

Responsive Web Design

Kevin Bluer



JS



Agenda

Agenda

- What is RWD?
- Adaptive vs Responsive
- Common Techniques
 - Media Queries
 - ViewPort
 - Frameworks
- Chrome Support
- Summary

What is RWD?

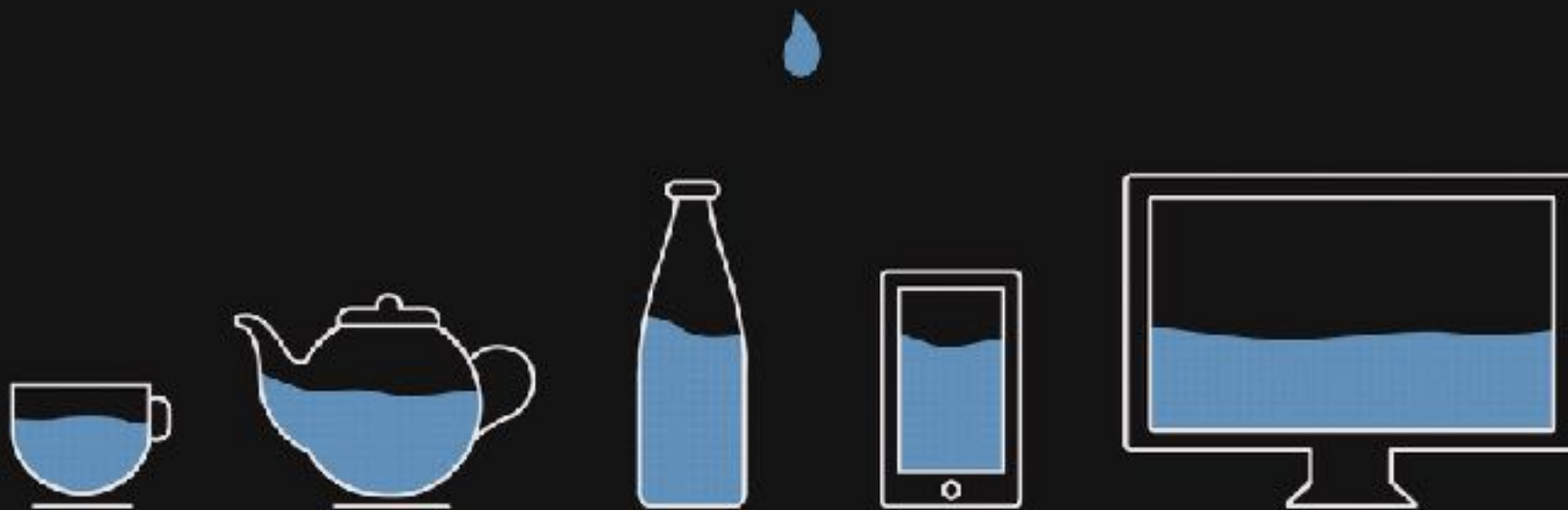
RWD, huh?

- © “Responsive web design (RWD) is an approach to web design aimed at crafting sites to provide an optimal viewing and interaction experience—easy reading and navigation with a minimum of resizing, panning, and scrolling—across a wide range of devices (from desktop computer monitors to mobile phones).”

RWD, huh?



CONTENT IS LIKE WATER



“ You put water into a cup it becomes the cup.
You put water into a bottle it becomes the bottle.
You put it in a teapot, it becomes the teapot. ”

