Responsive Web Design

Agenda

- Mobile Web Design
- Responsive Web Design
- CSS Styling for Print

Three Approaches to Mobile Web

1. Create a **separate website** hosted within your current domain targeted for mobile users, like:
   – m.whitehouse.gov
   – m.jyllands-posten.dk
   – mobil.bold.dk

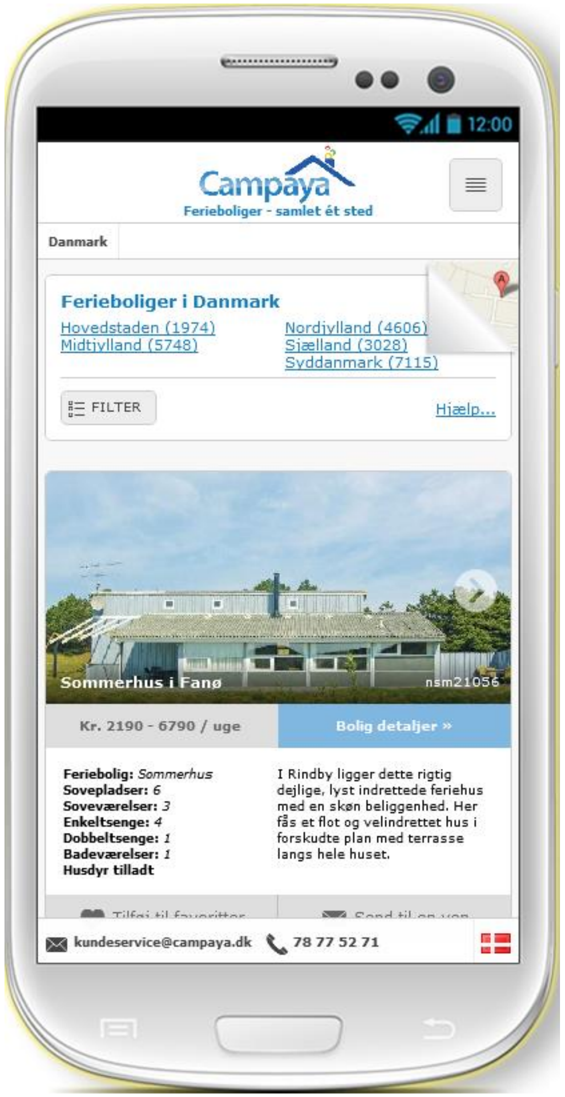
2. Sites that dynamically serve all devices on the same set of URLs, but each URL serves different HTML and CSS depending on whether the **user agent** is a desktop or a mobile device

3. **Use CSS** to configure your current website for display on both mobile, tablets and desktop devices
   – **This is Google's recommended configuration**

https://developers.google.com/webmasters/smartphone-sites/details

MOBILE WEB DESIGN

HTML5 compatibility on mobile and tablet browsers:
http://mobilehtml5.org/

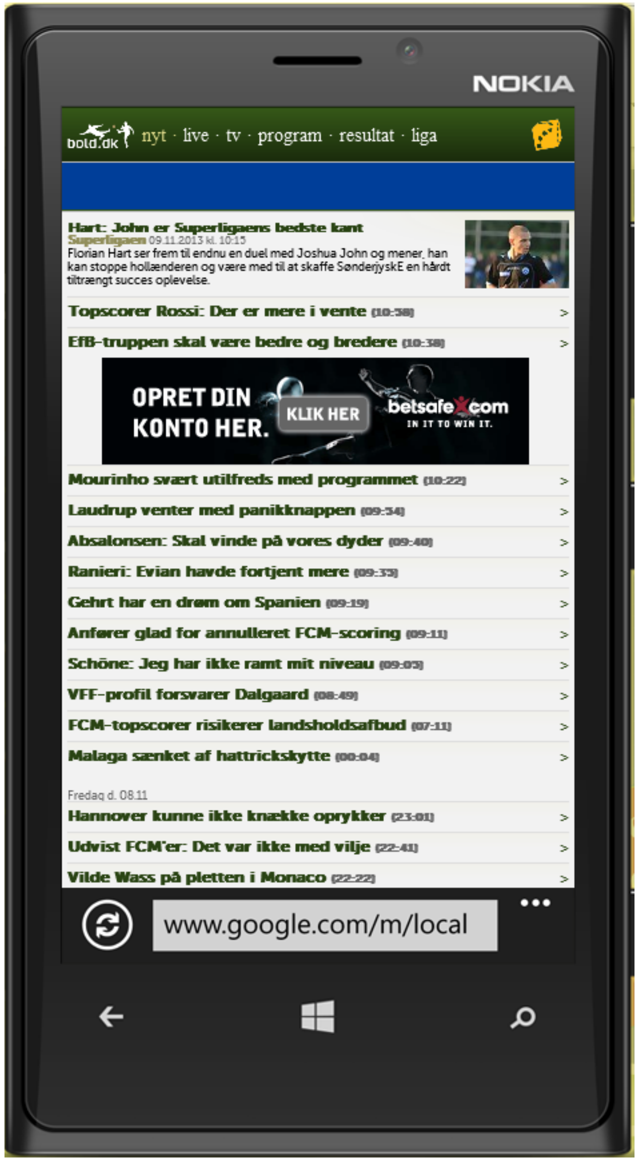


Feriebolig: *Sommerhus*
Sovepladser: *6*
Soveværelser: *3*
Enkeltsenge: *4*
Dobbeltsenge: *1*
Badeværelser: *1*
Husdyr tilladt

I Rindby ligger dette rigtig dejlige, lyst indrettede feriehus med en skøn beliggenhed. Her fås et flot og velindrettet hus i forskudte plan med terrasse langs hele huset.

Mobile Web Limitations

- Small Screen Size
- Maybe low bandwidth
- Limited fonts
- Awkward controls
- Lack of Flash support
- Cost per kilobyte
- Limited processor and memory
- (Limited color)



Hart: John er Superligaens bedste kant
Superligaen 09.11.2013 kl. 10:15
Florian Hart ser frem til endnu en duel med Joshua John og mener, han kan stoppe hollænderen og være med til at skaffe SønderjyskE en hårdt tiltrængt succes oplevelse.

