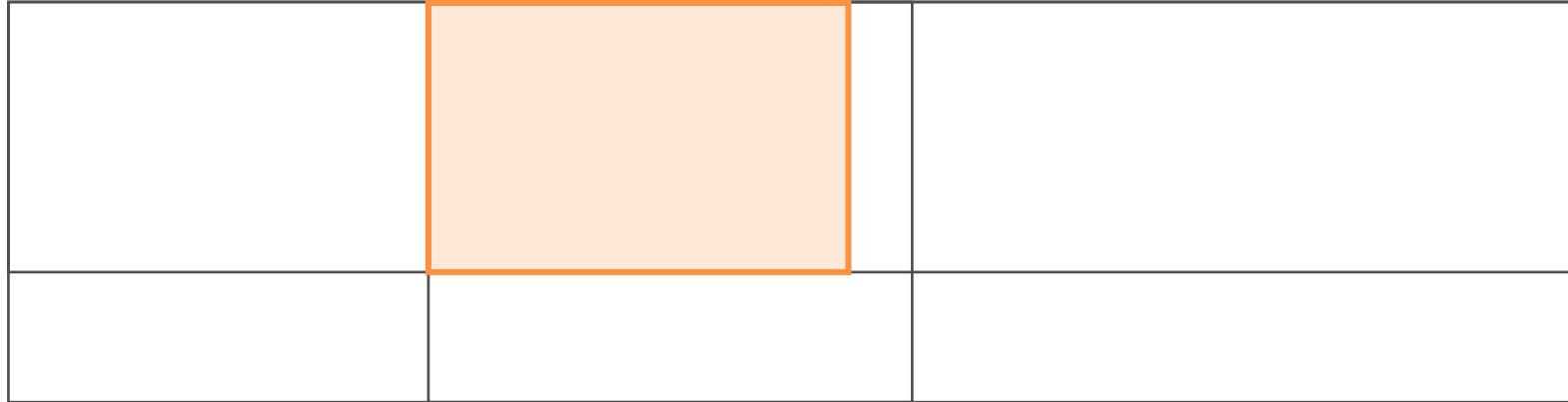
Row

Row

Row

Column

Column

Column

From a Grid Cell Perspective

```css
.sidebar {
  grid-area: sidebar;
}
```
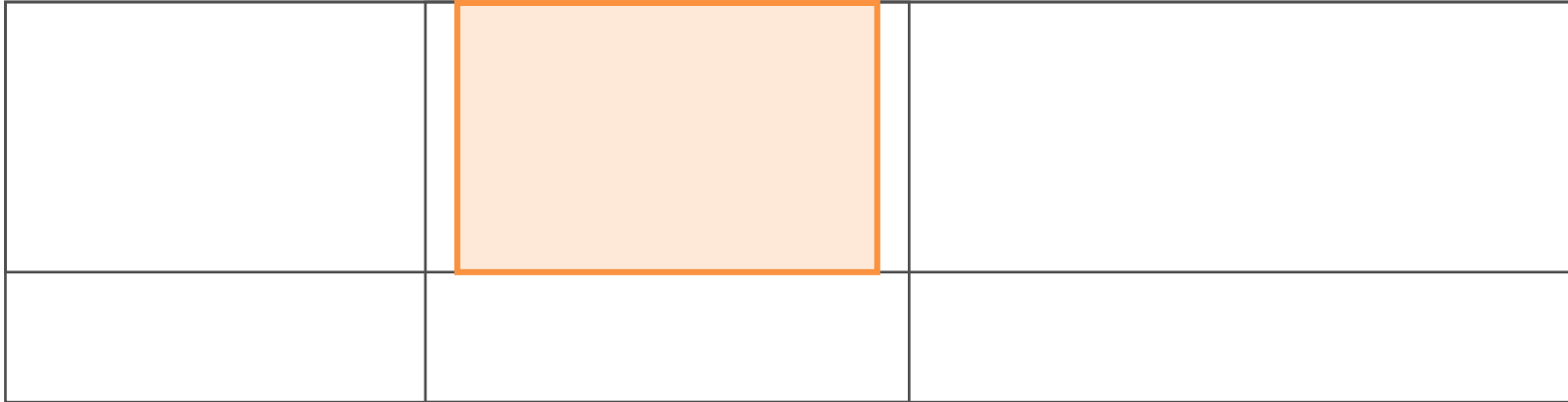
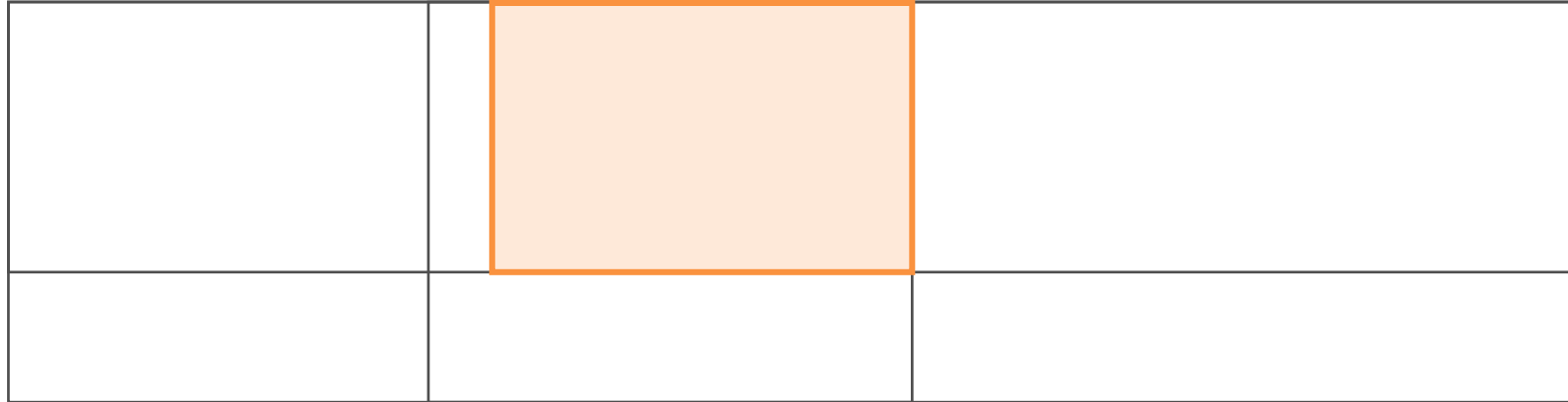# Grid