

Responsive Web Design Continued...

RWD I1

Today's agenda

- Recap
- Multiple breakpoints
- The Two Viewports
- Mobile Browsers
- Screen orientation
- Responsive Layout
- Exercise

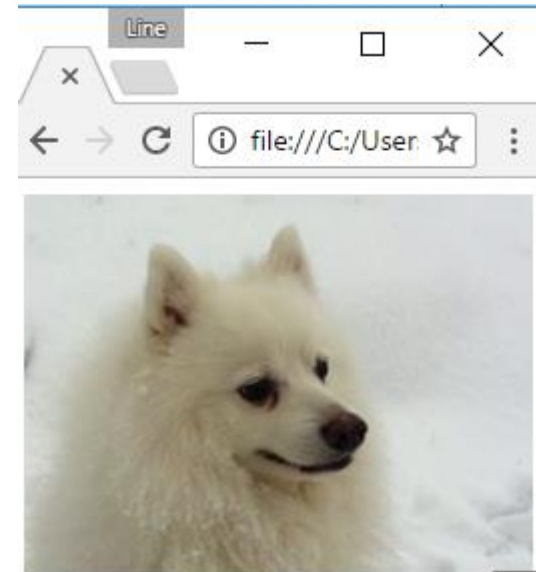
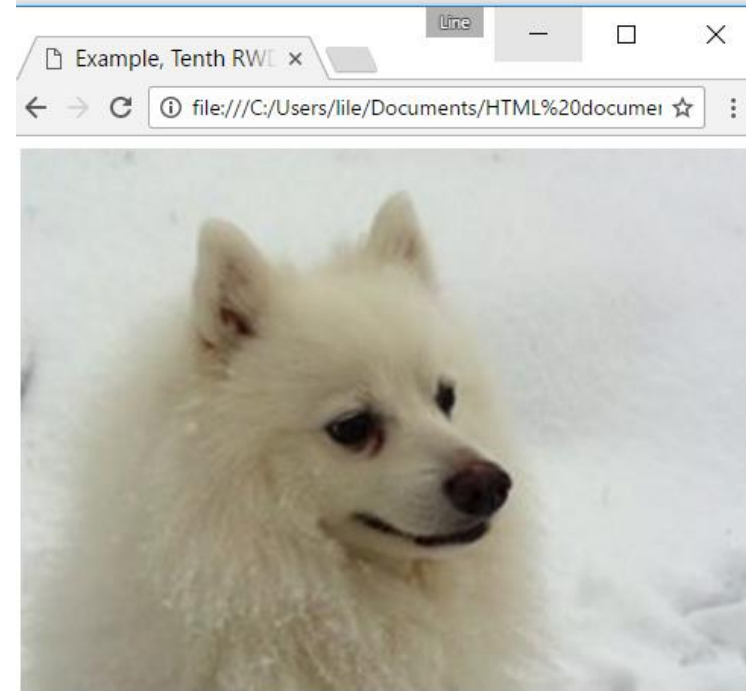
Exercises from last time

Questions

Recap: Relative Size: height

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

Element size is set relative to size of parent (in this case the browser window view/the `body` element).



RW

Recap: Relative Size: divisions

HTML:

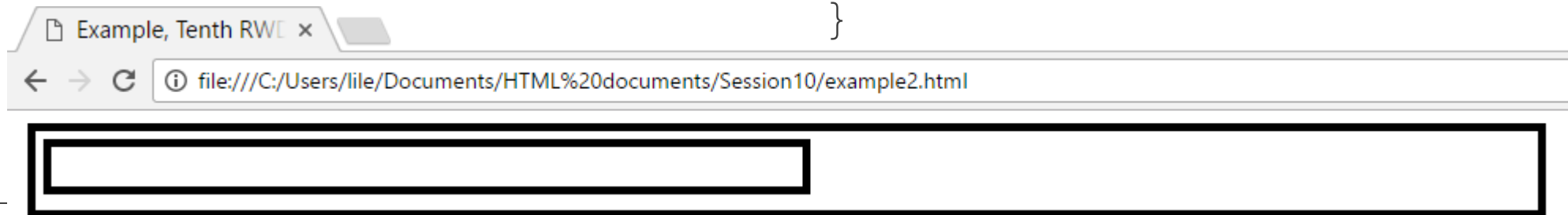
```
<body>
  <div id="firstBox">
    <div id="secondBox">

      </div>
    </div>
  </body>
```

CSS:

```
#firstBox {
  width: 50%;
  height: 50px;
  margin: 5px;
  border: 5px solid black;
}

#secondBox {
  width: 50%;
  height: 50%;
  margin: 5px;
  border: 5px solid black;
}
```



Recap: Media Query (responsive)

Change the CSS styling to adjust to the client media (screen resolution/window size).

