

# CSS Grid and Responsive Web Design Exercises

## Responsive design

### Understanding

My mantra: however, I encourage you to read around and find your own way of doing things as well. There's no common consensus on this btw.

- First: Mobile [Most user base]\*\*
- Second: Desktop [Second most user base]
- Third: Tablet

\*\*note: This rule changes if you are designing a desktop first application. Applications are generally designed desktop first and many times their mobile counterparts come later or there are just mobile apps and no mobile support. If you're working in a team, follow what they're doing.

But for webpages the above rule is GOLD.

### How does this work?

We do mobile first, we define a media query for tablet but don't populate it now. And then we define everything for the desktop.

```
/* Default styles for mobile devices */
body {
  font-size: 14px;
}

/* Styles for tablets (portrait) and up */
@media (min-width: 768px) {
  body {
    font-size: 16px;
  }
}

/* Styles for laptops and up */
@media screen and (min-width: 1024px) {
  body {
    font-size: 18px;
  }
}
```

## ex01: let's start with mobile-first

### Challenge

1. Make the background color of your screen (body) yellow.
2. Now, write a media query that makes the background color transparent when the screen size is more than 600px.

**\*\*note:** We are using 600px so that it is easier for me to teach. You can change the breakpoints to 1024 when publishing the website, or 768 if you want tablet to have desktop layouts.

### Understanding

1. By following the rule of cascade your second rule overrides the first. But it overrides only at specific screen intervals.
2. Now your mobile look will have a yellow background while your desktop look will have a transparent background.
3. This is mobile first design. You can even take the first rule and put it in `mobile.layout.css` and the other rule in `desktop.layout.css` and apply both CSS one after another. Just make sure that everything in `desktop.layout.css` file is in a media query.

### Solution

<https://codesandbox.io/s/first-media-query-solution-iogf29>

## ex02: responsive typography

### Understanding

1. It's important to change the font size when the screen size varies. Especially the headings should feel right on a desktop.
2. We should use relative units. Mostly rem.
  - a. Why? read/write a blog.
  - b. Why rem and not em? read/write a blog.

3. The table that I have shared below, you should keep it handy. For any machine coding question, or project, you can use these sizes. They're pretty standard.

You don't need to be a designer to make good UI, you just need to be ready with a good set of defaults and follow a theme. -Tanay Pratap

## Challenge

Create a simple webpage with responsive typography that adapts to different screen sizes. Your webpage should include a title, subtitle, and a paragraph. The font size of the text elements should change based on the screen size.

# Heading 1

## Heading 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero. Sed cursus ante dapibus diam. Sed nisi. Nulla quis sem at nibh elementum imperdiet. Duis sagittis ipsum. Praesent mauris.

Units

Breakpoint	Title (h1)	Subtitle (h2)	Paragraph (p)
Mobile	2rem	1.5rem	1rem
Tablet	2.5rem	1.75rem	1.125rem
Laptop	3rem	2rem	1.25rem

Solution

# Grids

Grids are much like flex.

Grids are useful more for layout in 2D while flex are in 1D. So, reach out to grids when you are laying out the floor plan for your app.

## Syntax

```
display: grid; /* step 1 */  
grid-template-columns: 200px 1fr auto;  
grid-column-gap: 16px;
```

```
/* to do rows */  
grid-template-rows: 1fr 1fr 1fr;  
grid-row-gap: 0.5rem;
```

