



ZURB Foundation



tutorialspoint
SIMPLY EASY LEARNING

www.tutorialspoint.com

About the Tutorial

Foundation is one of the advanced front-end frameworks for designing beautiful responsive websites. It works on all types of devices and provides you with HTML, CSS and JavaScript plugin. The tutorial is divided into sections, such that each section contains related topics with simple and useful examples.

Audience

This tutorial has been prepared for anyone who has a basic knowledge of HTML and CSS and has an urge to develop websites. After completing this tutorial, you will find yourself at a moderate level of expertise in developing web projects using Zurb Foundation Bootstrap.

Prerequisites

Before you proceed with this tutorial, we are assuming that you are already aware about the basics of HTML and CSS. If you are not well aware of these concepts then we will suggest you to go through our short tutorials on [HTML Tutorial](#) and [CSS Tutorial](#).

Copyright & Disclaimer

© Copyright 2017 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial	i
Audience.....	i
Prerequisites.....	i
Copyright & Disclaimer	i
Table of Contents	ii
1. Foundation – Overview	1
What is Foundation?	1
History	1
Why to use Foundation?	1
Features.....	1
Advantages	1
Disadvantages.....	2
2. Foundation – Installation	3
Download Foundation	3
File Structure	4
HTML Template	5
3. Foundation – Starter Projects	7
Basic Template	7
ZURB Template	7
Asset Copying	7
Page Compilation.....	7
SASS Compilation.....	8
JavaScript Compilation	8
Image Compression	8
BrowserSync	8
4. Foundation – Kitchen Sink	9
Foundation – Accordion	11
Foundation – Accordion Menu.....	13
Foundation – Badge.....	15
Foundation – Breadcrumbs	16
Foundation – Button	17
Foundation – Callout	19
Foundation – Close Button	21
Foundation – Drilldown Menu	22
Foundation – Dropdown Menu	25
Foundation – Dropdown Pane.....	27
Foundation – Flex Video	28
Foundation – Float Classes	30
Foundation – Forms.....	31
Foundation – Labels	33
Foundation – Media Object.....	34
Foundation – Menu	36
Foundation – Pagination	37
Foundation – Slider	38

Foundation – Switch	40
Foundation – Table	41
Foundation – Tabs	43
Foundation – Thumbnail	45
Foundation – Title Bar	47
Foundation – Tooltip	48
Foundation – Top Bar	49
Foundation – Orbit	51
FOUNDATION GENERAL	54
5. Foundation – Global Styles	55
Font Sizing	55
Colors.....	55
SASS Reference.....	55
6. Foundation – SASS	58
Compatibility	58
Loading the Framework.....	58
Adjusting CSS Output.....	59
The Settings File.....	60
7. Foundation – JavaScript	61
JavaScript Installing	61
File Structure	62
Initializing	62
Using Plugins	63
Configuring Plugins.....	64
Adding Plugins after Page Load	66
Programmatic Use.....	66
Events	67
8. Foundation – JavaScript Utilities	68
Box	68
Keyboard	69
MediaQuery.....	70
Motion & Move	70
Timer & Images Loaded.....	71
Touch.....	71
Triggers	71
Miscellaneous.....	72
9. Foundation – Media Queries	73
Changing the Breakpoints	73
SASS	74
JavaScript.....	74
SASS Reference.....	75
JavaScript Reference	76
10. Foundation – The Grid	77
Description	77
Grid Options	77
Basic Structure of a Foundation Grid	77

Foundation – Large Grid	78
Foundation – Medium Grid	81
Foundation – Small Grid	83
Advanced Grid	85
Foundation – Nesting	87
Foundation – Offsets	89
Foundation – Incomplete Rows	90
Foundation – Collapse/Uncollapse Rows	92
Foundation – Centered Columns	93
Foundation – Source Ordering	95
Foundation – Block Grids	96
Foundation – Building Semantically	98
SASS Reference	100
Foundation – SASS Variables	100
Foundation – SASS Mixins	100
11. Foundation – Flex Grid	105
Browser Support	105
Foundation – Flex Grid Importing	106
Foundation – Flex Grid Basics	106
Foundation – Flex Grid Advanced Sizing	107
Foundation – Flex Grid Responsive Adjustments	109
Automatic Stacking	110
Foundation – Flex Grid Column Alignment	111
Foundation – Flex Grid Source Ordering	114
Foundation – Flex Grid SASS Reference	116
12. Foundation – Forms	118
Foundation – Form Basics	118
Foundation – Forms Text Inputs	119
Foundation – Select Menus	124
Foundation – Checkboxes and Radio Buttons	125
Foundation – Fieldset Styles	127
Foundation – Help Text	128
Foundation – Label Positioning	130
Foundation – Inline Labels and Buttons	132
Foundation – Custom Controls	133
Foundation – Form SASS Reference	134
13. Foundation – Visibility Classes	138
Foundation – Show by Screen Size	138
Foundation – Hide by Screen Size	139
Orientation Detection	141
Accessibility	141
Foundation – Show for Screen Readers	141
Foundation – Hide for Screen Readers	143
Foundation – Skip Links	144
SASS Reference	145
14. Foundation – Base Typography	146
Foundation – Base Typography Paragraph	147
Foundation – Base Typography Header	148
Foundation – Base Typography Links	150

Foundation – Base Typography Dividers	151
Foundation – Base Typography Lists	152
Foundation – Base Typography Definition Lists	154
Foundation – Base Typography Blockquotes	155
Foundation – Base Typography Abbreviations and Code.....	156
Foundation – Base Typography Keystrokes.....	158
Foundation – Base Typography Accessibility.....	159
SASS Reference.....	160
15. Foundation – Typography Helpers	164
Foundation – Text Alignment.....	164
Foundation – Subheader	166
Foundation – Lead Paragraph	167
Foundation – Un-bulleted List.....	169
Foundation – Statistics	171
SASS Reference.....	172
16. Foundation – Basic Controls.....	174
Foundation – Basic Controls Button.....	174
Foundation – Button Basics.....	175
Foundation – Button Sizing	176
Foundation – Button Coloring	178
Foundation – Button Hollow Style.....	179
Foundation – Button Dropdown Arrows	180
Foundation – Button Accessibility	181
Foundation – Button SASS Reference.....	183
Foundation – Button Group	186
Foundation – Button Group Basics.....	186
Foundation – Button Group Sizing	188
Foundation – Button Group Coloring	189
Foundation – Even width Group.....	191
Foundation – Button Stacking	192
Foundation – Split Buttons.....	193
Foundation – Button Group SASS Reference	195
Foundation – Close Button.....	197
Foundation – Button SASS Reference.....	198
Foundation – Slider	199
Foundation – Slider Basics	199
Foundation – Slider Vertical	201
Foundation – Slider Disabled.....	202
Foundation – Slider Two Handles	203
Foundation – Slider Data Binding	205
Foundation – Slider SASS Reference	206
Foundation – Slider JavaScript Reference	208
Foundation – Switch	210
Foundation – Switch Basics	210
Foundation – Radio Switch.....	212
Foundation – Sizing Classes	213
Foundation – Inner Labels	215
Foundation – Switch SASS Reference	216
17. Foundation – Navigation.....	220
Foundation – Navigation Overview	221

Foundation – Navigation Menu	222
Foundation – Menu	222
Foundation – Item Alignment	224
Foundation – Vertical Menu	226
Foundation – Simple Style	227
Foundation – Nested Style	228
Foundation – Icons	230
Foundation – Menu SASS Reference	232
Foundation – Navigation Dropdown Menus	234
Foundation – Horizontal Dropdown Menus	235
Foundation – Vertical Dropdown Menus	237
Foundation – Dropdown SASS reference	238
Foundation – Dropdown JavaScript reference	239
Foundation – Navigation Drilldown Menu	241
Foundation – Navigation Drilldown Basic	241
Foundation – Drilldown JavaScript Reference	243
Foundation – Navigation Accordion Menu	244
Foundation – Accordion Menu	245
Foundation – Accordion Menu SASS Reference	247
Foundation – Accordion Menu JavaScript Reference	247
Foundation – Navigation Top Bar	249
Foundation – Top Bar	249
Foundation – Top Bar SASS Reference	251
Foundation – Responsive Navigation	252
Foundation – Responsive Navigation Menu	252
Foundation – Responsive Navigation Toggle	254
Foundation – Navigation Magellan	256
Foundation – Magellan Setup	256
Foundation – Magellan Sticky Navigation	260
Foundation – Magellan JavaScript Reference	264
Foundation – Navigation Pagination	265
Foundation – Pagination	265
Foundation – Centered Pagination	267
Foundation – Pagination SASS Reference	268
Foundation – Navigation Breadcrumbs	270
18. Foundation – Containers	272
Foundation – Accordion	272
Foundation – Basic Accordion	273
Foundation – Multi Expand Accordion	275
Foundation – Closed Accordion	277
Foundation – Sass Reference	279
Foundation – JavaScript Reference	280
Foundation – Callout	282
Foundation – Basic Callout	283
Foundation – Callout Coloring	284
Foundation – Callout Sizing	286
Foundation – Callout Closable	288
Foundation – Callout SASS Reference	289
Foundation – Dropdown-panes	291
Foundation – Basic Dropdown Pane	291
Foundation – Dropdown Pane Positioning	293
Foundation – Dropdown Pane SASS Reference	295

Foundation – Dropdown Pane JavaScript Reference	296
Foundation – Media Object.....	297
Foundation – Basic Media Object.....	298
Foundation – Media Object Section Alignment	299
Foundation – Media Object Stack on Small.....	301
Foundation – Nesting Media Objects	302
Foundation – Media Objects Sass Reference	305
Foundation – Off-canvas	306
Foundation – Off-canvas Setup	307
Foundation – Off-canvas Multiple Menus	308
Foundation – Off-canvas Title Bar	310
Foundation – Off-canvas Larger Screen Reveal	312
Foundation – Off-canvas SASS Reference	313
Foundation – Off-canvas JavaScript Reference	315
Foundation – Reveal	317
Foundation – Reveal Basics	318
Foundation – Reveal Modals in Modals	320
Foundation – Reveal Video.....	322
Foundation – Reveal Sizing.....	323
Small Size Modal.....	325
Large Size Modal.....	326
Foundation – Reveal Full-screen	328
Foundation – Reveal No Overlay	329
Foundation – Reveal Animations.....	331
Foundation – Reveal SASS Reference	332
Foundation – Reveal JavaScript Reference.....	334
Foundation – Basic Table.....	336
Foundation – Hover Table	338
Foundation – Stacked Table	340
Foundation – Scroll Table	342
Foundation – Table SASS Reference	345
Foundation – Tabs	347
Foundation – Basic Tabs	348
Foundation – Vertical Tabs	350
Foundation – Tabs SASS Reference	352
Foundation – Tabs JavaScript Reference	354
19. Foundation – Media.....	356
Foundation – Flex Video	356
Foundation – Flex Video Class	357
Foundation – Flex Video Class	358
Foundation – Flex Video SASS Reference	360
Foundation – Media Label	361
Foundation – Label Basics	361
Foundation – Label Multiple IDs.....	362
Foundation – Label Coloring.....	364
Foundation – Label SASS Reference	365
Foundation – Media Orbit	366
Foundation – Media Orbit Basics	366
Foundation – Media SASS Reference	369
Foundation – Media JavaScript Reference	372
Foundation – Progress Bar Basics.....	374
Foundation – Progress Bar Basics.....	375

Foundation – Progress Bar With Text.....	377
Foundation – Progress Bar SASS Reference	378
Foundation – Media Tooltips	379
Foundation – Media Tooltips Basics.....	379
Foundation – Tooltip SASS Reference	381
Foundation – Tooltip JavaScript Reference	382
20. Foundation – Plugins.....	385
Foundation – Plugins Abide.....	385
Foundation – Plugins Abide Demo	386
Foundation – Plugins Abide SASS Reference	389
Foundation – Plugins Abide JavaScript Reference.....	390
Foundation – Plugins Equalizer	393
Foundation – Plugins Basic Equalizer	393
Foundation – Plugins Nesting Equalizer	395
Foundation – Equalizer JavaScript Reference.....	399
Foundation – Plugins Interchange	400
Foundation – Interchange Use with Images.....	401
Foundation – Interchange Use with HTML.....	402
Foundation – Interchange Use with Background Images.....	403
Foundation – Interchange Named Media Queries.....	405
Foundation – Interchange Javascript Reference	406
Foundation – Plugins Toggler	407
Foundation – Plugins Toggle a CSS class.....	407
Foundation – Plugins Toggle with Animation.....	409
Foundation – Plugins Mark as Closable	410
Foundation – Toggler Javascript Reference.....	411
Foundation – Plugins Sticky	412
Foundation – Plugins Sticky Basic.....	413
Foundation – Plugins Sticky Advanced	418
Foundation – Plugins Sticky JavaScript Reference.....	421
FOUNDATION SASS	424
21. Foundation – SASS Functions	425
SASS Reference.....	425
22. Foundation – SASS Mixins	429
Importing.....	429
SASS Reference.....	429
FOUNDATION LIBRARIES	433
23. Foundation – Motion UI.....	434
Installing Motion UI	434
Built-in Transitions.....	434
Foundation – Motion UI Transitions.....	434
Custom Transitions.....	436
Animation	436
Foundation – Reveal Animations.....	436

1. Foundation – Overview

What is Foundation?

Foundation is one of the advanced front-end frameworks for designing beautiful responsive websites. It works on all types of devices and provides you with HTML, CSS and JavaScript plugins.

Foundation began as an internal style guide, built by ZURB in 2008. ZURB is a product design company in Campbell, California. It released Foundation 2.0 as open source in October 2011. The latest version of Foundation is 6.1.1, released in December 2015.

Why Use Foundation?

- It provides faster development by using Sass compiler, which works much faster than default compiler.
- It enriches your website with pricing tables, switches, joyride, range sliders, lightbox and many more.
- It comes with development package like Grunt and Libsass for faster coding and control.
- Foundation for sites provides you with HTML, CSS and JS to quickly build websites.
- Email framework provides you with responsive HTML emails, which can be read on any device.
- Foundation for Apps allows you to build fully responsive web apps.

Features

- It has powerful grid system and some useful UI components and cool JavaScript plugins.
- It provides responsive design, which serves all types of devices.
- It is optimized for mobile devices and truly supports mobile first approach.
- It provides HTML templates, which are customizable and extensible.

Advantages

- It is easy to learn, once you have the basic understanding of HTML and CSS.
- You can use Foundation freely as it is an open source.

- It provides you a bunch of templates, which helps you in start developing the website right away.
- Foundation supports preprocessors like SASS and Compass, which makes development faster.

Disadvantages

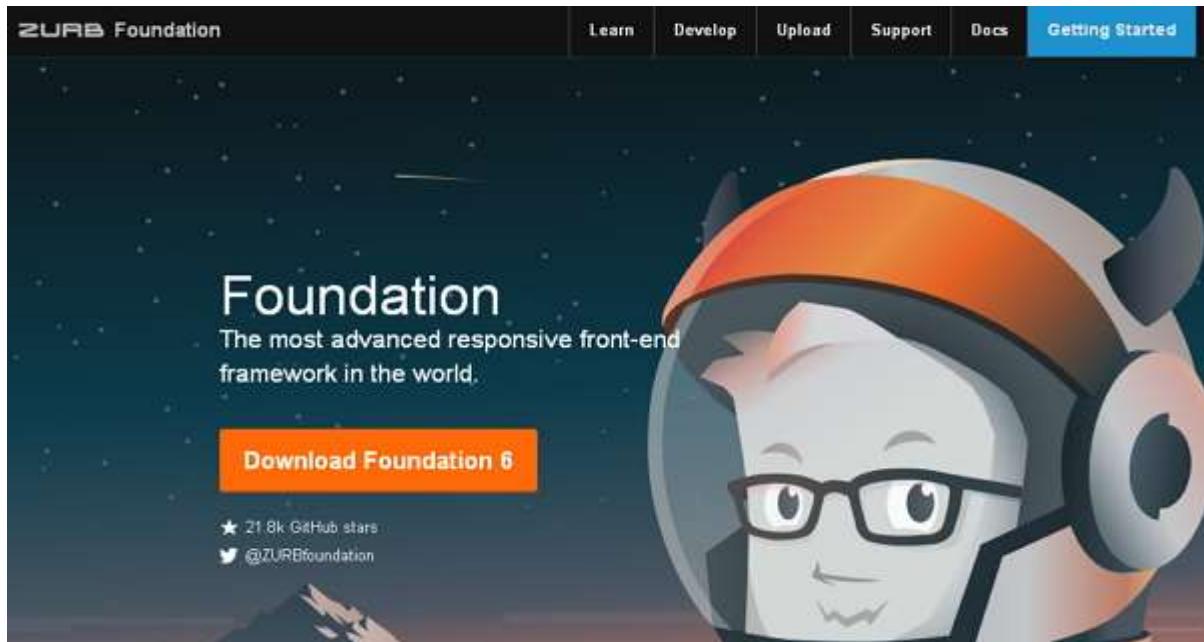
- Because of the popularity of the Twitter Bootstrap, the community support for Twitter Bootstrap is better than Foundation.
- It may take some time for beginners to learn and take advantage of the preprocessor support.
- Lack of wider support like QA sites and forums for fixing issues.
- Foundation has less themes compared to others.

2. Foundation – Installation

In this chapter, we will discuss about how to install and use Foundation on website.

Download Foundation

When you open the link foundation.zurb.com, you will see a screen as shown below -



Click the **Download Foundation 6** button, you will be redirected to another page.

Here you can see four buttons -

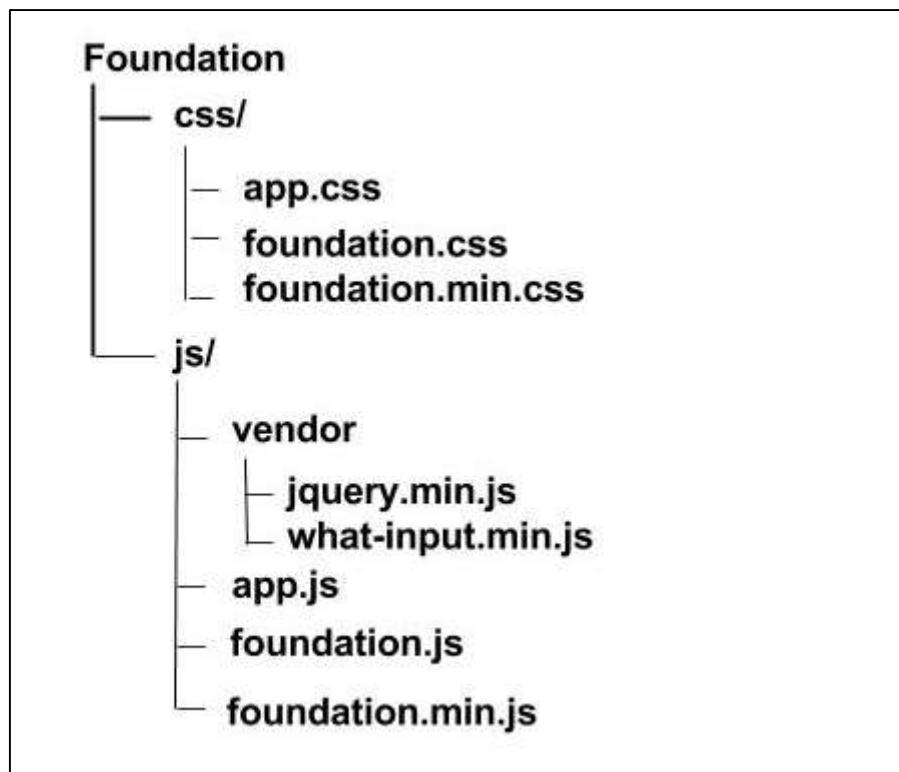
A screenshot of the Foundation 'FOUNDATION FOR SITES' download page. The top navigation bar is identical to the previous screenshot, with the 'Getting Started' tab selected. Below the navigation, the text "FOUNDATION FOR SITES" is centered. A large, bold "Download Foundation" button is prominently displayed. Underneath this, there are four categories: "Complete", "Essential", "Custom", and "Sass". Each category has a brief description and a corresponding orange "Download" button. The "Complete" section says: "Grab this version of Foundation if you want everything in the framework in simple, vanilla CSS and JS...". The "Essential" section says: "A simple, lighter version that includes typography, the grid, buttons, Reveal and Interchange... *59kb (okay, plus dependencies)." The "Custom" section says: "Include or remove certain elements and define the size of columns, colors, font size and more...". The "Sass" section says: "Foundation is built using SCSS, and you can work with it in the same way. Check out the instructions on the Install documentation page.".

- **Download Everything:** You can download this version of Foundation, if you wish to have everything in the framework i.e. vanilla CSS and JS.
- **Download Essentials:** It will download the simple version which includes the grid, buttons, typography etc.
- **Custom Download:** This will download the custom library for Foundation, it includes elements and define size of columns, font size, color etc.
- **Install via SCSS:** This will redirect you to the documentation page to install Foundation for sites.

You can click the **Download Everything** button to get everything in the framework i.e. CSS and JS. As the files consist all things in the framework so every time you don't need to include separate files for individual functionality. At the time of writing this tutorial, the latest version (Foundation 6) was downloaded.

File Structure

Once Foundation is downloaded, extract the ZIP file, and you will see the following file/directory structure-



As you can see, there are compiled CSS and JS (foundation.*), as well as compiled and minified CSS and JS (foundation.min.*).

We are using the CDN versions of the library throughout this tutorial.

HTML Template

A basic HTML template using Foundation is as shown below:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Foundation Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  </head>
  <body>
    <h1>Hello, world!</h1>
  </body>
</html>
```

The following sections describe the above given code in detail.

HTML5 doctype

Foundation consists of certain HTML elements and CSS properties that require the use of the HTML5 doctype. Therefore, the following code for HTML5 doctype should be included at the beginning of all your projects using Foundation.

```
<!DOCTYPE html>
<html>
  ....
</html>
```

Mobile First

It helps to be responsive to mobile devices. You need to include the **viewport meta tag** to the `<head>` element, to ensure proper rendering and touch zooming on mobile devices.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

- *width* property controls the width of the device. Setting it to device-width will make sure that it is rendered across various devices (mobiles, desktops, tablets...) properly.

- *initial-scale = 1.0* ensures that when loaded, your web page will be rendered at a 1:1 scale, and no zooming will be applied out of the box.

Initialization of Components

The jQuery script is required in Foundation for components like modals and dropdown.

```
<script  
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.  
js"></script>  
  
<script  
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m  
in.js"></script>
```

Output

Let us carry out the following steps to see how the above html code works:

- Save above html code in **firstexample.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Hello, world!

3. Foundation – Starter Projects

You can start with your project development with some available templates, which can be installed through Yeti Launch or Foundation CLI. You can start with the new project by using these templates by using *Gulp* build system for the processing of the Sass, JavaScript, copying files etc.

Basic Template

The basic template is something like SASS template, which includes the flat directory structure and compiles only SASS files and it is good to have this simple template while using only SASS. You can use the basic template by using *Yeti Launch* or using *Foundation CLI* using the following command:

```
$ foundation new --framework sites --template basic
```

To set up this, first run the *npm install*, *bower install* and use the *npm start* command to run it. You can also download the template files from the [Github](#).

ZURB Template

It is a combination of CSS/SCSS, JavaScript, Handlebars template, markup structure, image compression and uses SASS processing. You can use the ZURB template by using *Yeti Launch* or using *Foundation CLI* using the below command:

```
$ foundation new --framework sites --template zurb
```

To run this template, follow the same steps specified in the basic template. You can also download the template files from the [Github](#).

Asset Copying

You can copy the content in the *src/assets* folder, using *Gulp* in which *assets* will be your project folder. Important thing here is, the SASS files, JavaScript files and images will not come under this asset copying process, as they will be having their own process to copy the content.

Page Compilation

You can create HTML pages under three folders namely *pages*, *layouts* and *partials* which resides inside the *src/* directory. You can use the Panini flat file compiler that creates layouts for pages by using templates, pages, HTML partials. This process can be done by using Handlebars templating language.

SASS Compilation

You can compile the SASS to CSS using [Libsass](#) and the main SASS file will be stored under `src/assets/scss/app.scss` and also newly created SASS partials will be stored under this folder itself. The output of CSS will be like normal CSS, which is in the nested style. You can compress the CSS with [clean-css](#) and removes the unused CSS from the stylesheet using [UnCSS](#).

JavaScript Compilation

The JavaScript files will be stored under `src/assets/js` folder along with Foundation and all dependencies are tied together into `app.js` file. The files will be tied together as specified in the below order:

- Dependencies of Foundation.
- Files will be stored under `src/assets/js` folder.
- The files are bundled into one file called `app.js`.

Image Compression

By default, all images will be stored under `assets/img` folder under `dist` folder. You can compress the images while building for production using [gulp-imagemin](#) which supports JPEG, PNG, SVG and GIF files.

BrowserSync

You can create a [BrowserSync](#) server which is synchronized browser testing available at `http://localhost:8000` and able to see the compiled templates using this URL. While your server is running, the page refreshes automatically when you save the file and you could see the changes made to the page in real time as you work.

4. Foundation – Kitchen Sink

Description

It includes the Foundation elements to work with the web applications smoothly. The following table lists some of the Foundation components:

S. No.	Components & Description
1	Accordion Accordions contain vertical tabs, which are used on websites to expand and collapse large amount of data.
2	Accordion Menu It displays the collapsible menu with accordion effects.
3	Badge Badges are similar to labels, which are used to highlight the information such as important notes and messages.
4	Breadcrumbs It specifies the current location for a site within navigational hierarchy.
5	Buttons Foundation supports standard buttons with different styles.
6	Callout Callout is an element, which can be used to place the content inside.
7	Close Button It is used to dismiss the alert box.
8	Drilldown Menu Drilldown menu changes the nested lists into vertical drilldown menu.
9	Dropdown Menu Dropdown menu is used for displaying links in a list format.
10	Dropdown Pane Dropdown pane displays the content when you click the button.

11	Flex Video It is used to create video objects in the web pages.
12	Float Classes It is used to add utility classes to HTML elements.
13	Forms It is used to create form layout to collect user input.
14	Label Labels are inline styles, which define a label for an input element.
15	Media Object It is used to add media objects such as images, video, blog comments etc. which can be placed left or right of the content block.
16	Menu It provides access to different modes in the website.
17	Pagination It is a type of navigation that divides the content into a series of related pages.
18	Slider It specifies the range of values by dragging a handle.
19	Switch It is used to switch between on and off state.
20	Table It represents the data in rows and columns format.
21	Tabs It is a navigation based tab that displays the content into different panes without leaving the page.
22	Thumbnail It styles the images in thumbnail shape.
23	Title Bar It is used to display the current screen used by the user with other menu items.

24	Tooltip It is small pop-up box that describes the information when you hover the mouse on the link.
25	Top Bar It is used to create navigation header in the website.
26	Orbit It is an easy and powerful slider that swipes the elements using the <i>orbit</i> class.

Foundation – Accordion

Description

Accordions contain vertical tabs, which are used on websites to expand and collapse large amount of data. It makes the collapsing divisions of the page and content is broken into logical sections.

Example

The following example demonstrates the use of *accordion* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Accordion</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Accordion Example</h2>
  <ul class="accordion" data-accordion role="tablist">
    <li class="accordion-item is-active">
```

```

        <a href="#collapse1" role="tab" class="accordion-title" id="collapse1-
heading">Foundation</a>
        <div id="collapse1" class="accordion-content" role="tabpanel" data-tab-
content aria-labelledby="panel1d-heading">
            The most advanced responsive front-end framework.
        </div>
    </li>
    <li class="accordion-item">
        <a href="#collapse2" role="tab" class="accordion-title" id="collapse2-
heading">Bootstrap</a>
        <div id="collapse2" class="accordion-content" role="tabpanel" data-tab-
content aria-labelledby="panel1d-heading">
            Bootstrap is the most popular and powerful front-end framework.
        </div>
    </li>
    <li class="accordion-item">
        <a href="#collapse3" role="tab" class="accordion-title" id="collapse3-
heading">HTML</a>
        <div id="collapse3" class="accordion-content" role="tabpanel" data-tab-
content aria-labelledby="panel1d-heading">
            HTML is a markup language for describing web documents.
        </div>
    </li>
</ul>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above html code works:

- Save the above given html code in **accordion.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Accordion Example

[Foundation](#)

The most advanced responsive front-end framework.

[Bootstrap](#)

Foundation – Accordion Menu

Description

It displays the collapsible menu with accordion effects and provides support for auto collapse.

Example

The following example demonstrates the use of *accordion menu* in Foundation:

```
<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Accordion Menu</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
    <h2>Accordion Menu Example</h2>
    <ul class="vertical menu" data-accordion-menu>
        <li><a href="#">Two Wheeler</a>
            <ul class="menu vertical nested is-active">
                <li><a href="#">Honda</a>
```

```

<ul class="menu vertical nested">
    <li><a href="#">CB Shine</a></li>
    <li><a href="#">Unicorn</a></li>
    <li><a href="#">Dream Yuga</a></li>
</ul>
</li>
<li><a href="#">TVS</a></li>
<li><a href="#">Bajaj</a></li>
</ul>
</li>
<li><a href="#">Four Wheeler</a>
    <ul class="menu vertical nested">
        <li><a href="#">Mahindra</a></li>
        <li><a href="#">Ford</a></li>
    </ul>
</li>
<li><a href="#">Heavy Weight Vehicle</a></li>
</ul>

<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in the **accordion_menu.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Accordion Menu Example

Two Wheeler

Honda

TVS

Bajaj

Four Wheeler

Heavy Weight Vehicle

Foundation – Badge

Description

Badges are similar to labels, which are used to highlight the information such as important notes, messages and display with the rounded corners.

