



# INTRODUCING LEXICON CSS

Copyright ©2017 Liferay, Inc.

All Rights Reserved.

No material may be reproduced electronically or in print,  
distributed, copied, sold, resold, or otherwise exploited  
for any commercial purpose without express written  
consent of Liferay, Inc.

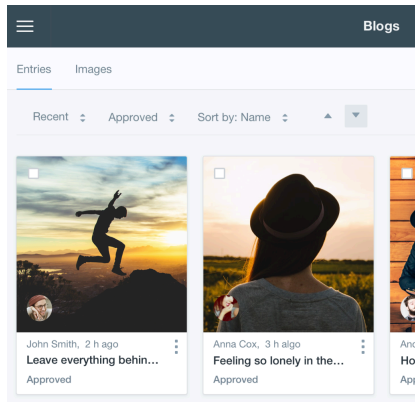
## WHAT IS LEXICON CSS?

- ❖ Lexicon CSS a web implementation of the Liferay's Lexicon Design Language.
- ❖ Lexicon CSS provides style guidelines and best practices for designing web applications.
- ❖ It is meant to be fluid and extensible while providing a complete collection of visual and interactive patterns.
- ❖ It is an extension of the Bootstrap 3 Framework and is built with Sass.
- ❖ Lexicon CSS fills the front-end gaps between Bootstrap and the specific needs of Liferay.
- ❖ Lexicon CSS is optimized for the most common operations and takes the best from the most popular modern UI patterns available.
- ❖ These patterns can be carefully combined, like Lego pieces, to obtain the desired user experience.



# LEXICON CSS FEATURES

- ❖ Lexicon CSS has added components and features to cover more use cases:
  - ❖ Aspect Ratio
  - ❖ Cards
  - ❖ Dropdown Wide and Dropdown Full
  - ❖ Figures
  - ❖ Nameplates
  - ❖ Sidebar/Sidenav
  - ❖ Stickers
  - ❖ SVG Icons
  - ❖ Timelines
  - ❖ Toggles



# REUSABLE PATTERNS

- ❖ We've also added a lot of reusable CSS patterns to help accomplish time-consuming tasks such as:
  - ❖ Truncating text
  - ❖ Content filling the remaining container width
  - ❖ Truncating text inside table cells
  - ❖ Table cells filling remaining container width and table cells only being as wide as their contents
  - ❖ Open and closed icons inside collapsible panels
  - ❖ Nested vertical navigations
  - ❖ Slideout panels
  - ❖ Notification icons/messages
  - ❖ Vertical alignment of content

# NEW LEXICON CSS ICONS

- ❖ In previous versions of Liferay, the Bootstrap implementation in AUI took advantage of Font Awesome and Glyphicons for scalable icons.
- ❖ Lexicon CSS adds its own set of SVG icons that are bundled in the base theme.
- ❖ SVG gives developers a greater amount of freedom in styling the icons, as well as a higher level of fidelity and clarity in the icons.
- ❖ The SVG icons also allow for multi-color styling.

 add-cell

 adjust

 align-left

 angle-left

 archive

 bars

 bookmarks

 cards

 caret-double-l

 add-column

 align-center

 align-right

 angle-right

 asterisk

 blogs

 calendar

 cards2

 caret-double

 add-row

 align-justify

 angle-down

 angle-up

 audio

 bold

 camera

 caret-bottom

 caret-top

# USING FONT AWESOME

- ❖ If you've used Font Awesome in the past, you can continue to use it in your theme by adding the following in your `_au_i_variables.scss` file:

```
// Icon paths
$FontAwesomePath: "au_i/lexicon/fonts/alloy-font-awesome/font";
$font-awesome-path: "au_i/lexicon/fonts/alloy-font-awesome/font";
$icon-font-path: "au_i/lexicon/fonts/";
```

 align-center

**B** bold

 columns

 eraser

 file-text-o

**H** header

 list

 align-justify

 chain (alias)

 copy (alias)

 file

 files-o

 indent

 list-alt

 align-left

 chain-broken

 cut (alias)

