

# CSS Media Queries

all the code is in this [archive folder](#)

- Media queries allow you to define conditional CSS rules
- These rules will only take effect while a condition is true, though normal cascade behavior still applies, so if two conditional rules conflict, the latter rule will be applied
- The conditional statements in media queries have to do with the capability of the browser and/or the device used to display the webpage

- A media query is declared as part of a CSS document as follows:

```
@media conditions { ...CSS rules... }
```

# media types

- Media types detect the device used to display a page
- There are available queries for projectors, brail readers, text-to-speech screen readers, printed media, and so on
- We are interested in two specific, very common types – **screen**:

**@media screen**

- and **print**:

**@media print**

# device dimensions

- You can query the dimensions of a given device. This is the easiest way to detect specific mobile phones

**@media (device-width: 640px)**

- You can also add max and min statements to detect devices in a range

**@media (min-device-width: 640px)**

- **device-width**
- **device-height**
- **min-device-width**
- **min-device-height**
- **max-device-width**
- **max-device-height**

# browser width

- On many devices, you can resize your browser window or viewport
- To detect the current browser viewport size (in real time) use the width and height statements

**@media (width: 640px)**

- The width and height conditions are different than device dimensions because they query the current size of the browser, not the size of the device (which is fixed)

- **width**
- **height**
- **min-width**
- **min-height**
- **max-width**
- **max-height**



# multiple conditions

- You may combine multiple **conditions** using the **and** keyword  
**@media screen and (device-width: 640px) and (device-height: 960px)**
- This query will only work on iPhone 4 and 4s, but no later devices.

# useful CSS properties

- It is often useful to redeclare container sizes as browser dimensions change:

**width, max-width**

- It is also prudent to redefine float behavior

**float: none;**

- Or make specific elements appear or disappear

**display: block;** or **display: inline;** or **display:none**

# the content property

- Using :after or :before selectors you can add pseudo elements following or preceding the styled element
- The content property will allow you to insert text into the pseudo element – static text must be surrounded by quotation marks ("")

**content: " over 800 pixels wide";**

- It is also possible to insert the attributes of the styled element into the pseudo element using the attr value followed by the attribute name in brackets

**content: attr(title)**

- You can combine attr values with static text using the "+" sign.

```
<span class="screenwidth">The screen is</span>
```

```
@media (min-width:800px){  
.screenwidth:after {content:" over 800 pixels wide";}  
}
```

# demo (resize the browser)

- no media query: [example](#)
- with media-query: [example](#)
- business card: [example](#)

# example: column layouts

- [link](#)
- changes on browser resize

# example: print

- colorful page
- changes when asked to print: file → print preview