

HTML and CSS

Margit Tennosaar

Homework

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

<https://www.markhendriksen.com/interactive-css-flexbox-tool>

<https://flexboxfroggy.com/>

Grid

- 960px - <https://960.gs/>
- 1200px - <https://1200px.com/>
- 1440px - <https://1440px.com/>

Display: flex;



Display: grid;

Two dimensions



guide-grid/

Image source: <https://hackernoon.com/the-ultimate-css-battle-grid-vs-flexbox-d40da0449faf>

Grid and flex

Flex layout is most appropriate to the components of an application, and small-scale layouts, while the Grid layout is intended for larger scale layouts.

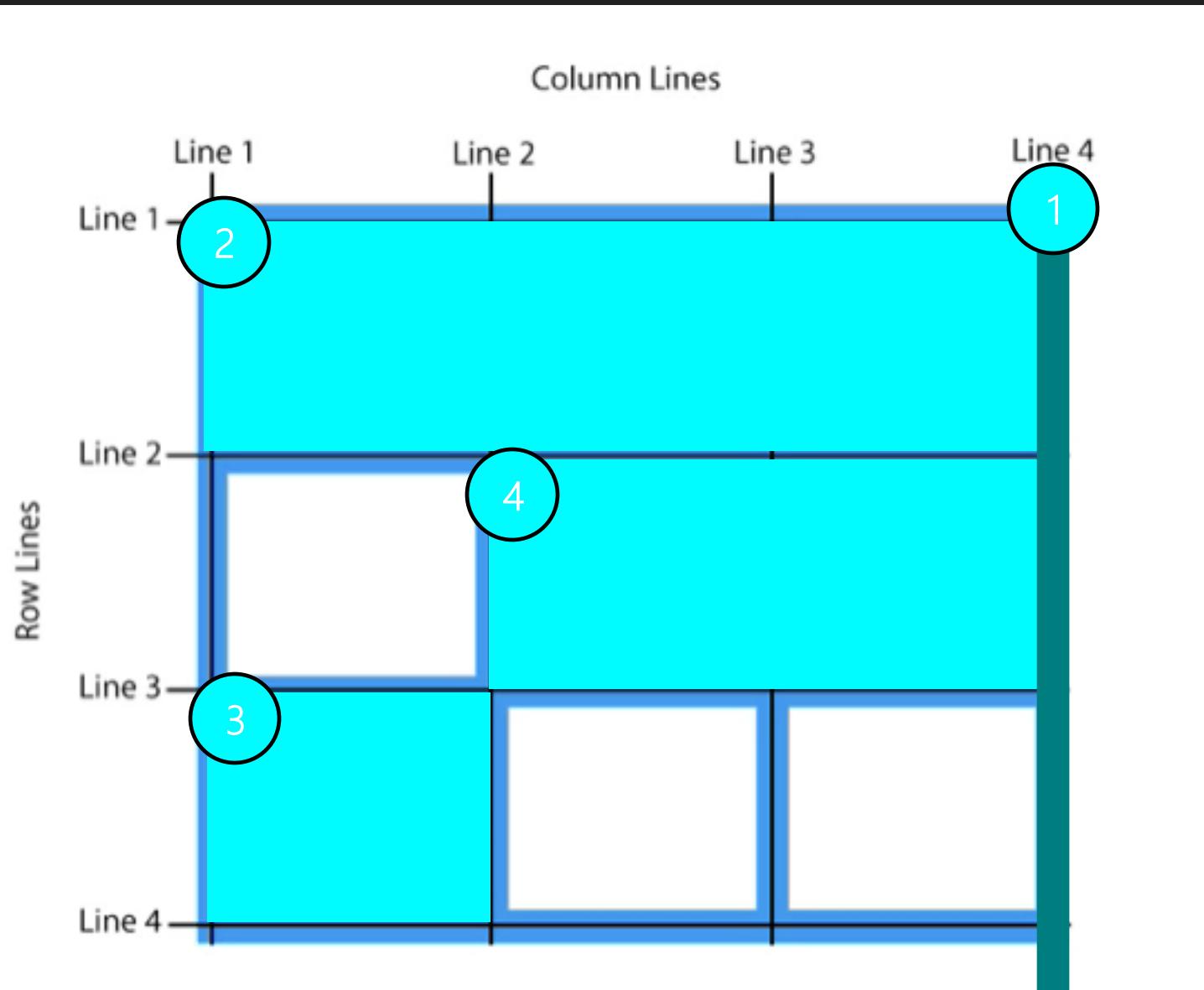
In flexbox layout, the size of a cell (flex-item) is defined inside the flex-item itself, and in the grid layout, the size of a cell (grid-item) is defined inside the grid-container.