Select different stylesheets depending on media:

```
<link rel="stylesheet" media="mediatype and|not|only (media feature)"  
href="myStylesheet.css" type="text/css">
```

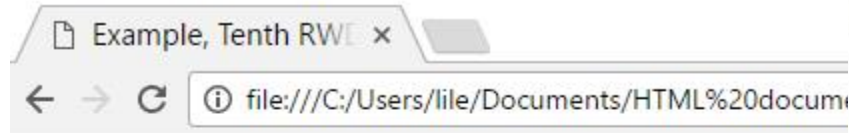
Or manipulate the CSS Styles directly:

```
@media mediatype and|not|only (media feature) {  
    CSS-Code;  
}
```

Recap: Media Query – In the CSS

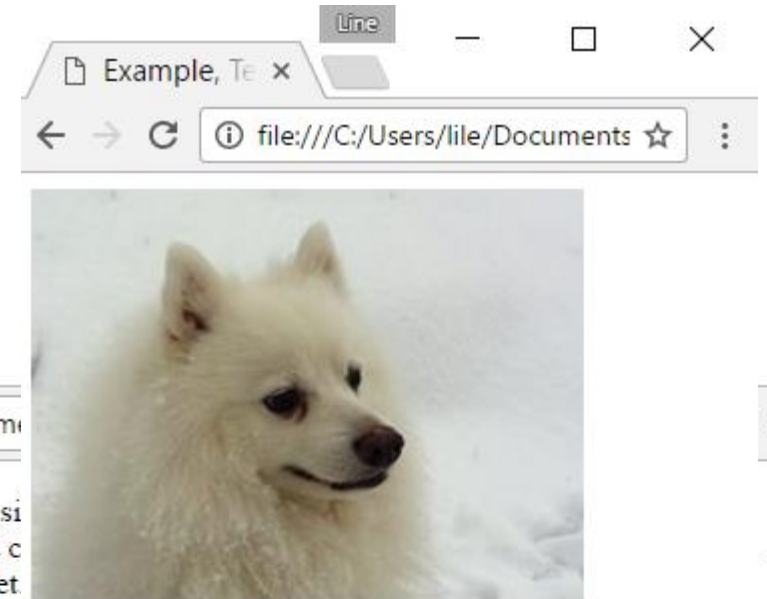
```
@media screen and (min-width: 800px) {  
  img {  
    float: left;  
  }  
  ...  
}
```

Breakpoint



Lorem ipsum dolor si
Praesent libero. Sed c
elementum imperdiet.
sed augue semper por
Class aptent taciti soc
himenaeos. Curabitur

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. Fusce nec tellus sed augue semper porta.
Mauris massa. Vestibulum lacinia arcu eget nulla.
Class aptent taciti sociosqu ad litora torquent per
conubia nostra, per inceptos himenaeos. Curabitur
sodales ligula in libero. Sed dignissim lacinia nunc.



To be Responsive or not to be Responsive

Try out these WEB sites for Life Style Magazines

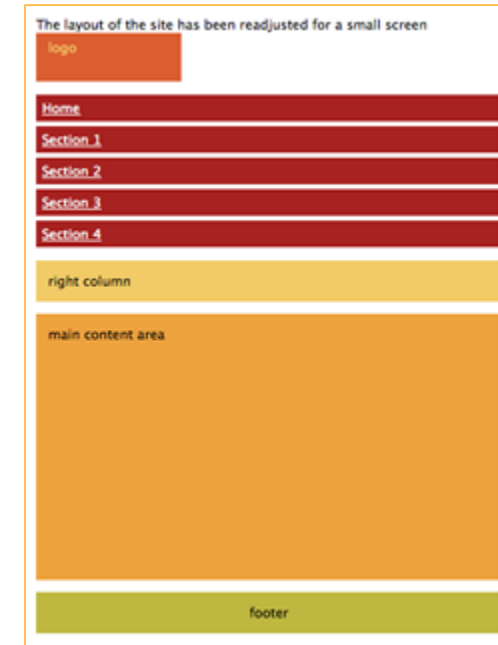
<http://www.lifestylesmagazine.com/website/>

<http://www.lifeandstylemag.com/>

Desktop/Mobile



Desktop Version



Mobile Version

Where to Set Breakpoints

- Your instinct might be to write media query breakpoints around common screen resolutions, such as 320px, 480px, 768px, 1024px, 1224px, and so forth,
 - This is a bad idea
- When building a responsive website it should adjust to an array of different window sizes, regardless of the device. Breakpoints should **only** be introduced when a website starts to break, look weird, or the experience is being hampered
- Additionally, new devices and resolutions are being released all the time. Trying to keep up with these changes could be an endless process

Multiple Breakpoints

```
@media screen and (max-width: 480px) {  
  ...  
}
```

```
@media screen and (min-width: 481px) and (max-width: 1024px) {  
  ...  
}
```

```
@media screen and (min-width: 1025px) {  
  ...  
}
```

