

Intro to Responsive Web Design

RWD I1

Today's Agenda

- Setting size for elements
- What is screen size?
- Media query
- Box-sizing
- Exercise

Pop Quiz



What is Responsive Web Design?

A technique that makes it possible to create webpages that can be shown with good overview and no need to scroll horizontally on the page on different devices.

- Screen size
- Screen orientation
- Media: Screen, TV, Print etc.

Notice that the layout of menus and contents change.

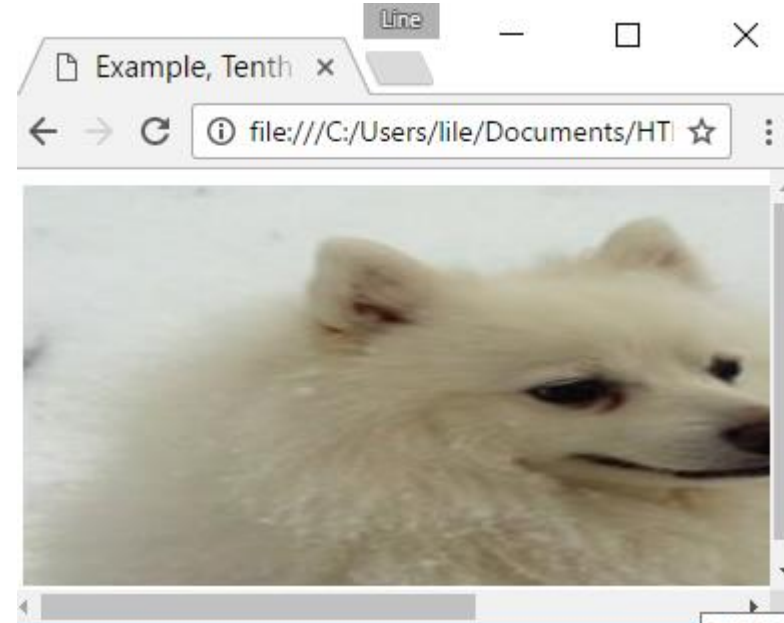
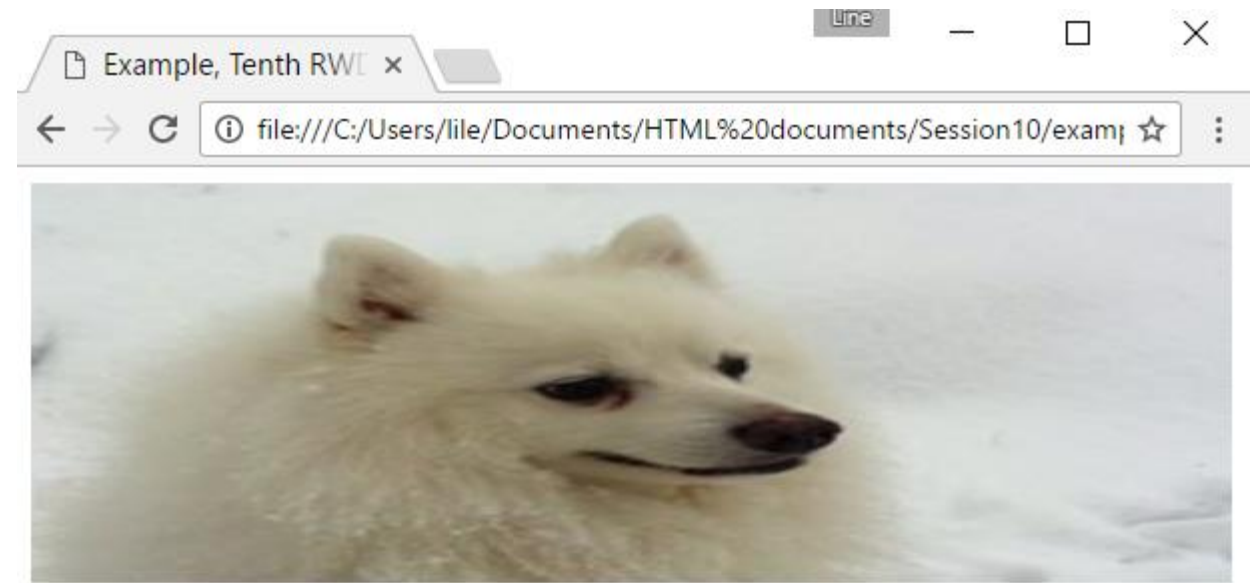
It is the same HTML code, everything is done in CSS.