Example

The following example demonstrates the use of *badge* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Badge</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Badge Example</h2>
  <a href="">Link<span class="secondary badge">A</span></a>

```

```

<span class="success badge">B</span>
<span class="alert badge">1</span>
<span class="warning badge">2</span>
<span class="dark badge">3</span>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **badge.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Badge Example

[Link](#) A B 1 2 3

Foundation – Breadcrumbs

Description

It specifies the current location for a site within the navigational hierarchy. It is useful when accessing a large number of the pages of a website.

Example

The following example demonstrates the use of *breadcrumbs* in Foundation:

```

<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Breadcrumbs</title>
    <link rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
        s">

```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Breadcrumbs Example</h2>
<ul class="breadcrumbs">
<li><a href="#">Home</a></li>
<li><a href="#">About Us</a></li>
<li class="disabled">Services</li>
<li>
    <span class="show-for-sr">Contact: </span> Contact Us
</li>
</ul>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **breadcrumbs.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Breadcrumbs Example

HOME / ABOUT US / SERVICES / CONTACT US

Foundation – Button

Description

Foundation supports the standard buttons with different styles. Use the *button* classes with button element.

Example

The following example demonstrates the use of *buttons* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Button</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Button Example</h2>
  <button type="button" class="success button">Login</button>
  <button type="button" class="alert button">Logout</button>
  <button type="button" class="warning button">Registration</button><br>

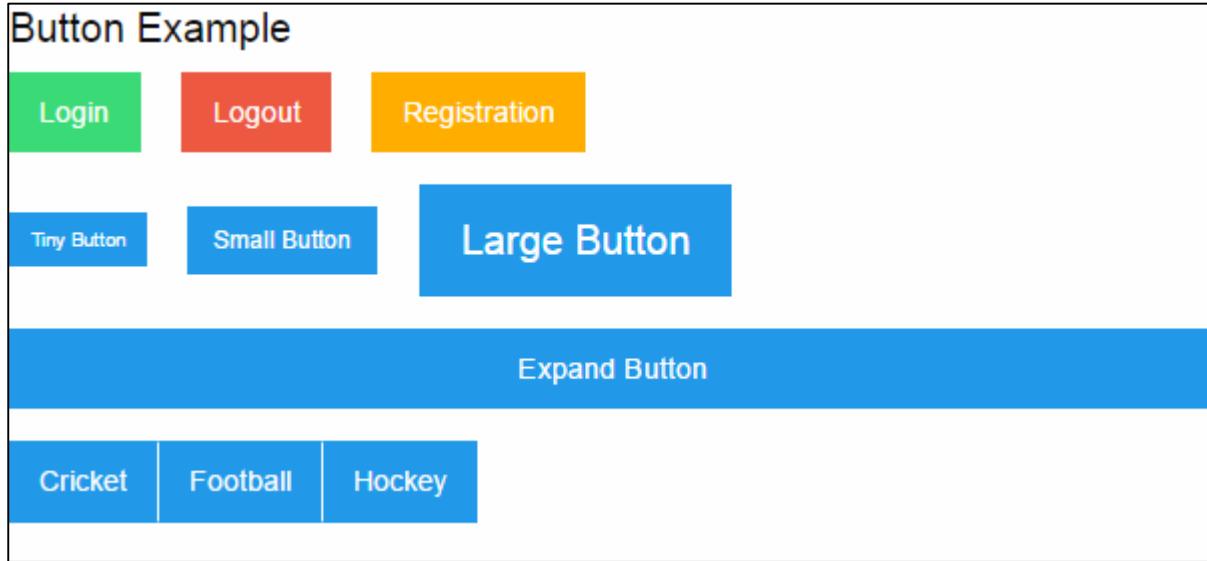
  <a class="tiny button" href="#">Tiny Button</a>
  <a class="small button" href="#">Small Button</a>
  <a class="large button" href="#">Large Button</a>
  <a class="expanded button" href="#">Expand Button</a>

  <div class="button-group">
    <a class="button">Cricket</a>
    <a class="button">Football</a>
    <a class="button">Hockey</a>
  </div>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **button.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Callout

Description

Callout is an element, which can be used to place the content inside.

Example

The following example demonstrates the use of *Callout* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Callout</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
```

```

</head>
<body>
<h2>Callout Example</h2>
<div class="callout">
  <h5>callout.</h5>
  <a href="#">Foundation is a responsive front-end framework.</a>
</div>

<div class="callout secondary">
  <h5>secondary callout</h5>
  <a href="#">Foundation is a responsive front-end framework.</a>
</div>

<div class="callout success">
  <h5>success callout</h5>
  <a href="#">Foundation is a responsive front-end framework.</a>
</div>

<div class="callout warning">
  <h5>warning callout</h5>
  <a href="#">Foundation is a responsive front-end framework.</a>
</div>

<div class="callout alert">
  <h5>alert callout</h5>
  <a href="#">Foundation is a responsive front-end framework.</a>
</div>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **callout.html** file.
- Open this HTML file in a browser. An output is displayed as shown below.

Callout Example

callout.

Foundation is a responsive front-end framework.

secondary callout

Foundation is a responsive front-end framework.

success callout

Foundation is a responsive front-end framework.

warning callout

Foundation is a responsive front-end framework.

alert callout

Foundation is a responsive front-end framework.

Foundation – Close Button

Description

The Close button is used to dismiss the alert box.

Example

The following example demonstrates the use of *close button* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Close Button</title>
```

```

<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Close Button Example</h2>
<div class="callout">
  <button class="close-button" aria-label="Close alert" type="button">
    <span aria-hidden="true">x</span>
  </button>
  <p>Example of static close button.</p>
</div>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **close_button.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Close Button Example

Example of static close button.



Foundation – Drilldown Menu

Description

Drilldown menu is a standard menu syntax that changes the nested lists into vertical drilldown menu.

Example

The following example demonstrates the use of *drilldown menu* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Drilldown Menu</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Drilldown Menu Example</h2>
  <ul class="vertical menu" data-drilldown style="width: 200px">
    <li>
      <a href="#">Cars</a>
      <ul class="vertical menu">
        <li>
          <a href="#">Mahindra</a>
          <ul class="vertical menu">
            <li><a href="#">Xylo</a></li>
            <li><a href="#">Scorpio</a></li>
            <li><a href="#">Bolero</a></li>
            <li><a href="#">Quanto</a></li>
          </ul>
        </li>
        <li><a href="#">Ford</a></li>
        <li><a href="#">Skoda</a></li>
        <li><a href="#">Toyota</a></li>
        <li><a href="#">Renault</a></li>
      </ul>
    </li>
  </ul>
</body>
```

```

</li>
<li>
  <a href="#">Sports</a>
  <ul class="vertical menu">
    <li><a href="#">Cricket</a></li>
    <li><a href="#">Football</a></li>
    <li><a href="#">Baseball</a></li>
    <li><a href="#">Hockey</a></li>
  </ul>
</li>
<li>
  <a href="#">Country</a>
  <ul class="vertical menu">
    <li><a href="#">India</a></li>
    <li><a href="#">Srilanka</a></li>
    <li><a href="#">Australia</a></li>
    <li><a href="#">England</a></li>
  </ul>
</li>
</ul>
<script>
$(document).ready(function() {
  $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **drilldown_menu.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Drilldown Menu Example

- Cars ►
- Sports ►
- Country ►

Foundation – Dropdown Menu

Description

The Dropdown menu is used for displaying the links in a list format.

Example

The following example demonstrates the use of *dropdown menu* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Dropdown Pane</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Dropdown Menu Example</h2>
  <ul class="dropdown menu" data-dropdown-menu>
    <li>
      <a>Cars</a>
      <ul class="menu">
        <li><a href="#">Ford</a></li>
```

```

<li>
    <a href="#">Mahindra</a>
    <ul class='menu'>
        <li><a href="#">Xylo</a></li>
        <li><a href="#">Scorpio</a></li>
        <li>
            <a href="#">XUV</a>
            <ul class='menu'>
                <li><a href="#">W4</a></li>
                <li><a href="#">W6</a></li>
            </ul>
        </li>
    </ul>
</li>
<li><a href="#">Skoda</a></li>
</ul>
</li>

<a href="#">Sports</a>
<ul class="menu">
    <li><a href="#">Cricket</a></li>
    <li><a href="#">Football</a></li>
</ul>
</li>

<li><a href="#">Country</a></li>
</ul>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **dropdown_menu.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Dropdown Menu Example

Cars ▾ Sports ▾ Country

Foundation – Dropdown Pane

Description

The Dropdown pane displays the content when you click the button.

Example

The following example demonstrates the use of *dropdown pane* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Dropdown Pane</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Dropdown Pane Example</h2>
  <button class="button" type="button" data-toggle="dropdown_toggle">Dropdown
  Pane</button>
  <div class="dropdown-pane" id="dropdown_toggle" data-dropdown>
    Foundation is a responsive front-end framework.
  </div>
  <script>
```

```

$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

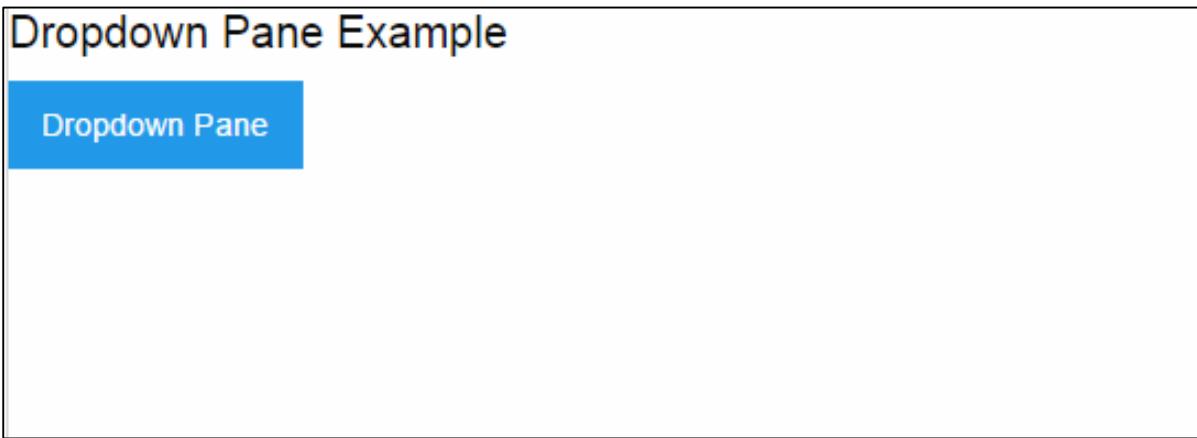
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **dropdown_pane.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Dropdown Pane Example



Dropdown Pane

Foundation – Flex Video

Description

The Flex Video is used to create video objects in the web pages by using *flex-video* class in the *div* tag.

Example

The following example demonstrates the use of *flex video* in Foundation:

```

<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Flex Video</title>
    <link rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
        s">

```

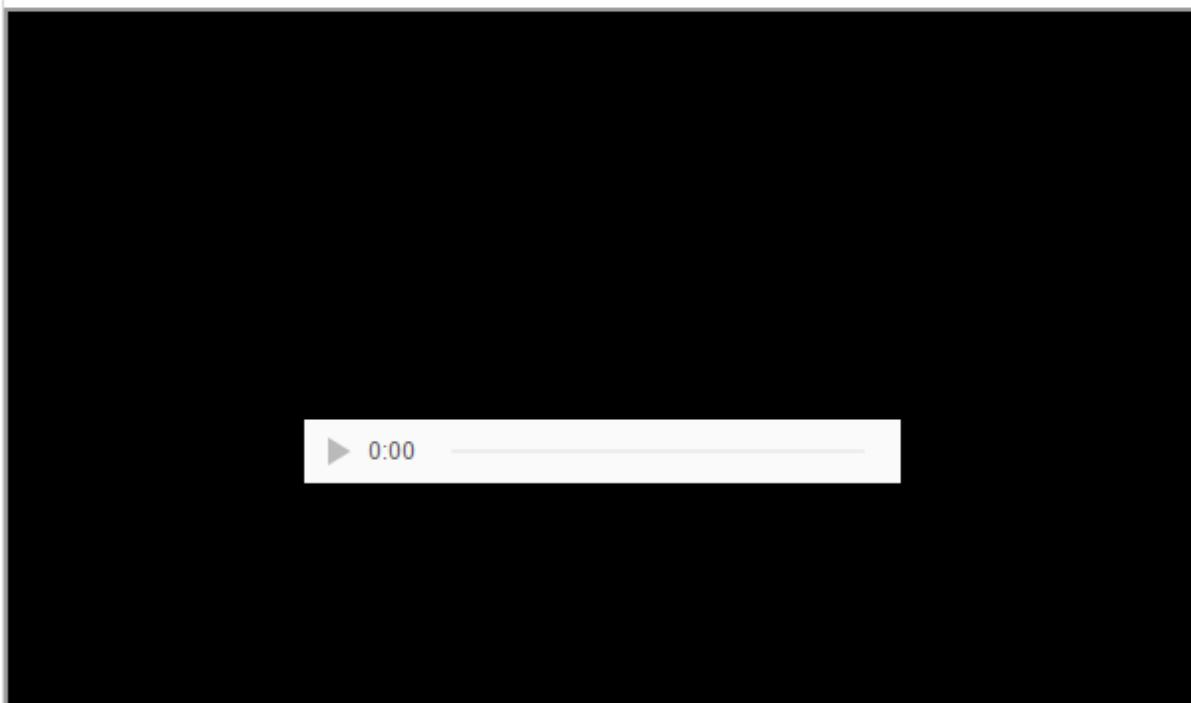
```
<script  
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m  
in.js"></script>  
  
<script  
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.  
js"></script>  
  
</head>  
  
<body>  
  
<h2>Flex Video Example</h2>  
  
<div class="flex-video">  
  
    <iframe width="200" height="200"  
src="/foundation/images/kitchen_sink/video.mp4"></iframe>  
  
</div>  
  
</body>  
  
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save above given html code in **flex_video.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Flex Video Example



Foundation – Float Classes

Description

Float classes are used to add utility classes to the HTML elements by adding `.clearfix` class to the parent element.

Example

The following example demonstrates the use of *float classes* in Foundation:

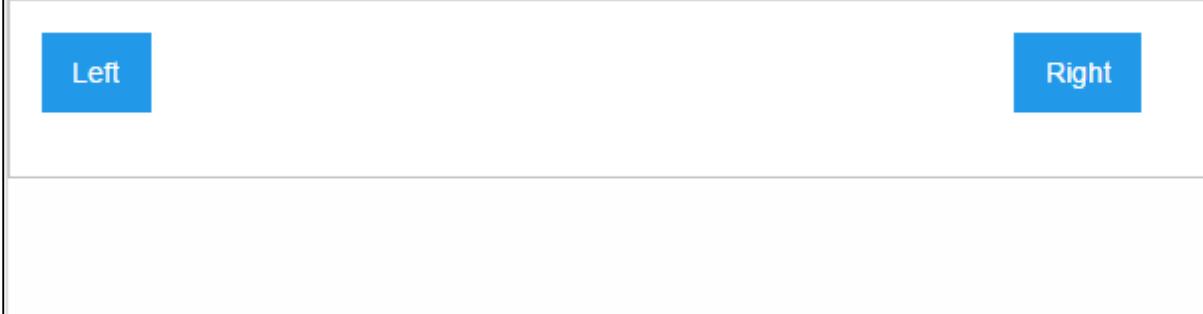
```
<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Float Classes</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
    <h2>Float Classes Example</h2>
    <div class="callout clearfix">
        <a class="button float-left">Left</a>
        <a class="button float-right">Right</a>
    </div>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **float_classes.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Float Classes Example



Foundation – Forms

Description

Form is used to create a form layout to collect user input.

Example

The following example demonstrates the use of *forms* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Forms</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Forms Example</h2>
  <form>
    <label>Input Label
      <input type="text" placeholder="enter your text">
    </label>
    <p class="help-text">This is help text..</p>
  </form>
</body>

```

```

<label>
  <textarea placeholder="This is text area"></textarea>
</label>
<label>Select Menu
  <select>
    <option value="india">India</option>
    <option value="newzealand">New Zealand</option>
    <option value="china">China</option>
    <option value="japan">Japan</option>
  </select>
</label>
<div class="row">
  <fieldset class="small-6 columns">
    <legend>Your Favorite Fruit?</legend>
    <input type="radio" name="fruit" value="Apple"><label for="fruitApp">Apple</label>
    <input type="radio" name="fruit" value="Grapes"><label for="fruitGrp">Grapes</label>
  </fieldset>
  <fieldset class="small-6 columns">
    <legend>Choose your car?</legend>
    <input id="checkbox1" type="checkbox"><label for="checkbox1">Scorpio</label>
    <input id="checkbox2" type="checkbox"><label for="checkbox2">XUV</label>
    <input id="checkbox3" type="checkbox"><label for="checkbox3">Xylo</label>
  </fieldset>
</div>
<div class="input-group">
  <span class="input-group-label">Amnt</span>
  <input class="input-group-field" type="url">
  <a class="input-group-button button">Submit</a>
</div>
</form>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **forms.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Forms Example

Input Label

enter your text

This is help text..

This is text area

Select Menu

India ▾

Your Favorite Fruit?	Choose your car?
<input checked="" type="radio"/> Apple <input type="radio"/> Grapes	<input type="checkbox"/> Scorpio <input type="checkbox"/> XUV <input type="checkbox"/> Xylo
Amnt	Submit

Foundation – Labels

Description

Labels are inline styles, which define labels for an input element.

Example

The following example demonstrates the use of *labels* in Foundation:

```
<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Labels</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
    s">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
    in.js"></script>
```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
js"></script>
</head>
<body>
<h2>Labels Example</h2>
<span class="secondary label">Label-Secondary</span>
<span class="success label">Label-Success</span>
<span class="alert label">Label-Alert</span>
<span class="warning label">Label-Warning</span>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **label.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Media Object

Description

Media Object is used to add media objects such as images, videos, blog comments etc., which can be placed left or right of the content block.

Example

The following example demonstrates the use of *media object* in Foundation:

```

<!doctype html>
<head>
<meta charset="utf-8" />

```

```

<meta http-equiv="x-ua-compatible" content="ie=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Media Object</title>
<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Media Object Example</h2>
<div class="media-object">
  <div class="media-object-section middle">
    <div class="thumbnail">
      <img src= "/foundation/images/kitchen_sink/winter.jpg">
    </div>
  </div>
  <div class="media-object-section">
    <h4>Media Object</h4>
    <p>Media objects are super useful components for displaying an item, usually an image, alongside some content, usually text.</p>
  </div>
</div>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **media_object.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Media Object Example

Media Object



Media objects are super useful components for displaying an item, usually an image, alongside some content, usually text.

Foundation – Menu

Description

The Menu provides access to different modes in the website.

Example

The following example demonstrates the use of *menu* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Menu</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Menu Example</h2>
  <ul class="menu">
    <li><a href="#">Home</a></li>
    <li><a href="#">About Us</a></li>
    <li><a href="#">Services</a></li>
    <li><a href="#">Contact Us</a></li>
  </ul>
```

```
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **foundtn_menu.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Menu Example

Home About Us Services Contact Us

Foundation – Pagination

Description

Pagination is a type of navigation that divides the content into series of related pages.

Example

The following example demonstrates the use of *pagination* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Pagination</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
in.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
js"></script>
</head>
```

```

<body>
<h2>Pagination Example</h2>
<ul class="pagination">
  <li class="disabled">Previous</li>
  <li class="current">1</li>
  <li><a href="#">2</a></li>
  <li><a href="#">3</a></li>
  <li><a href="#">4</a></li>
  <li class="ellipsis"></li>
  <li><a href="#">12</a></li>
  <li><a href="#">13</a></li>
  <li><a href="#">Next</a></li>
</ul>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **pagination.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Pagination Example

[Previous](#) [Next](#)

Foundation – Slider

Description

The Slider allows specifying the range value by dragging a handle.

Example

The following example demonstrates the use of *slider* in Foundation:

```

<!doctype html>
<head>

```

```

<meta charset="utf-8" />
<meta http-equiv="x-ua-compatible" content="ie=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Slider</title>
<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Slider Example</h2>
<div class="slider" data-slider data-initial-start="30" data-end="150">
  <span class="slider-handle" data-slider-handle role="slider"></span>
  <span class="slider-fill" data-slider-fill></span>
  <input type="hidden">
</div><br>

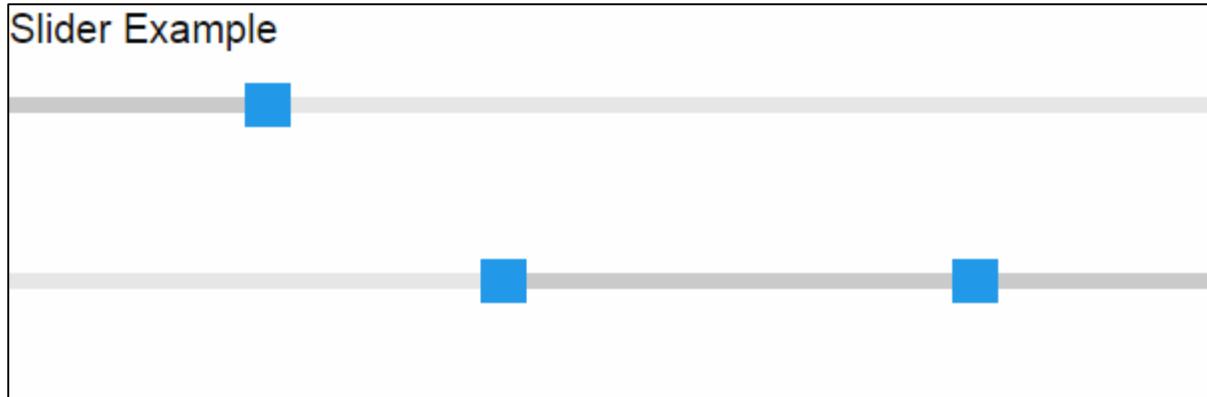
<div class="slider" data-slider data-initial-start="40" data-initial-end="80">
  <span class="slider-handle" data-slider-handle role="slider"></span>
  <span class="slider-fill" data-slider-fill></span>
  <span class="slider-handle" data-slider-handle role="slider"></span>
</div>
<script>
  $(document).ready(function() {
    $(document).foundation();
  })
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **slider.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Switch

Description

It is used to switch between on and off state.

Example

The following example demonstrates the use of *switch* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Switch</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Switch Example</h2>
  <div class="switch tiny">
    <input class="switch-input" id="switch_tiny" type="checkbox">
    <label class="switch-paddle" for="switch_tiny"></label>
  </div>
</body>
```

40

```

</div>

<div class="switch small">
  <input class="switch-input" id="switch_small" type="checkbox">
  <label class="switch-paddle" for="switch_small"></label>
</div>

<div class="switch large">
  <input class="switch-input" id="switch_large" type="checkbox">
  <label class="switch-paddle" for="switch_large"></label>
</div>
<script>
  $(document).ready(function() {
    $(document).foundation();
  })
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **switch.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Switch Example



Foundation – Table

Description

The Table represents the data in rows and columns format.

Example

The following example demonstrates the use of *table* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Table</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Table Example</h2>
<table>
  <thead>
    <tr>
      <th>Team</th>
      <th>Player Name</th>
      <th>Batsman/Bowler</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>India</td>
      <td>Sachin Tendulkar</td>
      <td>Batsman</td>
    </tr>
    <tr>
      <td>South Africa</td>
      <td>Shaun Pollock</td>
      <td>Bowler</td>
    </tr>
    <tr>

```

```

<td>West Indies</td>
<td>Brian Lara</td>
<td>Batsman</td>
</tr>
</tbody>
</table>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **table.html** file.
- Open this HTML file in a browser, an output is displayed as shown below .

Table Example

Team	Player Name	Batsman/Bowler
India	Sachin Tendulkar	Batsman
South Africa	Shaun Pollock	Bowler
West Indies	Brian Lara	Batsman

Foundation – Tabs

Description

It is a navigation-based tab that displays the content into different panes without leaving the page.

Example

The following example demonstrates the use of *tabs* in Foundation:

```

<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Tabs</title>

```

```

<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Tabs Example</h2>
<ul class="tabs" data-tabs id="tabs_example">
  <li class="tabs-title is-active"><a href="#tab1">Player 1</a></li>
  <li class="tabs-title"><a href="#tab2">Player 2</a></li>
  <li class="tabs-title"><a href="#tab3">Player 3</a></li>
  <li class="tabs-title"><a href="#tab4">Player 4</a></li>
  <li class="tabs-title"><a href="#tab5">Player 5</a></li>
</ul>

<div class="tabs-content" data-tabs-content="tabs_example">
  <div class="tabs-panel is-active" id="tab1">
    <p>First Player</p>
    <p>Sachin Tendulkar</p>
  </div>
  <div class="tabs-panel" id="tab2">
    <p>Second Player</p>
    <p>M S Dhoni</p>
  </div>
  <div class="tabs-panel" id="tab3">
    <p>Third Player</p>
    <p>Shane Warne</p>
  </div>
  <div class="tabs-panel" id="tab4">
    <p>Fourth Player</p>
    <p>Shaun Pollock</p>
  </div>
  <div class="tabs-panel" id="tab5">
    <p>Five Player</p>
    <p>Adam Gilchrist</p>
  </div>
</div>

```

```

</div>
<script>
    $(document).ready(function() {
        $(document).foundation();
    })
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **tabs.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Tabs Example

Player 1 Player 2 Player 3 Player 4 Player 5

First Player

Sachin Tendulkar

Foundation – Thumbnail

Description

Thumbnail styles the images in a thumbnail shape.

Example

The following example demonstrates the use of *thumbnail* in Foundation:

```

<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Thumbnail</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
    s">

```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Thumbnail Example</h2>
<div class="row">
  <div class="small-4 columns">
    <img class="thumbnail" src= "/foundation/images/kitchen_sink/winter.jpg"
alt="Winter">
  </div>
  <div class="small-4 columns">
    
  </div>
  <div class="small-4 columns">
    
  </div>
</div>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **thumbnail.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Thumbnail Example



Foundation – Title Bar

Description

Title Bar can be used to display the menu items on either side as a title bar.

Example

The following example demonstrates the use of *title bar* in Foundation:

```
<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Title Bar</title>
    <link rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
        src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
        src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
    <h2>Title Bar Example</h2>
    <div class="title-bar">
        <div class="title-bar-left">
            <span class="title-bar-title">Title Bar One</span>
        </div>
        <div class="title-bar-right">
            <span class="title-bar-title">Title Bar Two</span>
        </div>
    </div>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **title_bar.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Title Bar Example

Title Bar One

Title Bar Two

Foundation – Tooltip

Description

Tooltip is a small pop-up box that describes the information when you hover the mouse on the link.

Example

The following example demonstrates the use of *tooltip* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Tooltip</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Tooltip Example</h2>
  <p>The <span data-tooltip class="has-tip" title="It is small pop-up box">Tooltips</span> are a quick way to provide extended information on a term or action on a page. It is small pop-up box that describes the information when you hover the mouse on the link.</p>
  <script>
    $(document).ready(function() {
```

```

$(document).foundation();
}
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **tooltip.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Tooltip Example

The **Tooltips** are a quick way to provide extended information on a term or action on a page. It is small pop-up box that describes the information when you hover the mouse on the link.

Foundation – Top Bar

Description

Top Bar is used to create navigation header in the website.

