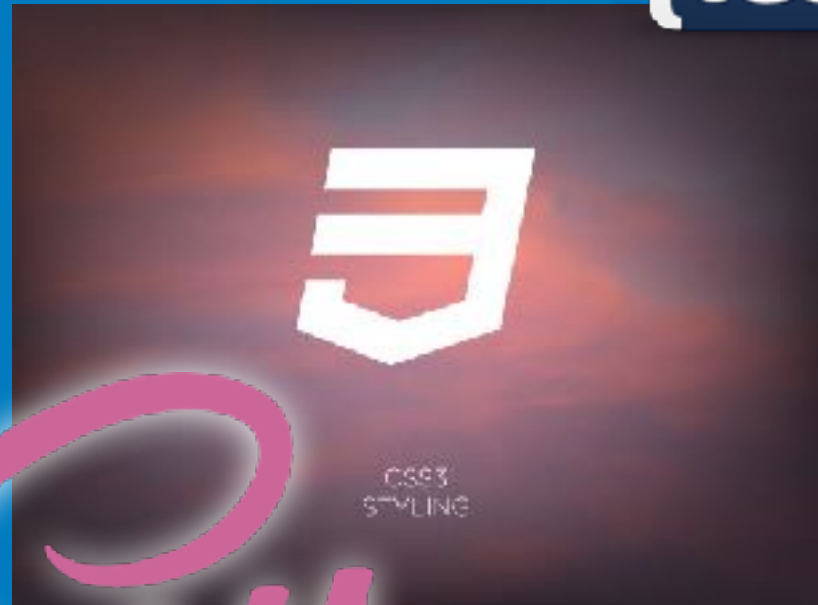


Building layouts with frameworks and preprocessors



Sass

Agenda

■ Building layouts

- Fixed layouts
- Fluid layouts
- Responsive layouts



- Getting started
- Mixins
- Nested rules
- Operations
- More LESS

■ LESS vs Sass

■ CSS frameworks

- 960 grid system
- Bootstrap, from Twitter



- Getting started
- Mixins and extending
- Nested rules
- Operations
- Control directives
- More Sass

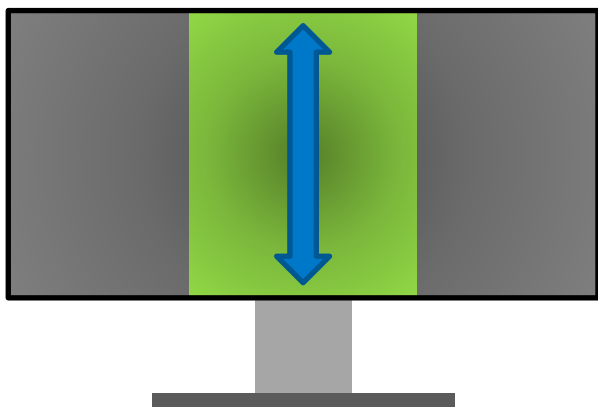
Layouts: The fixed (1/2)

■ Content area has a fixed width, e.g. 760 pixels

- Pros
- Tight control over the readable content area
 - Often easy to style

Cons

- The content area may appear narrow on wide screens
- Limits the reasonable amount of displayable content



Layouts: The fixed (2/2)

- A fixed width, often positioned in the middle of the page

```
#container {  
    width: 700px;  
    margin-left: auto;  
    margin-right: auto;  
}
```

CSS

```
#container {  
    width: 700px;  
    position: absolute;  
    left: 50%;  
    margin-left: -350px;  
}
```

CSS

Agenda

■ Building layouts

- Fixed layouts
- Fluid layouts
- Responsive layouts



- Getting started
- Mixins
- Nested rules
- Operations
- More LESS

■ LESS vs Sass

■ CSS frameworks

- 960 grid system
- Bootstrap, from Twitter



- Getting started
- Mixins and extending
- Nested rules
- Operations
- Control directives
- More Sass

Layouts: The fluid

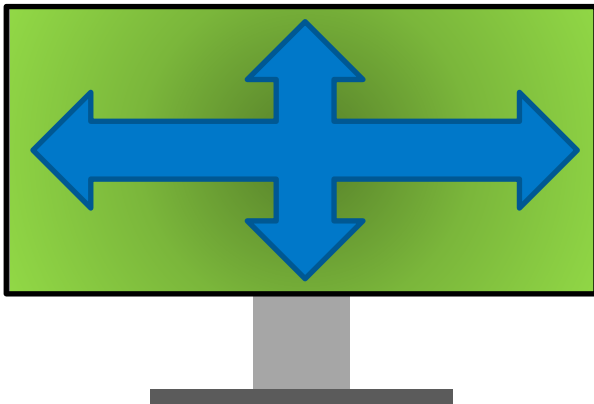
■ Content area has a relative width, e.g.

Pros 76%

- Makes good use of the available screen estate

Cons

- Readability suffers if text spans the entire screen
- More work to implement successfully
- Modern browsers support page zoom instead of text zoom



Agenda

■ Building layouts

- Fixed layouts
- Fluid layouts
- Responsive layouts



- Getting started
- Mixins
- Nested rules
- Operations
- More LESS

■ LESS vs Sass

■ CSS frameworks

- 960 grid system
- Bootstrap, from Twitter



- Getting started
- Mixins and extending
- Nested rules
- Operations
- Control directives
- More Sass

Layouts: The responsive (1/2)

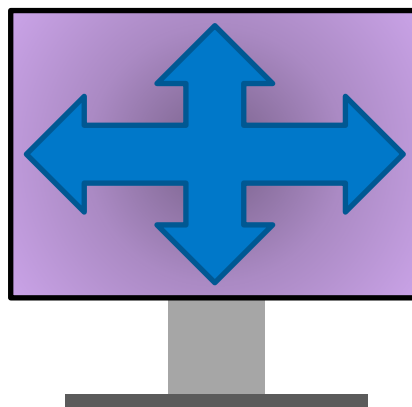
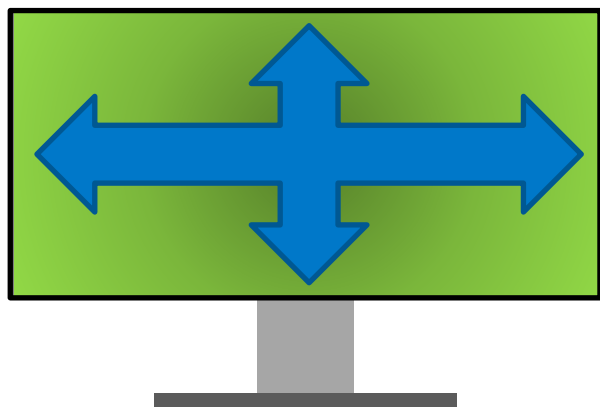
■ Respond with the layout that's best suited

Pros

- Offers an optimized user experience based on screen resolution

Cons

- More data to retrieve
- More work to implement



Layouts: The responsive (2/2)

- Respond to device capabilities with CSS3 media queries

- Use media queries to link external

```
<link href="mobile.css" rel="stylesheet"
      media="screen and (max-width: 400px)" />
```

HTML

```
@media screen and (orientation: portrait) {
    body {
        font-family: Verdana;
    }
}
```

CSS

Lab: Building layouts

- Exercise 1: Fully set up a layout
- Exercise 2: Make it responsive with a mobile layout
- Exercise 3: Build a form

A screenshot of a web browser displaying a mobile layout for a website called 'MyTrips'. The browser's address bar shows 'Near een website gaan' and the Google logo. The website has a dark background with three green buttons at the top: 'Home', 'My trips', and 'Tip a trip'. The 'Tip a trip' button is selected, and the page title is 'Tip a trip'. Below the title, there is a text input field with the placeholder 'Where?' and the label 'What's the location of the trip?'. Below that is another text input field with the placeholder 'When?' and the label 'When is it really nice there?'. Below that is a large text input field with the placeholder 'Why?' and the label 'Why should I go here?'. At the bottom right, there is a 'Tip it!' button. The footer of the page says 'Waldo © 2013'.