Fixed Size Elements

```
img {  
  width: 600px;  
  height: 200px;  
}
```

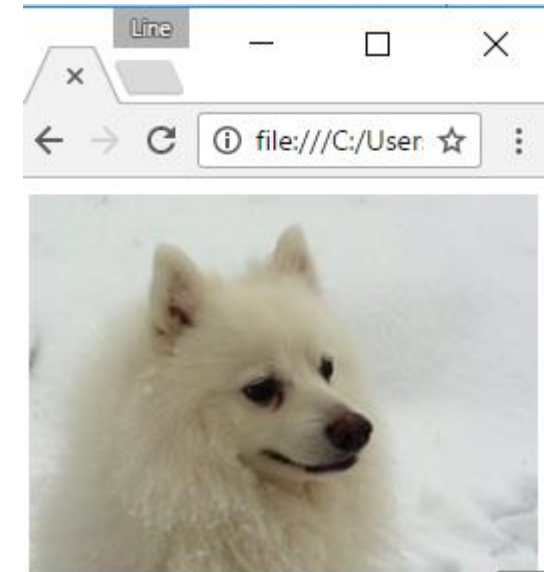
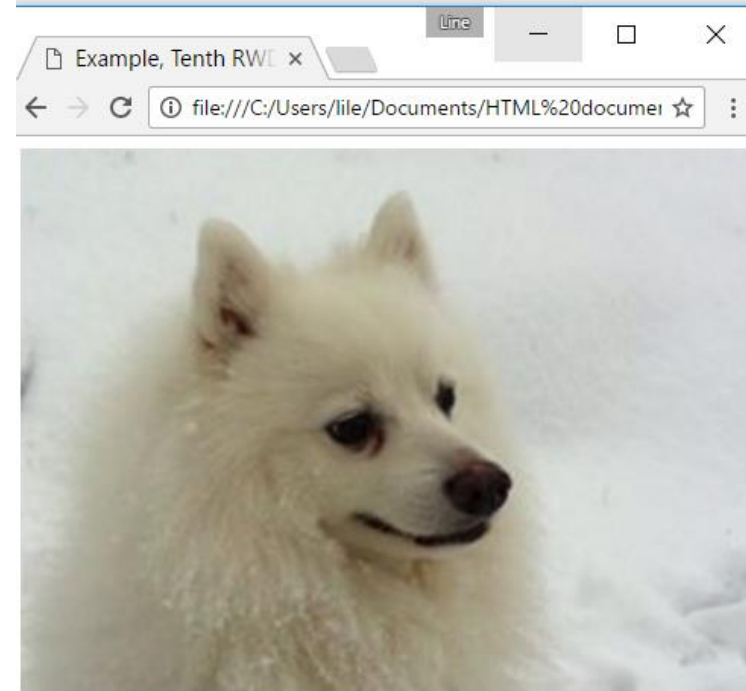
Element size is set regardless of the size of parent and children elements.



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

Relative Size: width

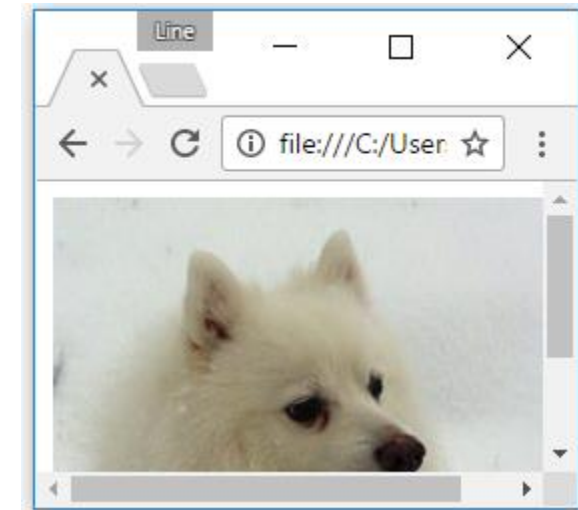
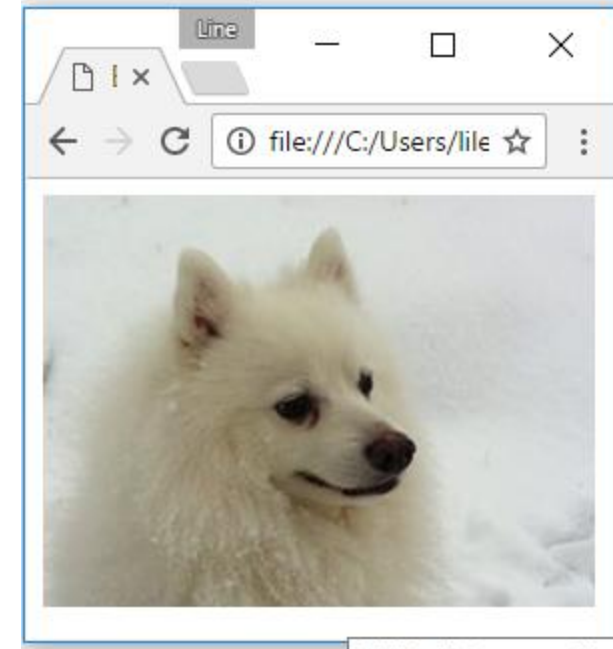
```
img {  
  width: auto;  
  height: 100%;  
}
```