Example

The following example demonstrates the use of *top bar* in Foundation:

```

<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Top Bar</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
    s">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
    in.js"></script>

```

```

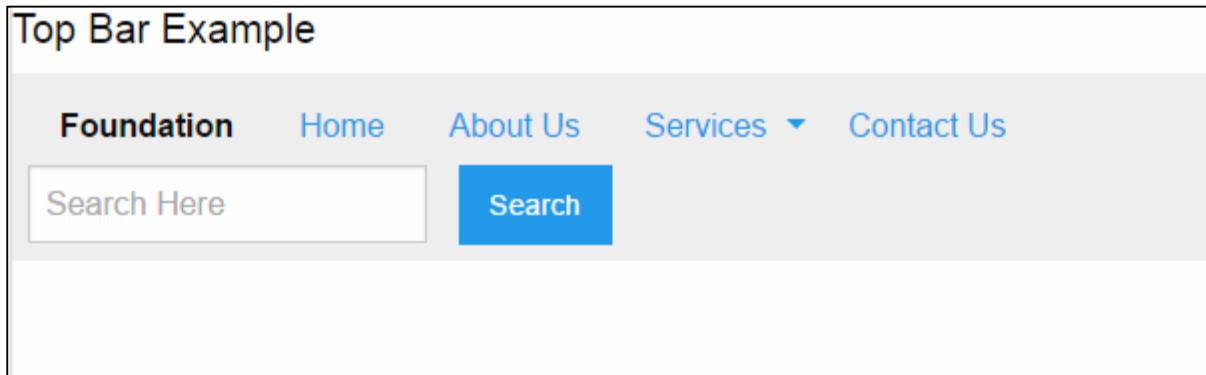
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Top Bar Example</h2>
<div class="top-bar">
  <div class="top-bar-left">
    <ul class="dropdown menu" data-dropdown-menu>
      <li class="menu-text">Foundation</li>
      <li><a href="#">Home</a></li>
      <li><a href="#">About Us</a></li>
      <li class="has-submenu">
        <a href="#">Services</a>
        <ul class="submenu menu vertical" data-submenu>
          <li><a href="#">Service1</a></li>
          <li><a href="#">Service2</a></li>
          <li><a href="#">Service3</a></li>
        </ul>
      </li>
      <li><a href="#">Contact Us</a></li>
    </ul>
  </div>
  <div class="top-bar-right">
    <ul class="menu">
      <li><input type="search" placeholder="Search Here"></li>
      <li><button type="button" class="button">Search</button></li>
    </ul>
  </div>
</div>
<script>
$(document).ready(function() {
  $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **top_bar.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Orbit

Description

Orbit is an easy and powerful slider that swipes the elements using the *orbit* class.

Example

The following example demonstrates the use of *orbit* in Foundation:

```
<html>
<head>
<title>Orbit</title>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/motion-ui/1.1.1/motion-ui.css">
<script src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2 class="text-center">Orbit Example</h2>
<div class="orbit" role="region" data-orbit data-use-m-u-i="false">
<ul class="orbit-container">
<button class="orbit-previous" aria-label="previous"><span class="show-for-sr">Previous Slide</span>◀</button>
```

```

<button class="orbit-next" aria-label="next"><span class="show-for-sr">Next
Slide</span>></button>

<li class="is-active orbit-slide">
  <div>
    <h3 class="text-center">This is your first slider</h3>
    <p class="text-center">Foundation is a family of responsive front-end
frameworks that make it easy to design beautiful responsive websites, apps and
emails that look amazing on any device. </p>
  </div>
</li>

<li class="orbit-slide">
  <div>
    <h3 class="text-center">This is your second slider</h3>
    <p class="text-center">Foundation is a family of responsive front-end
frameworks that make it easy to design beautiful responsive websites, apps and
emails that look amazing on any device. </p>
  </div>
</li>

<li class="orbit-slide">
  <div>
    <h3 class="text-center">This is your third slider</h3>
    <p class="text-center">Foundation is a family of responsive front-end
frameworks that make it easy to design beautiful responsive websites, apps and
emails that look amazing on any device. </p>
  </div>
</li>

<li class="orbit-slide">
  <div>
    <h3 class="text-center">This is your fourth slider</h3>
    <p class="text-center">Foundation is a family of responsive front-end
frameworks that make it easy to design beautiful responsive websites, apps and
emails that look amazing on any device. </p>
  </div>
</li>
</ul>

<nav class="orbit-bullets">
  <button class="is-active" data-slide="0"><span class="show-for-sr">First
slide</span><span class="show-for-sr">Current Slide</span></button>
  <button data-slide="1"><span class="show-for-sr">Second slide</span></button>
  <button data-slide="2"><span class="show-for-sr">Third slide</span></button>
  <button data-slide="3"><span class="show-for-sr">Fourth slide</span></button>
</nav>
```

```
</div>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **orbit.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Orbit Example

This is your first slider

Foundation is a family of responsive front-end frameworks that make it easy to design beautiful responsive websites, apps and emails that look amazing on any device.



Foundation General

5. Foundation – Global Styles

In this chapter, we will study about *Global Styles*. The global CSS of Foundation framework includes useful resets that makes sure styling is consistent across browsers.

Font Sizing

The font size of the browser style sheet is set to 100% by default. The default font size is set to 16 pixels. Depending on font size, grid size is calculated. To have distinct base font size and unaffected grid breakpoints, set \$rem-base to \$global-font-size value, which must be in pixels.

Colors

Interactive elements like *links* and *buttons* use default shades of blue, which comes from SASS variable `$primary-color`. Components can also have colors such as: *secondary*, *alert*, *success* and *warning*. For more information check [here](#).

SASS Reference

Variables

The following table lists the SASS variables, which are used to customize the default styles of components in your project `_settings.scss`.

S. No.	Name & Description	Type	Default Value
1	\$global-width It represents the site's global width. Used to determine the grid's row width.	Number	rem-calc(1200)
2	\$global-font-size It represents the font size applied to <code><html></code> and <code><body></code> . It is set 100% by default and the user's browser settings value will be inherited.	Number	100%
3	\$global-lineheight It represents all types of default line height. <code>\$global-lineheight</code> is 24px while <code>\$global-font-size</code> set to 16px.	Number	1.5
4	\$primary-color It gives color to the interactive components such as links and buttons.	Color	#2199e8

5	\$secondary-color It is used with components, which support <code>.secondary</code> class.	Color	#777
6	\$success-color It represents the positive status or action when used with <code>.success</code> class.	Color	#3adb76
7	\$warning-color It represents a caution status or action when used with <code>.warning</code> class.	Color	#ffae00
8	\$alert-color It represents a negative status or action when used with <code>.alert</code> class.	Color	#ec5840
9	\$light-gray It is used for light gray UI items.	Color	#e6e6e6
10	\$medium-gray It is used for medium gray UI items.	Color	#cacaca
11	\$dark-gray It is used for dark gray UI items.	Color	#8a8a8a
12	\$black It is used for black UI items.	Color	#0a0a0a
13	\$white It is used for white UI items.	Color	#fefefe
14	\$body-background It represents the background color of the body.	Color	\$white
15	\$body-font-color It represents the text color of the body.	Color	\$black
16	\$body-font-family It represents the list of fonts of the body.	List	'Helvetica Neue', Helvetica, Roboto, Arial, sans-serif
17	\$body-antialiased Antialiased type is enabled by setting this attribute to <code>true</code> using the CSS properties - <code>webkit-font-smoothing</code> and <code>-moz-osx-font-smoothing</code> .	Boolean	true

18	\$global-margin It represents the global margin value on components.	Number	1rem
19	\$global-padding It represents global padding value on components.	Number	1rem
20	\$global-margin It represents global margin value used between components.	Number	1rem
21	\$global-weight-normal It represents global font weight for normal type.	Keyword or Number	normal
22	\$global-weight-bold It represents global font weight for bold type.	Keyword or Number	bold
23	\$global-radius It represents the global value of all elements which has a border radius.	Number	0
24	\$global-text-direction It sets the text direction of the CSS to <i>ltr</i> or <i>rtl</i>		ltr

6. Foundation – SASS

SASS helps to make the code more flexible and customizable in Foundation.

Compatibility

To install SASS based version for foundation, Ruby should be installed on Windows. Foundation can be compiled with Ruby SASS and libsass. We recommend **node-sass 3.4.2+** version to compile SASS.

Autoprefixer Required

Autoprefixer handles SASS files. *gulp-autoprefixer* is used to build the process. The following autoprefixer setting is used to get proper browser support.

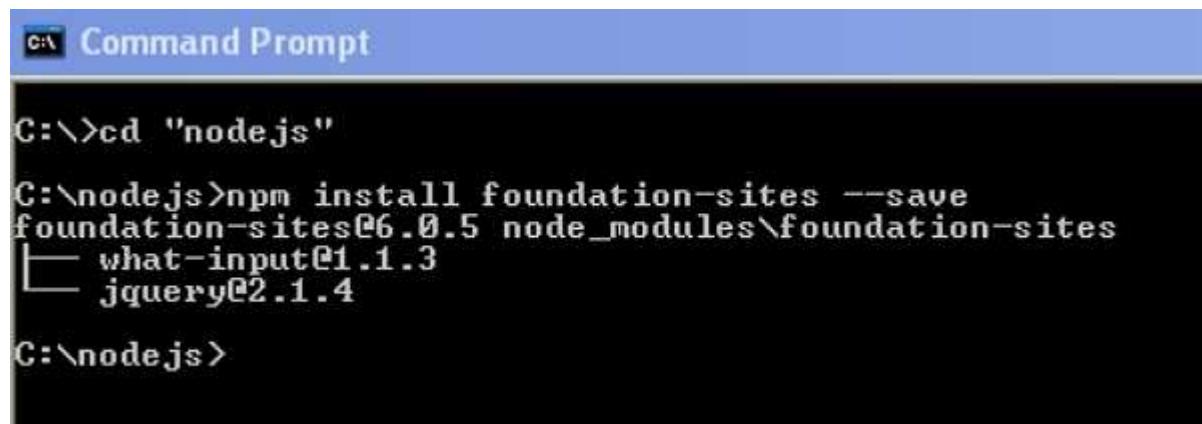
```
autoprefixer({
  browsers: ['last 2 versions', 'ie >= 9', 'and_chr >= 2.3']
});
```

Loading the Framework

We can install the framework files using NPM. Using command line interface (CLI), we can compile the Sass files. Following is the command to load the framework:

```
npm install foundation-sites --save
```

After running the above given command line code, you will get the following lines -



```
C:\>cd "nodejs"
C:\nodejs>npm install foundation-sites --save
foundation-sites@6.0.5 node_modules\foundation-sites
└── what-input@1.1.3
    └── jquery@2.1.4
C:\nodejs>
```

Compiling Manually

Framework files can be added as an import path depending upon on your build process, but the path will be the same - *packages_folder/foundation-sites/scss*. The *@import* statement is included at the top of the *foundation-sites.scss* file. The next line in the given code is explained in the **Adjusting CSS Output** section.

```
@import 'foundation';
@include foundation-everything;
```

Using Compiled CSS

You can include the pre-compiled CSS files. There are two types of CSS files, i.e. minified and unminified. Minified version is used for production and the unminified version is used to directly edit the framework CSS.

```
<link rel="stylesheet" href="node_modules/foundation-sites/dist/foundation-sites.css">
<link rel="stylesheet" href="node_modules/foundation-sites/dist/foundation-sites.min.css">
```

Adjusting CSS Output

For various components, Foundation output consist of many classes. It is used to control the CSS output of the framework. Add the following single line of code to include all the components at once.

```
@include foundation-everything;
```

Following is the list of the components imported when you write the above code in your scss file. The components, which are not necessary, can be commented. You can view the below given code lines in the *Your_folder_name/node_modules/foundation-sites/scss/foundation.scss* file.

```
@import 'foundation';

@import 'grid/grid';
@import 'typography/typography';
@import 'forms/forms';
@import 'components/visibility';
@import 'components/float';
@import 'components/button';
@import 'components/button-group';
@import 'components/accordion-menu';
@import 'components/accordion';
@import 'components/badge';
@import 'components/breadcrumbs';
@import 'components/callout';
@import 'components/close-button';
@import 'components/drilldown';
...
```

```
...
...
//so on....
```

The Settings File

A settings file is included in the entire foundation project i.e. `_settings.scss`. If you are using Yeti Launch or the CLI to create a Foundation for Sites project, you can find the settings file under `src/assets/scss/`.

We have installed Foundation using npm, hence you can find the settings file included under `your_folder_name/node_modules/foundation-sites/scss/settings/_settings.scss`. You can move this into your own Sass files to work with.

You can write your own CSS, if you are not able to customize with variables. Following is a set of variables, which change the default styling of buttons.

```
$button-padding: 0.85em 1em;
$button-margin: 0 $global-margin $global-margin 0;
$button-fill: solid;
$button-background: $primary-color;
$button-background-hover: scale-color($button-background, $lightness: -15%);
$button-color: #fff;
$button-color-alt: #000;
$button-radius: $global-radius;
$button-sizes: (
  tiny: 0.6rem,
  small: 0.75rem,
  default: 0.9rem,
  large: 1.25rem,
);
$button-opacity-disabled: 0.25;
```

7. Foundation – JavaScript

In this chapter, we will study about *JavaScript*. It is easy to set up *JavaScript* in Foundation; only thing you require is *jQuery*.

JavaScript Installing

Description

You can use ZIP download, package manager, or CDN to get Foundation *JavaScript* file. In your code you can provide links to *jQuery* and Foundation as *<script>* tags, placed before the closing *</body>* and check that Foundation is loaded after *jQuery*.

Example

The following example demonstrates the use of *JavaScript* in Foundation.

```
<html>
<head>
    <title>JavaScript Example</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
</head>
<body>
    <h2>Hello World!!!</h2>
    <button class="button" type="button" data-toggle="dropdown_toggle">Click Me</button>
    <div class="dropdown-pane" id="dropdown_toggle" data-dropdown>
        Welcome to TutorialsPoint
    </div>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script>
        $(document).ready(function() {
            $(document).foundation();
        })
    </script>
```

```
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **foundation_javascript_installation.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Hello World!!!

Click Me

File Structure

When you install Foundation through command line, Foundation plugins downloads as individual files such as *foundation.tabs.js*, *foundation.dropdownMenu.js*, *foundation.slider.js* and so on. All these files are combined into *foundation.js*, which provides all the plugins at one time. If you wish to use some plugin, first *foundation.core.js* should be loaded.

For instance:

```
<script src="js/jquery.min.js"></script>
<script src="js/foundation.core.js"></script>
<script src="js/foundation.tabs.js"></script>
```

Certain plugins may require particular utility libraries, which come with Foundation installation. You can study in detail about specific plugin requirements in the next chapter JavaScript Utilities.

Loading individual files creates network overhead, specifically for mobile users. For faster page loading, use of grunt or gulp is recommended.

Initializing

The *foundation()* function is used to initialize all the Foundation plugin at one time.

For instance:

```
(document).foundation();
```

Using Plugins

Description

Using data attributes, plugins are connected to HTML elements as they match the plugins' name. A single HTML element can have only one plugin at a time, although the majority of the plugins can be nested within other ones. For instance, tooltip link is created by adding *data-tooltip*.

Example

The following example demonstrates the use of connecting HTML elements to plugins in Foundation.

```
<html>
  <head>
    <title>JavaScript Example</title>
    <link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  </head>
  <body>
    <p><span data-tooltip aria-haspopup="true" class="has-tip" data-disable-hover='false' tabindex=1 title="Advanced Front-End Framework.">Foundation</span>
    has powerful grid system and some useful UI components and cool JavaScript
    plugins.</p>
    <script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script>
      $(document).ready(function() {
        $(document).foundation();
      })
    </script>
  </body>
</html>
```

Output

63

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **javascript_using_plugins.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Foundation has powerful grid system and some useful UI components and cool JavaScript plugins.

Configuring Plugins

Description

Plugins can be customized by using its configuration settings. For instance, you can set the speed of the *accordion* slides up and down. The plugin settings can be globally changed using the plugin's **DEFAULTS** property.

```
Foundation.Accordion.defaults.slideSpeed = 600;
Foundation.Accordion.defaults.multiExpand = true;
```

The plugins can be set with different settings by changing the HTML settings or in JavaScript. Every setting in HTML is defined as an individual data attribute.

Example

The following example demonstrates how to configure plugin in Foundation.

```
<html>
<head>
    <title>JavaScript Configuring Plugins</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
</head>
<body>
    <ul class="accordion" data-accordion data-slide-speed="600" role="tablist">
        <li class="accordion-item is-active">
            <a href="#collapse1" role="tab" class="accordion-title" id="collapse1-
heading">Foundation</a>
```

```

<div id="collapse1" class="accordion-content" role="tabpanel" data-tab-
content aria-labelledby="panel1d-heading">
    The most advanced responsive front-end framework.
</div>
</li>
<li class="accordion-item">
    <a href="#collapse2" role="tab" class="accordion-title" id="collapse2-
heading">CSS</a>
    <div id="collapse2" class="accordion-content" role="tabpanel" data-tab-
content aria-labelledby="panel1d-heading">
        CSS describes the presentation of an HTML documents.
    </div>
</li>
<li class="accordion-item">
    <a href="#collapse3" role="tab" class="accordion-title" id="collapse3-
heading">HTML</a>
    <div id="collapse3" class="accordion-content" role="tabpanel" data-tab-
content aria-labelledby="panel1d-heading">
        HTML is a markup language for describing web documents.
    </div>
</li>
</ul>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
in.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
js"></script>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given in **javascript_configuring_plugins.html** file.

- Open this HTML file in a browser, an output is displayed as shown below.

Adding Plugins after Page Load

When new HTML is added to the DOM, any of the plugins on those elements will not be initialized by default. You can check for new plugins by re-calling the `.foundation()` function.

For instance:

```
$.ajax('assets/partials/kitten-carousel.html', function(data) {
  $('#kitten-carousel').html(data).foundation();
});
```

Programmatic Use

In JavaScript, plugins can be created programmatically and each plugin is a global `Foundation` object's class, with a constructor which takes two parameters such as an element and an object.

```
var $accordion = new Foundation.Accordion($('#accordion'), {
  slideSpeed: 600,
  multiExpand: true
});
```

Majority of the plugins are provided with public API, which lets you manipulate it via JavaScript. You can look through the documentations of plugin to study the available functions and methods can be invoked easily.

For instance:

```
$('.tooltip').foundation('destroy'); // this will destroy all Tooltips on the page.
```

```
$('#reveal').foundation('open'); // this will open a Reveal modal with id `reveal`.
```

```
$('[data-tabs]').eq(0).foundation('selectTab', $('#example')); // this will change the first Tabs on the page to whatever panel you choose.
```

- You are allowed to choose any jQuery selector and if the selector holds multiple plugins, then they all will have identical chosen method called.
- Arguments can be passed just like passing arguments to JavaScript.
- Methods that are prefixed with *underscore(_)* are considered as a portion of internal API, meaning, that without warning they can break, change or even disappear.

Events

Whenever a specific function finishes, DOM triggers an event. For instance, whenever tabs are changed, it can be listened and create a return response to it. Each plugin can trigger list of events, which will be documented in plugin's documentation. In Foundation 6, callback plugins are removed and must be taken as event listeners.

For instance:

```
$('[data-tabs]').on('change.zf.tabs', function() {
  console.log('Tabs are changed!');
});
```

8. Foundation – JavaScript Utilities

Foundation includes JavaScript utilities that are used to add common functionalities. It is very helpful and easy to use. This JavaScript utilities library can be found in the folder `Your_folder_name/node_modules/foundation-sites/js`

Box

- `Foundation.Box` library consists of a couple of methods.
- The **js/foundation.util.box.js** is the script filename, which can be included while writing the code.
- Either jQuery objects or plain JavaScript elements can be pass to both methods.

```
var dims = Foundation.Box.GetDimensions(element);
```

The returned object specifies the dimension of the `element` as:

```
{  
  height: 54,  
  width: 521,  
  offset: {  
    left: 198,  
    top: 1047  
  },  
  parentDims: {  
    height: ... //The same format is share for parentDims and windowDims as the  
    element dimensions.  
  },  
  windowDims: {  
    height: ...  
  }  
}
```

- Function `ImNotTouchingYou` is included.
- Based on the passed element, a Boolean value is returned, which is either a conflict with edge of the window or optional or a parent element.

- The two options specified in the line given below i.e. leftAndRightOnly, topAndBottomOnly is used to identify the collision on only one axis.

```
var clear = Foundation.Box.ImNotTouchingYou(element [, parent, leftAndRightOnly,
topAndBottomOnly]);
```

Keyboard

- There are many methods in *Foundation.Keyboard*, which helps to make the keyboard event interaction easy.
- The **js/foundation.util.keyboard.js** is the script filename, which can be included while writing the code.
- The object *Foundation.Keyboard.keys* consist key/value pairs, which keys are used in the framework more frequently.
- Whenever the key is pressed then *Foundation.Keyboard.parseKey* is called to get a string. This helps to manage your own keyboard inputs.

The following code is used to find all focusable elements within the given *\$element*. Therefore, there is no need of writing any function and selector by you.

```
var focusable = Foundation.Keyboard.findFocusable($('#content'));
```

- The *handleKey* function is a main function of this library.
- This method is used to handle the keyboard event; it can be called whenever any plugin is registered with the utility.

```
Foundation.Keyboard.register('pluginName', {
  'TAB': 'next'
});
...//in event callback
Foundation.Keyboard.handleKey(event, 'pluginName', {
  next: function(){
    //do stuff
  }
});
```

The *Foundation.Keyboard.register* function can be called when you want to use your own key bindings.

MediaQuery

- *MediaQuery* library is a backbone of all responsive CSS technique.
- The **js/foundation.util.mediaQuery.js** is the script filename, which can be included while writing the code.
- The *Foundation.MediaQuery.atLeast('large')* is used to check if the screen is at least as wide as a breakpoint.
- The *Foundation.MediaQuery.get('medium')* gets the media query of a breakpoint.
- The *Foundation.MediaQuery.queries* are an array of media queries, Foundation uses for breakpoints.
- *Foundation.MediaQuery.current* is a string of the current breakpoint size.

```
Foundation.MediaQuery.get('medium');
Foundation.MediaQuery.atLeast('large');
Foundation.MediaQuery.queries;
Foundation.MediaQuery.current;
```

The following code broadcasts the media query change on the window.

```
$(window).on('changed.zf.mediaquery', function(event, newSize, oldSize){});
```

Motion & Move

- *Foundation.Motion* javascript is similar to Motion UI library, which is included in the Foundation 6. It is used to create custom CSS transitions and animations.
- The **js/foundation.util.motion.js** is the script filename, which can be included while writing the code.
- *Foundation.Move* is used to make CSS3 backed animation simple and elegant.
- **requestAnimationFrame();** method tells the browser to perform an animation; it requests that your animation function be called before the browser performs the next repaint.

```
Foundation.Move(durationInMS, $element, function(){
  //animation logic
});
```

When the animation is completed, *finished.zf.animate* is fired.

Timer & Images Loaded

Orbit uses both, the function timer and the image loaded. The **js/foundation.util.timerAndImageLoader.js** is the script filename, which can be included while writing the code.

```
var timer = new Foundation.Timer($element, {duration: ms, infinite: bool},  
callback);
```

The image-loaded method runs a callback function in your jQuery collection when images are completely loaded.

```
Foundation.onImagesLoaded($images, callback);
```

Touch

- The methods are used for adding pseudo drag events and swipe to elements.
- The **js/foundation.util.touch.js** is the script filename, which can be included while writing the code.
- The *addTouch* method is used to bind elements to touch events in the Slider plugin for mobile devices.
- The *spotSwipe* method binds the elements to swipe events in the Orbit plugin for mobile devices.

```
$('.selector').addTouch().on('mousemove', handleDrag);  
$('.selector').spotSwipe().on('swipeleft', handleLeftSwipe);
```

Triggers

- It triggers the specified event for the selected elements.
- The **js/foundation.util.triggers.js** is the script filename, which can be included while writing the code.
- The triggers are utilized in many Foundation plugin.

```
$('.selector').on('open.zf.trigger', handleOpen);  
$('.selector').on('close.zf.trigger', handleClose);  
$('.selector').on('toggle.zf.trigger', handleToggle);
```

The following two methods are used in this library i.e. resize and scroll.

- The *resize()* method triggers the resize event when a resize event occurs.
- The *scroll()* method triggers the scroll event when a scroll event occurs.

```
$('#someId').on('scrollme.zf.trigger', handleScroll);
$('#someId').on('resizeme.zf.trigger', handleResize);
```

Miscellaneous

- Foundation contains few features in the core library, which are used in many places.
- The **js/foundation.core.js** is the script filename, which can be included while writing the code.
- *Foundation.GetYoDigits([number, namespace])* returns a random base-36 uid with namespacing. It returns the string length of 6 characters long by default.
- *Foundation.getFnName(fn)* returns a JavaScript function name.
- *Foundation.transitionend* occurs when CSS transition is completed.

9. Foundation – Media Queries

Media queries are CSS3 module that include media features such as width, height, color and displays the content as per the specified screen resolution.

Foundation uses the following media queries to create breakdown ranges:

- **Small:** Used for any screen.
- **Medium:** It is used for screens of 640 pixels and wider.
- **Large:** It is used for screens of 1024 pixels and wider.

You can change the screen size by using the *breakpoint classes*. For instance, you can use *.small-6* class for small sized screens and *.medium-4* class for medium sized screens as shown in the following code snippet:

```
<div class="row">
  <div class="small-6 medium-4 columns"></div>
  <div class="small-6 medium-8 columns"></div>
</div>
```

Changing the Breakpoints

You can change the breakpoints, if your application uses SASS version of Foundation. You can place the breakpoints name under the *\$breakpoints* variable in the settings file as shown below:

```
$breakpoints: (
  small: 0px,
  medium: 640px,
  large: 1024px,
  xlarge: 1200px,
  xxlarge: 1440px,
);
```

You can change the breakpoints classes in the settings file by modifying the *\$breakpoint-classes* variable. If you want to use *.large* class in the CSS, then add it to the end of the list as shown below:

```
$breakpoints-classes: (small medium large);
```

Suppose, you want to use *.xlarge* class in the CSS, and then add this class to the end of the list as shown below:

```
$breakpoints-classes: (small medium large xlarge);
```

SASS

The Breakpoint Mixin

- You can write the media queries by using *breakpoint()* mixin along with *@include*.
- Use the *down* or *only* keywords along with the breakpoint value to change the behavior of the media query as shown in the following code format:

```
class_name {
    // code for medium screens and smaller
    @include breakpoint(medium down) { }
    // code for medium screens only
    @include breakpoint(medium only) { }
}
```

You can use three media queries- *portrait*, *landscape* and *retina* for device orientation or pixel density and they are not width based media queries.

Breakpoint Function

- You can use the functionality of *breakpoint()* mixin by using the internal function.
- The *breakpoint()* functionality can be used directly to write own media queries:

```
@media screen and #{breakpoint(medium)} {
    // code for medium screens and up styles
}
```

JavaScript

Working with Media Queries

The Foundation JavaScript provides the *MediaQuery.current* function to access current breakpoint name on the *Foundation.MediaQuery* object as specified below:

```
Foundation.MediaQuery.current
```

- The *MediaQuery.current* function displays *small*, *medium*, *large* as current breakpoint names.
- You can get the media query of breakpoint using the *MediaQuery.getfunction* as shown below:

```
Foundation.MediaQuery.get('small')
```

SASS Reference

Variables

The following table lists the SASS variables, which can be used to customize the default styles of the component:

S. No.	Name & Description	Type	Default Value
1	\$breakpoints It is a breakpoint name which can be used to write the media queries by using <code>breakpoint()</code> mixin.	Map	small: 0px medium: 640px large: 1024px xl: 1200px xxl: 1440px
2	\$breakpoint-classes You can change the CSS class output by modifying the <code>\$breakpoint-classes</code> variable.	List	small medium large

Mixins

Mixins creates a group of styles to build your CSS class structure for the Foundation components.

BREAKPOINT

It uses `breakpoint()` mixin to create media queries and includes the following activities:

- If string is passed, then mixin searches the string in the `$breakpoints` map and creates the media query.
- If you are using pixel value, then convert it to em value using `$rem-base`.
- If rem value is passed, then it changes its unit to em.
- If you are using em value, then it can be used as it is.

The following table specifies the parameter used by the breakpoint:

S. No.	Name & Description	Type	Default Value
1	\$value It processes the values by using breakpoint name, px, rem or em values.	keyword or number	None

Functions

BREAKPOINT

It uses `breakpoint()` mixin to create media queries with matching input value.

The following table specifies the possible input value used by the breakpoint:

S. No.	Name & Description	Type	Default Value
1	\$val It processes the values by using breakpoint name, px, rem or em values.	keyword or number	small

JavaScript Reference

Functions

There are two types of functions:

- **.atLeast**: It checks the screen. It must be wide at least as a breakpoint.
- **.get**: It is used to get the media query of the breakpoint.

The following table specifies the parameter used by above functions:

S. No.	Name & Description	Type
1	size It checks and gets the name of the breakpoint for the specified functions respectively.	String

10. Foundation – The Grid

Description

Foundation grid system scales up to 12 columns through the page. Grid systems are used to create page layouts through a series of rows and columns that house your content.

Grid Options

The following table tells briefly about how the Foundation grid system works in multiple devices.

	Small devices Phones(<640px)	Medium devices Tablets(>=640px)	Large devices Laptops & Desktops(>=1200px)
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints
Class prefix	.small-*	.medium-*	.large-*
Number of columns	12	12	12
Nestable	Yes	Yes	Yes
Offsets	Yes	Yes	Yes
Column ordering	Yes	Yes	Yes

Basic Structure of a Foundation Grid

Following is the basic structure of a Foundation grid:

```
<div class="row">
  <div class="small-*"></div>
  <div class="medium-*"></div>
  <div class="large-*"></div>
</div>
```

77

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

- First, create a *row* class to create horizontal groups of columns.
- Content should be placed within the columns, and only columns may be the immediate children of rows.
- Grid columns are created by specifying the number of twelve available columns you wish to span. For example, for four equal columns we would use *.large-3*

Following are the three classes used in the Foundation grid system:

S. No.	Basic Grid Classes & Description
1	Large The <i>large-*</i> class is used for the large devices.
2	Medium The <i>medium-*</i> class is used for the medium devices.
3	Small <i>small-*</i> class is used for the small devices.

Foundation – Large Grid

Description

The *large* grid displays easily on the large device i.e. 50%/50% but in small devices, it gets completely filled and displays vertically. The *large-** class is used for the large devices.