Alignment – Horizontal Start

```css
.grid-container {
  justify-items: start;
}
```

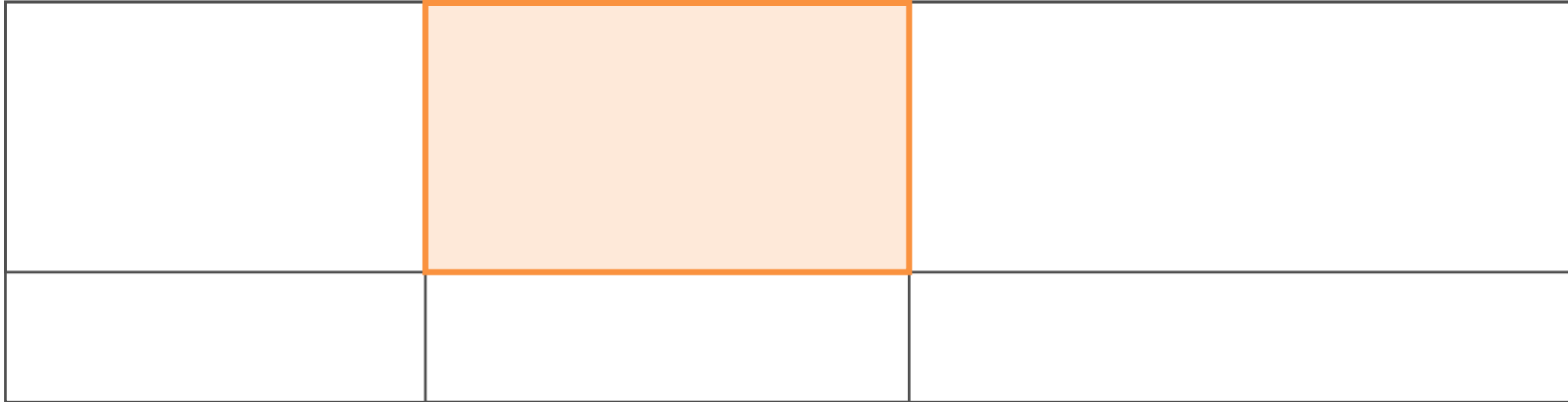# Grid Alignment - Horizontal Center

```css
.grid-container {
  justify-items: center;
}
```

