



Frontend #4 23.2.2016



Today's agenda (1/2)

Time	Activity
8.30	Welcome and overview
8.40	Exercise from lecture 3 + learnings Semantic HTML
9.15	Break
9.30	Responsive Web Design
9.45	Exercise 1
10.30	Break



Today's agenda (2/2)

Time	Activity
10.45	Exercise 2
11.30	Lunch
12.00	Exercise 3
13.45	Lessons learned today and next lecture



- Understand the main elements of the DOM
- Understand which tags that may be children of which tags (common content model)
- Understand the separation between markup and styling
- Understand how styling is injected in markup
- Understand importance, specificity and inheritance principles in CSS
- Use selected display properties (block, none, inline, inline-block)
- Understand the position types (static, absolute, relative, initial, inherit)
- Use and apply floating layout (float, clear)
- Use and apply the box model (and look further into flexbox and grid)
- Understand overflow
- Use color codes and names
- Use and apply background images
- Detect which browsers that renders which principles (use of caniuse.com and apply reset.css)
- Use validation of HTML and CSS including understand error messages in order to make adjustment of the code

Lecture 3 - learning objectives

- Understand the main elements of the DOM
- Understand which tags that may be children of which tags (common content model)
- Understand the separation between markup and styling
- Understand how styling is injected in markup
- Understand importance, specificity and inheritance principles in CSS
- Use selected display properties (block, none, inline, inline-block)
- Understand the position types (static, absolute, relative, initial, inherit)
- Use and apply floating layout (float, clear)
- Use and apply the box model (and look further into flexbox and grid)
- Understand overflow

<https://app.pluralsight.com/library/courses/css-positioning-1834/table-of-contents>

CSS Positioning

by Susan Simkins

Lecture 3 - learning objectives



- Understand the main elements of the DOM
- Understand which tags that may be children of which tags (common content model)
- Understand the separation between markup and styling
- Understand how styling is injected in markup
- Understand importance, specificity and inheritance principles in CSS
- Use selected display properties (block, none, inline, inline-block)
- Understand the position types (static, absolute, relative, initial, inherit)
- Use and apply floating layout (float, clear)
- Use and apply the box model (and look further iinto flexbox and grid)
- Understand overflow
- Use color codes and names

The CSS Overflow Property



CHRIS COYIER // MAY 27, 2009

<https://css-tricks.com/the-css-overflow-property/>

Lecture 3 - learning objectives

- Understand the main elements of the DOM
- Understand which tags that may be children of which tags (common content model)
- Understand the separation between markup and styling
- Understand how styling is injected in markup
- Understand importance, specificity and inheritance principles in CSS
- Use selected display properties (block, none, inline, inline-block)
- Understand the position types (static, absolute, relative, initial, inherit)
- Use and apply floating layout (float, clear)
- Use and apply the box model (and look further into flexbox and grid)
- Understand overflow
- Use color codes and names
- Use and apply background images

HTML & CSS TUTORIALS

Want to learn how to use color in your website? We've assembled a great set of short and easy tutorials that cover the basics of HTML, CSS and SCSS.

Lecture 3 - learning objectives

<http://htmlcolorcodes.com/tutorials/>

- Understand the main elements of the DOM
- Understand which tags that may be children of which tags (common content model)
- Understand the separation between markup and styling
- Understand how styling is injected in markup
- Understand importance, specificity and inheritance principles in CSS
- Use selected display properties (block, none, inline, inline-block)
- Understand the position types (static, absolute, relative, initial, inherit)
- Use and apply floating layout (float, clear)
- Use and apply the box model (and look further into flexbox and grid)
- Understand overflow
- Use color codes and names
- Use and apply background images
- Detect which browsers that renders which principles (use of caniuse.com and apply reset.css)
- Use validation of HTML and CSS including understand error messages in order to make adjustment of the code

<https://css-tricks.com/perfect-full-page-background-image/>

<https://developer.mozilla.org/en-US/docs/Web/CSS/background-image>

<https://developer.mozilla.org/en-US/docs/Web/CSS/background>

Lecture 3 - learning objectives

Exercise from lecture 3



Present your solution to the person next to you

How far did you get? (demo and code review)

Where were you challenged?

What may be unclear to you (HTML and CSS)?

Take notes and be ready to present the three most important findings for me at the table.

Exercise from lecture 3 (10 mins)



Semantic HTML



All five levels of “Journey into Mobile”

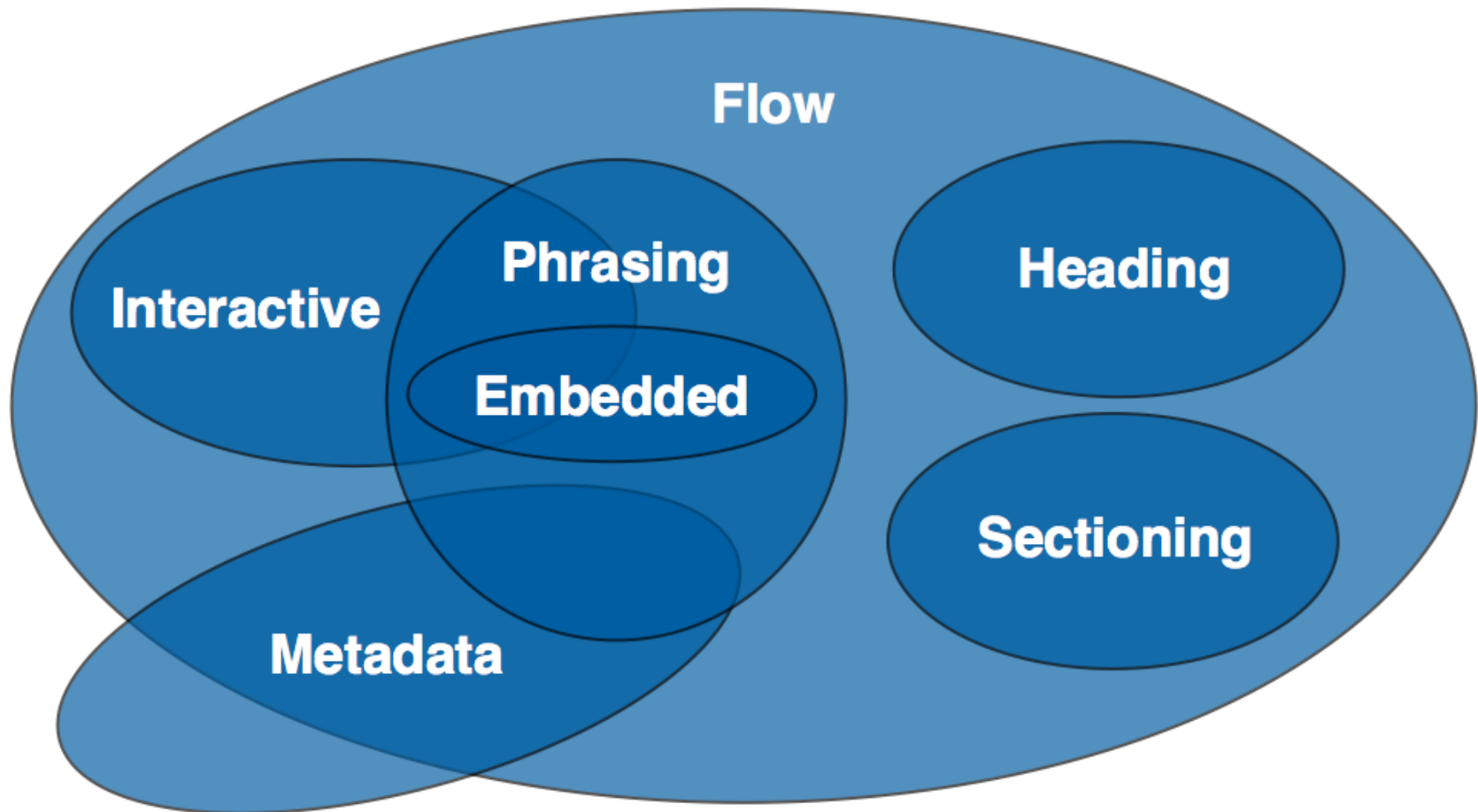


Semantic HTML (all videos)