COPY

## ex01: row with three columns

### Challenge

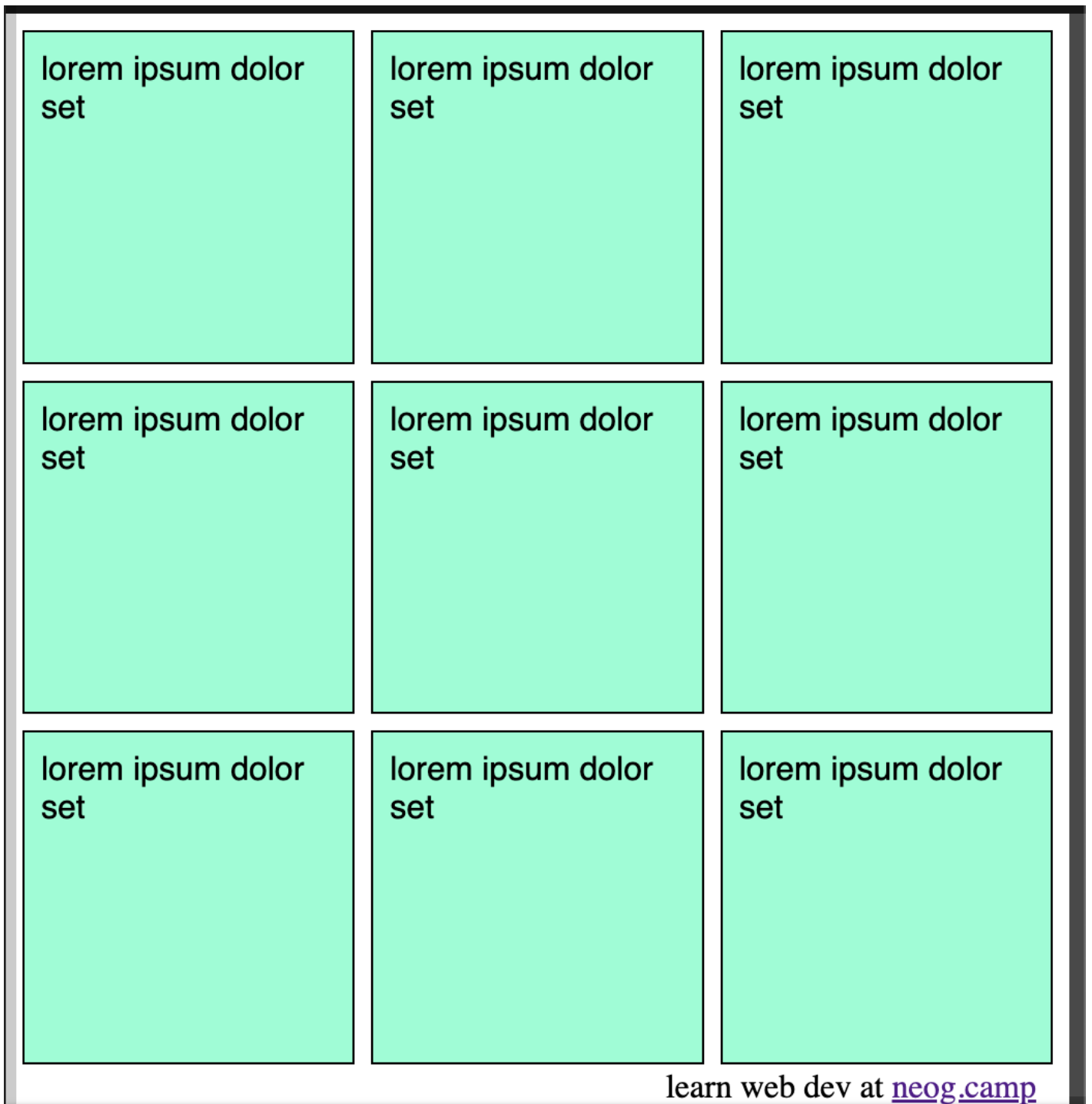
Create a layout class and have 3 rows and 3 columns.

3 rows - 1 row is 1fr | auto | 200px

3 columns - 1 column is 1fr

row gap: 1rem

column gap - 0.5rem



## Solution

<https://codesandbox.io/s/grid-column-rows-9gg66u>

## ex02: make your grid mobile friendly

### Challenge

1. Create a class row.

- 2. Put three cards in it.
- 3. For mobile, it should show two cards in one row.
- 4. For the desktop, it should show three cards in one row.

lorem ipsum dolor set

lorem ipsum dolor set

lorem ipsum dolor set

lorem ipsum dolor set

lorem ipsum dolor set

lorem ipsum dolor set

lorem ipsum dolor set

lorem ipsum dolor set

lorem ipsum dolor set

lorem ipsum dolor set

lorem ipsum dolor set

lorem ipsum dolor set



## Solution

<https://codesandbox.io/s/responsive-grids-otl85j>

## ex03: play with your earlier layout

### Challenge

Reset the whole class layout from the ex01 into three views as shown below

Mobile | Tablet | Desktop view

lorem ipsum dolor set	lorem ipsum dolor set
lorem ipsum dolor set	lorem ipsum dolor set
lorem ipsum dolor set	lorem ipsum dolor set



