

Responsive patterns with subgrid

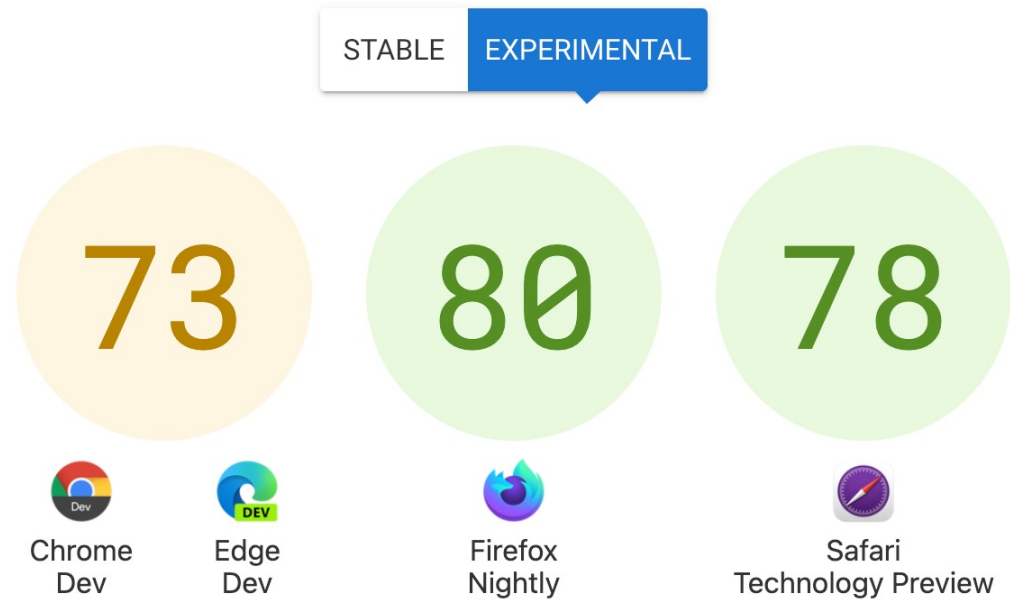
@rachelandrew



```
grid-template-rows: subgrid;  
grid-template-columns: subgrid;
```

Interop 2022 Dashboard

These scores represent how browser engines are doing in
15 focus areas and 3 joint investigation efforts.



April 30, 2022

Card One

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Two

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Three

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Four

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Five

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Six

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card One

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Two

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer. This footer has much more content than the designer expected.

Card Three

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Four

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Five

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Six

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Only **direct children**
become grid or flex items.

Their children return to normal flow.

The following content uses block and inline layout

Block elements extend to fill the container in the inline direction. They break onto a new line. Any inline elements such as a **span** do not break onto a new line.



Media Object

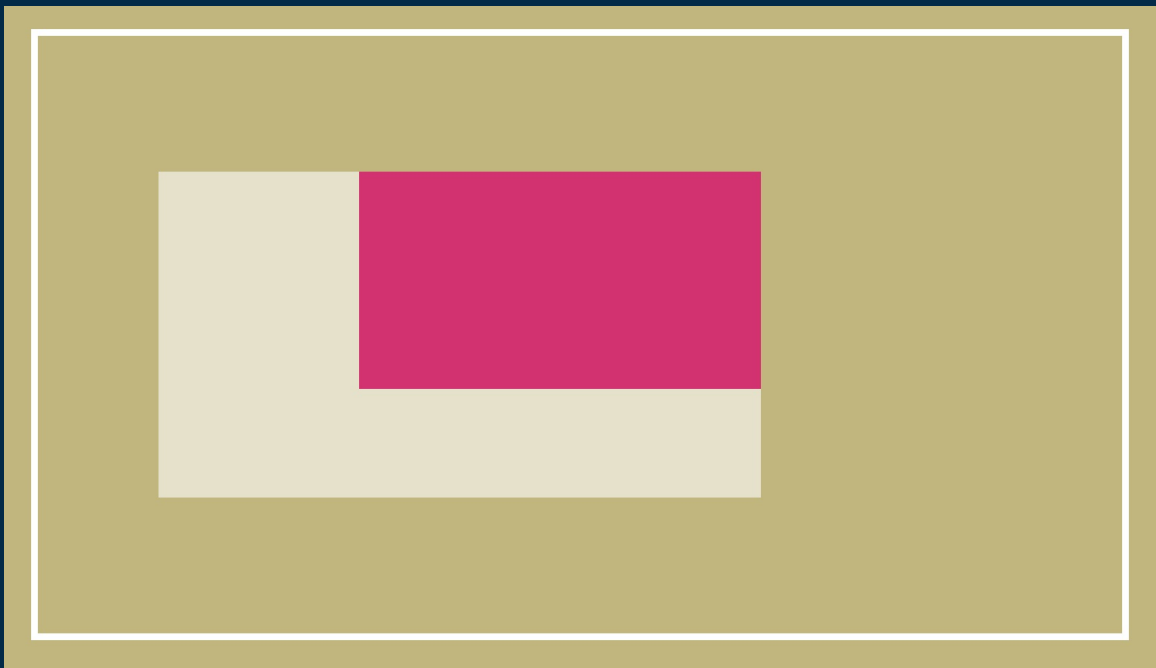
Content inside the flex item returns to normal flow layout.

Unless you change the value of display, elements continue to display using these block and inline rules.



Subgrid

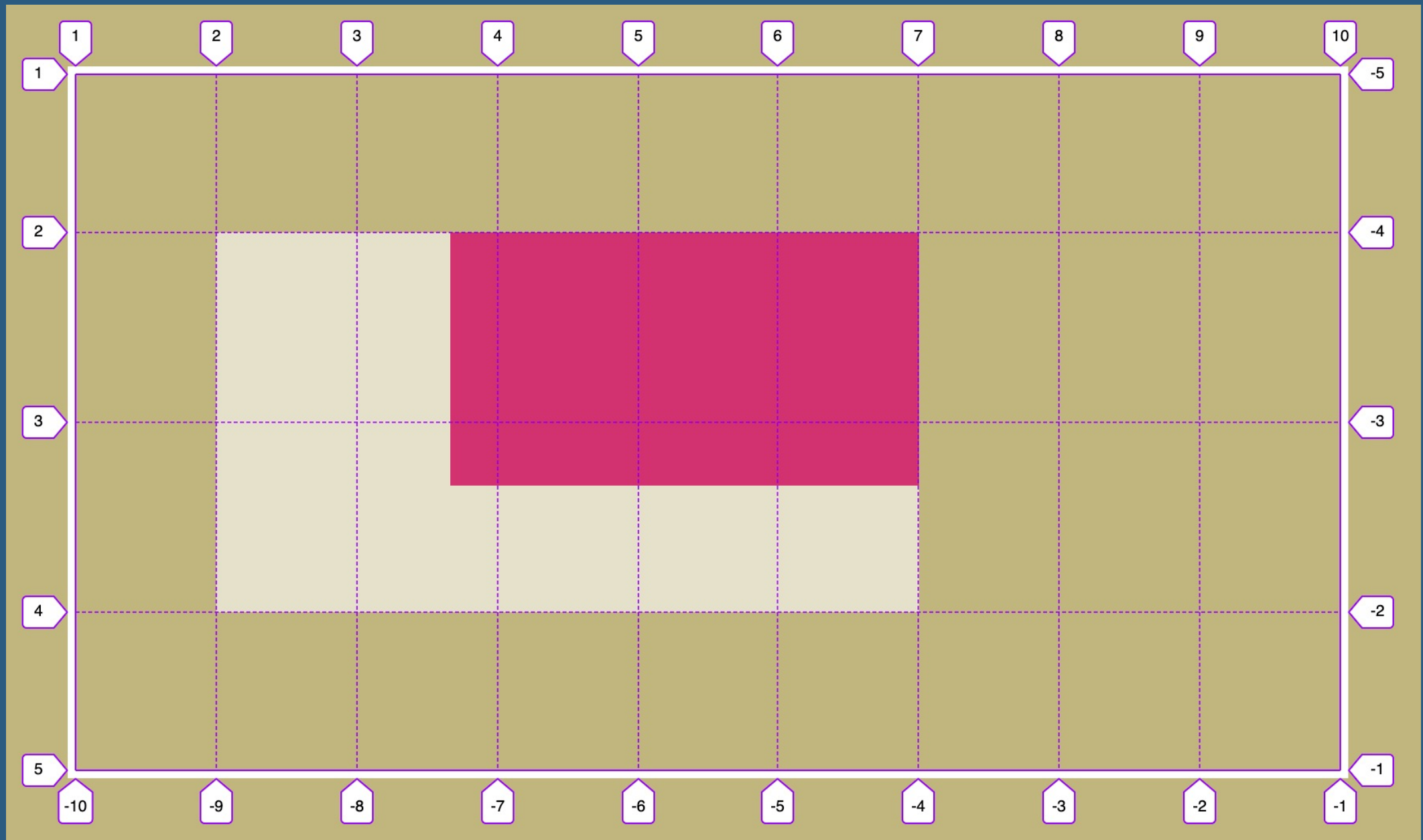
Create a grid on a grid item which uses the grid tracks defined on the parent – for rows, columns or both.

