

New Tools for CSS Layout

Rachel Andrew
@rachelandrew

I do web stuff.

Web developer since 1996, teaching CSS for almost as long. *Co-founder Perch CMS & Notist.* Author or co-author of 23 books on web development. *Smashing Magazine Editor in Chief.* Writer for MDN. *CSS Working Group Invited Expert.* Google Developer Expert. I like long runs and small airplanes.

A layout system for CSS

For the first time!

A layout system for CSS

Flexbox

Grid Layout

Multiple-column Layout

Floats

CSS Positioned Layout

Brought together by ...

Box Alignment

Intrinsic and Extrinsic Sizing

Logical Properties and Values

Flexbox

Layout in one dimension. A row or a column.

Small

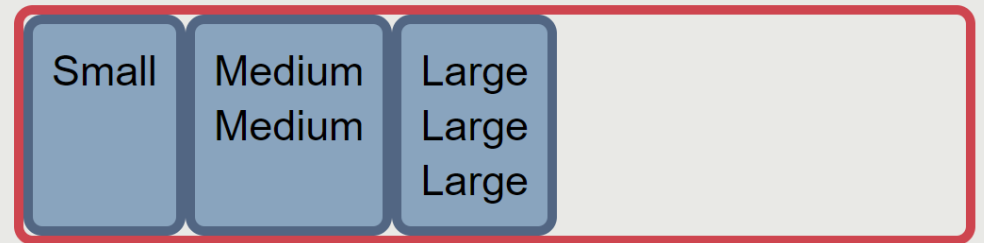
Medium
Medium

Large
Large
Large

Inline Dimension



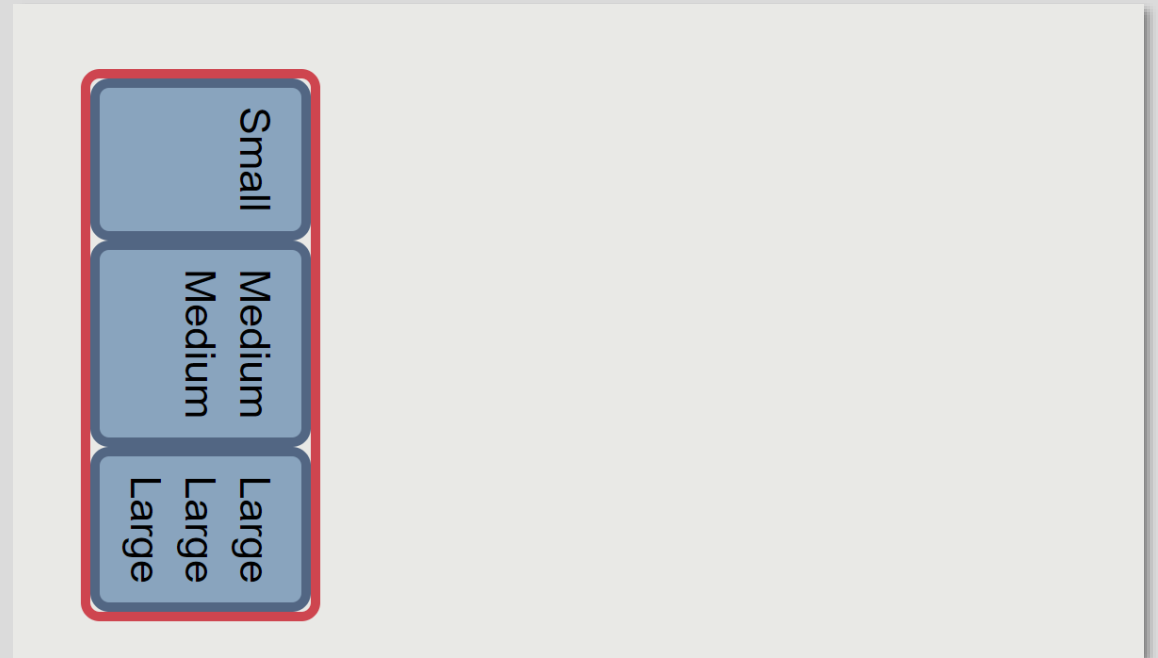
```
.flex {  
  display: flex;  
  width: 500px;  
}
```





```
.flex {  
  writing-mode: vertical-rl;  
  display: flex;  
  width: 500px;  
}
```

Block Dimension

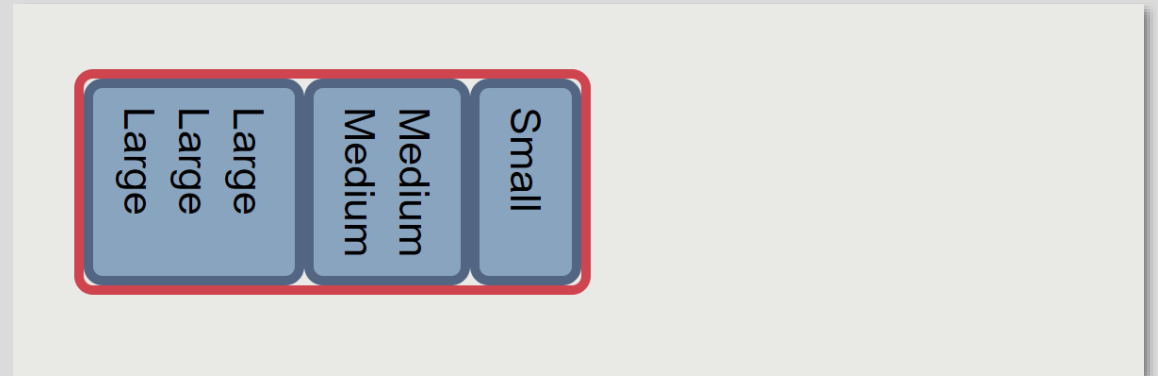


```
.flex {  
  writing-mode: vertical-rl;  
  display: flex;  
  width: 500px;  
}
```