## Solution

<https://codesandbox.io/s/mobile-friendly-grid-dpnxmv>

## ex04: layout docsite

### Understanding

<https://developer.mozilla.org/en-US/docs/Web/CSS/grid-template-areas>

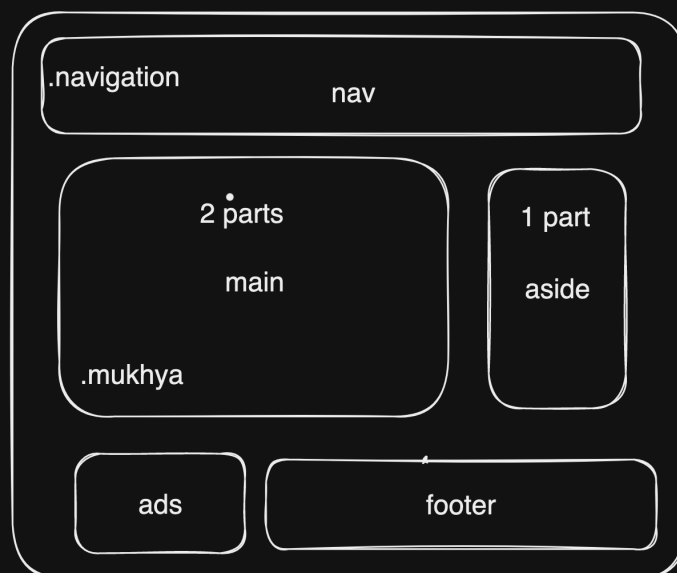
```
grid-template-areas:  
  "nav nav nav"  
  "main main aside"  
  "ads footer footer"
```

```
grid-area: nav
```

```
grid-area: main
```

```
grid-area: ads
```

```
.mukhya { grid-area: main }
```



## Syntax

```
.nav {  
  grid-area: nav;  
}  
  
.main {  
  grid-area: row;  
}  
  
.aside-main {  
  grid-area: aside;  
}  
  
.footer {  
  grid-area: footer;  
}  
  
.layout {  
  grid-template-areas:  
    'nav nav nav'  
    'row row aside'  
    'row row aside'  
    'footer footer footer';  
}
```

[COPY](#)

## Homework

Using the template-area layout a docsite as shown below. You can work out the details later.

- Avatar
- Badge
- Button
- Card
- Input

## introduction

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Aliquam minima, ut blanditiis molestiae eum nisi, rem deleniti, nobis saepe dolor voluptates provident illum consequuntur ab officiis quaerat aperiam ex nesciunt!

### avatar

Lorem, ipsum dolor sit amet consectetur adipisicing elit. A excepturi similique numquam necessitatibus quisquam. Quos perspiciatis porro ut assumenda, incidunt doloremque sint ipsum quisquam magnam dicta ratione? Mollitia, ab qui.

### badge

Lorem, ipsum dolor sit amet consectetur adipisicing elit. A excepturi similique numquam necessitatibus quisquam. Quos perspiciatis porro ut assumenda, incidunt doloremque sint ipsum quisquam magnam dicta ratione? Mollitia, ab qui.

### badge

Lorem, ipsum dolor sit amet consectetur adipisicing elit. A excepturi similique numquam necessitatibus quisquam. Quos perspiciatis porro ut assumenda, incidunt doloremque sint ipsum quisquam magnam dicta ratione? Mollitia, ab qui.

### badge

Lorem, ipsum dolor sit amet consectetur adipisicing elit. A excepturi similique numquam necessitatibus quisquam. Quos perspiciatis porro ut assumenda, incidunt doloremque sint ipsum quisquam magnam dicta ratione? Mollitia, ab qui.

### badge

Lorem, ipsum dolor sit amet consectetur adipisicing elit. A excepturi similique numquam necessitatibus quisquam. Quos perspiciatis porro ut assumenda, incidunt doloremque sint ipsum quisquam magnam dicta ratione? Mollitia, ab qui.

Solution #

<https://codesandbox.io/s/responsive-grids-mobile-view-forked-y1p1q1>