Does not work when the parent element is the browser view/the **body** element, but otherwise will.

What we see here is the default sizing.

To fix add:

```
html, body {  
  height: 100%;  
}
```

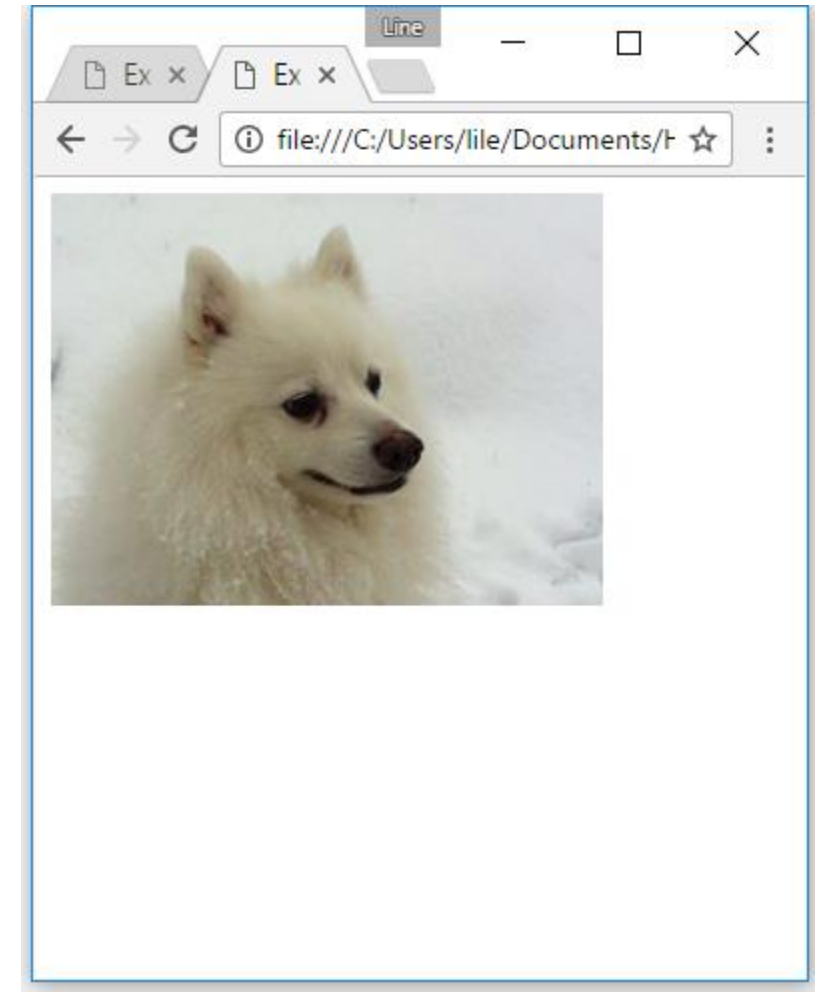
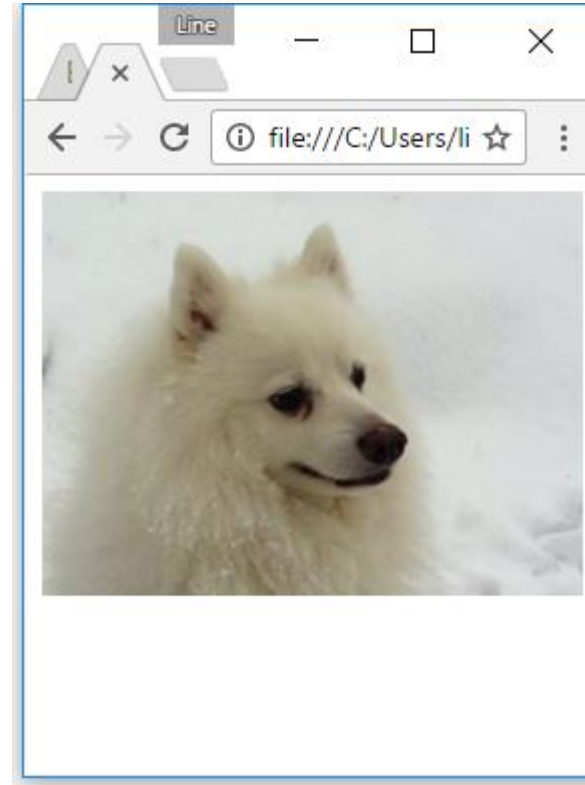


