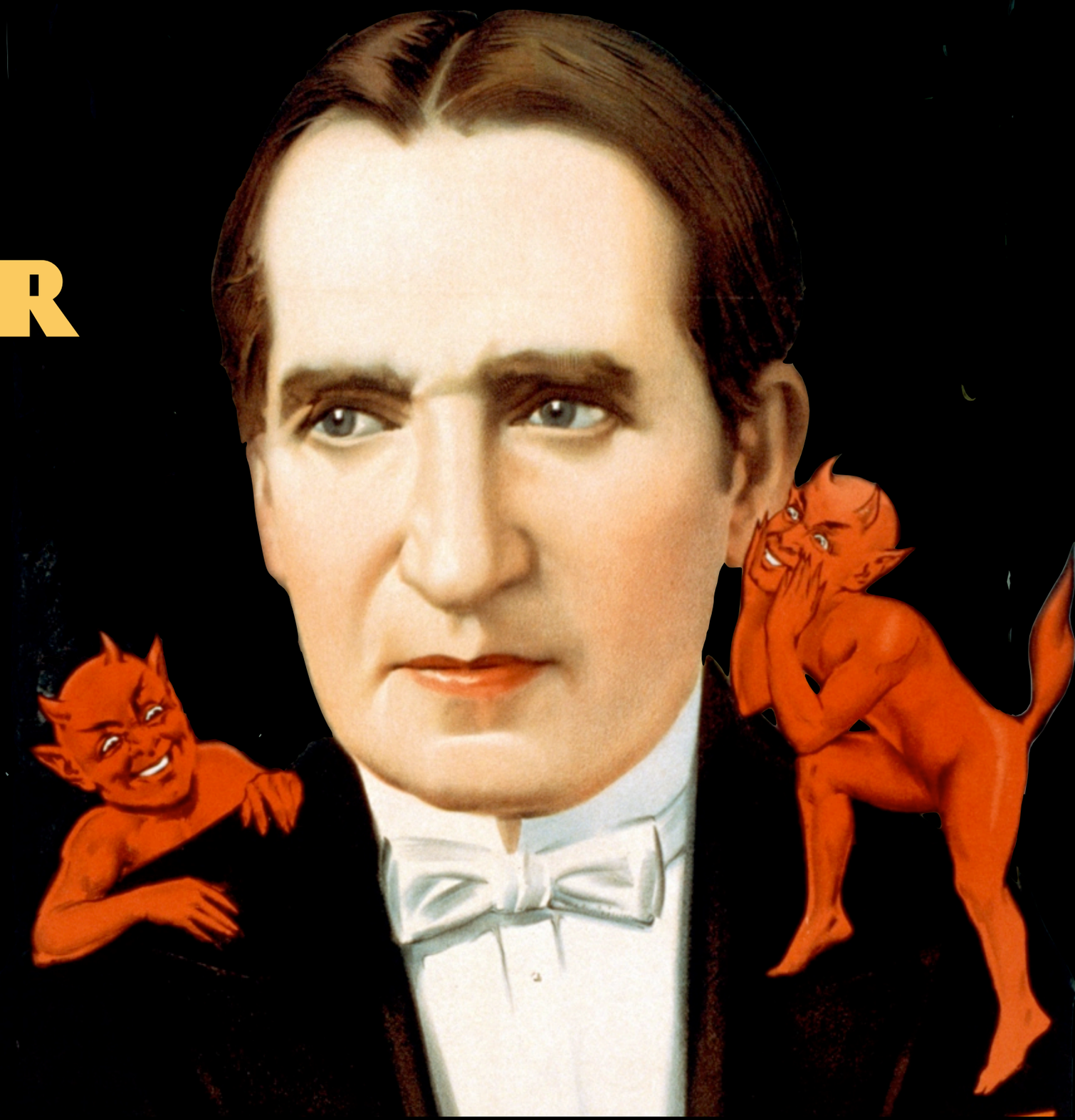
@media mobilism and (year: 2011)

META LAYOUT

A closer look at MEDIA QUERIES

by STEPHEN HAY

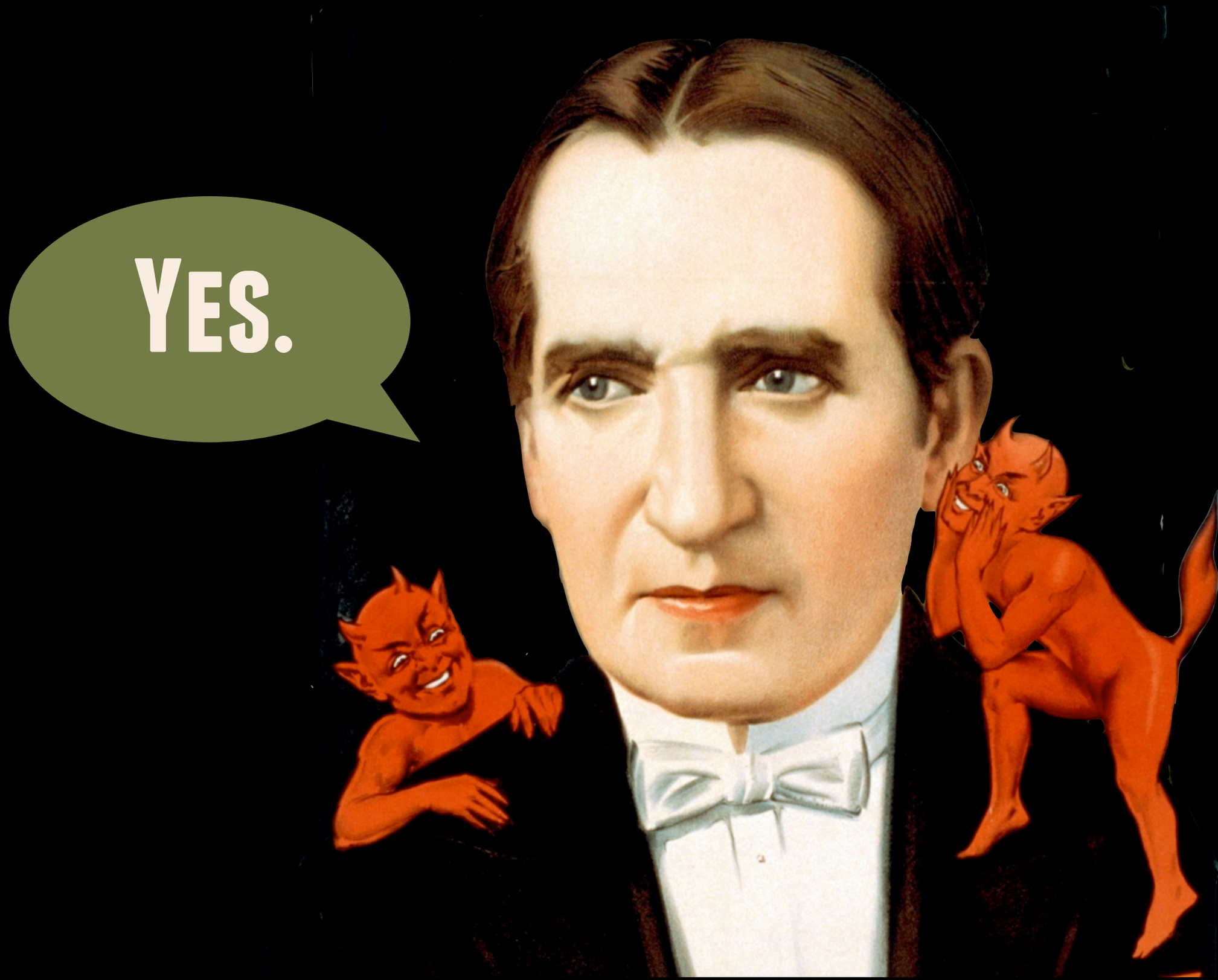
B. Rowser



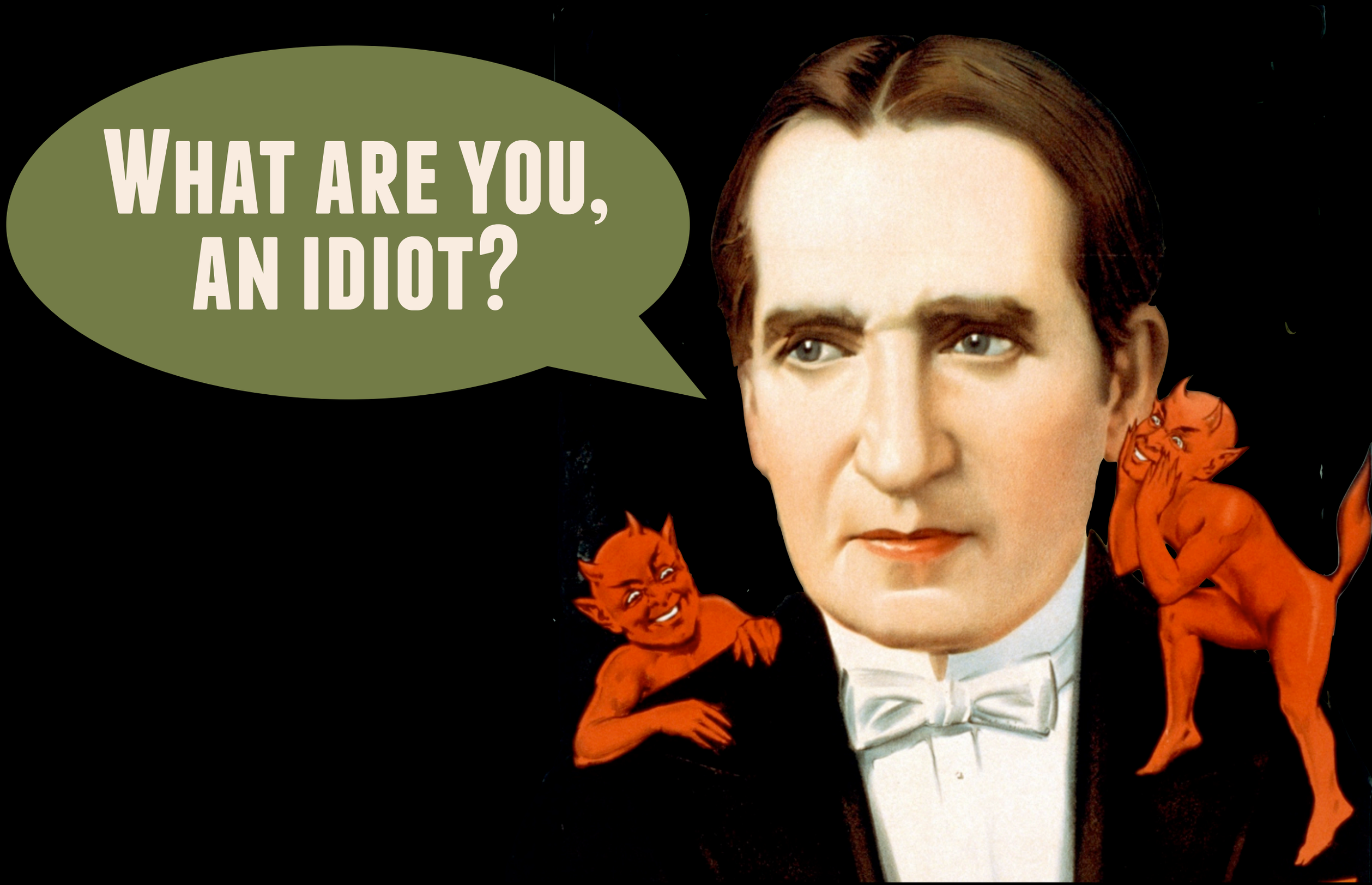
HI HANDSOME.