Example

The following example demonstrates the use of *large* class in Foundation.

```
<!DOCTYPE html>
<html>
<head>
    <title>Foundation Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
</head>
<body>
<h2>Example of Large Grid</h2>
<h4>Divided row in 2 columns</h4>
<div class="row">
  <div class="large-6 columns" style="background-color:#C0B0F0;">
    Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.
  </div>

  <div class="large-6 columns" style="background-color:#8BD6EE;">
    Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.
  </div>
</div>

<h4>Divided row in 3 columns</h4>
<div class="row">
  <div class="large-4 columns" style="background-color:#8BD6EE;">
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
  </div>
  <div class="large-4 columns" style="background-color:#C0B0F0;">
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
  </div>
  <div class="large-4 columns" style="background-color:#8BD6EE;">
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
  </div>
</div>

<h4>Divided row in 4 columns</h4>

```

```
<div class="row">
  <div class="large-3 columns" style="background-color:#C0B0F0;">
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
  </div>
  <div class="large-3 columns" style="background-color:#8BD6EE;">
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
  </div>
  <div class="large-3 columns" style="background-color:#C0B0F0;">
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
  </div>
  <div class="large-3 columns" style="background-color:#8BD6EE;">
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
  </div>
</div>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **largeExample.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Large Grid

Divided row in 2 columns

 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
 Lorem Ipsum has been the industry's standard dummy text ever since the
 1500s, when an unknown printer took a galley of type and scrambled it to make
 a type specimen book.

 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
 Lorem Ipsum has been the industry's standard dummy text ever since the
 1500s, when an unknown printer took a galley of type and scrambled it to make
 a type specimen book.

Divided row in 3 columns

 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
 Lorem Ipsum is simply dummy text of the printing and typesetting industry.

Divided row in 4 columns

 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
 Lorem Ipsum is simply dummy text of the printing and typesetting industry.
 Lorem Ipsum is simply dummy text of the printing and typesetting industry.

Foundation – Medium Grid

Description

The *medium-** class is used for the medium devices. The medium size device can inherit the styles from small.

Example

The following example demonstrates the use of *medium* class in Foundation:

```
<!DOCTYPE html>
<html>
<head>
  <title>Foundation Template</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
```

```

<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
</head>
<body>
<h2>Example of Large Grid</h2>
<h4>Divided row in 2 columns</h4>
<div class="row">
  <div class="medium-6 columns" style="background-color:#C0B0F0;">
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
    Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
    when an unknown printer took a galley of type and scrambled it to make a type
    specimen book.
  </div>

  <div class="medium-6 columns" style="background-color:#8BD6EE;">
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
    Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
    when an unknown printer took a galley of type and scrambled it to make a type
    specimen book.
  </div>
</div>

<h4>Divided row in 3 columns</h4>
<div class="row">
  <div class="medium-4 columns" style="background-color:#8BD6EE;">
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
    Lorem Ipsum has been the industry's standard dummy text.
  </div>
  <div class="medium-4 columns" style="background-color:#C0B0F0;">
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
  </div>
  <div class="medium-4 columns" style="background-color:#8BD6EE;">
    Lorem Ipsum is simply dummy text of the printing.
  </div>
</div>

```

```
</div>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **mediumgrid.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of medium Grid

Divided row in 2 columns

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Divided row in 3 columns

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text.

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

Lorem Ipsum is simply dummy text of the printing.

Foundation – Small Grid

Description

The *small* grid displays easily on large devices. *small-** class is used for the small devices.

Example

The following example demonstrates the use of *small* class in Foundation:

```
<!DOCTYPE html>
<html>
<head>
```

```

<title>Foundation Template</title>
<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
</head>
<body>
<h2>Example of Small Grid</h2>
<h4>Divided row in 2 columns</h4>
<div class="row">
  <div class="small-6 columns" style="background-color:#C0B0F0;">
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
    Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
    when an unknown printer took a galley of type and scrambled it to make a type
    specimen book.
  </div>
  <div class="small-6 columns" style="background-color:#8BD6EE;">
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
    Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
    when an unknown printer took a galley of type and scrambled it to make a type
    specimen book.
  </div>
</div>

<h4>Divided row in 4 columns</h4>
<div class="row">
  <div class="small-4 columns" style="background-color:#8BD6EE;">
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
    Lorem Ipsum has been the industry's standard dummy text.
  </div>
  <div class="small-3 columns" style="background-color:#C0B0F0;">
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
  </div>
  <div class="small-2 columns" style="background-color:#8BD6EE;">

```

```

    Lorem Ipsum is simply dummy text of the printing.

</div>

<div class="small-3 columns" style="background-color:#C0B0F0;">
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
</div>

</div>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **smallgrid.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of small Grid

Divided row in 2 columns

Lore m Ip su m is si mply du mmu y te xt of the prin ting and type settin g indu stry. Lore m Ip su m has been the industry's stan dard du mmu y te xt ever since the 1500s, when an unkno wn prin ter took a galley of type and scram bled it to make a type specimen book.	Lore m Ip su m is si mply du mmu y te xt of the prin ting and type settin g indu stry. Lore m Ip su m has been the industry's stan dard du mmu y te xt ever since the 1500s, when an unkno wn prin ter took a galley of type and scram bled it to make a type specimen book.
--	--

Divided row in 4 columns

Lore m Ip su m is si mply du mmu y te xt of the prin ting and type settin g indu stry. Lore m Ip su m has been the industry's stan dard du mmu y te xt ever since the 1500s, when an unkno wn prin ter took a galley of type and scram bled it to make a type specimen book.	Lore m Ip su m is si mply du mmu y te xt of the prin ting and type settin g indu stry. Lore m Ip su m has been the industry's stan dard du mmu y te xt ever since the 1500s, when an unkno wn prin ter took a galley of type and scram bled it to make a type specimen book.	Lore m Ip su m is si mply du mmu y te xt of the prin ting and type settin g indu stry. Lore m Ip su m has been the industry's stan dard du mmu y te xt ever since the 1500s, when an unkno wn prin ter took a galley of type and scram bled it to make a type specimen book.	Lore m Ip su m is si mply du mmu y te xt of the prin ting and type settin g indu stry. Lore m Ip su m has been the industry's stan dard du mmu y te xt ever since the 1500s, when an unkno wn prin ter took a galley of type and scram bled it to make a type specimen book.
--	--	--	--

Advanced Grid

Following are the advanced grid formats used in Foundation.

S. No.	Advanced Grids & Description
1	Combined Column/Row The <i>column</i> and <i>row</i> classes are used on the same element to get the full width column to use as a container.
2	Nesting We can nest the grid columns inside another columns.
3	Offsets Using <i>large-offset-*</i> or <i>small-offset-*</i> class, you can move the columns to the right.
4	Incomplete Rows Foundation floats the last element automatically to the right when the rows do not include columns up to 12.
5	Collapse/Uncollapse Rows Using the media query size, the collapse and uncollapse classes are included to the row element to show the paddings.
6	Centered Columns By including the class <i>small-centered</i> in the column, you can make the column at the center.
7	Source Ordering Source ordering class is used to shift the columns between the breakpoint.
8	Block Grids Block-grid is used to split the content.

Foundation – Combined Column/Row

Description

The *column* and *row* classes are used on the same element to get the full width column to use as container.

Example

The following example demonstrates the use of *combined column/row* class in Foundation:

```
<!DOCTYPE html>
<html>
<head>
  <title>Foundation Template</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
</head>
<body>
<h3>Example of combined column and row</h3>
<div class="column row" style="background-color:#8BD6EE;">
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
    Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
    when an unknown printer took a galley of type and scrambled it to make a type
    specimen book.
</div>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **combined_row_column.html** file.
 - Open this HTML file in a browser, an output is displayed as shown below.

Example of combined column and row

Foundation – Nesting

Description

We can nest the grids columns inside columns. In one column, we can define many more columns inside it.

Example

The following example demonstrates the use of *nesting* in Foundation:

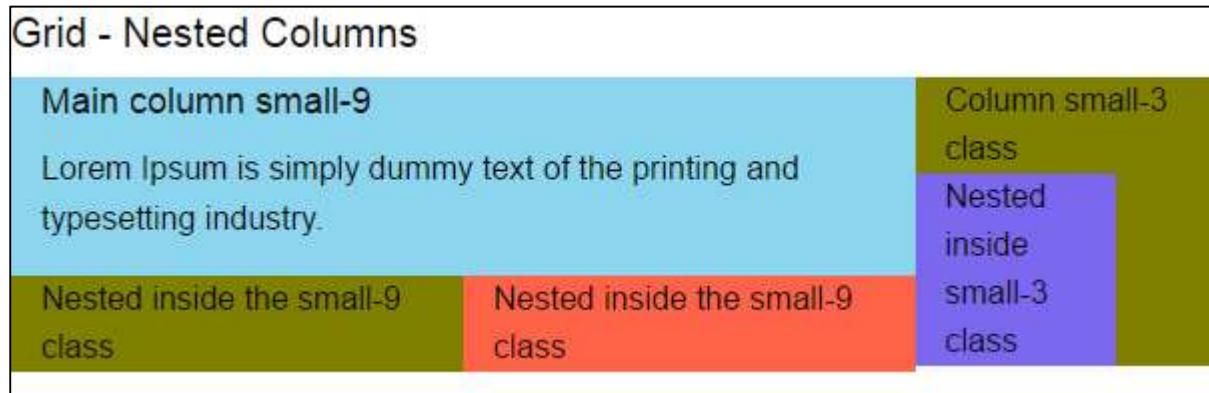
```
<!DOCTYPE html>
<html>
<head>
    <title>Foundation Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
</head>
<body>
<div class="row">
<h2>Grid - Nested Columns</h2>
<div class="small-9 columns" style="background-color:#8BD6EE;">
    <h4>Main column small-9</h4>
    <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry.</p>
    <div class="row">
        <div class="small-6 columns" style="background-color:#808000;">Nested
        inside the small-9 class</div>
        <div class="small-6 columns" style="background-color:#FF6347;">Nested
        inside the small-9 class </div>
    </div>
</div>
<div class="small-3 columns" style="background-color:#808000;">Column small-3
class
    <div class="row">
        <div class="small-8 columns" style="background-color:#7B68EE;">Nested
        inside small-3 class</div>
    </div>
</div>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **nested.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Offsets

Description

Using large-offset-* or small-offset-* class, you can move the columns to the right. The left margin of a column is increased by the number specified while defining the class in *. For example, large-offset-4.

Example

The following example demonstrates the use of *offsets* in Foundation:

```
<!DOCTYPE html>
<html>
<head>
    <title>Foundation Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
</head>
<body>
    <h2>Grid - Offsets</h2>
```

```

<div class="row">
  <div class="large-2 columns" style="background-color:#808000;">Two
  column</div>
  <div class="large-10 columns" style="background-color:#7B68EE;">10
  columns</div>
</div>
<div class="row">
  <div class="large-2 columns" style="background-color:#8BD6EE;">Two
  columns</div>
  <div class="large-8 small-offset-2 columns" style="background-
  color:#C0B0F0;">8 column, offset 2 column</div>
</div>
<div class="row">
  <div class="large-3 columns" style="background-color:#808000;">3
  columns</div>
  <div class="large-5 small-offset-4 columns" style="background-
  color:#8BD6EE;">5 columns, offset 4 column</div>
</div>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **offsets.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Grid - Offsets
Two column
10 columns
Two columns
8 column, offset 2 column
3 columns
5 columns, offset 4 column

Foundation – Incomplete Rows

Description

Foundation floats the last element automatically to the right when the rows do not include columns up to 12. You can include `.end` class to the last element to make the element float to the left.

Example

The following example demonstrates the use of `incomplete-rows` in Foundation:

```
<!DOCTYPE html>
<html>
<head>
<title>Foundation Template</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
</head>
<body>
<h2>Incomplete rows example</h2>
<div class="row">
<div class="small-4 columns" style="background-color:#808000;">small-4
columns</div>
<div class="small-4 columns" style="background-color:#8BD6EE;">small-4
columns</div>
</div>
<div class="row">
<div class="small-4 columns" style="background-color:#8BD6EE;">small-4
columns</div>
<div class="small-4 columns end" style="background-color:#808000;">small-4
columns end</div>
</div>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **incomplete_rows.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Incomplete rows example

```
small-4 columns          small-4 columns
small-4 columns          small-4 columns end
```

Foundation – Collapse/Uncollapse Rows

Description

Using the media-query size, include the collapse and uncollapse class to the row element to show the paddings.

Example

The following example demonstrates the use of *Collapse/Uncollapse Rows* in the Foundation. We have included the collapse class for medium media query and uncollapse for large media query.

```
<!DOCTYPE html>
<html>
<head>
    <title>Foundation Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
in.js"></script>
</head>
<body>
<h2>Collapse/Uncollapse Row</h2>
<div class="row medium-collapse large-uncollapse">
    <div class="small-6 columns" style="background-color:#8BD6EE;">
        Collapse and uncollapse row
    </div>
    <div class="small-6 columns" style="background-color:#C0B0F0;">
```

```
Collapse and uncollapse row
</div>
</div>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **collapse_row.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Collapse/Uncollapse Row

Collapse and uncollapse row Collapse and uncollapse row

Foundation – Centered Columns

Description

Including the class *small-centered* to column, you can make the column at the center. The *large-centered* class is used to center the column in large devices. To uncenter the columns in large devices or screen then include *large-uncenter* class.

Example

The following example demonstrates the use of *Centered Columns* in Foundation:

```
<!DOCTYPE html>
<html>
<head>
  <title>Foundation Template</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
</head>
<body>
<h2>Centered Columns</h2>
<div class="row">
  <div class="small-2 small-centered columns" style="background-color:#8BD6EE;">Small centered</div>
</div>
<div class="row">
  <div class="small-4 small-centered columns" style="background-color:#C0B0F0;">Small centered</div>
</div>
<div class="row">
  <div class="small-6 small-centered large-uncentered columns" style="background-color:#808000;">Large uncentered</div>
</div>
<div class="row">
  <div class="small-8 small-centered columns" style="background-color:#FF6347;">Small centered</div>
</div>
<div class="row">
  <div class="small-10 large-centered columns" style="background-color:#7B68EE;">Large centered</div>
</div>
</body>
</html>

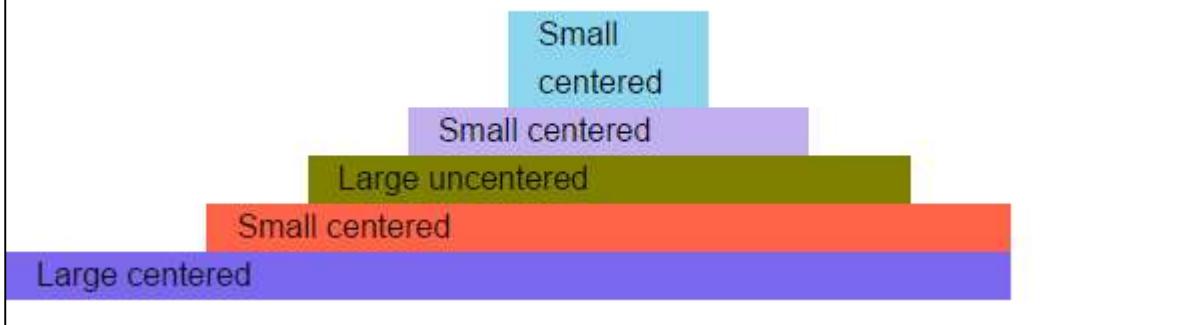
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **centered_columns.htm** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Centered Columns



Foundation – Source Ordering

Description

Source ordering class is used to shift the columns between the breakpoints. The sub-navigation can be an option to position the column either at the left or right side of the page in any device. The **push/pull** class can be applied to any size of device, which you wish to prefer, for example, **medium-push/pull-***, **large-push/pull-***, **small-push/pull-***.

Example

The following example demonstrates the use of *Source ordering* in Foundation:

```
<!DOCTYPE html>
<html>
<head>
  <title>Foundation Template</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
</head>
<body>
  <h2>Source Ordering</h2>
  <div class="row">
    <div class="small-5 small-push-7 columns" style="background-color:#8BD6EE;">Small push class</div>
    <div class="small-7 small-pull-5 columns" style="background-color:#C0B0F0;">small pull class</div>
  </div>
</body>
</html>
```

```

</div>
<div class="row">
  <div class="large-2 large-push-10 columns" style="background-color:#808000;">Large push class</div>
  <div class="large-10 large-pull-2 columns" style="background-color:#FF6347;">Large pull class</div>
</div>
<div class="row">
  <div class="medium-9 medium-push-3 columns" style="background-color:#C0B0F0;">Medium push class</div>
  <div class="medium-3 medium-pull-9 columns" style="background-color:#8BD6EE;">Medium pull class</div>
</div>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **source_ordering.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Source Ordering

small pull class	Small push class
Large push class	
Large pull class	
Medium push class	
Medium pull class	

Foundation – Block Grids

Description

Block-grid is used to split the content. The block grid can be merged into the main grid. Include the format **[size]-up-[n]** class to change the size of all columns for different screen or devices.

Example

The following example demonstrates the use of *Block Grids* in Foundation:

```
<!DOCTYPE html>
```

```

<html>
<head>
    <title>Foundation Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
</head>
<body>
<h2>Block Grids</h2>
<div class="row small-up-2 medium-up-3 large-up-4">
    <div class="column">
        
    </div>
    <div class="column">
        
    </div>
    <div class="column">
        
    </div>
    <div class="column">
        
    </div>
    <div class="column">
        
    </div>
</div>
</body>
</html>

```

Output

97

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **block_grids.html** file.
- Open this HTML file in a browser, an output is displayed as shown below. It is displayed on a medium sized screen.

Block Grids



Foundation – Building Semantically

Using the set of SASS mixins, a grid CSS is generated, which is used to build your own semantic grid.

Rows

The *grid-row()* mixin is used to create a row.

```
.container {
  @include grid-row;
}
```

Columns

The `grid-column()` mixin is used to create a column. The width of the column can be defined in a number of ways.

```
.container {
    @include grid-column;
    //sets 100% column count

    @include grid-column(3);
    //sets column count 25%

    @include grid-column(25%);
    //set percentage for column count

    @include grid-column(1 of 7);
    //custom fraction is set for columns
}
```

The grid column can also be accessed as a function. The percentage value is given without any grid column CSS.

```
.main-content {
    width: grid-column(1 of 7);
}
```

Multiple Grids

The `$grid-column-count` sets numbers of columns to all grids by default. It can be overridden within an instance of row.

```
.container {
    @include grid-row(16) {
        .main-content {
            @include grid-column(5);
        }

        .sidebar {
            @include grid-column(11);
        }
    }
}
```

Without outputting any row CSS, you can temporarily change the grid context using `grid-context()` mixin. Pairing with `breakpoint()` mixin you can make the grid responsive.

```
@include grid-context(7) {
    .sidebar {
        @include grid-column(4);
```

```
}
```

SASS Reference

Following are the SASS references for grids used in Foundation.

S. No.	Basic Grids & Description
1	Variables Using the sass variables we can modify the default styles of this component.
2	Mixins The final CSS output is build using the mixin.

Foundation – SASS Variables

Using the sass variables, we can modify the default styles of this component. Following are a few sass variables:

S. No.	Name & Description	Type	Default Value
1	\$grid-row-width Maximum width of row.	Number	\$global-width
2	\$grid-column-count If the default value of grid is changed then it will affect the grid mixins and the number of CSS classes output.	Number	12
3	\$grid-column-gutter Space between the columns.	Number	30px
4	\$grid-column-align-edge The column is aligned to the opposite edge of the row when it is set as true.	Boolean	true
5	\$block-grid-max When the block grid CSS is used then the <i>.x-up</i> classes highest number is available.	Number	8

Foundation – SASS Mixins

The final CSS output is built using the mixin. The mixin can be used to build your own class structure out of these components. Following are the mixins used to build the final CSS output.

grid-column

A grid column is created.

```
@include grid-column($columns, $gutter);
```

S. No.	Name & Description	Type	Default Value
1	\$columns Sets width of the column. <i>grid-column()</i> function is referred for the available values.	Mixed	\$grid-column-count
2	\$gutter Create space between the columns.	Number	\$grid-column-gutter

grid-column-row

Includes row and column class equivalently to same element. The **grid-col-row()** is the assumed function.

```
@include grid-column-row($gutter);
```

S. No.	Name & Description	Type	Default Value
1	\$gutter Sets width on the either side of the column row of the gutters.	Number	\$grid-column-gutter

grid-column-collapse

The gutters collapse on the column after the padding is removed. The assumed function is **grid-col-collapse()**

```
@include grid-column-collapse;
```

grid-column-uncollapse

The gutters un-collapse on a column after re-adding the padding. The assumed function is **grid-col-uncollapse()**

```
@include grid-column-uncollapse;
```

grid-layout

Sets sizes for the child elements so that the number specified at **\$n** appears in each row.

```
@include grid-layout($n, $selector);
```

S. No.	Name & Description	Type	Default Value
1	\$n Number of elements to display on per row.	Number	None
2	\$selector Selector(s) to use for child elements.	String	'.column'

grid-column-position

Columns are rearranged. The assumed function is **grid-col-pos()**.

```
@include grid-column-position($position);
```

S. No.	Name & Description	Type	Default Value
1	\$position Sets direction to move the column depending upon the specified column count. The negative numbers push the column to left and positive number to the right side.	Number	None

grid-column-unposition

The column position is reset. The assumed function is **grid-col-unpos()**.

```
@include grid-column-unposition;
```

grid-column-offset

Column is offset to the right by **\$n** columns. The assumed function is **grid-col-off()**.

```
@include grid-column-offset($n);
```

S. No.	Name & Description	Type	Default Value
1	\$n Any value is passed accepted by the grid-column() mixin.	Number or List	None

grid-column-end

The default behavior of the last column in a row is disabled that is aligning to the opposite edge. The assumed function is **grid-col-end()**.

```
@include grid-column-end;
```

grid-context

To use different columns, the behavior of the columns defined inside this mixin should be changed.

```
@include grid-context($columns, $root) { }
```

S. No.	Name & Description	Type	Default Value
1	\$columns Number of columns to use.	Number	None
2	\$root Inside the mixin the selectors will nest inside the parent selector when it is set as false.	Boolean	false

grid-row

A Grid row is created.

```
@include grid-row($columns, $behavior, $width, $cf) { }
```

S. No.	Name & Description	Type	Default Value
1	\$columns Sets number of columns for this row.	Number	Null
2	\$behavior The default grid style is modified.	Keywords	Null
3	\$width Maximum width of row.	Number	\$grid-row-width
4	\$cf Whether to include the clearfix or not.	Boolean	true

grid-column-size

Grid column width is set.

```
@include grid-column-size($width);
```

S. No.	Name & Description	Type	Default Value
1	\$width The <i>grid-column()</i> function accepts any passed value. Set width of the column.	Number or List	\$grid-column-count

11. Foundation – Flex Grid

The grid depends on the *flex* display property. It consists of a number of features that are available with the flexbox, like automatic stacking, source ordering, vertical alignment and horizontal alignment.

Browser Support

The flex grid is supported in Chrome, Firefox, Internet Explorer 10+, Safari 6+, Android 4+ and iOS 7+.

The following table describes the features of Flex Grid along with description.

S. No.	Features & Description
1	Importing It includes the export mixin for the flex grid to use the default CSS.
2	Basics The Flex grid structure is similar to that of float grid.
3	Advanced Sizing If the sizing class is not included in the column, then it expands itself and fills the remaining space in the column.
4	Responsive Adjustments If the explicit size of column in flex grid is not given, then it will auto-size the columns
5	Column Alignment The flex grid columns can be aligned horizontal or vertical axis in the parent row.
6	Source Ordering Source ordering helps to rearrange the column in different size of screens.
7	Sass Reference SASS (Syntactically Awesome Stylesheet) is a CSS pre-processor, which helps to reduce repetition with CSS and saves time.

Foundation – Flex Grid Importing

Description

It includes the export mixin for the flex grid to use the default CSS. By default, the flex grid is not enabled.

Let us see some simple formats of importing flex grid as shown below:

```
@import 'foundation';
// include foundation-grid;
@include foundation-flex-grid;
```

Foundation – Flex Grid Basics

Description

The Flex grid structure is similar to that of float grid. The .row class is used for the row and .column is used for the column. For different screen sizing the same classes of grids are used in flex grid also i.e. small-*, medium-* and large-*

Example

The following example demonstrates the use of Basic flex grid in Foundation:

```
<!DOCTYPE html>
<html>
<head>
    <title>Foundation Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
</head>
<body>
<h2>Example of Flex Grid</h2>
<div class="row">
    <div class="small-4 columns" style="background-color:#8BD6EE;">Small 4</div>
    <div class="small-4 columns" style="background-color:#FF6347;">Small 4</div>

```

```

<div class="small-4 columns" style="background-color:#8BD6EE;">Small 4</div>
</div>

<div class="row">
    <div class="medium-6 large-9 columns" style="background-color:#C0B0F0;">Medium 6/ Large 9</div>
    <div class="medium-6 large-3 columns" style="background-color:#7B68EE;">Medium 6/ Large 3</div>
</div>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **flex-grid-basic.html** file.
- Open this HTML file in a browser, an output is displayed as shown below. It is displayed on a medium sized screen.

Example of Flex Grid

Small 4	Small 4	Small 4
Medium 6/ Large 9		
Medium 6/ Large 3		

Foundation – Flex Grid Advanced Sizing

Description

If the sizing class is not included in the column then it expands itself and fills the remaining space in the column. Multiple columns also keep the equal space.

shrink class is used to make the column shrink i.e. it takes the horizontal space, which the content needs.

Example

The following example demonstrates the use of *advanced sizing* in Foundation:

```

<!DOCTYPE html>
<html>
<head>

```

```

<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Foundation Template</title>
<link rel="stylesheet" href="css/foundation.css" />
<script src="js/vendor/jquery.min.js"></script>
<script src="js/foundation.min.js"></script>
</head>
<body>
<h2>Example of Flex Grid</h2>
<div class="row">
  <div class="small-5 columns" style="background-color:#8BD6EE;">Small 5 columns</div>
  <div class="columns" style="background-color:#C0B0F0;">Expanded columns</div>
  <div class="columns" style="background-color:#8BD6EE;">Expanded columns</div>
</div><br/>
<h2>Shrink column</h2>
<div class="row">
  <div class="shrink columns" style="background-color:#FF6347;">Shrink column</div>
  <div class="columns" style="background-color:#7B68EE;">Expanded columns</div>
</div>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **advanced_sizing.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Flex Grid

Small 5 columns
Expanded columns
Expanded columns

Shrink column

Shrink columns!
Expanded columns

Foundation – Flex Grid Responsive Adjustments

Description

If the explicit size of a column in flex grid is not given then it will auto-size the columns. For smaller screens, use `.small-*` class. The `medium-expand` or `large-expand` class is used to expand the behavior.

Example

The following example demonstrates the use of *responsive adjustment* in Foundation:

```
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Foundation Template</title>
    <link rel="stylesheet" href="css/foundation.css" />
    <script src="js/vendor/jquery.min.js"></script>
    <script src="js/foundation.min.js"></script>
</head>
<body>
    <h2>Example of Flex Grid</h2>
    <div class="row">
        <div class="small-6 large-expand columns" style="background-color:#8BD6EE;">First column</div>
        <div class="small-6 large-expand columns" style="background-color:#C0B0F0;">Second column</div>
        <div class="small-6 large-expand columns" style="background-color:#8BD6EE;">Third column</div>
        <div class="small-6 large-expand columns" style="background-color:#C0B0F0;">Forth column</div>
        <div class="small-6 large-expand columns" style="background-color:#8BD6EE;">Fifth column</div>
        <div class="small-6 large-expand columns" style="background-color:#C0B0F0;">Sixth column</div>
    </div>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code **in responsive_adjustment.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Flex Grid

```
First column
Second column
Third column
Forth column
Fifth column
Sixth column
```

Automatic Stacking

Automatic Stacking is the shorthand class used for the responsive adjustment behavior. The `.[size]-unstack` classes is used to stack all columns in a row by default, and unstack in the larger screen.

Example

The following example demonstrates the use of *automatic stacking* in Foundation:

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Foundation Template</title>
  <link rel="stylesheet" href="css/foundation.css" />
  <script src="js/vendor/jquery.min.js"></script>
  <script src="js/foundation.min.js"></script>
</head>
<body>
  <h2>Example of Flex Grid</h2>
  <div class="row medium-unstack">
    <div class="columns" style="background-color:#FF6347;">First</div>
    <div class="columns" style="background-color:#7B68EE;">Second</div>
    <div class="columns" style="background-color:#FF6347;">Third</div>
    <div class="columns" style="background-color:#7B68EE;">Forth</div>
    <div class="columns" style="background-color:#FF6347;">Fifth</div>
    <div class="columns" style="background-color:#7B68EE;">Sixth</div>
  </div>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **automatic_stacking.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Flex Grid

First
Second
Third
Forth
Fifth
Sixth

Foundation – Flex Grid Column Alignment

Description

The flex grid columns can be aligned along the horizontal or vertical axis in the parent row.

Horizontal Alignment

The columns can be aligned in the same manner as that of the content aligned in paragraph.

- Using *.align-[dir]* class in the flex row, you can align the columns according to your wish. By default, the column is aligned to the left.
- The *align-spaced* class is used to keep the space between each column.
- *align-justify* class aligns the columns to the left and right edge with a space between the column.

Example

The following example demonstrates the use of *Horizontal alignment* in Foundation:

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Foundation Template</title>
  <link rel="stylesheet" href="css/foundation.css" />
  <script src="js/vendor/jquery.min.js"></script>
```

```

<script src="js/foundation.min.js"></script>
</head>
<body>
<h2>Example of Flex Grid</h2>
<div class="row">
  <div class="column small-3" style="background-color:#FF6347;">Left</div>
  <div class="column small-3" style="background-color:#C0B0F0;">Side</div>
</div>

<h2>Aligned to right</h2>
<div class="row align-right">
  <div class="column small-4" style="background-color:#FF6347;">Right</div>
  <div class="column small-4" style="background-color:#C0B0F0;">Side</div>
</div>

<h2>Aligned in Middle</h2>
<div class="row align-center">
  <div class="column small-5" style="background-color:#FF6347;">Aligned in</div>
  <div class="column small-5" style="background-color:#C0B0F0;">Middle</div>
</div>

<h2>Aligned to the Edges</h2>
<div class="row align-justify">
  <div class="column small-3" style="background-color:#FF6347;">Left Edge</div>
  <div class="column small-3" style="background-color:#C0B0F0;">Right Edge</div>
</div>

<h2>Aligned to Space around</h2>
<div class="row align-spaced">
  <div class="column small-4" style="background-color:#FF6347;">Spaces</div>
  <div class="column small-4" style="background-color:#C0B0F0;">Spaces</div>
</div>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **horizontal_alignment.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Flex Grid

Left

Side

Aligned to right

Right

Side

Aligned in Middle

Aligned in

Middle

Aligned to the Edges

Left Edge

Right Edge

Aligned to Space around

Spaces

spaces

Vertical Alignment

The flex grid is aligned to the top, by default. You can change the behavior with the other set of alignment classes. In vertical alignment, options are available such as *top*, *middle*, *bottom* and *stretch*.

Example

The following example demonstrates the use of *Vertical Alignment* in Foundation:

```
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
```

```

<title>Foundation Template</title>
<link rel="stylesheet" href="css/foundation.css" />
<script src="js/vendor/jquery.min.js"></script>
<script src="js/foundation.min.js"></script>
</head>
<body>
  <h2>Example of Flex Grid</h2>
  <div class="row align-bottom">
    <div class="columns" style="background-color:#8BD6EE;">Lorem Ipsum is simply dummy text of the printing and typesetting industry.</div>
    <div class="columns" style="background-color:#7B68EE;">Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</div>
  </div>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **vertical_alignment.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Flex Grid

Lorem Ipsum is simply dummy text of the printing and typesetting industry.
 Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Foundation – Flex Grid Source Ordering

Description

Source ordering helps to rearrange the column in different size of screens. Using the *order* property, the columns are sorted into the row, i.e. the lowest number is arranged at the first place.