CSS Animation with Transition and Transform (ch. 1)

**Chapter 1-4 in “Jump start Responsive Web Design”
(available at Fronter)**

Preparation



Visualisation of content categories

<https://www.w3.org/TR/html5/dom.html#content-models>

How does semantic HTML differ from non-semantic?

Who benefits from semantic HTML?

What is an HTML outline?

What are sectioning and grouping elements?

What is the purpose of heading elements?

What is Semantic HTML and why?
(5 questions)



Sectioning Elements

Elements whose intended use is to divide content up into logical sections.

<main>

<nav>

<footer>

<article>

<aside>

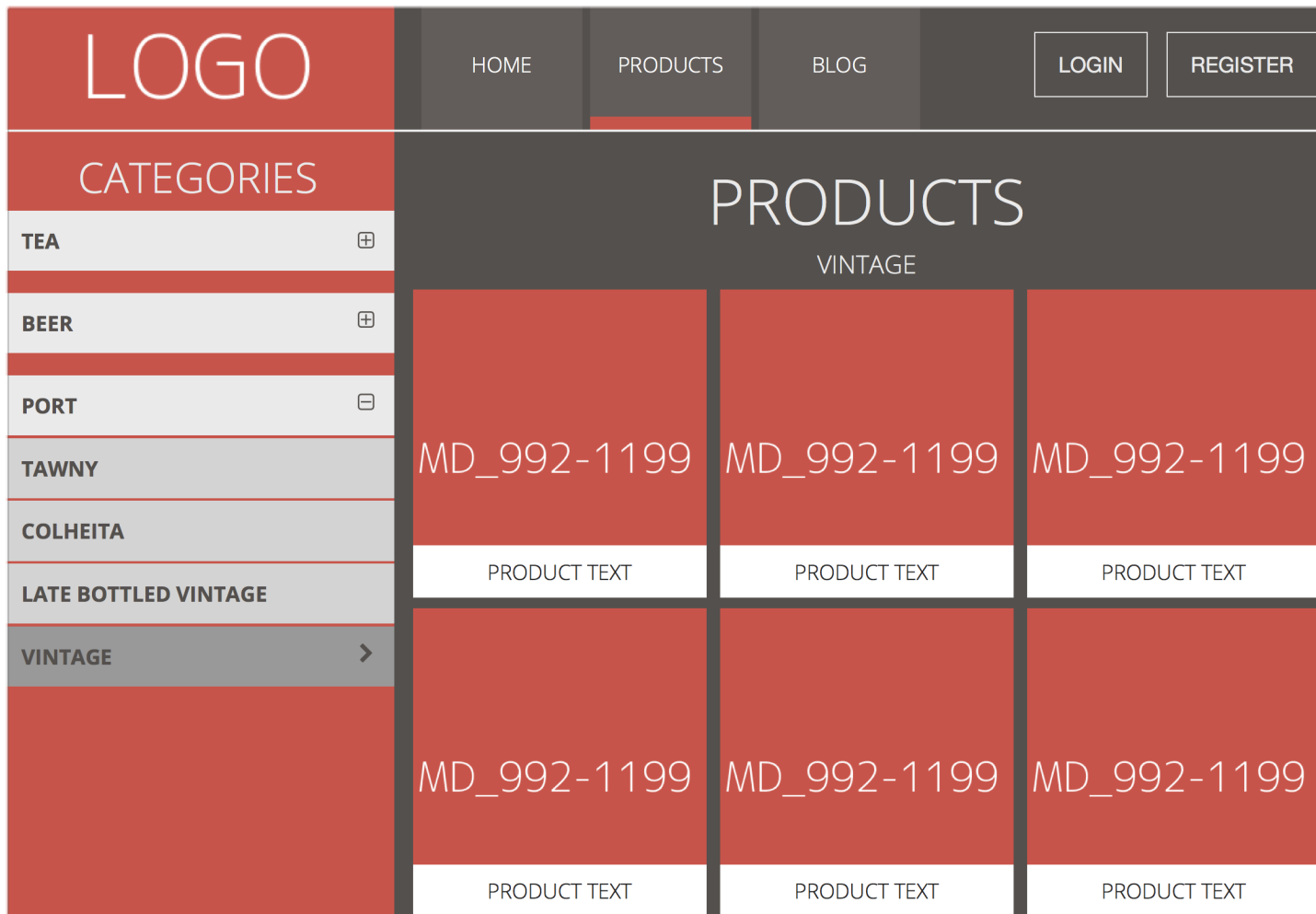
<address>

<section>

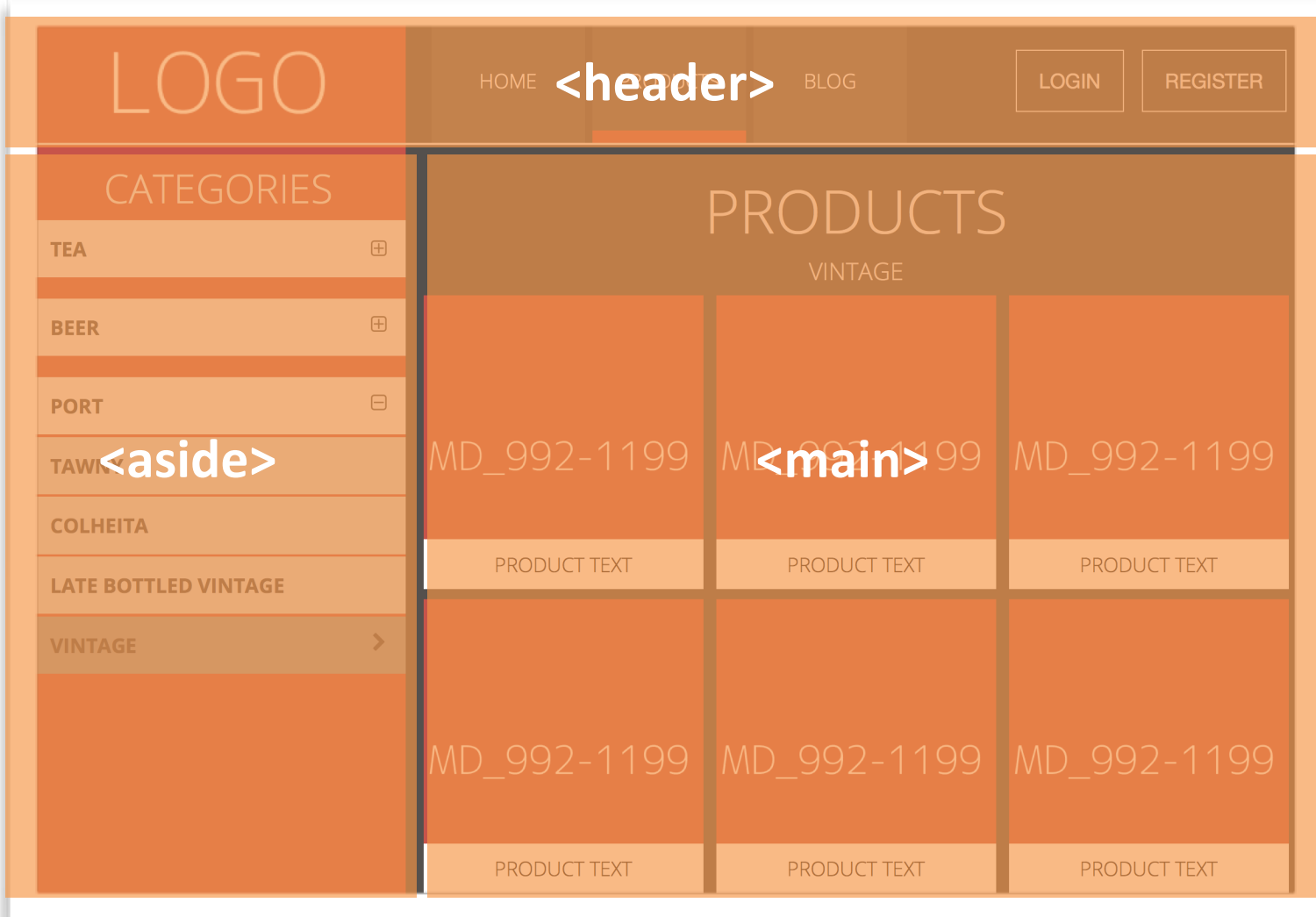
<header>

**Contact information -
not postal address**

Sectioning

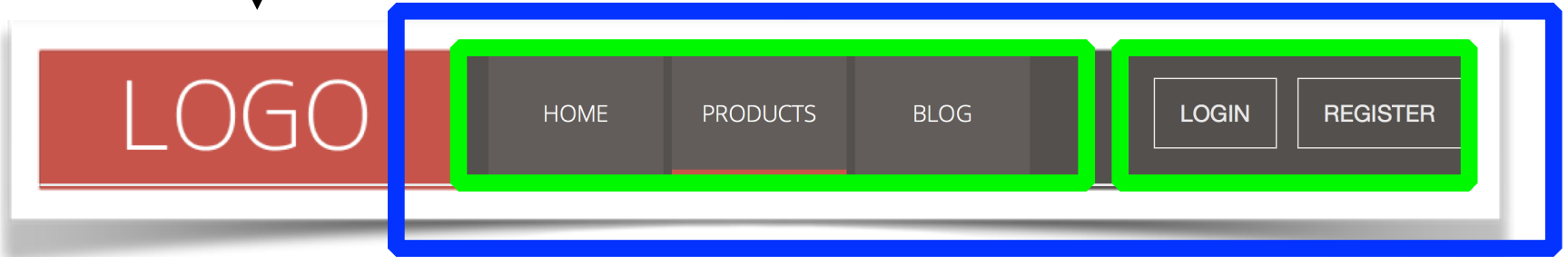


Which elements should be on first level?



Which elements should be on first level?

Background image (set in CSS)



