

# RESPONSIVE CSS

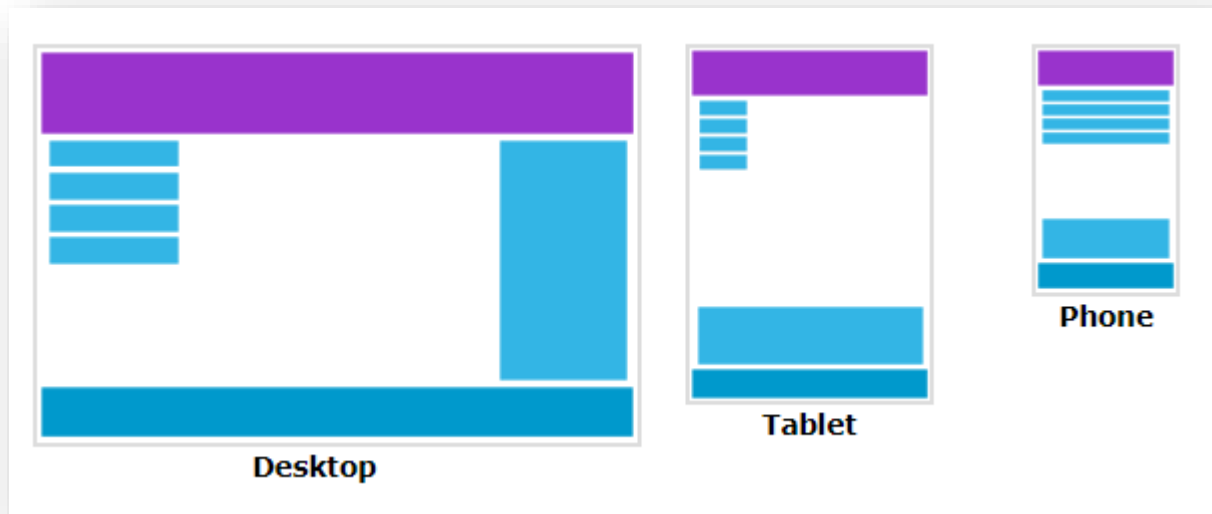
# RESPONSIVE

In a **responsive design**, the page “responds” to changes in the browser size that go beyond the width scaling of a liquid layout.

One of the problems of a liquid layout is that

■ images and horizontal navigation elements tend to take up a fixed size, and when the browser window shrinks to the size of a mobile browser.

# RESPONSIVE WEB DESIGN



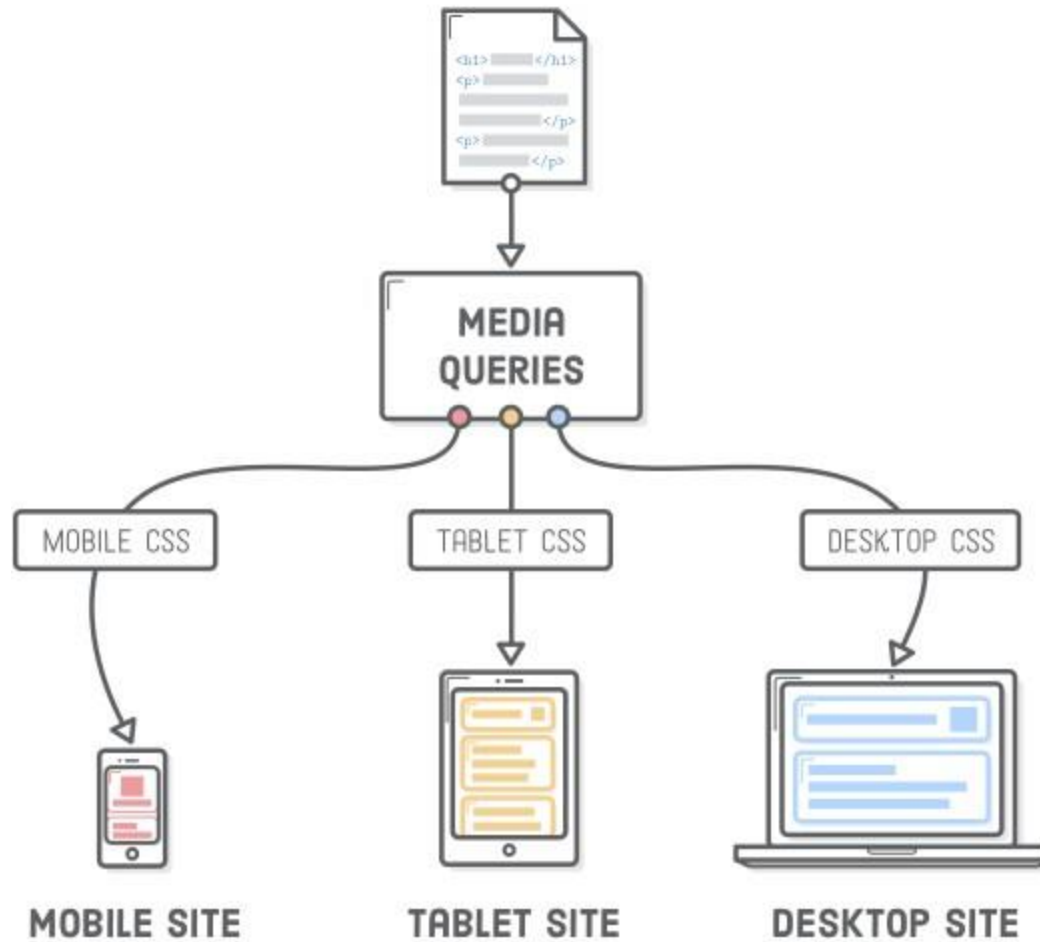
# MOBILE FIRST DESIGN

One of the most influential recent approaches to web design is sometimes referred to as **mobile first design**.

The rationale for the mobile-first approach lies

- ▮ the increasingly larger audience whose principal technology for accessing websites is a smaller device such as a phone or a tablet.
- ▮ Forces the designers and site architects to focus on the most important component of any site: the content.

## HTML CONTENT



# KEY COMPONENTS

Four key components make responsive design work. They are:

1. Liquid/Fluid layouts
2. Scaling images to the viewport size
3. Setting viewports via the `<meta>` tag

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```
4. Customizing the CSS for different viewports using media queries

# 1. LIQUID/FLUID LAYOUT

## 2. IMAGE SCALING

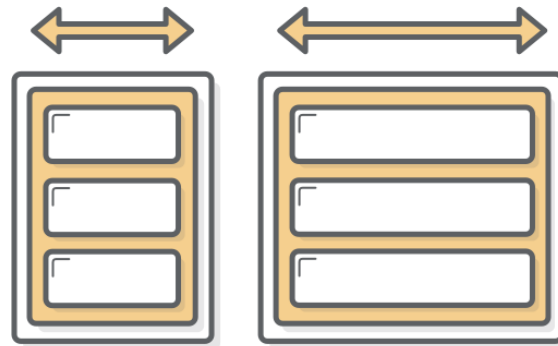
Responsive designs begin with a liquid layout

one in which most elements have their widths specified as percentages.

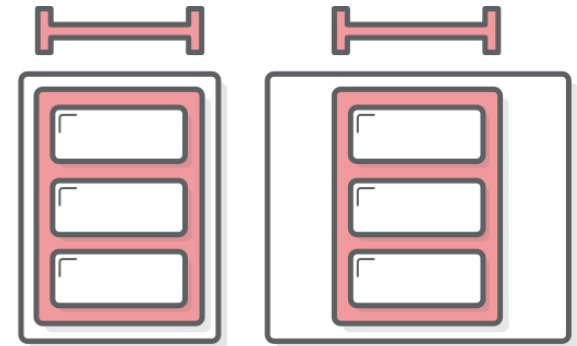
```
img {
```

```
  max-width: 100%;
```

```
}
```



FLUID LAYOUT



FIXED-WIDTH LAYOUT

# 3. SETTING VIEWPORT

A key technique in creating responsive layouts makes use of the ability of current mobile browsers to shrink or grow the web page to fit the width of the screen.

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

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



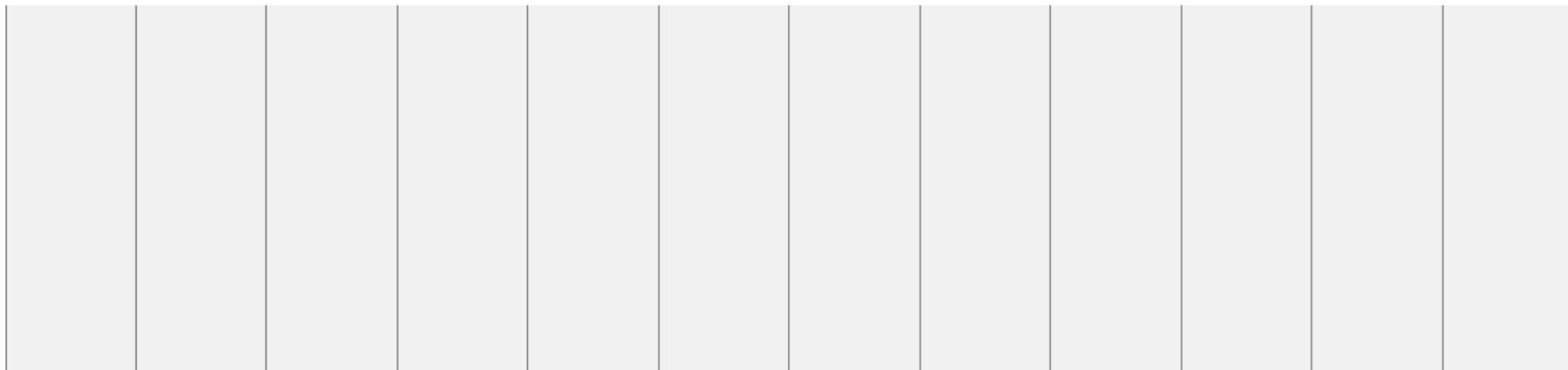
**Without the viewport meta tag**



**With the viewport meta tag**

# What is a Grid-View?

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



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



# LIVE EXAMPLES

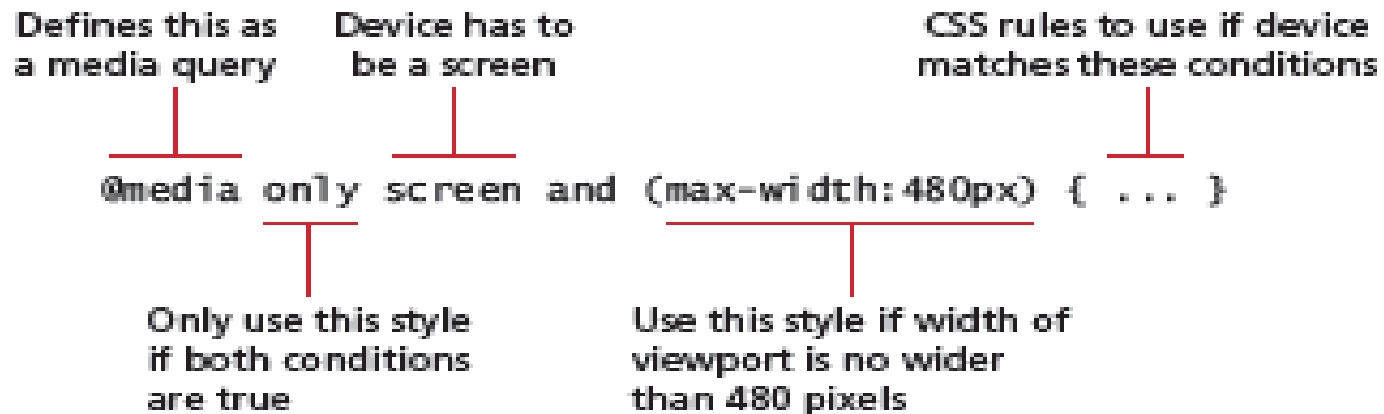
[https://www.w3schools.com/CSS/tryit.asp?filename=tryresponsive\\_webpage](https://www.w3schools.com/CSS/tryit.asp?filename=tryresponsive_webpage)

[https://www.w3schools.com/CSS/css\\_rwd\\_grid.asp](https://www.w3schools.com/CSS/css_rwd_grid.asp)

# 4. MEDIA QUERIES

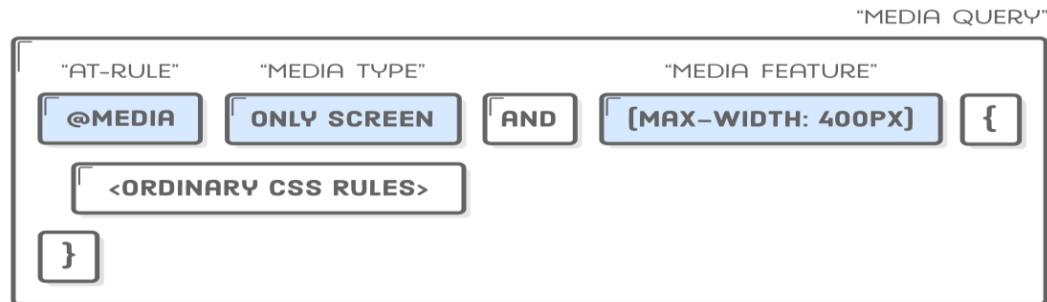
The other key component of responsive designs is **CSS media queries**.

A **media query** is a way to apply style rules based on the medium that is displaying the file.



**FIGURE 5.33** Sample media query

# 4.1 FEATURES



Feature	Description
<code>width</code>	Width of the viewport
<code>height</code>	Height of the viewport
<code>device-width</code>	Width of the device
<code>device-height</code>	Height of the device
<code>orientation</code>	Whether the device is portrait or landscape
<code>color</code>	The number of bits per color

TABLE 5.3 Browser Features You Can Examine with Media Queries

## 4.2 ADD BREAKPOINTS

```
/* Extra small devices (phones, 600px and down) */  
@media only screen and (max-width: 600px) {...}
```

```
/* Small devices (portrait tablets and large phones, 600px and up) */  
@media only screen and (min-width: 600px) {...}
```

```
/* Medium devices (landscape tablets, 768px and up) */  
@media only screen and (min-width: 768px) {...}
```

```
/* Large devices (laptops/desktops, 992px and up) */  
@media only screen and (min-width: 992px) {...}
```

```
/* Extra large devices (large laptops and desktops, 1200px and up) */  
@media only screen and (min-width: 1200px) {...}
```

## 4.3 ADJUSTING FOR DEVICES

```
#logo {  
  background: url(images/logo.png);  
  width: 600px;  
  border: 1px #ccc solid; }
```

```
@media only screen and (max-device-width: 480px) {  
  #logo {  
    background: url(images/logo_mobile.png);  
    width: 440px; }  
}
```

```
/* This part is read by every device/ viewport */  
body{ font-size: 14px;}  
/* This part is only read by viewports wider than 1200 pixels */  
@media (min-width: 1200px)  
{body{ font-size: 16px;}}
```

## 4.3 ADJUSTING FOR DEVICES

Hide Elements:

```
/* If the screen size is 600px wide or less, hide the element */
@media only screen and (max-width: 600px) {
  div.example {
    display: none;
  }
}
```

Images:

```
/* For width smaller than 400px: */
body {
  background-image: url('img_smallflower.jpg');
}

/* For width 400px and larger: */
@media only screen and (min-width: 400px) {
  body {
    background-image: url('img_flowers.jpg');
  }
}
```



## 4.4 PRINT AND SCREEN

```
@media screen{
```

```
Body    {background: #ff0000;}
```

```
}
```

```
@media print {
```

```
Body    {background: #fff;}
```

```
}
```

## 4.5 MULTIPLE CSS FILES

`<link href="general.css" rel="stylesheet" media="screen">`

`<link href="blue.css" rel="stylesheet" media="print">`

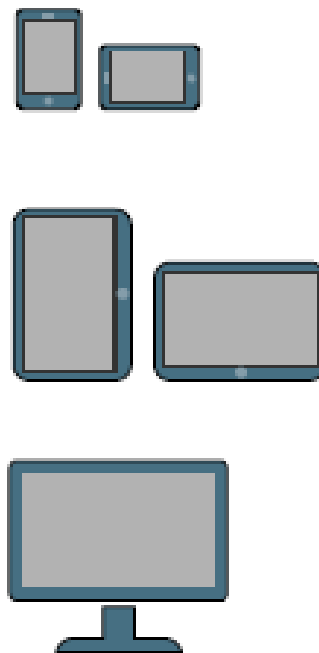
Value	Description
all	Used for all media type devices
print	Used for printers
screen	Used for computer screens, tablets, smart-phones etc.
speech	Used for screenreaders that "reads" the page out loud

# MEDIA QUERIES

## PROGRESSIVE ENHANCEMENT

Contemporary responsive sites will typically provide CSS rules for phone displays first, then tablets, then desktop monitors, an approach called **progressive enhancement**,

in which a design is adapted to progressively more advanced devices,



styles.css

```
/* rules for phones */
@media only screen and (max-width:480px)
{
    #slider-image { max-width: 100%; }
    #flash-ad { display: none; }
    ...
}

/* CSS rules for tablets */
@media only screen and (min-width: 481px)
and (max-width: 768px)
{
    ...
}

/* CSS rules for desktops */
@media only screen and (min-width: 769px)
{
    ...
}
```

Instead of having all the rules in a single file, we can put them in separate files and add media queries to `<link>` elements.

```
<link rel="stylesheet" href="mobile.css" media="screen and (max-width:480px)" />
<link rel="stylesheet" href="tablet.css" media="screen and (min-width:481px)
and (max-width:768px)" />
<link rel="stylesheet" href="desktop.css" media="screen and (min-width:769px)" />

<!--[if lt IE 9]>
<link rel="stylesheet" media="all" href="style-ie.css"/>
<![endif]-->
```

Handles Internet Explorer 8 and earlier using IE conditional comments.

**FIGURE 5.34** Media queries in action

# EXERCISE

Q1. Create 4 breakpoints based on screen sizes, and set a different font face and font size for each of it.

Q2. Create a media query which specifies blue color for screens larger than 720px and light blue for screens smaller than 480 px.

Q3. Create a media query that displays white background and font size of 12 points for print media and light-blue background and 14 font size for screen media.

Q4. Create a media query that displays all elements as block in portrait orientation and as inline-block in landscape orientation.

Q5. Create a query that checks if a screen has orientation portrait and width between 450 and 720 pixels then table border is set to 1 px green color. For all else it will be 2 px blue color.

# EXERCISE - SOLUTIONS

Q1. Create a media query which specifies blue color for screens larger than 720px and light blue for screens smaller than 480 px.

```
@media only screen and (min-width: 720px){color:blue;}
```

```
@media (max-width: 480px){color:light-blue;}
```

Q2. Create a media query that displays white background and font size of 12 points for print media and light-blue background and 14 font size for screen media.

```
@media screen{  
  body{  
    background: lightblue;  
    font-size: 14pt;}}
```

```
@media print{  
  body  
  { background: lightblue;  
    font-size: 12pt;}}
```

# EXERCISE - SOLUTIONS

Q3. Create a media query that displays all p elements as block in portrait orientation and as inline-block in landscape orientation.

```
@media (orientation: landscape){p{display:block;}}  
@media (orientation: portrait){p{display:inline-block;}}
```

Q4. Create 4 breakpoints based on screen sizes, and set a different font face and font size for each of it.

```
@media only screen and (max-width: 600px){p{font-family: "Comic Sans MS";font-size: 1pt;}}  
@media only screen and (min-width: 600px){p{font-family: "Times New Roman";font-size: 2pt;}}  
@media only screen and (min-width: 768px){p{font-family: "Ariel";font-size: 3pt;}}  
@media only screen and (min-width: 992px){p{font-family: "Comic Sans MS";font-size: 4pt;}}  
@media only screen and (min-width: 1200px){p{font-family: "Comic Sans MS";font-size: 5pt;}}
```

# EXERCISE - SOLUTIONS

Q5. Create a query that checks if a screen has orientation portrait and width between 450 and 720 pixels then table border is set to 1 px green color. For all else it will be 2 px blue color.

```
table{ border: 2px solid blue; }  
@media only screen and (orientation: portrait) and (min-  
width:450px) and(max-width:720px)  
{  
    table{ border: 1px solid green; }  
}
```



# EXERCISE

**<style>**

body { background-color: lightblue;}

ul li {display: inline;}

p { display:inline-block; border: 1px solid black; width:30%;}

**@media screen and (min-width: 300px) and (max-width:440px) {**

body {background-color: lightgreen;        }                }

**</style>**

**<body>**

**<h1>Resize the browser window to see the effect!</h1>**

**<div>**

**<p>The media query will apply on what media type and at what width??.</p>**

**<p>The media query will apply on what media type and at what width??.</p>**

**<p>The media query will apply on what media type and at what width??.</p>**

**</div>   <div>email me if you need any assistance</div> </body>**

Change the code for devices having width more than 450px but less than 650px a background-image:url('mountain.jpg') should appear. Also, a text *"I am using media query"* should appear before the text "Resize the browser window...".

# PARTIAL SOLUTION

```
@media screen and (min-width:720px)
{
  table {
    border: 1px solid green;
  }
  body {
    /*backgorund-image: url("mountain.jpg");*/
    background-color: lightblue;
  }
  h1.tohide
  {
    display: none;
  }
}
```

# REFERENCES

MDN Docs: Media Queries

[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS media queries/Using media queries](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_media_queries/Using_media_queries)

MDN Docs: Responsive Design

[https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS layout/Responsive Design](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design)

W3Schools: Responsive Web Design

[https:// www.w3schools.com/css/css rwd intro.asp](https://www.w3schools.com/css/css_rwd_intro.asp)