# TITON

## Toolkit

TITON

# Toolkit

Front-end library for the responsive and mobile web

titon.io/toolkit
github.com/titon/toolkit

# About Titon

# About Titon

- A personal project for the past 5 years

# About Titon

- A personal project for the past 5 years
- Provides front-end and back-end (PHP) libraries

# About Titon

- A personal project for the past 5 years
- Provides front-end and back-end (PHP) libraries
- Because coding is fun

# About Titon

- A personal project for the past 5 years
- Provides front-end and back-end (PHP) libraries
- Because coding is fun
- And a great learning exercise

# About Titon

- A personal project for the past 5 years
- Provides front-end and back-end (PHP) libraries
- Because coding is fun
- And a great learning exercise
- Also gives back to the community

# What is Toolkit?

# What is Toolkit?

- A front-end library that provides pre-built user interface components

# What is Toolkit?

- A front-end library that provides pre-built user interface components
- Requires jQuery or MooTools for the JS layer

# What is Toolkit?

- A front-end library that provides pre-built user interface components
- Requires jQuery or MooTools for the JS layer
- Uses Sass and Compass for the CSS layer

# What is Toolkit?

- A front-end library that provides pre-built user interface components
- Requires jQuery or MooTools for the JS layer
- Uses Sass and Compass for the CSS layer
- Utilizes Grunt for development and builds

# What is Toolkit?

- A front-end library that provides pre-built user interface components
- Requires jQuery or MooTools for the JS layer
- Uses Sass and Compass for the CSS layer
- Utilizes Grunt for development and builds
- Available through Bower, Ruby Gems, and NPM

# Philosophies

# Philosophies

- Mobile first

# Philosophies

- Mobile first
- Responsive (fluid, em, media queries)

# Philosophies

- Mobile first
- Responsive (fluid, em, media queries)
- Semantic markup (HTML5)

# Philosophies

- Mobile first
- Responsive (fluid, em, media queries)
- Semantic markup (HTML5)
- CSS3 transitions and animations

# Philosophies

- Mobile first
- Responsive (fluid, em, media queries)
- Semantic markup (HTML5)
- CSS3 transitions and animations
- Styleless

# Philosophies

- Mobile first
- Responsive (fluid, em, media queries)
- Semantic markup (HTML5)
- CSS3 transitions and animations
- Styleless
- Extensible (options, templates, styles, components)

# Philosophies

- Mobile first
- Responsive (fluid, em, media queries)
- Semantic markup (HTML5)
- CSS3 transitions and animations
- Styleless
- Extensible (options, templates, styles, components)
- Progressive enhancement

# Philosophies

- Mobile first
- Responsive (fluid, em, media queries)
- Semantic markup (HTML5)
- CSS3 transitions and animations
- Styleless
- Extensible (options, templates, styles, components)
- Progressive enhancement
- Graceful degradation

# Why use it?

# Why use it?

- Engineered for mobile, tablet, and desktop devices

# Why use it?

- Engineered for mobile, tablet, and desktop devices
- Decoupled modular architecture; custom builds

# Why use it?

- Engineered for mobile, tablet, and desktop devices
- Decoupled modular architecture; custom builds
- Extremely extensible, encourages a configuration over convention approach

# Why use it?

- Engineered for mobile, tablet, and desktop devices
- Decoupled modular architecture; custom builds
- Extremely extensible, encourages a configuration over convention approach
- Easy to integrate

# Why use it?

- Engineered for mobile, tablet, and desktop devices
- Decoupled modular architecture; custom builds
- Extremely extensible, encourages a configuration over convention approach
- Easy to integrate
- Easy to customize and style

# Why use it?

- Engineered for mobile, tablet, and desktop devices
- Decoupled modular architecture; custom builds
- Extremely extensible, encourages a configuration over convention approach
- Easy to integrate
- Easy to customize and style
- Provides more features compared to current competitors

# Why use it?

- Engineered for mobile, tablet, and desktop devices
- Decoupled modular architecture; custom builds
- Extremely extensible, encourages a configuration over convention approach
- Easy to integrate
- Easy to customize and style
- Provides more features compared to current competitors
- Built in-house so allows for quick bug fixing, new feature requests, agile development, peer reviewing and collaboration

# Why not use it?

# Why not use it?

- Already using Foundation, Bootstrap, etc in a project

# Why not use it?

- Already using Foundation, Bootstrap, etc in a project
- Want pre-designed components as Toolkit does not provide default aesthetics