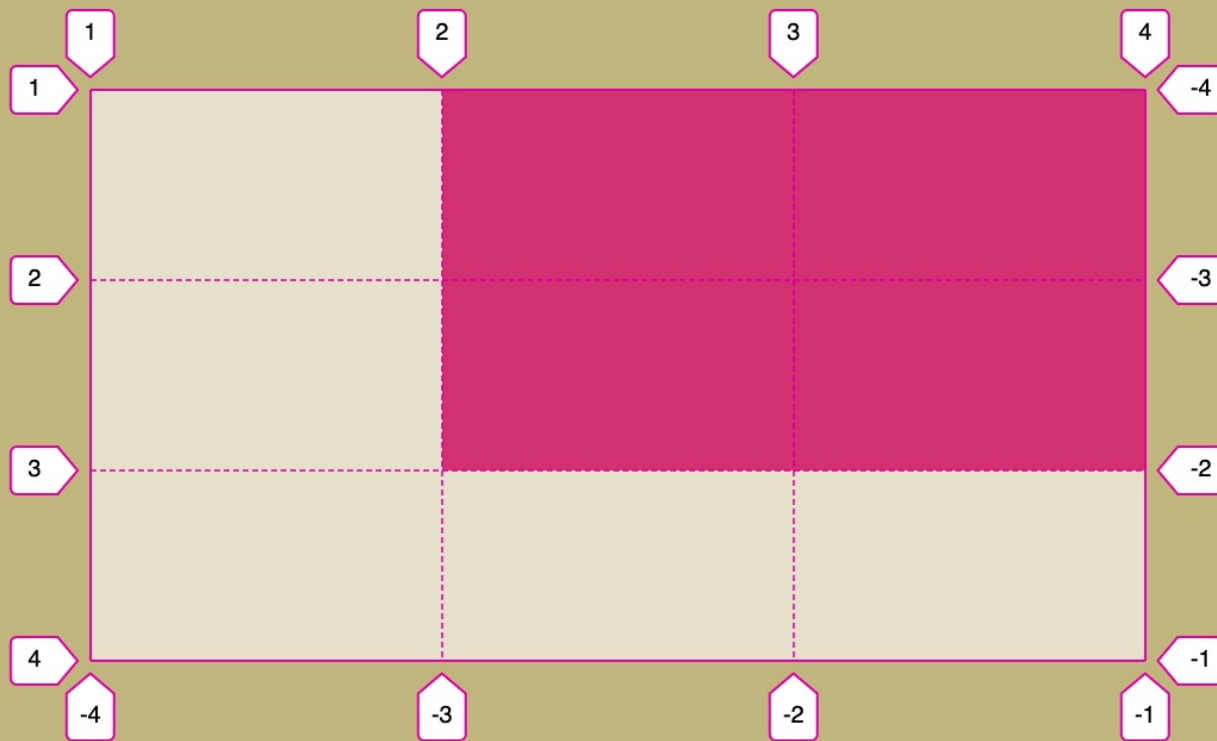


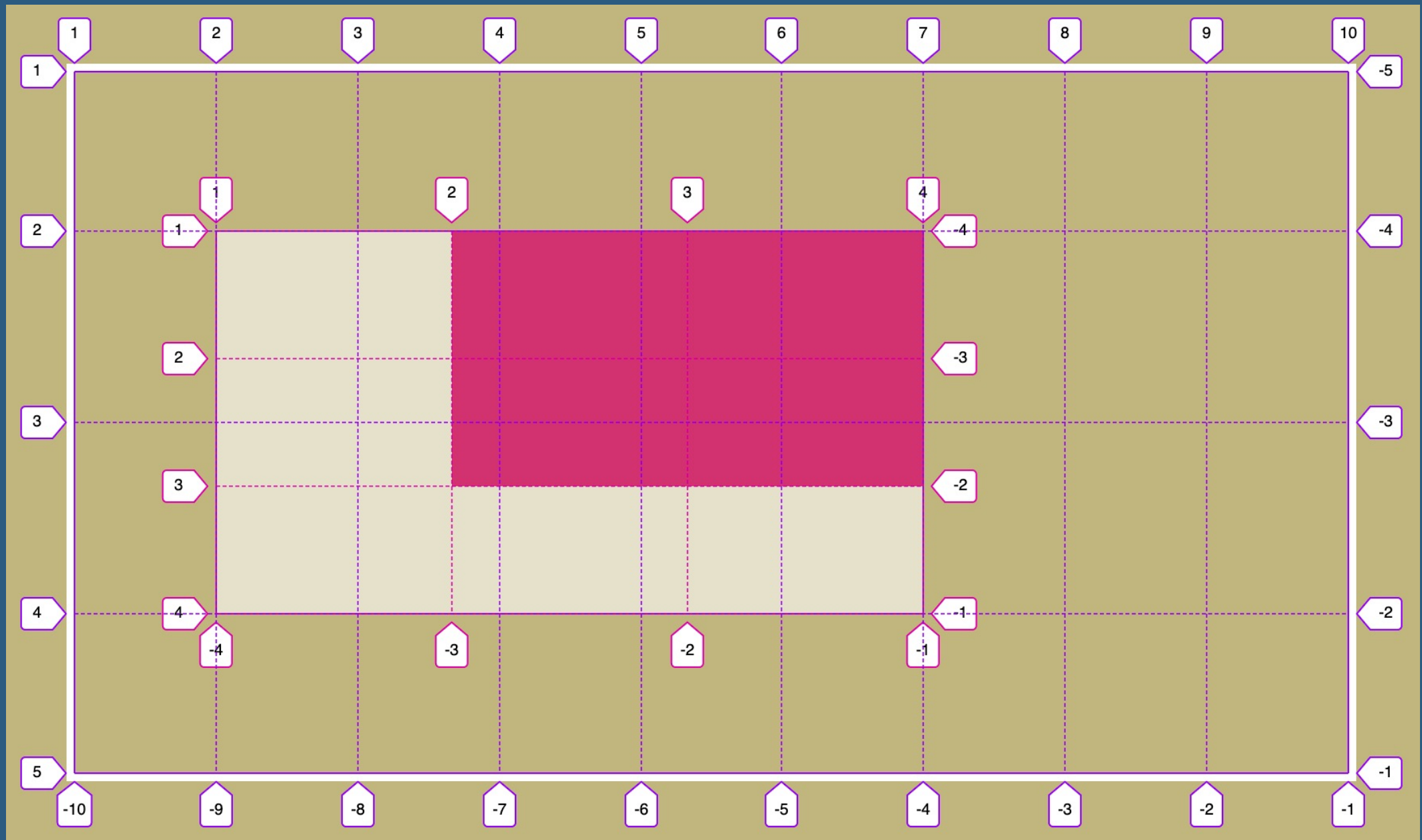
```
.grid {  
  display: grid;  
  grid-template-columns: repeat(9, 1fr);  
  grid-template-rows: repeat(4, minmax(100px,  
auto));  
}
```