Example

The following example demonstrates the use of *Source Ordering* in Foundation:

```
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Foundation Template</title>
    <link rel="stylesheet" href="css/foundation.css" />
    <script src="js/vendor/jquery.min.js"></script>
    <script src="js/foundation.min.js"></script>
</head>
<body>
    <h2>Example of Flex Grid</h2>
    <div class="row">
        <div class="column order-1 medium-order-2" style="background-color:#8BD6EE;">
            Appears first on small screen
        </div>
        <div class="column order-2 medium-order-1" style="background-color:#7B68EE;">
            Appears second on small screen
        </div>
    </div>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **flex_grid_source_ordering.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Flex Grid

Appears first on small screen
Appears second on small screen

Foundation – Flex Grid SASS Reference

Description

SASS (Syntactically Awesome Stylesheet) is a CSS pre-processor, which helps to reduce repetition with CSS and saves time.

Mixins

The final CSS output is built using the mixin. The mixin can be used to build your own class structure out of these components. Following are the mixins used to build the final CSS output.

flex-grid-row

A Container is created for flex grid row, using this mixin.

```
@include flex-grid-row($behavior, $width, $columns, $base) { }
```

S. No.	Name & Description	Type	Default Value
1	\$behavior The default grid style is modified.	Keyword or List	null
2	\$width Maximum width of row.	Number	\$grid-row-width
3	\$columns Sets number of columns for this row.	Keyword or List	null
4	\$base It is useful for calling mixin twice on the same element as it creates duplicate output.	Boolean	true

flex-grid-column

Flex grid columns are created using this mixin. Using unstack class on the parent flex row, the column can be stretched to the full width of the container.

```
@include flex-grid-column($columns, $gutter);
```

S. No.	Name & Description	Type	Default Value
1	\$columns <i>flex-grid-column()</i> function is referred for the available values	Mixed	null
2	\$gutter Create space between the columns	Number	\$grid-column-gutter

flex-grid-order

The source of flex grid column is changed. The lowest column number appears at first in the layout.

```
@include flex-grid-order($order);
```

Name & Description	Type	Default Value
\$order Apply order number.	Number	0

Functions

Following is the function of a flex grid:

flex-grid-column

flex property is calculated for the flex grid column. The same value is accepted as basic *grid-column()* function along with the following parameters:

- **null**: Expands the column to full space.
- **shrink**: Contracts the column.

```
flex-grid-column($columns)
```

Name & Description	Type	Default Value
\$columns Set the particular column width	Mixed	null

12. Foundation – Forms

In this chapter, we will study about Forms. Foundation provides powerful, easy and versatile layout system for Forms, which combines form styles and grid support.

The following table lists the form elements used in Foundation.

S. No.	Form Elements & Description
1	Form Basics Creation of forms is easy and very flexible, which are built with a combination of standardized form elements and powerful grid system.
2	Help Text It is used to notify the user about the purpose of the element and is usually placed below a field.
3	Label Positioning You can position your labels left or right of your inputs.
4	Inline Labels and Buttons Extra text or controls can be attached to the left/right of an input field.
5	Custom Controls Custom controls like date pickers, switches or sliders require some attention to access it.
6	SASS Reference You can change the styles of the components by using SASS Reference.

Foundation – Form Basics

Description

Creation of forms is easy and very flexible, which are built with a combination of standardized form elements and powerful grid system.

The following table lists the form basics elements.

S. No.	Form Basics & Description
1	Text Inputs Foundation supports input types, which are used to create a text field.
2	Select Menus You can join many options into one menu by using select menu.
3	Checkboxes and Radio Buttons Checkboxes can be used for selecting multiple options from a list and radio button for selecting only one option.
4	Fieldset Styles It is used to group the related elements together by using <code><fieldset></code> .

Foundation – Forms Text Inputs

Description

Foundation supports input types like `text`, `password`, `time`, `email`, `date`, `datetime-local`, `month`, `number`, `search`, `tel`, `url` and `week`. `Text area` elements are used to create a multi-line text input.

In most of the browsers, `number` input type elements will display up/down controls, which are called as *spin buttons*. The numbers can be incremented and decremented inside the field. You can disable *spin buttons* by setting SASS variable `$input-number-spinners` to false.

Example

The following example demonstrates the use of *input types* in Foundation.

```

<html>
<head>
    <title>Form Text Inputs</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
    s">
</head>
<body>
    <form>
        <div class="row">
            <div class="medium-6 columns">
                <label>Input Text
                    <input type="text">

```

```
</label>
</div>

<div class="medium-6 columns">
    <label>Input search
        <input type="search">
    </label>
</div>
</div>

<div class="row">
    <div class="medium-4 columns">
        <label>Input time
            <input type="time">
        </label>
    </div>
    <div class="medium-4 columns">
        <label>Input Date
            <input type="date">
        </label>
    </div>
    <div class="medium-4 columns">
        <label>Datetime
            <input type="datetime-local">
        </label>
    </div>
</div>

<div class="row">
    <div class="medium-4 columns">
        <label>Number
            <input type="number">
        </label>
    </div>
    <div class="medium-4 columns">
        <label>Telephone
            <input type="tel">
        </label>
    </div>
</div>
```

```
<div class="medium-4 columns">
    <label>Month
        <input type="month">
    </label>
</div>
</div>

<div class="row">
    <div class="medium-6 columns">
        <label>Input url
            <input type="url">
        </label>
    </div>
    <div class="medium-6 columns">
        <label>Input week
            <input type="week">
        </label>
    </div>
</div>

<div class="row">
    <div class="medium-6 columns">
        <label>Email
            <input type="email">
        </label>
    </div>
    <div class="medium-6 columns">
        <label>Password
            <input type="password">
        </label>
    </div>
</div>

<div class="row">
    <div class="medium-12 columns">
        <label>Textarea
            <textarea>Hi .. This is from textarea.....</textarea>
    </div>
</div>
```

```
        </label>
    </div>
</div>

</form>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **forms_text_input.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Input Text	<input type="text"/>
Input search	<input type="text"/>
Input time	<input type="text"/> --:-- --
Input Date	<input type="text"/> mm/dd/yyyy
Datetime	<input type="text"/> mm/dd/yyyy --:-- --
Number	<input type="text"/>
Telephone	<input type="text"/>
Month	<input type="text"/> ----- ---- ▼ ▲
Input url	<input type="text"/>
Input week	<input type="text"/> Week --, ----
Email	<input type="text"/>
Password	<input type="password"/>
Textarea	<input type="text"/> Hi .. This is from textarea.....

Foundation – Select Menus

Description

You can join many options into one menu by using the Select menu.

Example

The following example demonstrates the use of select menu in Foundation.

```

<html>
<head>
    <title>Form Select Menus</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
</head>
<body>
    <form>
        Select your car:
        <select>
            <option value="Rolls-Royce Wraith">Rolls-Royce Wraith</option>
            <option value=" Lamborghini Huracan"> Lamborghini Huracan</option>
            <option value="Bentley Flying Spur">Bentley Flying Spur</option>
            <option value="Porsche 911 Turbo S">Porsche 911 Turbo S</option>
        </select>
    </form>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
in.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
js"></script>
    <script>
        $(document).ready(function() {
            $(document).foundation();
        })
    </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **forms_select_menus.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Select your car:

Rolls-Royce Wraith

Foundation – Checkboxes and Radio Buttons

Description

- Checkboxes can be used for selecting multiple options from a list; radio button can be used to select only one option.
- Enclose a set of checkboxes and radio buttons in a *fieldset* element and provide them a common text by using the *legend* element.
- Every single control in *fieldset* element must have separate label, which can be created using a *label* tag.

Example

The following example demonstrates the use of *checkboxes and radio buttons* in Foundation.

```
<html>
<head>
    <title>Checkboxes and Radio Buttons</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
</head>
<body>
    <form>
        <div class="row">
            <fieldset class="medium-12 columns">
                <legend>Select your vehicle</legend>
```

```

        <input type="radio" name="vehicle" value="XUV" id="vehicleXUV"
required><label for="vehicleXUV">XUV</label>

        <input type="radio" name="vehicle" value="XYLO" id="vehicleXYLO"><label for="vehicleXYLO">XYLO</label>

        <input type="radio" name="vehicle" value="SCORPIO" id="vehicleSCORPIO"><label for="vehicleSCORPIO">SCORPIO</label>

    </fieldset>
</div>
<div class="row">
    <fieldset class="medium-12 columns">
        <legend>Choose your favourite company</legend>
        <input id="audi" type="checkbox"><label for="audi">Audi</label>
        <input id="mahindra" type="checkbox"><label
for="mahindra">Mahindra</label>
        <input id="benz" type="checkbox"><label for="benz">Benz</label>
    </fieldset>
</div>
</form>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **forms_check_radio.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Select your vehicle

XUV XYLO SCORPIO

Choose your favourite company

Audi Mahindra Benz

Foundation – Fieldset Styles

Description

Fieldset is used to group the related elements together. It can be implemented in Foundation by using `.fieldset` class.

Example

The following example demonstrates the use of *fieldset styles* in Foundation.

```
<html>
<head>
    <title>Form Fieldset Styles</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
</head>
<body>
    <fieldset class="fieldset">
        <legend>Select your favourite sport.</legend>
        <input id="football" type="checkbox"><label
for="football">Football</label>
        <input id="cricket" type="checkbox"><label for="cricket">Cricket</label>
        <input id="baseball" type="checkbox"><label
for="baseball">Baseball</label>
        <input id="basketball" type="checkbox"><label
for="basketball">Basketball</label>
    </fieldset>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
in.js"></script>
```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
js"></script>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **forms_fieldset_styles.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

The screenshot shows a single-line input field with the placeholder text "Select your favourite sport.". Below the input field is a horizontal row of four radio buttons. From left to right, the labels next to the radio buttons are "Football", "Cricket", "Baseball", and "Basketball". The radio buttons are standard gray squares with a small circle inside. The entire form is contained within a light gray rectangular frame.

Foundation – Help Text

Description

Help text is used to notify the user about the purpose of the element and is usually placed below a field. A unique ID is provided to it and the *aria-describedby* attribute is added to the input.

Example

The following example demonstrates the use of *help text* in Foundation.

```

<html>
<head>
    <title>Form Help Text</title>

```

```

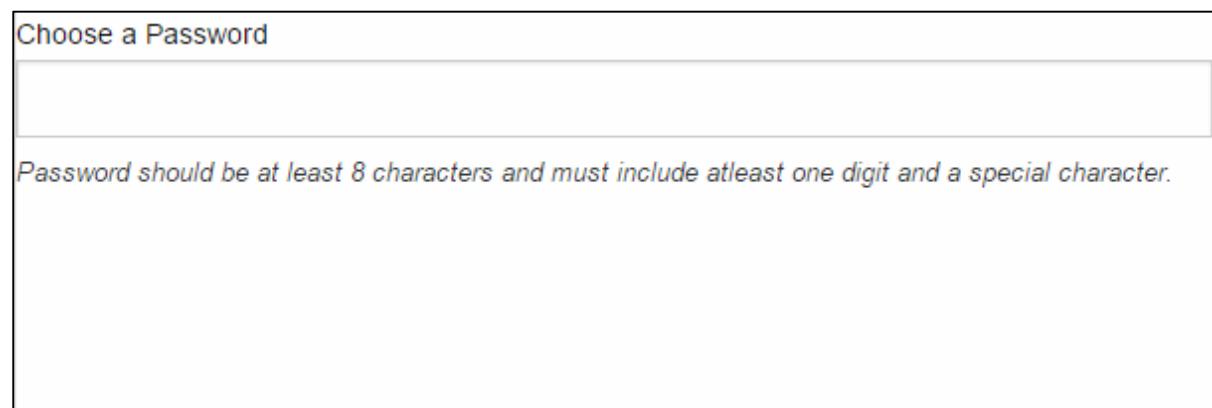
<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
</head>
<body>
  <label>Choose a Password
    <input type="password" aria-describedby="pwdHelpText">
  </label>
  <p class="help-text" id="pwdHelpText">Password should be at least 8 characters and must include atleast one digit and a special character.</p>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
  <script>
    $(document).ready(function() {
      $(document).foundation();
    })
  </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **form_help_text.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



The screenshot shows a web page with a single input field. The placeholder text inside the field is "Choose a Password". Below the input field, there is a help text message: "Password should be at least 8 characters and must include atleast one digit and a special character." This message is displayed in a smaller, gray font.

Foundation – Label Positioning

Description

- You can place your labels to the *left* or *right* of your inputs.
- To place label on right, use *.text-right* or *.float-right* class.
- To place label on left, use *.text-left* or *.float-left* class.
- You can add *.middle* class to align the label vertically middle with its input.

Example

The following example demonstrates the use of *label positioning* in Foundation.

```
<html>
<head>
    <title>Form Label Positioning</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
</head>
<body>
    <form>
        <div class="row">
            <div class="small-6 columns">
                <label for="right-label" class="text-right">Label</label>
            </div>
            <div class="small-6 columns">
                <input type="text" id="right-label" placeholder="Right aligned">
            </div>
        </div>

        <div class="row">
            <div class="small-6 columns">
                <label for="left-label" class="text-left">Label</label>
            </div>
            <div class="small-6 columns">
                <input type="text" id="left-label" placeholder="Left aligned">
            </div>
        </div>
    </form>
</body>
```

```

<div class="row">
    <div class="small-6 columns">
        <label for="middle-label" class="text-right middle">Label</label>
    </div>
    <div class="small-6 columns">
        <input type="text" id="middle-label" placeholder="Right and middle aligned text input">
    </div>
</div>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **form_label_positioning.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Inline Labels and Buttons

Description

Extra text or controls can be attached to the left/right of an input field. Enclose the elements in container `.input-group`, and then add the following classes to the elements inside:

- `input-group-field`: add on the text field.
- `input-group-label`: add on the text label.
- `input-group-button`: add on a button.

Example

The following example demonstrates the use of *label positioning* in Foundation.

```
<html>
<head>
    <title>Form Inline Labels and Buttons</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
</head>
<body>
    <form>
        <div class="input-group">
            <span class="input-group-label">$</span>
            <input class="input-group-field" type="number">
            <a class="input-group-button button">Submit</a>
        </div>
    </form>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script>
        $(document).ready(function() {
            $(document).foundation();
        })
    </script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **form_inline_labels_buttons.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

The screenshot shows a simple form interface. On the left, there is a grey rectangular input field containing a gold-colored dollar sign (\$). To the right of this field is a large, empty white area for input. In the top right corner of the form area, there is a solid blue rectangular button with the word "Submit" written in white capital letters.

Foundation – Custom Controls

Description

Custom controls like date pickers, switches or sliders require some attention to access it. Help text or custom inputs with labels, requires *aria-labelledby* and *aria-describedby* attributes added to them. This will help the screen readers to describe the control.

Example

The following example demonstrates the use of *custom controls* in Foundation.

```
<html>
<head>
    <title>Form Custom Controls</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
</head>
<body>
    <form>
        <label id="rateLabel">Rate Yourself</label>
        <div class="slider" aria-labelledby="rateLabel" aria-describedby="rateHelpText" data-slider data-initial-start='80' data-end='100'>
            <span class="slider-handle" data-slider-handle role="slider" tabindex="1"></span>
            <span class="slider-fill" data-slider-fill></span>
            <input type="hidden">
    
```

```

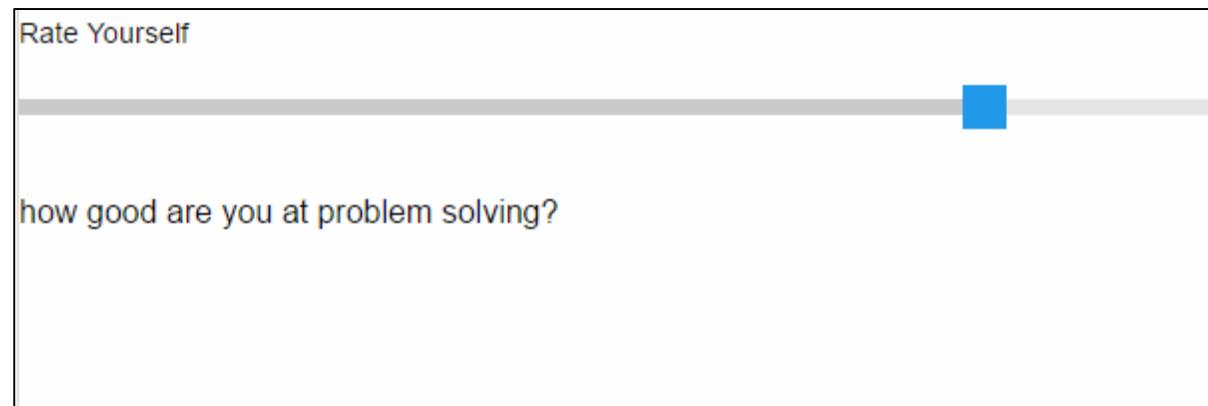
</div>
<p id="rateHelpText">how good are you at problem solving?</p>
</form>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **form_custom_controls.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Form SASS Reference

Description

You can change the styles of the components by using SASS Reference.

Variables

The following table lists the SASS variables in a project's settings file that make this component's default styles to get customized.

S. No.	Name & Description	Type	Default Value
1	\$fieldset-border Custom fieldset's default border.	Border	1px solid \$medium-gray
2	\$fieldset-padding Custom fieldset's default internal padding.	Number	rem-calc(20)
3	\$fieldset-margin Custom fieldset's default margin around.	Number	rem-calc(18 0)
4	\$legend-padding Padding between legend text and fieldset border by default.	Number	rem-calc(0 3)
5	\$form-spacing Form element's global spacing.	Number	rem-calc(16)
6	\$helptext-color Default color of help text.	Color	#333
7	\$helptext-font-size Font size of the help text by default.	Number	rem-calc(13)
8	\$helptext-font-style Font style of the help text by default.	Keyword	italic
9	\$input-prefix-color Represents the color of labels prefixed to an input.	Color	\$black
10	\$input-prefix-background Label's background color prefixed to an input.	Color	\$light-gray
11	\$input-prefix-border Border around labels which will be prefixed to an input.	Border	1px solid \$medium-gray
12	\$input-prefix-padding Pre/postfixed input label's left/right padding.		1rem
13	\$form-label-color Represents color for form labels.	Color	\$black
14	\$form-label-font-size Represents font size for form labels.	Number	rem-calc(14)

15	\$form-label-font-weight Represents font weight for form labels.	Keyword	\$global-weight-normal
16	\$form-label-line-height Line height for labels of the form. Higher number represents more space between label and input field.	Number	1.8
17	\$select-background Represents background color for select menus.	Color	#fafafa
18	\$select-triangle-color Dropdown triangle's color. Set <i>transparent</i> to remove it entirely.	Color	#333
19	\$select-radius Represents the default radius for select menus.	Color	\$global-radius
20	\$input-color Represents font color of text inputs.	Color	\$black
21	\$input-font-family Represents font family of text inputs.	Font	inherit
22	\$input-font-size Represents font size of text inputs.	Number	rem-calc(16)
23	\$input-background Represents background color of text inputs.	Color	\$white
24	\$input-background-focus Represents background color of focused text inputs.	Color	\$white
25	\$input-background-disabled Background color of text inputs which are disabled.	Color	\$light-gray
26	\$input-border Represents border around text inputs.	Border	1px solid \$medium-gray
27	\$input-border-focus Represents border around focused text inputs.	Color	1px solid \$dark-gray

28	\$input-shadow Represents box shadow within text inputs when not focused.	Shadow	inset 0 1px 2px rgba(\$black, 0.1)
29	\$input-shadow-focus Represents box shadow outside text inputs when focused.	Shadow	0 0 5px \$medium-gray
30	\$input-cursor-disabled Represents the cursor to use when hovered over text input which is disabled.	Cursor	default
31	\$input-transition Represents properties to transition on text inputs.	Transition	box-shadow 0.5s, border-color 0.25s ease-in-out
32	\$input-number-spinners Enables the up/down buttons that is added to <code><input type='number'></code> elements by Chrome and Firefox.	Boolean	true
33	\$input-radius Represents radius for text inputs.	Border	\$global-radius

13. Foundation – Visibility Classes

Description

- Foundation uses visibility classes to show or hide elements based upon device orientation (portrait and landscape) or screen size (small, medium, large or xlarge screen).
- It allows a user to use elements based on the browsing environment.

The following table lists visibility classes of Foundation, which control the elements based on their browsing environment:

S. No.	Visibility Class & Description
1	Show by Screen Size It shows the elements based on the device by using <code>.show</code> class.
2	Hide by Screen Size It hides the elements based on the device by using <code>.hide</code> class.

Foundation supports some classes where you can hide the content by using the `.hide` and `.invisible` classes and displays nothing on the page.

Foundation – Show by Screen Size

Description

- It shows the elements based on the device, by using `.show` class, on the page, which the user is viewing.
- You can display the content for small devices using the `show-for-small` class, for medium devices, use the `show-for-medium` class and for large devices, use the `show-for-large` class.

Example

The following example demonstrates the use of `.show` class in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

```

<title>Show by Screen Size</title>
<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Show by Screen Size Example</h2>
<p class="show-for-small">This line appears on small or medium screen.</p>
<p class="show-for-medium">This line appears on medium screen or larger.</p>
<p class="show-for-large">This line appears on large screen or larger.</p>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **show_screen.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Show by Screen Size Example

This line appears on small or medium screen.

Foundation – Hide by Screen Size

Description

- It hides the elements based on the device by using *.hide* class.
- You can hide the elements for small devices by using the *hide* class instead of using *.hide-for-small* class.

- For medium device, you can hide the content by using *hide-for-medium* class and for large device, use the *hide-for-large* class.

Example

The following example demonstrates the use of *.hide* class in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Hide by Screen Size</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Hide by Screen Size Example</h2>
  <p class="hide-for-small">This line disappears on small or medium screen.</p>
  <p class="hide-for-medium">This line disappears on medium screen or larger.</p>
  <p class="hide-for-large">This line disappears on large screen or larger.</p>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **hide_screen.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Hide by Screen Size Example

This line disappears on small or medium screen.

This line disappears on medium screen or larger.

This line disappears on large screen or larger.

Orientation Detection

The devices can determine different orientations by using *landscape* and *portrait* functionality. The hand held devices such as mobile phones specifies the different orientations when you rotate them. For desktop, the orientation will be landscape always.

Accessibility

The following table lists the accessibility techniques for screen readers, which hides the content while making it readable by screen readers:

S. No.	Accessibility Class & Description
1	Show for Screen Readers It uses <i>show-for-sr</i> class to hide the content while preventing screen readers from reading it.
2	Hide for Screen Readers It uses <i>aria-hidden</i> attribute, which makes text visible but could not be read by a screen reader.
3	Creating Skip Links Screen reader will create a skip link to get the navigation to your site's content.

Foundation – Show for Screen Readers

Description

It hides the content while preventing screen readers from reading it. You can hide the content by making it readable by screen readers by using the class *show-for-sr*.

Example

The following example demonstrates the use of *.show-for-sr* class in Foundation:

```

<!doctype html>
<head>
<meta charset="utf-8" />
<meta http-equiv="x-ua-compatible" content="ie=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Show Screen Readers</title>
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Show Screen Readers Example</h2>
<p class="show-for-sr">Welcome to TutorialsPoint!!!</p>
<p>you can't see a line of text above this one.</p>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **show_screen_readers.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Show Screen Readers Example

you can't see a line of text above this one.

Foundation – Hide for Screen Readers

Description

It uses `aria-hidden="true"` attribute, which makes text visible but cannot be read by a screen reader.

Example

The following example demonstrates the use of `aria-hidden` attribute in the Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Hide Screen Readers</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Hide Screen Readers Example</h2>
  <p aria-hidden="true">Welcome to TutorialsPoint!!! You can see this line, but
  won't be read by a screen reader</p>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **hide_screen_readers.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Hide Screen Readers Example

Welcome to Tutorialspoint!!! You can see this line, but won't be read by a screen reader

Foundation – Skip Links

Description

- If your site contains lot of navigation, then it is a bit difficult for a screen reader to get the site content by reading through the entire navigation.
- To solve this problem, you can use *skip link* at the top of your page and use the *.show-on-focus* class to hide the content and *tabindex="0"*attribute to make an element focusable.

Example

The following example demonstrates the use of skip links attribute in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Creating Skip Links</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Skip Links Example</h2>
  <p><a class="show-on-focus" href="#myval">Skip this info</a></p>
```

```
<main id="myval" role="main" tabindex="0">
  <p>Hide Screen Readers Example</p>
</main>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **skip_links.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Skip Links Example

Hide Screen Readers Example

SASS Reference

Foundation uses the following mixins to display CSS output, which allows building own class structure for your components:

S. No.	Mixin & Description	Parameter	Type
1	show-for By default, it hides an element and displays it above certain screen size.	\$size	Keyword
2	show-for-only By default, it hides an element and displays it within breakpoint.	\$size	Keyword
3	hide-for By default, it shows an element and hides it above a certain screen size.	\$size	Keyword
4	hide-for-only By default, it shows an element and hides it above a certain screen size.	\$size	Keyword

The default value of all these mixins will be set to *none*.

14. Foundation – Base Typography

Description

Typography in Foundation defines headings, paragraphs, lists and other inline elements which create attractive and simple default styles for elements.

The following table lists the different types of typography used in Foundation:

S. No.	Typography & Description
1	Paragraphs Paragraph is a group of sentences defined with different font size, highlighted words, line height etc.
2	Header It defines HTML headings from h1 to h6.
3	Links It creates a hyperlink that opens another document when you click on the text or an image.
4	Dividers It is used to give a break between the sections by using <hr> tag.
5	Ordered and Unordered Lists Foundation supports ordered lists, unordered lists to list the things.
6	Definition Lists Definition Lists are used to display name value pairs.
7	Blockquotes It represents block of text, which defines much bigger than normal.
8	Abbreviations and Code Abbreviation defines a shortened term of word or phrase and code represents a piece of code.

9	Keystrokes It is used to perform a specific function.
10	Accessibility Foundation provides some guidelines to access the content of the page.

Foundation – Base Typography Paragraph

Description

Paragraph is a group of sentences defined with different font size, highlighted words, line height etc. and includes more content, which makes the paragraph look good in different CSS styles.

Example

The following example demonstrates the use of *base typography paragraphs* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Base Typography Paragraphs</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Base Typography Paragraphs Example</h2>
  <p>A Framework for any device, medium, and accessibility. <strong>Foundation</strong> is a family of responsive front-end frameworks that make it easy to design beautiful responsive websites, apps and emails that look amazing on any device. <i>Foundation</i> is semantic, readable, flexible, and completely customizable.</p>
```

```

<p> Foundation for Sites provides you with HTML, CSS, & JavaScript to help
you quickly prototype. It's easy to customize & extend to get your project
polished & shipped.</p>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **paragraphs.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Base Typography Paragraphs Example

A Framework for any device, medium, and accessibility. **Foundation** is a family of responsive front-end frameworks that make it easy to design beautiful responsive websites, apps and emails that look amazing on any device. *Foundation* is semantic, readable, flexible, and completely customizable.

Foundation for Sites provides you with HTML, CSS, & JavaScript to help you quickly prototype. It's easy to customize & extend to get your project polished & shipped.

Foundation – Base Typography Header

Description

It defines styling for all HTML headings from h1 to h6. You can make the text small by using the `<small>` tag in the header elements.