`<nav>`

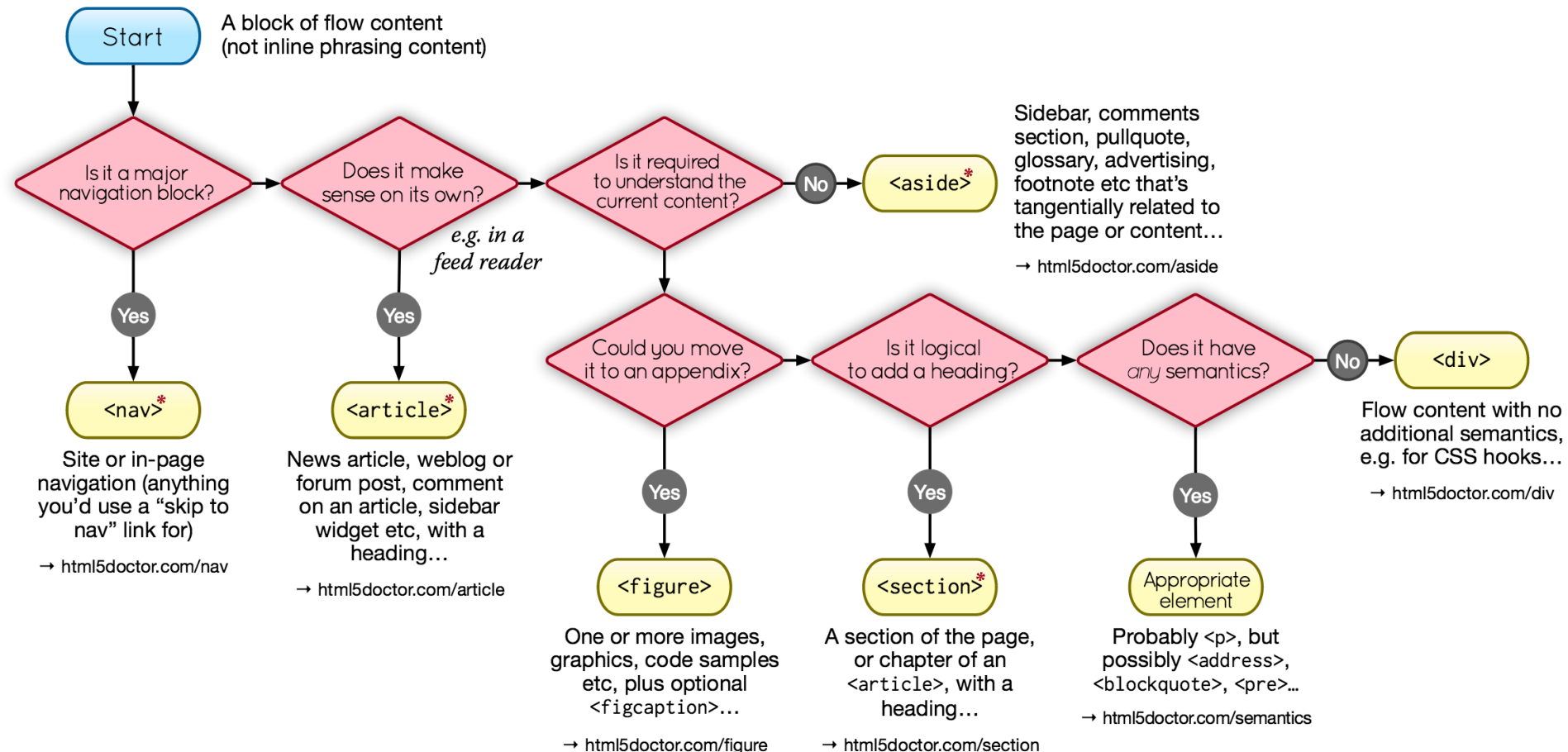
Header elements



HTML5 Element Flowchart

Sectioning content elements and friends

By @riddle & @boblet
www.html5doctor.com



*Sectioning content element

These four elements (and their headings) are used by HTML5's outlining algorithm to make the document's outline
→ html5doctor.com/outline

Kahoot!

Responsive Web Design



Understand and use the four elements of RWD

Media Queries (@media rules)

Fluid Grid (px -> %)

Flexible Images (max-width: 100%; height: auto)

Viewport meta tag

Hiding and showing DOM elements

Learning objectives



Fluid Grid (percentages instead of pixels)

Flexible Images (max-width: 100%; height: auto)

Media Queries (@media)

Viewport meta tag

Central elements of RWD



Design mobile first



CSS selectors as usual

@media and (min-width: 600px) { ... }

@media and (min-width: 1024px) { ... }

Mobile First

Screen resolutions

320px	mobile portrait
480px	mobile landscape
600px	small tablet
768px	tablet portrait
1024px	tablet landscape/netbook/desktop
1280px and greater	desktop large

Screen resolutions (Jumpstart, p. 96)

	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
<code>.visible-xs-*</code>	Visible	Hidden	Hidden	Hidden
<code>.visible-sm-*</code>	Hidden	Visible	Hidden	Hidden
<code>.visible-md-*</code>	Hidden	Hidden	Visible	Hidden
<code>.visible-lg-*</code>	Hidden	Hidden	Hidden	Visible
<code>.hidden-xs</code>	Hidden	Visible	Visible	Visible
<code>.hidden-sm</code>	Visible	Hidden	Visible	Visible
<code>.hidden-md</code>	Visible	Visible	Hidden	Visible
<code>.hidden-lg</code>	Visible	Visible	Visible	Hidden