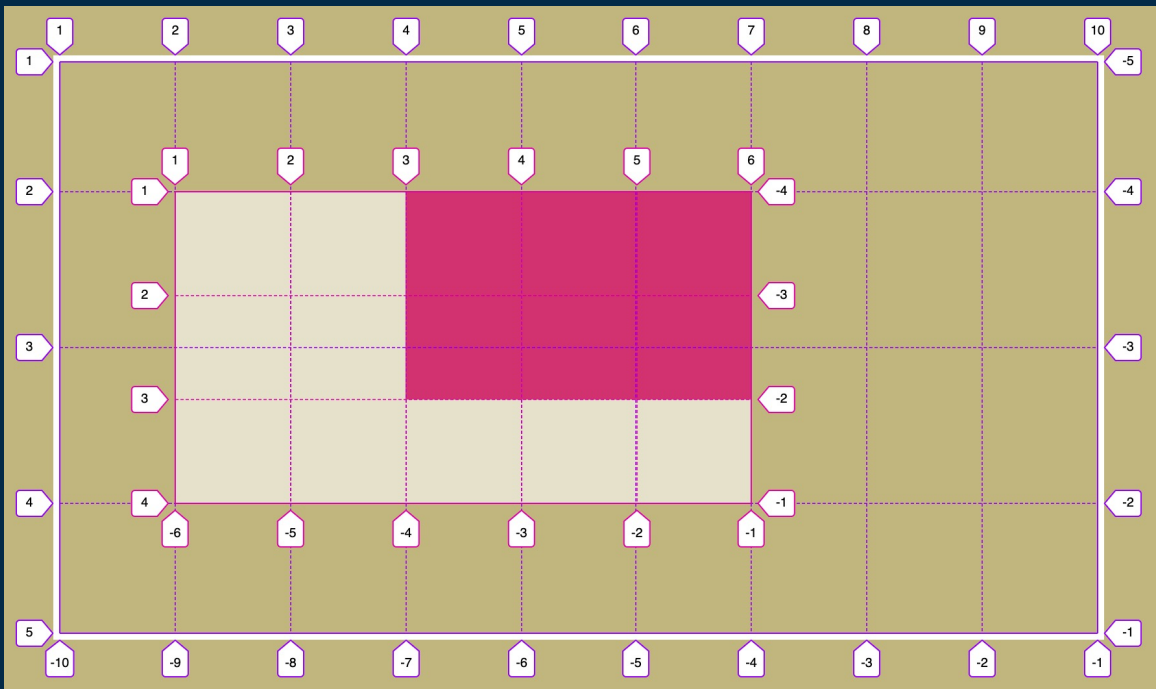
```
.item {  
  grid-column: 2 / 7;  
  grid-row: 2 / 4;  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-template-rows: repeat(3, 80px);  
}
```

```
.subitem {  
  grid-column: 2 / 4;  
  grid-row: 1 / 4;  
}
```





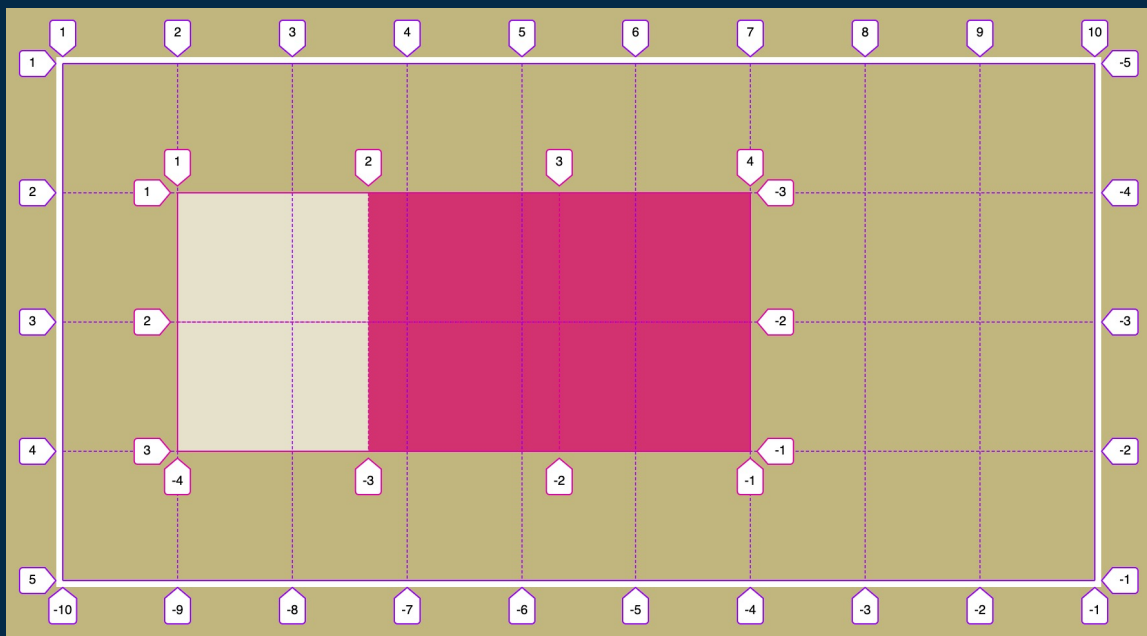




```
.grid {
  display: grid;
  grid-template-columns: repeat(9, 1fr);
  grid-template-rows: repeat(4, minmax(100px,
auto));
}
```

```
.item {
  grid-column: 2 / 7;
  grid-row: 2 / 4;
  display: grid;
  grid-template-columns: subgrid;
  grid-template-rows: repeat(3, 80px);
}
```