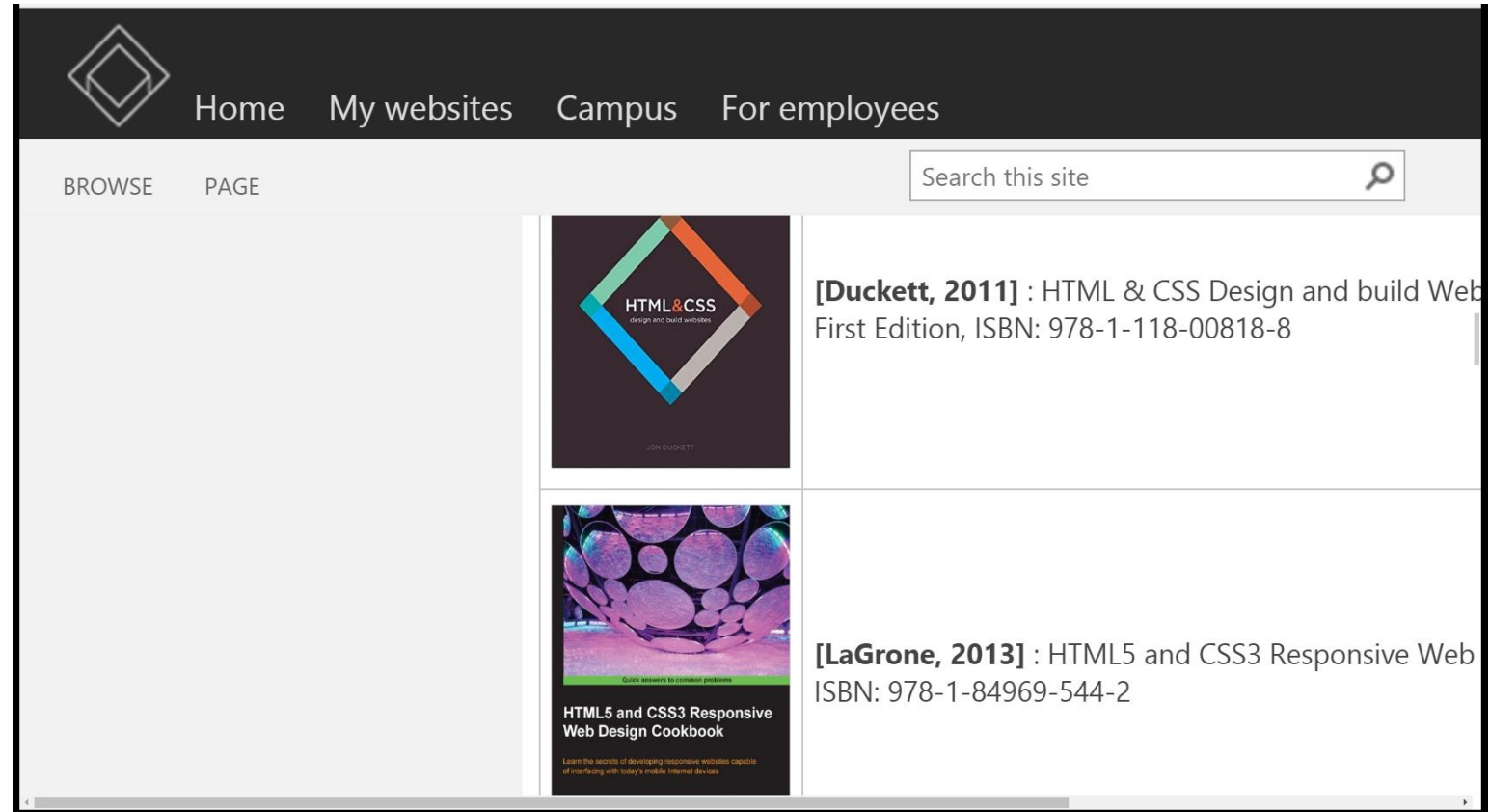
The two viewports

Instead of trying to show the whole webpage on the screen, browsers operate with two viewports instead:

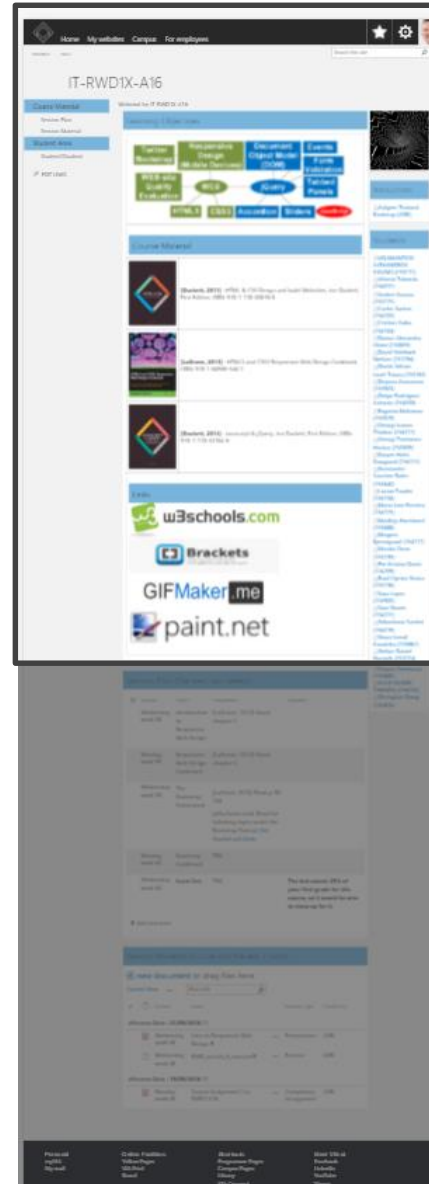
- The Visual Viewport
 - The part of the page that is currently shown on-screen.
 - Same size as the browser window.
- The Layout Viewport
 - Can be considerably bigger than the visual viewport.
 - Contains elements that could not fit into the browser window due to pixel width limitations.