Agenda

■ Building layouts

- Fixed layouts
- Fluid layouts
- Responsive layouts



- Getting started
- Mixins
- Nested rules
- Operations
- More LESS

■ LESS vs Sass

■ CSS frameworks

- 960 grid system
- Bootstrap, from Twitter



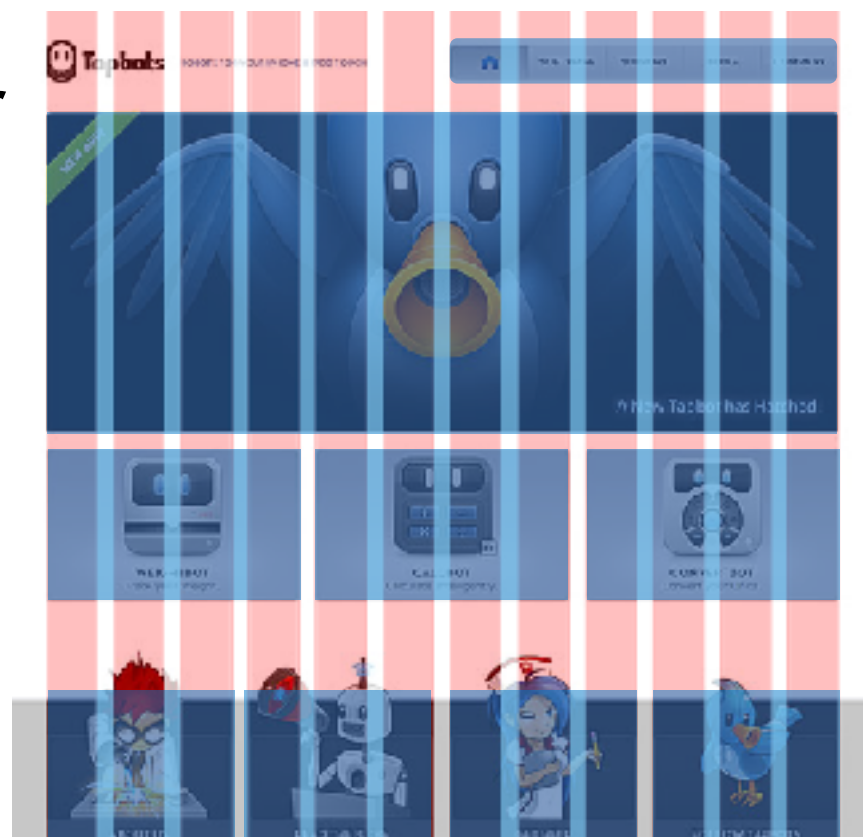
- Getting started
- Mixins and extending
- Nested rules
- Operations
- Control directives
- More Sass

960 grid system

- A CSS template to build your layout upon



- Divide 960 pixels over 12, 16 or 24 columns



960 grid system: Usage (1/3)

■ Declare the grid container

```
<div class="container_12">  
  ...  
</div>
```

Use 12
columns for
the grid

HTML

■ Use the grid

```
<div class="container_12">  
  <div class="grid_3">  
    ...  
  </div>  
</div>
```

Let this DIV span
across three columns

HTML

960 grid system: Usage (2/3)

■ Indent with prefixes and suffixes

HTML

```
<div class="container_12">
```

```
  <div class="grid_3 prefix_2 suffix_4">
```

```
    ...
```

```
  </div>
```

```
</div>
```

Provides two columns
of indenting **before**
the DIV

Provides four columns
of indenting **after** the
DIV

- Uses padding-left and padding-right

960 grid system: Usage (3/3)

■ Push and pull space

- Rearrange elements independent of the order in the markup

```
<div class="container_12">  
  <div class="grid_3 pull_2">  
    ...  
  </div>  
</div>
```

HTML

“Pulls” the element
two columns to the left

- Uses relative positioning along with `left` and `right`

Agenda

■ Building layouts

- Fixed layouts
- Fluid layouts
- Responsive layouts



- Getting started
- Mixins
- Nested rules
- Operations
- More LESS

■ LESS vs Sass

■ CSS frameworks

- 960 grid system
- Bootstrap, from Twitter



- Getting started
- Mixins and extending
- Nested rules
- Operations
- Control directives
- More Sass

Bootstrap, from Twitter

- A framework to speed up front-end development
- Features:
 - **Positioning** with a 12-column responsive grid
 - **Styling** of tables, fonts, buttons and forms
 - **UI components**, including dropdown menus, progress indicators, alerts and pagination
 - **jQuery plugins**, including modal dialogs, tabs, tooltips, button states and scrollspy



Bootstrap: Usage

■ Include Bootstrap CSS for styling

```
<link href="bootstrap/css/bootstrap.css"
      rel="stylesheet" />
```

HTML

- Bootstrap uses CSS3, so make sure the HTML5 doctype is set

■ Include Bootstrap JavaScript to make use of components

- Include jQuery as well!

```
<script src="scripts/jquery.js"></script>
<script src="bootstrap/js/bootstrap.js"
      type="text/javascript"></script>
```

HTML

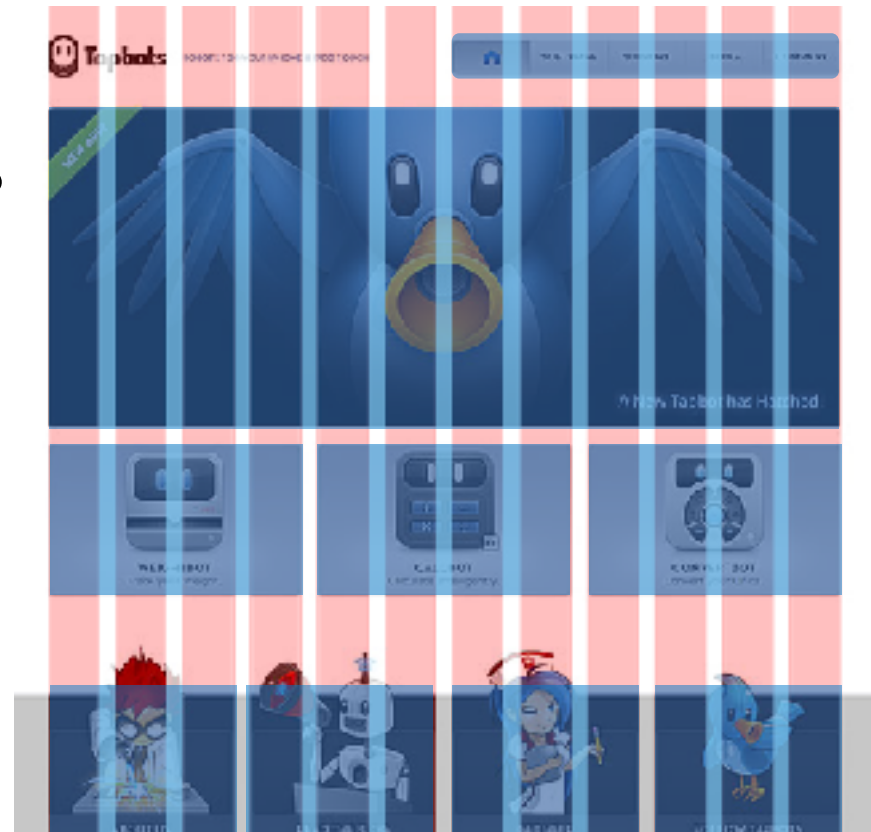
Bootstrap, from Twitter

- A framework to speed up front-end development
- Features:
 - **Positioning** with a 12-column responsive grid
 - **Styling** of tables, fonts, buttons and forms
 - **UI components**, including dropdown menus, progress indicators, alerts and pagination
 - **jQuery plugins**, including modal dialogs, tabs, tooltips, button states and scrollspy



B Positioning

- A grid that:
 - Stretches over 940 pixels
 - Has 12 columns
 - Has capabilities for fixed and fluid layouts
- Support for building responsive layouts



B Positioning: Grid (1/3)

■ Provide a fixed-width layout

```
<div class="container">
  ...
</div>
```

HTML

■ And use the grid

```
<div class="row">
  <div class="span4">
    ...
  </div>
</div>
```

Let this DIV span
across four columns

HTML

B Positioning: Grid (2/3)

■ Provide a fluid layout

```
<div class="container-fluid">
  ...
</div>
```

HTML

■ And use the grid

```
<div class="row-fluid">
  <div class="span4">
    ...
  </div>
</div>
```

HTML

← Let this DIV span
across four columns

B Positioning: Grid (3/3)