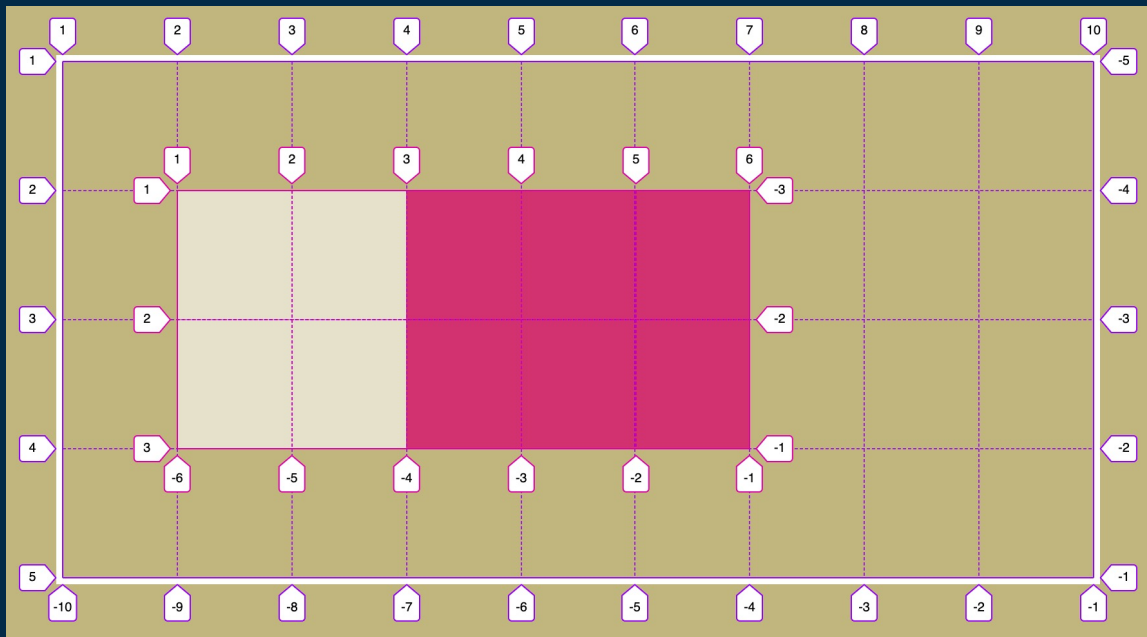
```
.subitem {
  grid-column: 3 / 6;
  grid-row: 1 / 4;
}
```



```
.grid {
  display: grid;
  grid-template-columns: repeat(9, 1fr);
  grid-template-rows: repeat(4, minmax(100px,
auto));
}
```

```
.item {
  grid-column: 2 / 7;
  grid-row: 2 / 4;
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-template-rows: subgrid;
}
```

```
.subitem {
  grid-column: 2 / 4;
  grid-row: 1 / 3;
}
```



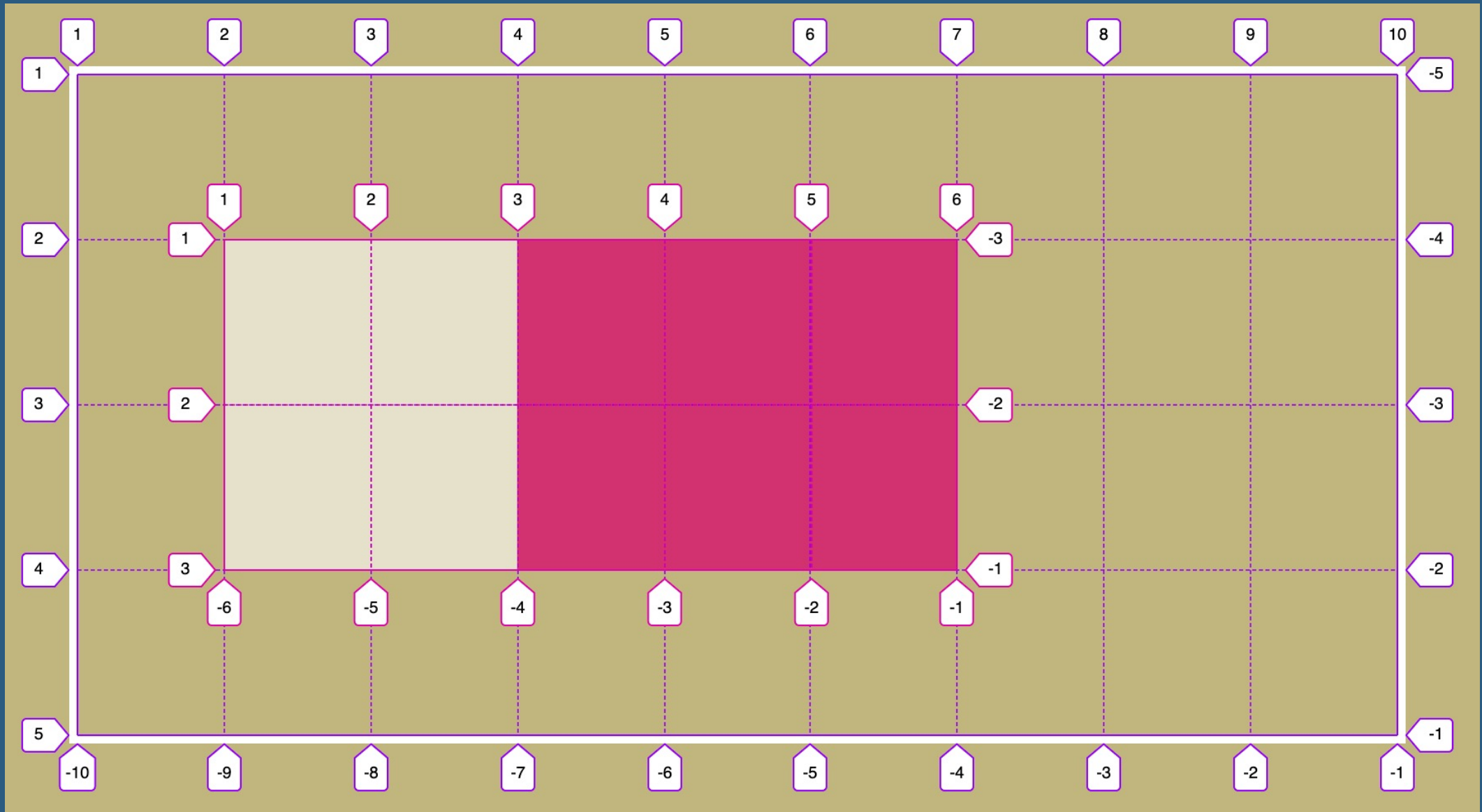
```
.grid {
  display: grid;
  grid-template-columns: repeat(9, 1fr);
  grid-template-rows: repeat(4, minmax(100px,
auto));
}
```

```
.item {
  grid-column: 2 / 7;
  grid-row: 2 / 4;
  display: grid;
  grid-template-columns: subgrid;
  grid-template-rows: subgrid;
}
```

```
.subitem {
  grid-column: 3 / 6;
  grid-row: 1 / 3;
}
```