The two viewports

Unless the webpage contains elements with CSS pixel sizes that have greater width than the visual viewport or the user zooms in, the layout viewport will only extend in the vertical direction.



The two viewports

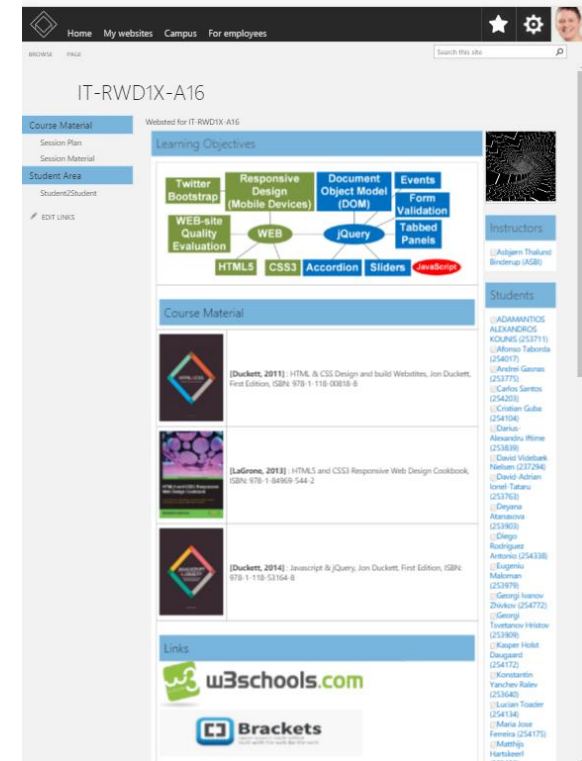
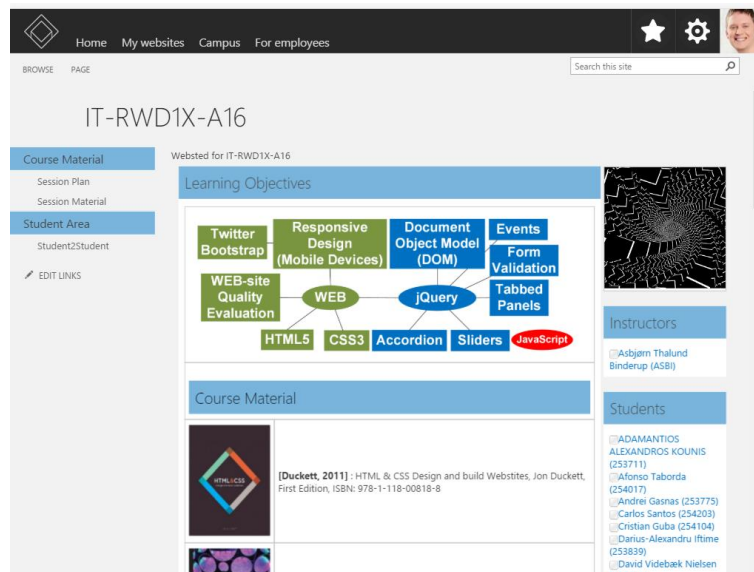


The Visual Viewport

The Layout Viewport

Mobile and Tablet Browsers

- Has far less device pixels than a laptop/desktop screen.
- Can quickly switch horizontal/vertical orientation.



Screen orientation

```
@media screen and (max-width: 680px) and (orientation:  
landscape) {
```

```
...
```

```
}
```

```
@media screen and (max-width: 680px) and (orientation:  
portrait) {
```

```
...
```