CSS grid

- CSS Grid is a powerful tool for creating responsive layouts.
- Allows precise control over rows and columns.
- Works well for both simple and complex designs.
- A layout system that divides a container into rows and columns.
- Enables easy placement and alignment of elements.

How grid is made:

1. **Line or gap** – space between cells
2. **Track** - full line row or column
3. **Cell** - one piece in grid
4. **Area** – more than one cell



Grid

```
.container {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-template-rows: auto;  
  gap: 10px;  
}
```

```
.container {  
  justify-items: center;  
  align-items: center;  
}
```

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

Grid areas

```
.container {  
  grid-template-areas:  
    "header header"  
    "sidebar main"  
    "footer footer";  
}  
.header { grid-area: header; }  
.main { grid-area: main; }
```

Task

<https://css-tricks.com/snippets/css/complete-guide-grid/>

<https://cssgridgarden.com/>

Meta data

Meta data

- Provides important page information.
- Helps SEO, social sharing, and browser settings.

Essential meta tags

```
<title>Title</title>

<meta name="description" content="Learn HTML meta tags for SEO.">

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta property="og:title" content="Learn Meta Tags">
<meta name="twitter:card" content="summary_large_image">

<meta name="robots" content="index, follow">
```

Mobile first approach

Mobile first

- Design for mobile first, then adjust for larger screens.
- Improves user experience and SEO.

Mobile-First Approach

Starts with a **default mobile layout**, then **scales up** using min-width.

Desktop-First Approach

Starts with a **default desktop layout**, then **scales down** using max-width.



The **@media** rule in CSS is used for **responsive design**, allowing styles to change based on screen size, resolution, or other device characteristics. This helps create layouts that **adapt** to different devices, such as smartphones, tablets, and desktops.

```
@media (condition) {  
    /* CSS styles apply only if the condition is true */  
}
```

```
@media (min-width: 768px) {  
    body {  
        background-color: lightblue;  
    }  
}
```

Breakpoint Type	Screen Size Range	Mobile-First Approach	Desktop-First Approach
Extra Small (XS)	0 - 575px	Default (No media query needed)	@media (max-width: 575px)
Small (SM)	576px - 767px	@media (min-width: 576px)	@media (max-width: 767px)
Medium (MD)	768px - 991px	@media (min-width: 768px)	@media (max-width: 991px)
Large (LG)	992px - 1199px	@media (min-width: 992px)	@media (max-width: 1199px)
Extra Large (XL)	1200px - 1399px	@media (min-width: 1200px)	@media (max-width: 1399px)
Extra Extra Large (XXL)	1400px+	@media (min-width: 1400px)	No upper limit

```
body {  
    font-size: 16px;  
    line-height: 1.5;  
}  
.container {  
    display: flex;  
    flex-direction: column;  
}  
  
@media (min-width: 768px) {  
.container {  
    flex-direction: row;  
    justify-content: space-between;  
}  
}
```

Mobile vs desktop first

```
/* Default mobile styles (no media query needed)
*/
body {
  font-size: 16px;
}

/* Apply styles when the screen is at least 768px */

@media (min-width: 768px) {
  body {
    font-size: 18px;
  }
}
```

```
/* Default desktop styles */

body {
  font-size: 18px;
}

/* Apply styles when the screen is 767px or smaller */

@media (max-width: 767px) {
  body {
    font-size: 16px;
  }
}
```

Responsive images

```
<picture>
  <source srcset="image-small.webp" type="image/webp" media="(max-width: 600px)">
  <source srcset="image-large.webp" type="image/webp">
  
</picture>
```

```

```

```

```