■ Indent with offsets

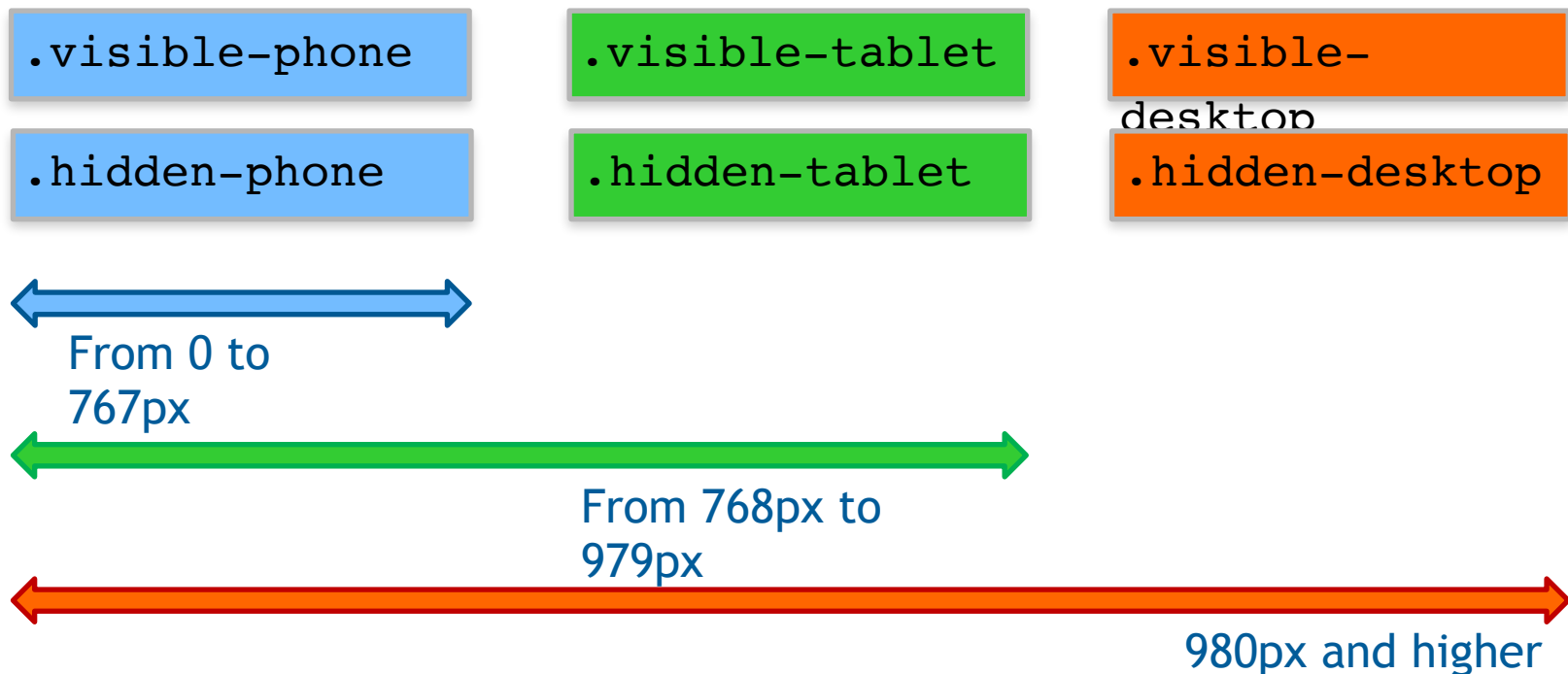
HTML

```
<div class="row">  
  <div class="span4 offset2">  
    ...  
  </div>  
</div>
```

Provides two columns
of indenting before
the DIV

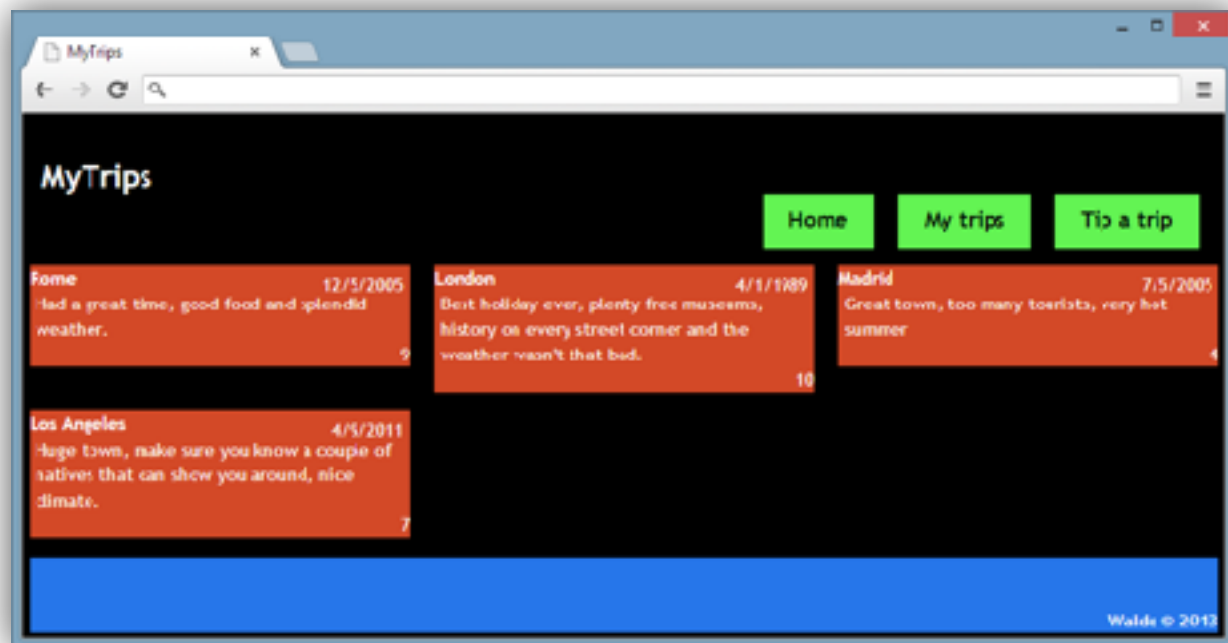
– Uses `margin-left`

- Bootstrap offers **utility classes** for showing/hiding content **by device width**



Lab: Bootstrap

■ Exercise 4: Build a layout with Bootstrap



Bootstrap, from Twitter

- A framework to speed up front-end development

- Features:

- **Positioning** with a 12-column responsive grid
- **Styling** of tables, fonts, buttons and forms
- **UI components**, including dropdown menus, progress indicators, alerts and pagination
- **jQuery plugins**, including modal dialogs, tabs, tooltips, button states and scrollspy



■ Common elements

- ``, `<i>`, `<abbr>`, `<blockquote>`, ``, ``,
`<dl>`, `<code>` and `<pre>`

■ Typical usage scenarios

- Zebra-striped tables
- Displaying form input fields
- Buttons and images
- Emphasis classes

B Styling: Tables (1 / 6)

■ Use table styling

```
<table class="table">
  ...
</table>
```

HTML

#	First name	Last name	Date of birth
1	Frank	Young	July 5th, 1981
2	Susan	Chi	November 22th, 1966
3	Paul	Wineless	March 1st, 1991

B Styling: Tables (2/6)

■ Add more explicit borders

HTML

```
<table class="table table-bordered">
  ...
</table>
```

#	First name	Last name	Date of birth
1	Frank	Young	July 5th, 1981
2	Susan	Chi	November 22th, 1966
3	Paul	Wineless	March 1st, 1991

B Styling: Tables (3/6)

■ Add zebra stripes to a table

```
<table class="table table-striped">  
  ...  
</table>
```

HTML

#	First name	Last name	Date of birth
1	Frank	Young	July 5th, 1981
2	Susan	Chi	November 22th, 1966
3	Paul	Wineless	March 1st, 1991

- Uses CSS3's `nth-child()` and does not work in older browsers

B Styling: Tables (4/6)

■ Make tables more compact

HTML

```
<table class="table table-condensed">
  ...
</table>
```

#	First name	Last name	Date of birth
1	Frank	Young	July 5th, 1981
2	Susan	Chi	November 22th, 1966
3	Paul	Wineless	March 1st, 1991

B Styling: Tables (5/6)

■ Enable a hover state

```
<table class="table table-hover">
  ...
</table>
```

HTML

#	First name	Last name	Date of birth
1	Frank	Young	July 5th, 1981
2	Susan	Chi	November 22th, 1966
3	Paul	Wineless	March 1st, 1991