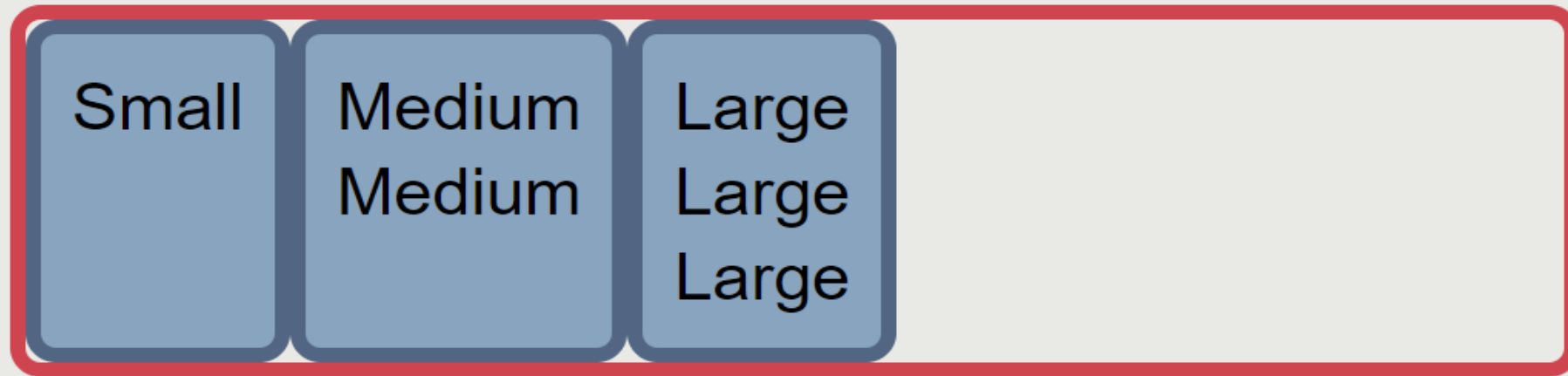
```
.flex {  
  writing-mode: vertical-rl;  
  flex-direction: column;  
  display: flex;  
}
```

Block Dimension

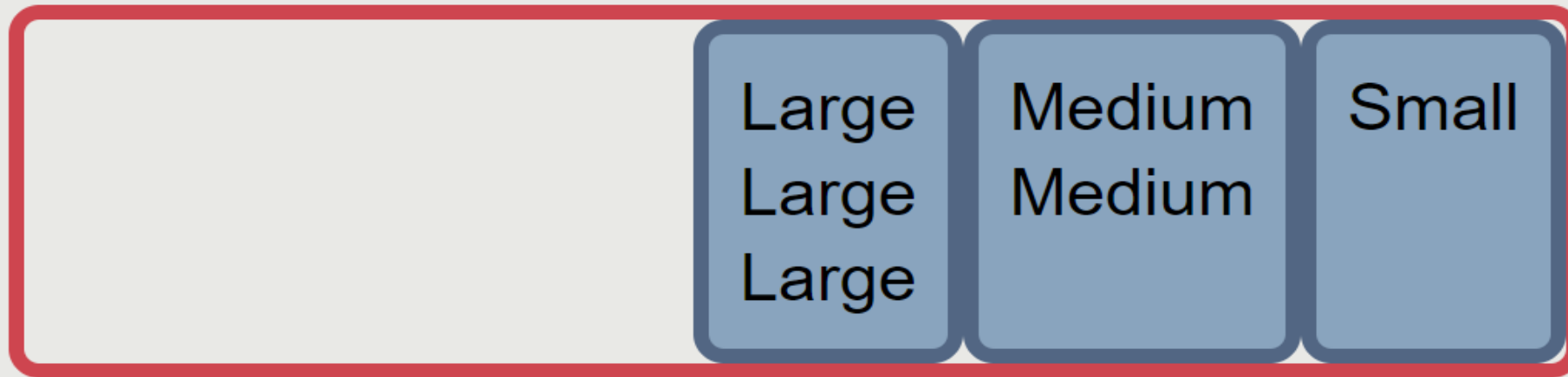


```
.flex {  
  writing-mode: vertical-rl;  
  flex-direction: column;  
  display: flex;  
}
```

Start Left to Right



Start Right to Left

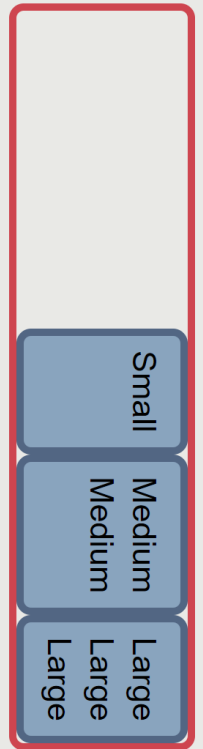
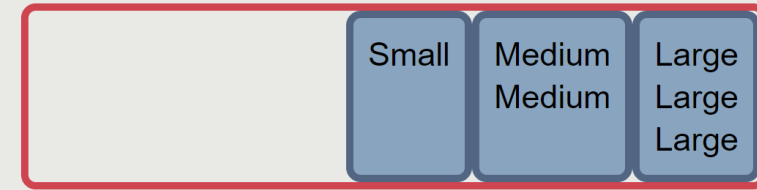




```
.flex {  
  display: flex;  
  width: 500px;  
  justify-content: flex-end;  
}  
  
.flex.vertical {  
  writing-mode: vertical-rl;  
  width: auto;  
  inline-size: 500px;  
}
```