Screen resolutions (getbootstrap.com)

Viewport

vh - 1/100 viewport height

vw - 1/100 viewport width

vmin - 1/100 smallest side of viewport

vmax - 1/100 largest side of viewport

1 vmax = 1vh in portrait mode

1 vmax = 1vw in landscape mode

Viewport units

**Is there anything you need to know
before you can create a responsive website?**

Questions to RWD and Mobile First



Exercise Responsive Web Design

Fluid Grid, Flexible Images, Media Queries, Viewport meta

Good practice for CSS file order

Font Awesome, dummyimage, lorempixel

<picture> element

overflow: auto - 100% height of html, body

Topics I might talk about during the exercise



General elements (body, h1, aside...)

Generic UI elements (.pull-right, .hidden-sm...)

Overwrites of 3rd party CSS (i.e. Font Awesome)

Follow the structure of the markup (header, aside, main, footer...)

Media queries

Good practice for CSS order

BUSINESS ACADEMY
AARHUS

<http://code.tutsplus.com/tutorials/30-css-best-practices-for-beginners--net-6741>

```
4  /** GENERAL ELEMENTS **/  
5  
6 ▶ html, body { ... }  
9  
10 ▶ body { ... }  
14  
15 ▶ h1, h2, h3 { ... }  
18  
19 ▶ h1 { ... }  
24  
25 ▶ h2 { ... }  
28  
29 ▶ h3 { ... }  
33  
34 ▶ img { ... }  
38  
39 ▶ aside { ... }  
45  
46 ▶ main { ... }
```

Example of structure

```

50  /** GENERAL UI ELEMENTS **/
51
52 ▶ .container { ... }
59
60 ▶ .btn { ... }
69
70 ▶ .btn-default { ... }
75
76 ▶ .btn-default:hover { ... }
80
81 ▶ .btn-category { ... }
89
90 ▶ .btn-category-child { ... }
97
98 ▶ .btn.toggled { ... }
101
102 ▶ .pull-left { ... }
106
107 ▶ .pull-right { ... }
111
112 ▶ .hidden { ... }
115
116 ▶ .hidden-xs, .hidden-sm, .hidden-md, .hidden-lg { ... }
119
120 .visible-sm-block, .visible-sm-inline-block, .visible-sm-inline,
121 .visible-md-block, .visible-md-inline-block, .visible-md-inline,
122 ▶ .visible-lg-block, .visible-lg-inline-block, .visible-lg-inline { ... }
125

```

Example of structure

```
138  /** FONT AWESOME OVERWRITES **/  
139  
140 ► .fa.pull-left, .fa.pull-right { ... }  
143
```

Example of structure



```
179  /** ASIDE - CATEGORIES MENU **/  
180  
181  
182 ▶ aside h2 { ... }  
186  
187 ▶ aside .fa { ... }  
190  
191 ▶ aside .btn { ... }  
194  
195 ▶ .category { ... }  
198  
199 ▶ .category:first-child { ... }  
202  
203 ▶ .category-parent { ... }  
206  
207 .category-child { /* ul */  
208 }  
209  
210 ▶ .category-child li { ... }  
213  
214 ▶ .selected { ... }
```