B Styling: Tables (6/6)

■ Apply contextual classes

.success

.error

.warning

.info

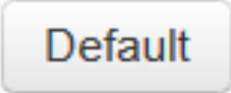
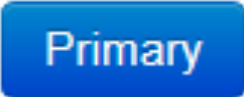




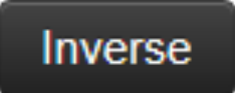

```
<table class="table">
  <tr class="success">
    ...
  </tr>
</table>
```

HTML

#	First name	Last name	Date of birth
1	Frank	Young	July 5th, 1981
2	Susan	Chi	November 22th, 1966
3	Paul	Wineless	March 1st, 1991

B Styling: Buttons (1/3)

■ Bootstraps offers buttons that aid the user

Button	CSS class(es)	Meaning
	<code>btn</code>	Default button
	<code>btn btn-primary</code>	Identifies primary action in a set of buttons
	<code>btn btn-info</code>	Alternative to the default styles
	<code>btn btn-success</code>	Indicates successful or positive action
	<code>btn btn-warning</code>	Indicates caution should be taken
	<code>btn btn-danger</code>	Indicates a dangerous or negative action
	<code>btn-inverse</code>	Alternate dark gray button, no specific meaning
	<code>btn btn-link</code>	Deemphasize a button

B Styling: Buttons (2/3)

- Use bigger or smaller buttons to attract more or less attention

```
<button class="btn btn-large">Large</button>  
<button class="btn">Default</button>  
<button class="btn btn-small">Small</button>  
<button class="btn btn-mini">Mini</button>
```

HTML



- Block-level buttons

```
<button class="btn btn-large btn-block">100% width</button>
```

HTML

100% width

B Styling: Buttons (3/3)

■ Apply the button look to the element you

```
<a class="btn" href="#">Link</a>  
<button class="btn" type="submit">Button</button>  
<input class="btn" type="button" value="Input">  
<input class="btn" type="submit" value="Submit">
```

HTML

Link

Button

Input

Submit

- Bootstrap styles form elements
 - Input fields get rounded corners, padding and a glow on focus
 - Contextual classes for validation
 - Building the form gets easier with helpful additional classes

■ Display a form inline

HTML

```
<form class="form-inline">
  <input type="text" class="input-small"
    placeholder="Email">
  <input type="password" class="input-small"
    placeholder="Password">
  <label class="checkbox">
    <input type="checkbox">
    Remember me
  </label>
  <button type="submit" class="btn">Sign in</button>
</form>
```

B Styling: Forms (3/8)

■ Display elements on the same line

HTML

```
<form class="form-horizontal">
  <div class="control-group">
    <label class="control-label"
      for="inputEmail">E-mail</label>
    <div class="controls">
      <input type="text" id="inputEmail"
        placeholder="E-mail">
    </div>
  </div>
  [...more control groups with labels and controls...]
</form>
```

E-mail

Password

Sign in

B Styling: Forms (4/8)

■ Use contextual classes for validation

.success

.error

.warning

.info

```
<form class="form-horizontal">  
  <div class="control-group error">  
    [...labels and controls...]  
  </div>  
  [...more control groups with labels and controls...]  
</form>
```

HTML

E-mail

E-mail

Password

Password

Sign in

B Styling: Forms (5/8)

■ Default support for search forms

```
<form class="form-search">  
  <input type="text" class="input-medium search-  
query">  
  <button type="submit" class="btn">Search</button>  
</form>
```

HTML

B Styling: Forms (6/8)

■ Prepend an input field

```
<div class="input-prepend">  
  <span class="add-on">@</span>  
  <input type="text" placeholder="Twitter name">  
</div>
```

HTML

■ Append an input field

```
<div class="input-prepend input-append">  
  <span class="add-on">&euro;</span>  
  <input class="span1" type="text">  
  <span class="add-on">,.00</span>  
  <button class="btn">Submit</button>  
</div>
```

HTML

Buttons are
automatically
appended

€	500	,00	Submit
---	-----	-----	--------

B Styling: Forms (7/8)

■ Size input fields

HTML

```
<input class="input-mini" type="text"
      placeholder=".input-mini">
<input class="input-medium" type="text"
      placeholder=".input-medium">
<input class="input-xxlarge" type="text"
      placeholder=".input-xxlarge">
<input class="span3" type="text" placeholder=".span3">
```

.input-mini

.input-medium

.input-xxlarge

.span3

Size is set to three
grid columns

■ Emphasize form actions

HTML

```
<div class="form-actions">  
  <button type="submit"  
    class="btn btn-primary">Save changes</  
button>  
  <button class="btn">Cancel</button>  
</div>
```

- Buttons are automatically lined up with the form controls

E-mail address

Password

Lab: Bootstrap

■ Exercise 5: Set up a form with Bootstrap

A screenshot of a web browser displaying a web application titled 'MyTrips'. The browser's address bar shows 'MyTrips'. The application has a dark theme with a navigation bar at the top containing three green buttons: 'Home', 'My trips', and 'Tip a trip'. The 'Tip a trip' button is currently selected, and its label is highlighted in white. Below the navigation bar, the page title 'Tip a trip' is displayed in white. A red horizontal line separates the title from the form instructions. The instructions read: 'Use the form below to give a tip on what might possibly be a fantastic trip for me.' The form consists of three rows, each with a label on the left and a text input field on the right. The labels are 'Where?', 'When?', and 'Why?'. The corresponding input fields are empty. To the right of each input field, there is a small text label: 'What's the location of the trip?', 'When is it really nice there?', and 'Why should I go here?'. At the bottom of the form, there are two buttons: 'Save charges' (in blue) and 'Cancel' (in grey). The footer of the application is a solid blue bar with the text 'Waldo © 2013' in the bottom right corner.

Bootstrap, from Twitter

- A framework to speed up front-end development
- Features:
 - **Positioning** with a 12-column responsive grid
 - **Styling** of tables, fonts, buttons and forms
 - **UI components**, including dropdown menus, progress indicators, alerts and pagination
 - **jQuery plugins**, including modal dialogs, tabs, tooltips, button states and scrollspy



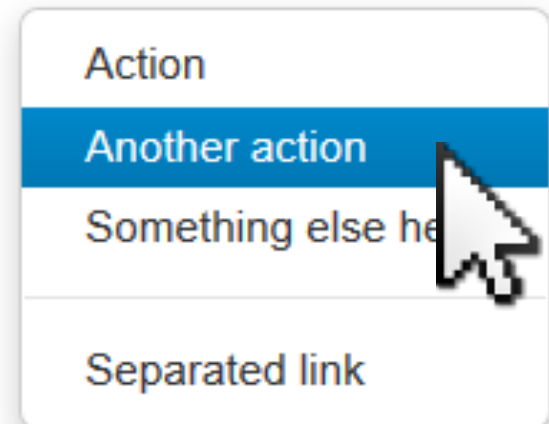
B UI components: Dropdowns

■ Support for dropdown menus

HTML

```
<ul class="dropdown-menu">
  <li><a href="#">Action</a></li>
  <li><a href="#">Another action</a></li>
  <li><a href="#">Something else here</a></li>
  <li class="divider"></li>
  <li><a href="#">Separated link</a></li>
</ul>
```

- Highly useful in combination with other UI components, e.g. buttons



B UI components: Alerts

■ Display notifications

HTML

```
<div class="alert">
  <button type="button" class="close"
    data-dismiss="alert">&times;</button>
  <strong>Warning!</strong> Something went wrong.
</div>
```

Warning! Something went horribly,
horribly wrong.



■ Support for context

```
<div class="alert alert-
info">
  ...
</div>
```

Warning! Something went horribly,
horribly wrong.