Inline

Block

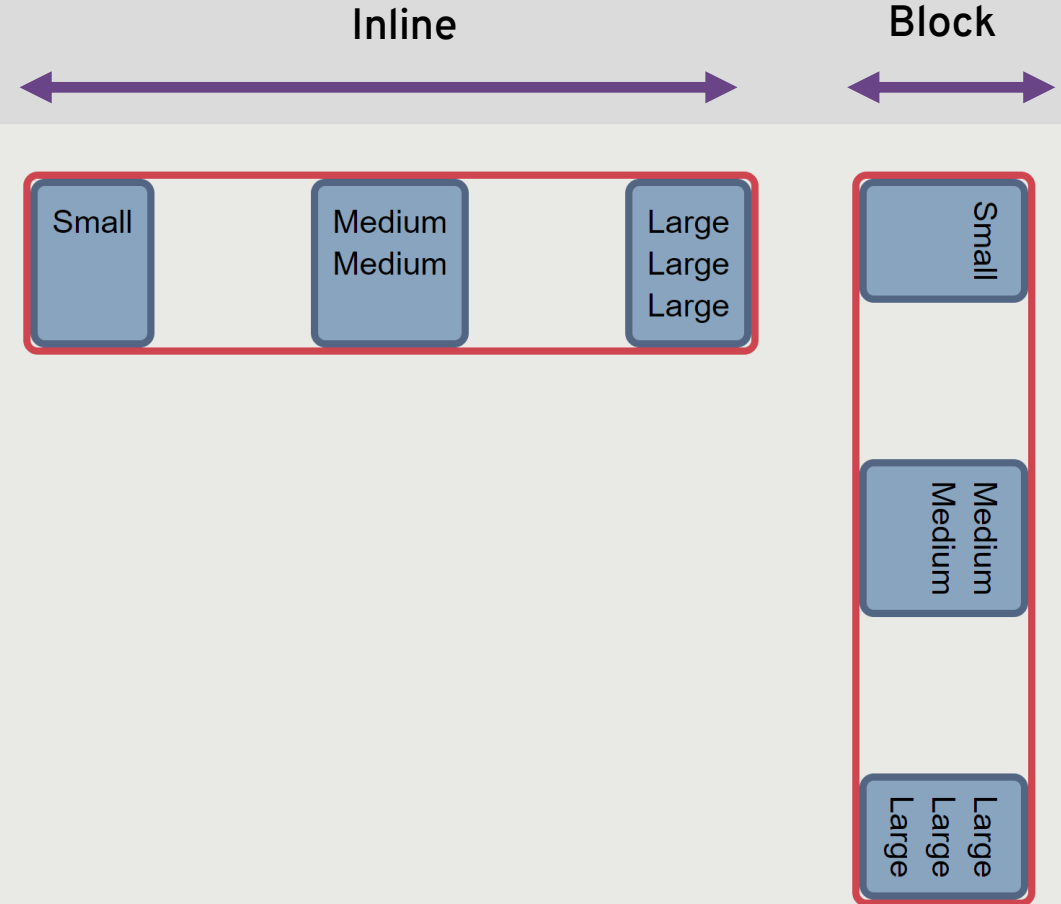


```
.flex {  
  display: flex;  
  width: 500px;  
  justify-content: flex-end;  
}  
  
.flex.vertical {  
  writing-mode: vertical-rl;  
  width: auto;  
  inline-size: 500px;  
}
```




```
.flex {  
  display: flex;  
  width: 500px;  
  justify-content: space-between;  
}  
  
.flex.vertical {  
  writing-mode: vertical-rl;  
  width: auto;  
  inline-size: 500px;  
}
```

