Lesson - 9

Responsive Web Design

SkillUp by Dobrea Vladislav

Lesson Plan

1. **Review HW (merging branches GitHub)**
2. **What is RWD?**
3. **Grid View**
4. **Media Queries**
5. **Meta tag viewport**
6. **Responsive Images**
7. **Working with Figma**

What is Responsive Web Design?

Responsive web design (**RWD**) is an approach to web design that makes web pages render well on a variety of devices and window or screen sizes. Recent work also considers the viewer proximity as part of the viewing context as an extension for RWD.[1] Content, design and performance are necessary across all devices to ensure usability and satisfaction.

Responsive web design makes your web page look good on all devices.

Responsive web design uses only HTML and CSS.

Responsive web design is not a program or a JavaScript.

RWD key points

A site designed with RWD adapts the layout to the viewing environment by using fluid, proportion-based grids, flexible images, and CSS3 media queries, an extension of the `@media` rule, in the following ways:

- The fluid grid concept calls for page element sizing to be in relative units like percentages, rather than absolute units like pixels or points.
- Flexible images are also sized in relative units, so as to prevent them from displaying outside their containing element.
- Media queries allow the page to use different CSS style rules based on characteristics of the device the site is being displayed on, e.g. width of the rendering surface (browser window width or a physical display size).
- Responsive layouts automatically adjust and adapt to any device screen size, whether it is a desktop, a laptop, a tablet, or a mobile phone.

RWD and SEO?

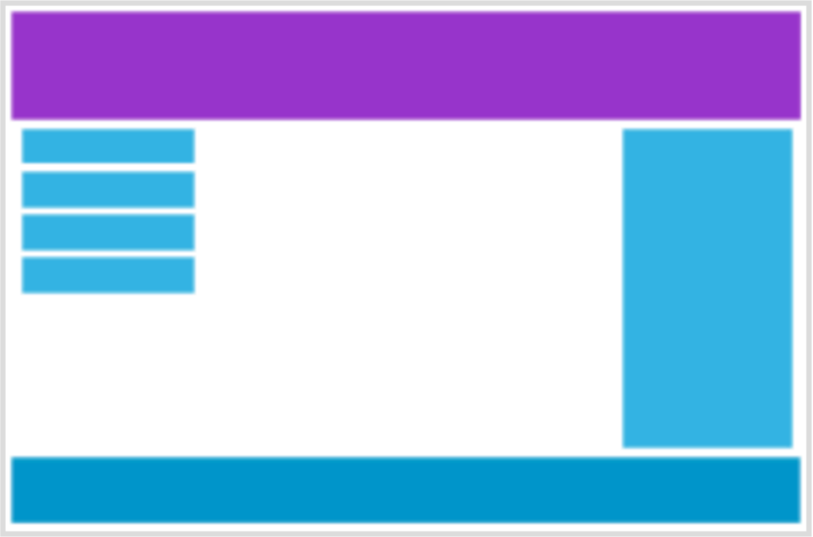
Responsive web design in SEO allows visitors to navigate your website easily. No matter whatever the device is. It provides a better user experience to your website because they could get the information anywhere and anytime. If your website does not have responsive web design Google will penalize for providing a bad user experience.

Important Note: Though you have a well-designed website, If your page is not responsive and mobile-friendly on smartphones you cannot rank higher on SERP

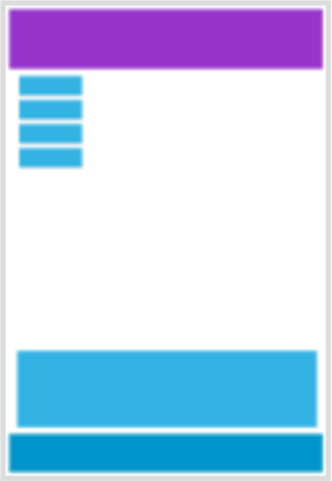
Designing For The Best Experience For All Users

Web pages can be viewed using many different devices: desktops, tablets, and phones. Your web page should look good, and be easy to use, regardless of the device.

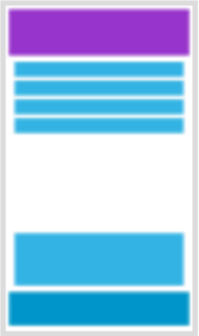
Web pages should not leave out information to fit smaller devices, but rather adapt its content to fit any device:



Desktop



Tablet



Phone

Responsive Web Design - The Viewport

What is The Viewport?