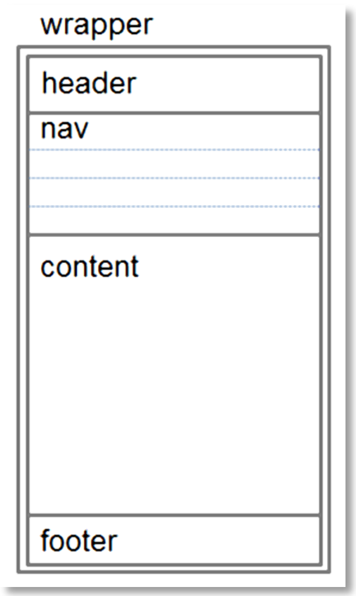
Topscorer Rossi: Der er mere i vente (10:38) >
EfB-truppen skal være bedre og bredere (10:38) >
Mourinho svært utilfreds med programmet (10:22) >
Laudrup venter med panikknappen (09:54) >
Absalonsen: Skal vinde på vores dyder (09:40) >
Ranieri: Evian havde fortjent mere (09:35) >
Gehrt har en drøm om Spanien (09:19) >
Anfører glad for annulleret FCM-scoring (09:11) >
Schöne: Jeg har ikke ramt mit niveau (09:05) >
VFF-profil forsvarer Dalgaard (08:49) >
FCM-topscorer risikerer landsholdsafbud (07:11) >
Malaga sænket af hattrickskytte (00:04) >

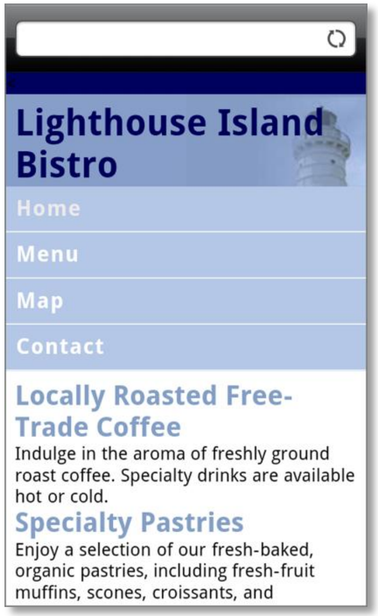
Fredag d. 08.11
Hannover kunne ikke knække oprykker (23:01) >
Udvist FCM'er: Det var ikke med vilje (22:41) >
Vilde Wass på pletten i Monaco (22:22) >

www.google.com/m/local

Design Techniques for Mobile Web

- Single column design
- Avoid floats, tables, frames
- Descriptive page title
- Descriptive heading tags
- Optimize images
- Descriptive alt text for images
- Eliminate unneeded images
- Navigation in lists
- Em or percentage font size units
- Common font typefaces
- Good contrast between text and background colors
- Provide "Skip to Content" hyperlink
- Provide "Back to Top" hyperlink





Locally Roasted Free-Trade Coffee

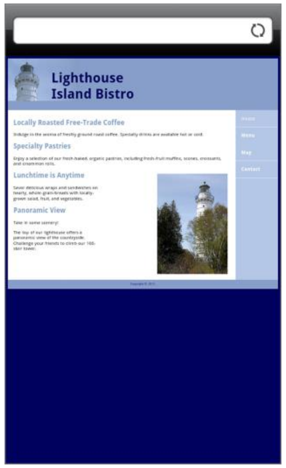
Indulge in the aroma of freshly ground roast coffee. Specialty drinks are available hot or cold.

Specialty Pastries

Enjoy a selection of our fresh-baked, organic pastries, including fresh-fruit muffins, scones, croissants, and

Viewport Meta Tag

- Default action for most mobile devices is to zoom out and scale the web page

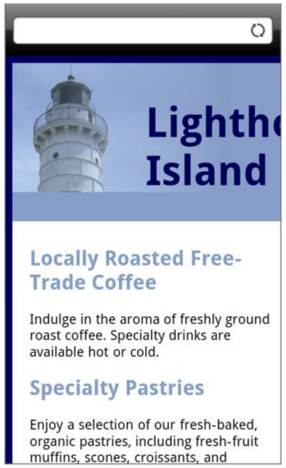


- Viewport Meta Tag
  - Created as an Apple extension to configure display on mobile devices
  - Configures width and initial scale of browser viewport

```
<meta name="viewport"
    content="width=device-width,initial-scale=1.0">
```

  - Or

```
<meta name="viewport" content="width=device-width,
minimum-scale=1.0, maximum-scale=1.0">
```



Locally Roasted Free-Trade Coffee

Indulge in the aroma of freshly ground roast coffee. Specialty drinks are available hot or cold.

Specialty Pastries

Enjoy a selection of our fresh-baked, organic pastries, including fresh-fruit muffins, scones, croissants, and

Telephone & Text Message Hyperlinks

- Telephone Scheme:

  ```
  <a href="tel:888-555-5555">Call 888-555-5555</a>
  ```

  - Many mobile browsers will initiate a phone call when the hyperlink is clicked

- SMS Scheme:

  ```
  <a href="sms:888-555-5555">Text 888-555-5555</a>
  ```

  - Many mobile browsers will initiate a text message to the phone number when the hyperlink is clicked

Make Your Mobile Pages Faster

- Google Analytics data shows that the average web page takes over 10 seconds to load on mobile
- We know that a user’s thought process is typically interrupted after waiting for just one second, resulting in that user starting to become disengaged
- So at a minimum:

the “above the fold” content of a web page

should render in less than one second

Rendering in under one second

- If we estimate 3G network round trip time at 250ms
  - 4G RTT is 55ms (20 – 170 depending on operator and location)
  - Cable RTT is 21ms
- Assuming no blocking external JavaScript or CSS, we incur three round trips for DNS, TCP, and request/response, for a total of 750ms
- Plus 100ms for backend time
- This brings us to 850ms
- As long as render-blocking JavaScript and CSS is inlined and the size of the initial HTML payload is kept to a minimum (e.g. under 15kB compressed), the time it takes to parse and render should be well under 100ms, bringing us in at 950ms, just under our one second target

http://calendar.perfplanet.com/2012/make-your-mobile-pages-render-in-under-one-second/

http://calendar.perfplanet.com/2013/network-latency-4g/

Optimized Page Structure

```
<html>
<head>
  <style>
    .main { ... }
    .leftnav { ... }
    /* ..any other styles needed for the initial render here */
  </style>
  <script>
    // Any script needed for initial render here.
    // Ideally, there should be no JS needed for the initial render
  </script>
</head>
<body>
  <div class="topnav"> Perhaps there is a top nav bar here. </div>
  <div class="main">
     Here is my content. </div>
  ...
  <link rel="stylesheet" href="my_leftover.css">
  <script src="my_leftover.js"></script>
</body>
</html>
```