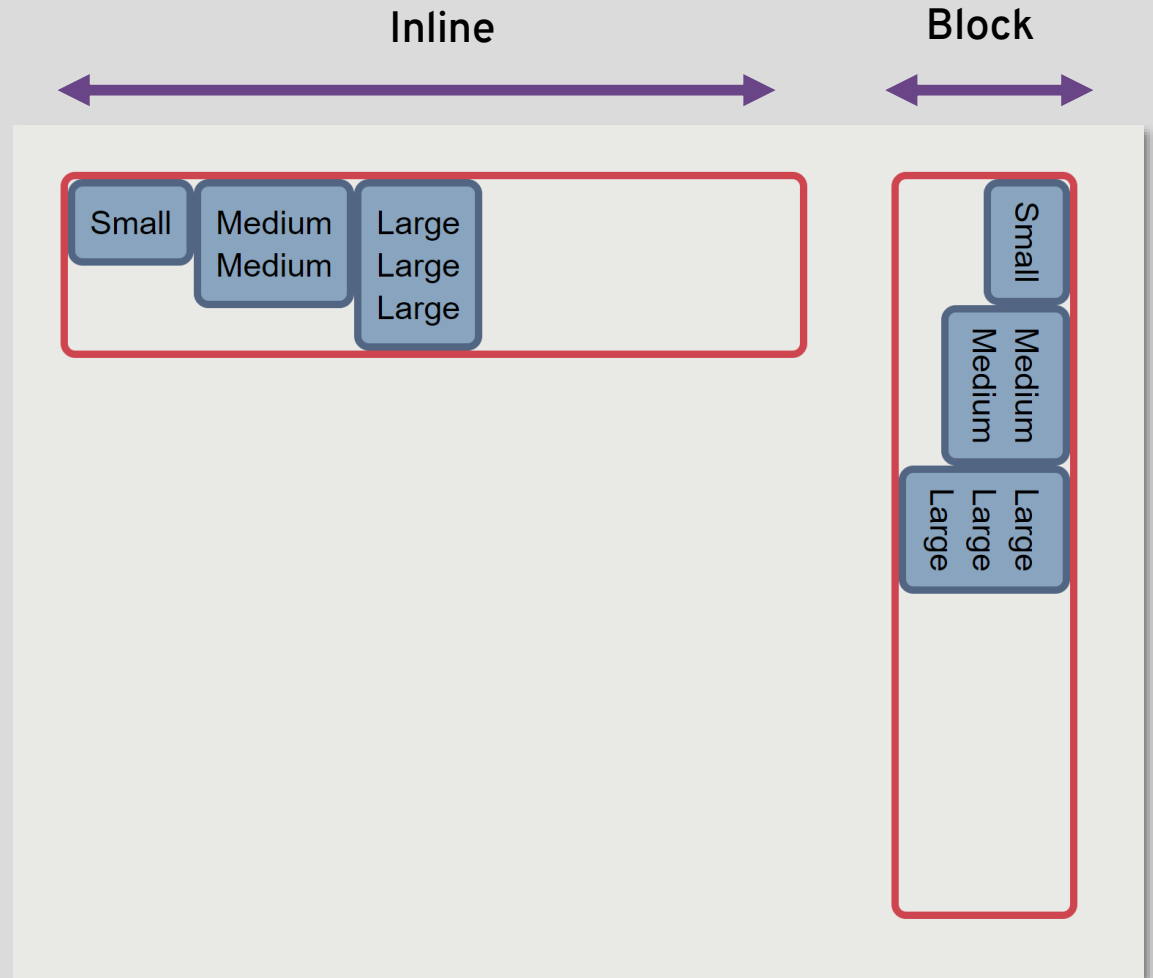
```
.flex {  
  display: flex;  
  width: 500px;  
  justify-content: space-between;  
}  
  
.flex.vertical {  
  writing-mode: vertical-rl;  
  width: auto;  
  inline-size: 500px;  
}
```





```
.flex {  
  display: flex;  
  align-items: flex-start;  
}  
  
.flex.vertical {  
  writing-mode: vertical-rl;  
}
```

```
.flex {  
  display: flex;  
  align-items: flex-start;  
}  
  
.flex.vertical {  
  writing-mode: vertical-rl;  
}
```





```
.flex {  
  display: flex;  
  align-items: center;  
  justify-content: center;  
}
```

```
.flex {  
  display: flex;  
  align-items: center;  
  justify-content: center;  
}
```



CSS Grid Layout

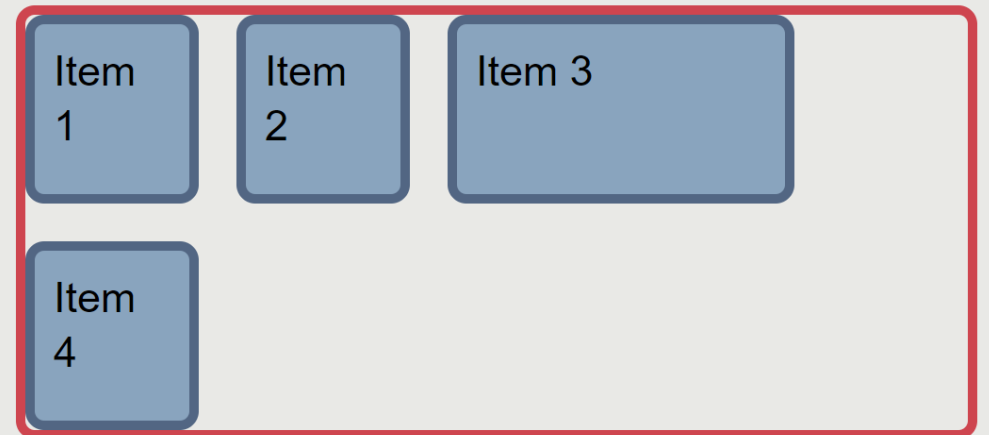
Layout in two dimensions. Rows and Columns.

