Responsive Web Design

Dino Esposito
http://software2cents.wordpress.com
@despos

pluralsight
hardcore dev and IT training

Key Points

Conditional styling via CSS Media Queries

Flexible grids, **fluid layouts**, relative sizes and multi-resolution images

Taking care of **device-specific** scenarios

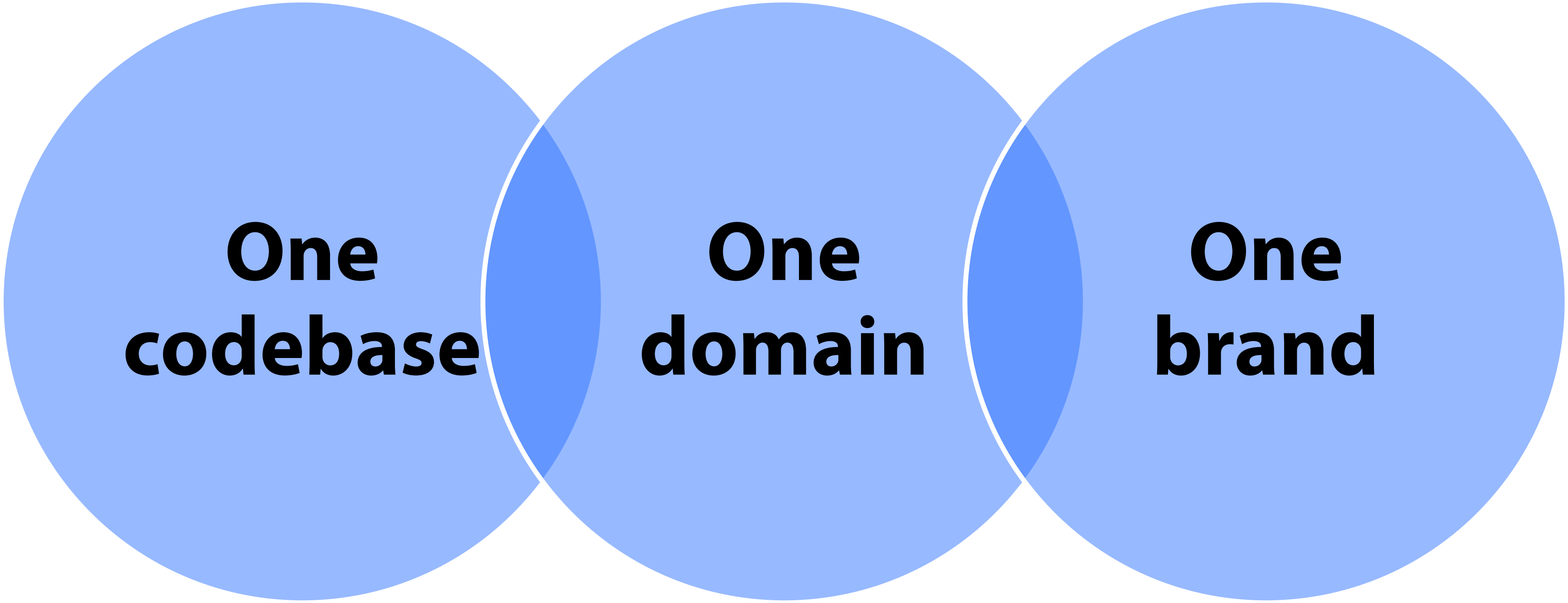
Detecting devices is hard?

Don’t do that. Be smart and let the browser do the job. Through CSS.

RWD at a Glance…

RWD at a Glance…

1% inspiration



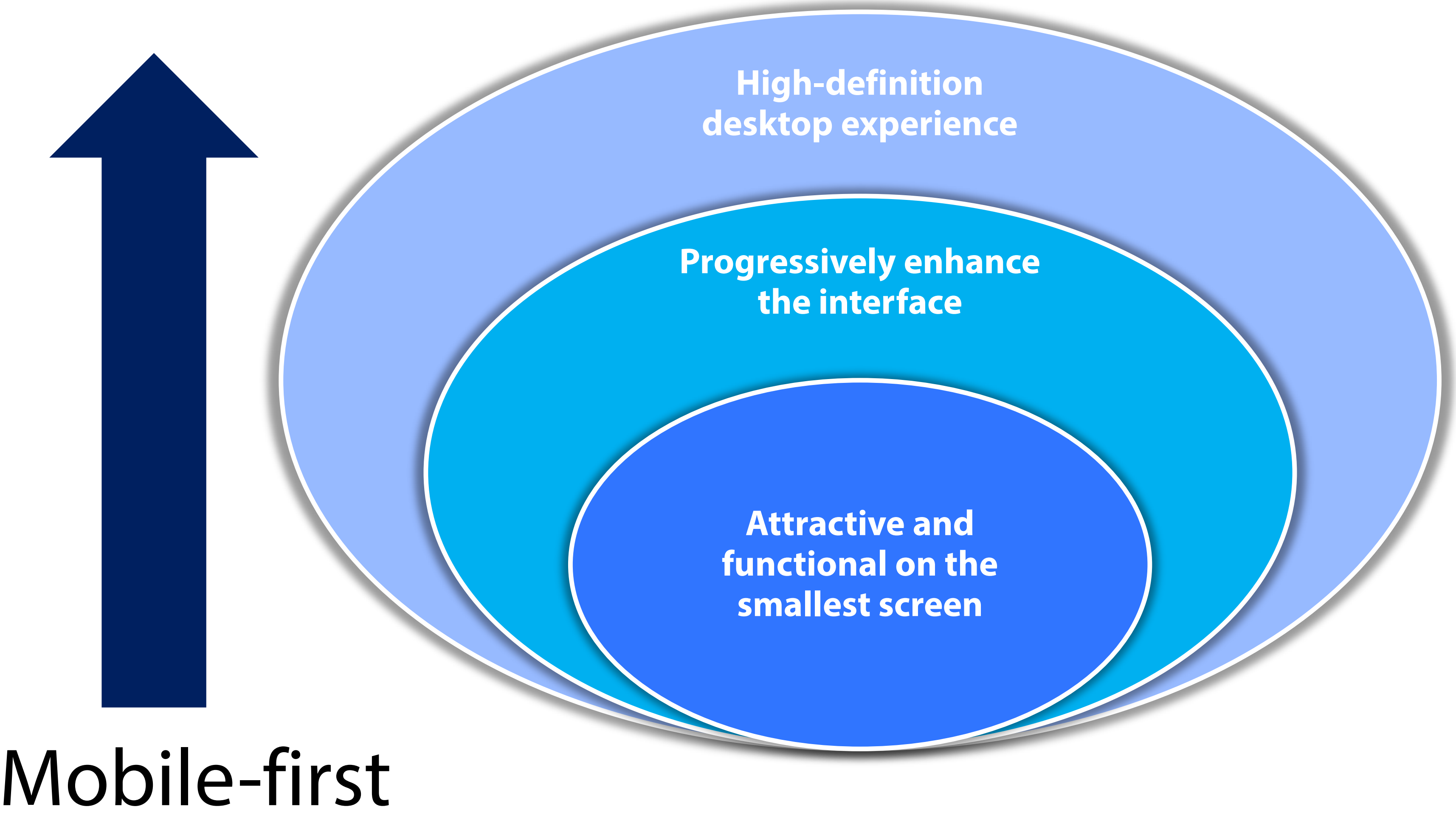
One codebase

One domain

One brand

RWD at a Glance…

99% perspiration



High-definition desktop experience

Progressively enhance the interface

Attractive and functional on the smallest screen

Mobile-first

FACTS

- **Same content downloaded to just any user agents**
- **No distinction whatsoever**
  - Regardless of the requesting URL
  - Regardless of the requesting user agent string
  - Regardless of the hosting platform
  - Regardless of the actual capabilities of the requesting device
- **Once the content is on the client …**

some magic happens

There's a Trick; and You Can See It!

It's all in the capable hands of the CSS support in the browser

- In some browsers …
- In most browsers …
- In all browsers of today and tomorrow …

All that can happen once content is downloaded is … all you can do with CSS within a browser.

As a web developer…

What can you force the browser to do with CSS?