 file-o

 floppy-o

*I* italic

 list-ol

 align-right

 clipboard

 dedent (alias)

 file-text

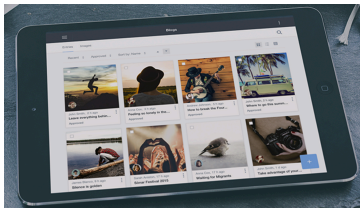
**A** font

 link

 list-ul

# LEXICON CSS RESPONSIVE DESIGN

- ❖ Lexicon CSS has replaced the previous `respond-to` mixin with explicit media queries.
- ❖ Now, to include custom responsive design, you can use the following:
  - ❖ **Phone:** `include media-query(null, $screen-xs-max)`
  - ❖ **Tablet:** `include media-query(sm, $screen-sm-max)`
  - ❖ **Phone - Tablet:** `include media-query(null, $breakpoint_tablet - 1)`
  - ❖ **Tablet - Desktop:** `include sm`
  - ❖ **Desktop:** `include media-query($breakpoint_tablet, null)`



## LEXICON BASE

- ❖ When building a theme, developers start with the *Lexicon Base* theme.
- ❖ *Lexicon Base* is our Bootstrap API extension and includes all of the default Lexicon CSS styles.
- ❖ When a theme is created, developers can choose from the following options as a starting point for customization:
  - ❖ Building on the *Unstyled* base:
    - ❖ This base theme does not include the *Lexicon Base*, allowing developers to have full control over styling for every aspect of the platform.
  - ❖ Building on the *Styled* base:
    - ❖ This base theme includes the *Lexicon Base* theme.
    - ❖ It adds all the features and components we need and inherits Bootstrap's styles.



# ATLAS

- ❖ Though *Lexicon Base* is the default import of a *Styled* theme, developers can also easily use what is called the *Atlas* theme.
- ❖ Atlas is Liferay's custom Bootstrap theme, used in the Classic Theme.
- ❖ Its purpose is to overwrite and manipulate Bootstrap and Lexicon Base to create our Classic look-and-feel.
- ❖ The benefit to using Atlas is that you'll gain some classic styles without trying to customize the existing Classic Theme itself.

## USING ATLAS IN YOUR CUSTOM THEME

- ❖ Using *Atlas* instead of *Lexicon Base* is a simple switch in imports.
- ❖ To start, you can import Atlas in an `aui.scss` file:  
`@import "aui/lexicon/atlas";`
- ❖ Then you can import the various packages, variables, and mixins for Atlas in an `_imports.scss` file:

```
@import "bourbon";  
@import "mixins";  
@import "aui/lexicon/atlas-variables";  
@import "aui/lexicon/bootstrap/mixins";  
@import "aui/lexicon/lexicon-base/mixins";  
@import "aui/lexicon/atlas-theme/mixins";
```

- ❖ Building on the *Styled* base is recommended for theme development as it will contain both *Lexicon Base* and *Atlas*.

## CUSTOMIZING THE CLASSIC THEME

- ❖ If you would like to include all of the Classic Theme instead of just the base styles of *Atlas*, you can copy over the files located in:

```
frontend-theme-classic/src/css  
frontend-theme-classic/src/images  
frontend-theme-classic/src/js  
frontend-theme-classic/src/templates
```

- ❖ This can also be accomplished using the Liferay Theme Tasks module `gulp kickstart` command and following the prompts.

```
[09:26:53] Starting 'kickstart'...  
[09:26:53] Warning: the kickstart task will potentially overwrite files in your  
src directory  
? Where would you like to search?  
  Search globally installed npm modules  
  Search npm registry (published modules)  
  Search npm registry (published modules)  
> Classic
```

# USING LEXICON CSS

- ❖ Lexicon CSS can be used in both theme development and template development.
- ❖ Later, we'll create our own custom themes and templates that take advantage of these features.
- ❖ For a list of components, elements, etc., visit:  
<http://liferay.github.io/clay/>



Notes: