Media queries
step 1 to responsive websites
Motivation for responsive websites

- Media queries allow you to group styles together and target them to specific devices.
- You can target them to devices based on a specified criteria.
- For example: target by width, height, orientation (landscape v. portrait).
- Screen size is the most salient feature between laptop and hand held device.

*different web page layouts from the same HTML*
CSS3 Introduces Media Queries

Media queries in CSS3 extend the CSS2 media types idea: Instead of looking for a type of device, they look at the capability of the device.

Media queries can be used to check many things, such as:

- width and height of the viewport
- width and height of the device
- orientation (is the tablet/phone in landscape or portrait mode?)
- resolution

Using media queries are a popular technique for delivering a tailored style sheet to tablets, iPhone, and Androids.

<table>
<thead>
<tr>
<th>Property</th>
<th>🌐</th>
<th>🍎</th>
<th>🌐</th>
<th>🌐</th>
<th>🌐</th>
<th>🌐</th>
</tr>
</thead>
<tbody>
<tr>
<td>@media</td>
<td>21.0</td>
<td>12.0</td>
<td>9.0</td>
<td>3.5</td>
<td>4.0</td>
<td>9.0</td>
</tr>
</tbody>
</table>

The numbers in the table specifies the first browser version that fully supports the @media rule.
Basic syntax of media queries

```css
@media not only mediatype and (expressions) {
  CSS-Code;
}
```

Within those curly braces you have your styles, it's like a stylesheet within a style sheet.

If TRUE, then the styles within the {} will apply.

Media feature (resolves to true or false)

```css
@media (max-width: 767px) {
  p {
    color: blue;
  }
}
```

You can have more than one media feature combine together using logical operators.
Media types:

You can also have different stylesheets for different media:

```html
<link rel="stylesheet" media="mediatype and\ not\ only (expressions)" href="print.css">  
```

### CSS3 Media Types

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>Used for all media type devices</td>
</tr>
<tr>
<td>print</td>
<td>Used for printers</td>
</tr>
<tr>
<td>screen</td>
<td>Used for computer screens, tablets, smart-phones etc.</td>
</tr>
<tr>
<td>speech</td>
<td>Used for screenreaders that &quot;reads&quot; the page out loud</td>
</tr>
</tbody>
</table>
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.

---

**Example**

If the browser window is smaller than 500px, the background color will change to lightblue:

```css
@media only screen and (max-width: 500px) {
  body {
    background-color: lightblue;
  }
}
```
Common features for media queries

```html
<style>
  @media (max-width: 767px) {
    p {
      color: blue;
    }
  }

  @media (min-width: 800px) { ... }

  @media (orientation: portrait) { ... }

  @media screen { ... }

  @media print { ... }
</style>
```

- **max-width** and **min-width** are the most common features used for `@media` queries because it is the most common way to target different devices.

Target “only screens” vs targeting “only print”
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>Used for all media type devices</td>
</tr>
<tr>
<td>aural</td>
<td>Deprecated. Used for speech and sound synthesizers</td>
</tr>
<tr>
<td>braille</td>
<td>Deprecated. Used for braille tactile feedback devices</td>
</tr>
<tr>
<td>embossed</td>
<td>Deprecated. Used for paged braille printers</td>
</tr>
<tr>
<td>handheld</td>
<td>Deprecated. Used for small or handheld devices</td>
</tr>
<tr>
<td>print</td>
<td>Used for printers</td>
</tr>
<tr>
<td>projection</td>
<td>Deprecated. Used for projected presentations, like slides</td>
</tr>
<tr>
<td>screen</td>
<td>Used for computer screens, tablets, smart-phones etc.</td>
</tr>
<tr>
<td>speech</td>
<td>Used for screenreaders that &quot;reads&quot; the page out loud</td>
</tr>
<tr>
<td>tty</td>
<td>Deprecated. Used for media using a fixed-pitch character grid, like teletypes and terminals</td>
</tr>
<tr>
<td>tv</td>
<td>Deprecated. Used for television-type devices</td>
</tr>
</tbody>
</table>
## Media Features