Josh Clark (originally Bruce Lee) - Seven deadly mobile myths

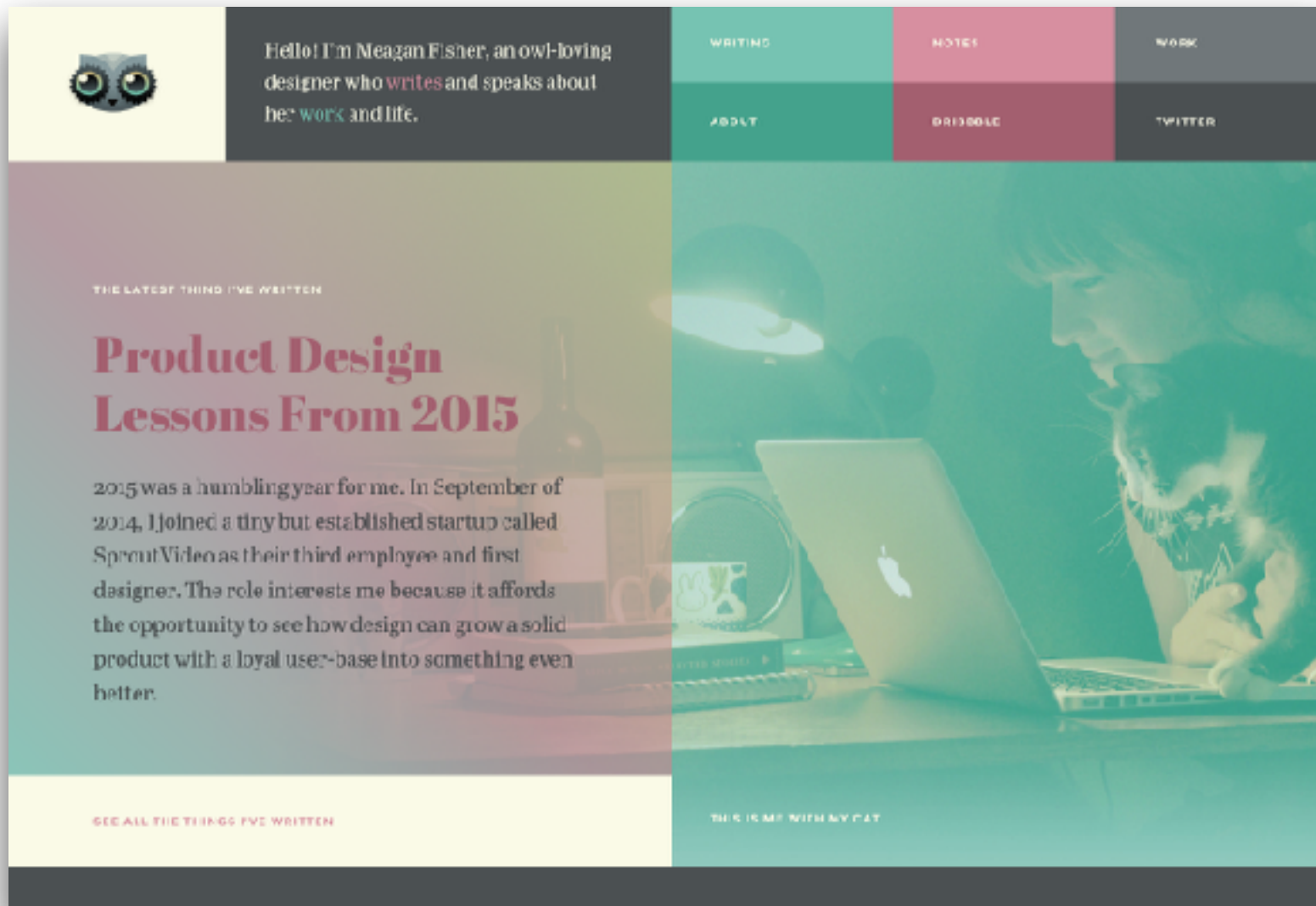
Illustration by Stéphanie Walter

Adaptive vs Responsive

- What is **adaptive** (fixed break points) design?
 - Designing for specific resolutions
 - Useful when adapting an existing website
- What is **responsive** (fluid grids, etc) design?
 - Creating interfaces that adapt to the user's capabilities (in terms of both form and function)
- Of course a combination is completely fine :)

Example

Example



© <http://owltastic.com>



It can even include print...



Common Techniques

Common Techniques

- Media queries
- Responsive images
- Content overflow
- Relative sizing (em, rem, vh, vw, etc)
- Fluid grids
- Adjusting the viewport metatag
- Responsive frameworks (such as Bootstrap)

Media Query Examples

- Styles here only apply when the screen is between 320 and 480px wide
- Styles here apply when the screen is at least 321px wide
- Styles here apply when the screen is at most 320px wide

```
@media only screen  
and (min-device-width : 320px)  
and (max-device-width : 480px) {  
  /* Styles */  
}
```

```
/* Smartphones (landscape) */  
@media only screen  
and (min-width : 321px) {  
  /* Styles */  
}
```

```
/* Smartphones (portrait) */  
@media only screen  
and (max-width : 320px) {  
  /* Styles */  
}
```

Responsive Images

- Using width

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

- Background Images (with “background-size”)

```
div {  
  width: 100%;  
  height: 400px;  
  background-image: url('img_flowers.jpg');  
  background-size: cover;  
  border: 1px solid red;  
}
```

Content Overflow

- © The **overflow** property determines what happens if it's content overflows the element's container