Example

The following example demonstrates the use of *base typography header* in Foundation:

```

<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Base Typography Header</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Base Typography Header Example</h2>
<h1>This is heading one</h1>
<h2>This is heading <small>two</small></h2>
<h3>This is heading three</h3>
<h4>This is heading four</h4>
<h5>This is heading five</h5>
<h6>This is heading six</h6>
</body>
</html>

```

Note: In the heading tag h2, we have used the <small> element.

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **header.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Base Typography Header Example

This is heading one

This is heading two

This is heading three

This is heading four

This is heading five

This is heading six

Foundation – Base Typography Links

Description

It creates a hyperlink, that opens another document, when you click on the text or an image.

Example

The following example demonstrates the use of base typography links in Foundation:

```
<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Base Typography Links</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
in.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
js"></script>
</head>
<body>
    <h2>Base Typography Links Example</h2>
    <p>A <a target="_blank" rel="nofollow"
    href="http://foundation.zurb.com/">Foundation</a> is a family of responsive
    front-end frameworks that make it easy to design beautiful responsive websites,
    apps and emails that look amazing on any device.</p>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **links.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Base Typography Links Example

A [Foundation](#) is a family of responsive front-end frameworks that make it easy to design beautiful responsive websites, apps and emails that look amazing on any device.

Foundation – Base Typography Dividers

Description

It is used to give a break between the sections using the `<hr>` tag.

Example

The following example demonstrates the use of *base typography dividers* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Base Typography Dividers</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Base Typography Dividers Example</h2>
  <section>
    <h2>Foundation</h2>
    <p>Foundation is a family of responsive front-end frameworks that make it easy to design beautiful responsive websites.</p>
  </section>
  <hr>
```

```

<section>
    <h2>Bootstrap</h2>
    <p>Bootstrap is a sleek, intuitive, and powerful, mobile first front-end framework for faster and easier web development.</p>
</section>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **dividers.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Base Typography Dividers Example

Foundation

Foundation is a family of responsive front-end frameworks that make it easy to design beautiful responsive websites.

Bootstrap

Bootstrap is a sleek, intuitive, and powerful, mobile first front-end framework for faster and easier web development.

Foundation – Base Typography Lists

Description

An ordered list is a list that specifies some sort of sequential order and is displayed by numbers. An unordered list is a list that does not have any particular order and is displayed with bullets.

Example

The following example demonstrates the use of *ordered and unordered lists* in Foundation:

```

<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">

```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Base Typography Lists</title>
<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Example of Ordered List</h2>
<ol>
    <li>List 1</li>
    <li>List 2</li>
    <li>List 3</li>
    <li>List 4</li>
</ol>
<h2>Example of UnOrdered List</h2>
<ul>
    <li>List 1</li>
    <li>List 2</li>
    <li>List 3</li>
    <li>List 4</li>
</ul>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **unordered_ordered_lists.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Ordered List

1. List 1
2. List 2
3. List 3
4. List 4

Example of UnOrdered List

- List 1
- List 2
- List 3
- List 4

Foundation – Base Typography Definition Lists

Description

Definition Lists are used to display the name value pairs by using both `<dt>` and the `<dd>` elements. The element `<dt>` stands for *definition term*, which is the term that is being defined and the element `<dd>` is the definition of the `<dt>`.

Example

The following example demonstrates the use of *definition lists* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Base Typography Definition Lists</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
```

```

<h2>Base Typography Definition Lists Example</h2>
<dl>
  <dt>Foundation</dt>
  <dd>The most advanced responsive front-end framework that make it easy to design responsive websites..</dd>
  <dt>Bootstrap</dt>
  <dd>Bootstrap is a powerful, mobile first front-end framework for easier web development.</dd>
</dl>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **definition_lists.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Base Typography Definition Lists Example

Foundation

The most advanced responsive front-end framework that make it easy to design responsive websites..

Bootstrap

Bootstrap is a powerful, mobile first front-end framework for easier web development.

Foundation – Base Typography Blockquotes

Description

It represents a block of text, which defines much bigger than the normal text and spans the width of the text.

Example

The following example demonstrates the use of *blockquotes* in Foundation:

```

<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">

```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Base Typography Blockquotes</title>
<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Base Typography Blockquotes Example</h2>
<blockquote>
    A Foundation is a family of responsive front-end frameworks that make it
    easy to design beautiful responsive websites, apps and emails that look
    amazing on any device.
    <cite>Maintained by ZURB</cite>
</blockquote>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **blockquotes.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Base Typography Blockquotes Example

A Foundation is a family of responsive front-end frameworks that make it easy to design beautiful responsive websites, apps and emails that look amazing on any device.

— *Maintained by ZURB*

Foundation – Base Typography Abbreviations and Code

Description

Abbreviation defines a shortened term of a word or a phrase and **Code** represents a piece of code.

Example

The following example demonstrates the use of *abbreviations and code* in Foundation:

```
<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Base Typography Abbreviations and Code</title>
    <link rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
        src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
        src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
    <h2>Example of Abbreviations</h2>
    <blockquote>
        <p><abbr title="Hyper Text Markup Language">HTML</abbr> which is the most widely used language on Web to develop web pages.</p>
    </blockquote>
    <h2>Example of Code</h2>
    <p>You can use break between the sections by using <code><hr></code> tag.</p>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **abbreviations_code.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Abbreviations

HTML which is the most widely used language on Web to develop web pages.

Example of Code

You can use break between the sections by using `<hr>` tag.

Foundation – Base Typography Keystrokes

Description

It is used to perform a specific function by using the `<kbd>` element.

Example

The following example demonstrates the use of *keystrokes* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Base Typography Keystrokes</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Base Typography Keystrokes Example</h2>
  <p>Use <kbd>ctrl + esc</kbd> to open the start menu.</p>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **keystrokes.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Base Typography Keystrokes Example

Use `ctrl + esc` to open the start menu.

Foundation – Base Typography Accessibility

Description

Foundation provides the following guidelines to access the content of the page.

Text vs. Images

It is not possible to read an image and text inside an image. You can use the actual text over the text in graphic. In addition, text in an image cannot be resized by a browser like a normal text. You can use the *alt* attribute of the image to read the text.

Contrast

The low vision users can read only if there is a high contrast between color of an element text and its background. The minimum recommended contrast ratio is 4.5:1. There are no automated tools for checking color combination, but you can check the color contrast using the tool such as [WebAIM](#) color contrast checker.

Type Size

You can use the *rem* and *em* units for font size, padding, margins etc. The user can ensure the text size by scaling up and down the units and able to resize the browser up to 200%. The Foundation uses *rem-calc()* function which takes pixel values and convert them to *rem* values.

SASS Reference

You can change the styles of the components by using the following SASS variables as listed in the table.

S. No.	Name & Description	Type	Default Value
1	\$header-font-family Specifies the font family for header elements.	String or List	\$body-font-family
2	\$header-font-weight Specifies font weight of headers.	String	\$global-weight-normal
3	\$header-font-style Provides font style of headers.	String	normal
4	\$font-family-monospace Font stack used for elements that use monospaced type, such as code samples.	String or List	Consolas, 'Liberation Mono', Courier, monospace
5	\$header-sizes Defines screen sizes of headings and each key is a breakpoint, and each value is a map of heading sizes.	Map	<pre>small: ('h1': 24 'h2': 20 'h3': 19 'h4': 18 'h5': 17 'h6': 16) medium: ('h1': 48 'h2': 40 'h3': 31 'h4': 25 'h5': 20 'h6': 16)</pre>
6	\$header-color Provides color of the headers.	Color	inherit
7	\$header-lineheight Defines line height of headers.	Number	1.4
8	\$header-margin-bottom Provides bottom margin of headers.	Number	0.5rem
9	\$header-text-rendering Defines method for text rendering.	String	optimizeLegibility

S. No.	Name & Description	Type	Default Value
10	\$small-font-size Specifies font size for <code><small></code> elements.	Number	80%
11	\$paragraph-margin-bottom Specifies bottom margin of paragraphs.	Number	1rem
12	\$paragraph-text-rendering Methods for text rendering paragraph.	String	optimizeLegibility
13	\$code-color Provides text color to code samples.	Color	\$black
14	\$code-font-family Provides font family to the code samples.	String or List	\$font-family-monospace
15	\$code-border Specifies the border around the code.	List	1px solid \$medium-gray
16	\$code-padding Specifies the padding around the text.	Number or List	rem-calc(2 5 1)
17	\$anchor-color Default color for links.	Color	\$primary-color
18	\$anchor-color-hover Specifies the default color for links on hover.	Color	scale-color(\$anchor-color, \$lightness: -14%)
19	\$anchor-text-decoration Default text decoration for links.	String	none
20	\$anchor-text-decoration-hover Default text decoration for links on hover.	String	none
21	\$hr-width Defines maximum width of a divider.	Number	\$global-width
22	\$hr-border Specifies default border for a divider.	List	1px solid \$medium-gray
23	\$hr-margin Default margin for a divider.	Number or List	rem-calc(20) auto
24	\$list-lineheight It defines line height for items in a list.	Number	\$paragraph-lineheight

S. No.	Name & Description	Type	Default Value
25	\$list-style-type Provides bullet type for unordered lists.	String	disc
26	\$list-style-position It defines positioning for bullets on unordered lists.	String	outside
27	\$list-side-margin Defines left side (or right) margin.	Number	1.25rem
28	\$deflist-term-weight Provides font weight for <code><dt></code> elements.	String	\$global-weight-bold
29	\$deflist-term-margin-bottom Provides spacing between <code><dt></code> and <code><dd></code> elements.	Number	0.3rem
30	\$blockquote-color It applies text color of <code><blockquote></code> elements.	Color	\$dark-gray
31	\$blockquote-padding Provides padding inside a <code><blockquote></code> elements.	Number or List	rem-calc(9 20 0 19)
32	\$blockquote-border It gives side border for the <code><blockquote></code> elements.	List	1px solid \$medium-gray
33	\$cite-font-size Defines font size for the <code><cite></code> elements.	Number	rem-calc(13)
34	\$cite-color Provides text color for <code><cite></code> elements.	Color	\$dark-gray
35	\$keystroke-font Defines font family for the <code><kbd></code> elements.	String or List	\$font-family-monospace
36	\$keystroke-color Defines text color for the <code><kbd></code> elements.	Color	\$black

S. No.	Name & Description	Type	Default Value
37	\$keystroke-background Provides background color for the <kbd> elements.	Color	\$light-gray
38	\$keystroke-padding Specifies padding for the <kbd> elements.	Number or List	rem-calc(2 4 0)
39	\$keystroke-radius Displays the border radius for the <kbd> elements.	Number or List	\$global-radius
40	\$abbr-underline Provides the bottom border style for the <abbr> elements.	List	1px dotted \$black

15. Foundation – Typography Helpers

In this chapter we will study about **Typography Helpers**. Typography helpers are used to format your text in semantically relevant ways. Helper classes in Foundation let you to scaffold some typographic styles quicker.

The following table lists the *Typography Helpers*, which are used in Foundation.

S. No.	Typography helpers & Description
1	Text Alignment It helps in changing the element's text alignment like <i>left</i> , <i>right</i> , <i>center</i> and <i>justify</i> .
2	Subheader Subheaders can be added to any header element by using the <i>.subheader</i> class.
3	Lead Paragraph It is considerably larger block of text compared to the normal text, which can be used for advertisement or other descriptive text.
4	Un-bulleted List By default, the <i></i> is a bulleted list in Foundation. To remove the bullets, you can use <i>.no-bullet</i> class.
5	Statistics Whenever you are dealing with dashboard, you need to highlight some important numbers. You can achieve this by using <i>.stat</i> class.

Foundation – Text Alignment

Description

Text of an element can be aligned by adding *.text-left*, *.text-right*, *.text-center* or *.text-justify* classes. By putting a breakpoint in front of a text alignment class, will make the alignment only to apply on that screen size or a larger one. For instance, *.medium-text-center* will align the text to the left on smallest screens, but changes to the center when screen size is medium and larger.

Example

The following example demonstrates the use of *text alignment* in Foundation.

```
<html>
<head>
    <title>Text Alignment</title>
```

```

<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
</head>
<body>
    <p class="text-left">Foundation is one of the advanced front-end framework for designing beautiful responsive websites. It works on all types of devices and provides you with HTML, CSS and JavaScript plugins.</p>
    <p class="text-right">Foundation is one of the advanced front-end framework for designing beautiful responsive websites. It works on all types of devices and provides you with HTML, CSS and JavaScript plugins.</p>
    <p class="text-center">Foundation is one of the advanced front-end framework for designing beautiful responsive websites. It works on all types of devices and provides you with HTML, CSS and JavaScript plugins.</p>
    <p class="text-justify">Foundation is one of the advanced front-end framework for designing beautiful responsive websites. It works on all types of devices and provides you with HTML, CSS and JavaScript plugins.</p>
    <p class="medium-text-center">Foundation is one of the advanced front-end framework for designing beautiful responsive websites. It works on all types of devices and provides you with HTML, CSS and JavaScript plugins.</p>
    <script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script>
        $(document).ready(function() {
            $(document).foundation();
        })
    </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **typography_text_alignment.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Foundation is one of the advanced front-end framework for designing beautiful responsive websites. It works on all types of devices and provides you with HTML, CSS and JavaScript plugins.

Foundation is one of the advanced front-end framework for designing beautiful responsive websites. It works on all types of devices and provides you with HTML, CSS and JavaScript plugins.

Foundation is one of the advanced front-end framework for designing beautiful responsive websites. It works on all types of devices and provides you with HTML, CSS and JavaScript plugins.

Foundation is one of the advanced front-end framework for designing beautiful responsive websites. It works on all types of devices and provides you with HTML, CSS and JavaScript plugins.

Foundation is one of the advanced front-end framework for designing beautiful responsive websites. It works on all types of devices and provides you with HTML, CSS and JavaScript plugins.

Foundation – Subheader

Description

Subheader lightens the color of the header by using the *.subheader* class. It can be added to any header element.

Example

The following example demonstrates the use of *subheader* in Foundation.

```
<html>
<head>
    <title>Subheader</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
</head>
<body>
    <h1 class="subheader">This is heading one</h1>
    <h2 class="subheader">This is heading two</h2>
    <h3 class="subheader">This is heading three</h3>
    <h4 class="subheader">This is heading four</h4>
```

```

<h5 class="subheader">This is heading five</h5>
<h6 class="subheader">This is heading six</h6>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script>
$(document).ready(function() {
  $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **typography_subheader.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

This is heading one

This is heading two

This is heading three

This is heading four

This is heading five

This is heading six

Foundation – Lead Paragraph

Description

It is considerably a larger block of text compared to the normal text, which can be used for advertisement or other descriptive text.

Example

The following example demonstrates the use of *lead paragraph* in Foundation.

```
<html>
```

```

<head>
    <title>Lead Paragraph</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
</head>
<body>
    <p class="lead">This is the paragraph with <em>lead</em> class. Foundation is
    one of the advanced front-end framework for designing beautiful responsive
    websites. It works on all types of devices and provides you with HTML, CSS and
    JavaScript plugins.</p>
    <p>This is the normal paragraph. Foundation is one of the advanced front-end
    framework for designing beautiful responsive websites. It works on all types of
    devices and provides you with HTML, CSS and JavaScript plugins.</p>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script>
        $(document).ready(function() {
            $(document).foundation();
        })
    </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **typography_lead_paragraph.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

This is the paragraph with *lead* class. Foundation is one of the advanced front-end framework for designing beautiful responsive websites. It works on all types of devices and provides you with HTML, CSS and JavaScript plugins.

This is the normal paragraph. Foundation is one of the advanced front-end framework for designing beautiful responsive websites. It works on all types of devices and provides you with HTML, CSS and JavaScript plugins.

Foundation – Un-bulleted List

Description

By default, the element `` is a bulleted list in Foundation. To remove bullets, you can use `.no-bullet` class.

Example

The following example demonstrates the use of *un-bulleted list* in Foundation.

```

<html>
<head>
    <title>Un-bulleted List</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
</head>
<body>
    <p>List with <i>no-bullet</i> class.</p>
    <ul class="no-bullet">
        <li>List item 1</li>
        <li>List item 2
            <ul>
                <li>Nested list item 1</li>
                <li>Nested list item 2</li>
                <li>Nested list item 3</li>
            </ul>
        </li>
        <li>List item 3</li>
    </ul>
    <p>List without <i>no-bullet</i> class.</p>

```

```

<ul>
    <li>List item 1</li>
    <li>List item 2
        <ul>
            <li>Nested list item 1</li>
            <li>Nested list item 2</li>
            <li>Nested list item 3</li>
        </ul>
    </li>
    <li>List item 3</li>
</ul>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **typography_un_bulleted_list.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

List with *no-bullet* class.

```
List item 1
List item 2
  Nested list item 1
  Nested list item 2
  Nested list item 3
List item 3
```

List without *no-bullet* class.

- List item 1
- List item 2
 - Nested list item 1
 - Nested list item 2
 - Nested list item 3
- List item 3

Foundation – Statistics

Description

Whenever you are dealing with dashboard, you need to highlight some important numbers. You can achieve this by using the *.stat* class.

Example

The following example demonstrates the use of *statistics* in Foundation.

```
<html>
<head>
  <title>Statistics</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
  s">
</head>
<body>
  <p>Number of online users</p>
  <div class="stat">7080</div>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
  in.js"></script>
```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
js"></script>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **typography_statistics.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



SASS Reference

Variables

The following table lists the SASS variables in the project's settings file that makes the component's default styles to get customized.

S. No.	Name & Description	Type	Default Value
1	\$lead-font-size Font size for lead paragraphs by default.	Number	\$global-font-size * 1.25
2	\$lead-lineheight Line height for lead paragraph by default.	String	1.6
3	\$subheader-lineheight Subheader's default line height.	Number	1.4

4	\$subheader-color Subheader's default font color.	Color	\$dark-gray
5	\$subheader-font-weight Subheader's default font weight.	String	\$global-weight-normal
6	\$subheader-margin-top Subheader's default top margin.	Number	0.2rem
7	\$subheader-margin-bottom Subheader's default bottom margin.	Number	0.5rem
8	\$stat-font-size Static number's default font size	Number	2.5rem

16. Foundation – Basic Controls

In this chapter, we will study about **basic controls**. Foundation provides basic controls such as *buttons*, *sliders* and *switches*.

The following table lists the basic controls used in Foundation.

S. No.	Basic Controls & Description
1	Button Foundation supports many button styles that can be customized according to your needs.
2	Button Group They are containers for corresponding action elements. It works fine when group of actions is displayed in a bar.
3	Close Button Close button is used when you want something to go away by clicking on it.
4	Slider Sliders are very useful for setting certain values inside a range.
5	Switch It allows you to turn off or turn on the switch by clicking on it.

Foundation – Basic Controls Button

Description

Foundation supports many button styles that can be customized according to your needs. When you require more traditional actions, *buttons* are appropriate tools.

The following table lists the basic controls used in Foundation.

S. No.	Button Controls & Description
1	Basics Buttons can be created with less markup tags like <code><a></code> and <code><button></code> .
2	Sizing Size and shape of the buttons can be changed by using additional classes.
3	Coloring Buttons can be colored to give them a additional meaning.

4	Hollow Style You can use <code>.hollow</code> class to give your button a hollow style.
5	Dropdown Arrows You can use <code>.dropdown</code> class to add dropdown arrow to your button.
6	Accessibility When your button does not contain readable text, you can add screen reader-only text to tell it's purpose.
7	SASS Reference You can change the styles of the components by using SASS Reference.

Foundation – Button Basics

Description

Buttons can be created with fewer markup. Buttons in Foundation can be created by two tags which are listed below:

- **<a>** : You can use it when your button is a link to another page or an anchor in page. Usually JavaScript is not required to work with anchors.
- **<button>** : You can use it when some action has to be performed by a button. You always require JavaScript function to work with `<button>` elements.

Example

The following example demonstrates how to use `button` in Foundation.

```

<html>
<head>
  <title>Button Basics</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
</head>
<body>
  <a href="contact.html" class="button">Contact Us</a>
  <a href="#gallery" class="button">Gallery</a>
  <button type="button" class="success button">Success</button>
  <button type="button" class="alert button">Alert</button>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>

```

```

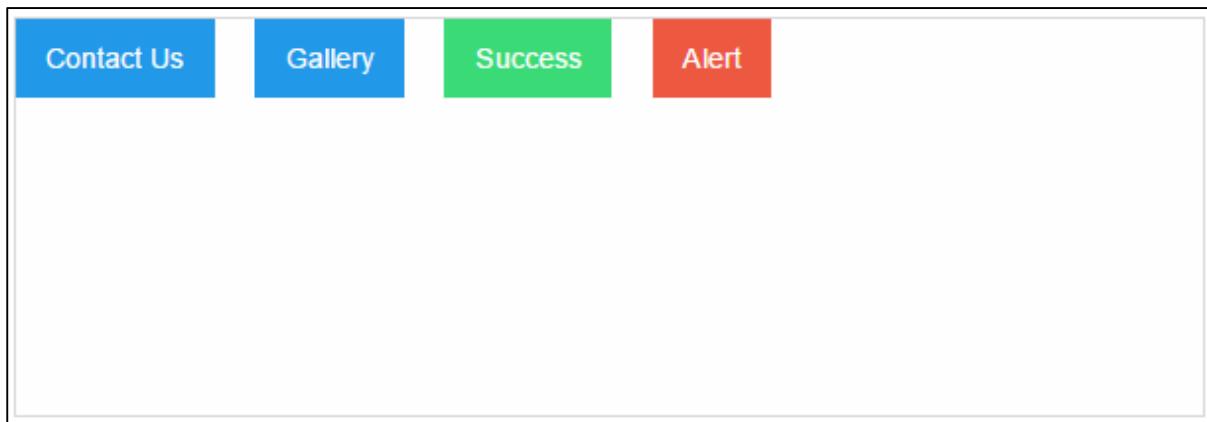
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
js"></script>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **button_basics.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Button Sizing

Description

Size and shape of the buttons can be changed by using additional classes such as *tiny*, *small*, *large*, *expanded* and *small expanded*.

Example

The following example demonstrates how to change the *size of the buttons* in Foundation.

```

<html>
<head>
    <title>Button Sizing</title>

```

```

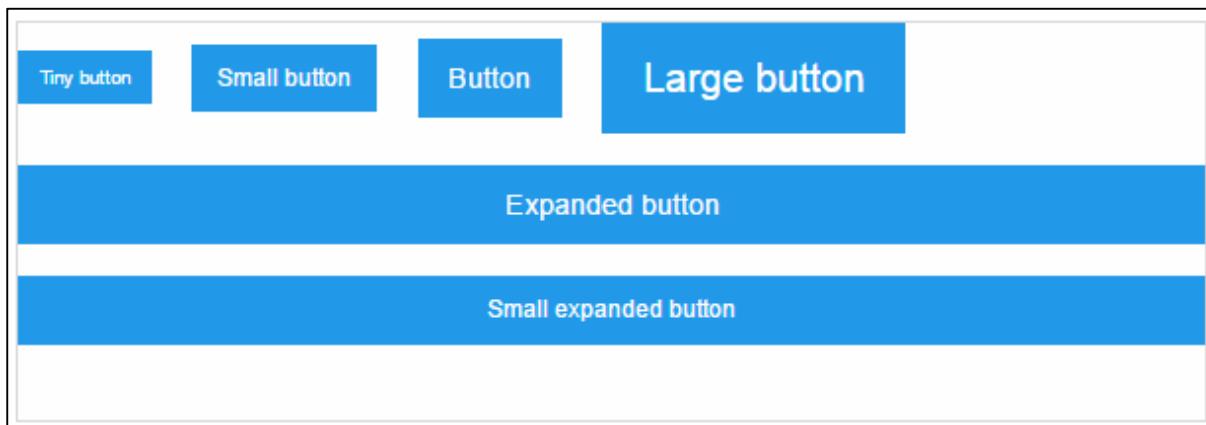
<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
</head>
<body>
  <a class="tiny button" href="#">Tiny button</a>
  <a class="small button" href="#">Small button</a>
  <a class="button" href="#">Button</a>
  <a class="large button" href="#">Large button</a>
  <a class="expanded button" href="#">Expanded button</a>
  <a class="small expanded button" href="#">Small expanded button</a>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
  <script>
    $(document).ready(function() {
      $(document).foundation();
    })
  </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **button_sizing.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Button Coloring

Description

Buttons can be colored to give them an additional meaning. Buttons can be disabled by using `.disabled` class, which displays the button as faded and it will not disable the control. If you want to disable the `<button>` elements, add `disabled` attribute to it. If you want to disable a link, add `aria-disabled` attribute.

Example

The following example demonstrates how to *color the buttons* in Foundation.

```

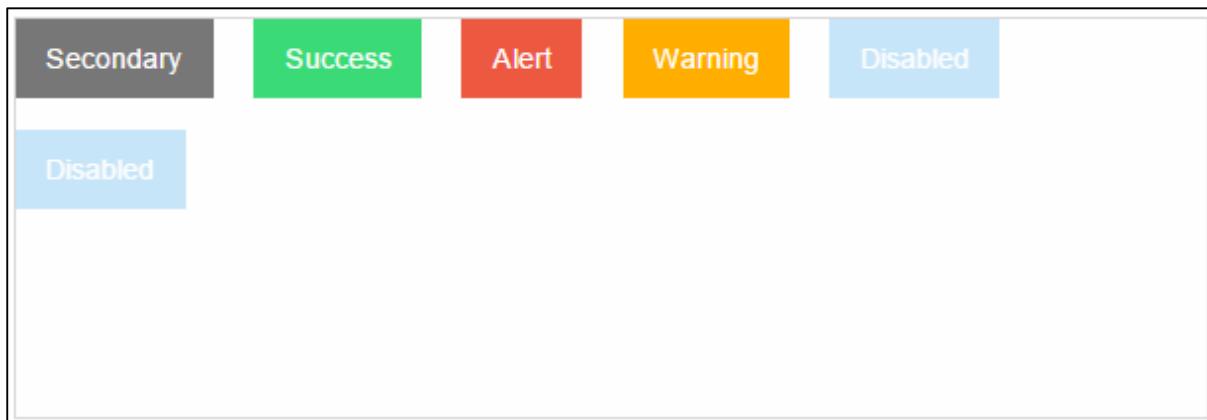
<html>
<head>
    <title>Button Coloring</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
</head>
<body>
    <a class="secondary button" href="#">Secondary</a>
    <button type="button" class="success button">Success</button>
    <button type="button" class="alert button">Alert</button>
    <a class="warning button" href="#">Warning</a>
    <a class="disabled button" href="#">Disabled</a>
    <button type="button" class="disabled button">Disabled</button>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
in.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.m
in.js"></script>
    <script>
        $(document).ready(function() {
            $(document).foundation();
        })
    </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **button_coloring.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Button Hollow Style

Description

Use `.hollow` class to give your button a hollow style appearance. To make a default style, change the `$button-fill` variable in your settings file.

Example

The following example demonstrates how to use *hollow styles* in Foundation.

```
<html>
<head>
    <title>Button Hollow Style</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
</head>
<body>
    <button class="hollow button" href="#">Primary</button>
    <button class="secondary hollow button" href="#">Secondary</button>
    <button class="success hollow button" href="#">Success</button>
    <button class="alert hollow button" href="#">Alert</button>
    <button class="warning hollow button" href="#">Warning</button>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
```

```

<script>
    $(document).ready(function() {
        $(document).foundation();
    })
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **button_hollow_style.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Button Dropdown Arrows

Description

You can use `.dropdown` class to add dropdown arrow to your button. It will not automatically add dropdown functionality to your button. To achieve this you have to attach the [Dropdown plugin](#).

Example

The following example demonstrates how to add *dropdown arrows to buttons* in Foundation.

```

<html>
<head>
    <title>Dropdown Arrows</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
    s">
</head>

```

180

```

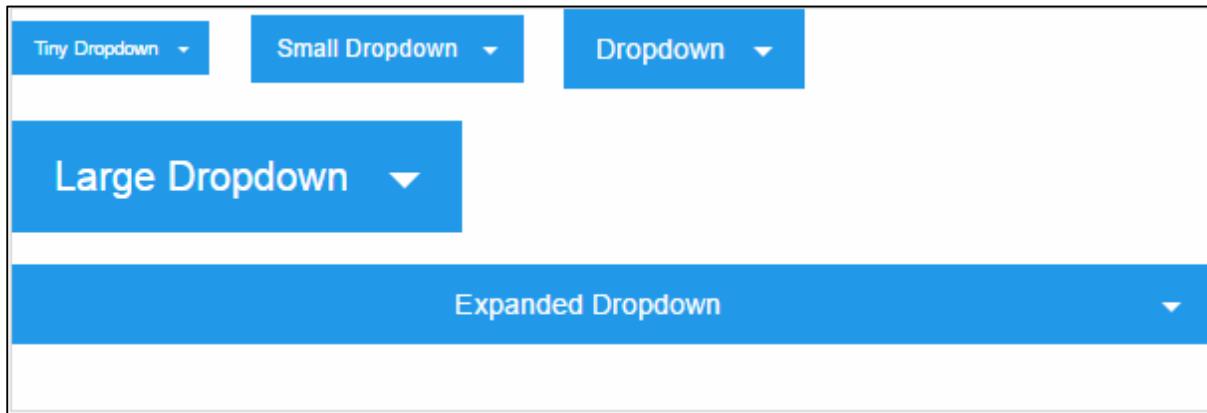
<body>
  <button class="tiny dropdown button">Tiny Dropdown</button>
  <button class="small dropdown button">Small Dropdown</button>
  <button class="dropdown button">Dropdown</button>
  <button class="large dropdown button">Large Dropdown</button>
  <button class="expanded dropdown button">Expanded Dropdown</button>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
  <script>
    $(document).ready(function() {
      $(document).foundation();
    })
  </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **button_dropdown_arrows.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Button Accessibility

Description

When your button does not contain readable text, you can add screen reader-only text to tell its purpose. The symbol or icon must be placed in an element with *aria-hidden*-

"true" attribute, which avoids the screen readers from attempting to pronounce the icon or symbol. To define screen readers-only text use `.show-for-sr` class.

Example

The following example demonstrates the use of *button accessibility* in Foundation.