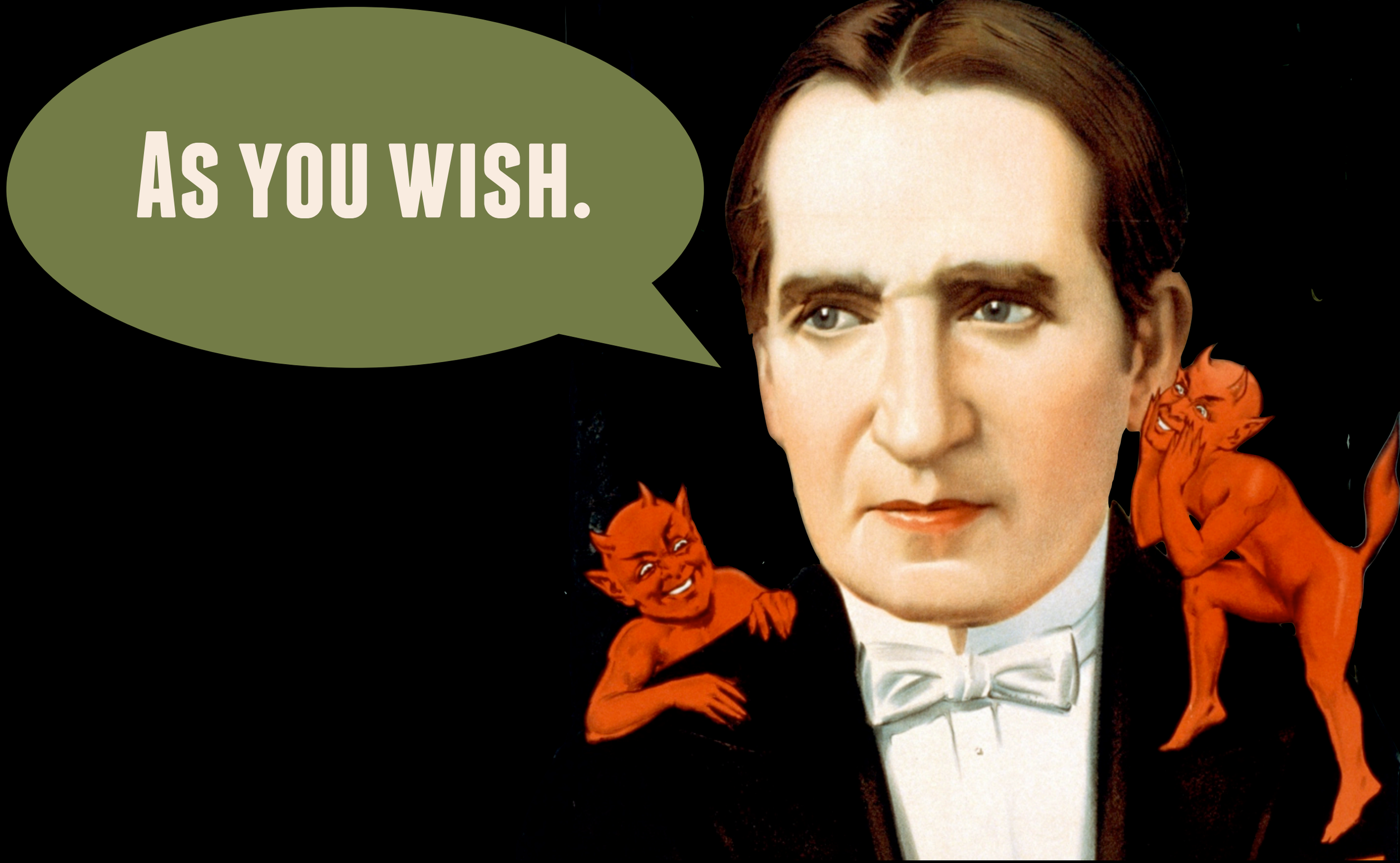
```
@media mood and (min-level: fine)
```

```
@media screen and (min-width: 600px)
```



```
@media all and (orientation: landscape)
```




As you wish.

QuickTime Player File Edit View Share Window Help Stop Recording Thu 3:59 PM

Yiibu – Lovingly crafted mobile experiences

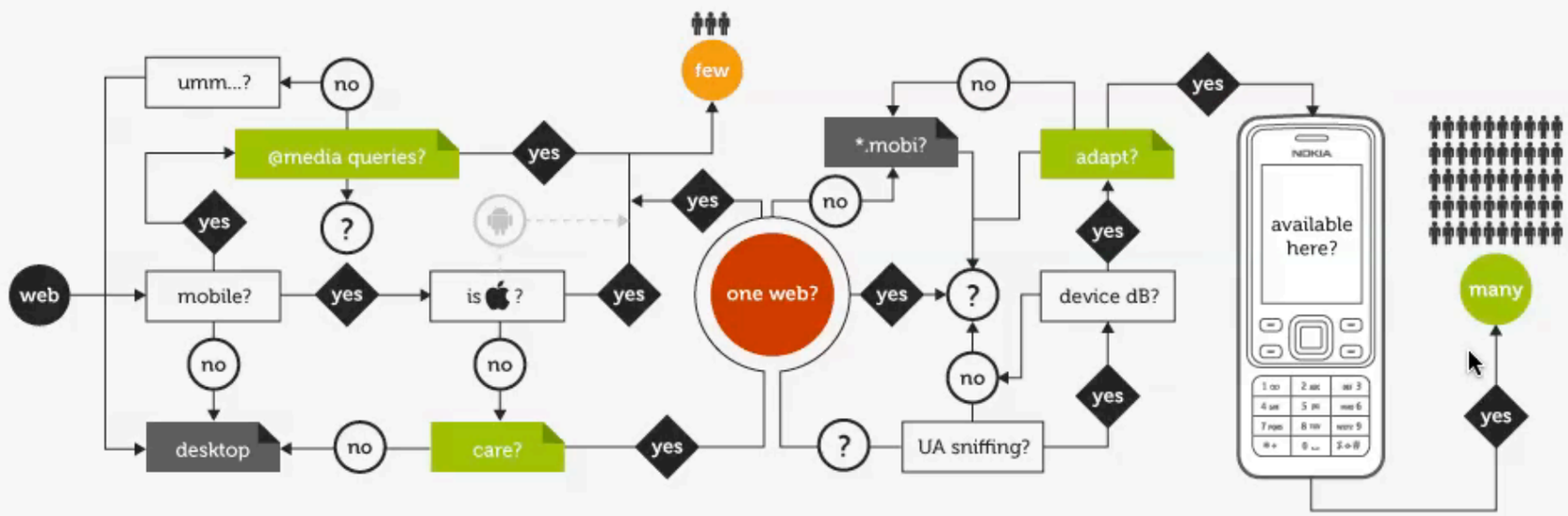
Web yiibu.com Search with Google

yiibu articles | projects search... go

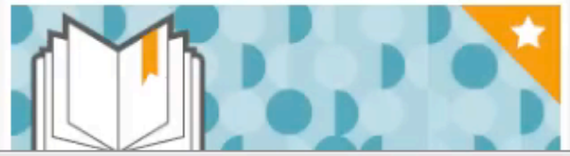
WORK IN PROGRESS

Are we there yet?

This site is a proof of concept for many of the ideas described in Rethinking the Mobile Web. Over the coming months, we will release articles covering our approach in detail. In the meantime why not discover how these ideas came to be and how this site was developed.



This site works beautifully on most modern and legacy smartphones, tablets, and of course desktop browsers. Find out how we did it.



View (100%)

QuickTime Player File Edit View Share Window Help Stop Recording Thu 4:05 PM

Ethan Marcotte is a web designer & developer who lives in Boston.

Web ethanmarcotte.com Search with Google

ABOUT writing portfolio contact

ETHAN MARCOTTE

is a web designer & developer, a speaker and author.

Ethan lives in Boston, Massachusetts, and is passionate about beautiful design, elegant code, and the intersection of the two. He is a popular and experienced speaker, having been featured at An Event Apart, Carsonified Workshop, and South by Southwest, and co-hosts the Handcrafted CSS seminar series with Dan Cederholm.

More fun facts: Ethan has a blog, and blathers incessantly on Twitter. His clientele has included *New York Magazine*, Stanford University, and the World Wide Web Consortium. Also, he is tall.

If you'd like someone to speak at your conference or company, or to work on your next design project, Ethan is available for freelance engagements from

View (100%)

MEDIA QUERIES

```
@media [not|only] type [and] (expr) {
  rules;
}
```

== Feature detection

MEDIA TYPES

```
@media screen { ... }
@media print { ... }
@media handheld { ... }
```

Media Types

```
@media screen { ... }
@media print { ... }
@media handheld { ... }
```

Let's add some logic...

```
@media screen and { ... }
```

MEDIA TYPES

```
@media screen { ... }
@media print { ... }
@media handheld { ... }
```

LET'S ADD SOME LOGIC...

```
@media screen and { ... }
```

AND A FEATURE TO QUERY

```
@media screen and (color) { ... }
```

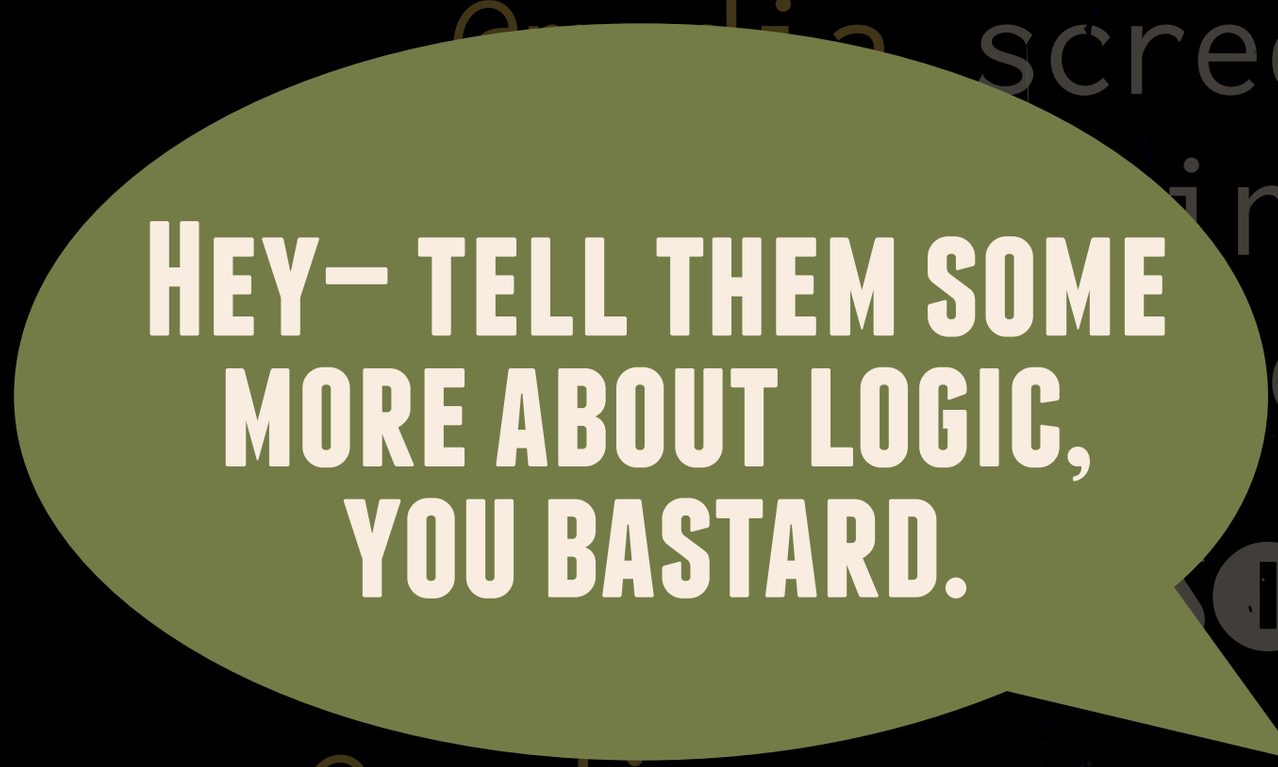
MEDIA TYPES

```
@media screen {
```

```
@media screen a
```

AND A FEAT

```
@media screen
```



LOGICAL AND

```
@media screen and (min-width: 600px)
and (max-width: 1200px)
```

CONSTRAINS A QUERY

LOGICAL OR

```
@media screen and (min-resolution: 300dpi), screen and (-webkit-min-device-pixel-ratio: 2)
```

THE COMMA IS AN OR

LOGICAL NOT

```
@media not screen and (color)
```

LOGICAL NOT

```
@media not screen and (color)
```

BE CAREFUL...