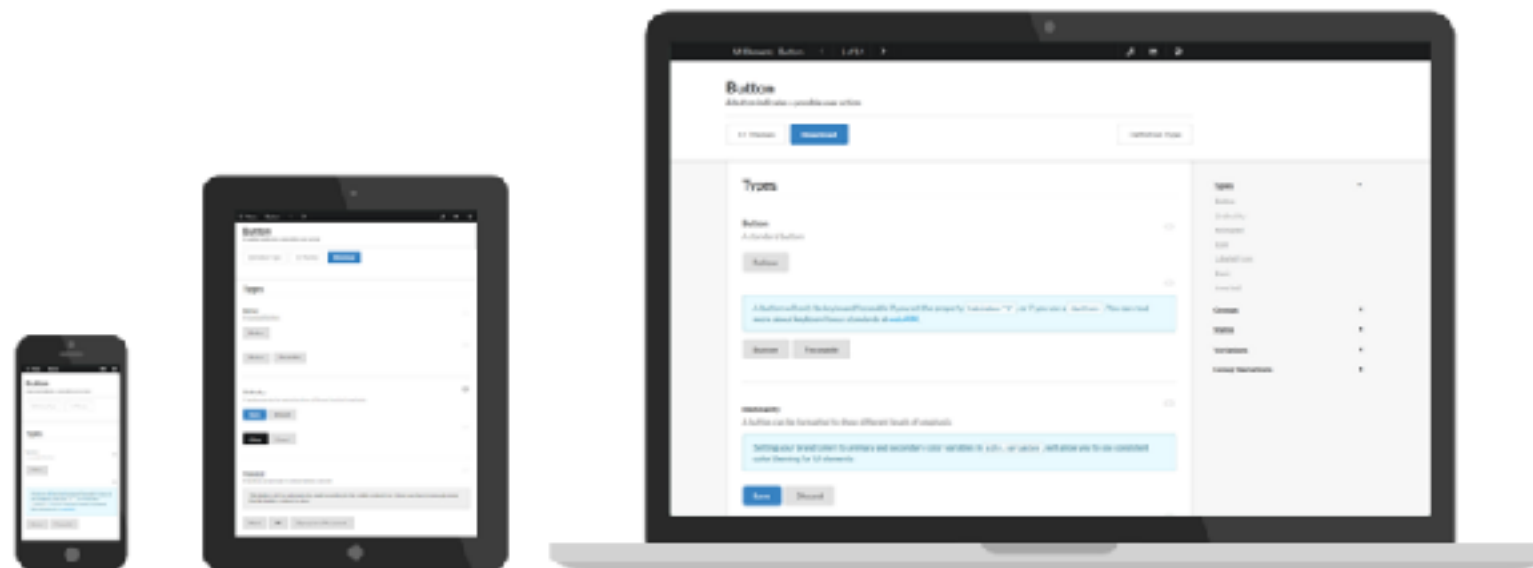
Value	Description
visible	The overflow is not clipped. It renders outside the element's box. This is default
hidden	The overflow is clipped, and the rest of the content will be invisible
scroll	The overflow is clipped, but a scroll-bar is added to see the rest of the content
auto	If overflow is clipped, a scroll-bar should be added to see the rest of the content
inherit	Specifies that the value of the overflow property should be inherited from the parent element

Viewport

- The viewport is the user's visible area of a page
- The viewport `<meta>` tag provides instructions on how to control the page's dimensions and scaling
- 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

Responsive Frameworks

- Frameworks such as Bootstrap, Foundation, Semantic UI, and Material Design all have built in support