Speed Summary

To make your mobile web page render in under one second, you should:

- Keep server backend time to generate HTML to a minimum (under 100ms)
- Avoid HTTP redirects for the main HTML resource
- Avoid loading blocking external JavaScript and CSS before the initial render
- Inline just the JavaScript and CSS needed for the initial render
- Delay or async load any JavaScript and CSS not needed for the initial render
- Keep HTML payload needed to render initial content to under 15kB compressed

Google label 'mobile friendly' websites

- Google:

  "Websites can earn a "mobile-friendly" badge that is displayed with search results"
- Requirements:
  - avoids requiring software such as Adobe Systems' Flash
  - text is readable without zooming,
  - content is sized so that horizontal and vertical scrolling isn't required
  - links are spaced far enough apart for easy tapping
- Google has a page to test if a site will be considered mobile friendly: https://www.google.com/webmasters/tools/mobile-friendly/
- Guide: https://developers.google.com/webmasters/mobile-sites/?utm_source=wmc-blog&utm_medium=referral&utm_campaign=mobile-friendly

Testing Mobile Display

- Web-Based Iphone Emulator
  - www.testiphone.com
- Mobilizer
  - www.springbox.com/mobilizer
- Web-Based Opera Mini Simulator
  - http://www.opera.com/da/developer/opera-mini-simulator

RESPONSIVE WEB DESIGN

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

From Wikipedia

Why Responsive Design

- **Using a single URL for a piece of content makes it easier for your users to interact with, share, and link to your content**, and a single URL for the content helps Google's algorithms assign the indexing properties for the content
- No redirection is needed for users to get to the device-optimized view, which reduces loading time. Also, user agent-based redirection is error-prone and can degrade your site's user experience
- It saves resources for both your site and Google's crawlers. For responsive web design pages, any Googlebot user agents needs to crawl your pages once, as opposed to crawling multiple times with different user agents, to retrieve your content. This improvement in crawling efficiency can indirectly help Google index more of the site's contents and keep it appropriately fresh

The Goal for Responsive Design

1. The pages should render legibly at any screen resolution
   - From 320 px and up
2. We mark up one set of content, making it viewable on any device
   - Only one version of every html file
3. We should never show a horizontal scrollbar, whatever the window size
   - From 320 px and up

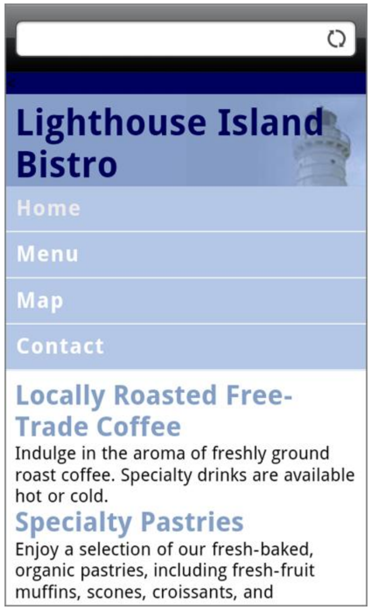
Responsive web design – How TO?

- A site designed with RWD adapts the layout to the viewing environment by using:
  1. CSS3 media queries
  2. Fluid, proportion-based grid layout
  3. Flexible images

CSS3 Media Queries

- Determines the capability of the mobile device, such as screen resolution
- Directs the browser to styles configured specifically for those capabilities
- Example:

```
<link href="lighthousemobile.css" rel="stylesheet"
      media="only screen and (max-device-width: 640px)">
```



Lighthouse Island Bistro

Home
Menu
Map
Contact

Locally Roasted Free-Trade Coffee

Indulge in the aroma of freshly ground roast coffee. Specialty drinks are available hot or cold.

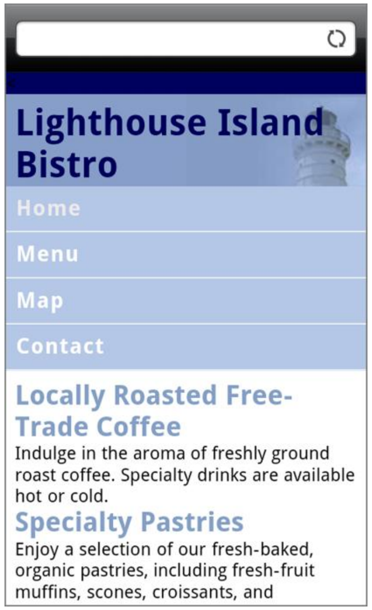
Specialty Pastries

Enjoy a selection of our fresh-baked, organic pastries, including fresh-fruit muffins, scones, croissants, and

CSS3 Media Queries

- You can also include Media Queries in your CSS as part of a @media rule:
- Example:

```
@media screen and (max-device-width: 640px){
  .column {
    float: none;
  }
}
```


Lighthouse Island Bistro

Locally Roasted Free-Trade Coffee

Indulge in the aroma of freshly ground roast coffee. Specialty drinks are available hot or cold.

Specialty Pastries

Enjoy a selection of our fresh-baked, organic pastries, including fresh-fruit muffins, scones, croissants, and

CSS3 Media Queries - Retina

- Recent Webkit-specific media query to target the iPhone's high-resolution Retina display:

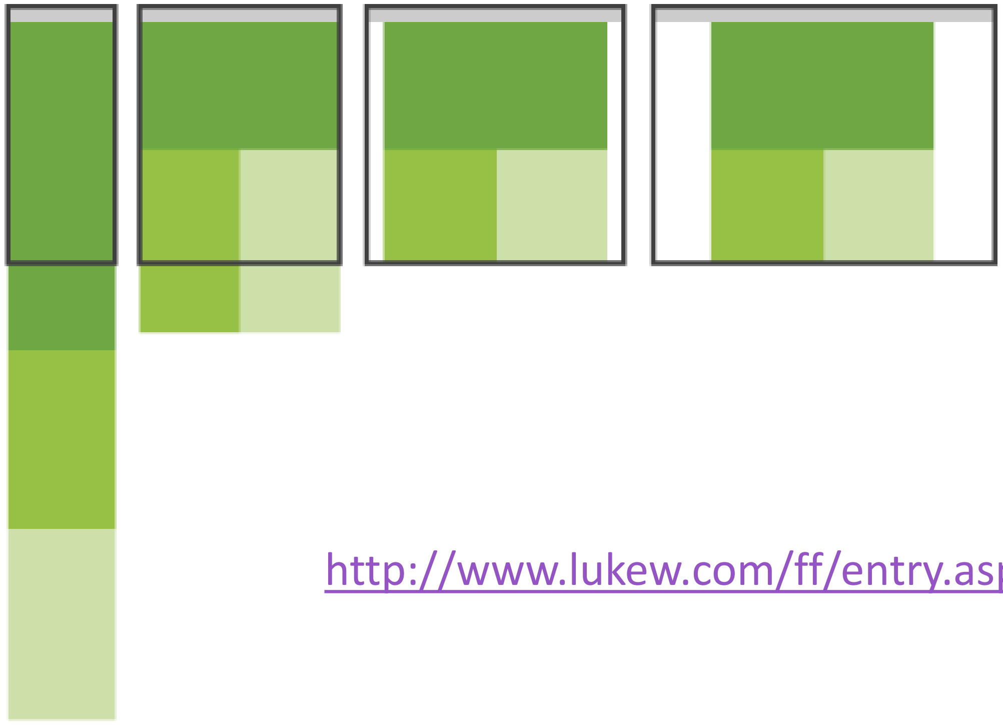
```
@media only screen and (-webkit-min-device-pixel-ratio: 2) {
  /* CSS goes here */
}
```

Media Queries Magic Numbers

- You don’t need to guess the magic numbers in media queries – search the Web, and you will find working media queries, E.g. :
  - http://css-tricks.com/snippets/css/retina-display-media-query/
  - http://blog.safaribooksonline.com/2012/02/24/responsive-mobile-design-media-queries-css3-ios-2/
- Or Google “Responsive web design”
- Or use Bootstrap

Responsive Layout: Mostly Fluid

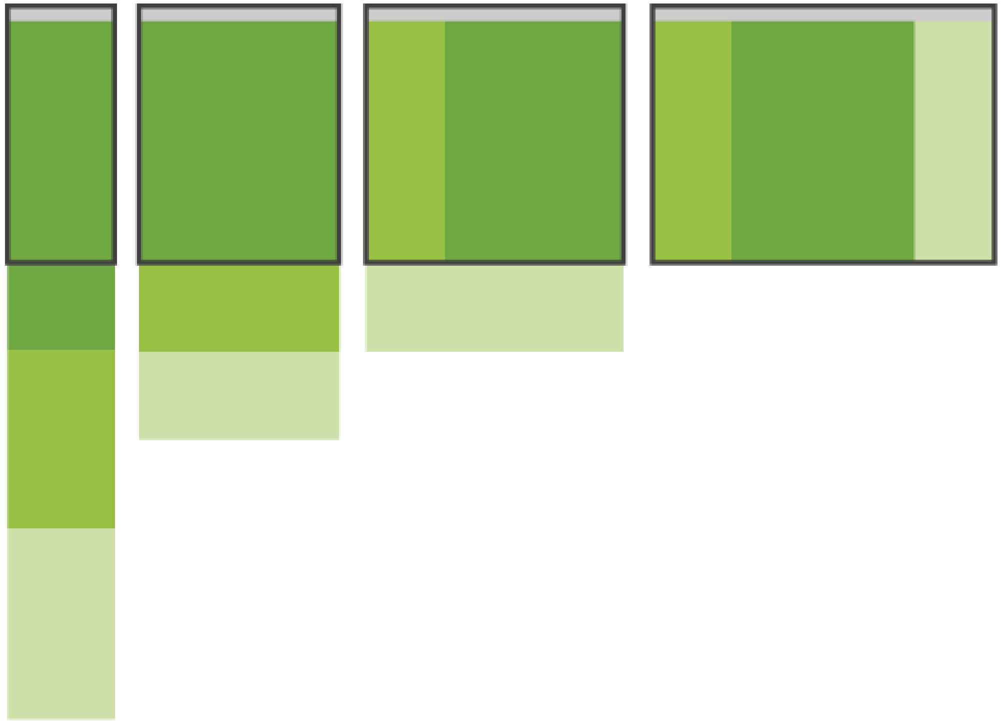
- A multi-column layout that introduces larger margins on big screens, relies on fluid grids and images to scale from large screens down to small screen sizes, and stacks columns vertically in its narrowest incarnations



http://www.lukew.com/ff/entry.asp?1514

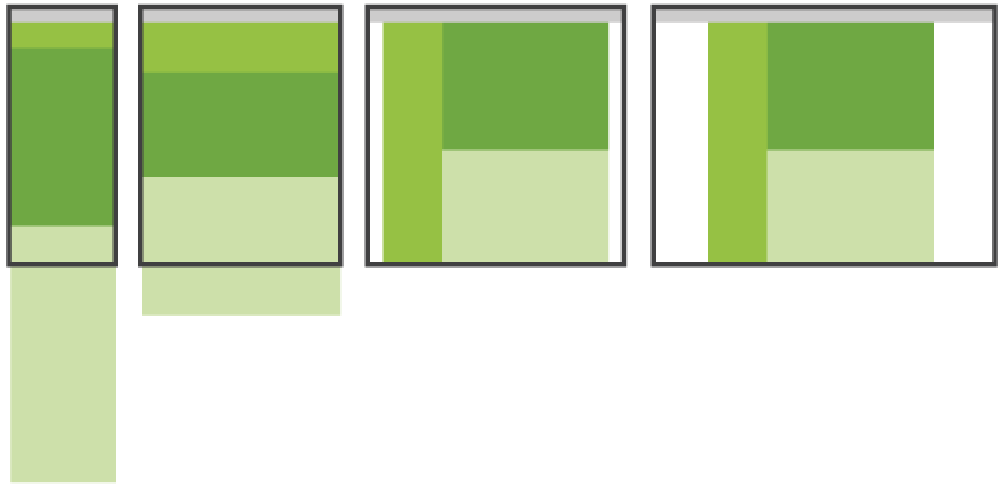
Responsive Layout: Column Drop

- Starts with a multi-column layout and ends up with a single column layout, dropping columns along the way as screen sizes get narrower. Unlike the Mostly Fluid pattern, the overall size of elements in this layout tend to stay consistent. Adapting to various screen sizes instead relies on stacking columns