# Why not use it?

- Already using Foundation, Bootstrap, etc in a project
- Want pre-designed components as Toolkit does not provide default aesthetics
- Takes a bit longer for integration as everything must be configured and styled

# Why not use it?

- Already using Foundation, Bootstrap, etc in a project
- Want pre-designed components as Toolkit does not provide default aesthetics
- Takes a bit longer for integration as everything must be configured and styled
- When only 1 or 2 components are needed, the JS and CSS overhead may be too much

# Compared to the Competition

# Compared to the Competition

- Supports a vendor namespacing system for component and states classes

# Compared to the Competition

- Supports a vendor namespacing system for component and states classes
- Has 36 (17 JS) unique components compared to Bootstrap’s 31 (11 JS) and Foundation’s 39 (15 JS) [link]

# Compared to the Competition

- Supports a vendor namespacing system for component and states classes
- Has 36 (17 JS) unique components compared to Bootstrap's 31 (11 JS) and Foundation's 39 (15 JS) [link]
- More configuration options within JS components

# Compared to the Competition

- Supports a vendor namespacing system for component and states classes
- Has 36 (17 JS) unique components compared to Bootstrap’s 31 (11 JS) and Foundation’s 39 (15 JS) [link]
- More configuration options within JS components
- Easier to customize CSS styles (does not use Sass variables)

# Compared to the Competition

- Supports a vendor namespacing system for component and states classes
- Has 36 (17 JS) unique components compared to Bootstrap's 31 (11 JS) and Foundation's 39 (15 JS) [link]
- More configuration options within JS components
- Easier to customize CSS styles (does not use Sass variables)
- Lighter distribution files (JS 55kb, CSS 61kb) compared to Bootstrap (JS 29kb, CSS 101kb) and Foundation (JS 70kb, CSS 107kb)

# Compared to the Competition

- Supports a vendor namespacing system for component and states classes
- Has 36 (17 JS) unique components compared to Bootstrap's 31 (11 JS) and Foundation's 39 (15 JS) [link]
- More configuration options within JS components
- Easier to customize CSS styles (does not use Sass variables)
- Lighter distribution files (JS 55kb, CSS 61kb) compared to Bootstrap (JS 29kb, CSS 101kb) and Foundation (JS 70kb, CSS 107kb)
- Provides a powerful JS Component class for custom components [link]

# Compared to the Competition

- Supports a vendor namespacing system for component and states classes
- Has 36 (17 JS) unique components compared to Bootstrap’s 31 (11 JS) and Foundation’s 39 (15 JS) [link]
- More configuration options within JS components
- Easier to customize CSS styles (does not use Sass variables)
- Lighter distribution files (JS 55kb, CSS 61kb) compared to Bootstrap (JS 29kb, CSS 101kb) and Foundation (JS 70kb, CSS 107kb)
- Provides a powerful JS Component class for custom components [link]
- Provides Sass mixins and functions [link]

# Compared to the Competition

- Supports a vendor namespacing system for component and states classes
- Has 36 (17 JS) unique components compared to Bootstrap's 31 (11 JS) and Foundation's 39 (15 JS) [link]
- More configuration options within JS components
- Easier to customize CSS styles (does not use Sass variables)
- Lighter distribution files (JS 55kb, CSS 61kb) compared to Bootstrap (JS 29kb, CSS 101kb) and Foundation (JS 70kb, CSS 107kb)
- Provides a powerful JS Component class for custom components [link]
- Provides Sass mixins and functions [link]
- Extends jQuery and MooTools with new methods

# Installation

# Installation

- Download

# Installation

- Download
- Git clone

# Installation

- Download
- Git clone
    - Custom builds with Grunt

# Installation

- Download
- Git clone
    - Custom builds with Grunt
- Ruby gem + Compass extension

# Installation

- Download
- Git clone
    - Custom builds with Grunt
- Ruby gem + Compass extension
- Bower

# Installation

- Download
- Git clone
    - Custom builds with Grunt
- Ruby gem + Compass extension
- Bower
- NPM

# Installation

- Download
- Git clone
    - Custom builds with Grunt
- Ruby gem + Compass extension
- Bower
- NPM
- Composer

# Demos

Live interactive demos can be found within the GitHub repository. Each component has a respective demo allowing options to be tested and configured.

Interacting with the demos will require a running PHP web server.

# Thanks!

Slides presented by Miles Johnson, developer behind the Titon Toolkit and Framework.

https://github.com/milesj

https://twitter.com/mileswjohnson

Feedback and comments are welcome! Thanks again for checking out Titon.