Grid Alignment - Horizontal End


```css
.grid-container {
  justify-items: end;
}
```

Grid Alignment - Horizontal Stretch
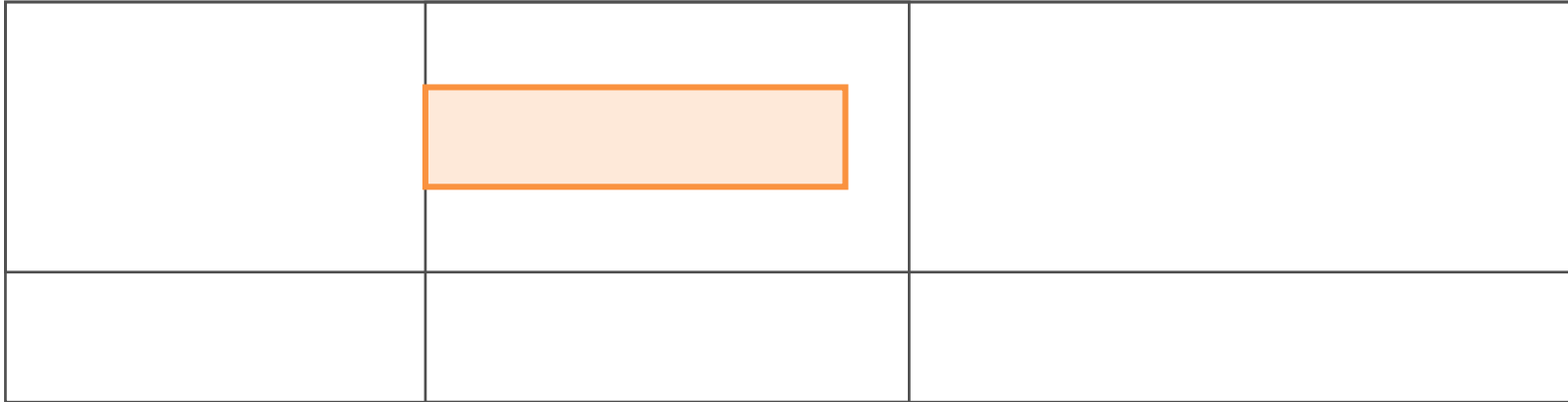
```css
.grid-container {
  justify-items: stretch;
}
```