Responsive Layout: Layout Shifter

- This pattern does the most to adapt across different screen sizes. That is, different layouts are used on large, medium, and small screens. Because this inherently requires more work, it seems to be less popular than the previous two patterns



Responsive - Bootstrap

- You don’t have to reinvent the wheel – use twitter bootstrap
- **http://twitter.github.io/bootstrap/** or **http://blog.getbootstrap.com/**
- Bootstrap was made to not only look and behave great in the latest desktop browsers, but also in tablet and smartphone browsers via responsive CSS
  - Free HTML snippets for Bootstrap: **http://bootsnipp.com/**
- Other responsive templates:
  http://html5boilerplate.com/
- OR use **flexbox**
  https://css-tricks.com/snippets/css/a-guide-to-flexbox/

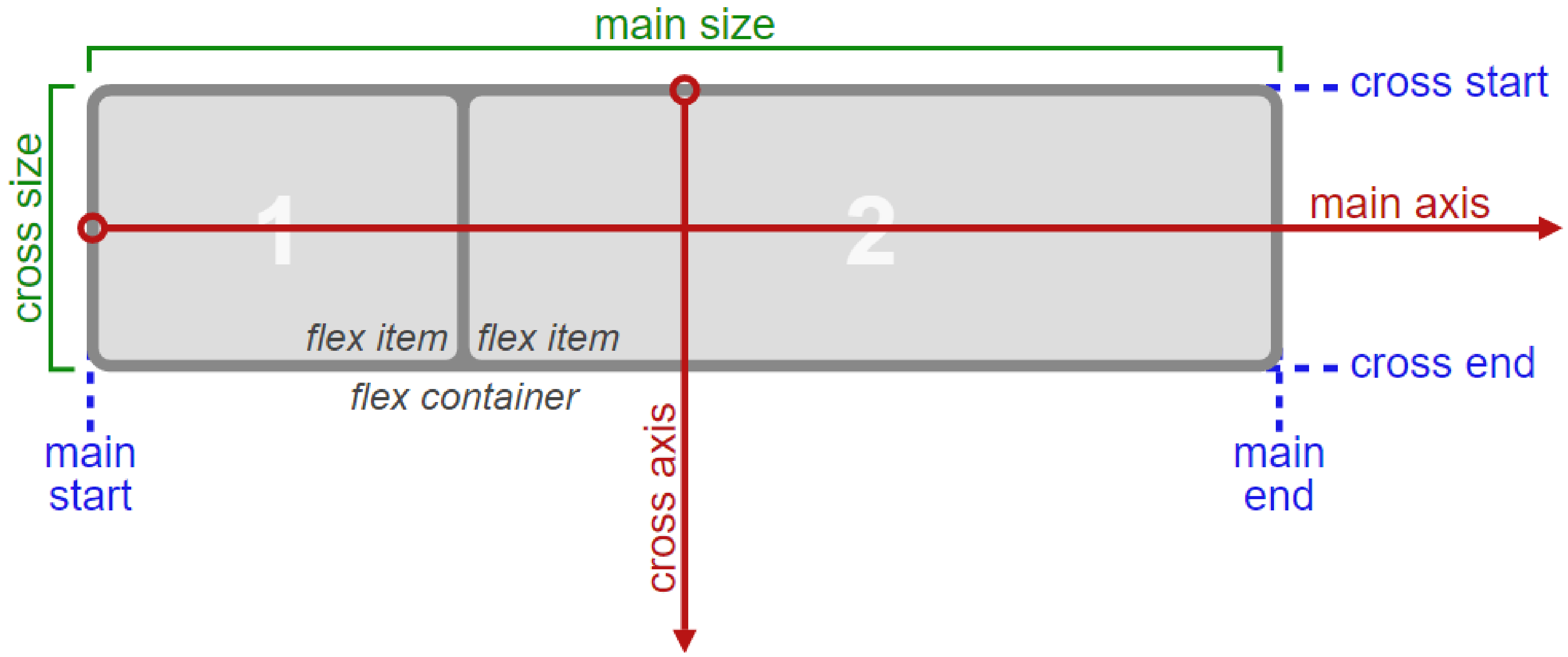
FLEXBOX

Flexbox

- Is a new layout mode in CSS3
- Is a flexible, float-free CSS layout method that accommodates different screen sizes and display devices
- Allows the browser to alter the width or height of elements to best fill the available space on any display device
  - Expand elements to fill all available free space
  - Shrink elements to prevent overflow
- With flexbox the designer can lay out elements vertically or horizontally, taking up all the space provided or the least amount of space necessary
  - Or create elements of equal height or width with just a few lines of CSS

Flex Layout Terminology

- A **flex container** is the box generated by an element with a display property of flex or inline-flex
- Children of a flex container are called **flex items** and are laid out using the flex layout model



Flex Container

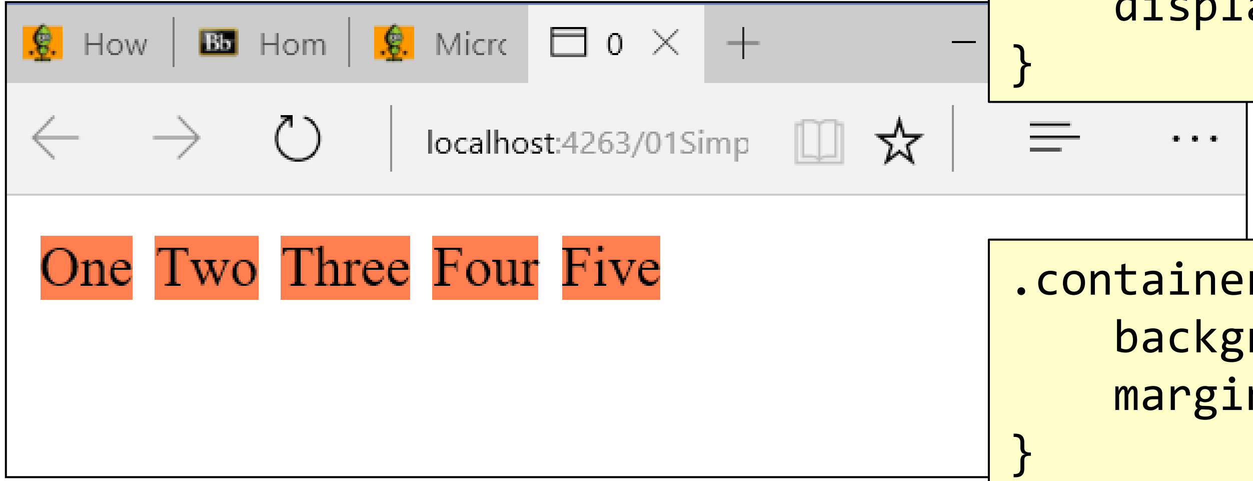
- You define a **containing block** as the container of flexible item
  - either an inline or a block-level flex container

```
<div class="container">
    <p >One</p>
    <p>Two</p>
    <p>Three</p>
    <p>Four</p>
    <p>Five</p>
</div>
```

```
.container {
    display: inline-flex;
}
```