LOGICAL NOT

```
@media not screen and (min-width: 600px)
```

LOGICAL NOT

```
@media not screen and (min-width: 600px)
```

DOES NOT EVALUATE AS:

```
@media (not screen) and (min-width: 600px)
```

LOGICAL NOT

```
@media not screen and (min-width: 600px)
```

DOES NOT EVALUATE AS:

```
@media (not screen) and (min-width: 600px)
```

DOES EVALUATE AS:

```
@media (not (screen and (min-width: 600px)))
```

ONLY

```
@media only all and
(device-aspect-ratio: 16/9)
```

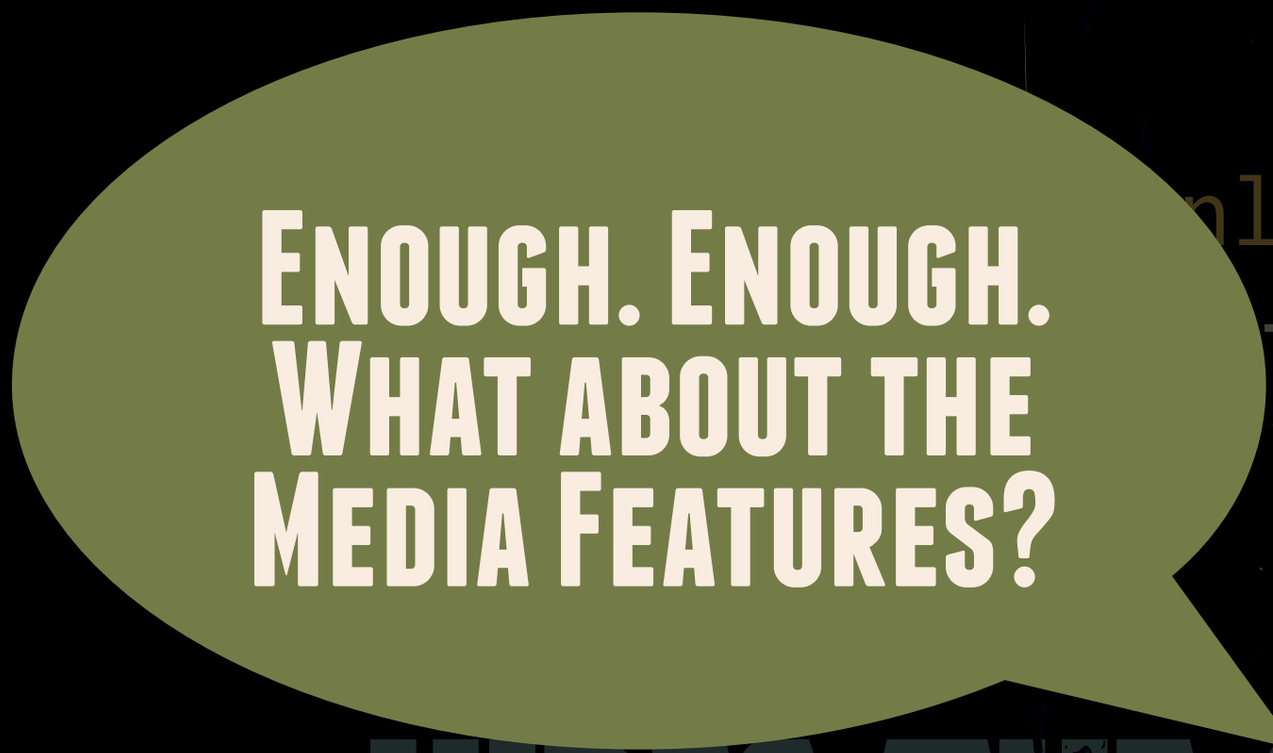
HIDES THE CSS FROM OLDER BROWSERS

ONLY

nly

HIDES THE C

OLDER



Media Features

width
height
device-width
device-height
orientation
aspect-ratio
device-aspect-ratio

color
color-index
monochrome
resolution
scan
grid

Media Features

width
height
device-width
device-height
orientation
aspect-ratio
device-aspect-ratio

color
color-index
monochrome
resolution
scan
grid

Media Features

width
height
device-width
device-height
orientation
aspect-ratio
device-aspect-ratio

color
color-index
monochrome
resolution
scan
grid

Media Features

width
height
device-width
device-height
orientation
aspect-ratio
device-aspect-ratio

color
color-index
monochrome
resolution
scan
grid

Media Features

width
height
device-width
device-height
orientation
aspect-ratio
device-aspect-ratio

color
color-index
monochrome
resolution
scan
grid

Media Features

width
height
device-width
device-height
orientation
aspect-ratio
device-aspect-ratio

color
color-index
monochrome
resolution
scan
grid

Media Features

width
height
device-width
device-height
orientation
aspect-ratio
device-aspect-ratio

color
color-index
monochrome
resolution
scan
grid

Media Features

width
height
device-width
device-height
orientation
aspect-ratio
device-aspect-ratio

color
color-index
monochrome
resolution
scan
grid

Media Features

width
height
device-width
device-height
orientation
aspect-ratio
device-aspect-ratio

color
color-index
monochrome
resolution
scan
grid

Media Features

likely you'll use these most:

width
height
device-width
device-height
orientation
aspect-ratio
device-aspect-ratio

color
color-index
monochrome
resolution
scan
grid

Testing

CSS3 Media Queries: Various Tests

This test page is a work in progress. Last updated 7 May 2011.

Basic media query support

Media queries in the `media` attribute of the `link` element in HTML are supported.

Media queries within CSS are supported.

Media queries in `@import` rules are supported.

Basic media feature support

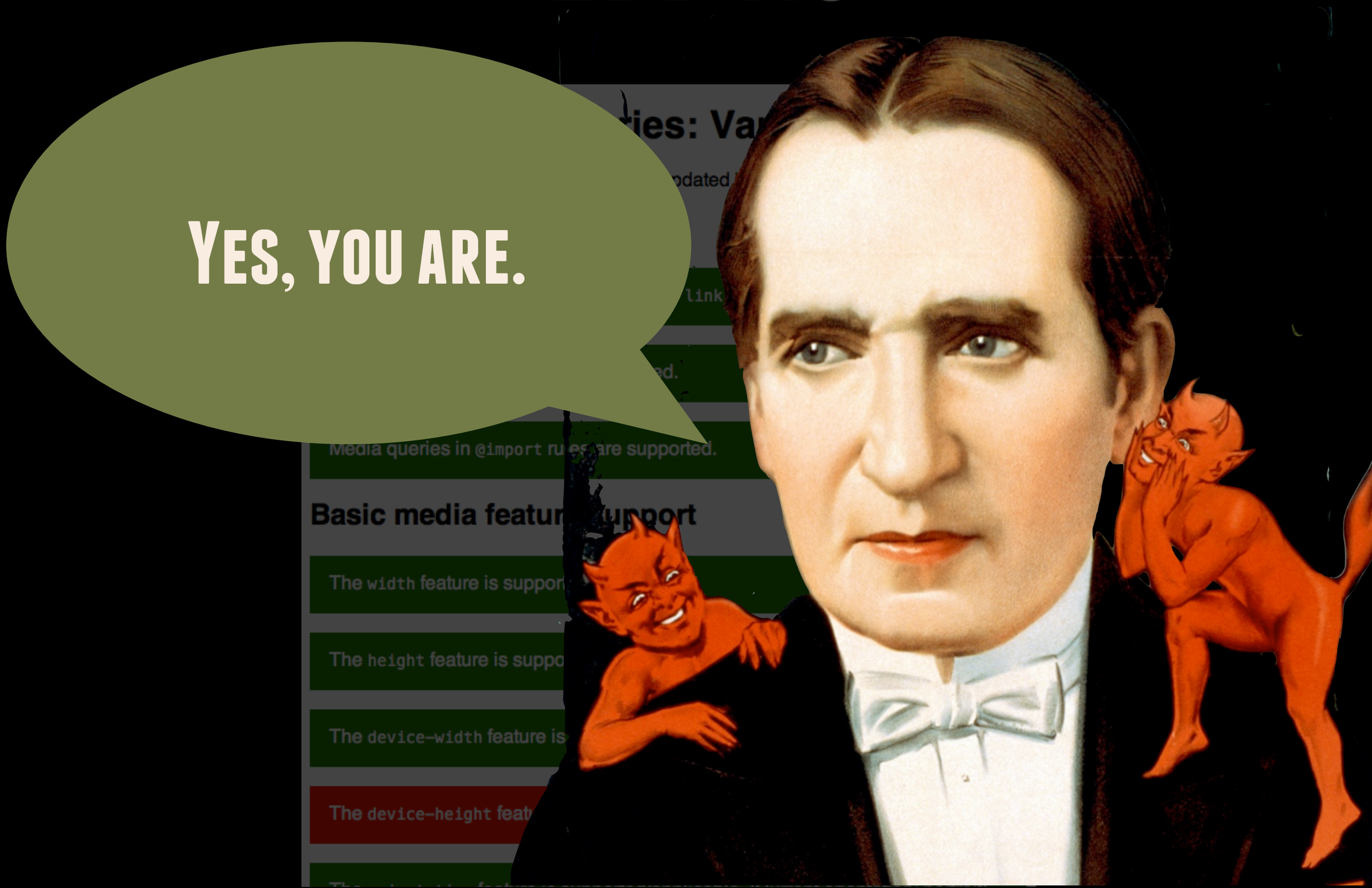
The `width` feature is supported/applicable.

The `height` feature is supported/applicable.

The `device-width` feature is supported/applicable.

The `device-height` feature is not supported/applicable.

Testing


Yes, you are.
Media queries in @import rules are supported.
Basic media feature support
The width feature is suppor
The height feature is suppo
The device-width feature is
The device-height feat

Prefixed Features

Prefixed Features

How many device pixels per CSS pixel?

Prefixed Features

How many device pixels per CSS pixel?

-webkit-[min|max]-device-pixel-ratio
[min|max]--moz-device-pixel-ratio
-o-[min|max]-device-pixel-ratio

Prefixed Features

How many device pixels per CSS pixel?

```
-webkit-[min|max]-device-pixel-ratio
[min|max]--moz-device-pixel-ratio
-o-[min|max]-device-pixel-ratio
```

```
-webkit-min-device-pixel-ratio: 1.5
-o-min-device-pixel-ratio: 3/2
```

Prefixed Features

A pixel is not a pixel.

Prefixed Features

Is this a touch device?

`-moz-touch-enabled`

Prefixed Features

Is this a touch device?