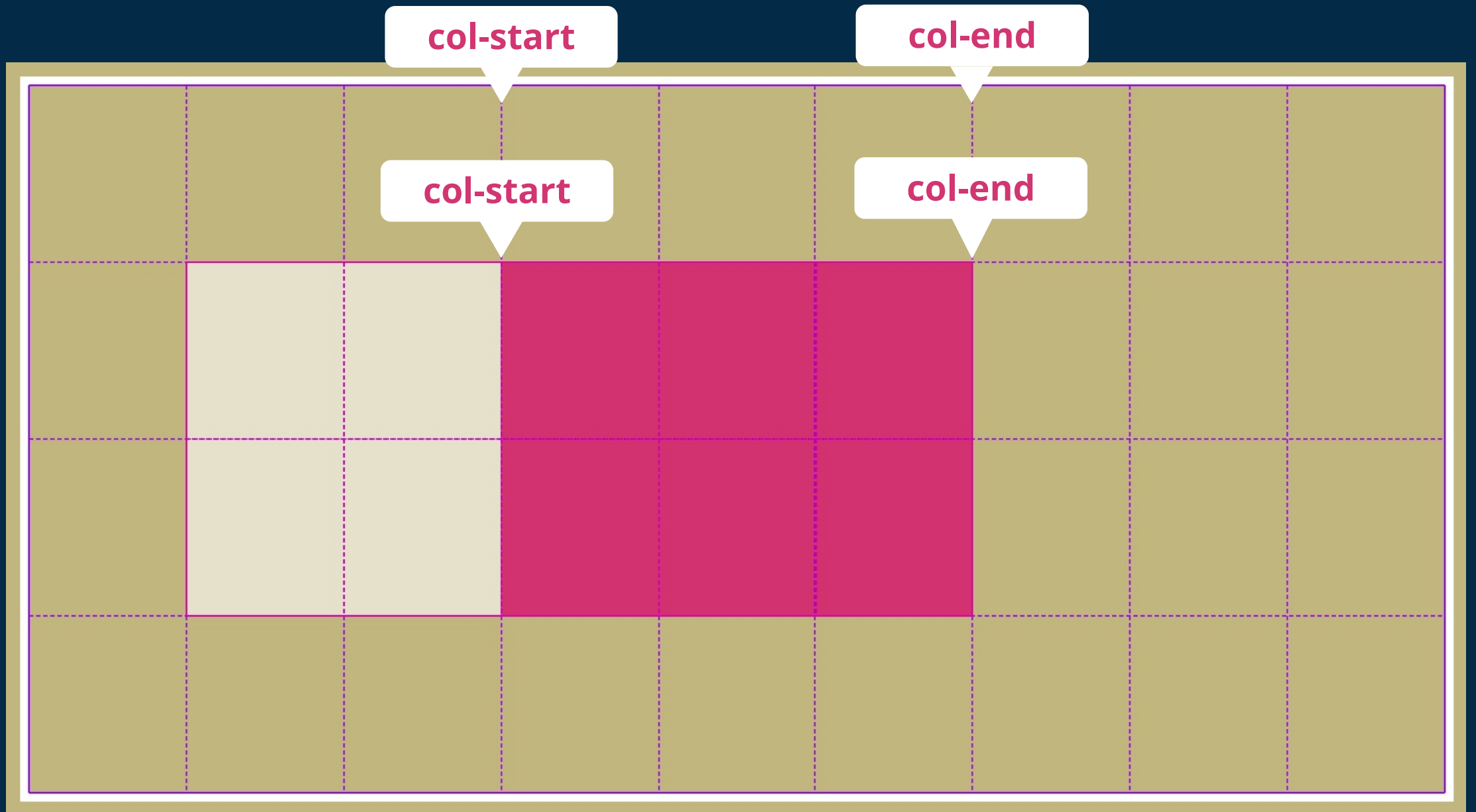

Line **numbers** start from 1 inside the subgrid.

You position child items in the subgrid according to the subgrid line numbering, not those of the parent.



Line **names** on the parent are passed into the subgrid.

If you have named lines on the parent grid they will be passed into the subgrid and added to any names defined there.



```
.grid {  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr [col-start] 1fr 1fr 1fr [col-end] 1fr 1fr 1fr;  
  grid-template-rows: repeat(4, minmax(100px, auto));  
}
```

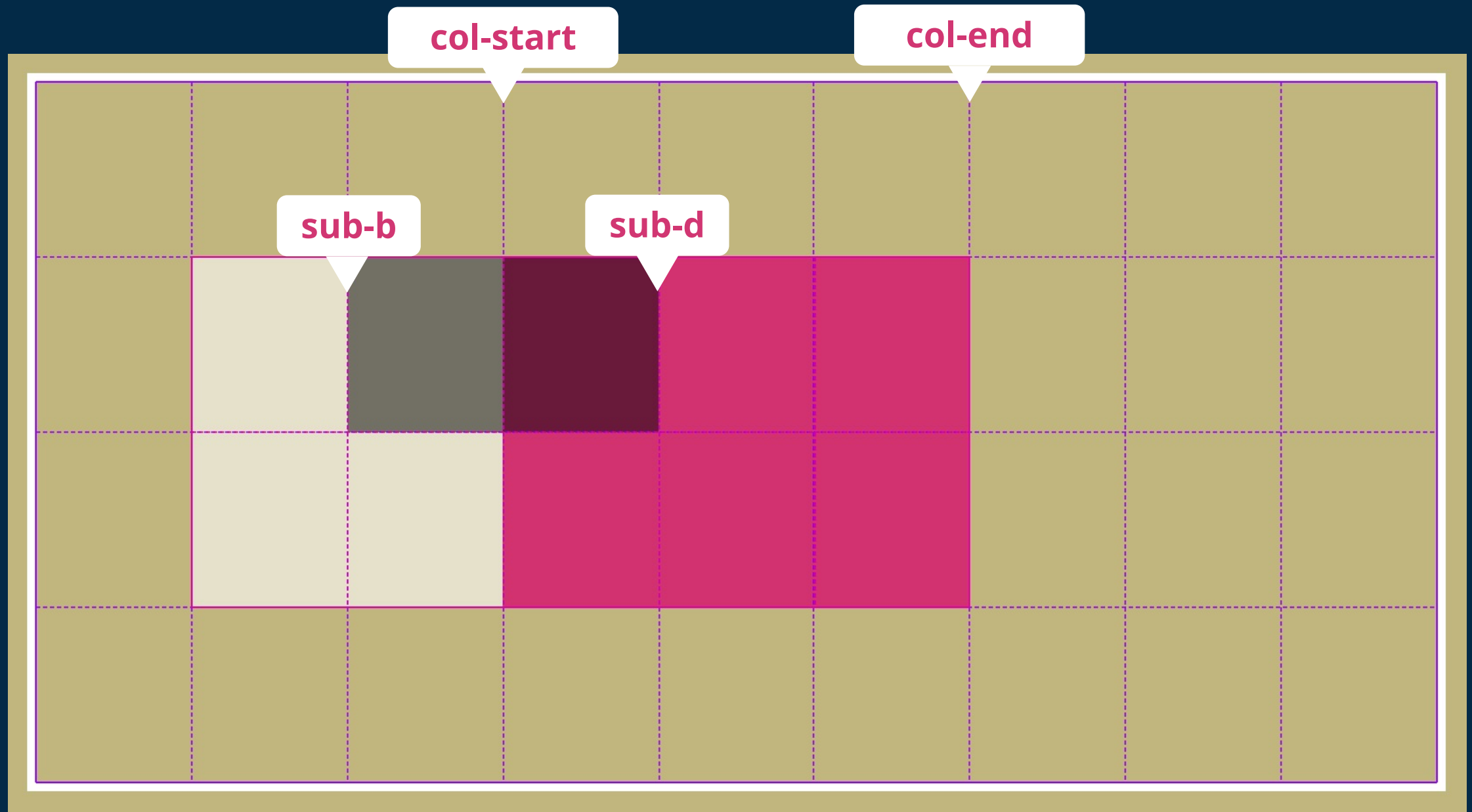
```
.item {  
  grid-column: 2 / 7;  
  grid-row: 2 / 4;  
  display: grid;  
  grid-template-columns: subgrid;  
  grid-template-rows: subgrid;  
}
```

```
.subitem {  
  grid-column: col-start / col-end;  
  grid-row: 1 / 3;  
}
```

You can **add** named lines to the subgrid.