Relative Size: max-width

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

Element size is set relative to size of the child element (the original pixels size of the image) or downsized if the parent element is smaller.

This only works on images though.



Relative Size: divisions

HTML:

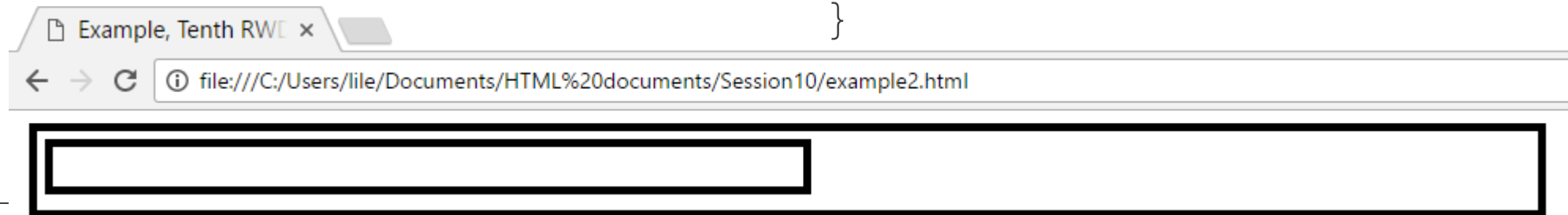
```
<body>
  <div id="firstBox">
    <div id="secondBox">

    </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;
}
```



Relative Size

Size is generally always inherited outward to inward.

`<body>` ← Size is relative to screen/window resolution and size

`<div>` ← Size is relative to parent

`` ← Size is relative to parent

`</div>`

`</body>`

What is Screen Size?

Screen Width/Height

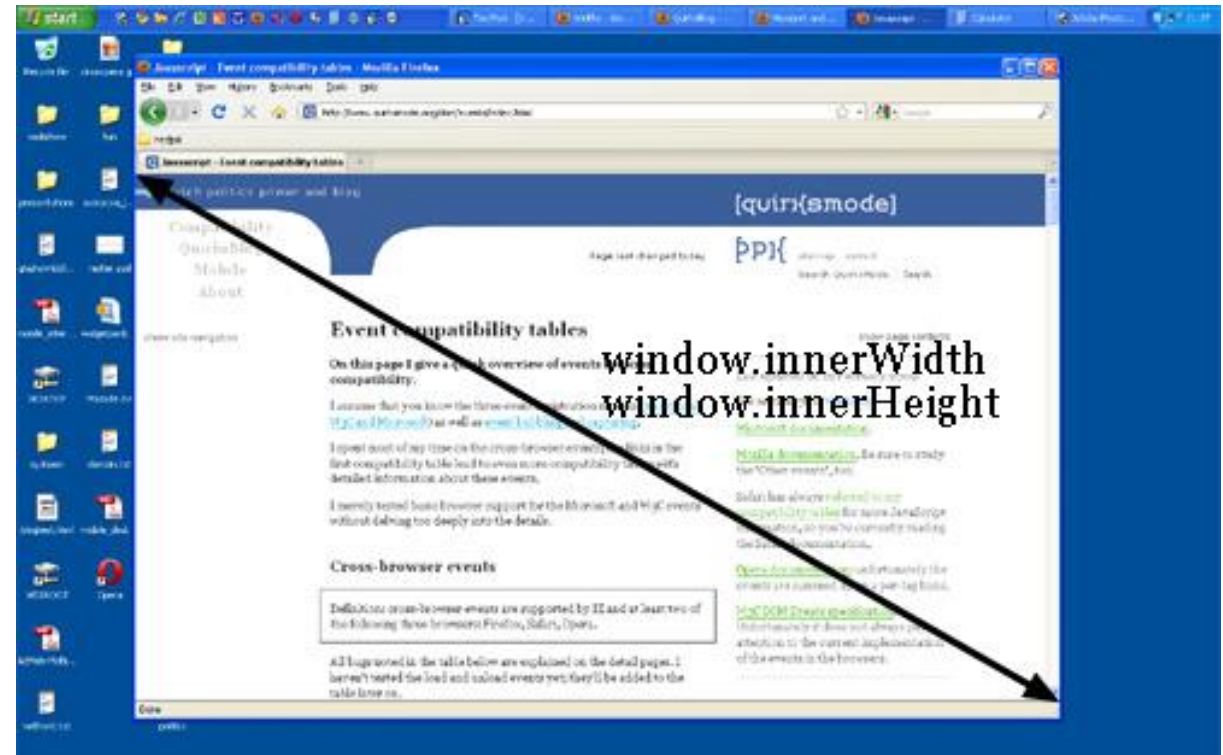
Physical pixels Horizontal and Vertical of whole screen.