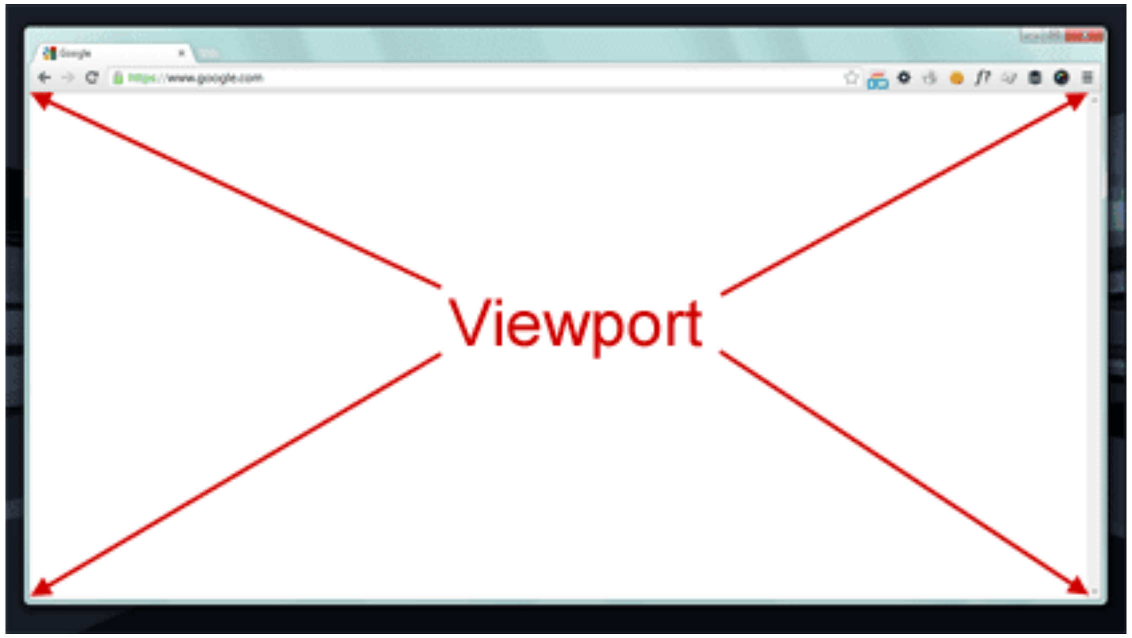
The viewport is the user's visible area of a web page.

The viewport varies with the device, and will be smaller on a mobile phone than on a computer screen.

Before tablets and mobile phones, web pages were designed only for computer screens, and it was common for web pages to have a static design and a fixed size.

Then, when we started surfing the internet using tablets and mobile phones, fixed size web pages were too large to fit the viewport. To fix this, browsers on those devices scaled down the entire web page to fit the screen.

This was not perfect!! But a quick fix.



Setting The Viewport

HTML5 introduced a method to let web designers take control over the viewport, through the `<meta>` tag.

You should include the following `<meta>` viewport element in all your web pages:

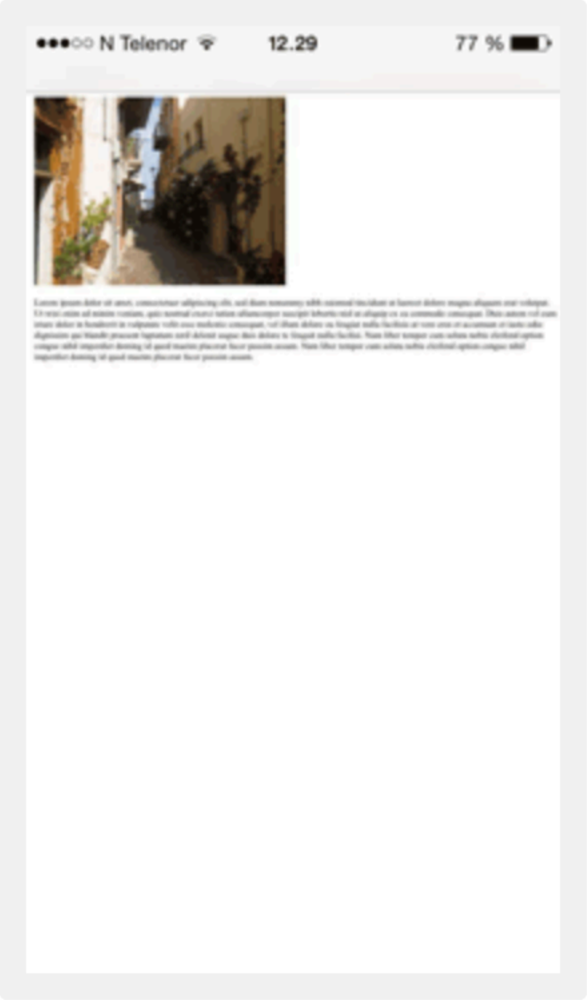
```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

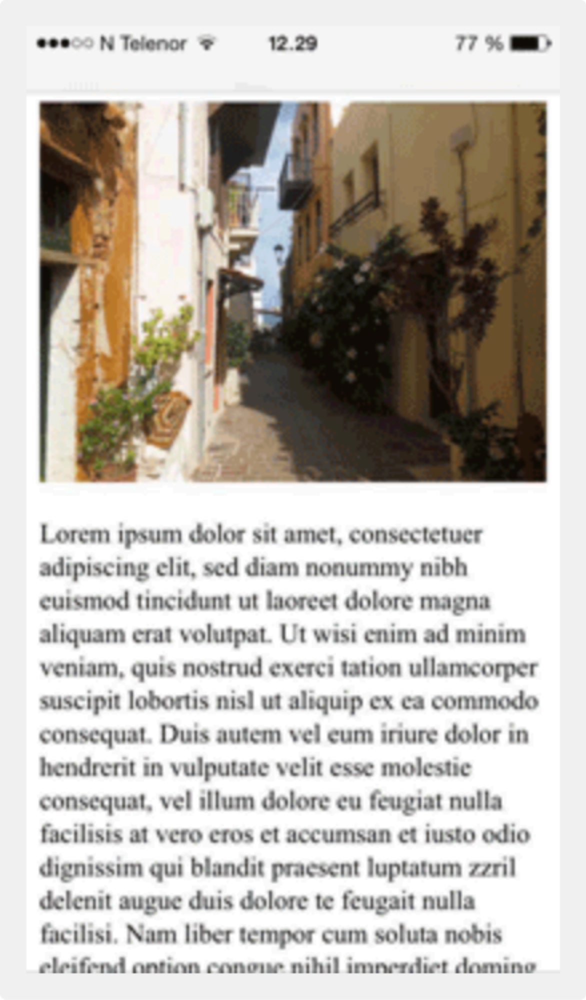
A `<meta>` viewport element gives the browser instructions on how to control the page's dimensions and scaling.

The `width=device-width` part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The `initial-scale=1.0` part sets the initial zoom level when the page is first loaded by the browser.

Viewport Difference





Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming

[Without the viewport meta tag]

[With the viewport meta tag]

Size Content to The Viewport

Users are used to scroll websites vertically on both desktop and mobile devices - but not horizontally!

So, if the user is forced to scroll horizontally, or zoom out, to see the whole web page it results in a poor user experience.

Some additional rules to follow:

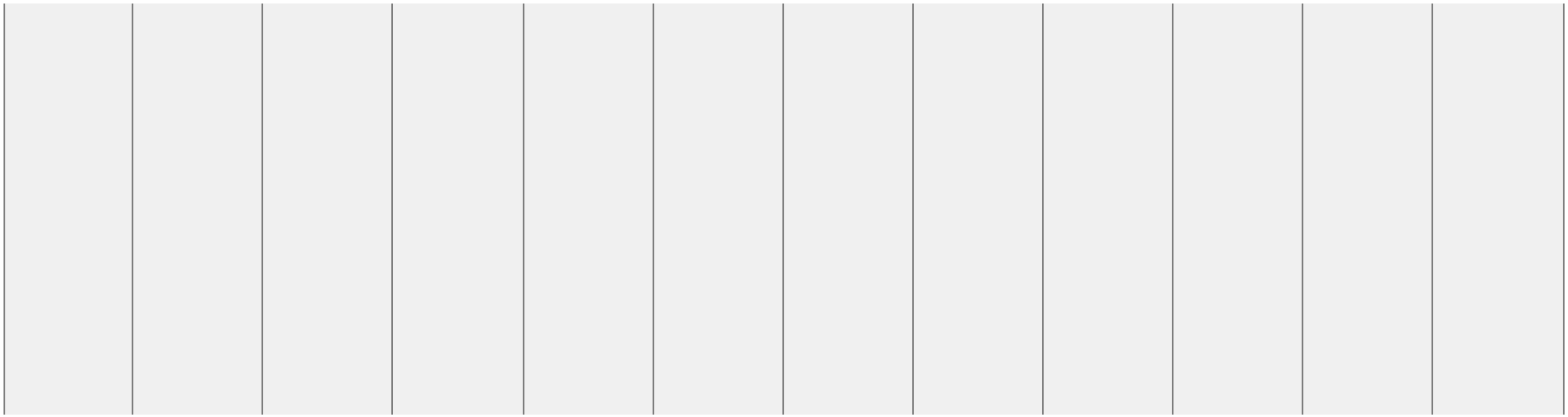
1. Do NOT use large fixed width elements - For example, if an image is displayed at a width wider than the viewport it can cause the viewport to scroll horizontally. Remember to adjust this content to fit within the width of the viewport.

2. Do NOT let the content rely on a particular viewport width to render well - Since screen dimensions and width in CSS pixels vary widely between devices, content should not rely on a particular viewport width to render well.

3. Use CSS media queries to apply different styling for small and large screens - Setting large absolute CSS widths for page elements will cause the element to be too wide for the viewport on a smaller device. Instead, consider using relative width values, such as width: 100%. Also, be careful of using large absolute positioning values. It may cause the element to fall outside the viewport on small devices.

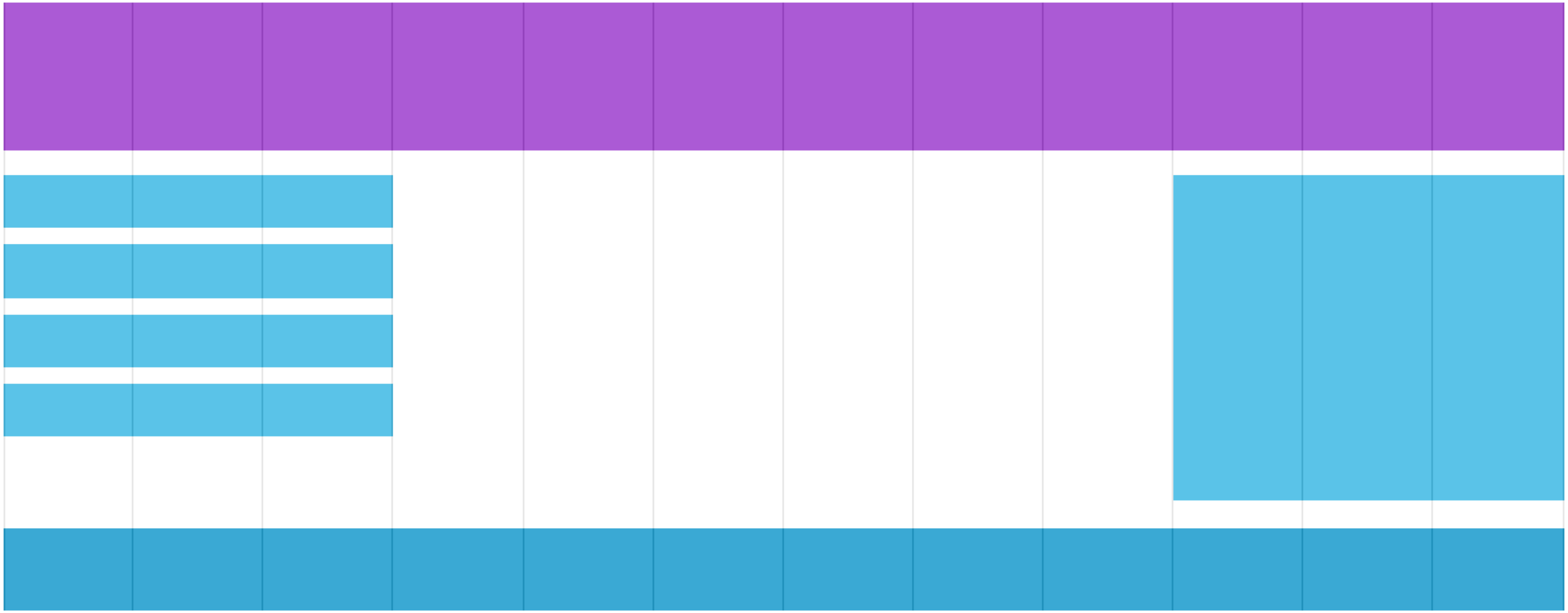
What is a Grid-View?

Many web pages are based on a grid-view, which means that the page is divided into columns:

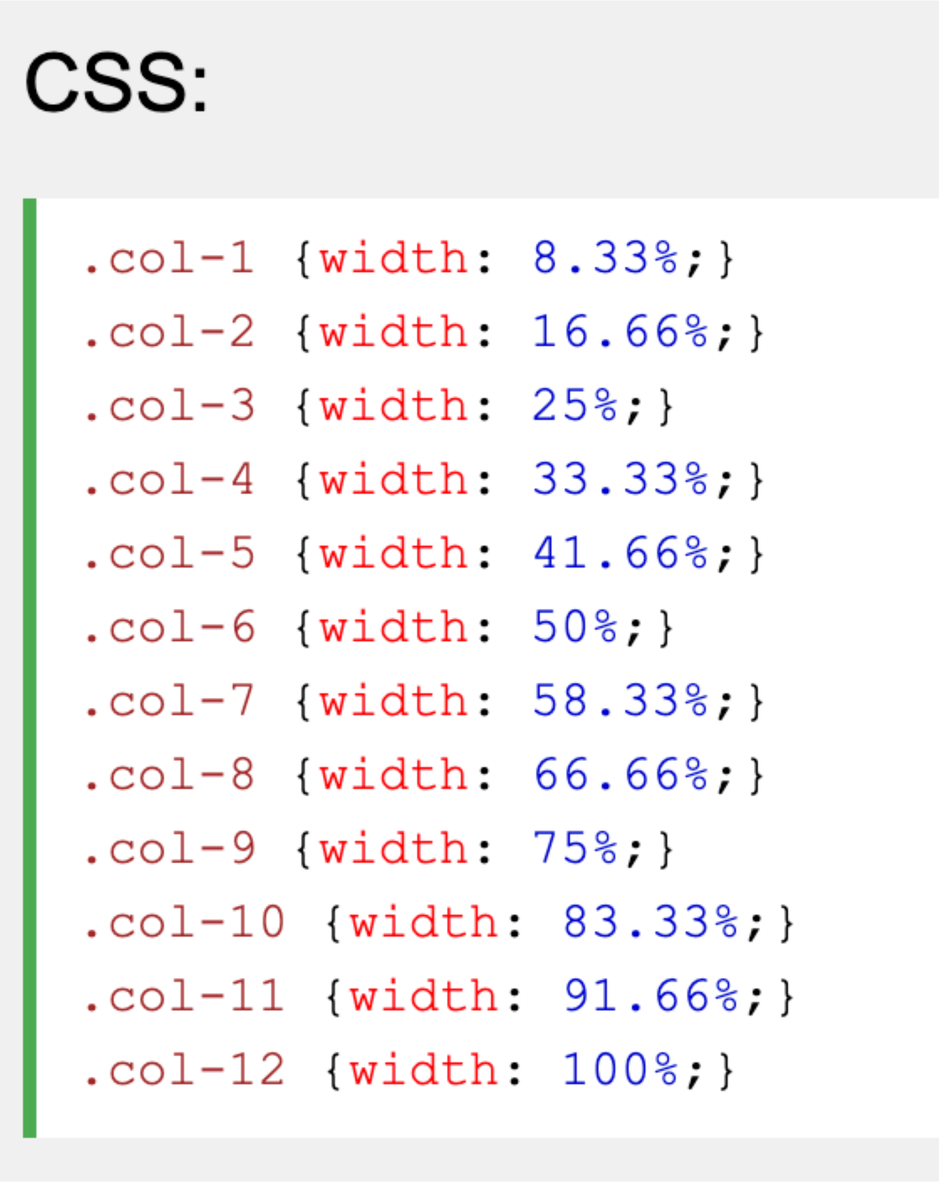


Helps in placing items on the page

Using a grid-view is very helpful when designing web pages. It makes it easier to place elements on the page.



Very Basic Implementation



CSS:

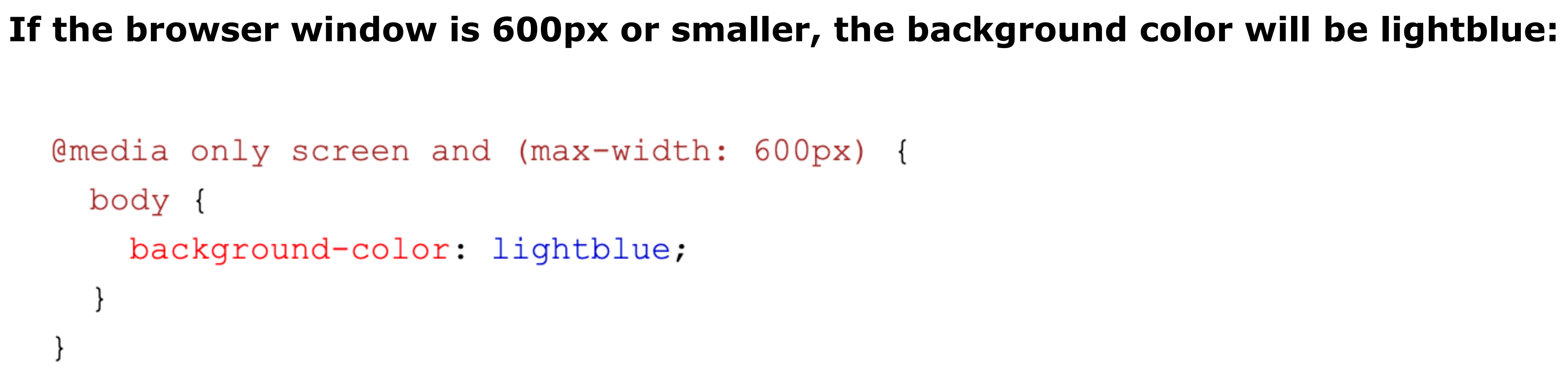
```
.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}
```

What is a Media Query?

Media query is a CSS technique introduced in CSS3.

It uses the `@media` rule to include a block of CSS properties only if a certain condition is true.

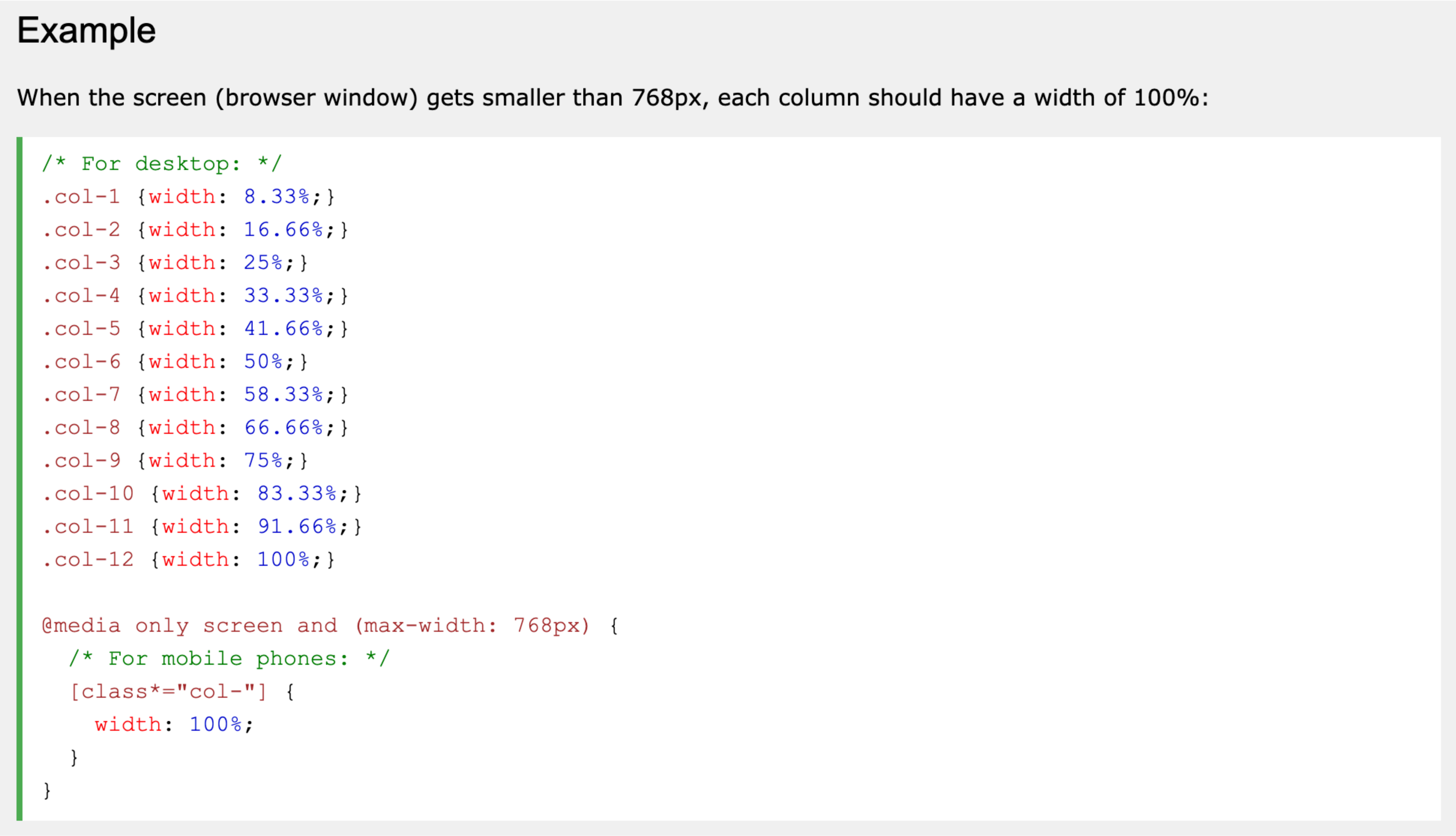
This is also called adding a breakpoint



If the browser window is 600px or smaller, the background color will be lightblue:

```
@media only screen and (max-width: 600px) {
  body {
    background-color: lightblue;
  }
}
```

Example



Example

When the screen (browser window) gets smaller than 768px, each column should have a width of 100%:

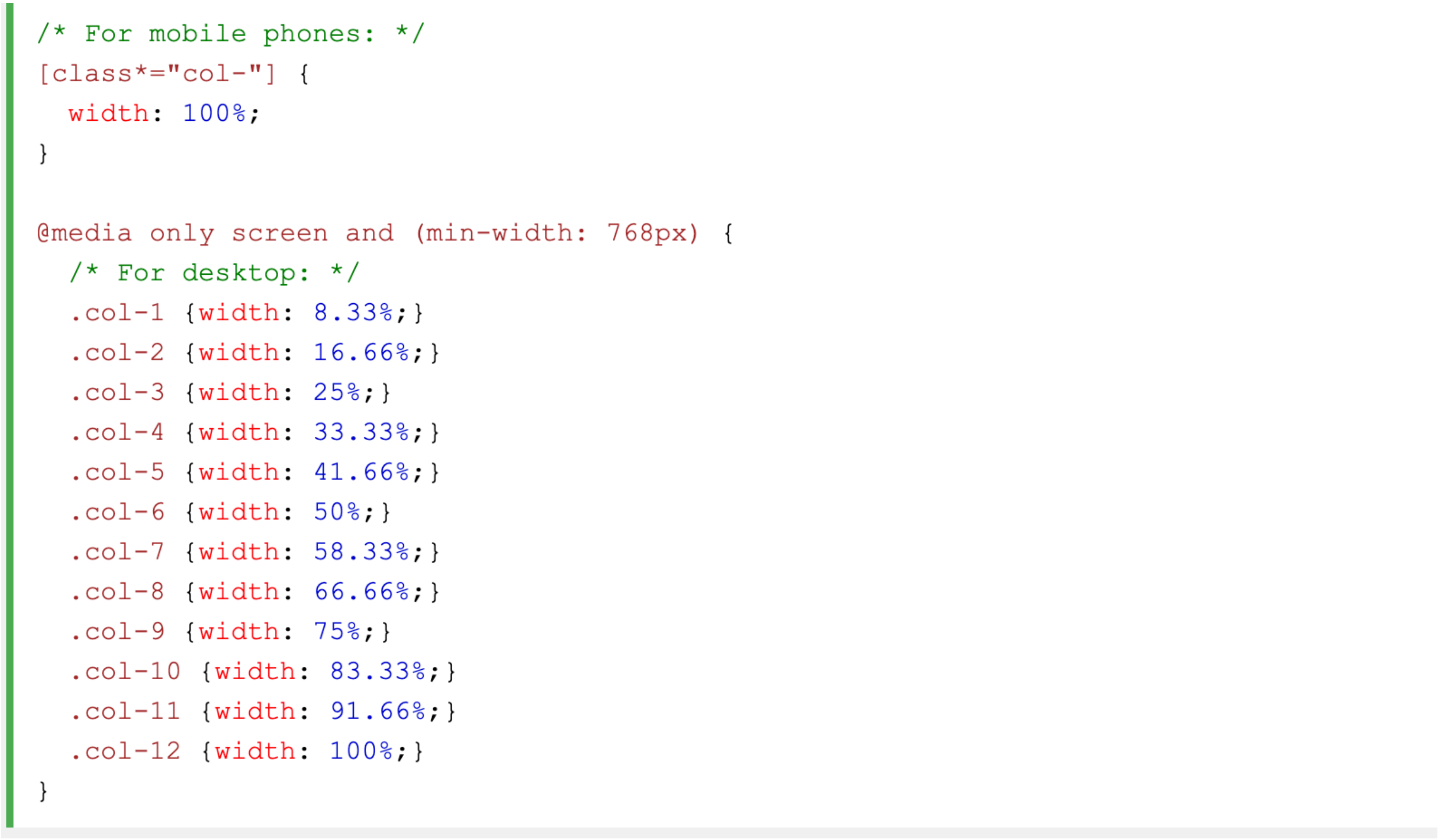
```css
/* For desktop: */
.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}