Line names are added after the subgrid keyword.

```
grid-template-columns: subgrid [sub-a] [sub-b];
```



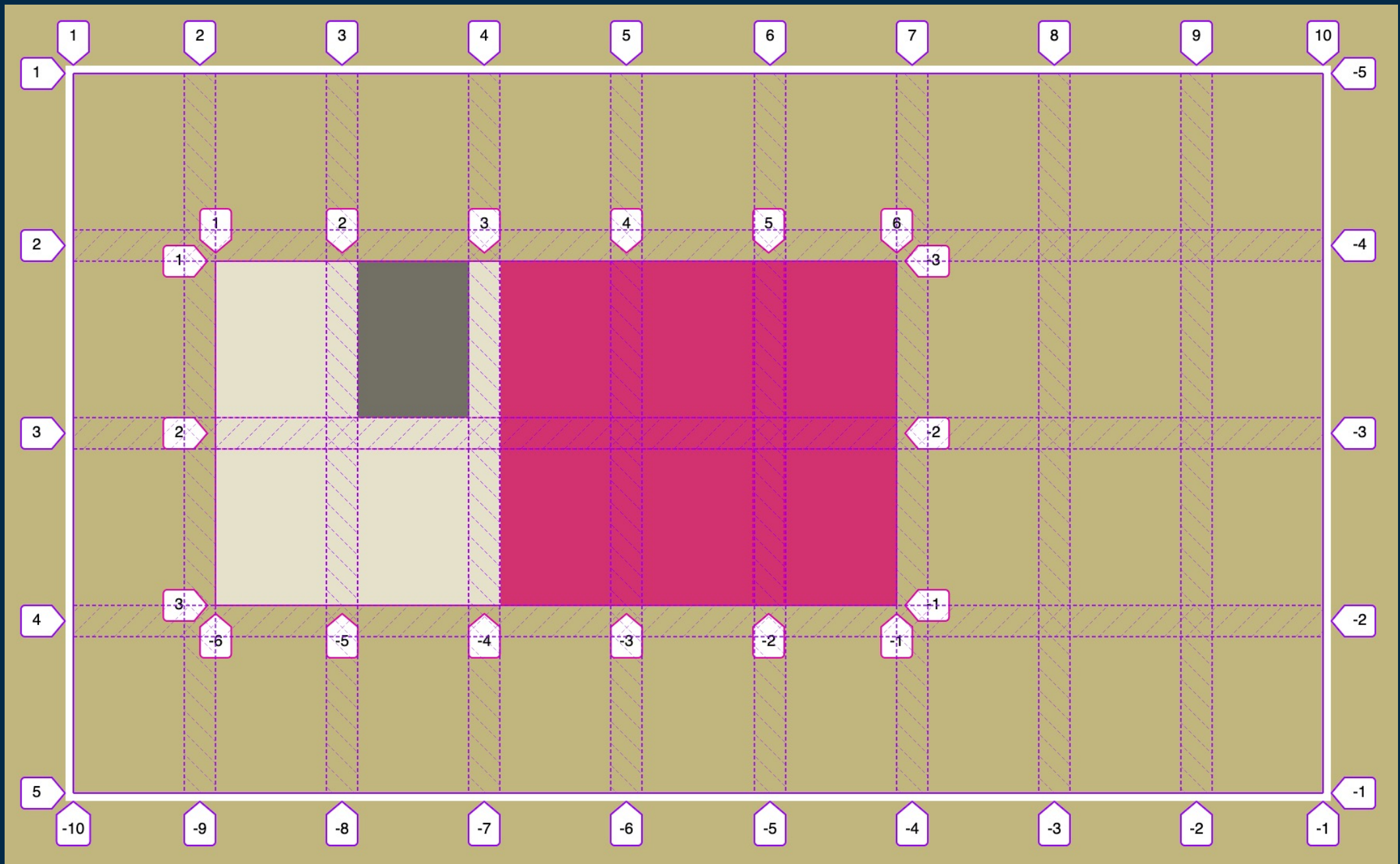

```
.grid {  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr [col-start] 1fr 1fr 1fr [col-end] 1fr 1fr 1fr;  
  grid-template-rows: repeat(4, minmax(100px, auto));  
}
```

```
.item {  
  grid-column: 2 / 7;  
  grid-row: 2 / 4;  
  display: grid;  
  grid-template-columns: subgrid [sub-a] [sub-b] [sub-c] [sub-d] [sub-e] [sub-f];  
  grid-template-rows: subgrid;  
}
```

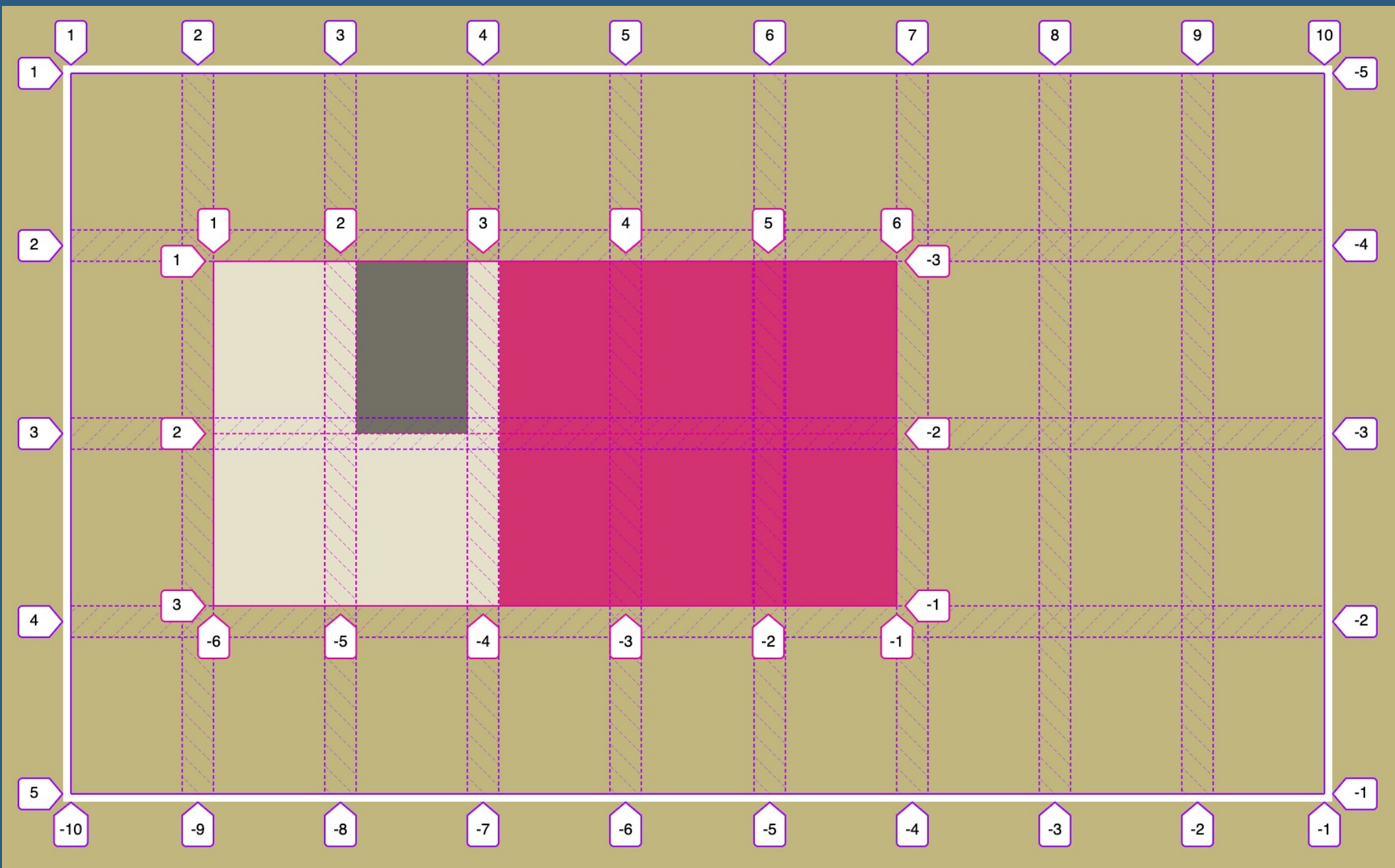
```
.subitem {  
  grid-column: col-start / col-end;  
  grid-row: 1 / 3;  
}
```

```
.subitem2 {  
  grid-column: sub-b / sub-d;  
  grid-row: 1 / 3;  
}
```

The subgrid inherits the
gaps from the parent.