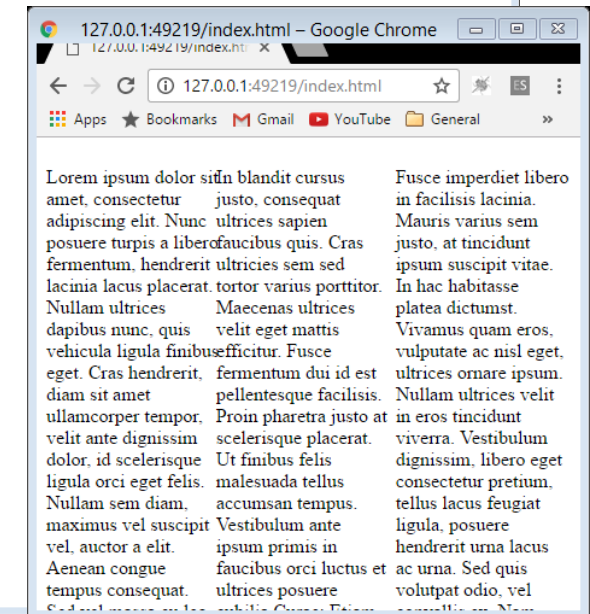
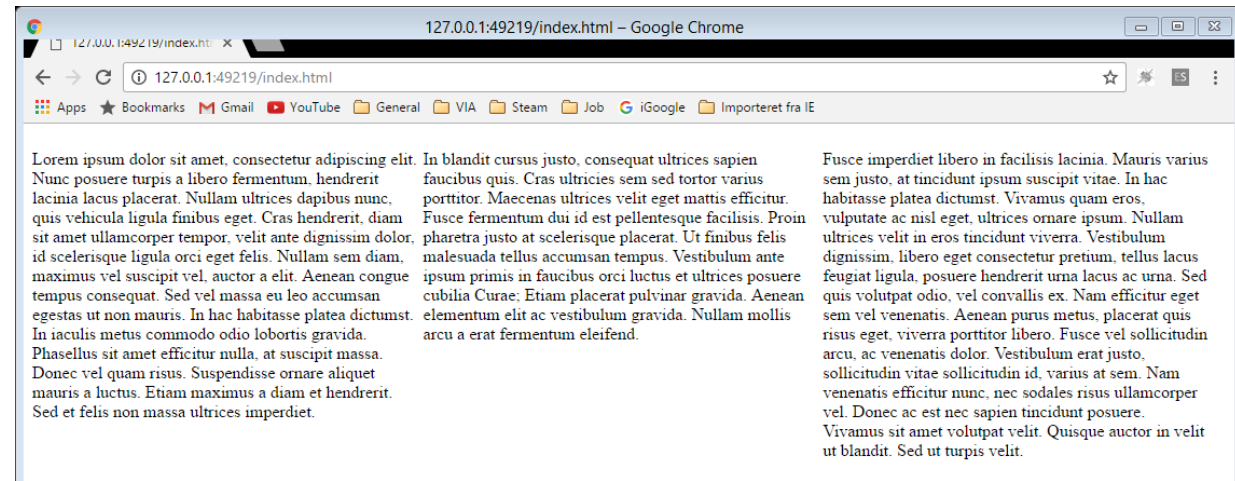
```
}
```


Responsive Positioning

```
p {  
  width: 32%;  
  position: fixed;  
}
```

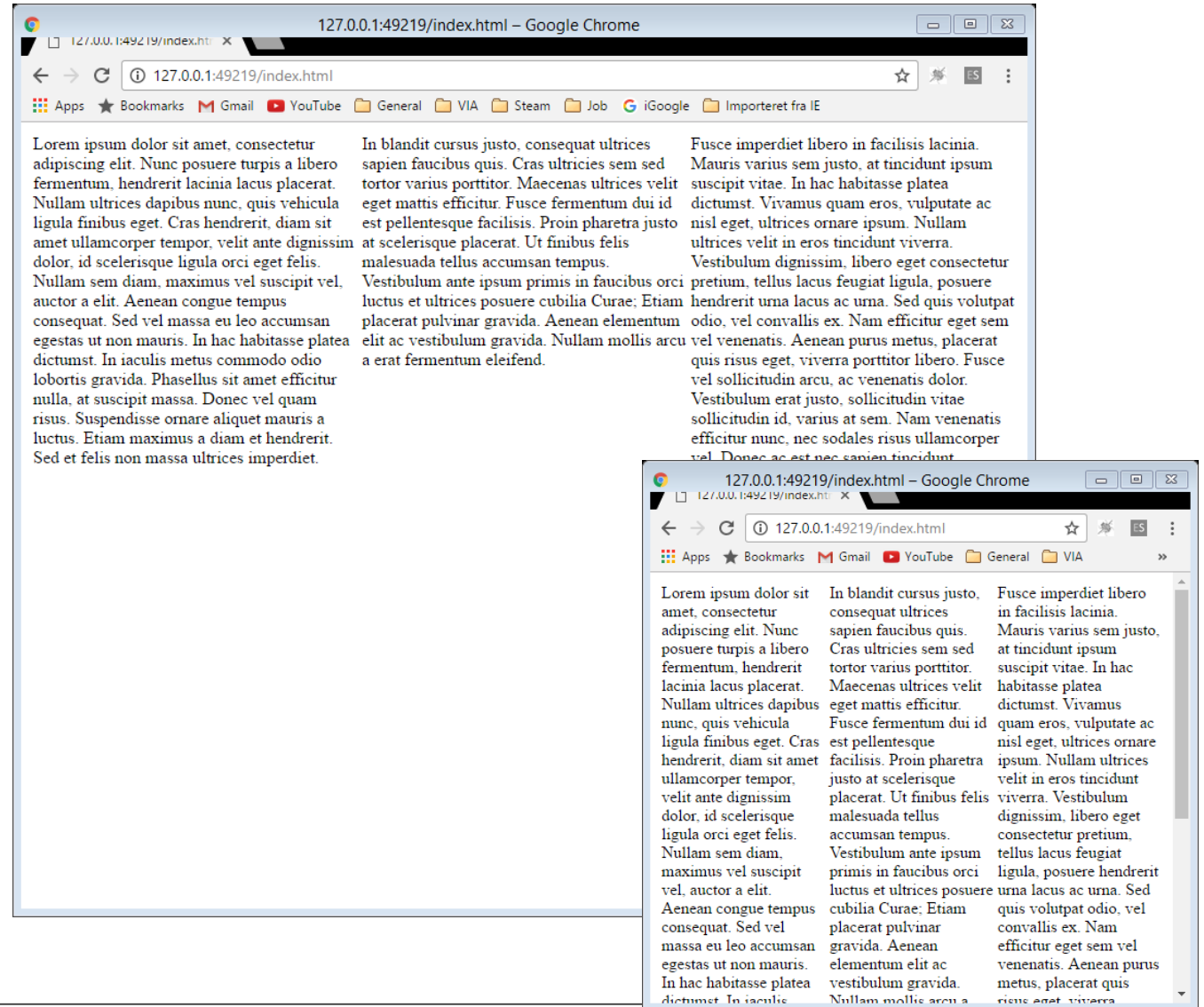
```
#paragraph2 {  
  left: 33%;  
}
```

```
#paragraph3 {  
  left: 66%;  
}
```