-moz-touch-enabled

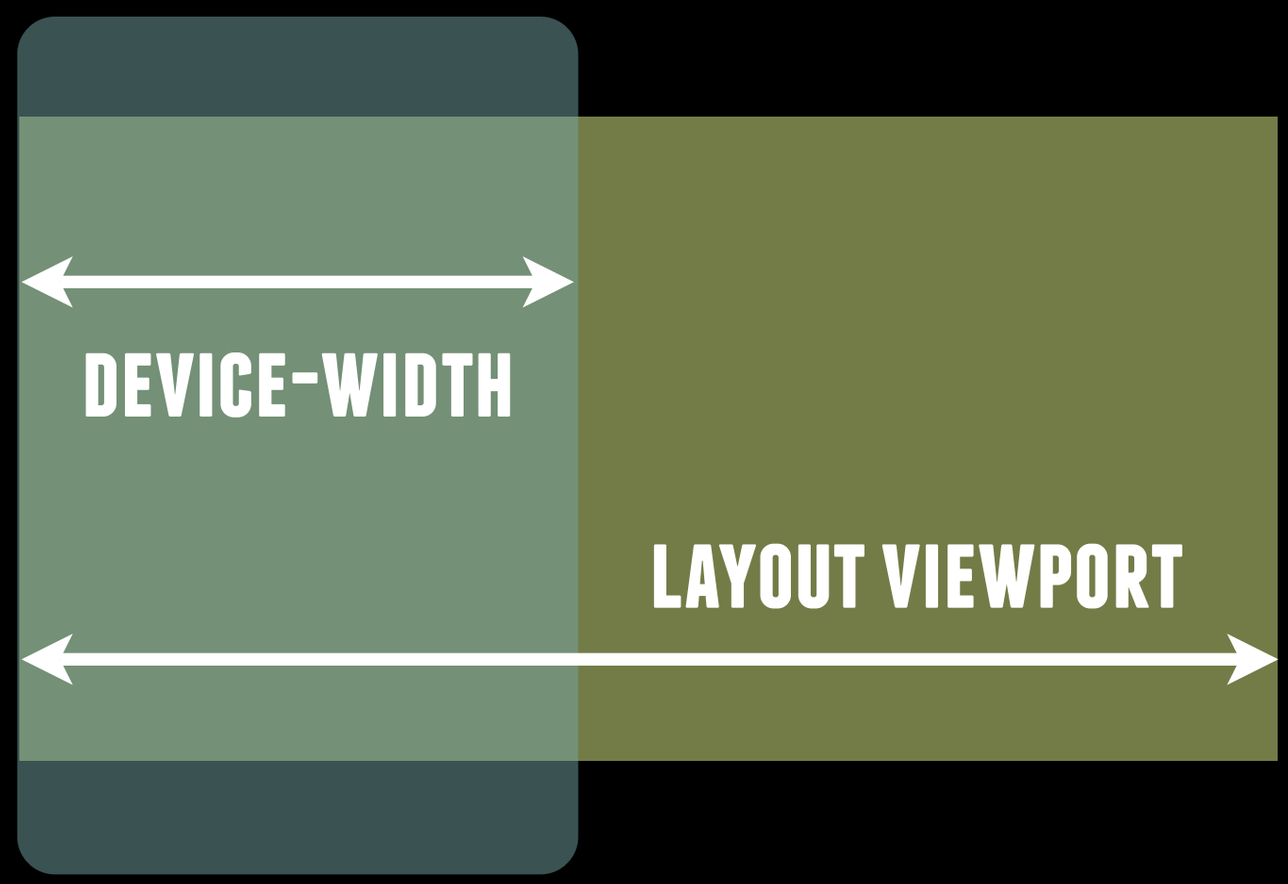
```
@media screen and (-moz-touch-enabled) {
  /* touchy styles */
}
```

Viewport

```
@media screen and (max-device-width: 320px)
```

Viewport

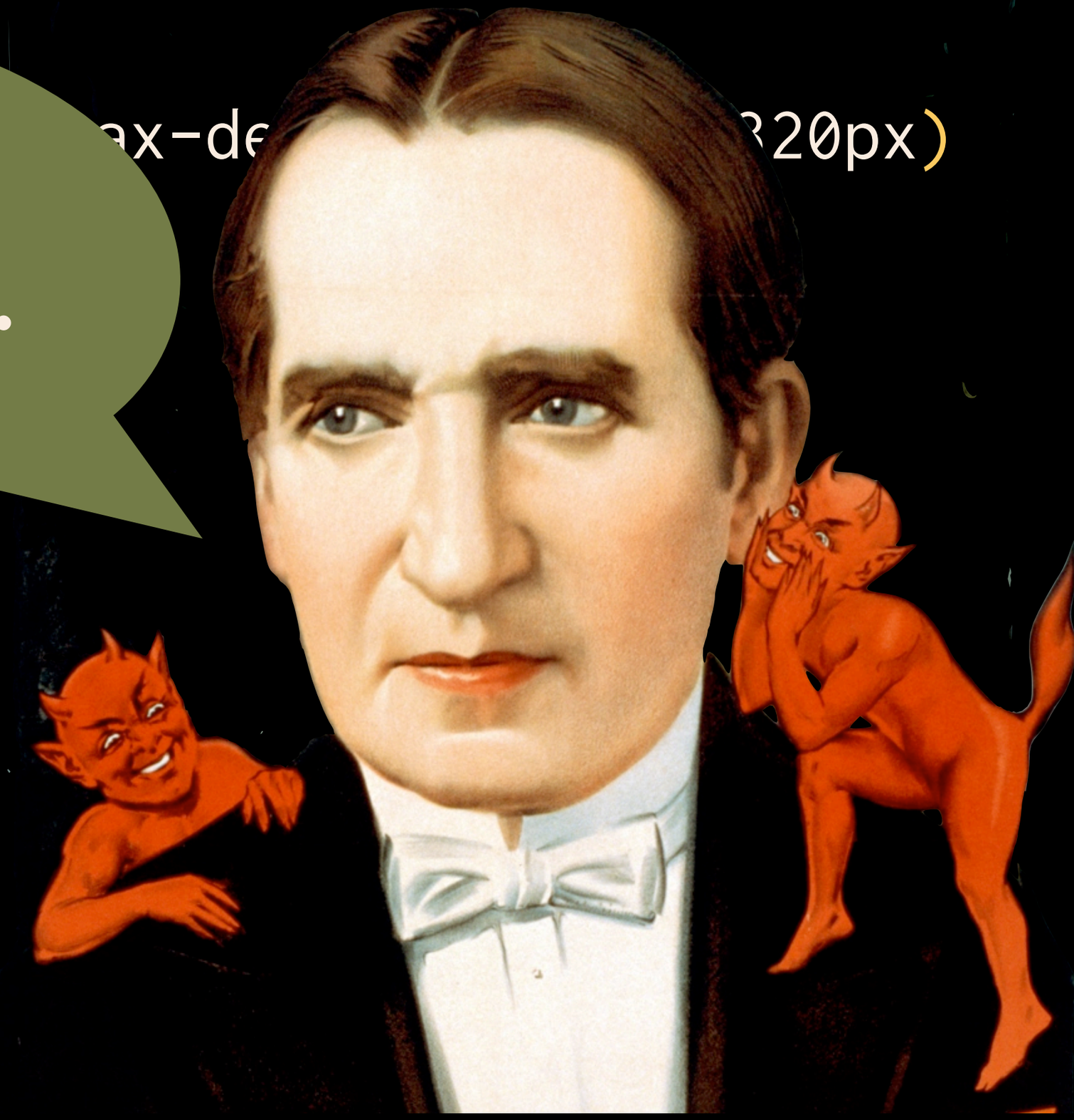
```
@media screen and (max-device-width: 320px)
```



Viewport

```
@ax-de 320px)
```

Meta viewport.



Meta viewport

```
<meta name="viewport"
content="width=device-width">
```

Meta viewport

```
<meta name="viewport"
content="width=device-width">
```

Use that in combination with width

```
@media screen and (max-width: 320px) {
  div {
    width: 80%;
  }
}
```

Meta viewport

```
<meta name="viewport"
content="width=device-width,
initial-scale=1">
```

Use that in combination with width

```
@media screen and (max-width: 320px) {
  div {
    width: 80%;
  }
}
```

VIEWPORT IN CSS?

```
@viewport {
  width: device-width;
  zoom: 1; /* No, not THAT zoom */
}
```

VIEWPORT IN CSS?

Applying media queries

Applying media queries

as blocks in CSS

```
@media screen and (max-width: 960px) {
  .inner {
    background: url(../img/bg-page-top.png) repeat-x;
    }
  .tertiary {
    float: none;
    width: 100%;
    margin: 0 auto;
    max-width: 960px;
    }
```

Applying media queries

as blocks in CSS

```
@media screen and (max-width: 960px) {
  .inner {
    background: url(../img/bg-page-top.png) repeat-x;
    }
  .tertiary {
    float: none;
    width: 100%;
    margin: 0 auto;
    max-width: 960px;
    }
```

media attribute of link element

```
<link rel="stylesheet" href="/_css/2011/base.css?20110321" media="screen, handheld">
<link rel="stylesheet" href="/_css/2011/small.css?20110302b" media="only screen and (min-width: 600px)">
<link rel="stylesheet" href="/_css/2011/large.css?20110302b" media="only screen and (min-width: 900px)">
```

Applying media queries

as blocks in CSS

```
@media screen and (max-width: 960px) {
  .inner {
    background: url(../img/bg-page-top.png) repeat-x;
    }
  .tertiary {
    float: none;
    width: 100%;
    margin: 0 auto;
    max-width: 960px;
    }
```

media attribute of link element

```
<link rel="stylesheet" href="/_css/2011/base.css?20110321" media="screen, handheld">
<link rel="stylesheet" href="/_css/2011/small.css?20110302b" media="only screen and (min-width: 600px)">
<link rel="stylesheet" href="/_css/2011/large.css?20110302b" media="only screen and (min-width: 900px)">
```

@import

```
@import url('import.css') only all;
```

Applying media queries

```
<link rel="stylesheet" href="/_css/2011/base.css?20110321" media="screen, handheld">
<link rel="stylesheet" href="/_css/2011/small.css?20110302b" media="only screen and (min-width: 600px)">
<link rel="stylesheet" href="/_css/2011/large.css?20110302b" media="only screen and (min-width: 900px)">
<!-- Responsive Design ftw! -->
<!--[if (lt IE 9) & (!IEMobile)]>
<link rel="stylesheet" href="/_css/2011/small.css?2011031">
<link rel="stylesheet" href="/_css/2011/large.css?2011031">
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
<!-- Old IE's ftl! -->
```

```
href="resources/styles/basic.css" media="screen, handheld" />
href="resources/styles/mobile.css" media="only screen and (min-width: 320px) and (max-width: 640px)" />
href="resources/styles/desktop.css" media="only screen and (min-width: 640px)" />
```

You don't always need them

QuickTime Player File Edit View Share Window Help Stop Recording Thu 3:58 PM

Joshua Davis | Studios

Web www.joshuadavis.com Search with Google

Joshua Davis | Studios

home

interactive
print
exhibitions
products

artwork

contact

calendar of events

twitter
facebook
flickr

server hosted by (mt)

shop

After several requests, we finally got a shop set up with the great guys over at Big Cartel.

While we still offer limited editions through our gallery partnerships... this shop allows us to offer affordable, non-editioned work for purchase.

click here to view.

Biography

Joshua Davis is a New York based artist, designer, and technologist producing both public and private work for companies, collectors, and institutions.

Since 1995, Joshua Davis has been using technology and computers as a medium to create projects that a recent Apple Pro Profile touted as infinitely interesting.

Manifesto

A key advantage to working with Joshua Davis Studios is our small boutique team. We are devoted to working with our clients and their needs, and translating those needs into a dynamic experience.

Research and development is at the heart of our process. At the studio, our friends, partners and adventurers work with us to advance this medium, not replicate it.

Clients

Each year Joshua Davis Studios accepts a limited number of projects with progressive clients.

BMW, Kanye West, Motorola, Amp'd Mobile, Nike, Volkswagen, Umbra, Miquelrius, Sony, Motown Records, Barneys, Puff Daddy, Bad Boy Entertainment, TOOL, Universal Records, Atlantic Records, HBO, Canon, Nokia, Charles Schwab, New Riders Publishers and others.

Studio

New York

Joshua Davis
234 Jefferson Ave
Mineola, New York 11501

studio@joshuadavis.com

tel : 1.516.417.3111

New Business

Chicago

Ben Arditti
3701 N. Ravenswood Ave, # 202
Chicago, Illinois 60613

ben@joshuadavis.com

tel : 1.773.818.4438

2010

the SAK, Flower Power
10 . 2010 / products

Aim High, Keep Moving!
06 . 2010 / print

the HYPE Framework
01 . 2010 / interactive

2009

Cup and Saucer
11 . 2009 / products

Reflect

REFLECT is an Apple iPhone Art App we've been working on that uses multi-touch gestures to randomly generate a composition which gets reflected into a kaleidoscopic pattern.

You can then save your reflection as a background wallpaper, or send to a friend.

Visit sidewaysmobile.com for more info and availability.

Reflect
06 . 2009 / interactive

OFFF Oeiras Bezier Grid
05 . 2009 / exhibition

Outside the Lines Exhibition
04 . 2009 / exhibition

Carmel Apple
04 . 2009 / color

Outside the Lines
04 . 2009 / print

Outside the Line Tests
04 . 2009 / development

View (100%)

Adapt.js - Adaptive CSS

960 GRID SYSTEM

What is this?

Adapt.js is a lightweight (755 bytes minified) JavaScript file that determines which CSS fil
to load before the browser renders a page. If the browser tilts or resizes, Adapt.js simply
checks its width, and serves only the CSS that is needed, when it is needed.

A potential drawback of Adapt.js is the possibility of a brief flash of unstyled content as a
new stylesheet is being fetched (think of it as "Ajax" for CSS). I have done my best to
mitigate this by keeping CSS files small (3 KB). It is worth noting this is a proposed, not
prescribed, approach to a problem with multiple solutions.

Other methods include: Build a separate site for mobile. Or, use media queries to adjust
layout, with a polyfill for older browser support, and conditional Internet Explorer commen
for Windows phones. Also a factor is how to handle multiple image resolutions without
adding file size. Filament Group is advocating context aware image sizing.

DEFAULT CSS FILES & WIDTHS

File Name	Screen Width
mobile.css	below 760px
720.css	760px to 980px
960.css	980px to 1280px
1200.css	1280px to 1600px
1560.css	1600 to 1920px
fluid.css	above 1920px

DOWNLOAD — This page's demo code.

```
// Edit to suit your needs.
var ADAPT_CONFIG = {
  // Where is your CSS?
  path: 'assets/css/',

  // false = Only run once, when page first loads.
```

Configuration

To use Adapt.js, all you have to do is set up a few parameters: `path` is the directory structure in which your stylesheets reside, `dynamic` is a boolean (`true` or `false`) that tells Adapt.js whether to watch the `window` for its `resize` event, also

Layout is not design.

Design is the art of putting form and content together.

Paul Rand said that

The media query is a screwdriver.

Layout is only one of the components of design.

HEY DUDE, REMEMBER WHEN WE DESIGNED WEB PAGES?