Not very important for a web developer.



Window Width/Height

- CSS pixels Horizontal and Vertical of the available window
- Very important for a web developer
- The space/size that can be used

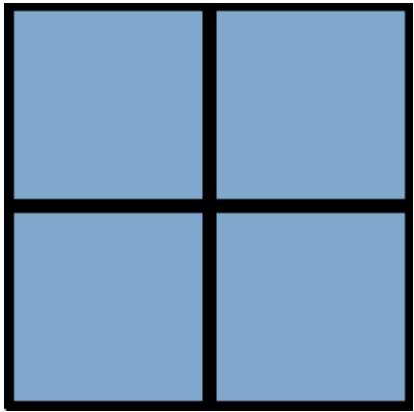


A pixel is not a Pixel

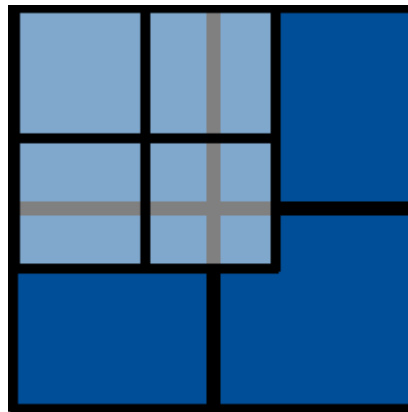
- Device pixels
 - Physical pixels on screen
- CSS pixels
 - Element pixels
- Zooming
 - Stretch CSS pixels over more Device pixels
 - Doesn't change element size
 - Zooming to 200% makes one CSS pixel grow to four times the size of one device pixels
 - At zoom level 100% one CSS pixel is exactly equal to one device pixel

The point here is that ***you are only interested in CSS pixels***
It's those pixels that dictate how your style sheet is rendered

Zooming



100% Zoom
CSS pixels fully
overlaps device
pixels



Zoom out
CSS pixels shrink
one device pixel
overlaps several
CSS pixels



Zoom in
CSS pixels grow
one CSS pixel
overlaps several
device pixels

CSS3 @media rule

- The @media rule is used to define different style rules for different media types/devices
- Media queries look at the capability of the device, and can be used to check many things, such as:
 - width and height of the viewport (the browser window)
 - width and height of the device
 - orientation (is the tablet/phone in landscape or portrait mode?)
 - resolution
 - and much more

See more here:

http://www.w3schools.com/cssref/css3_pr_mediaquery.asp

Media Query – How To Make Things Really 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;  
}
```

Media Query – Select Different Stylesheets