You can **change** the gaps
on the subgrid.



Sizing of items in the subgrid can change the size of the parent tracks.

Card One

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Two

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer. This footer has much more content than the designer expected.

Card Three

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Four

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Five

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

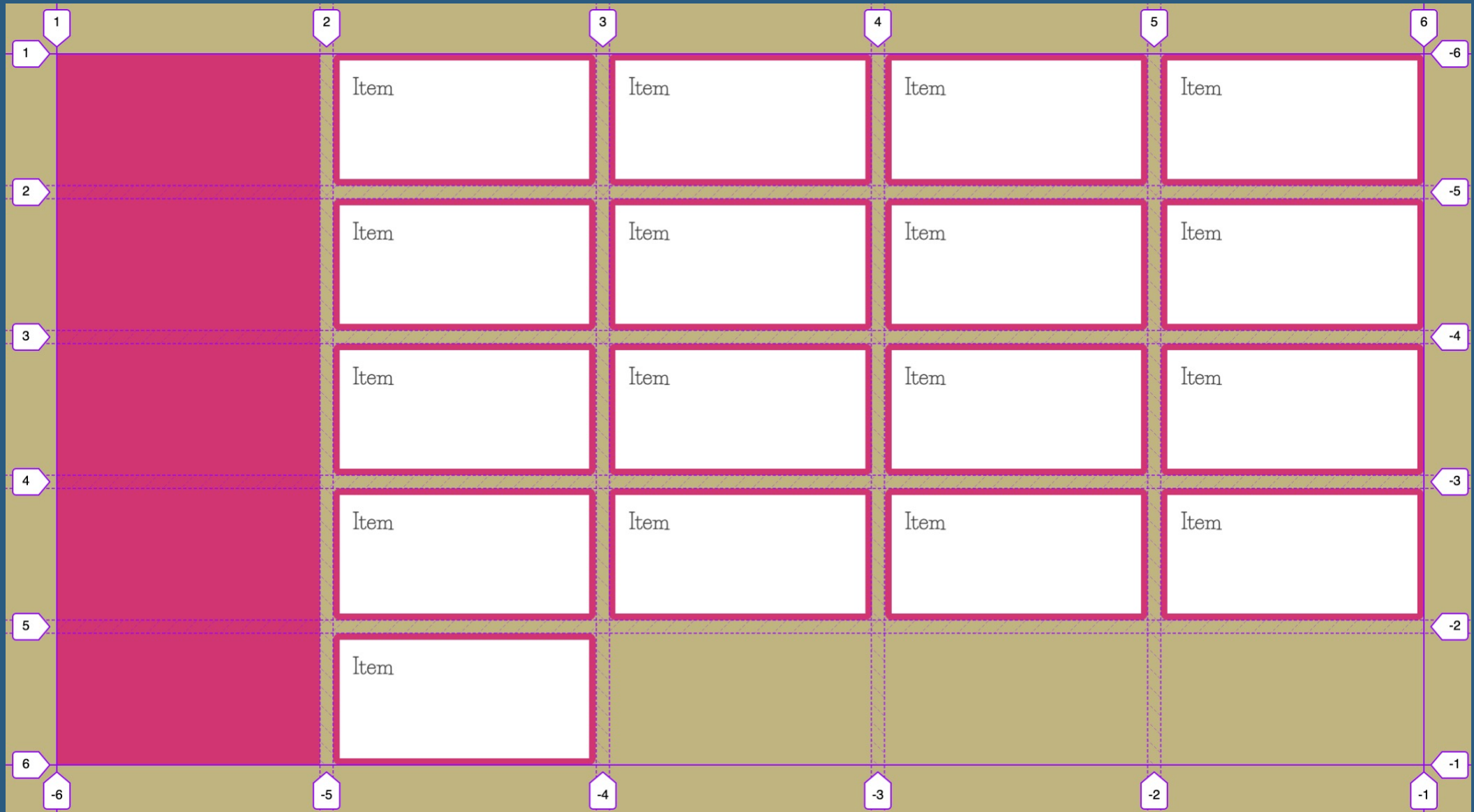
This is the footer.

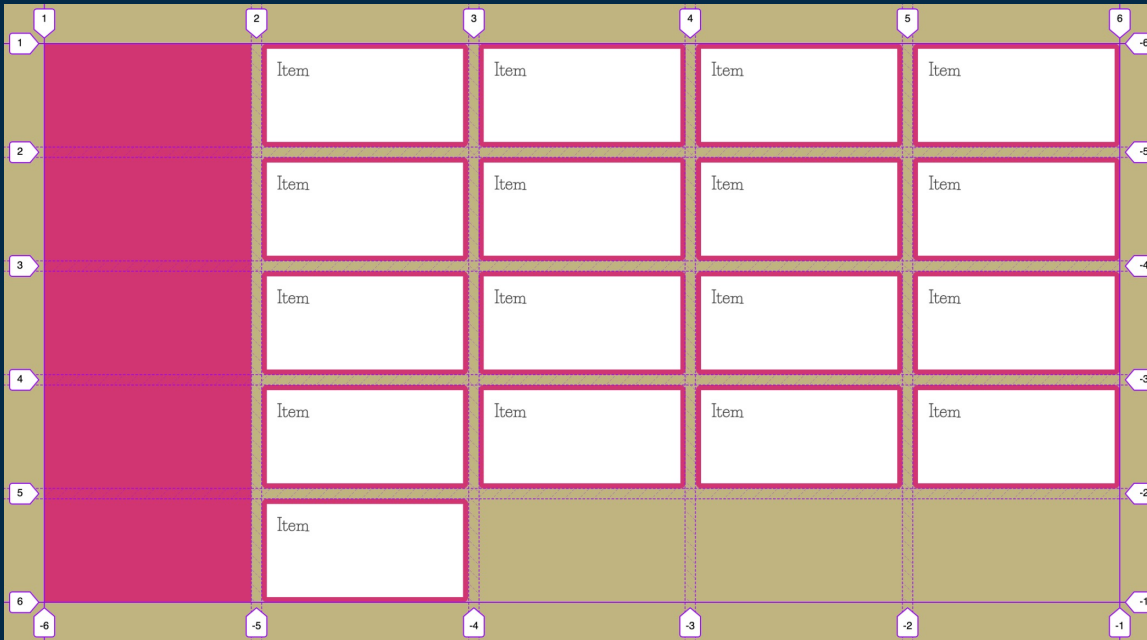
Card Six

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Subgrid enables some
previously **difficult patterns.**





```
.wrapper {  
  display: grid;  
  gap: 10px;  
  grid-template-columns: repeat(5, 1fr);  
  /* 5 explicit rows */  
  grid-template-rows:  
    repeat(5, minmax(100px, auto));  
}  
  
.fullheight {  
  background-color: rgb(209,54,114);  
  grid-row: 1 / -1;  
}
```

Line -1 is the end line of
the **explicit grid**.