Or

```
.container {
    display: flex;
}
```

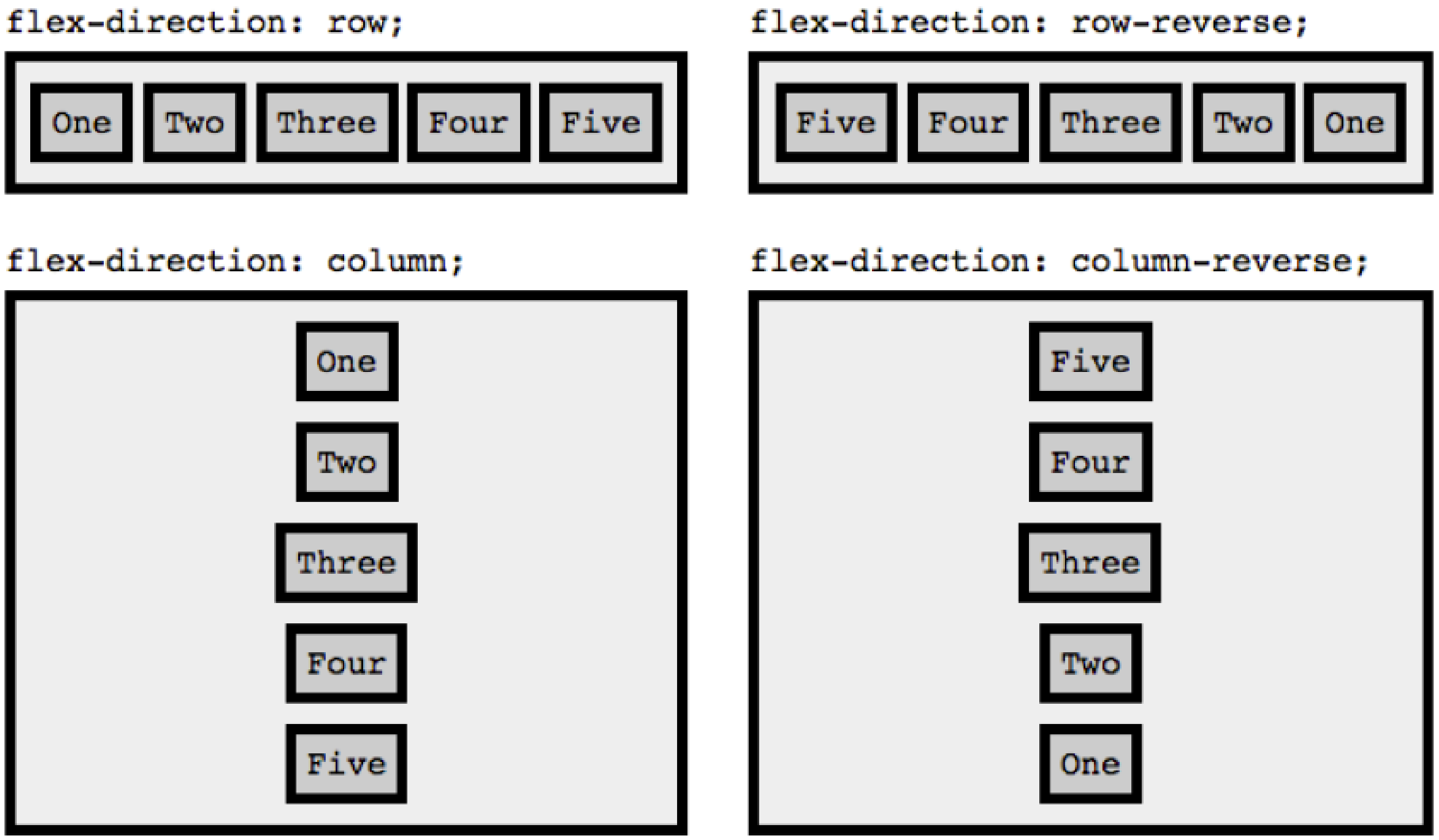


One Two Three Four Five

```
.container > p {
    background-color: coral;
    margin : 0.2em;
}
```

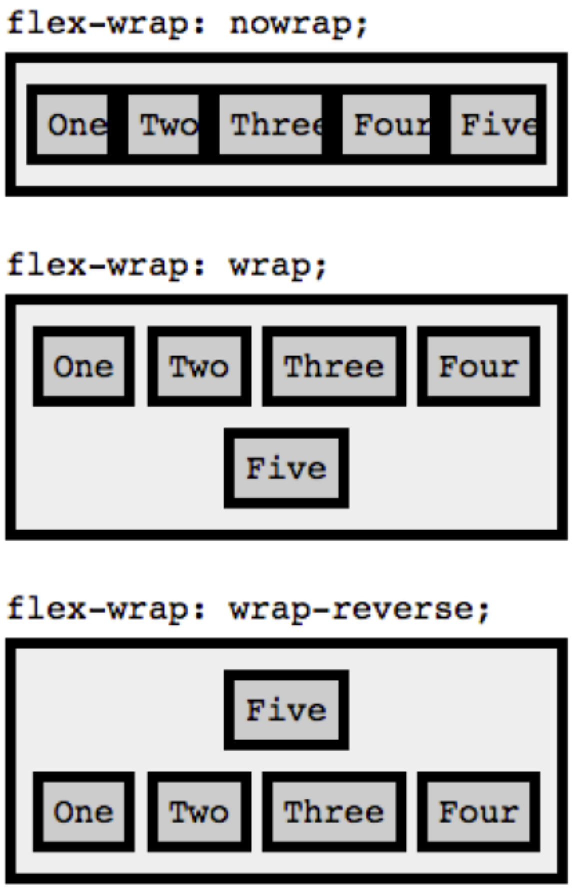
flex-direction

- Defines the axis along which the flex items follow each other
  - `flex-direction: row` *is default*