```
<html>
<head>
    <title>Button Accessibility</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundicons/3.0.0/foundation-
    icons.css">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
    in.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
    js"></script>
</head>
<body>
    <p>When he reached
        <button class="button" type="button">
            <span class="show-for-sr">home</span>
            <!-- Screen readers will see "home" but Visual users will see the home
            icon, but not the "home" text -->
            <span aria-hidden="true"><i class="fi-home"></i></span>
        </button> the children were playing.
    </p>
    <p>Copy and paste the above sentence to see what it is.</p>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **button_accessibility.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

When he reached  the children were playing.

Copy and paste the above sentence to see what it is.

Foundation – Button SASS Reference

Description

You can change the styles of the components by using SASS Reference.

Variables

The following table lists the SASS variables in a project's settings file that makes this component's default styles to get customized.

S. No.	Name & Description	Type	Default Value
1	\$button-padding Padding within buttons.	List	0.85em 1em
2	\$button-margin Margin around buttons.	List	0 0 \$global-margin 0
3	\$button-fill Fill for buttons, by default. It can be either <i>solid</i> or <i>hollow</i> .	Keyword	solid
4	\$button-background Default background color of the buttons.	Color	\$primary-color
5	\$button-background-hover On hovering, background color of buttons	Color	scale-color(\$button-background, \$lightness: - 15%)
6	\$button-color Font color of the buttons.	List	#fff
7	\$button-color-alt Font color of the button, if the background is light.	List	#000
8	\$button-radius Represents border radius of buttons, defaulted to global-radius.	Number	\$global-radius

9	\$button-sizes Button sizes.	Map	tiny: 0.6rem small: 0.75rem default: 0.9rem large: 1.25rem
10	\$button-opacity-disabled Opacity of a button, which is disabled.	List	0.25

Mixins

To build this component's final CSS output, the following mixins can be used. To build your own class structure using Foundation components, you can use the mixins yourself.

button-expand

```
@include button-expand($expand);
```

It is used to expand a button to its full-width.

Parameter & Description	Type	Default Value
\$expand Set to <i>true</i> to enable expand or <i>false</i> otherwise.	Boolean	true

button-style

```
@include button-style($background, $background-hover, $color);
```

It sets the button's visual style.

S. No.	Parameter & Description	Type	Default Value
1	\$background Button's background color.	Color	\$button-background
2	\$background-hover On hover background color of the button. Set to <i>auto</i> will automatically generate a color for hover.	Color	\$button-background-hover
3	\$color Text color on button. Set to <i>auto</i> will automatically generate a color depending on background color.	Color	\$button-color

button-hollow

```
@include button-hollow;
```

It removes on hover background fill and hollow button focus.

button-disabled

```
@include button-disabled;
```

By fading the element, resetting cursor and disabling events of pointers, it adds disabled styles to a button.

button-dropdown

```
@include button-dropdown($size, $color, $offset);
```

It adds a dropdown arrow for a button.

S. No.	Parameter & Description	Type	Default Value
1	\$size It represents the size of the arrow. Use of <i>em</i> value is recommended so the triangle scales when used within different button sizes.	Number	0.4em
2	\$color Color of the arrow.	Color	white
3	\$offset Represents the distance between a button's text and arrow.	Number	\$button-padding

button

```
@include button($expand, $background, $background-hover, $color, $style);
```

It adds every style for a button.

S. No.	Parameter & Description	Type	Default Value
1	\$expand To make button full-width set it to <i>true</i> .	Boolean	false
2	\$background Button's background color.	Color	\$button-background
3	\$background-hover On hover, set the background color. Set to <i>auto</i> to make the mixin generate hover color automatically.	Color	\$button-background-hover
4	\$color Represents the button's text color. Set to <i>auto</i> to generate a color depending on background color automatically.	Color	\$button-color

5	\$style To create hollow button, set it to <i>hollow</i> . <i>\$background</i> color is used as button primary color.	Keyword	solid
---	---	---------	-------

Foundation – Button Group

Description

They are containers for corresponding action elements. It works fine when a group of actions is displayed in a bar.

The following table lists the button group in Foundation

S. No.	Button Group & Description
1	Basics Any number of buttons can be placed inside a container by using <i>.button-group</i> .
2	Sizing Sizing of the button groups can be done by using standard buttons.
3	Coloring Each button in a button group can be colored individually or every button can be colored by using the same class.
4	Even width Group The <i>.expanded</i> class can be used to make buttons full-width automatically.
5	Stacking Buttons can be stacked by using <i>.stacked</i> class or <i>.stack-for-small</i> class for small screen only.
6	Split Buttons To create split buttons, you must add <i>.arrow-only</i> class and create button groups of two buttons.
7	SASS Reference You can change the styles of the components by using SASS Reference.

Foundation – Button Group Basics

Description

Any number of buttons can be placed inside a container by using *.button-group*. Each button in the group is separated by a small gap.

Example

The following example demonstrates the use of *basic button group* in Foundation.

```

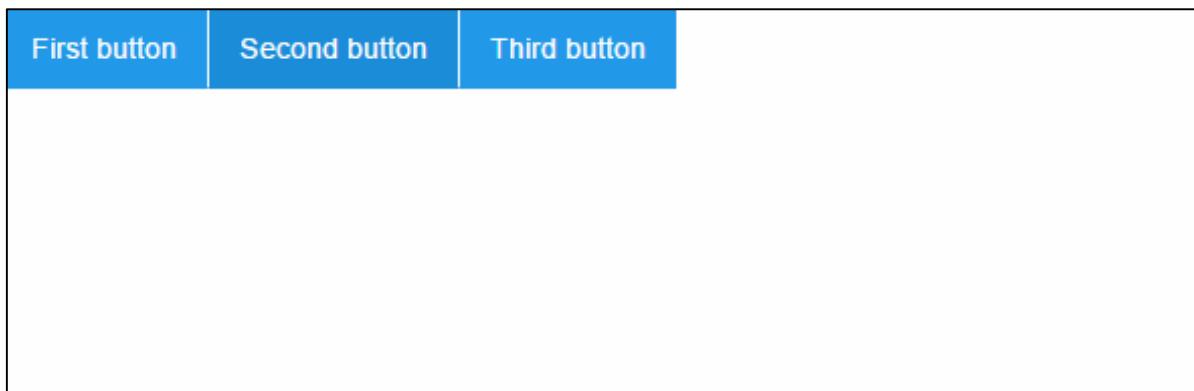
<html>
<head>
    <title>Button Group Basics</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
</head>
<body>
    <div class="button-group">
        <a class="button">First button</a>
        <a class="button">Second button</a>
        <a class="button">Third button</a>
    </div>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script>
        $(document).ready(function() {
            $(document).foundation();
        })
    </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **button_group_basics.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Button Group Sizing

Description

Sizing of the button groups can be done by using standard buttons such as `.tiny`, `small` and `large`.

Example

The following example demonstrates how to *size the button group* in Foundation.

```

<html>
  <head>
    <title>Button Group Sizing</title>
    <link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  </head>
  <body>
    <div class="small button-group">
      <a class="button">Small Button</a>
      <a class="button">Small Button</a>
      <a class="button">Small Button</a>
    </div>
    <script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script>
      $(document).ready(function() {
        $(document).foundation();
      })
    </script>
  </body>
</html>

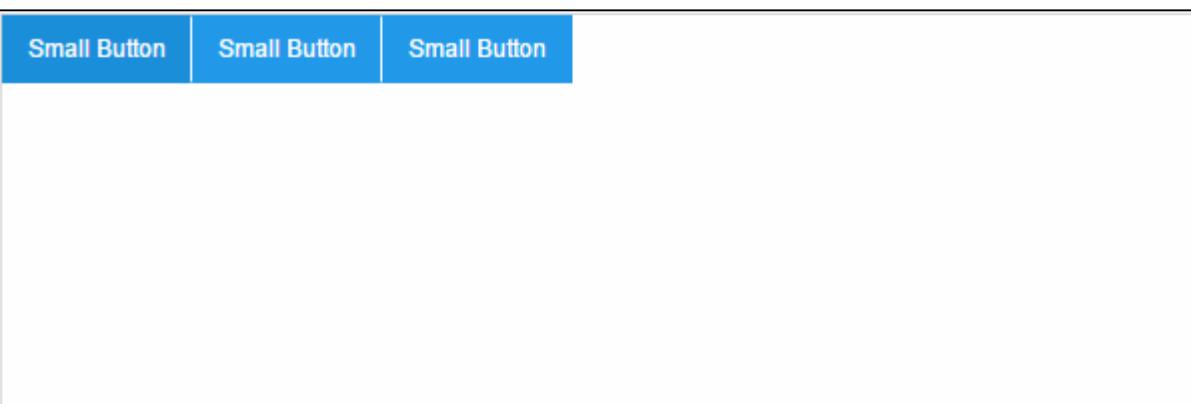
```

```
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **button_group_sizing.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Small Button Small Button Small Button

Foundation – Button Group Coloring

Description

Each button in the button group can be colored individually or every button can be colored by using the same class.

Example

The following example demonstrates how to *color button group* in Foundation.

```
<html>
<head>
    <title>Button Group Coloring</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
    s">
</head>
<body>
    <div class="button-group">
        <a class="secondary button">Secondary</a>
        <a class="success button">Success</a>
        <a class="warning button">Warning</a>
        <a class="alert button">Alert</a>
    </div>
</body>
</html>
```

```

</div>

<div class="success button-group">
  <a class="button">Button</a>
  <a class="button">With</a>
  <a class="button">Success</a>
  <a class="button">Class</a>
</div>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script>
$(document).ready(function() {
  $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **button_group_coloring.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Even width Group

Description

The `.expanded` class can be used to display the buttons to full-width automatically. Depending upon the number of buttons, they can be expanded to full-width; maximum of six buttons can be expanded.

Example

The following example demonstrates how to use *even width button group* in Foundation.

```
<html>
<head>
    <title>Even-width Group</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
</head>
<body>
    <div class="expanded button-group">
        <a class="button">One</a>
        <a class="button">Two</a>
        <a class="button">Three</a>
        <a class="button">Four</a>
        <a class="button">Five</a>
        <a class="button">Six</a>
    </div>
    <div class="expanded button-group">
        <a class="button">One</a>
        <a class="button">Two</a>
        <a class="button">Three</a>
        <a class="button">Four</a>
        <a class="button">Five</a>
        <a class="button">Six</a>
        <a class="button">Seven</a>
    </div>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script>
```

```

$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **button_group_even_width.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

One	Two	Three	Four	Five	Six
One	Two	Three	Four	Five	Six

Foundation – Button Stacking

Description

Buttons can be stacked by using *.stacked* class or *.stack-for-small* class for small screen only.

Example

The following example demonstrates how to *stack buttons* in Foundation.

```

<html>
<head>
    <title>Button Stacking</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
    s">
</head>
<body>
    <div class="stacked button-group">
        <a class="button">One</a>

```

```

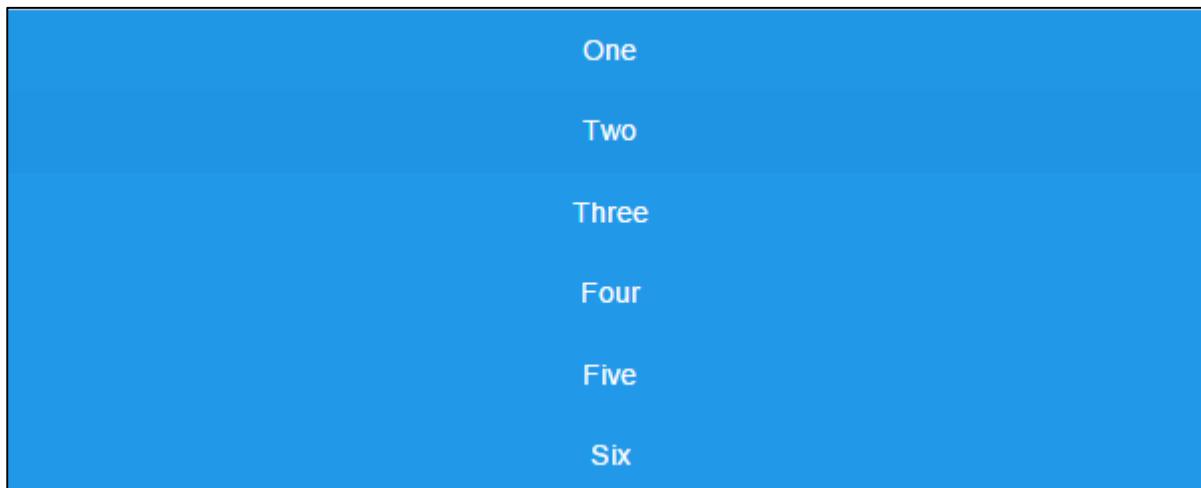
<a class="button">Two</a>
<a class="button">Three</a>
<a class="button">Four</a>
<a class="button">Five</a>
<a class="button">Six</a>
</div>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script>
$(document).ready(function() {
  $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **button_stacking.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Split Buttons

Description

To create split buttons, you must add `.arrow-only` class and create button groups of two buttons. You must use a label for screen readers by using `.show-for-sr`.

Example

The following example demonstrates how to use *split buttons* in Foundation.

```
<html>
<head>
    <title>Split Buttons</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundicons/3.0.0/foundation-
    icons.css">
</head>
<body>
    <div class="button-group">
        <a class="button">Shut Down</a>
        <a class="dropdown button arrow-only">
            <span class="show-for-sr"> Show options </span>
        </a>
    </div>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
    in.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
    js"></script>
    <script>
        $(document).ready(function() {
            $(document).foundation();
        })
    </script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **button_split.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Button Group SASS Reference

Description

You can change the styles of the components by using SASS Reference.

Variables

The following table lists the SASS variables in project's settings file that makes the component's default styles to get customized.

S. No.	Name & Description	Type	Default Value
1	\$buttongroup-margin It represents margin for button groups.	Number	1rem
2	\$buttongroup-spacing It represents margin between every buttons in button group.	Border	1px
3	\$buttongroup-child-selector It represents the selector for the buttons within a button group.	String	'.button'
4	\$buttongroup-expand-max It represents maximum buttons, which can be in button group with even-width.	Number	6

Mixins

To build this component's final CSS output, the following mixins can be used. To build your own class structure using Foundation components, you can use the mixins yourself.

button-group

```
@include button-group($child-selector);
```

It adds styles for a container of button group. The following table lists the parameter used for this purpose.

S. No.	Parameter & Description	Type	Default Value
1	\$child-selector It represents the selector for the buttons within a button group.	String	\$button-group-child-selector

button-group-expand

```
@include button-group-expand($count, $selector);
```

A full-width button group is created, which makes every button having equal width. It uses parameters, which are listed in the following table.

S. No.	Parameter & Description	Type	Default Value
1	\$count Represents number of buttons within the button group. Setting it to <i>auto</i> , will generate CSS that considers many number of buttons.	Keyword or Number	auto
2	\$selector Represents the selector for the buttons within a button group.	String	\$button-group-child-selector

button-group-stack

```
@include button-group-stack($selector);
```

It stacks the buttons in a group. The following table lists the parameter used for this purpose.

S. No.	Parameter & Description	Type	Default Value
1	\$selector Represents the selector for the buttons within a button group.	String	\$button-group-child-selector

button-group-unstack

```
@include button-group-unstack($selector);
```

It un-stacks the buttons in a group. The following table lists the parameter used for this purpose.

S. No.	Parameter & Description	Type	Default Value
1	\$selector Represents the selector for the buttons within a button group.	String	\$button-group-child-selector

Foundation – Close Button

Description

The Close button is used when you want something to go away by clicking on it. Close button is a `<button>` element, which uses the `.close-button` class. The multiplication symbol (`x`) is used as the X icon which will be enclosed within the `aria-hidden="true"` attribute, so the X icon will not be read by screen readers. To clarify the purpose of the button, it is labeled with `aria-label`.

Clicking on close button will not close the element, but it can be used with [Toggler](#), [Reveal](#), [Off-canvas](#) and other plugins that includes open/close actions.

Example

The following example demonstrates how to use *close button* in Foundation.

```

<html>
  <head>
    <title>Close Button</title>
    <link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  </head>
  <body>
    <div class="callout">
      <button class="close-button" aria-label="Close alert" type="button">
        <span aria-hidden="true">x</span>
      </button>
      <p>This is an example of close button!</p>
    </div>
    <script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script>
      $(document).ready(function() {
        $(document).foundation();
      })
    </script>
  </body>
</html>

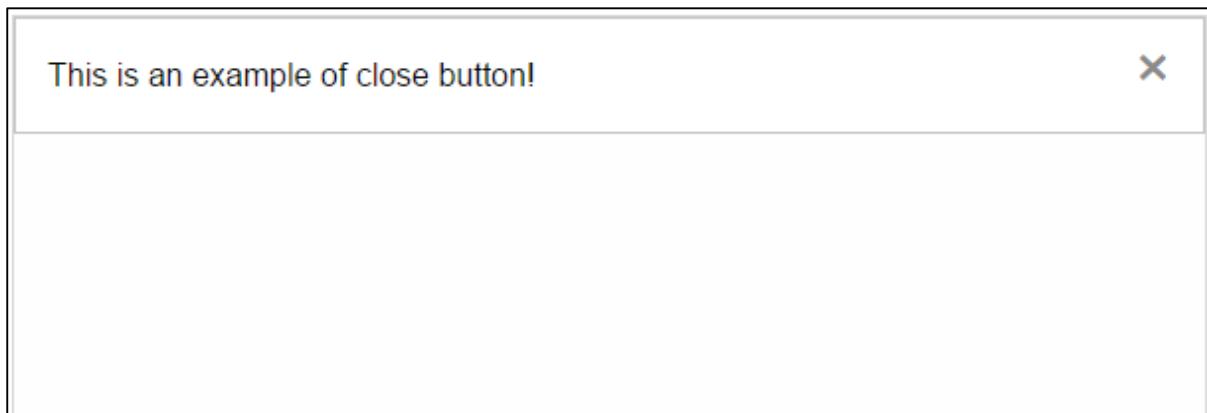
```

Output

197

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in foundation_close_button.html file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Button SASS Reference

Variables

The following table lists the SASS variables in the project's settings file that makes the component's default styles to get customized.

S. No.	Name & Description	Type	Default Value
1	\$closebutton-position It represents the close button default position. The first value and second value should be <i>right</i> or <i>left</i> and <i>top</i> or <i>bottom</i> respectively.	List	right top
2	\$closebutton-offset-horizontal It represents right (or left) offset for a close button.	Number	1rem
3	\$closebutton-offset-vertical It represents top (or bottom) offset for a close button.	Number	0.5rem
4	\$closebutton-size It represents the close button's default font size.	Number	2em
5	\$closebutton-lineheight It represents the close button's line-height, affecting the spacing of elements.	Number	1
6	\$closebutton-color It represents the close button's default color.	Color	\$dark-gray
7	\$closebutton-color-hover It represents the close button's default color when hovered.	Color	\$black

Mixins

To build this component's final CSS output, the following mixins can be used. To build your own class structure using Foundation components, you can use the mixins yourself.

CLOSE-BUTTON

```
@include close-button;
```

By using the styles in settings variables it adds close button's styles.

Foundation – Slider

Description

Sliders are very useful for setting certain values inside a range. The following table shows the various sliders used in Foundation.

S. No.	Sliders & Description
1	Basics The basic slider has a handle and an active fill.
2	Vertical It is used to display vertical slider by using <i>verticle</i> class.
3	Disabled By using <i>disabled</i> class you can disable the slider.
4	Two Handles To use two handles you need to use <i>.slider-handle</i> span and <i>input</i> field.
5	Data binding You can use this to display a visible input and a slider, so that changing one of them will match other one.
6	SASS Reference You can change the styles of the components by using SASS Reference.
7	JavaScript Reference Foundation provides JavaScript components for the sliders.

Foundation – Slider Basics

Description

The basic slider has a handle and an active fill. It uses the class *slider* and *data-initial-start* and *data-end* attributes.

Example

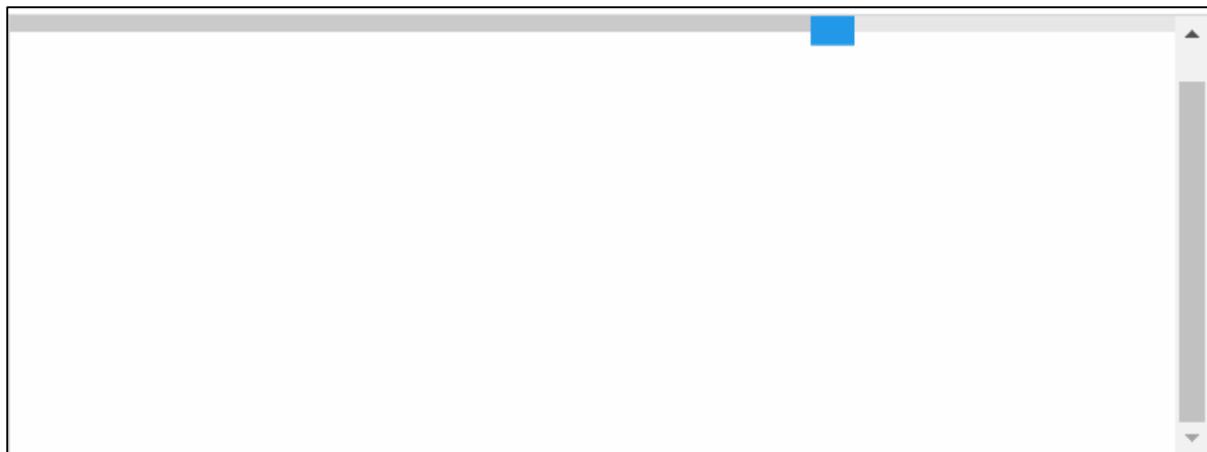
The following example demonstrates how to use *slider* in Foundation.

```
<html>
<head>
    <title>Slider Basics</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
</head>
<body>
    <div class="slider" data-slider data-initial-start="70" data-end="100">
        <span class="slider-handle" data-slider-handle role="slider"
        tabindex="1"></span>
        <span class="slider-fill" data-slider-fill></span>
        <input type="hidden">
    </div>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
in.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
js"></script>
    <script>
        $(document).ready(function() {
            $(document).foundation();
        })
    </script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **slider_basics.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Slider Vertical

Description

Vertical slider is used to display vertical slider by using *vertical* class and set *data vertical* to *true*.

Example

The following example demonstrates how to use *vertical slider* in Foundation.

```
<html>
<head>
    <title>Slider Vertical</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
</head>
<body>
    <div class="slider vertical" data-slider data-initial-start="25" data-
end="100" data-vertical="true">
        <span class="slider-handle" data-slider-handle role="slider"
tabindex="1"></span>
        <span class="slider-fill" data-slider-fill></span>
        <input type="hidden">
    </div>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
in.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
js"></script>
    <script>
        $(document).ready(function() {
```

```

$(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **slider_vertical.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Slider Disabled

Description

The *disabled* class is used to disable the slider. You need to add the *disabled* class to the *slider div*.

Example

The following example demonstrates how to use *disabled slider* in Foundation.

```

<html>
<head>
    <title>Slider Disabled</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
</head>
<body>
    <div class="slider disabled" data-slider data-initial-start="45">
        <span class="slider-handle" data-slider-handle role="slider"
tabindex="1"></span>

```

```

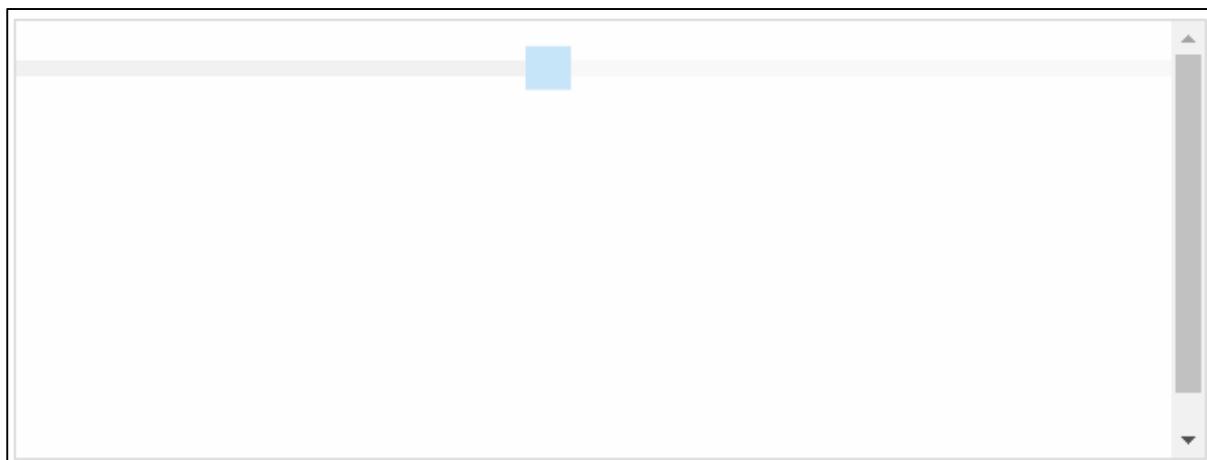
<span class="slider-fill" data-slider-fill></span>
<input type="hidden">
</div>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script>
$(document).ready(function() {
  $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **slider_disabled.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Slider Two Handles

Description

To use two handles you need to use *.slider-handle* span and *input* field. It works for both horizontal and vertical sliders. The JavaScript will allocate ids for input fields, until you change it. If you do not change, then the handles and inputs match in the way they are in the DOM tree.

Example

The following example demonstrates how to use *two handles slider* in Foundation.

```

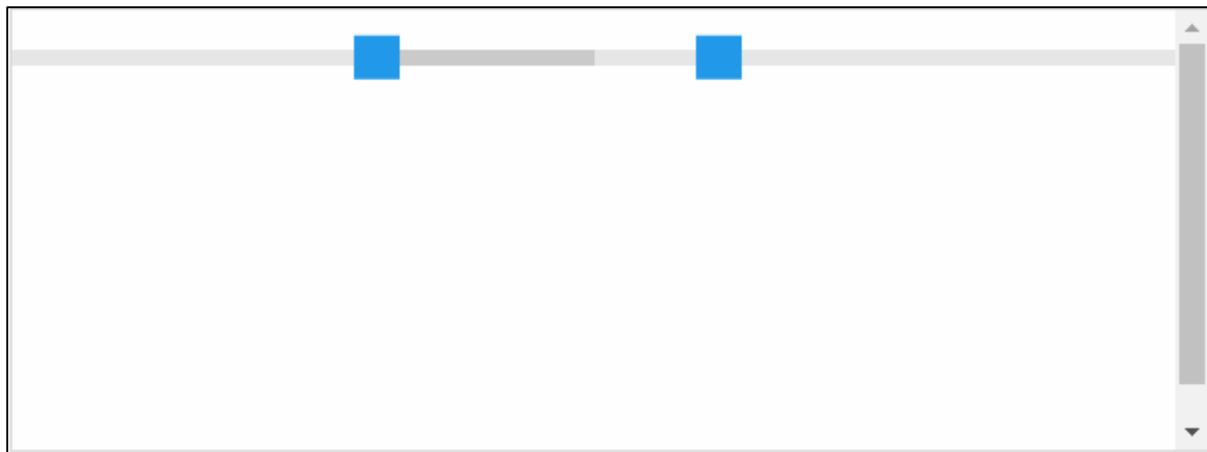
<html>
<head>
    <title>Slider Two Handles</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
</head>
<body>
    <div class="slider" data-slider data-initial-start="30" data-initial-
end="60">
        <span class="slider-handle" data-slider-handle role="slider"
tabindex="1"></span>
        <span class="slider-fill" data-slider-fill></span>
        <span class="slider-handle" data-slider-handle role="slider"
tabindex="1"></span>
        <input type="hidden">
        <input type="hidden">
    </div>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
in.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
js"></script>
    <script>
        $(document).ready(function() {
            $(document).foundation();
        })
    </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **slider_two_handles.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Slider Data Binding

Description

You can use this to display a visible input and a slider, so that changing one of them will match the other one. You have to assign `id` for `input` and add the `aria-controls="idOfInput"`.