Pages

to

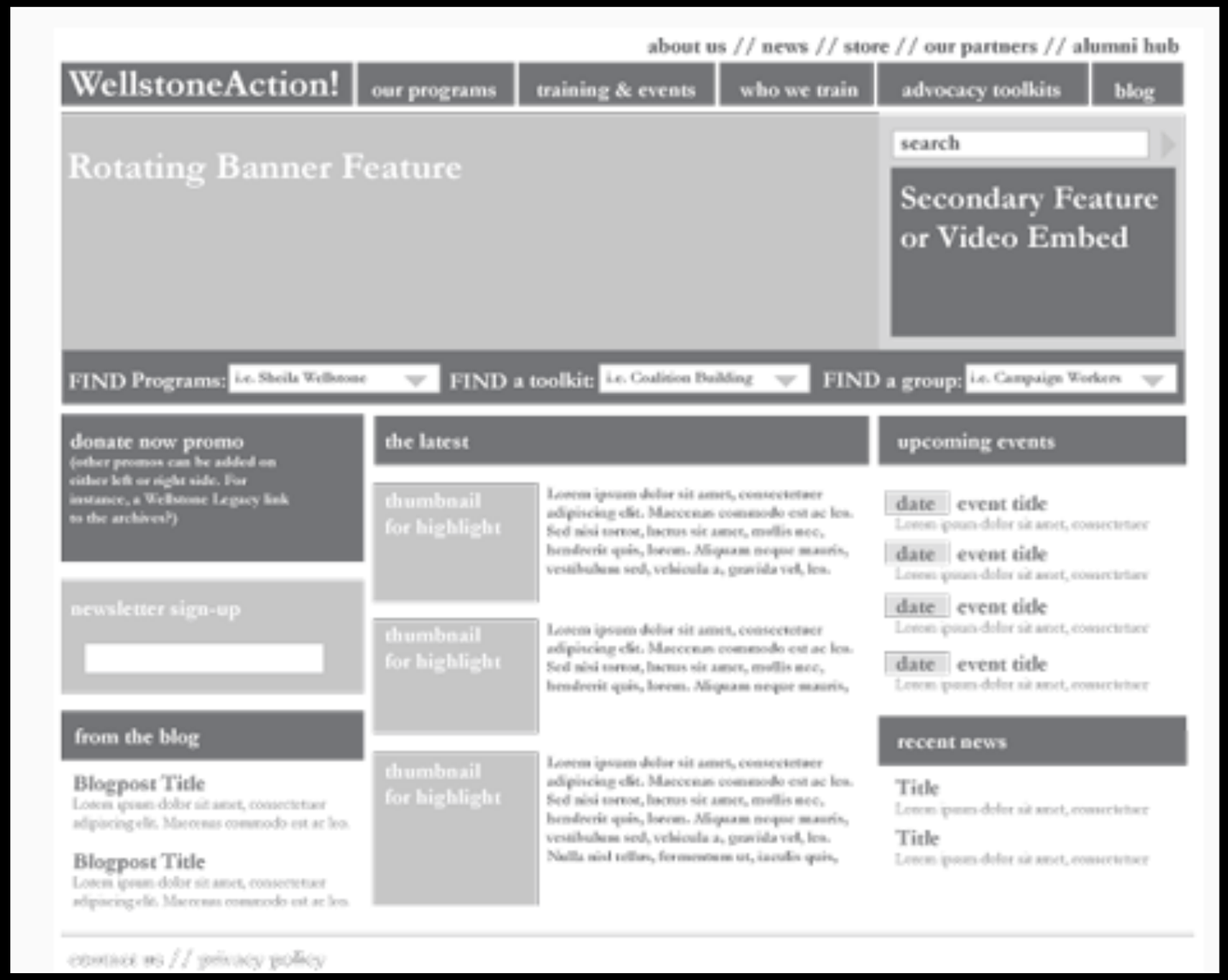
Components

to

Systems

Layout strategy

wireframes



drupal.org/node/413910

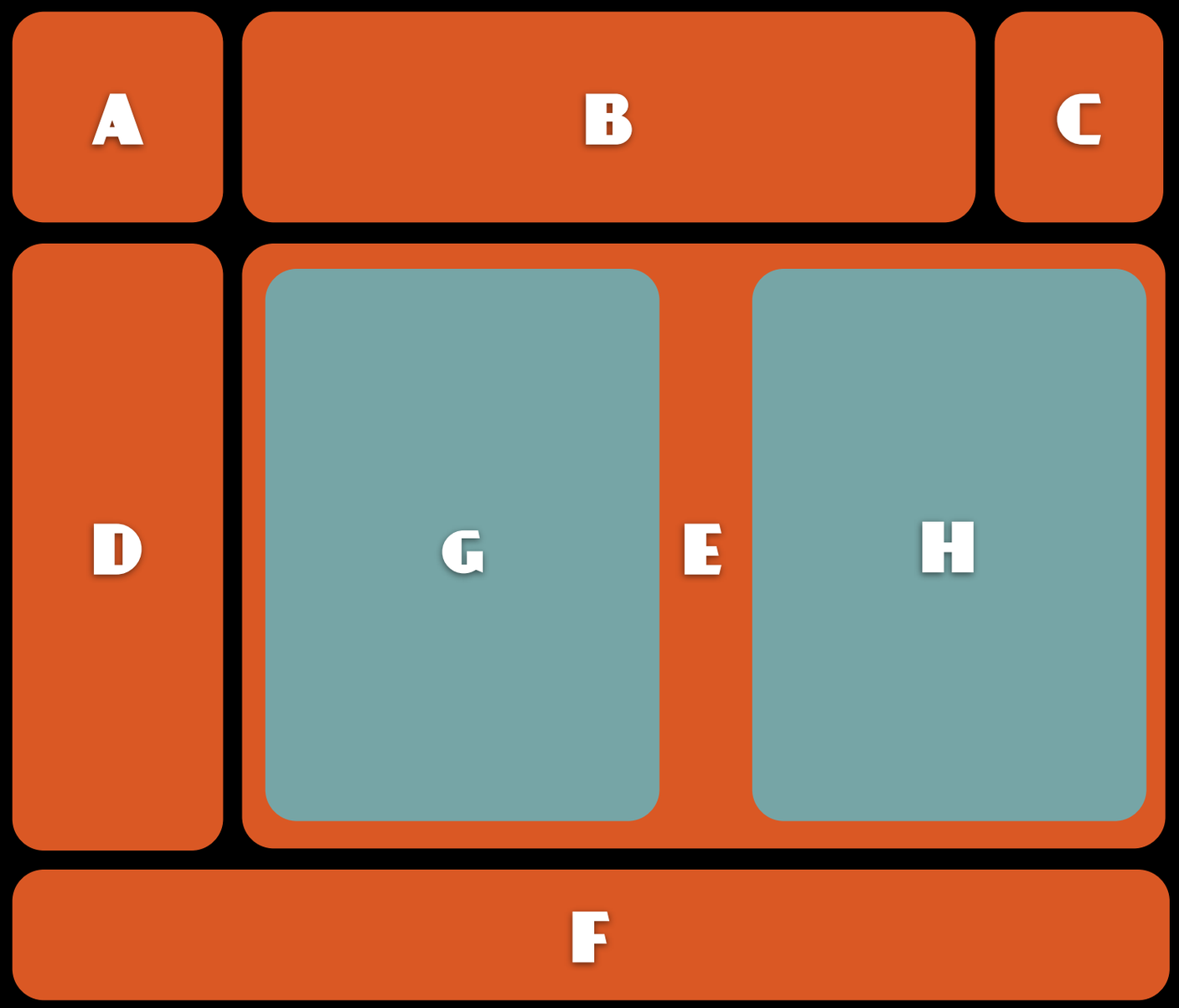
Layout strategy

wireframes

graffletopia.com/stencils/358

Layout strategy

Reference wireframing

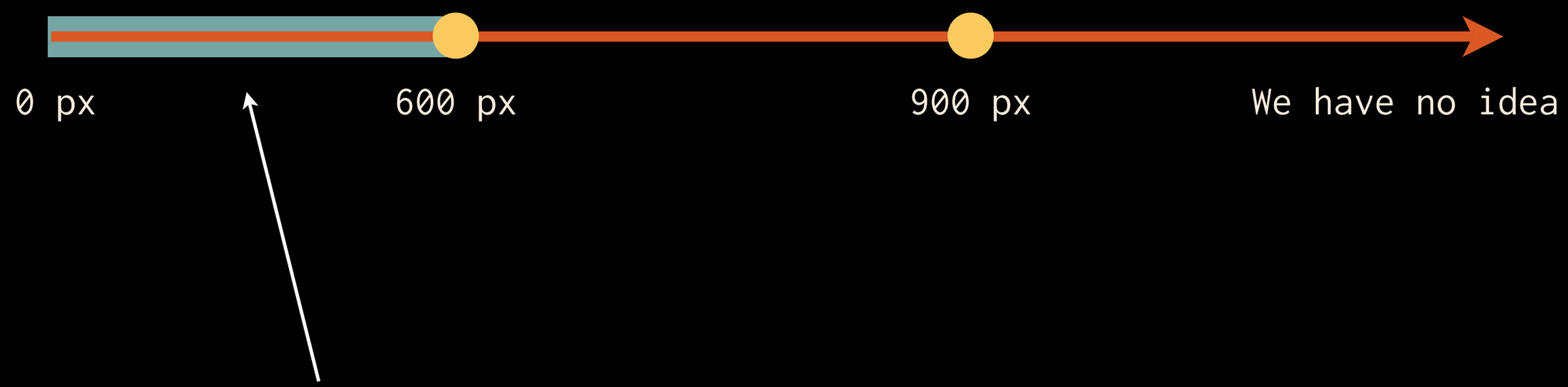


Layout strategy

Breakpoint graphs

Layout strategy

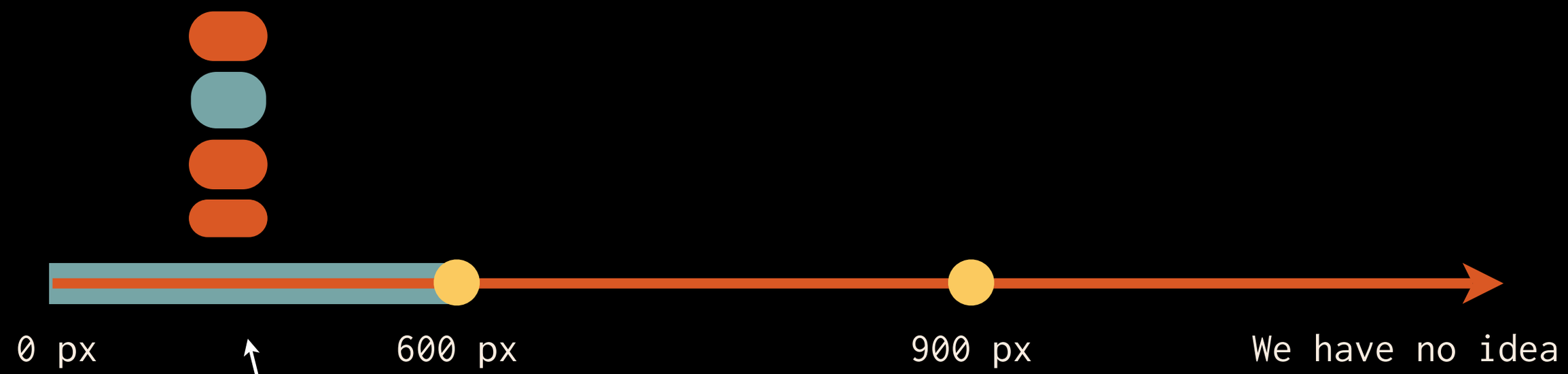
Breakpoint graphs



Structured content first

Layout strategy

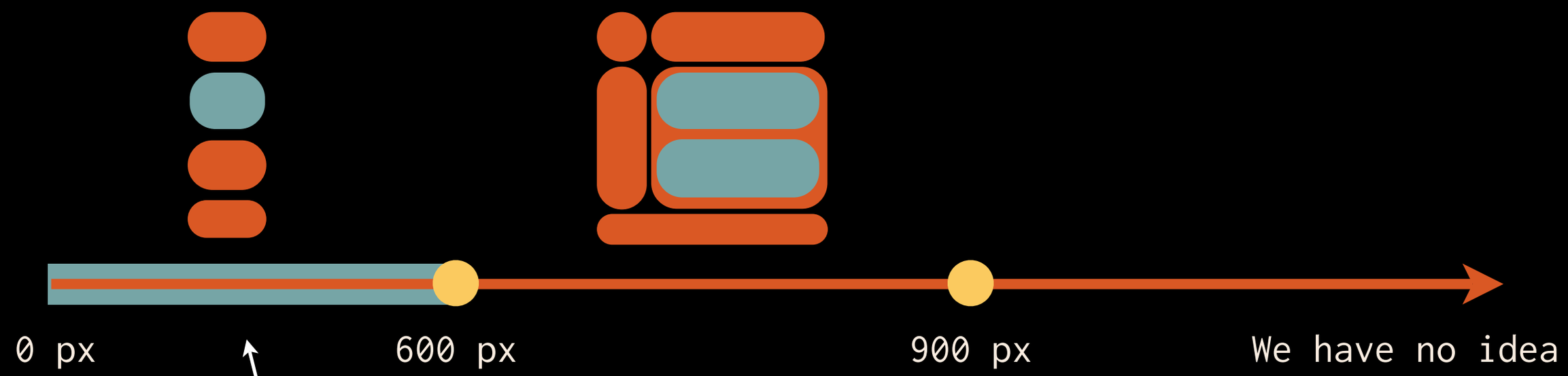
Breakpoint graphs



Structured content first

Layout strategy

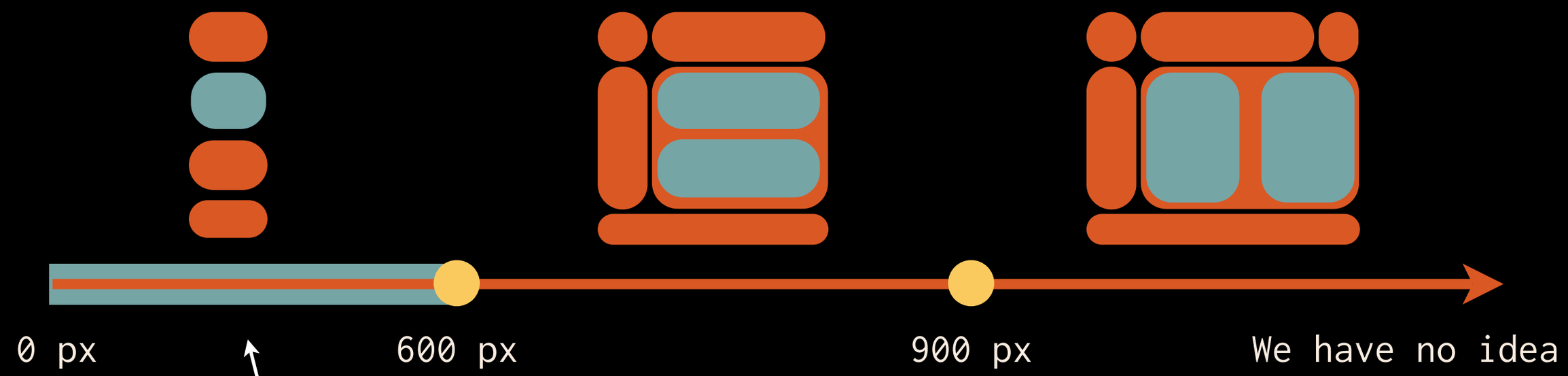
Breakpoint graphs



Structured content first

Layout strategy

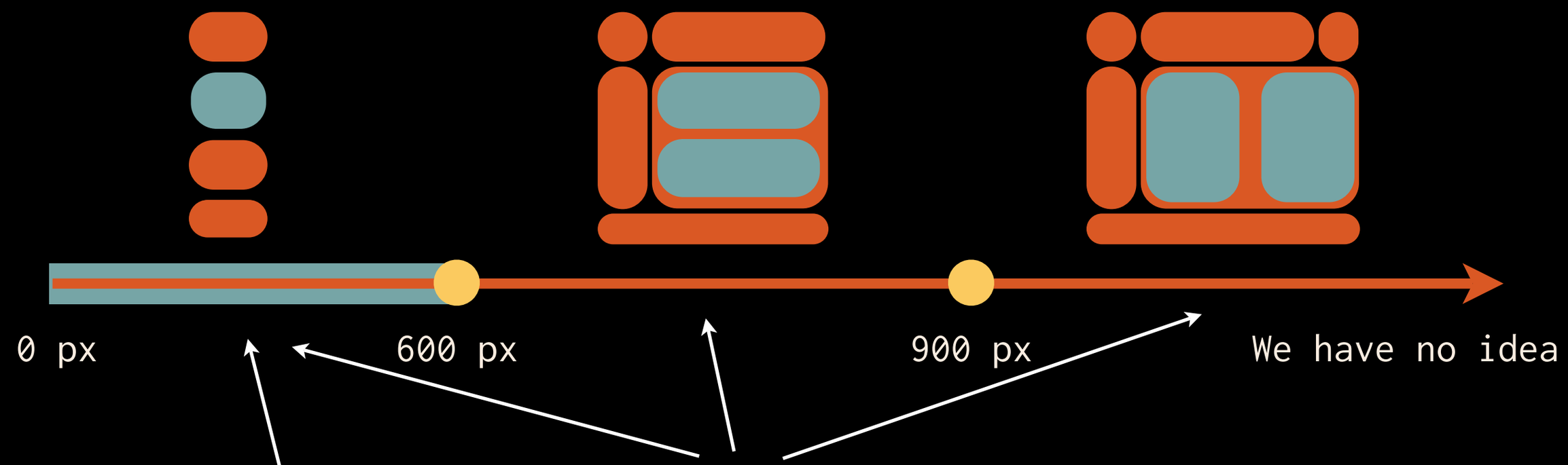
Breakpoint graphs



Structured content first

Layout strategy

Breakpoint graphs



Fluid layout

Structured content first

Order-independent layout mechanisms

```html
<section>
    <article id="news"></article>
    <article id="sports"></article>
    <article id="entertainment"></article>
</section>
```

```html
<section>
    <article id="news"></article>
    <article id="sports"></article>
    <article id="entertainment"></article>
</section>
```

N

S

E

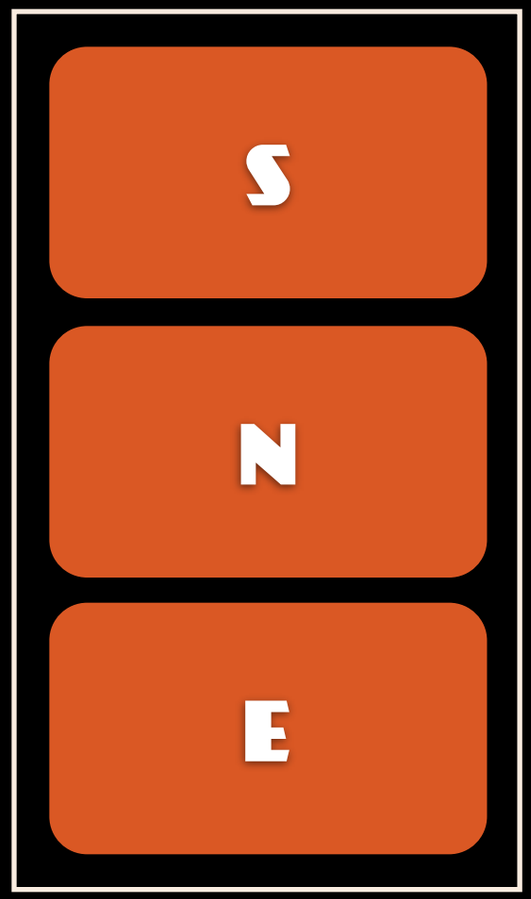
Flexible Box Layout module

```
<section>
    <article id="news"></article>
    <article id="sports"></article>
    <article id="entertainment"></article>
</section>
```

```
section {
    width: 960px;
    display: flexbox;
    flex-direction: lr;
    }
article {
    width: 200px;
    }
#news, #sports {
    width: flex(1);
    }
#sports {
    flex-order: 1;
    }
#news, #entertainment {
    flex-order: 2;
    }
```