flex-wrap

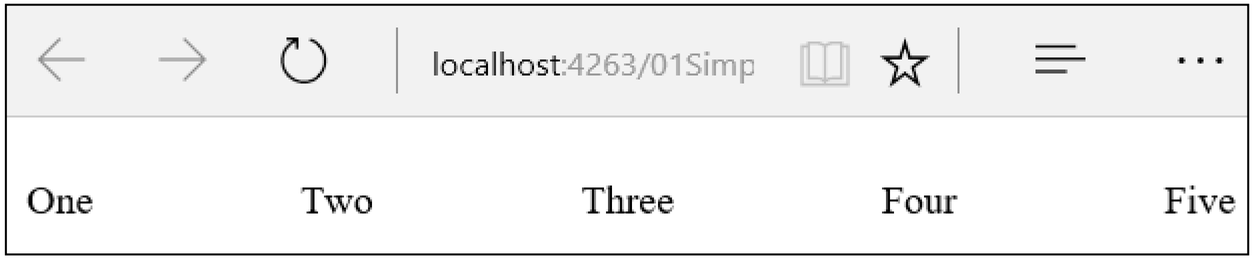
- Controls whether the flex container is single-line, multiline, or multi-lined with each new line coming visually before the previous line
- Possible values: `nowrap`, `wrap`, and `wrap-reverse`



AARHUS UNIVERSITY
AARHUS UNIVERSITY SCHOOL OF ENGINEERING

justify-content

- Defines how flex items are laid out on the current line
  - `flex-start` (default)
    - groups items to the left of the line (or the top)
  - `flex-end`
    - groups the items to the right of the line (or bottom)
  - `center`
    - groups the items in the center of the line
  - `space-between`
    - will push the first item to the left or top, and the last item to the right or bottom
  - `space-around`
    - value divides the white space equally between all the items, including around the first and last items



align-items

- Defines how flex items are laid out along the opposite axis as set by flex-direction (the cross axis)
  - flex-start
    - Items are placed at the top of the container
  - `flex-end`
    - Items are placed at the bottom
  - `center`
    - Will center the items vertically
  - `Stretch` (default)
    - Will stretch the items so that they’re all equal height
    - Taking up 100% of the height when there’s a single row
  - `Baseline`
    - Similar to flex-start but items are aligned along their baselines

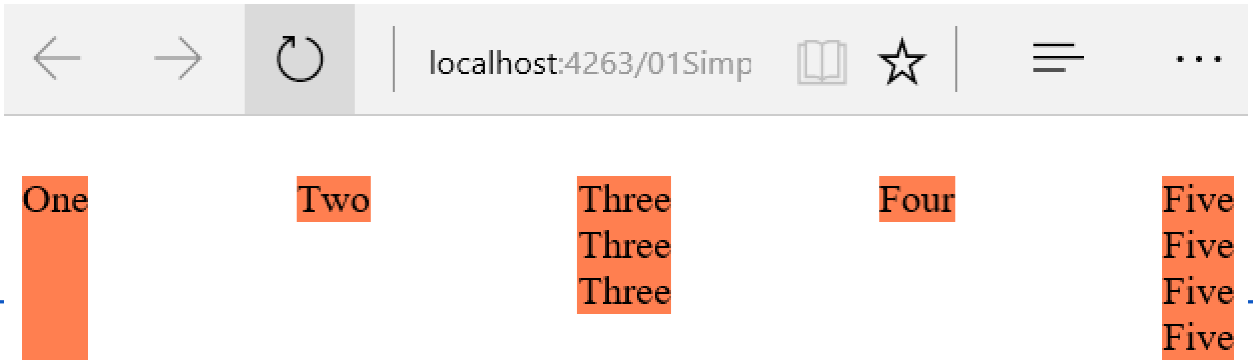
alignself

- You can override the align-items property for individual flex items by setting the alignself property on a single flex item

```
.container2 {
            display: flex;
            -ms-flex-flow: row nowrap;
            -webkit-flex-flow: row nowrap;
            flex-flow: row nowrap;
            justify-content: space-between;
            align-items:  flex-start;
        }

        .flexItem {
            background-color: coral;
        }

            .flexItem:first-of-type {
                align-self: stretch;
            }
```