What CSS Can Do For You

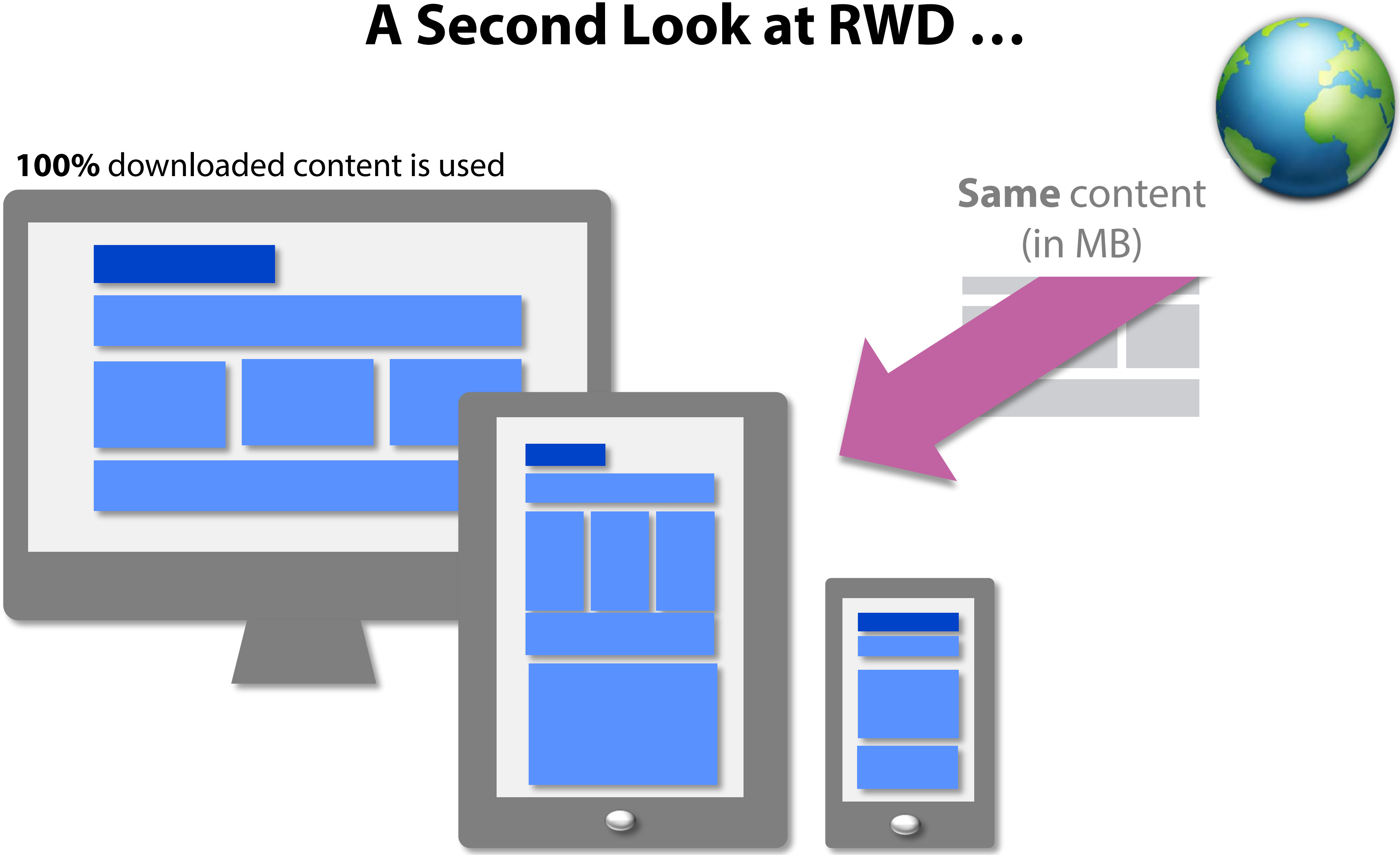
- **Reposition HTML elements**
  - Playing with relative and absolute positioning
- **Reflow HTML elements within containers**
  - Playing with the **float** attribute you can flow elements to the left/right and move to the bottom when content reaches the edge
- **Use relative sizes**
  - Width of containers/elements as a percentage of the parent’s width
  - Bounce content that exceed horizontal boundaries
  - Omit height to enable vertical scrolling
  - Force images to a given size
- **Show/Hide elements**

That’s it!

What CSS Can’t Do For You

- **Switch to different layouts**
  - Without serving the content for all possible layouts and then hiding those that don’t apply
- **Support different experiences and use-cases**
  - CSS works on top of the SAME view
  - Renders the SAME view differently
- **Can’t serve different and device-specific views**
  - Can’t optimize markup and images
  - Can’t minimize requests
  - Can’t change graphics on iOS and Android smartphones

A Second Look at RWD …



100% downloaded content is used

Same content
(in MB)

~80% downloaded
content is used

~40% downloaded
content is used

Really?

Let’s find out more about the internals of RWD.