.alert-success

.alert-error

.alert-info

B UI components: Navbar (1/2)

■ Display a navigation bar

HTML

```
<div class="navbar">
  <div class="navbar-inner">
    <a class="brand" href="#">Title</a>
    <ul class="nav">
      <li class="active"><a href="#">Home</a></li>
      <li><a href="#">Link</a></li>
      <li><a href="#">Link</a></li>
    </ul>
  </div>
</div>
```

Title

Home

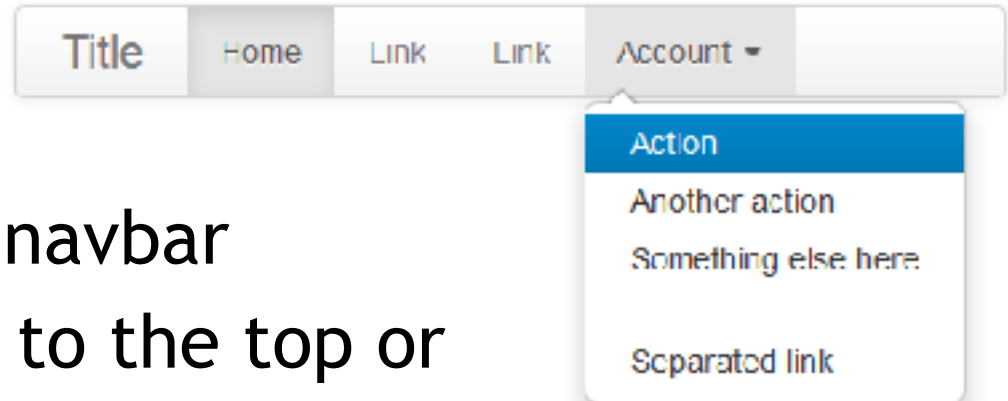
Link

Link

B UI components: Navbar (2/2)

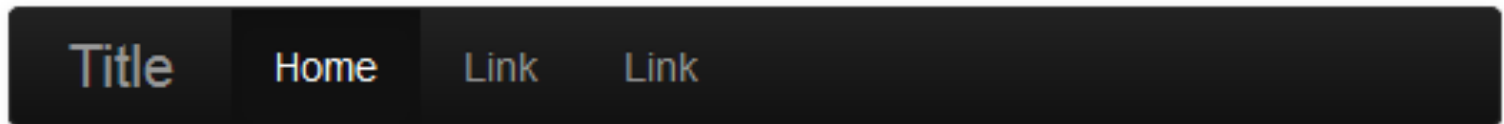
■ The navbar has built-in support for:

- Dropdowns for menu items
- Forms within the navbar
- Fixing the navbar to the top or bottom of the page
- An alternative “inverted” look



```
<div class="navbar navbar-inverse">
  ...
</div>
```

HTML

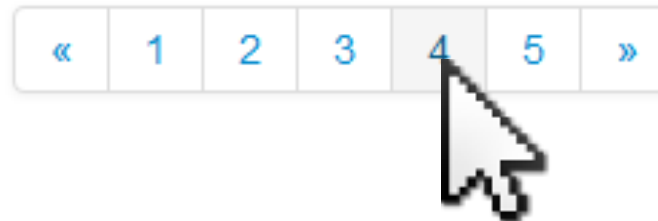


B UI components: Pagination

■ Display a pagination bar

HTML

```
<div class="pagination">
  <ul>
    <li><a href="#">&laquo;</a></li>
    <li><a href="#">1</a></li>
    <li><a href="#">2</a></li>
    <li><a href="#">3</a></li>
    <li><a href="#">4</a></li>
    <li><a href="#">5</a></li>
    <li><a href="#">&raquo;</a></li>
  </ul>
</div>
```



B UI components: Progress (1/3)

Show a progress indicator

```
<div class="progress">
  <div class="bar" style="width: 60%;"></div>
</div>
```

HTML

CSS



Indicate status along with progress

.progress-success

.progress-warning

.progress-danger

.progress-info

```
<div class="progress progress-danger">
  <div class="bar" style="width: 60%;"></div>
</div>
```

HTML

CSS



B UI components: Progress (2/3)

■ Stack progress bars

```
<div class="progress">
  <div class="bar bar-success"
    style="width: 35%;"></div>
  <div class="bar bar-warning"
    style="width: 20%;"></div>
  <div class="bar bar-danger"
    style="width: 10%;"></div>
</div>
```

HTML

CSS



B UI components: Progress (3/3)

■ Additional possibilities

– Striped progress bar

```
<div class="progress progress-striped">  
  <div class="bar" style="width: 60%;"></div>  
</div>
```

HTML

CSS



– Add animation to the striped progress bar

```
<div class="progress progress-striped active">  
  <div class="bar" style="width: 60%;"></div>  
</div>
```

HTML

CSS

Bootstrap, from Twitter

- A framework to speed up front-end development
- Features:
 - **Positioning** with a 12-column responsive grid
 - **Styling** of tables, fonts, buttons and forms
 - **UI components**, including dropdown menus, progress indicators, alerts and pagination
 - **jQuery plugins**, including modal dialogs, tabs, tooltips, button states and scrollspy



B jQuery plugins: Modal dialog

■ Display a modal dialog

```
<a href="#myModal" data-toggle="modal">Launch modal</a>
```

HTML

```
<a>
```

```
<div id="myModal" class="modal hide fade">
```

HTML

```
  <div class="modal-header">
```

```
    <button data-dismiss="modal"
            class="close">&times;</button>
```

```
    <h3>Some heading here</h3>
```

```
  </div>
```

```
  <div class="modal-body">
```

```
    <p>Lorem ipsum dolor sit amet, ...</p>
```

```
  </div>
```

```
  <div class="modal-footer">
```

```
    <button class="btn" data-dismiss="modal"
            aria-hidden="true">Close</button>
```

```
    <button class="btn btn-primary">Save</button>
```

```
  </div>
```

```
</div>
```


B jQuery plugins: Tabs

■ Divide content over tabs

HTML

```
<ul class="nav nav-tabs">
  <li><a href="#home" data-toggle="tab">General</a></li>
  <li><a href="#videos" data-toggle="tab">Videos</a></li>
  <li><a href="#extras" data-toggle="tab">Extras</a></li>
</ul>
<div class="tab-content">
  <div class="tab-pane active" id="home">General</div>
  <div class="tab-pane" id="videos">Some videos</div>
  <div class="tab-pane"
div>
</div>
```

General

Videos

Extra info

Some videos

B jQuery plugins: Tooltips

■ Show a tooltip with helpful information

```
<a href="#" rel="tooltip" title="Cool tip">hover  
here</a>
```

HTML

- Due to performance reasons, tooltips must be explicitly activated

```
$( "a[rel=tooltip]" ).tooltip();
```

JS



■ Control placement

```
<a href="#" rel="tooltip" title="Cool tip"  
data-placement="bottom">hover here</a>
```

HTML



B jQuery plugins: Popovers

■ Display an extended tooltip

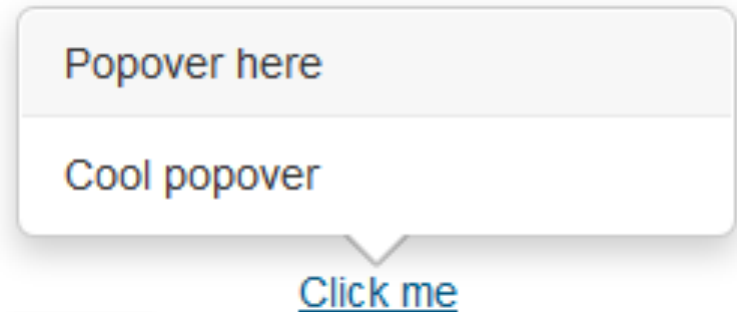
```
<a href="#" rel="popover" data-content="Cool popover" data-title="Popover here">Click me</a>
```

HTML

- Due to performance reasons, popovers must be explicitly activated

```
$( "a[rel=popover]" ).popover( )
```

JS



■ Control placement

```
<a href="#" rel="popover" data-placement="right" data-content="Cool popover" data-title="Popover here!">Click me</a>
```

HTML

B jQuery plugins: Buttons (1 / 2)

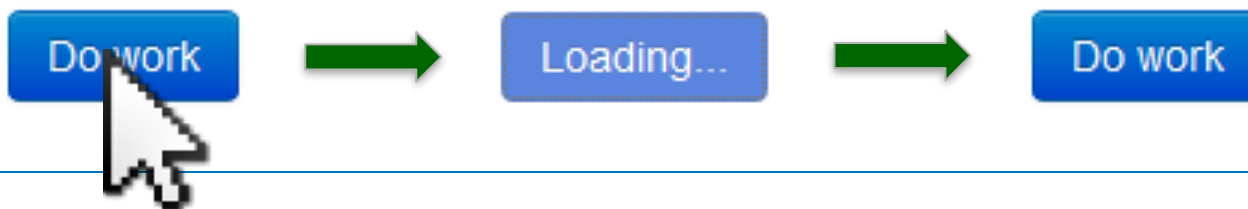
- Do more with buttons by supplying feedback

```
<button id="btnDoWork" data-loading-text="Loading...">Do work</button>
```