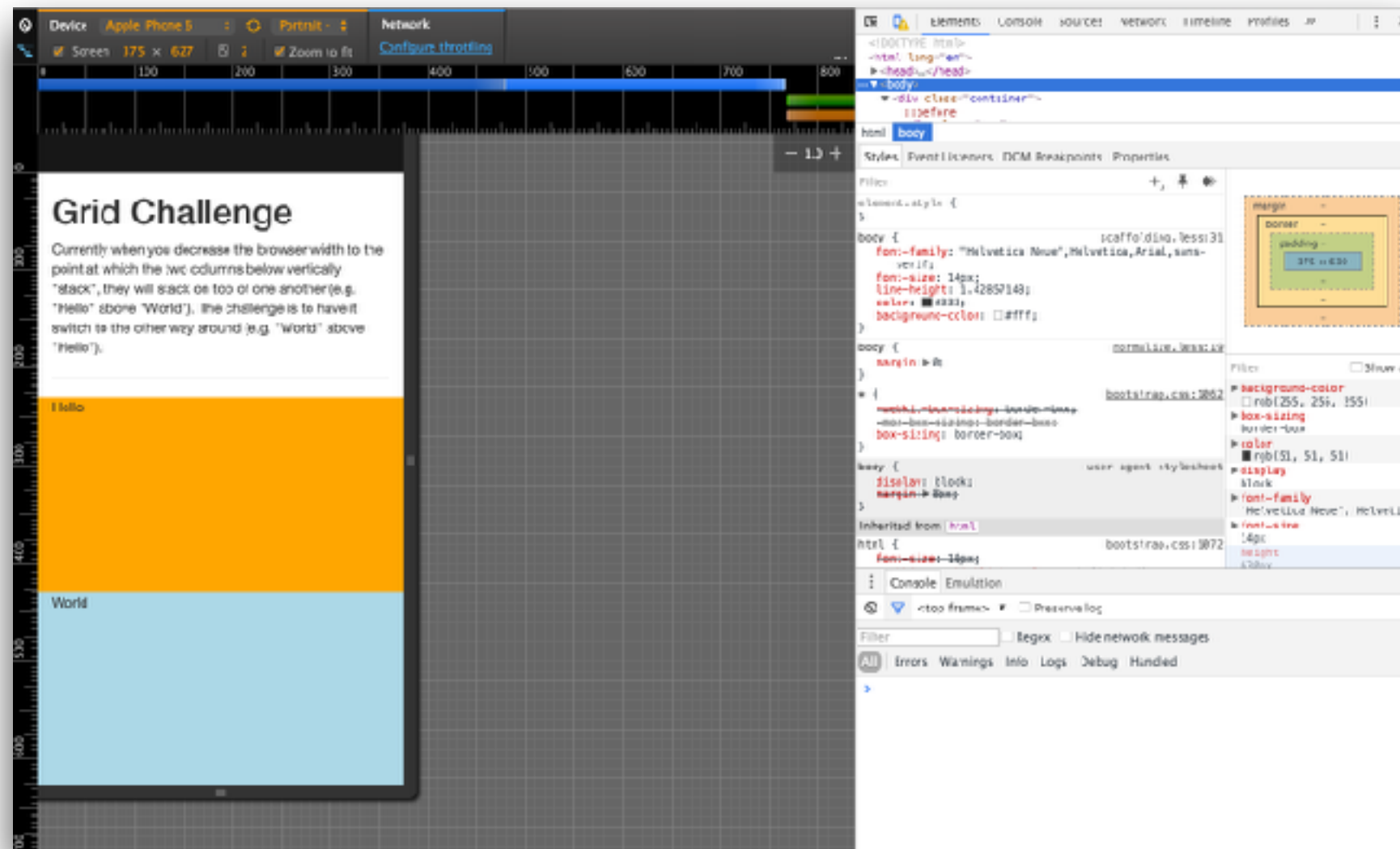





Chrome Support

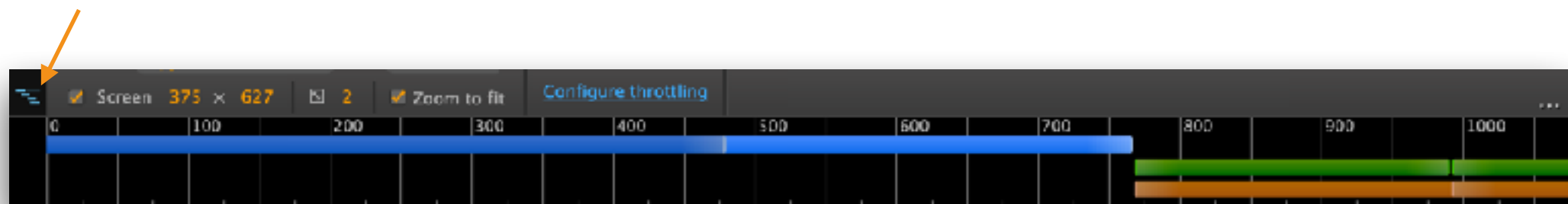
Chrome Support

- © Thankfully Chrome is our friend!



Tips

- Get used to the shortcut to toggle the Developer Tools on and off (Ctrl+Shift+I on Windows and Cmd+Alt+I on Mac)
- Dock the Developer Tools on the right-side of the screen
- Note the “Device Mode”  and the Devices
- Turn on the Media Query viewer



Exercise

Test the Developer Tools

Test the Developer Tools

- Open a few of your favorite sites and see responsive they are
- Try a few different device types / orientations
- Use the element inspector to explore the RWD techniques that they've used



Exercise

Media Queries

Media Queries

- <https://github.com/kevinbluer/epam-media-query>
- The current mobile view (e.g. iPhone 6, 6 Plus, etc) is currently broken
- Your mission is to fix it :)



Exercise

Fooooood! + RWD

Fooooood! + RWD

- Use some of the techniques in your Fooooood! site
- Begin by seeing how it currently looks on tablets and mobile devices
- Extend with a few media queries to adjust colors, font sizes, etc



Summary

Summary

- Mobile is (obviously) important!
- Depending on the customer needs / project, it's something you should consider from the beginning
- Thankfully there's a range of techniques to get your site or application working smoothly :)

Further Reading

Further Reading

- Media Queries: <http://mobile.smashingmagazine.com/2010/07/19/how-to-use-css3-media-queries-to-create-a-mobile-version-of-your-website/>
- More: <http://webdesignerwall.com/tutorials/css3-media-queries>
- Viewport: <http://webdesign.tutsplus.com/articles/quick-tip-dont-forget-the-viewport-meta-tag--webdesign-5972>
- Clearing Floats: <https://css-tricks.com/the-how-and-why-of-clearing-floats/>