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aspect-ratio</td>
<td>The ratio between the width and the height of the viewport</td>
</tr>
<tr>
<td>color</td>
<td>The number of bits per color component for the output device</td>
</tr>
<tr>
<td>color-index</td>
<td>The number of colors the device can display</td>
</tr>
<tr>
<td>device-aspect-ratio</td>
<td>The ratio between the width and the height of the device</td>
</tr>
<tr>
<td>device-height</td>
<td>The height of the device, such as a computer screen</td>
</tr>
<tr>
<td>device-width</td>
<td>The width of the device, such as a computer screen</td>
</tr>
<tr>
<td>grid</td>
<td>Whether the device is a grid or bitmap</td>
</tr>
<tr>
<td>height</td>
<td>The viewport height</td>
</tr>
<tr>
<td>max-aspect-ratio</td>
<td>The maximum ratio between the width and the height of the display area</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>min-color</td>
<td>The minimum number of bits per color component for the output device</td>
</tr>
<tr>
<td>min-color-index</td>
<td>The minimum number of colors the device can display</td>
</tr>
<tr>
<td>min-device-aspect-ratio</td>
<td>The minimum ratio between the width and the height of the device</td>
</tr>
<tr>
<td>min-device-width</td>
<td>The minimum width of the device, such as a computer screen</td>
</tr>
<tr>
<td>min-device-height</td>
<td>The minimum height of the device, such as a computer screen</td>
</tr>
<tr>
<td>min-height</td>
<td>The minimum height of the display area, such as a browser window</td>
</tr>
<tr>
<td>min-monochrome</td>
<td>The minimum number of bits per &quot;color&quot; on a monochrome (greyscale) device</td>
</tr>
<tr>
<td>min-resolution</td>
<td>The minimum resolution of the device, using dpi or dpcm</td>
</tr>
<tr>
<td>min-width</td>
<td>The minimum width of the display area, such as a browser window</td>
</tr>
<tr>
<td>monochrome</td>
<td>The number of bits per &quot;color&quot; on a monochrome (greyscale) device</td>
</tr>
<tr>
<td>orientation</td>
<td>The orientation of the viewport (landscape or portrait mode)</td>
</tr>
</tbody>
</table>

... and many more...

[http://www.w3schools.com/cssref/css3_pr_mediaquery.asp](http://www.w3schools.com/cssref/css3_pr_mediaquery.asp)
Common Logical Operators (AND/OR)

@media (min-width: 768px) and (max-width: 991px) { ... }

Devices within a width range

@media (max-width: 767px) , (min-width: 992px) { ... }

Comma is equivalent to OR

any device that falls within the range of its width being anywhere from 768 pixels until 991 pixels

targeting any device whose width is no larger than 767 pixels or any device that is at least 992 pixels in width
Common approach to structuring media queries

Start your design with base styles

Base styles will apply across the board no matter what screen size you actually are viewing the website on.

Now add your properties for the other devices – by adding or taking away styles

Warning: Do not overlap range boundaries!!
Example: Goldilocks and the Three Bears

```
<style>
  * {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
  }

  /* Base styles */
  h1 {
    margin-bottom: 15px;
  }

  p {
    border: 1px solid black;
    margin-bottom: 15px;
  }

  #p1 {
    background-color: #a52a2a;
    width: 300px;
    height: 300px;
  }

  #p2 {
    background-color: #deb887;
    width: 50px;
    height: 50px;
  }
</style>

<body>
  <h1>Goldilocks & The Three Bears - Media Queries</h1>
  <p id="p1"></p>
  <p id="p2"></p>
</body>
```
What happens if we change the browser width?

<table>
<thead>
<tr>
<th>Goldilocks &amp; The Three Bears - Media Queries</th>
<th>Media Queries</th>
</tr>
</thead>
</table>

Make browser window smaller.... Nothing happens.
Add a @media query for large devices >= 1200px

For devices that are at least 1200px wide!

```css
/** Large devices only **/
@media (min-width: 1200px) {
  #p1 {
    width: 80%;
  }
  #p2 {
    width: 150px;
    height: 150px;
  }
}
```

Target large devices only:
For screens width 1200px or larger
Apply the following styles
(within the braces associated with the media query)

The width of #p1 is now a % of the screen/window size.

Make #p2 triple the size for windows at least 1200px wide.
What happens how if the browser window is resized?
Use chrome developer tools to really see what is going on...

Notice the size of the screen in pixels.
Click on this icon to emulate a device!

Size of the screen: 1373px × 533px
Goldilocks & The Three Bears - Media Queries

Notice the size changes #p1 & #p2
Add code for medium sized devices...

```html
/********** Medium devices only **********/
@media (min-width: 992px) and (max-width: 1199px) {
  #p1 {
    width: 50%;
  }
  #p2 {
    width: 100px;
    height: 100px;
  }
}

Don’t overlap with previous media query.

https://scotch.io/quick-tips/default-sizes-for-twitter-bootstraps-media-queries
```
Goldilocks & The Three Bears - Media Queries

+1200px
Goldilocks & The Three Bears - Media Queries

```html
<html>
<head>
    <meta charset="utf-8">
</head>
<body>

<style>
    element.style {
        // Your styles here
    }
</style>
</body>
</html>
```

between 992px and 1199px
Goldilocks & The Three Bears - Media Queries

less than 992px
Visualizing the breakpoints using Chrome dev tools

Goldilocks & The Three Bears
Basic syntax of a media query
- @media (media feature)
- @media (media feature) logical operator (media feature)

Remember not to overlap breakpoints

Usually, you provide base styling
- Then, change or add to them in each media query