@media only screen and (max-width: 768px) {
  /* For mobile phones: */
  [class*="col-"] {
    width: 100%;
  }
}
```

Always Design for Mobile First

Mobile First means designing for mobile before designing for desktop or any other device (This will make the page display faster on smaller devices).

This means that we must make some changes in our CSS.



```css
/* For mobile phones: */
[class*="col-"] {
  width: 100%;
}

@media only screen and (min-width: 768px) {
  /* For desktop: */
  .col-1 {width: 8.33%;}
  .col-2 {width: 16.66%;}
  .col-3 {width: 25%;}
  .col-4 {width: 33.33%;}
  .col-5 {width: 41.66%;}
  .col-6 {width: 50%;}
  .col-7 {width: 58.33%;}
  .col-8 {width: 66.66%;}
  .col-9 {width: 75%;}
  .col-10 {width: 83.33%;}
  .col-11 {width: 91.66%;}
  .col-12 {width: 100%;}
}
```

Responsive Images

If the `width` property is set to a percentage and the height is set to "auto", the image will be responsive and scale up and down:

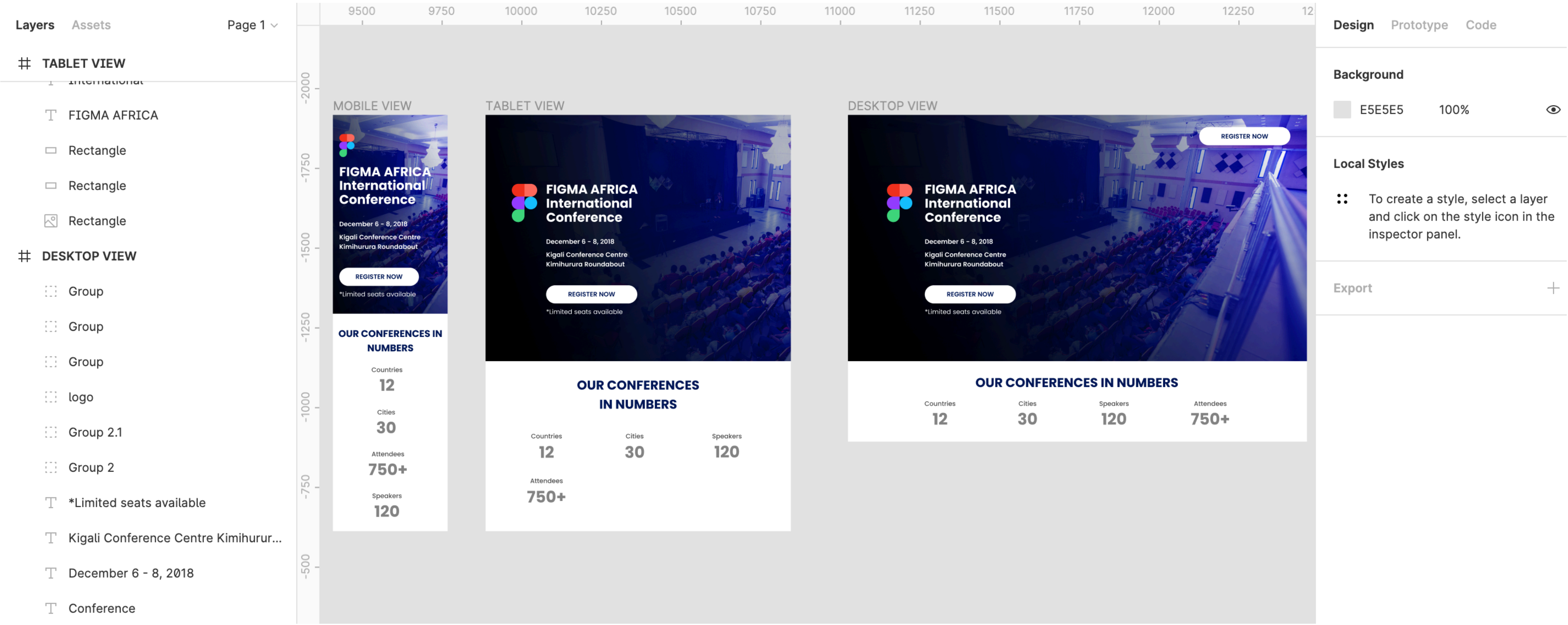
```
img {
  width: 100%;
  height: auto;
}
```

Notice that in the example above, the image can be scaled up to be larger than its original size. A better solution, in many cases, will be to use the `max-width` property instead.

```
img {
  max-width: 100%;
  height: auto;
}
```

Let’s practice

Figma Project



Background

E5E5E5 100%

To create a style, select a layer and click on the style icon in the inspector panel.

Go to https://www.figma.com/file/NFChPUQyWb2NZpqTgeelLn/Event-Landing-Page-(Copy)?node-id=0%3A1

Learning Resources

1. **Responsive Web Design**
    1. **https://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/**
    2. **https://www.w3schools.com/css/css_rwd_intro.asp**
    3. **https://www.freecodecamp.org/news/learn-responsive-web-design-in-5-minutes/**
    4. **https://www.youtube.com/watch?v=srvUrASNj0s**
2. **Media Queries**
    1. **https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries**
    2. **https://medium.com/@pbojinov/media-queries-explained-9bf20a85731f**
3. **Figma**
    1. **https://blog.prototypr.io/getting-started-with-figma-fc0db85c852c**

Home Work Instructions

1. Open figma project https://www.figma.com/file/NFChPUQyWb2NZpqTgeelLn/Event-Landing-Page-(Copy)?node-id=0%3A1
2. Create a new folder and start git project
3. Implement it using HTML/CSS and using RWD techniques.
4. Push it to GitHub. Name it Lesson-9-HW