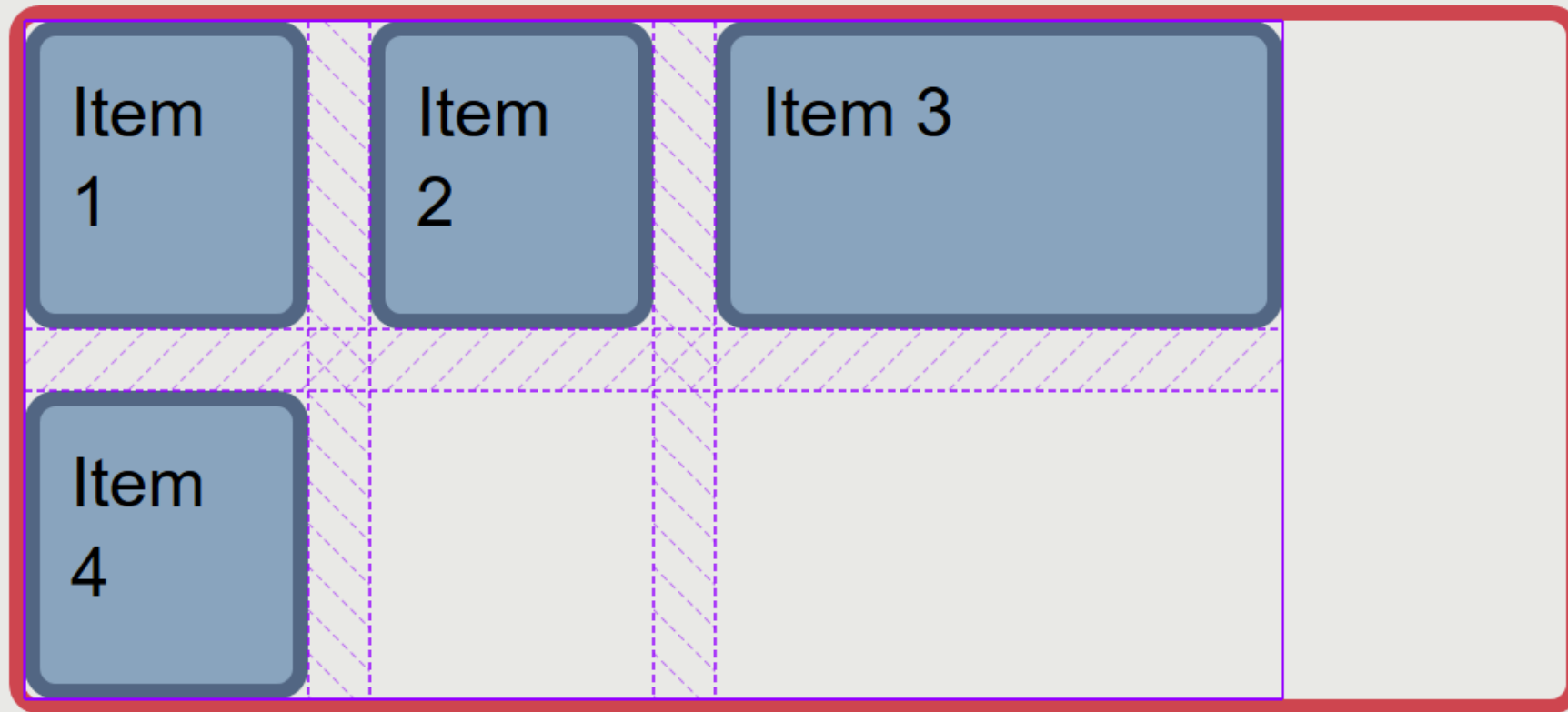


```
.grid {  
  display: grid;  
  width: 500px;  
  grid-template-columns: .2fr .2fr .4fr;  
  grid-template-rows: 100px 100px;  
  grid-gap: 20px;  
}
```



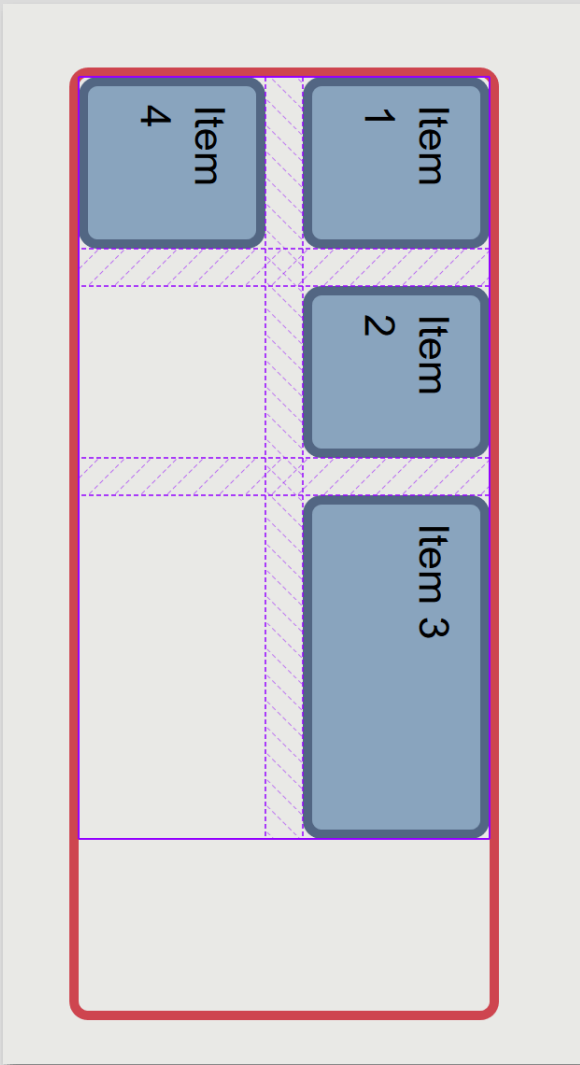
```
.grid {  
  display: grid;  
  width: 500px;  
  grid-template-columns: .2fr .2fr .4fr;  
  grid-template-rows: 100px 100px;  
  grid-gap: 20px;  
}
```





Firefox Grid Inspector

https://developer.mozilla.org/en-US/docs/Tools/Page_Inspector/How_to/Examine_grid_layouts