HTML

```
$( '#btnDoWork' ).click(function () {  
    $(this).button('loading');  
  
    // Do amazing work  
  
    $(this).button('reset');  
});
```

JS



B jQuery plugins: Buttons (2/2)

■ Make buttons behave like radiobuttons

```
<div data-toggle="buttons-radio">
  <button class="btn btn-info">Left</button>
  <button class="btn btn-info">Middle</button>
  <button class="btn btn-info">Right</button>
</div>
```

HTML

Left

Middle

Right

■ Or like checkboxes

```
<div class="btn-group" data-toggle="buttons-
checkboxes">
  <button class="btn btn-danger">Left</button>
  <button class="btn btn-danger">Middle</button>
  <button class="btn btn-danger">Right</button>
</div>
```

HTML

Left

Middle

Right

B jQuery plugins: Typeahead

■ Support for autocompletion

HTML

```
<input type="text" data-provide="typeahead" data-items="5" data-source=['Alabama", "Alaska", "Arizona", [...], "Washington", "West Virginia", "Wisconsin", "Wyoming"]'>
```

■ Autocompletion also possible with JavaScript

```
$( '.typeahead' ).typeahead({  
  source: function (typeahead, qry) {  
    return $.get('/ta', { query: qry }, function  
(data) {  
      return typeahead.process(data);  
    });  
  }  
});
```

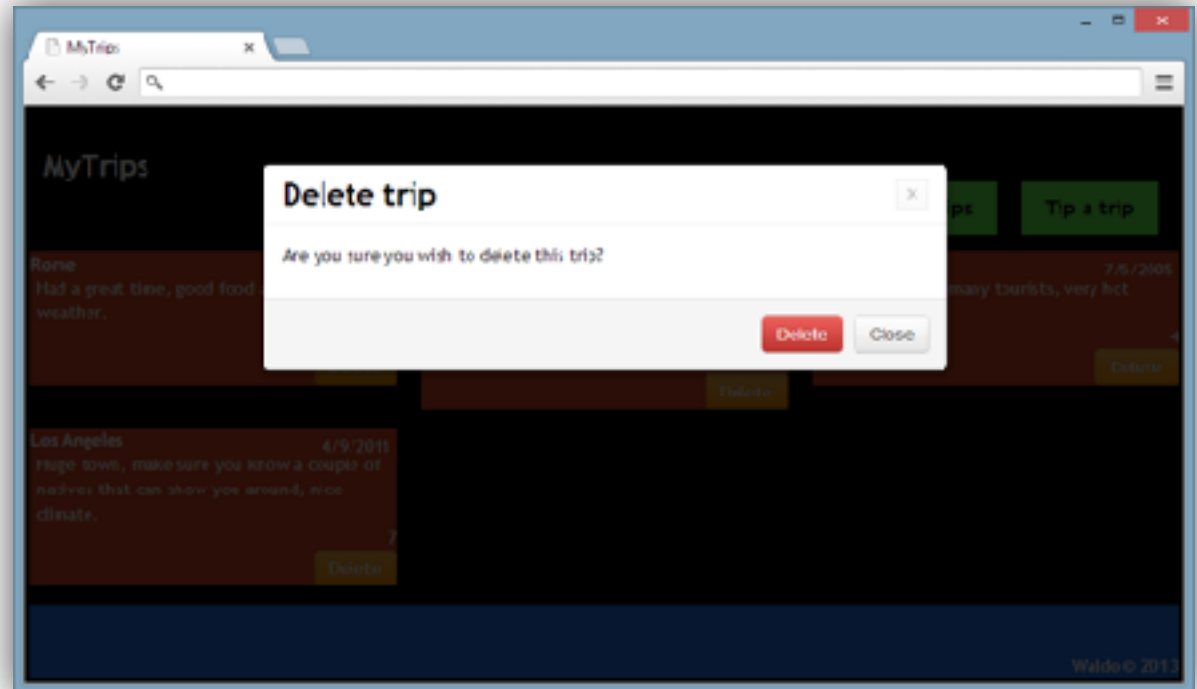
ar

- Arizona
- Arkansas
- Delaware
- Maryland
- North Carolina

JS

Lab: Bootstrap

■ Exercise 6: Use modal dialogs for deleting trips



CSS frameworks wrap-up

- CSS frameworks speed up front-end development
 - Support for positioning, typography and common UI components
- There are many
 - 960 and Bootstrap, but also Blueprint, Skeleton, jQuery UI, Formalize, 1140 CSS Grid, Boilerplate and many more

Agenda

■ Building layouts

- Fixed layouts
- Fluid layouts
- Responsive layouts



- Getting started
- Mixins
- Nested rules
- Operations
- More LESS

■ LESS vs Sass

■ CSS frameworks

- 960 grid system
- Bootstrap, from Twitter



- Getting started
- Mixins and extending
- Nested rules
- Operations
- Control directives
- More Sass



- A dynamic stylesheet language that translates to CSS
- Backwards compatible with CSS
- Helps in maintaining CSS by offering
 - Calculations through formulas
 - Let `<h1>` be 8 pixels bigger than the default font size
 - Variables
 - Define a color once and use it multiple times
 - Nested styling

- LESS files can be translated on both the **client-side** and **server-side**
- Client-side translation
 - JavaScript as the translator - www.lesscss.org
- Server-side
 - ASP.NET: .Less - www.dotlesscss.org
(also available as a NuGet package: dotless)
 - Java: Asual - www.asual.com/lesscss/

■ Link your LESS stylesheets

```
<link href="default.less"  
      rel="stylesheet/less" type="text/css" />
```

HTML

■ Parse the LESS stylesheets with JavaScript

```
<script src="less.js" type="text/javascript"></script>
```

HTML

↖ Downloadable from
www.lesscss.org

■ Drawbacks

- The page renders a bit slower
- JavaScript is required

■ Define variables in your stylesheet

```
@main-color: #5c87b2;  
body {  
  background-color: @main-color;  
}
```

– Note: They're really constants

■ Examples of variable definitions

```
@var1: blue; // Color
```

```
@var2: 12px; // Unit
```

```
@var3: 1.1em; // Unit
```

```
@var4: @var2 + 8px; // Formula
```

Agenda

■ Building layouts

- Fixed layouts
- Fluid layouts
- Responsive layouts



- Getting started
- Mixins
- Nested rules
- Operations
- More LESS

■ LESS vs Sass

■ CSS frameworks

- 960 grid system
- Bootstrap, from Twitter



- Getting started
- Mixins and extending
- Nested rules
- Operations
- Control directives
- More Sass

■ Define a mixin

```
.border {  
  border-top: 3px dotted #666666;  
  border-bottom: 4px solid black;  
}
```

← Note: This class
also appears in
your compiled CSS

■ And reuse the defined CSS properties

LESS

```
.message {  
  .border;  
}  
#nav li {  
  .border;  
}
```



CSS

```
.message {  
  border-top: 3px dotted  
#666666;  
  border-bottom: 4px solid  
black;  
}  
#nav li {  
  border-top: 3px dotted  
#666666;
```

■ Supply parameters to a mixin

```
.transition(@duration) {  
  -webkit-transition: all @duration ease;  
  -o-transition: all @duration ease;  
  -ms-transition: all @duration ease;  
  -moz-transition: all @duration ease;  
  transition: all @duration ease;  
}
```

■ And use the mixin

LESS

```
#nav a {  
  .transition(0.4s);  
}
```

CSS

```
#nav a {  
  -webkit-transition: all 0.4s  
  ease;  
  -o-transition: all 0.4s ease;  
  [...]  
}
```


■ Support for overloading

```
.transition(@duration) {  
  transition: all @duration ease;  
}
```

```
.transition(@property, @duration) {  
  transition: @property @duration ease;  
}
```

■ Usage

```
p {  
  .transition(5s)  
;  
}
```

```
p {  
  .transition(background-color,  
3s);  
}
```

■ Support for default values

```
.border(@thickness: 5px) {  
  border: @thickness solid black;  
}
```