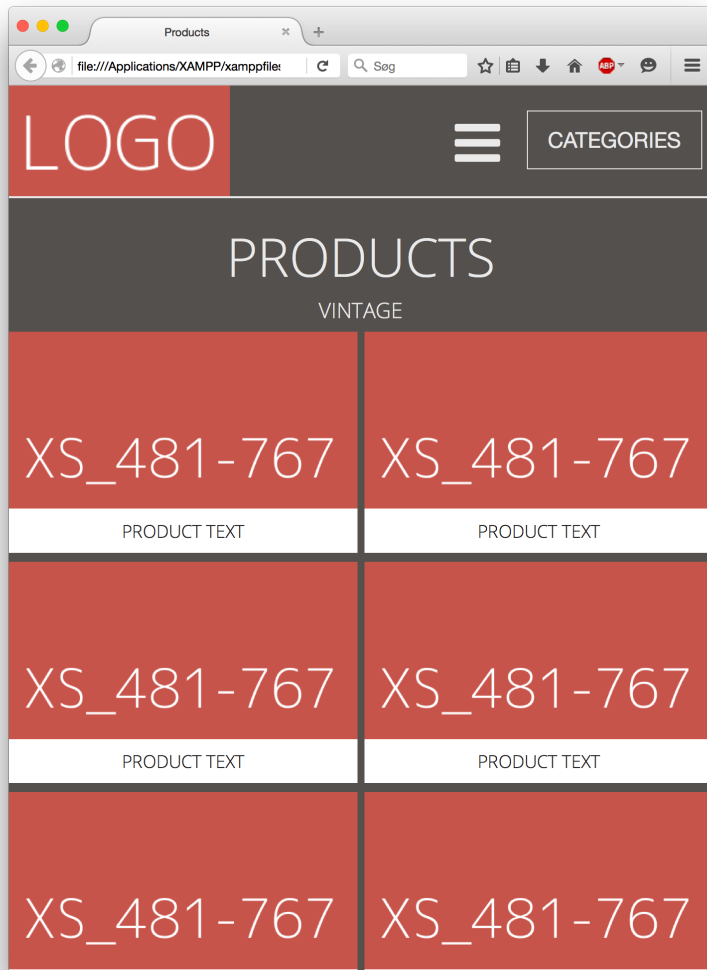
Example of structure

```
219  /** MAIN **/  
220  
221 ▶ .products { ... }  
224  
225 ▶ .product { ... }  
234  
235 ▶ .product p { ... }
```

Example of structure

```
247  /***** EXTRA SMALL (481-767px) *****/
248
249 ▶ @media screen and (min-width: 481px) { ... } /* EXTRA SMALL 481-767 END */
267
268  /***** SMALL (768 - 991px) *****/
269
270 ▶ @media screen and (min-width: 768px) { ... } /* SMALL END */
296
297
298  /***** MEDIUM (992 - 1199px) *****/
299
300 ▶ @media screen and (min-width: 992px) { ... } /* MEDIUM END */
419
420
421  /***** LARGE (> 1200px) *****/
422
423 ▶ @media screen and (min-width: 1200px) { ... } /* LARGE END */
```

Example of structure

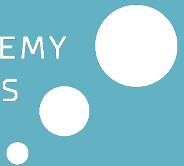


**Exercise on Fronter
in today's folder.**

(with hints)

Mobile First - Products page - exercise

Lessons Learned



Understand and use the four elements of RWD

Media Queries (@media rules)

Fluid Grid (px -> %)

Flexible Images (max-width: 100%; height: auto)

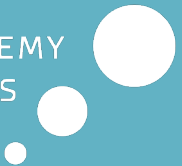
Viewport meta tag

Hiding and showing DOM elements

Learning objectives



Next lecture (Colors - then JavaScript)



Next time:

Topic: Colours to set the mood

+

The Basics of web navigation.

Read / watch:

[Dikun] 16 min video on colour – Must watch all three! [##](#)

[Krug] Chapter 6. Pages 55 – 83 on navigation – Must Read!













[Price] 23 min video on colour – With great examples [##](#)

Preparation (next Tuesday - Colours)



JavaScript Fundamentals

by Liam McLennan

	Introduction to JavaScript		18m 46s	
	The Building Blocks of JavaScript Programs		26m 40s	
	Functions		20m 35s	
	Control Flow		14m 9s	

Preparation 15.3 (JavaScript)

BUSINESS ACADEMY
AARHUS