```
<head>
  <link rel="stylesheet" media="screen and (max-width: 480px)"
    href="css/smallStylesheet.css" type="text/css">
  <link rel="stylesheet" media="screen and (min-width: 481px)"
    href="css/largeStylesheet.css" type="text/css">
  ...
</head>
```

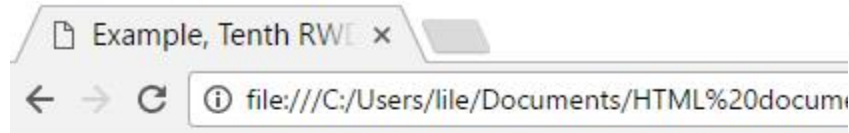
Breakpoints

When **mediatype** is **screen** and **media feature** is **max-width/min-width**, you are comparing against browser window size and not the actual device size (to do that, use **max-device-width/min-device-width** as **media feature** instead).

Media Query – In the CSS

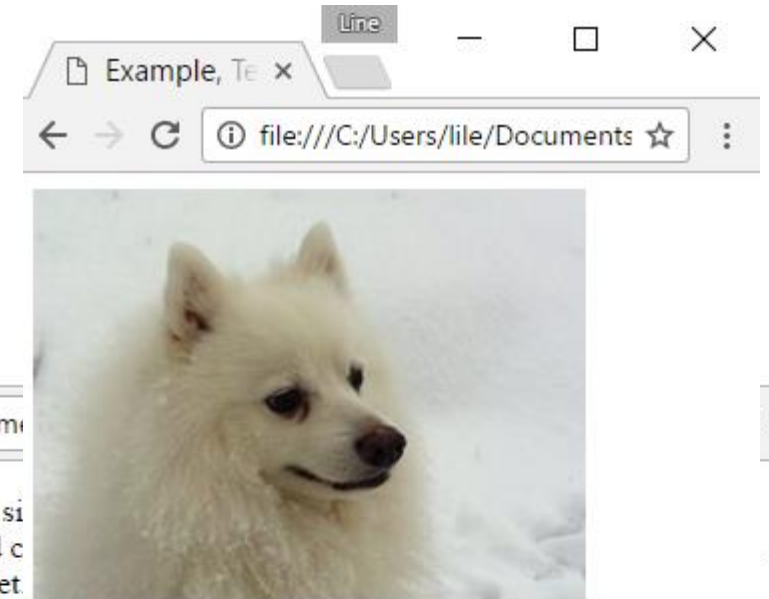
```
@media screen and (min-width: 800px) {  
  img {  
    float: left;  
  }  
  ...  
}
```

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.



Media Query: Hide Elements

Elements like divisions, images etc. can be hidden (not displayed).

```
@media screen and (max-width: 800px) {  
  img {  
    display: none;  
  }  
  ...  
}
```

```
@media screen and (min-width: 801px) {  
  img {  
    display: block;  
  }  
}
```

Box-sizing

What part of the element does **width** and **height** refer to?

```
* {  
  box-sizing: content/border-box;  
}
```

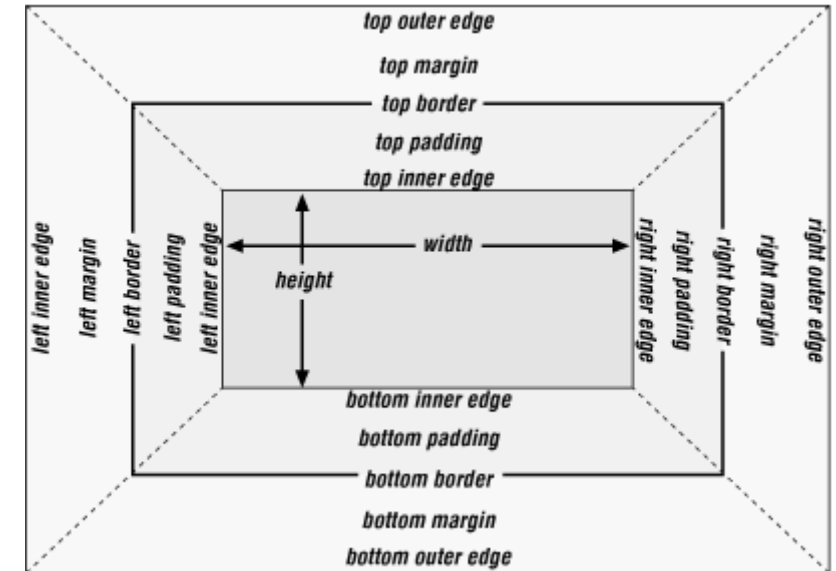
content-box

Default value. The width and height properties (and min/max properties) includes only the content. Border, padding, or margin are not included

border-box

The width and height properties (and min/max properties) includes content, padding and border, **but not the margin**

- makes it easier to scale correctly



Box-sizing Example

```
img {  
  width: 10%;  
  height: auto;  
  background-color: greenyellow;  
  border: 8px solid black;  
}  
  
#image2, #image4 {  
  padding: 8px;  
}  
  
#image3, #image4 {  
  box-sizing: border-box;  
}
```



Exercises

- Find the exercises on studienet (HTML and CSS file)
- Try out basic responsive features