■ Usage

```
p {  
  .border;  
}
```

```
p {  
  .border();  
}
```

```
p {  
  .border(10px)  
;  
}
```

■ Parametric mixins without parameters

```
.border() {  
  border-top: 3px dotted #666666;  
  border-bottom: 4px solid black;  
}
```

- Advantage: `.border` does not appear in the compiled CSS anymore

■ Usage is unchanged

```
.message {  
  .border;  
}  
#nav li {  
  .border;  
}
```

```
.message {  
  .border();  
}  
#nav li {  
  .border();  
}
```

■ Match patterns by supplying static values

```
.mixin (dark, @color) {  
    background-color: darken(@color, 10%);  
}  
.mixin (light, @color) {  
    background-color: lighten(@color, 10%);  
}  
.mixin (@_, @color) {  
    display: block;  
}
```

```
@switch: dark;  
#div1 {  
    .mixin(@switch, #888);  
}
```

■ Match on expressions with guards

```
.mixin (@a) when (lightness(@a) >= 50%) {  
  background-color: black;  
}  
.mixin (@a) when (lightness(@a) < 50%) {  
  background-color: white;  
}
```

LESS

```
#div1 { .mixin(#ddd); }  
#div2 { .mixin(#555); }
```

Note: Not supported
by all implementations

CSS

```
#div1 {  
  background-color:  
black;  
}  
#div2 {  
  background-color:  
white;  
}
```

■ Helpful functions for using with guards

```
iscolor(@color);
```

```
isnumber(@number);
```

```
ispercentage(@percentage)
```

```
:
```

```
isem(@number);
```

```
isstring(@string);
```

```
iskeyword(@keyword);
```

```
isurl(@url);
```

```
ispixel(@number);
```

```
.mixin (@a, @b: black) when (iscolor(@b)) { ... }
```

```
.mixin (@a) when (isnumber(@a)) and (@a > 0) { ... }
```

```
.mixin (@a) when (isnumber(@a)) and not (@a > 0) { ...  
}
```

Agenda

■ Building layouts

- Fixed layouts
- Fluid layouts
- Responsive layouts



- Getting started
- Mixins
- Nested rules
- Operations
- More LESS

■ LESS vs Sass

■ CSS frameworks

- 960 grid system
- Bootstrap, from Twitter



- Getting started
- Mixins and extending
- Nested rules
- Operations
- Control directives
- More Sass

■ Without nested rules...

CSS

```
#nav {  
    background-color: #660066;  
}  
#nav a {  
    display: block;  
    float: left;  
    padding: 10px;  
    text-transform: uppercase;  
    font-weight: bold;  
    text-decoration: none;  
    color: White;  
}  
#nav a:hover, nav a.current-page {  
    background-color: #bb00bb;  
}
```


■ With nested rules...

```
#nav {  
  background-color: #660066;  
  a {  
    display: block;  
    float: left;  
    padding: 10px;  
    text-transform: uppercase;  
    font-weight: bold;  
    text-decoration: none;  
    color: White;  
    &:hover, &.current-page {  
      background-color: #bb00bb;  
    }  
  }  
}
```

& to concatenate to its parent
instead of acting as a
descendant

Agenda

■ Building layouts

- Fixed layouts
- Fluid layouts
- Responsive layouts



- Getting started
- Mixins
- Nested rules
- Operations
- More LESS

■ LESS vs Sass

■ CSS frameworks

- 960 grid system
- Bootstrap, from Twitter



- Getting started
- Mixins and extending
- Nested rules
- Operations
- Control directives
- More Sass

■ Perform operations

```
h1 {  
  font-size: 14px + 4px; // 18px  
}
```

```
@base-color: #888 / 4;  
h1 {  
  font-size: @base-color + #111; // #333333  
}
```

```
@container-width: 960px;  
body {  
  width: (@container-width / 3) + 10; // 330px  
}
```

■ Math operations

```
round(@number);
```

```
floor(@number);
```

```
ceil(1.7);
```

```
percentage(0.5);
```

■ Useful during calculations

```
@container-width: 1000px;  
#container {  
  width: ceil(@container-width / 3); // 334px  
}
```

Agenda

■ Building layouts

- Fixed layouts
- Fluid layouts
- Responsive layouts



- Getting started
- Mixins
- Nested rules
- Operations
- More LESS

■ LESS vs Sass

■ CSS frameworks

- 960 grid system
- Bootstrap, from Twitter



- Getting started
- Mixins and extending
- Nested rules
- Operations
- Control directives
- More Sass

- Extensive support for dealing with **colors**
 - Retrieving color information
 - Transforming colors

■ Retrieve color information

```
hue(@color);
```

```
saturation(@color);
```

```
lightness(@color);
```

```
alpha(@color);
```

■ Useful when creating new colors

```
#menu {  
  background-color: hsl(hue(@base-color), 45%, 72%);  
}
```

■ When transforming, colors are first converted to HSL

```
lighten(@base-color, 10%);
```

```
darken(@base-color, 10%);
```

```
saturate(@base-color, 10%);
```

```
desaturate(@base-color,  
10%);
```

```
mix(@base-color, @new-  
color);
```

```
fadein(@base-color,  
10%);
```

```
fadeout(@base-color,  
10%);
```

```
fade(@base-color, 10%);
```

```
spin(@base-color, 10);
```

```
spin(@base-color, -10);
```

```
h1 {  
  color: darken(@base-color, 25%);  
}
```

■ Import LESS stylesheets

- Reuse the defined variables and mixins

```
@import 'general.less';
```

```
@import 'general';
```

■ Import CSS stylesheets

```
@import 'general.css';
```

CSS

- This statement simply ends up in the CSS output

■ Parse variables within strings

```
@base-url: "http://www.myserverhere.com";  
body  
{  
    background-image: url( "@{base-url}/images/bg.png" );  
}
```

■ Support for single-line comments

```
@var1: blue; // Color
```

- These comments **do not appear** in the CSS output

■ Multi-line comments do remain visible in the CSS output

```
/* Navigation */  
#nav {  
    background-color: #660066;  
}  
[...]
```

CSS

Lab: LESS is more

■ Exercise 7: Apply LESS to your stylesheets



Agenda

■ Building layouts

- Fixed layouts
- Fluid layouts
- Responsive layouts



- Getting started
- Mixins
- Nested rules
- Operations
- More LESS

■ LESS vs Sass

■ CSS frameworks

- 960 grid system
- Bootstrap, from Twitter



- Getting started
- Mixins and extending
- Nested rules
- Operations
- Control directives
- More Sass

Syntactically Awesome Stylesheets

- **A dynamic stylesheet language that translates to CSS**
- **Helps in maintaining CSS by offering**
 - Calculations through formulas
 - Let `<h1>` be 8 pixels bigger than the default font size
 - Variables
 - Define a color once and use it in various places
 - Support for reusing styles and nesting styles
 - All that and more

■ Server-side

- Originated in Ruby
- ASP.NET: SassAndCoffee <http://nuget.org/packages/SassAndCoffee>

■ Client-side translation is ongoing

- JavaScript as the translator

■ Sass actually comes in two syntaxes

■ SCSS

- A superset of CSS3

```
$blue: #3bbfce;
$margin: 16px;

.content-navigation {
  border-color: $blue;
}

.border {
  padding: $margin / 2;
  border-color: $blue;
}
```

■ Sass

- The “indented syntax”

```
$blue: #3bbfce
$margin: 16px

.content-navigation
  border-color: $blue

.border
  padding: $margin / 2
  border-color: $blue
```

■ Define variables in your stylesheet

```
$main-color: #5c87b2;  
body {  
  background-color: @main-color;  
}
```

■ Examples of variable definitions

```
$var1: blue; // Color
```

```
$var2: 12px; // Unit
```

```
$var3: 1.1em; // Unit
```

```
$var4: $var2 + 8px; // Formula
```


Agenda

■ Building layouts

- Fixed layouts
- Fluid layouts
- Responsive layouts



- Getting started
- Mixins
- Nested rules
- Operations
- More LESS

■ LESS vs Sass

■ CSS frameworks

- 960 grid system
- Bootstrap, from Twitter



- Getting started
- Mixins and extending
- Nested rules
- Operations
- Control directives
- More Sass

■ Define a mixin

```
@mixin border {  
  border-top: 3px solid #666666;  
  border-bottom: 4px solid black;  
}
```