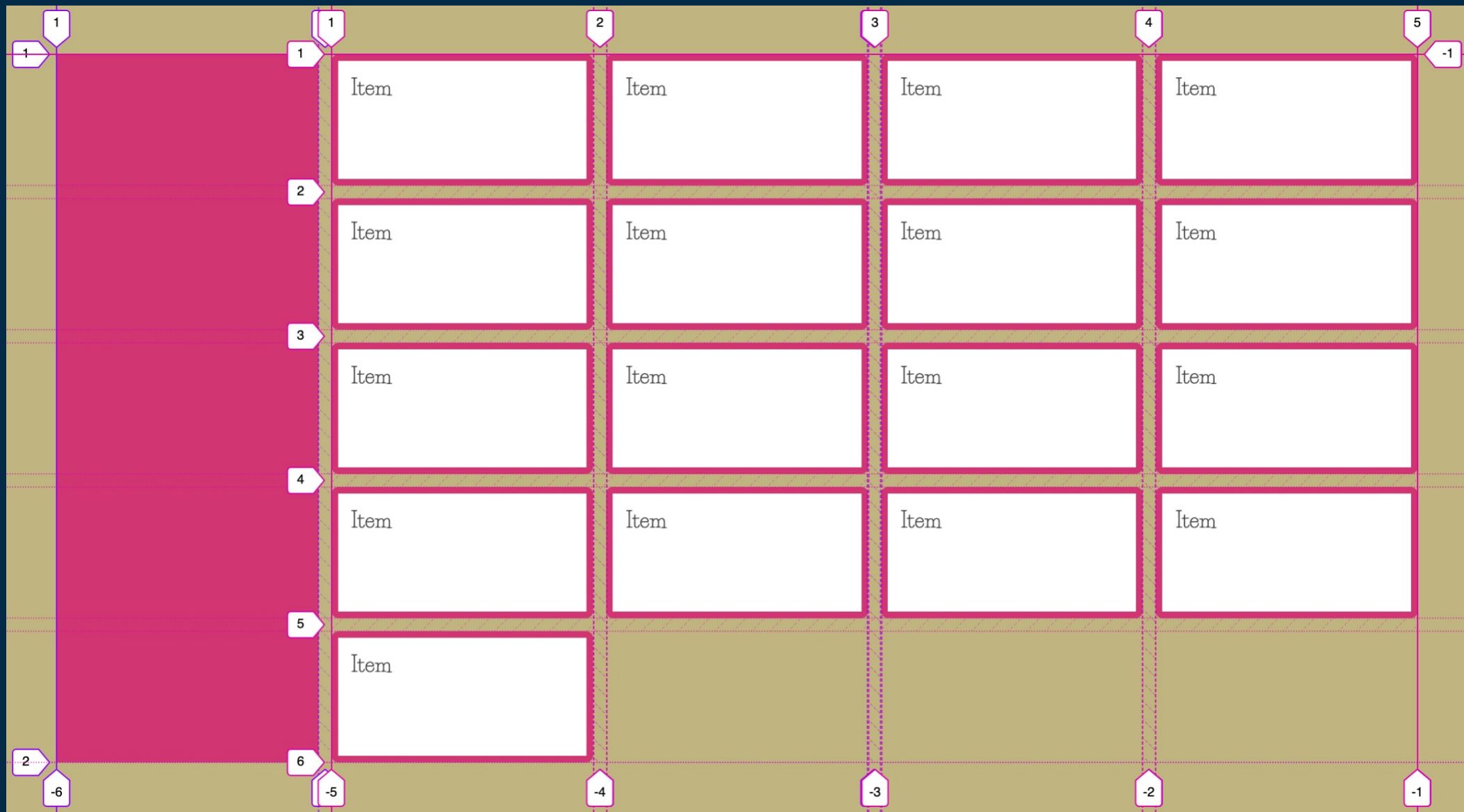


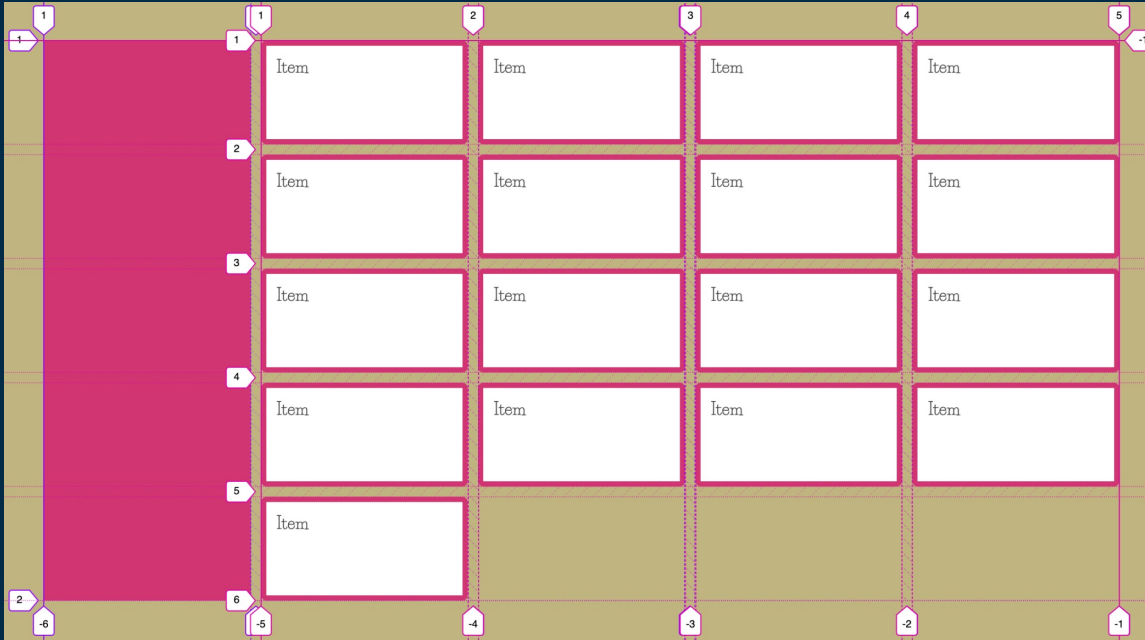


```
.wrapper {  
  display: grid;  
  gap: 10px;  
  grid-template-columns: repeat(5, 1fr);  
  /* no defined explicit rows */  
  grid-auto-rows: minmax(100px, auto);  
}  
  
.fullheight {  
  background-color: rgb(209,54,114);  
  grid-row: 1 / -1;  
}
```

We can't target the end
line of the **implicit grid**.

Place the items in a
container which uses a
subgrid for columns.





```
.wrapper {  
  display: grid;  
  gap: 10px;  
  grid-template-columns: repeat(5, 1fr);  
  /* no defined explicit rows */  
  grid-auto-rows: minmax(100px, auto);  
}  
.items {  
  grid-column: 2 / -1;  
  display: grid;  
  grid-template-columns: subgrid;  
  grid-auto-rows: minmax(100px, auto);  
}  
.fullheight {  
  background-color: rgb(209,54,114);  
  grid-row: 1 / -1;  
}
```

Content

Subitem

Content



```
.grid {  
  display: grid;  
  grid-template-columns:  
    [full-start] minmax(1em, 1fr)  
    [main-start] minmax(0, 40em) [main-end]  
    minmax(1em, 1fr) [full-end];  
}  
  
.grid > * {  
  grid-column: main ;  
}  
  
.grid > .full {  
  grid-column: full;  
  display: grid;  
  grid-template-columns: subgrid;  
}  
  
.center {  
  grid-column: main;  
}
```

The future

Container Queries and subgrid?



A nice picture of a balloon.



A nice picture of a
balloon.

```
@container grid (min-width: 600px) {  
  .item {  
    display: grid;  
    grid-template-columns: subgrid;  
  }  
}
```

Thank you

<https://noti.st/rachelandrew/pFcmZW/>