Grid Alignment – Vertical Start


```css
.grid-container {
  align-items: start;
}
```
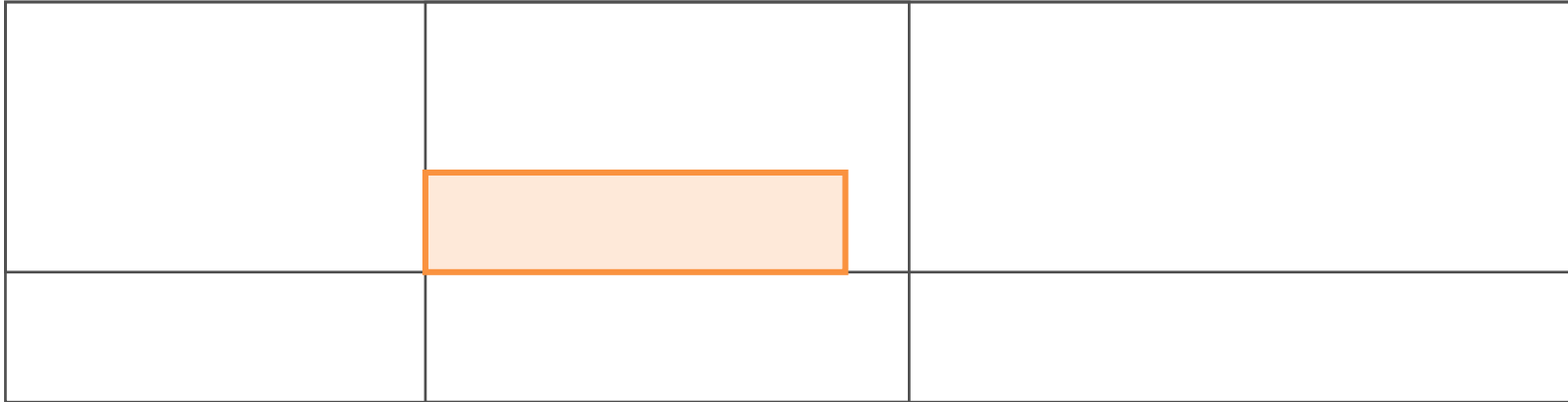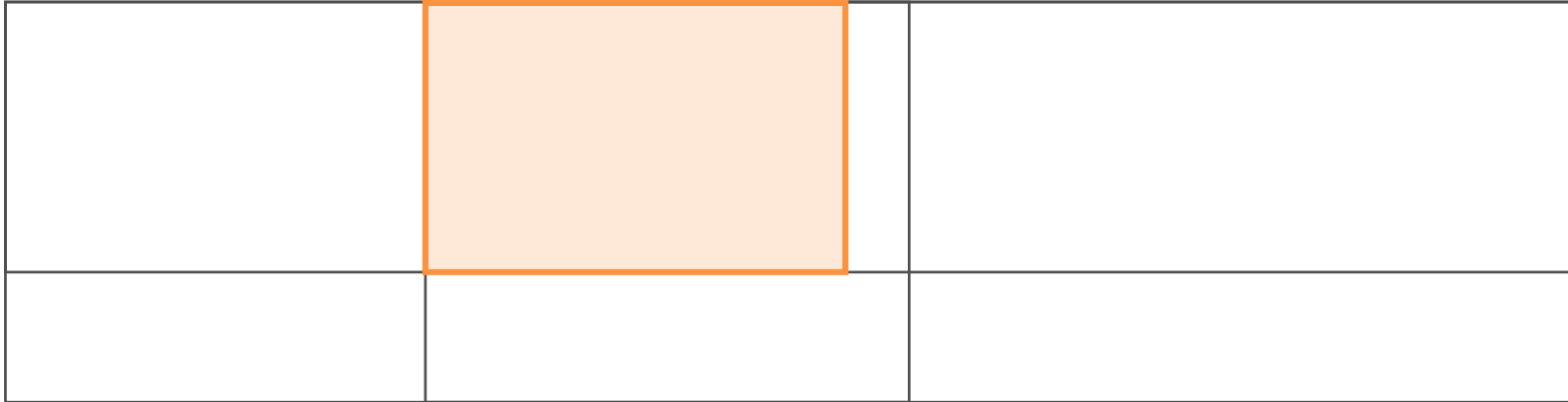
Grid Alignment - Vertical Center

```css
.grid-container {
  align-items: center;
}
```