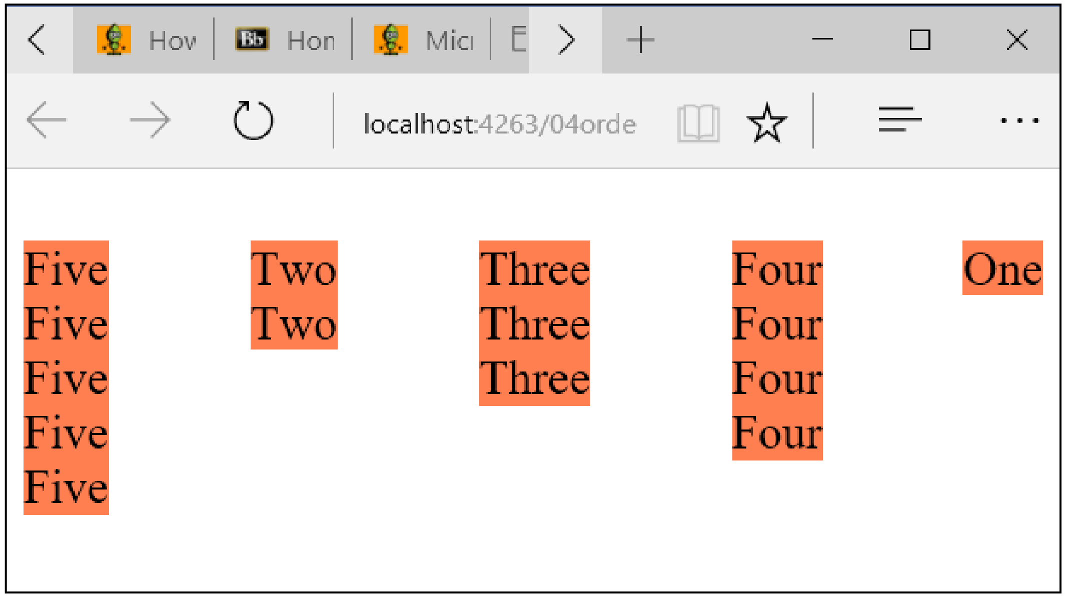
AARHUS UNIVERSITY
AARHUS UNIVERSITY SCHOOL OF ENGINEERING

order

- You can give each (or some) flex item an `order` property with an integer value
- The browser will display the items in ascending order according to the value of the order property instead of the html source order
- The default value of `order` is 0
- If 2 or more items have the same order value then the browser will use the html source order

```
.flexItem:first-of-type {
    order: 1;
}

.flexItem:last-of-type {
    order: -1;
}
```



Five Five Five Five Five

Two Two

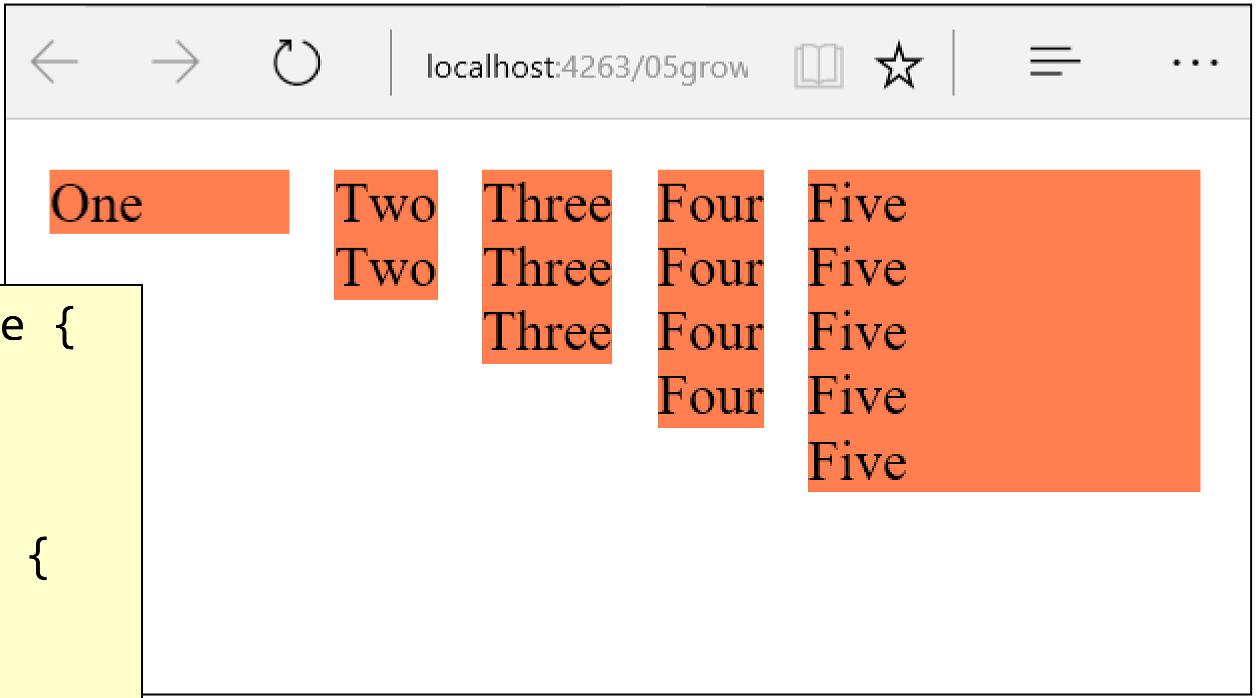
Three Three Three

Four Four Four Four

One

flex-grow

- Defines how the flex item should grow if there is room
- Is a unitless value for each flex-item that defines the proportion that the particular flex-item should occupy



```
.flexItem:first-of-type {
    flex-grow: 2;
}

.flexItem:last-of-type {
    flex-grow: 4;
}
```

flex-basis

- Defines the default width of each flex item
- Possible values:
  - auto (default)
  - A length value like 200px, 10em or 30%

flex-shrink

- Defines the ability for a flex item to shrink if necessary
- Possible values: an integer without units
- If there isn’t enough room for all the flexitems, the flex items with the largest flex-shrink value will shrink first

flex

- Is the shorthand for flex-grow, flex-shrink and flex-basis combined
- The second and third parameters (flex-shrink and flex-basis) are optional
- Default is 0 1 auto

CSS STYLING FOR PRINT

CSS Styling for Print

- Create an external style sheet with the configurations for browser display
- Create a second external style sheet with the configurations for printing
- Connect both of the external style sheets to the web page using two <**link** > elements.

```
<link rel="stylesheet" href="wildflower.css" type="text/css" media="screen">
<link rel="stylesheet" href="wildflowerprint.css" type="text/css" media="print">
```

Print Styling Best Practices

- **Hide non-essential content**
  Example:
  ```
  #nav { display: none; }
  ```
- **Configure font size and color for printing**
  - Use pt font sizes, use dark text color
- **Control page breaks**
  Example:
  ```
  .newpage { page-break-before: always; }
  ```
- **Print URLs for hyperlinks**
  Example:
  ```
  #sidebar a:after { content: " (" attr(href) ") "; }
  ```

For more info on Print

- Printing The Web
  http://drublic.de/blog/printing-the-web/

References & Links

- "Web Development and Design Foundations with HTML5"
- "HTML5 & CSS3 for the Real World"
- Logical Breakpoints For Your Responsive Design (use em) http://www.smashingmagazine.com/2013/03/01/logical-breakpoints-responsive-design/
- Responsive design is about more than just layout http://www.smashingmagazine.com/2013/05/06/new-defaults-web-design/
- Building Smartphone-Optimized Websites (Google optimized) https://developers.google.com/webmasters/smartphone-sites/details
- Make the mobile web faster https://developers.google.com/speed/articles/mobile
- **HTML5 compatibility on mobile and tablet browsers** http://mobilehtml5.org/
- **The Mobile Checker http://validator.w3.org/mobile-alpha/**