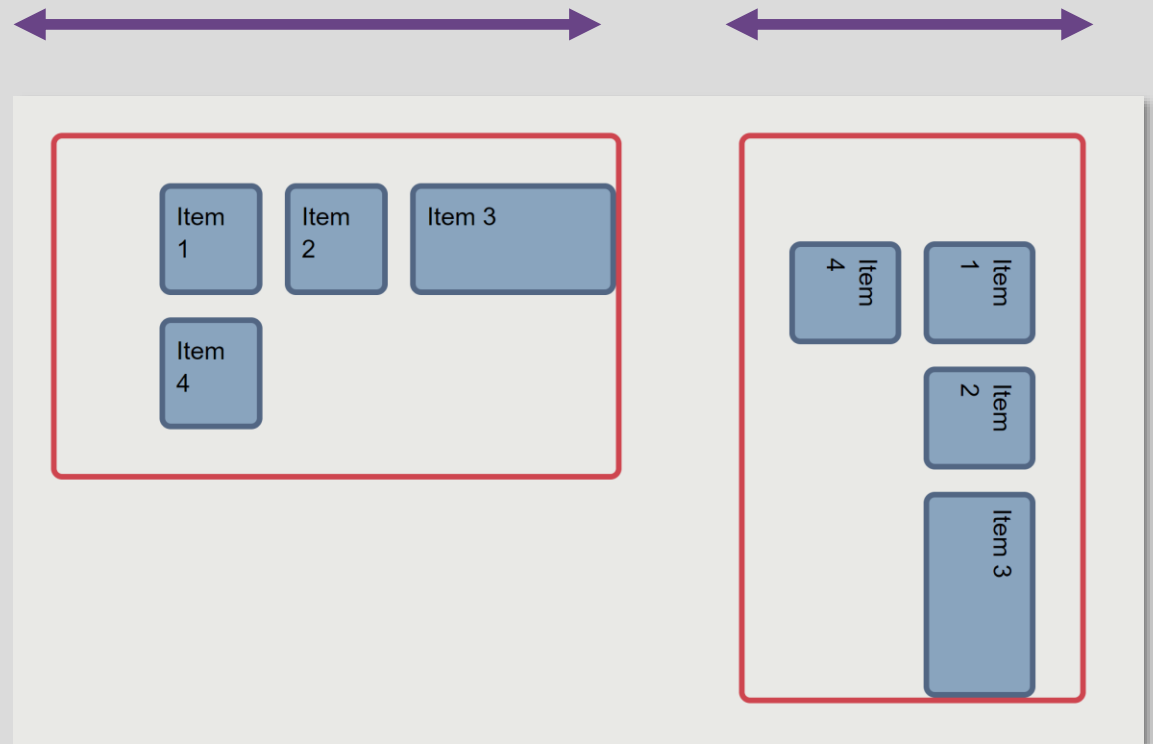


```
.grid {  
  display: grid;  
  width: 500px;  
  height: 300px;  
  grid-template-columns: 1fr 1fr 2fr;  
  grid-template-rows: 100px 100px;  
  grid-gap: 20px;  
  justify-content: end;  
  align-content: center;  
}  
  
.grid.vertical {  
  writing-mode: vertical-rl;  
  width: auto;  
  inline-size: 500px;  
  block-size: 300px;  
}
```

```
.grid {  
  display: grid;  
  width: 500px;  
  height: 300px;  
  grid-template-columns: 1fr 1fr 2fr;  
  grid-template-rows: 100px 100px;  
  grid-gap: 20px;  
  justify-content: end;  
  align-content: center;  
}  
  
.grid.vertical {  
  writing-mode: vertical-rl;  
  width: auto;  
  inline-size: 500px;  
  block-size: 300px;  
}
```

Inline

Block



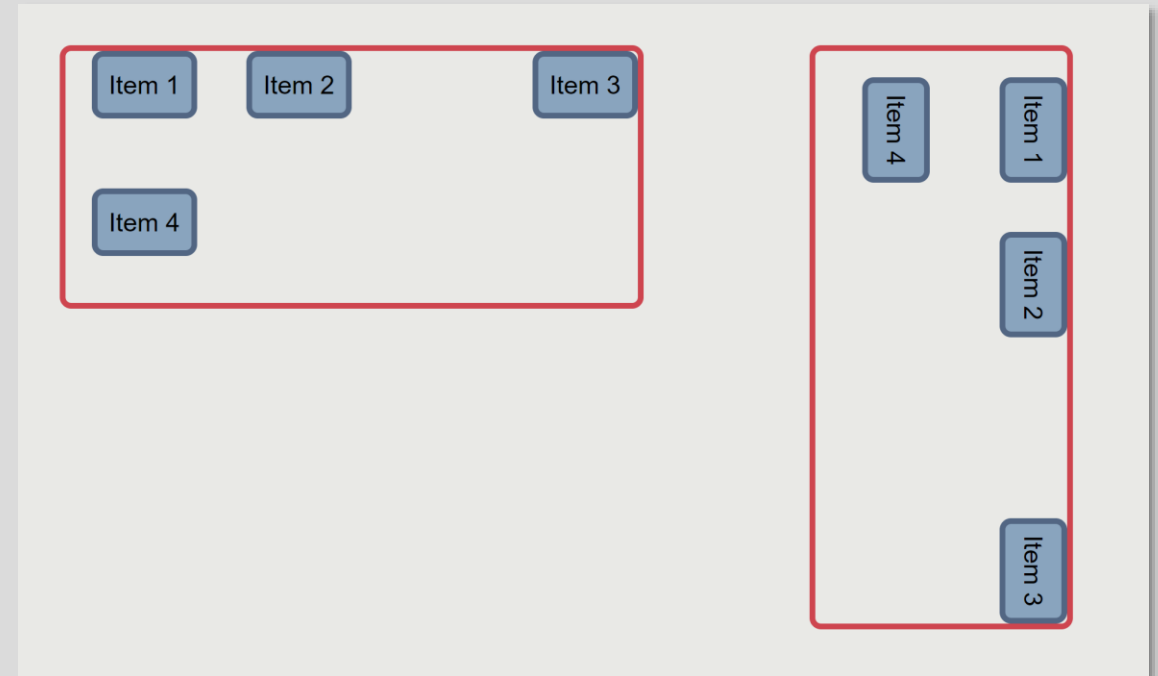


```
.grid {  
  display: grid;  
  width: 500px;  
  grid-template-columns: 1fr 1fr 2fr;  
  grid-template-rows: 100px 100px;  
  grid-gap: 20px;  
  align-items: start;  
  justify-items: end;  
}  
  
.grid.vertical {  
  writing-mode: vertical-rl;  
  width: auto;  
  inline-size: 500px;  
}
```

```
.grid {  
  display: grid;  
  width: 500px;  
  grid-template-columns: 1fr 1fr 2fr;  
  grid-template-rows: 100px 100px;  
  grid-gap: 20px;  
  align-items: start;  
  justify-items: end;  
}  
  
.grid.vertical {  
  writing-mode: vertical-rl;  
  width: auto;  
  inline-size: 500px;  
}
```