Grid Alignment - Vertical End
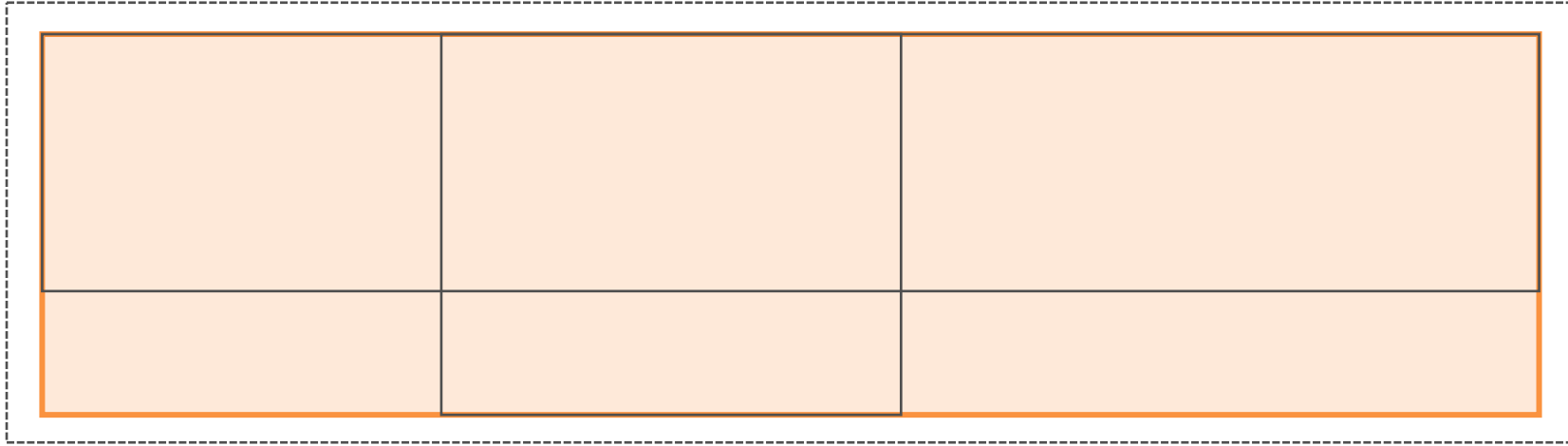
```css
.grid-container {
  align-items: end;
}
```

Grid Alignment - Vertical Stretch

```css
.grid-container {
  align-items: stretch;
}
```
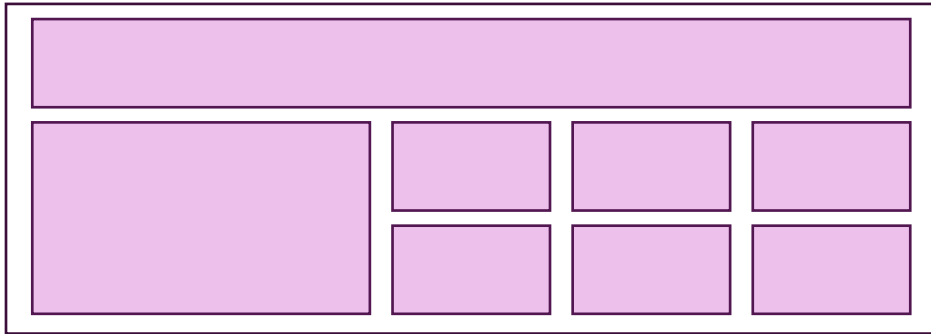
Grid Alignment – Align Grid Itself


```css
.grid-container {
  justify-content: start | end | center | stretch | space-around | space-
between | space-evenly;
  align-content: start | end | center | stretch | space-around | space-
between | space-evenly;
}
```

CSS Grid vs Flexbox

CSS Grid
Two-dimensional Positioning

CSS Flexbox
One-dimensional Positioning