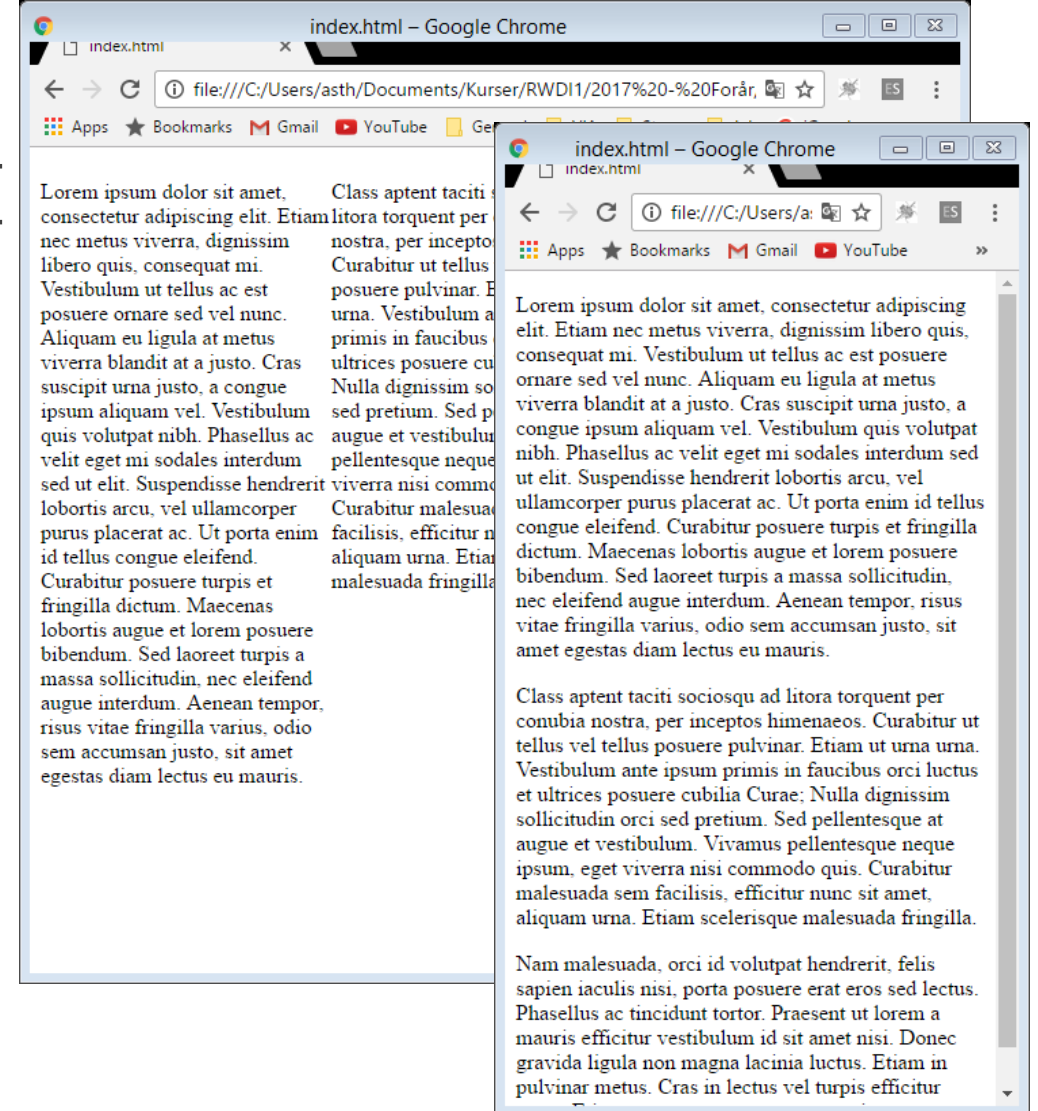
Responsive Table

```
td {  
    width: 32%;  
    vertical-align: top;  
}
```