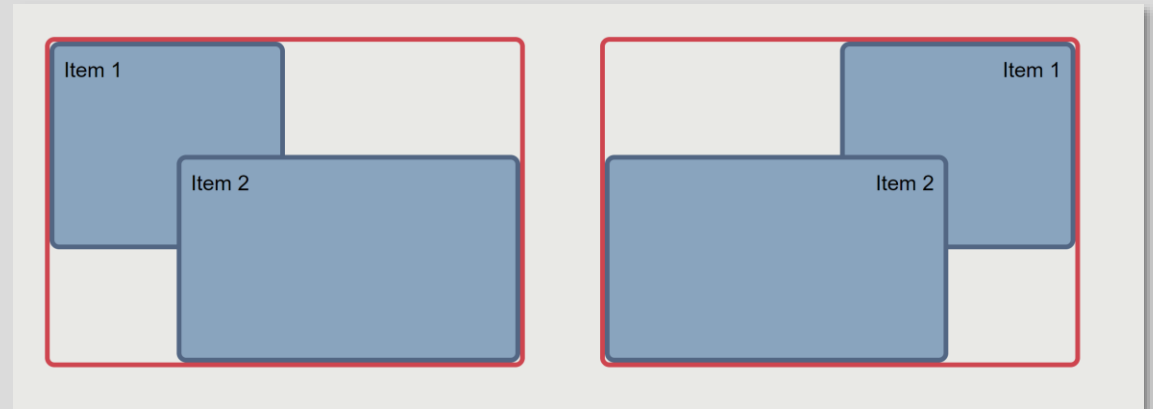
Inline

Block

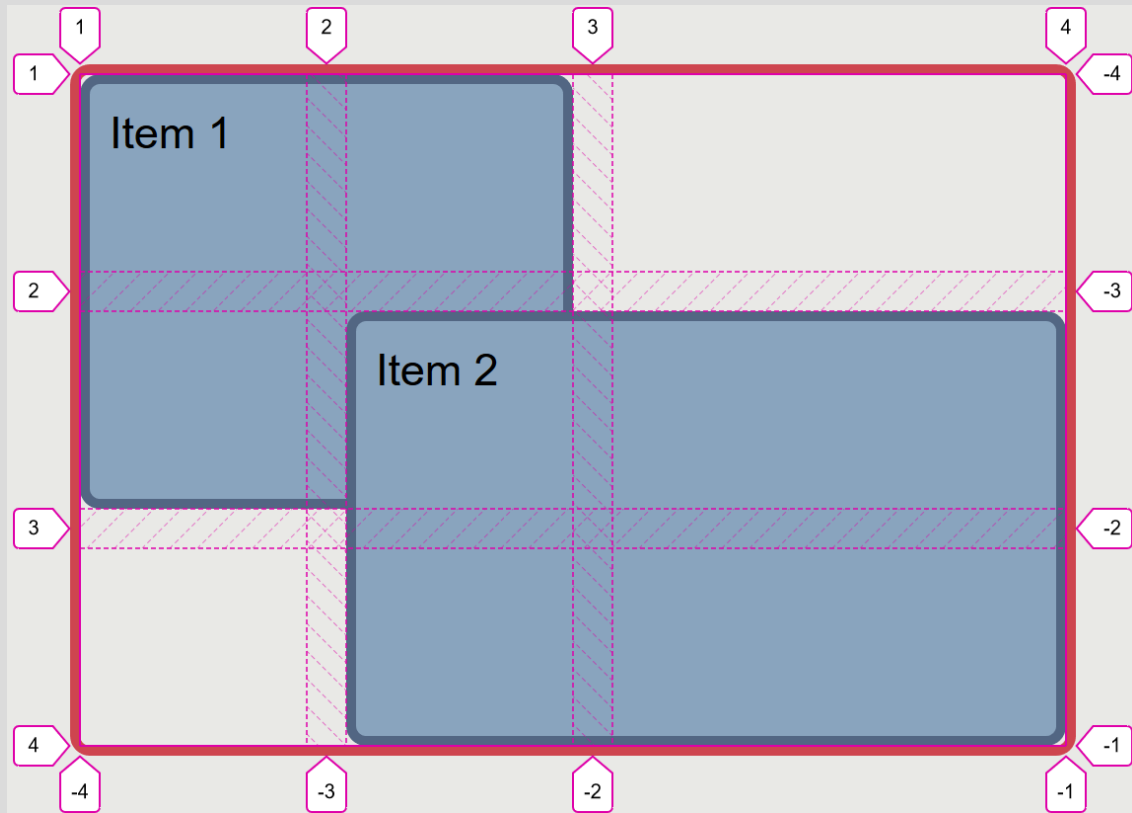



```
.grid {  
  display: grid;  
  width: 500px;  
  grid-template-columns: 1fr 1fr 2fr;  
  grid-template-rows: 100px 100px 100px;  
  grid-gap: 20px;  
}  
  
.grid-rtl { direction: rtl; }  
  
.grid div:nth-child(1) {  
  grid-column: 1 / 3;  
  grid-row: 1 / 3;  
}  
  
.grid div:nth-child(2) {  
  grid-column: -1 / -3;  
  grid-row: -1 / -3;  
}
```