■ And reuse the defined CSS properties

Sass

```
.message {  
  @include  
border;  
}  
#nav li {  
  @include  
border;  
}
```



CSS

```
.message {  
  border-top: 3px dotted  
#666666;  
  border-bottom: 4px solid  
black;  
}  
#nav li {  
  border-top: 3px dotted  
#666666;
```

■ Supply parameters to a mixin

```
@mixin transition($duration) {  
  -webkit-transition: all $duration ease;  
  -o-transition: all $duration ease;  
  -ms-transition: all $duration ease;  
  -moz-transition: all $duration ease;  
  transition: all $duration ease;  
}
```

■ And use the mixin

Sass

```
#nav a {  
  @include transition(0.4s);  
}
```

CSS

```
#nav a {  
  [...]  
  transition: all 0.4s  
  ease;  
}
```

■ Support for default values

```
@mixin border($thickness: 5px) {  
  border: $thickness solid black;  
}
```

■ Usage

```
p {  
  @include border;  
}
```

```
p {  
  @include border();  
}
```

```
p {  
  @include border(3px);  
}
```

■ Inheritance in CSS

Sass

```
.error {  
  border: 1px solid red;  
  background: pink;  
}  
.error.intrusion {  
  background: darkred;  
}  
.serious-error {  
  @extend .error;  
  border-width: 3px;  
}
```

CSS

```
.error, .serious-error {  
  border: 1px solid red;  
  background: pink;  
}  
.error.intrusion, .intrusion.serious-error {  
  background: darkred;  
}  
.serious-error {  
  border-width: 3px;  
}
```

– Multiple extensions are supported

Agenda

■ Building layouts

- Fixed layouts
- Fluid layouts
- Responsive layouts



- Getting started
- Mixins
- Nested rules
- Operations
- More LESS

■ LESS vs Sass

■ CSS frameworks

- 960 grid system
- Bootstrap, from Twitter



- Getting started
- Mixins and extending
- Nested rules
- Operations
- Control directives
- More Sass

■ Without nested rules...

CSS

```
#nav {  
  background-color: #660066;  
}  
#nav a {  
  display: block;  
  float: left;  
  padding: 10px;  
  text-transform: uppercase;  
  font-weight: bold;  
  text-decoration: none;  
  color: White;  
}  
#nav a:hover, nav a.current-page {  
  background-color: #bb00bb;  
}
```

■ With nested rules...

```
#nav {  
  background-color: #660066;  
  a {  
    display: block;  
    float: left;  
    padding: 10px;  
    text-transform: uppercase;  
    font-weight: bold;  
    text-decoration: none;  
    color: White;  
    &:hover, &.current-page {  
      background-color: #bb00bb;  
    }  
  }  
}
```

& to concatenate to its parent
instead of acting as a
descendant

■ With nested rules...

```
#nav {  
  background-color: #660066;  
  a {  
    display: block;  
    float: left;  
    padding: 10px;  
    text-transform: uppercase;  
    font-weight: bold;  
    text-decoration: none;  
    border: {  
      left: 3px solid #660000;  
      right: 10px solid #000080;  
    }  
  }  
}
```

& to concatenate to its parent
instead of acting as a
descendant

Agenda

■ Building layouts

- Fixed layouts
- Fluid layouts
- Responsive layouts



- Getting started
- Mixins
- Nested rules
- Operations
- More LESS

■ LESS vs Sass

■ CSS frameworks

- 960 grid system
- Bootstrap, from Twitter



- Getting started
- Mixins and extending
- Nested rules
- Operations
- Control directives
- More Sass

■ Perform operations

```
h1 {  
  font-size: 14px + 4px; // 18px  
}
```

```
$base-color: #888 / 4;  
h1 {  
  font-size: $base-color + #111; // #333333  
}
```

```
@container-width: 960px;  
body {  
  width: ($container-width / 3) + 10; // 330px  
}
```

■ Math operations

```
round($number);
```

```
floor($number);
```

```
ceil(1.7);
```

```
percentage(0.5);
```

```
min(8px, 1px, 4px);
```

```
max(8px, 1px, 4px);
```

■ Useful during calculations

```
$container-width: 1000px;  
#container {  
  width: ceil($container-width / 3); // 334px  
}
```

Agenda

■ Building layouts

- Fixed layouts
- Fluid layouts
- Responsive layouts



- Getting started
- Mixins
- Nested rules
- Operations
- More LESS

■ LESS vs Sass

■ CSS frameworks

- 960 grid system
- Bootstrap, from Twitter



- Getting started
- Mixins and extending
- Nested rules
- Operations
- Control directives
- More Sass

■ Support for loops with @for

Sass

```
@for $i from 1 through 3 {  
  .item-#{ $i } { width: 2em * $i; }  
}
```

```
.item-1 { width: 2em; }  
.item-2 { width: 4em; }  
.item-3 { width: 6em; }
```

CSS

■ Support for loops with @each

Sass

```
@each $animal in puma, sea-slug, egret, salamander {  
  .#{ $animal }-icon {  
    background-image: url( '/images/#{ $animal }.png' );  
  }  
}
```

CSS

```
.puma-icon {  
  background-image: url( "/images/puma.png" );  
}  
.sea-slug-icon {  
  background-image: url( "/images/sea-slug.png" );  
}  
[...]
```

■ Support for loops with @while

Sass

```
$i: 6;
@while $i > 0 {
  .item-#{ $i } { width: 2em * $i; }
  $i: $i - 2;
}
```

```
.item-6 { width: 12em; }
.item-4 { width: 8em; }
.item-2 { width: 4em; }
```

CSS

■ Support for loops with @if

Sass

```
$type: monster;
p {
  @if $type == ocean { color: blue; }
  @else if $type == monster { color: green; }
  @else { color: black; }
}
```

CSS

```
p {
  color: green;
}
```

Agenda

■ Building layouts

- Fixed layouts
- Fluid layouts
- Responsive layouts



- Getting started
- Mixins
- Nested rules
- Operations
- More LESS

■ LESS vs Sass

■ CSS frameworks

- 960 grid system
- Bootstrap, from Twitter



- Getting started
- Mixins and extending
- Nested rules
- Operations
- Control directives
- More Sass

- Extensive support for dealing with **colors**
 - Retrieve color information
 - Transform colors

■ Retrieve color information

```
hue(@color);
```

```
saturation(@color);
```

```
lightness(@color);
```

```
alpha(@color);
```

■ Useful when creating new colors

```
#menu {  
  background-color: hsl(hue(@base-color), 45%, 72%);  
}
```

■ Support for transforming colors

```
lighten($base-color, 10%);
```

```
invert($base-color);
```

```
darken($base-color, 10%);
```

```
grayscale($base-color);
```

```
saturate($base-color, 10%);
```

```
fade-in($base-color,  
0.1);
```

```
desaturate($base-color,  
10%);
```

```
fade-out($base-color,  
0.1);
```

```
mix($base-color, $new-  
color);
```

```
rgba($acolor, $alpha);
```

```
h1 {  
  color: darken($base-color, 25%);  
}
```

■ Import Sass stylesheets

- Reuse the defined variables and mixins

```
@import 'general.scss';
```

```
@import 'general';
```

■ Import CSS stylesheets

```
@import 'general.css';
```

CSS

- This statement simply ends up in the CSS output

■ Parse variables within strings

```
$base-url: "http://www.myserverhere.com";  
body  
{  
  background-image: url( "#{$base-url}/images/bg.png" );  
}
```

■ Support for single-line comments

```
$var1: blue; // Color
```

- These comments **do not appear** in the CSS output

■ Multi-line comments do remain visible in the CSS output

```
/* Navigation */  
#nav {  
  background-color: #660066;  
}  
[...]
```

CSS

Agenda

■ Building layouts

- Fixed layouts
- Fluid layouts
- Responsive layouts



- Getting started
- Mixins
- Nested rules
- Operations
- More LESS

■ LESS vs Sass

■ CSS frameworks

- 960 grid system
- Bootstrap, from Twitter



- Getting started
- Mixins and extending
- Nested rules
- Operations
- Control directives
- More Sass



- Basic reuse and readability improvements
- Client-side parser
- Website niceitude



- Basic reuse and readability improvements
- More explicit with `@extend`, `@include` and `@mixin`
- Respects CSS more with `$`
- More robust
 - Control directives
 - Compass
- Community presence

Winner:



Questions