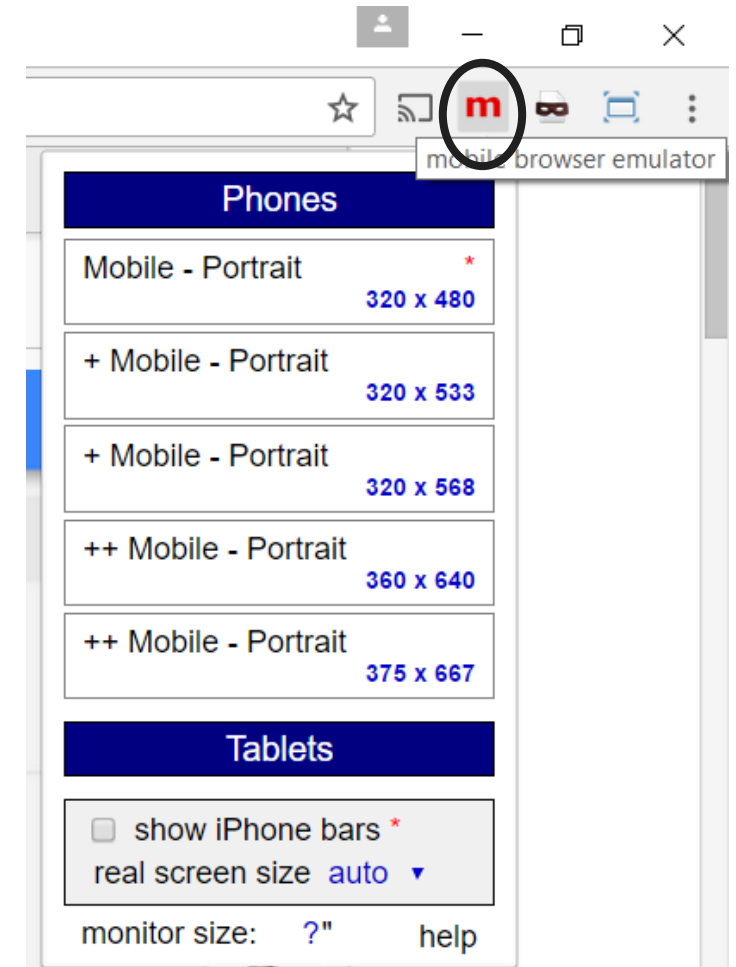
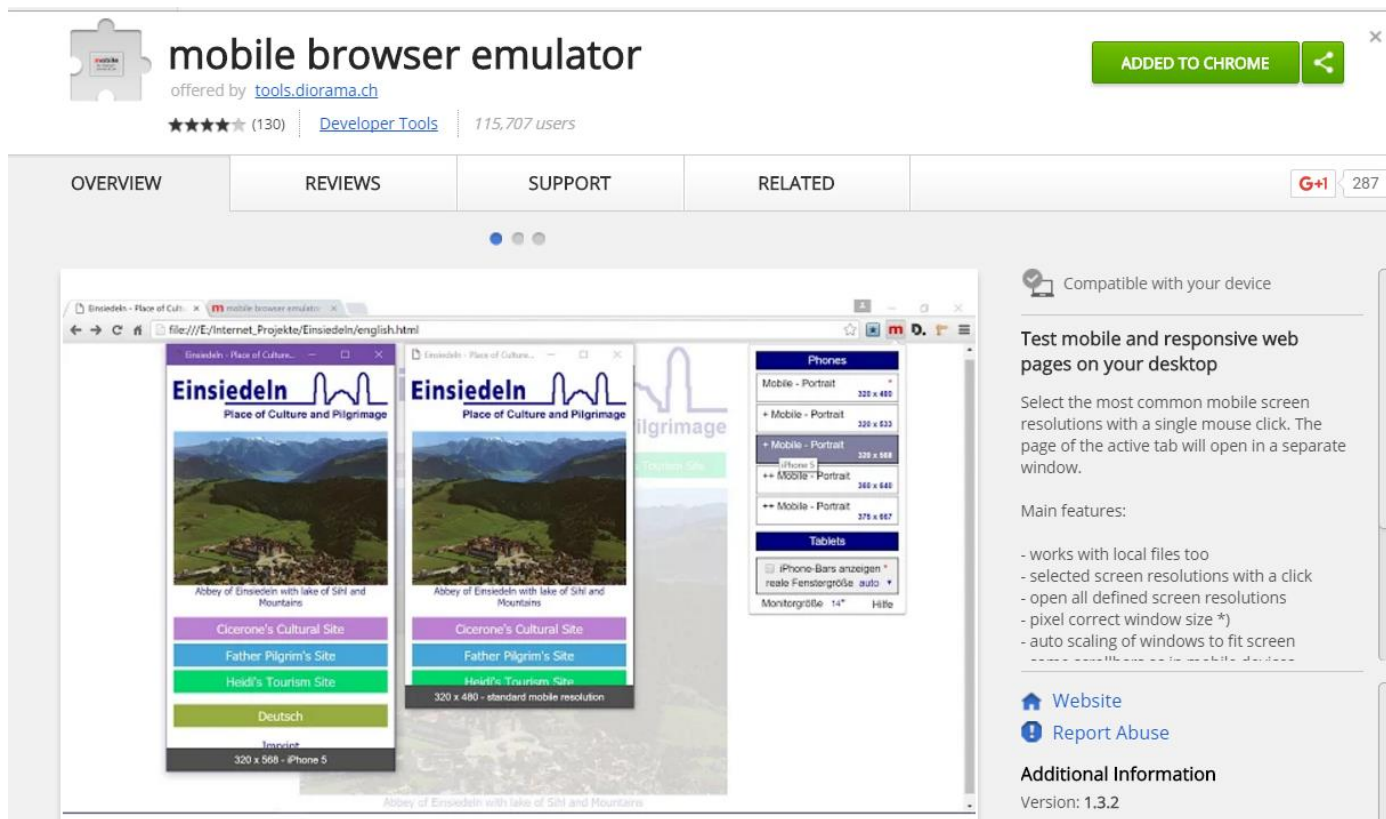
Responsive Float

```
@media screen and (min-width: 600px) {  
  p {  
    width: 32%;  
    float: left;  
  }  
}
```



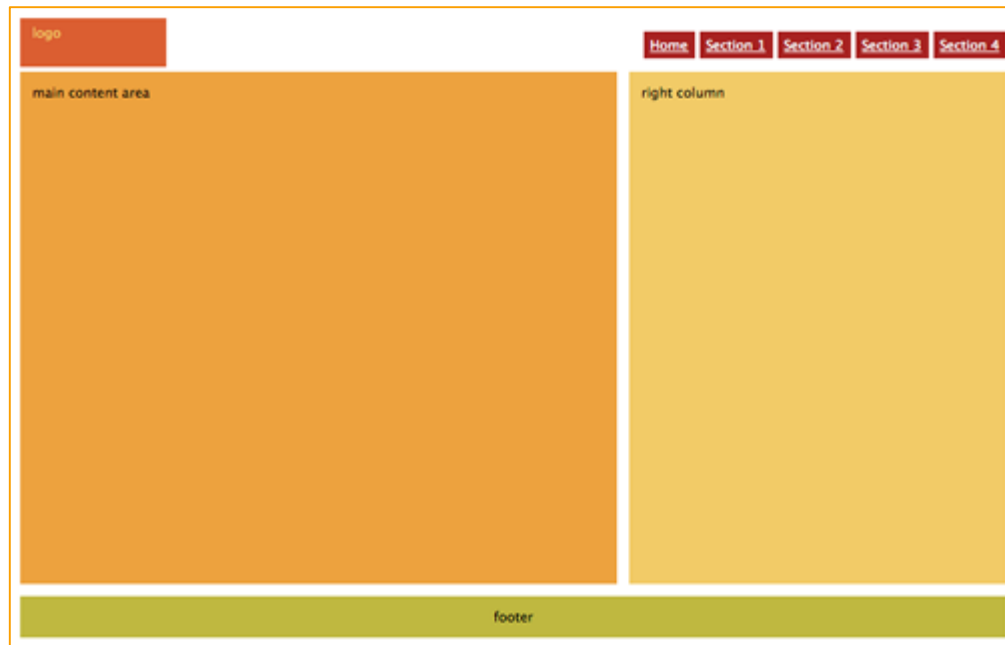
The mobile device emulator in Chrome

<https://chrome.google.com/webstore/detail/mobile-browser-emulator/lbofcampnkJmiomohpbaihdcbjhbfepf>

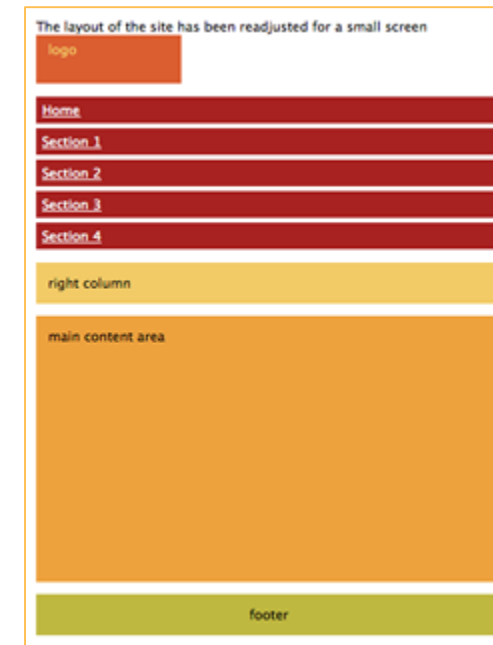


Exercises

- Make layouts that look something like this:



Desktop Version



Mobile Version