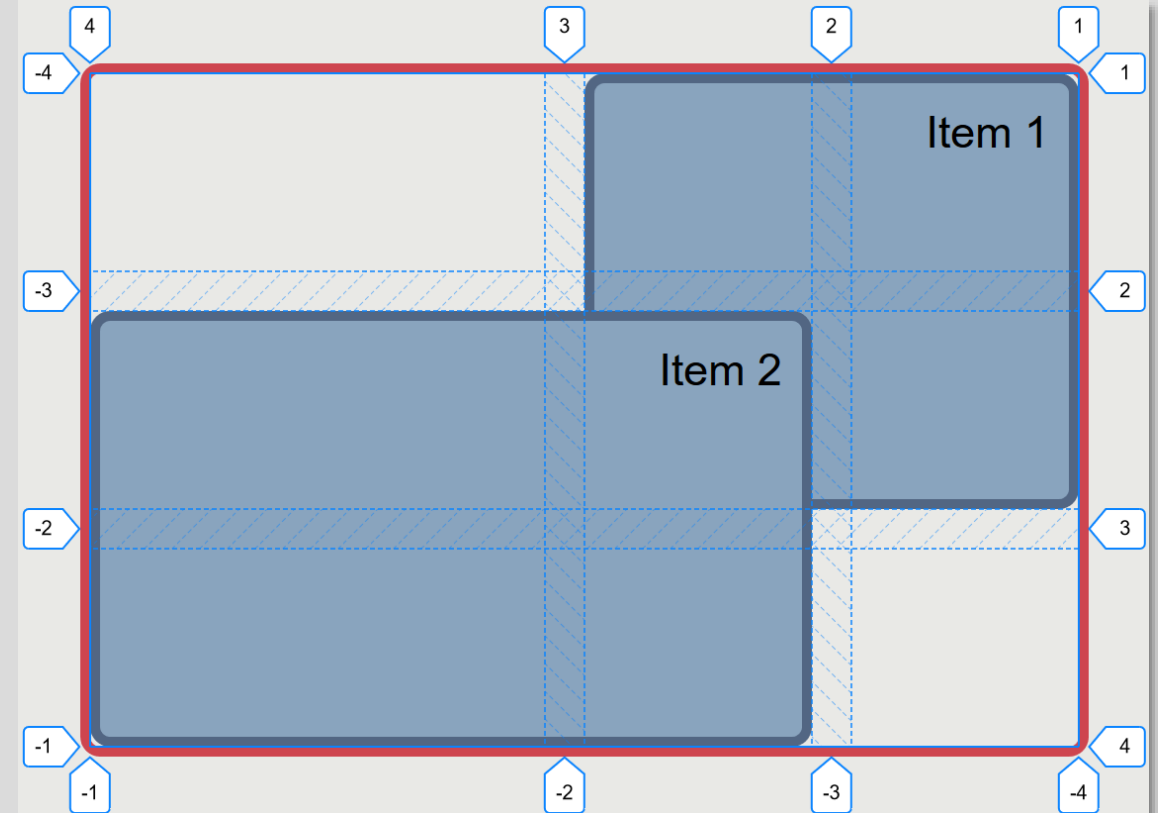
```
.grid {  
  display: grid;  
  width: 500px;  
  grid-template-columns: 1fr 1fr 2fr;  
  grid-template-rows: 100px 100px 100px;  
  grid-gap: 20px;  
}  
  
.grid-rtl { direction: rtl; }  
  
.grid div:nth-child(1) {  
  grid-column: 1 / 3;  
  grid-row: 1 / 3;  
}  
  
.grid div:nth-child(2) {  
  grid-column: -1 / -3;  
  grid-row: -1 / -3;  
}
```



Left to Right



Right to Left



A true system for layout

Designed for the job. Writing mode aware. Consistent.

What's next?

CSS Grid Level 2 “subgrids”



Articles

Design & development



Books

Physical & digital books



Events

Conferences & workshops



Jobs

Find work & employees



Membership

Webinars & early-birds

🔍 Topics



JULY 3, 2018 • [8 comments](#)

CSS Grid Level 2: Here Comes Subgrid

QUICK SUMMARY ↗ CSS Grid Level 2 is already in the process of being specified, and the main feature of this level of the spec is to bring us subgrid. In this article, Rachel Andrew explains the new features.

ABOUT THE AUTHOR

Rachel Andrew is not only editor-in-chief of Smashing Magazine, but also a web developer, writer and speaker. She is the author of a number of

WE ARE NOW OVER A YEAR ON FROM CSS GRID LAYOUT

landing in the majority of our browsers, and the CSS Working Group are already working on Level 2 of the specification. In this article, I'm going to explain what is currently part of the Working and Editor's

📅 13 min read

📍 [CSS](#), [Browsers](#), [Layouts](#)

🐦 Share on [Twitter](#) or [LinkedIn](#)

What's next?

Logical Properties & Values



Articles

Design & development

Books

Physical & digital books

Events

Conferences & workshops

Jobs

Find work & employees

Membership

Webinars & early-birds

🔍 Topics



MARCH 29, 2018 • [8 comments](#)

Understanding Logical Properties And Values

QUICK SUMMARY ↔ CSS Logical Properties and Values aren't quite ready to be used yet, however learning about them can help you to understand CSS Layout, and the interaction with Writing Modes.

ABOUT THE AUTHOR

Rachel Andrew is not only editor-in-chief of Smashing Magazine, but also a web developer, writer and speaker. She is the author of a number of books including [More](#)

IN THE PAST, CSS HAS TIED ITSELF TO PHYSICAL dimensions and directions, physically mapping the placement of elements to the left, right and top and bottom. We `float` an element

📅 9 min read

📌 [CSS](#), [Layouts](#), [Browsers](#)

🐦 Share on [Twitter](#) or [LinkedIn](#)

What's next?

Box Alignment in Block Layout

What's next?

What's important to you?

Thank you!

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