Foundation of RWD

CSS Media Queries

- Conditional CSS style sheets
- *http://www.w3.org/TR/css3-mediaqueries*
- Newer draft in the works at W3C

CSS selected on the fly

- Visual breakpoints
- Automatically applied by browsers

Proportional layouts

- Fluid content
- Logical grids
- Percentage-based sizes

True Foundation of RWD

- Conditional CSS style sheets
- *http://www.w3.org/TR/css3-mediaqueries*

- **Newer** draft in the works

CSS Media Queries

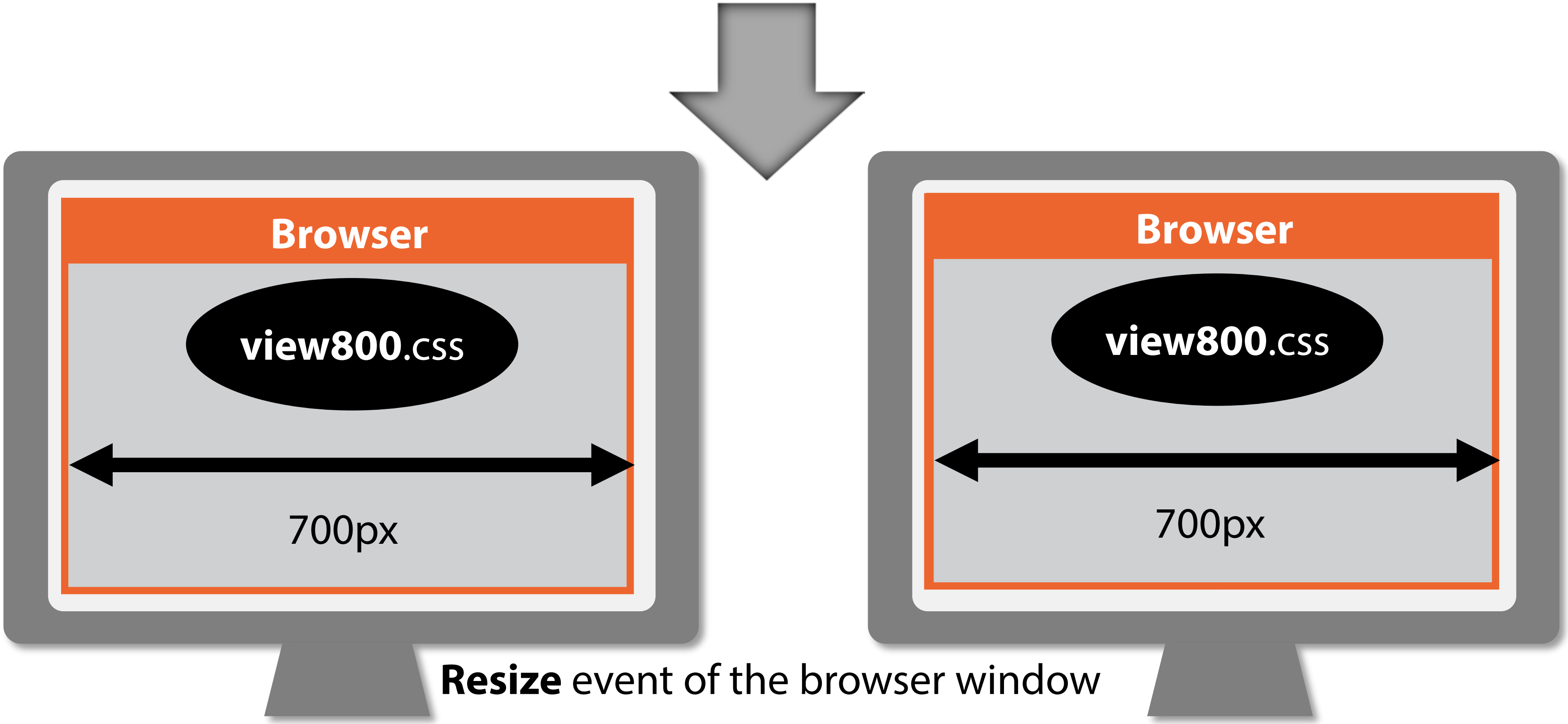
CSS Media Queries

```
<link type="text/css"
     rel="stylesheet"
     href="view480.css"
     media="only screen and (max-width: 480px)">


<link type="text/css"
     rel="stylesheet"
     href="view800.css"
     media="only screen and (max-width: 800px)">
```

Effect of CSS Media Queries

Same amount of content



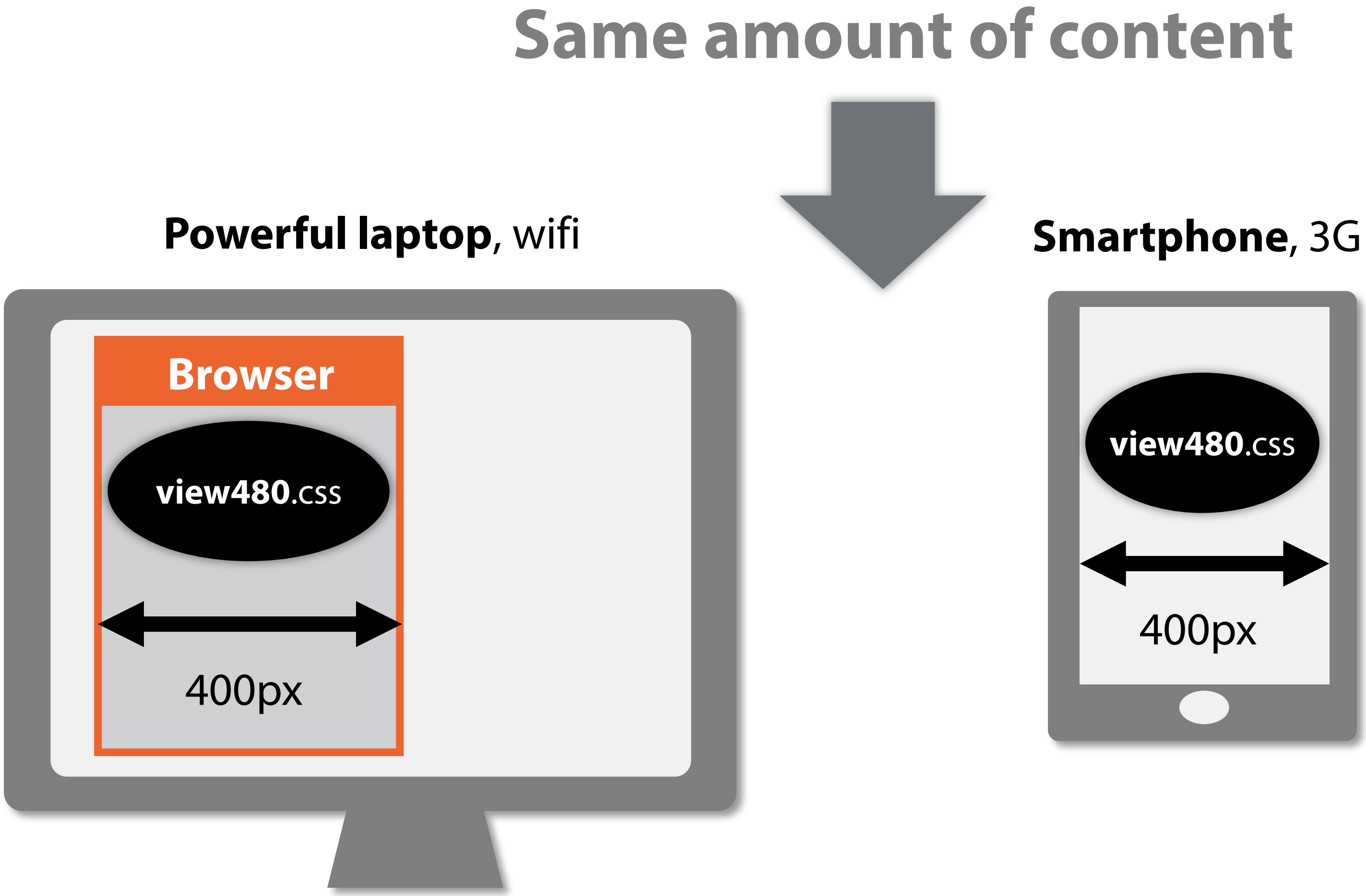
Resize event of the browser window

but

More than everything else

…

Effect of CSS Media Queries



Same amount of content

Powerful laptop, wifi

Browser

view480.css

400px

Smartphone, 3G

view480.css

400px

CSS Media Queries in Action

Browser Property	Description
width, **height**	Width and height of the rendering viewport—i.e., the browser's window.
orientation	Returns **portrait** when height is greater or equal than width. Otherwise, it returns **landscape**.
device-width, **device-height**	Width and height of the physical device screen.
aspect-ratio	Indicates the ratio between width and height. It's a value such as "16/9".
device-aspect-ratio	Indicates the ratio between device-width and device-height. It's a value such as "16/9".

DEMO About RWD

Media Queries

Grid system

Automatic resize

How to Do RWD Responsibly?

Layout of the content

- Content inventory
- Content hierarchy and structure
- HTML wireframes
- Visual breakpoints

Flexible Images

- SVG
- Adaptive images (whatever that means)

Testing on devices

- Which devices to test on
- Getting those devices
- Need of testing on real devices

Navigating the content

- No fixed guidelines (toolbar, drop-down menu)
- Need for a strong graphical idea

Old browsers?

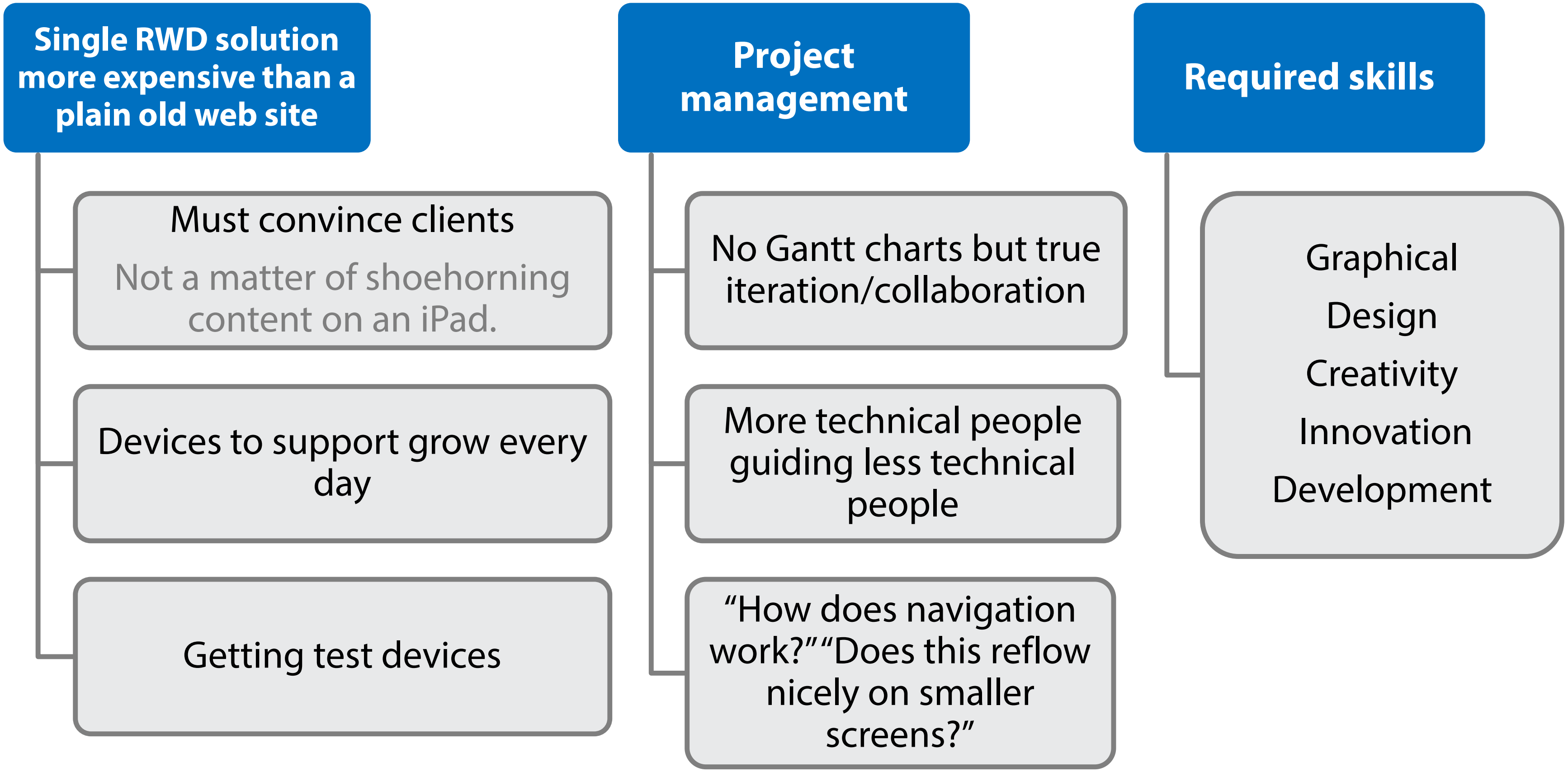
- No media queries
- Limited rendering
- Things like IE7/IE8

“From scratch” approach

- No shoehorned design
- No adaptation from fixed-width sites
- No rigid tables

Costs of a RWD Solution

More or less same costs as building device-specific solutions



- **Single RWD solution more expensive than a plain old web site**
  - Must convince clients
    Not a matter of shoehorning content on an iPad.
  - Devices to support grow every day
  - Getting test devices
- **Project management**
  - No Gantt charts but true iteration/collaboration
  - More technical people guiding less technical people
  - “How does navigation work?”“Does this reflow nicely on smaller screens?”
- **Required skills**
  - Graphical
    Design
    Creativity
    Innovation
    Development

With RWD

One Solution Gets Built.

Twitter Bootstrap

http://get**bootstrap**.com

Current de-facto standard in modern web development

- Developed internally at Twitter; then open-sourced
- Included in the Visual Studio releases
- Herald of responsive design

Essentially, a CSS file

- Bunch of jQuery plugins (packaged in JavaScript modules for download)
- Glyphs
- (Recommended) HTML templates

New web taxonomy

- DIV elements styled with Bootstrap classes become "new" elements
- Tabs, accordions, modal dialogs, progress bars, carousels, etc…

Twitter Bootstrap

http://get**bootstrap**.com

- **Enable responsive design**
  - □ Predefined visual breakpoints: 480, 768, 992, 1200+ pixels
  - □ Modular layout: **container** > **row** > **span**

```
<div class="container">
    <div class="row">
        :
    </div>
    <div class="row">
        :
    </div>
</div>
```

Bootstrap
predefined styles

Twitter Bootstrap

http://getbootstrap.com

- **Built-in grid system**
  - Each row has up to 12 cells
  - Columns in a row can be nested and have offset
- **Screen prefixes**
  - Number of cells goes with screen sizes: **lg**, **md**, **sm**, **xs**

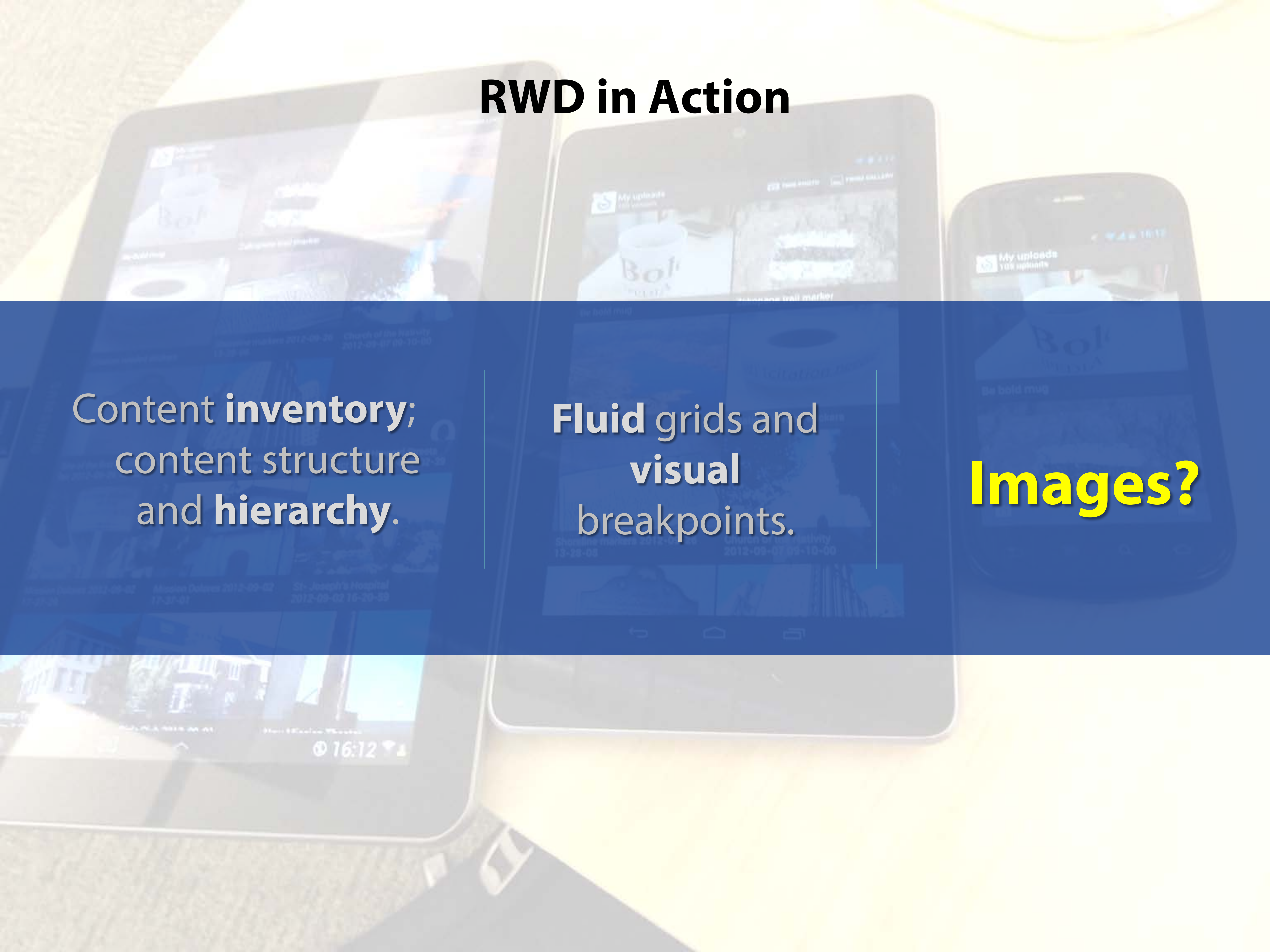
```html
<div class="row">
   <div class="col-md-4 col-xs-6"> ... </div>
   <div class="col-md-4"> ... </div>
   <div class="col-md-4 col-xs-6"> ... </div>
</div>
```