Flexible Box Layout module

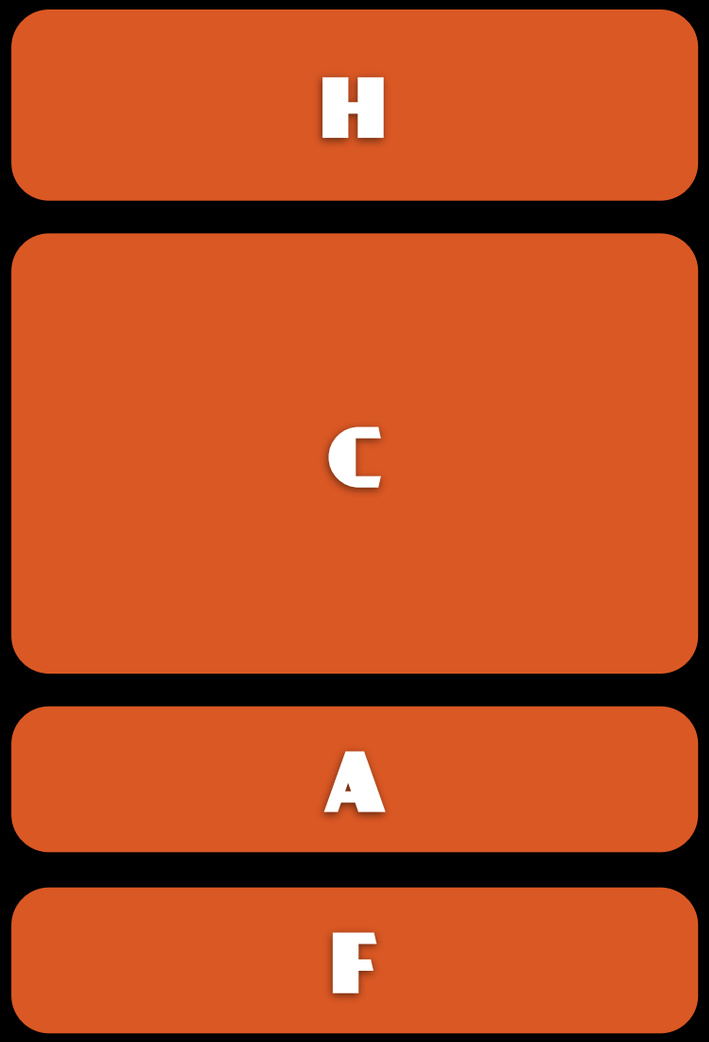
```
<section>
    <article id="news"></article>
    <article id="sports"></article>
    <article id="entertainment"></article>
</section>
```




```
section {
    width: 960px;
    display: flexbox;
    flex-direction: tb
    }
article {
    width: 200px;
    }
#news, #sports {
    width: flex(1);
    }
#sports {
    flex-order: 1;
    }
#news, #entertainment {
    flex-order: 2;
    }
```

```html
<div id="page">
    <header></header>
    <section id="content"></section>
    <aside></aside>
    <footer></footer>
</div>
```

```html
<div id="page">
    <header></header>
    <section id="content"></section>
    <aside></aside>
    <footer></footer>
</div>
```

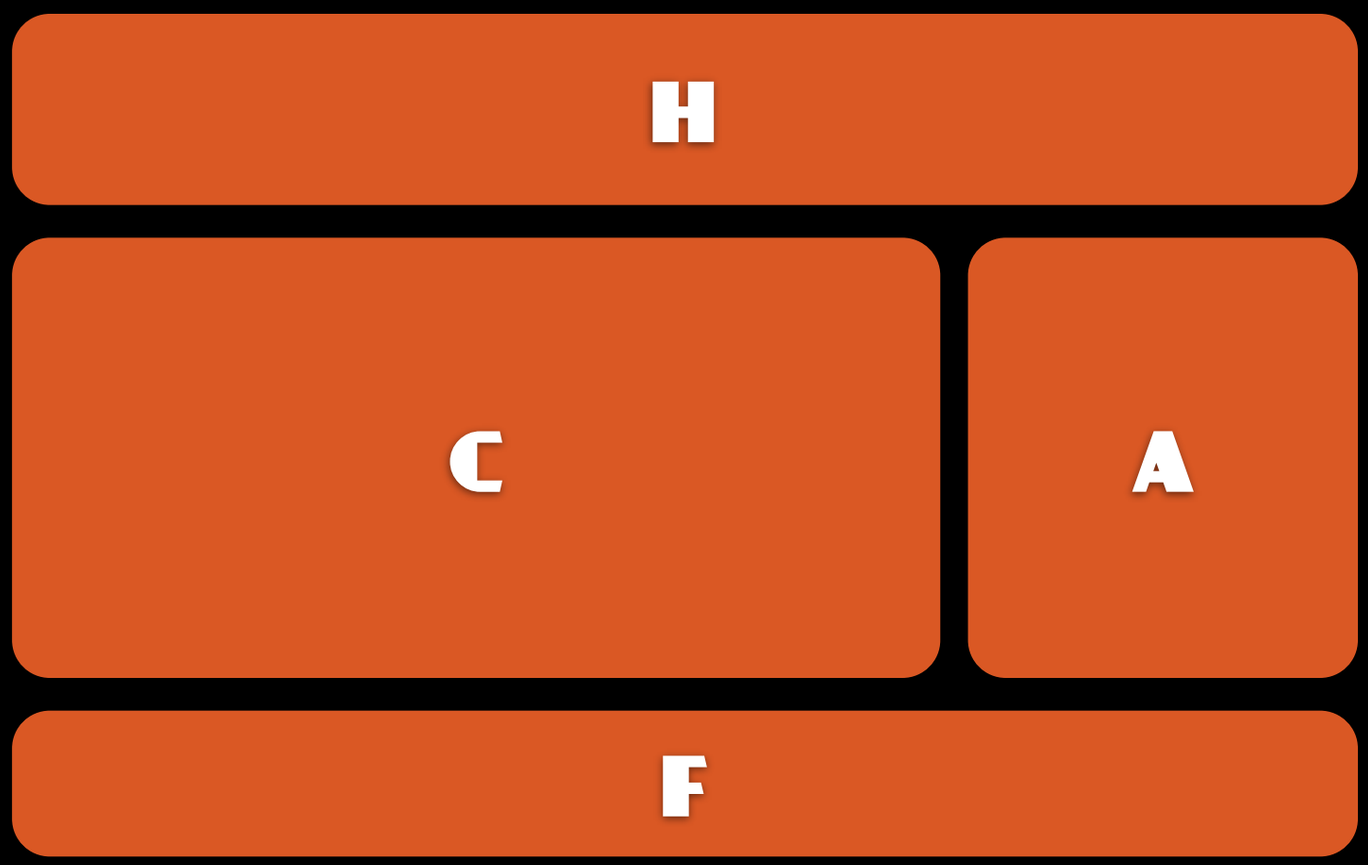


```
#page {
    display: grid;
    grid-columns: auto minmax(min-content, 1fr);
    grid-rows: auto minmax(min-content, 1fr) auto;
    }
header { grid-column: 1; grid-row: 1; grid-column-span: 2; }
#content { grid-column: 1, grid-row: 2; }
aside { grid-column: 2; grid-row: 2; }
footer { grid-column: 1; grid-row: 3; grid-column-span: 2; }
```

Grid layout module

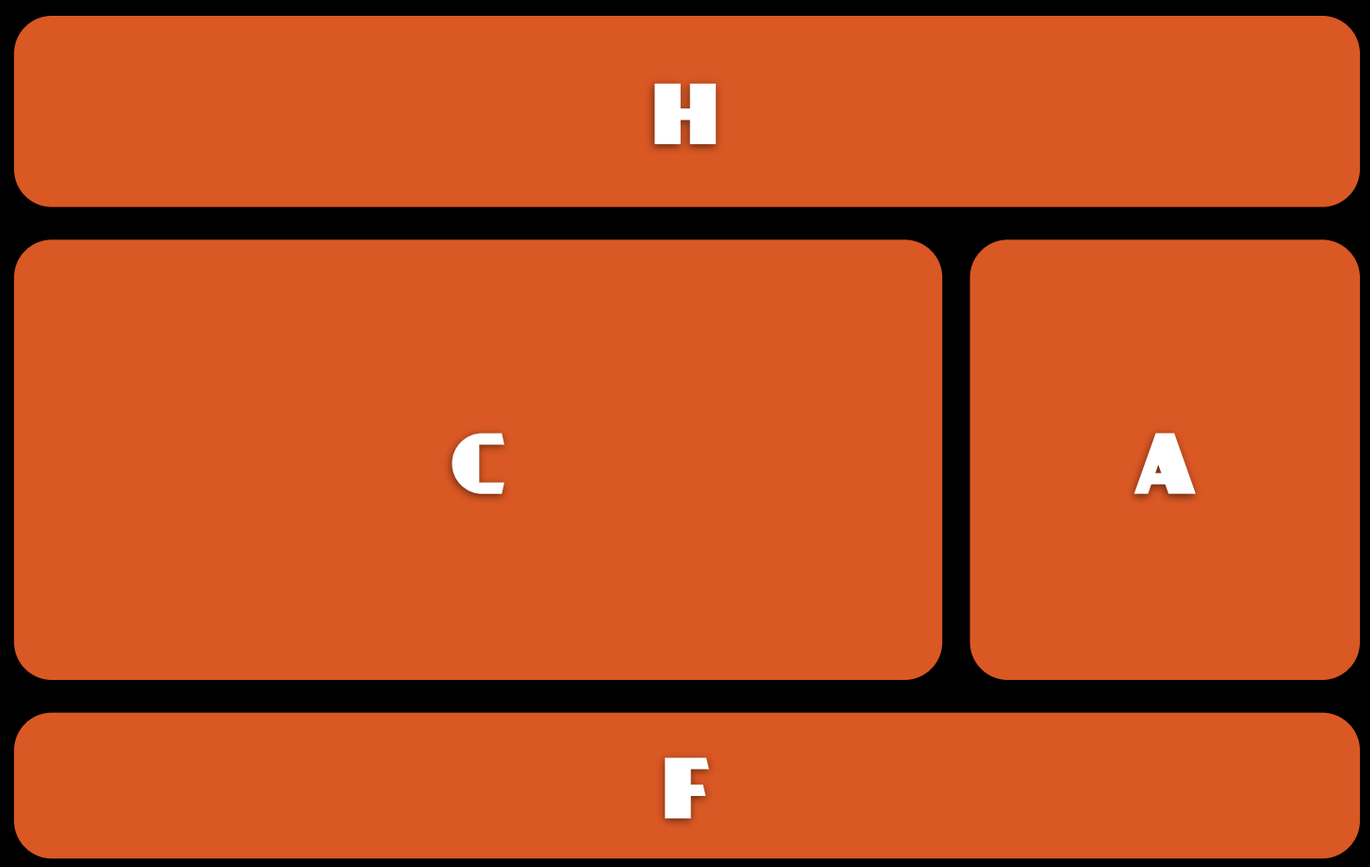
```
#page {
    display: grid;
    grid-columns: auto minmax(min-content, 1fr);
    grid-rows: auto minmax(min-content, 1fr) auto;
    }
header { grid-column: 1; grid-row: 1; grid-column-span: 2; }
#content { grid-column: 1, grid-row: 2; }
aside { grid-column: 2; grid-row: 2; }
footer { grid-column: 1; grid-row: 3; grid-column-span: 2; }
```

Grid layout module



```
#page {
    display: grid;
    grid-template: "hh"
                   "ca"
                   "ff";
    grid-columns: auto minmax(min-content, 1fr);
    grid-rows: auto minmax(min-content, 1fr) auto;
    }
header { grid-cell: "h"; }
#content { grid-cell: "c"; }
aside { grid-cell: "a"; }
footer { grid-cell: "f"; }
```

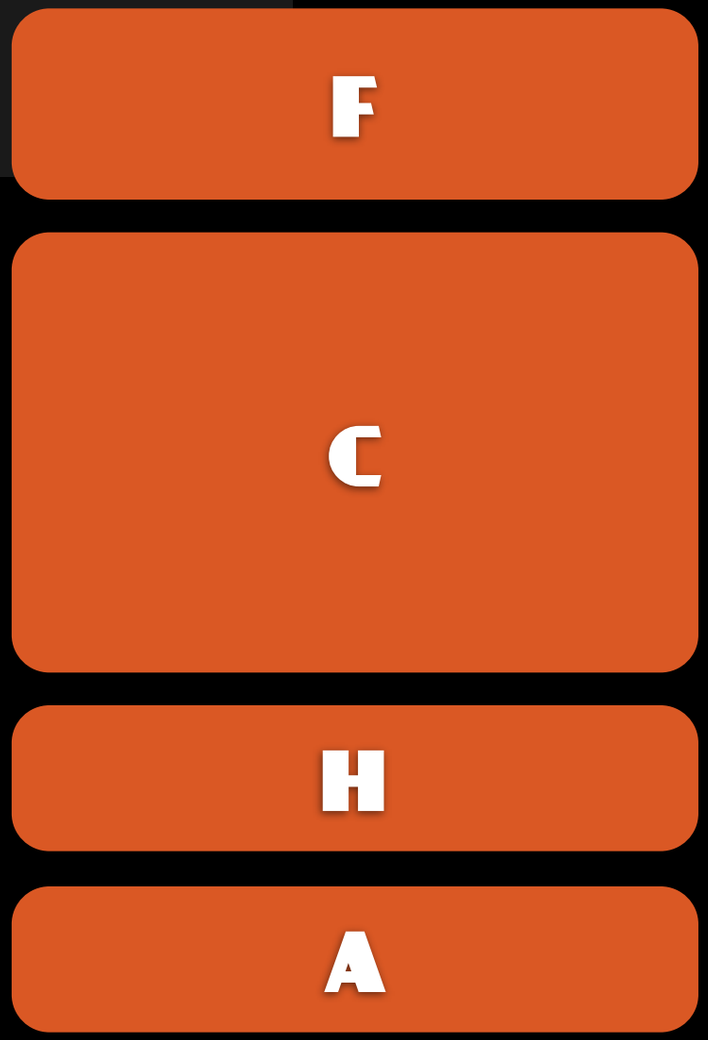
Grid layout module (templates)



```
@media screen and (max-width: 600px) {
    #page {
        grid-template: "f"
                       "c"
                       "h"
                       "a";
    }
}
```

Grid layout module (templates)

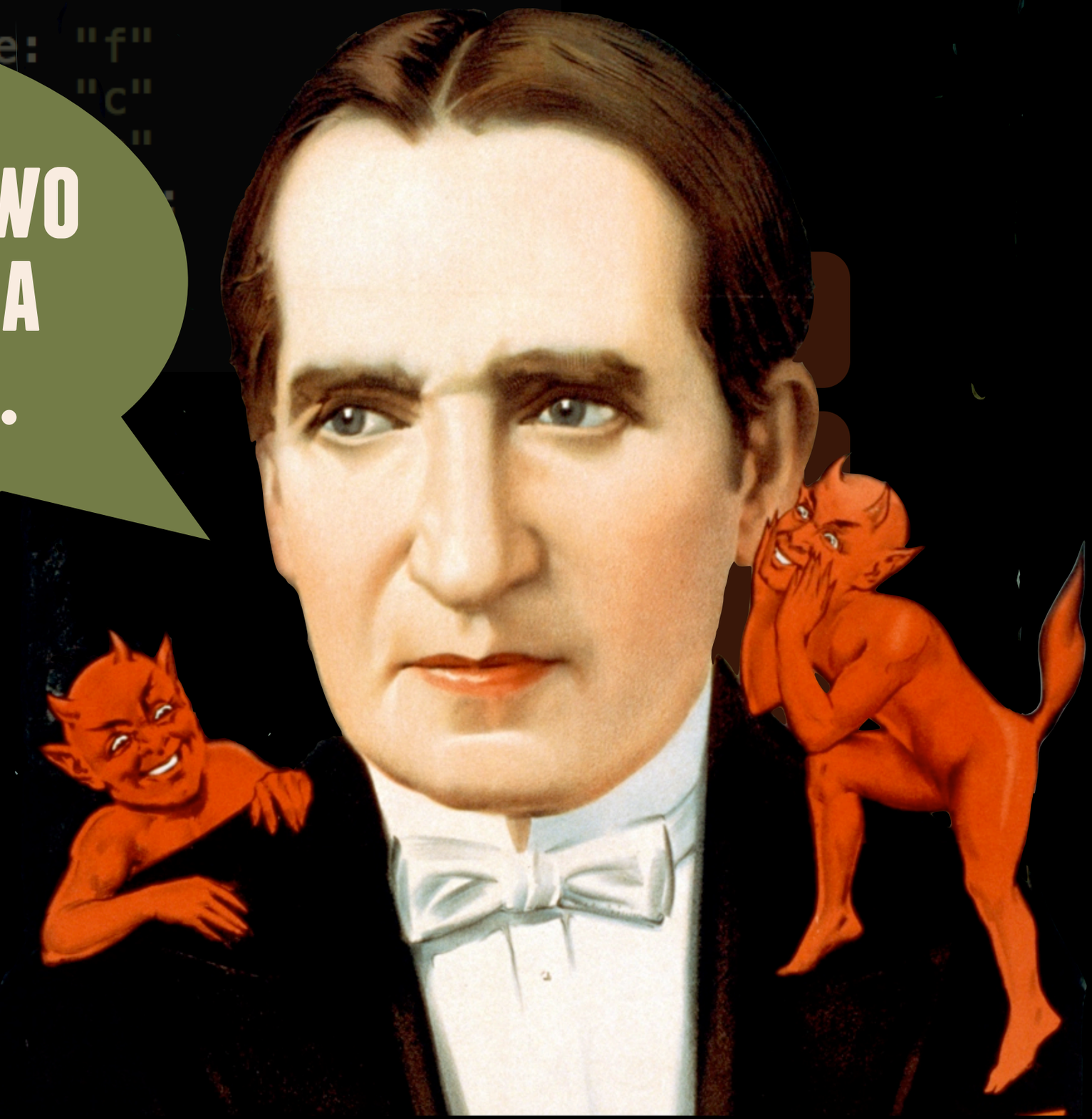
```
@media screen and (max-width: 600px) {
    #page {
        grid-template: "f"
                       "c"
                       "h"
                       "a";
    }
}
```



Grid layout module (templates)

```
@media screen and (max-width: 600px) {
    #page {
```

Not as cool as two bears fighting a hockey player.



Media queries / (meta) viewport

Media queries / (meta) viewport

fluid layouts

Media queries / (meta) viewport

fluid layouts

device/browser support profiles

Media queries / (meta) viewport

fluid layouts

device/browser support profiles

content reference wireframing

Media queries / (meta) viewport

fluid layouts

device/browser support profiles

content reference wireframing

layout breakpoint graphs

Media queries / (meta) viewport

fluid layouts

device/browser support profiles

content reference wireframing

layout breakpoint graphs

new layout mechanisms

Think in terms of design principles & systems

The art of letting go

We need to think up new ways to design layout, and new ways to communicate these new designs to our colleagues and our clients.

Thank you!

Starring magician Howard Thurston
as B. Rowser

http://www.delicious.com/stephenhay/mobilismconf2011
http://www.slideshare.net/stephenhay/mobilism2011

Thank you!

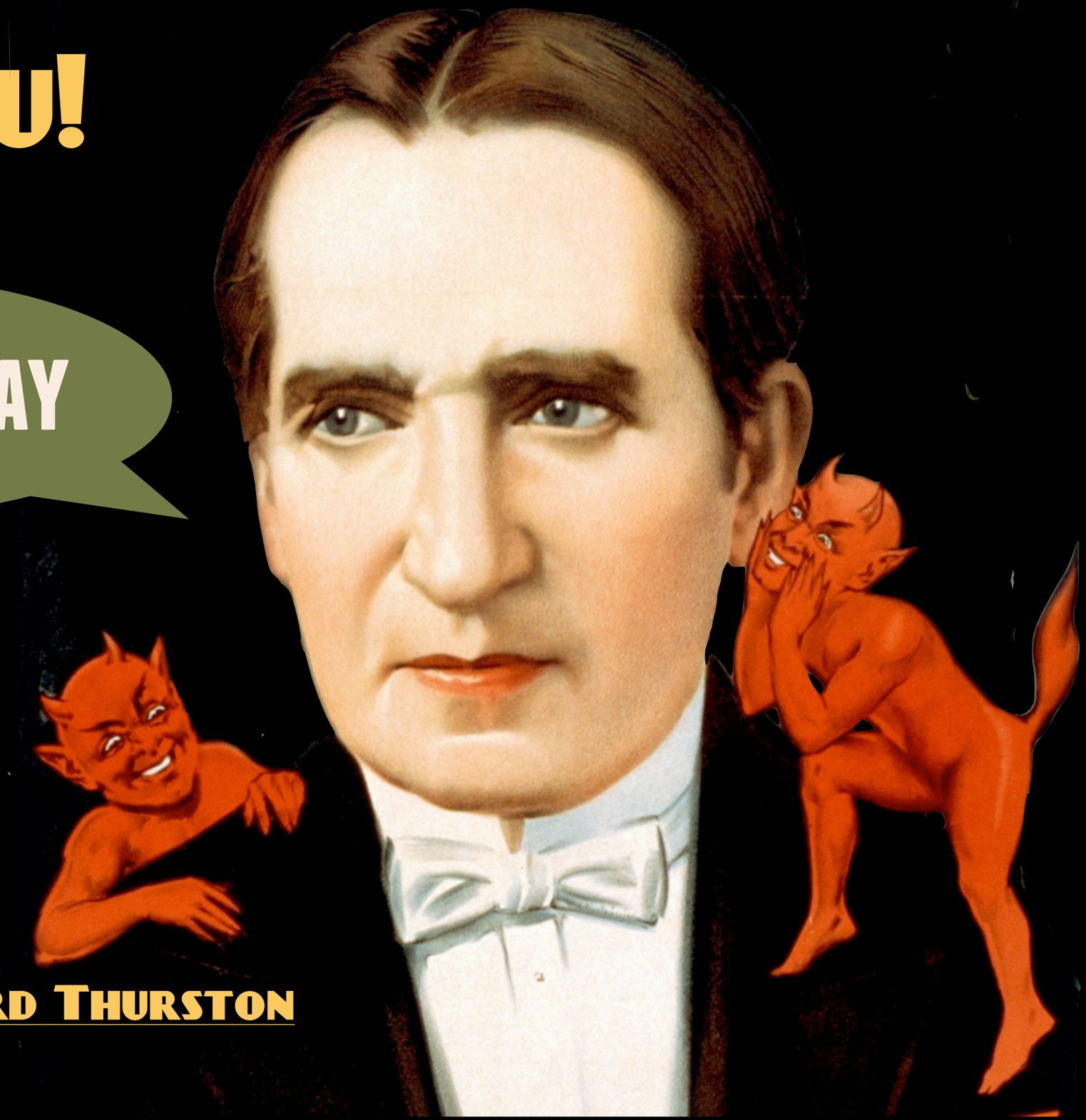
Starring magician Howard Thurston
as B. Rowser

http://www.delicious.com/stephenhay/mobilismconf2011

http://www.slideshare.net/stephenhay/mobilism2011

Thank you!

@stephenhay



Starring magician Howard Thurston as B. Rowser