3 columns on medium (**md**) devices
2 columns on extra-small (**xs**) devices

Real-world Bootstrap …

- **Doesn't remove need of great web design**
  - All sites may look the same: is aesthetics important?
- **Doesn't remove need of great device UX**
  - All devices are NOT the same
  - Twitter, for example, doesn't blindly use Bootstrap on devices…
- **Higher level than plain HTML**
  - Overrides HTML semantic
  - Presentation no longer completely separate from content
- **All-or-nothing approach**
  - Hard to jump on ongoing projects
  - Over 100K of CSS plus 30K max of script (at most)

RWD in Action



Content **inventory**; content structure and **hierarchy**.

Fluid grids and **visual** breakpoints.

Images?

Dealing with Images

Small and high-quality: not an easy trade-off to make.

- **All that we have, however, is the IMG element**
  - No support for multi-resolution images
  - PICTURE or SRCSET elements coming in future standards
- **Idea is binding images to media queries expressions**

```
<img sizes="(min-width: 40em) 80vw, 100vw"
     srcset="../examples/images/medium.jpg 375w,
             ../examples/images/medium.jpg 480w,
             ../examples/images/large.jpg 768w">
```

FROM: http://github.com/scottjehl/picturefill

WURFL Image Tailor

- **Serve resized images to meet the capabilities of the device**
  - Automatic server-side service: detect and resize
  - Truly mobile solution for images
  - Detect device and its known screen size
  - Use logic to calculate optimal size of the image
- **Single URL**
  - No multiple images for you to create/maintain on the server

```
<img src="http://wit.wurfl.io/http://yourserver/images/sea.jpg">
```

WURFL Image Tailor

```
<img src="http://wit.wurfl.io/http://yourserver/images/sea.jpg">
```

- **Absolute URL of the image**
  - JPG, PNG (8- and 24-bit), GIF, BMP supported
- **Default settings for resizing**
  - *http://wurfl.io/documentation/wit-directives.php*
- **Free of charge**
  - Subject to monthly quota
  - Exceeding quota just causes the service to redirect to original image
  - SLA available

“It is **not** the strongest of the species that survive, nor the most intelligent, but **the one most responsive** to change.”

— **Charles Darwin**

> As long as changes implemented end up working well in a real-world scenario.

(corollary)

Summary

- **Device-friendly web sites are a must**
- **RWD delivers a solution that works with different screens**
  - Mostly about reflowing/reducing content for different screens
- **RWD subject to the power of CSS**
  - Can’t have completely different layouts for tablets and smartphones
  - Can’t serve intelligent markup
- **Ideal?**
  - One web site to serve different views (markup)
  - Device detection is key
  - Effective device detection

Next topic