Example

The following example demonstrates how to use *slider data binding* in Foundation.

```
<html>
<head>
    <title>Slider Data Binding</title>
    <link rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
</head>
<body>
    <div class="small-10 columns">
        <div class="slider" data-slider data-initial-start="30">
            <span class="slider-handle" data-slider-handle role="slider"
                tabindex="1" aria-controls="sliderOutput"></span>
            <span class="slider-fill" data-slider-fill></span>
        </div>
    </div>
    <div class="small-2 columns">
```

```

<input type="number" id="sliderOutput">
</div>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **slider_data_binding.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Slider SASS Reference

Description

You can change the styles of the components by using SASS Reference.

Variables

The following table lists the SASS variables in the project's settings file that makes the component's default styles to get customized.

S. No.	Name & Description	Type	Default Value
1	\$slider-height It represents slider's default height.	Number	0.5rem

2	\$slider-width-vertical It represents slider's default width of a vertical slider.	Number	\$slider-height
3	\$slider-background It represents slider's track default background color.	Color	\$light-gray
4	\$slider-fill-background It represents default color of the active fill color of the slider.	Color	\$medium-gray
5	\$slider-handle-height It represents slider handle's default height.	Number	1.4rem
6	\$slider-handle-width It represents slider handle's default width.	Number	1.4rem
7	\$slider-handle-background It represents slider handle's default color.	Color	\$primary-color
8	\$slider-opacity-disabled It represents disabled slider's default fade amount.	Number	0.25
9	\$slider-radius It represents slider's default radius.	Number	\$global-radius
10	\$slider-transition It represents transition properties to set to the slider handle and fill.	Transition	all 0.2s ease-in-out

Mixins

To build this component's final CSS output, the following mixins can be used. To build your own class structure using Foundation components, you can use the mixins yourself.

slider-container

```
@include slider-container;
```

It adds general sliders styles.

slider-fill

```
@include slider-fill;
```

It adds general styles for slider's active fill.

slider-handle

```
@include slider-handle;
```

It adds general styles for the handles of slider.

Foundation – Slider JavaScript Reference

Description

Foundation provides JavaScript components for the sliders.

Initializing

Along with the *foundation.core.js* plugin, you need to include *foundation.slider.js* in your JavaScript to make use of this plugin. This plugin requires the following listed utility libraries-

- **foundation.util.motion.js**
- **foundation.util.triggers.js**
- **foundation.util.keyboard.js**
- **foundation.util.touch.js**

Foundation.Slider

It is used to create a new instance of a drilldown menu.

```
var elem = new Foundation.Slider(element);
```

The following table lists the parameters used by *Foundation.slider*.

S. No.	Name & Description	Type
1	element jQuery object is used to make into an accordion menu.	jQuery
2	options Overrides the default settings of the plugin.	Object

Plugin Options

To customize a slider instance, Plugin options can be used. These options can be set as an object passed to the constructor of the plugin, individual data attributes or one combined *data-options* attribute. The following table shows how to initialize JavaScript plugins.

S. No.	Name & Description	Example
1	start Slider scale minimum value.	0

2	end Slider scale maximum value.	100
3	step It represents the minimum value change per change event. Not implemented currently.	
4	initialStart Value at which the left handle/first input must be set to on initialization.	0
5	initialEnd Value at which the right handle/second input must be set to on initialization.	100
6	binding It allows the input to be placed outside container and visible.	false
7	clickSelect It allows you to click or tap on the slider bar to choose a value.	true
8	vertical When set to <i>true</i> and use <i>vertical</i> class allows you to alter alignment to vertical.	false
9	draggable Allows you to drag slider handle(s) to choose a value.	true
10	disabled Prevents the event listeners from being applied and disables the slider. Crosschecked by JS with 'disabledClass'.	false
11	doubleSided It allows the use of two handles and cross checked by the JS.	false
12	decimal Represents how many decimal places the plugin must go for floating point precision.	2
13	moveTime It is the time to animate the movement of a slider handle when the user clicks on the bar. If updating the transition time in Sass settings, then it needs to be manually set.	200
14	disabledClass Used to disable the slider	'disabled'

Events

These events triggers from any element with an attached Slider plugin. The *moved.zf.slider* triggers when the handle movement is completed.

Functions

The *.destroy* is used to destroy the slider plugin.

Foundation – Switch

Switch allows you to turn off or turn on the switch by clicking on it. The following table lists the switch types used in Foundation.

S. No.	Sliders & Description
1	Basics By using <code>.switch</code> class you can create a switch in Foundation.
2	Radio Switch It can be used, when the user needs to select only one out of many options.
3	Sizing Classes You can change the size of the switches by using <code>.tiny</code> , <code>.small</code> or <code>.large</code> classes.
4	Inner Labels You can use inner labels to display active and inactive switch text.
5	SASS Reference You can change the styles of the components by using SASS Reference.

Foundation – Switch Basics

Description

- To create a switch, you need to add `.switch` class to an element.
- Within the switch add `<input type="checkbox">` with `.switch-input` class.
- Next, you need to create a `<label>` with `.switch-paddle` class.
- Provide a unique ID for `<input>` and point the `<label>` to `<input>` with `for` attribute, which makes the switch clickable.
- Within the switch label uses `.show-for-sr` class, which is screen reader-only text to mask the text visually.

Example

The following example demonstrates how to use `switch` in Foundation.

```

<html>
<head>
    <title>Switch</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">

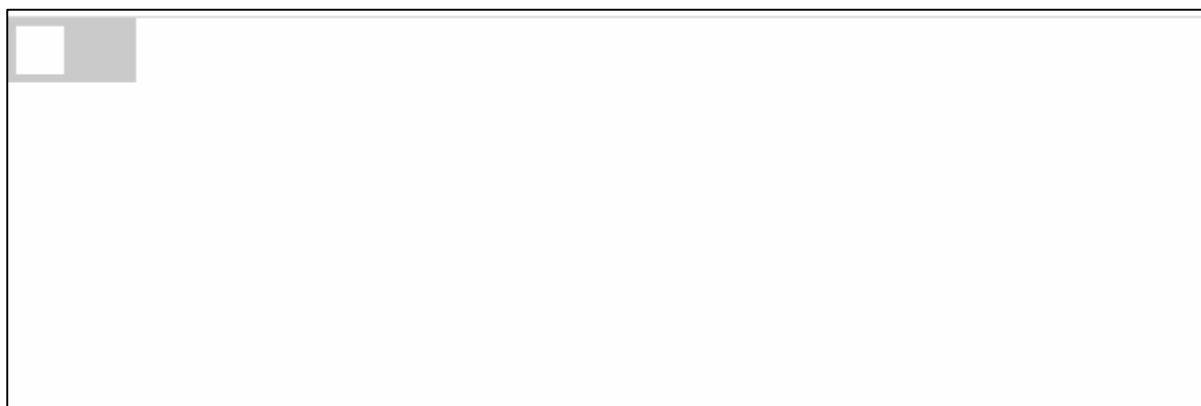
```

```
</head>
<body>
  <div class="switch">
    <input class="switch-input" id="demoSwitch" type="checkbox"
name="demoSwitch">
    <label class="switch-paddle" for="demoSwitch">
      <span class="show-for-sr">Switch</span>
    </label>
  </div>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
  <script>
    $(document).ready(function() {
      $(document).foundation();
    })
  </script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **switch_basics.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Radio Switch

Description

Radio switch is used to select only one option out of many. You can use `<input type="radio">` to create a list of options.

Example

The following example demonstrates how to use *radio switch* in Foundation.

```

<html>
<head>
    <title>Radio Switch</title>
    <link rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
</head>
<body>
    <div>
        <div class="switch">
            <input class="switch-input" id="exampleRadioSwitch1" type="radio"
                checked name="testGroup">
            <label class="switch-paddle" for="exampleRadioSwitch1">
                <span class="show-for-sr">Radio Switch 1</span>
            </label>
        </div>
        <div class="switch">
            <input class="switch-input" id="exampleRadioSwitch2" type="radio"
                checked name="testGroup">
            <label class="switch-paddle" for="exampleRadioSwitch2">
                <span class="show-for-sr">Radio Switch 2</span>
            </label>
        </div>
        <div class="switch">
            <input class="switch-input" id="exampleRadioSwitch3" type="radio"
                checked name="testGroup">
            <label class="switch-paddle" for="exampleRadioSwitch3">
                <span class="show-for-sr">Radio Switch 3</span>
            </label>
        </div>
    </div>

```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>

<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>

</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **radio_switch.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Sizing Classes

Description

You can change the size of the switches by using *.tiny*, *.small* or *.large* classes.

Example

The following example demonstrates how to change *size of switch* in Foundation.

```

<html>
<head>
    <title>Sizing Classes</title>

```

```

<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
</head>
<body>
  <div class="switch tiny">
    <input class="switch-input" id="tinySwitch" type="checkbox"
          name="demoSwitch">
    <label class="switch-paddle" for="tinySwitch">
      <span class="show-for-sr">Tiny Switch</span>
    </label>
  </div>

  <div class="switch small">
    <input class="switch-input" id="smallSwitch" type="checkbox"
          name="demoSwitch">
    <label class="switch-paddle" for="smallSwitch">
      <span class="show-for-sr">Small Switch</span>
    </label>
  </div>

  <div class="switch large">
    <input class="switch-input" id="largeSwitch" type="checkbox"
          name="demoSwitch">
    <label class="switch-paddle" for="largeSwitch">
      <span class="show-for-sr">Large Switch</span>
    </label>
  </div>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
  <script>
    $(document).ready(function() {
      $(document).foundation();
    })
  </script>
</body>

```

```
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in radio_switch_sizing_classes.html file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Inner Labels

Description

You can use inner labels to display active and inactive switch text. The `.switch-active` is used to indicate switch is on, and `.switch-inactive` to indicate switch is off. The active/inactive text is enclosed in `<label>` of the switch.

Example

The following example demonstrates the use of *inner label for switch* in Foundation.

```
<html>
<head>
    <title>Inner Labels</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
</head>
<body>
    <p>Would you like to follow us?</p>
    <div class="switch large">
        <input class="switch-input" id="inner_label" type="checkbox"
name="demoSwitch">
        <label class="switch-paddle" for="inner_label">
            <span class="show-for-sr">Would you like to follow us?</span>
    
```

```

<span class="switch-active" aria-hidden="true">Yes</span>
<span class="switch-inactive" aria-hidden="true">No</span>
</label>
</div>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **inner_labels.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Would you like to follow us?

No

Foundation – Switch SASS Reference

Description

You can change the styles of the components by using SASS Reference.

Variables

The following table lists the SASS variables in project's settings file that makes the component's default styles to get customized.

S. No.	Name & Description	Type	Default Value
1	\$switch-background It represents background color of a switch.	Color	\$medium-gray
2	\$switch-background-active It represents the switch's background active color.	Color	\$primary-color
3	\$switch-height It represents the switch's height, without any class applied.	Number	2rem
4	\$switch-height-tiny It represents the switch's height with <i>.tiny</i> class.	Number	1.5rem
5	\$switch-height-small It represents the switch's height with <i>.small</i> class.	Number	1.75rem
6	\$switch-height-large It represents the switch's height with <i>.large</i> class.	Number	2.5rem
7	\$switch-radius It represents switch's border radius.	Number	\$global-radius
8	\$switch-margin It represents border around a modal.	Number	\$global-margin
9	\$switch-paddle-background It represents background color for switch container and paddle.	Color	\$white
10	\$switch-paddle-offset It represents the spacing between edge of the body and switch paddle.	Number	0.25rem
11	\$switch-paddle-radius It represents the switch paddle's border radius.	Number	\$global-radius
12	\$switch-paddle-transition It represents switch transition.	Number	all 0.25s ease-out

Mixins

To build this component's final CSS output, the following mixins can be used. To build your own class structure using Foundation components, you can use the mixins yourself.

switch-container

```
@include switch-container;
```

It adds switch container's styles. Apply it to a container class.

switch-input

```
@include switch-input;
```

It adds switch input styles. Within a switch you have to apply it to an `<input>`.

switch-paddle

```
@include switch-paddle;
```

It adds styles for paddle and background of a switch. Within a switch you have to apply it to a `<label>`.

switch-text

```
@include switch-text;
```

Within a switch, it adds base styles for active or inactive text. You have to apply this to text elements within `<label>`.

switch-text-active

```
@include switch-text-active;
```

It adds styles for switch's active state text.

switch-text-inactive

```
@include switch-text-inactive;
```

It adds styles for switch's inactive state text.

switch-size

```
@include switch-size($font-size, $width, $height, $paddle-width, $paddle-offset);
```

By changing the size of the body and paddle, the `switch-size` alters the switch's size. You have to apply this to a container of the switch.

The following table lists the parameters that `switch-size` accepts.

S. No.	Parameter & Description	Type	Default Value
1	\$font-size It represents the label's font size inside the switch.	Number	1rem
2	\$width It represents width of the switch body.	Number	4rem

3	\$height It represents height of the switch body.	Number	2rem
4	\$paddle-width It represents width of the switch paddle.	Number	1.5rem
5	\$paddle-width It represents the spacing between the edge of the switch body and switch paddle.	Number	0.25rem

17. Foundation – Navigation

Description

Foundation provides a few different options for styling navigation elements. Many simple navigation patterns are bundled; it can be integrated in the form for robust responsive navigation solution.

The following table describes the different types of navigation along with description.

S. No.	Types & Description
1	Navigation Overview Navigation contains links to the other section and consists of many navigation patterns.
2	Menu Menu is used to build many navigation components.
3	Dropdown Menu The dropdown menu plugin is used to create submenus below the main menu.
4	Drilldown Menu The drilldown menu plugin is used to create submenus in main menu in the slider format.
5	Accordion Menu It displays the collapsible menu with accordion effects and provides support for auto collapse using the Accordion Menu plugin.
6	Top Bar Top bar helps to display the complex navigation bar easily on different size of the screen.
7	Responsive Navigation Responsive menu plugin allocates the menu in different sizes of the screen.
8	Magellan Magellan creates a navigation, which is in a fixed position; it automatically tracks the navigation list on a page, based on scroll position.
9	Pagination Pagination, an unordered list is handled by Bootstrap like a lot of other interface elements.
10	Breadcrumbs

	A breadcrumb specifies the current location for a site within navigational hierarchy.
--	---

Foundation – Navigation Overview

Description

Navigation contains links to other sections, it consist of many navigation patterns. Following are the various patterns of navigation available in Foundation.

Basic Menu

Menu is used to build many navigation components. It can be aligned in many ways i.e. vertical, horizontal or in a nested format. We will study about this in detail in the chapter [Menu](#)

Top Bar

Top bar is used to wrap all the menu. It helps to display the complex menu bar on all types of screen i.e. small, large and medium screen. We will study about this in detail in the chapter [Top Bar](#).

Menu Plugins

Following are the three types of menu plugins which create different styles for multiple menus by using the same markup.

Dropdown Menu

The dropdown menu plugin is used to create sub-menus below the main menu. The **data-dropdown-menu** helps in converting the nested menu into the list of dropdown menu. We will study about this in detail in the chapter [Dropdown Menu](#).

Drilldown Menu

The drilldown menu plugin is used to create sub menus in main menu in the slider format. The **data-drilldown** helps in converting the nested menu into the list of sliding menu. We will study about this in detail in the chapter [Drilldown Menu](#)

Accordion Menu

The accordion menu plugin is used to create sub menus in main menu in the collapsed manner. The **data-accordion-menu** helps in converting the nested menu into the list of collapsed menu. We will study about this in detail in the chapter [Accordion Menu](#)

Responsive Navigation

Responsive menu plugin allocates the menu in the different size of screen. **data-responsive-menu** is used to change the drill down menu into dropdown menu for large screen. We will study about this in detail in the chapter [Responsive Navigation](#).

Foundation – Navigation Menu

Description

Menu is used to build many navigation components. It can be aligned in many ways i.e. vertical, horizontal or in a nested format.

The following table describes the different types of menu along with the description.

S. No.	Types & Description
1	Basic Menu The menu is horizontally oriented by default.
2	Item Alignment Item alignment is to align the menu bar at the left or right side of the page.
3	Vertical Menu The menu can be displayed vertically.
4	Simple Style .simple class is included in a menu to create a simple menu bar.
5	Nested Style Nested style is creating sub-menus in the main menu.
6	Icons Menu items can also contain icons. Icons can be included by adding the element <i>.
7	Sass Reference Change the styles of the components by using SASS variables and mixins.

Foundation – Menu

Description

By default, the menu is horizontally oriented and <u>/> is filled with </i> element that can be used for all version of menus, which contains menu name and links. For linking the menu, <a> tag is used.

Example

The following example demonstrates the use of menu class in the Foundation:

```
<!DOCTYPE html>
<html>
<head>
<title>Foundation Template</title>
```

```

<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
</head>
<body>
<h2>Example of Menu</h2>
<ul class="menu">
<li><a href="#">Home</a></li>
<li><a href="#">About</a></li>
<li><a href="#">Service</a></li>
<li><a href="#">Contact</a></li>
</ul>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **menu.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Menu

[Home](#) [About](#) [Service](#) [Contact](#)

Foundation – Item Alignment

Description

Item alignment is to align the menu bar at the left or right side of the page. By default, the menu is aligned to the left side. The menu can be aligned to the right side by using the class `.align-right`.

Example

The following example demonstrates the use of item alignment in Foundation:

```
<!DOCTYPE html>
<html>
<head>
    <title>Foundation Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
        src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script
        src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
</head>
<body>
    <h2>Example of Menu</h2>
    <ul class="menu align-right">
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Service</a></li>
        <li><a href="#">Contact</a></li>
    </ul>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **item_alignment.htm** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Menu

Contact Service About Home

The items can be expanded in the remaining space. The *.expanded* class is used to expand the menu bar. It automatically expands the menu depending upon the space left in the row.

Example

The following example demonstrates the use of menu expanded in Foundation:

```
<!DOCTYPE html>
<html>
<head>
    <title>Foundation Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
</head>
<body>
    <h2>Example of Menu</h2>
    <ul class="menu expanded">
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Service</a></li>
```

```

<li><a href="#">Contact</a></li>
</ul>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **menu_expand.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Menu

Home About Service Contact

Foundation – Vertical Menu

Description

The menu can be displayed vertically by using the class *.vertical*.

Example

The following example demonstrates the use of vertical menu in the Foundation:

```

<!DOCTYPE html>
<html>
<head>
    <title>Foundation Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
in.js"></script>

```

```

</head>
<body>
    <h2>Example of Vertical Menu</h2>
    <ul class="menu vertical">
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Service</a></li>
        <li><a href="#">Contact</a></li>
    </ul>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **vertical_menu.htm** file.
- Open this HTML file in a browser, an output is displayed as shown below..

Example of Vertical Menu

Home
About
Service
Contact

Foundation – Simple Style

Description

Simple menu can be created without using padding and color. The *.simple* class is included in the menu to create a simple menu bar.

Example

The following example demonstrates the use of simple style in the Foundation:

```

<!DOCTYPE html>
<html>
<head>
    <title>Foundation Template</title>

```

```

<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
</head>
<body>
<h2>Example of Simple Style</h2>
<ul class="menu simple">
<li><a href="#">Home</a></li>
<li><a href="#">About</a></li>
<li><a href="#">Service</a></li>
<li><a href="#">Contact</a></li>
</ul>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **simple_style.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Simple Style

Home About Service Contact

Foundation – Nested Style

Description

Nested style means creating sub menus in the main menu. The nested menu can be created inside the `<i>` element of one particular menu by including the class `.nested`.

Example

The following example demonstrates the use of nested style in Foundation:

```
<!DOCTYPE html>
<html>
<head>
    <title>Foundation Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
        src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script
        src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
</head>
<body>
    <h2>Example of Nested Style</h2>
    <ul class="menu vertical">
        <li><a href="#">Home</a>
            <ul class="menu vertical nested">
                <li><a href="#">Home1</a></li>
                <li><a href="#">Home2</a></li>
            </ul>
        </li>
        <li><a href="#">About</a></li>
        <li><a href="#">Service</a></li>
        <li><a href="#">Contact</a></li>
    </ul>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **nested_style.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Nested Style

[Home](#)

[Home1](#)

[Home2](#)

[About](#)

[Service](#)

[Contact](#)

Foundation – Icons

Description

Menu items can also contain icons. Icons can be included by adding the element `<i>`. The icon can also be shifted above the text menu by including the class `icon-top` to the menu.

Example

The following example demonstrates the use of icons in menu in Foundation:

```
<!DOCTYPE html>
<html>
<head>
    <title>Foundation Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundicons/3.0.0/foundation-
    icons.css">

    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
    js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
    in.js"></script>
</head>
```

```
<body>
  <h2>Example of Icons</h2>
  <ul class="menu">
    <li><a href="#"><i class="fi-home"></i>Home</a></li>
    <li><a href="#"><i class="fi-book"></i>About</a></li>
    <li><a href="#"><i class="fi-pencil"></i>Service</a></li>
    <li><a href="#"><i class="fi-wrench"></i>Contact</a></li>
  </ul>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **icons.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Example

The following example demonstrates the use of icons on top in menu in Foundation:

```
<!DOCTYPE html>
<html>
<head>
  <title>Foundation Template</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
  s">
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundicons/3.0.0/foundation-
  icons.css">
```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
</head>
<body>
<h2>Example of Icons</h2>
<ul class="menu icon-top">
<li><a href="#"><i class="fi-home"></i>Home</a></li>
<li><a href="#"><i class="fi-book"></i>About</a></li>
<li><a href="#"><i class="fi-pencil"></i>Service</a></li>
<li><a href="#"><i class="fi-wrench"></i>Contact</a></li>
</ul>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **icon_top.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Icons



Foundation – Menu SASS Reference

Variables

You can change the styles of the components by using the following SASS variables as listed in the table.

S. No.	Name & Description	Type	Default Value
--------	--------------------	------	---------------

1	\$menu-margin Sets margin in menu.	Number	0
2	\$menu-margin-nested Sets margin for left side in a nested menu.	Number	1rem
3	\$menu-item-padding Sets padding in the menu.	Number	0.7rem 1rem
4	\$menu-icon-spacing Sets space between the text and icon in the menu item.	Number	0.25rem
5	\$menu-expand-max Includes maximum number of <i>expand-n</i> classes in the CSS.	Number	6

Mixins

You can use the mixins to build the CSS class structure for your components as listed in the table.

menu-base

Base style is created for menu.

```
@include menu-base;
```

menu-expand

Expand the menu item at the same width.

```
@include menu-expand($count);
```

Parameter	Type	Default Value	Description
\$count	Keyword or Number	auto	Number of items inside the menu.

menu-direction

Menu direction is set.

```
@include menu-direction($dir);
```

Parameter	Type	Default Value	Description
\$dir	Keyword	horizontal	Sets menu direction horizontally or vertically.

menu-simple

Simple menu is created without padding or hover state.

```
@include menu-simple;
```

menu-nested

Styles are included for the nested menu. Include **margin-left** to the menu.

```
@include menu-nested($padding);
```

Parameter	Type	Default Value	Description
\$padding	Keyword or Number	auto	Sets the margin length.

menu-icons

Icons are included to menu item.

```
@include menu-icons($position, $base);
```

Parameter	Type	Default Value	Description
\$position	Keyword	side	Set position for the icons.
\$base	Boolean	true	To call mixin multiple times on same element then set it as false.

Foundation – Navigation Dropdown Menus

Description

The dropdown menu plugin is used to create submenus below the main menu. The *data-dropdown-menu* attribute helps in converting the nested menu into the list of dropdown menu.

The following table describes the different types of dropdown menus along with the description.

S. No.	Features & Description
1	Horizontal Set the dropdown by including the attribute <code>data-dropdown-menu</code> and class <code>dropdown</code> to the menu container.
2	Vertical

	The menu can be displayed vertically by including the class vertical to the top level menu.
3	Sass Reference Change the styles of the components by using SASS variables.
4	JavaScript Reference Foundation provides JavaScript components for a dropdown menu.

Foundation – Horizontal Dropdown Menus

Description

Set the dropdown by including the attribute *data-dropdown-menu* and class *dropdown* to the menu container. Inside the **** tag of **** you can create one more new tag **** to create many levels of dropdown.

Example

The following example demonstrates the use of Horizontal dropdown menu in Foundation:

```
<!DOCTYPE html>
<html>
<head>
    <title>Foundation Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.js">
</script>
</head>
<body>
<h2>Example of Dropdown menu</h2>
<ul class="menu dropdown" data-dropdown-menu>
    <li><a href="#">Home</a>
        <ul class="menu">
            <li><a href="#">Home 1</a></li>
        </ul>
    </li>
</ul>

```

```

<li><a href="#">Home 2</a></li>
<li><a href="#">Home 3</a>
    <ul class="menu">
        <li><a href="#">Home3 1</a></li>
        <li><a href="#">Home3 2</a></li>
        <li><a href="#">Home3 3</a></li>
    </ul>
</li>
</ul>
</li>
<li><a href="#">About</a></li>
<li><a href="#">Service</a></li>
<li><a href="#">Contact</a></li>
</ul>

<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **dropdown_horizontal.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Dropdown menu

Home ▾ About Service Contact

Foundation – Vertical Dropdown Menus

Description

The menu can be displayed vertically by including the class `.vertical` to the top level menu. The sub menus are automatically aligned vertically.

Example

The following example demonstrates the use of Vertical dropdown menu in Foundation:

```
<!DOCTYPE html>
<html>
<head>
  <title>Foundation Template</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.js">
</script>
</head>
<body>
<h2>Example of Dropdown menu</h2>
<ul class="vertical dropdown menu" data-dropdown-menu>
  <li><a href="#">Home</a>
    <ul class="menu">
      <li><a href="#">Home 1</a></li>
      <li><a href="#">Home 2</a>
        <ul class="menu">
          <li><a href="#">Home2 1</a></li>
          <li><a href="#">Home2 2</a></li>
        </ul>
      </li>
    </ul>
  </li>
  <li><a href="#">About</a></li>
  <li><a href="#">Service</a></li>
```

```

<li><a href="#">Contact</a></li>
</ul>

<script>
$(document).ready(function() {
  $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **dropdown_vertical.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Dropdown menu

Home

About

Service

Contact

Foundation – Dropdown SASS reference

Variables

You can change the styles of the components by using the following SASS variables as listed in the table.

S. No.	Name & Description	Type	Default Value
1	\$dropdownmenu-arrows Arrows are enabled for items with dropdown menus.	Boolean	true
2	\$dropdownmenu-min-width Sets minimum width of the dropdown sub-menus	Length	200px

3	\$dropdownmenu-background Sets background color for dropdown menu.	color	\$white
4	\$dropdownmenu-border Sets border for dropdown panes.	List	1px solid \$medium-gray

Foundation – Dropdown JavaScript reference

Foundation provides JavaScript components for a dropdown menu as listed below.

Initializing

You can initialize the dropdown menu in the JavaScript by using `foundation.dropdownMenu.js` and `foundation.core.js` plugins. The plugin requires the following libraries:

- **foundation.util.keyboard.js**
- **foundation.util.box.js**
- **foundation.util.nest.js**

Foundation.DropdownMenu

It specifies instance of a dropdown menu as defined below:

```
var elem = new Foundation.DropdownMenu(element);
```

S. No.	Name & Description	Type
1	Element It creates a jQuery object into a dropdown menu.	jQuery
2	Options Default plugin setting is override.	Object

Plugin Options

You can use the following plugin to customize the dropdown menu instance. You can set the plugin option as individual data attributes.

S. No.	Name	Description	Example
--------	------	-------------	---------

1	disableHover	Hover event is disabled from opening submenus.	false
2	autoclose	Automatically, the sub menus are closed when the mouse leaves the event.	true
3	hoverDelay	It delays the time while opening the submenus on hover event.	50
4	clickOpen	The submenus are allowed to remain open on parent click event.	true
5	closingTime	It delays the time while closing the submenus on mouseleave event.	500
6	alignment	Depending upon the menu, the direction is set to open the submenus.	'left'
7	closeOnClick	The open submenus are closed by a click on the body.	true
8	verticalClass	To set the menu vertical, the class <i>vertical</i> is used.	'vertical'
9	rightClass	Set the menu at the right side.	'align-right'
10	forceFollow	To perform default action, the Boolean is force to override links of clicking on second touch event for mobile.	false

Events

The dropdown menus plugin attached to any element can trigger the following event.

S. No.	Name	Description
1	show.zf.dropdown menus	It triggers an event when new dropdown pane is visible.
2	hide.zf.dropdown menus	It triggers an event when the open menus are closed.

Functions

.destroy

The plugin is destroyed.

Foundation – Navigation Drilldown Menu

Description

The drilldown menu plugin is used to create submenus in main menu in the slider format. The *data-drilldown* helps in converting the nested menu into the list of sliding menu.

The following table describes the features of drilldown menu along with description.

S. No.	Features & Description
1	Basic Standard menu syntax is used for the drilldown menu.
2	JavaScript Reference Foundation provides JavaScript components for a drilldown menu.

Foundation – Navigation Drilldown Basic

Description

Standard menu syntax is used for the drilldown menu. The attribute *data-drilldown* is included into the root ** and *.menu* for every nested menu.

Example

The following example demonstrates the use of drilldown dropdown menu in Foundation:

```
<!DOCTYPE html>
<html>
<head>
    <title>Foundation Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.js"></script>
</head>
<body>
    <h2>Example of Drilldown menu</h2>

```

```

<ul class="vertical menu" data-drilldown>
  <li><a href="#">Home</a>
    <ul class="vertical menu">
      <li><a href="#">Home 1</a></li>
      <li><a href="#">Home 2</a>
        <ul class="vertical menu">
          <li><a href="#">Home2 1</a></li>
          <li><a href="#">Home2 2</a></li>
        </ul>
      </li>
    </ul>
  </li>
  <li><a href="#">About</a></li>
  <li><a href="#">Service</a></li>
  <li><a href="#">Contact</a></li>
</ul>
<script>
$(document).ready(function() {
  $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **drilldown_basic.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Drilldown menu

[Home](#)

[About](#)

[Service](#)

[Contact](#)

Foundation – Drilldown JavaScript Reference

Foundation provides JavaScript components for a drilldown menu as listed below.

Initializing

You can initialize the drilldown menu in JavaScript by using the *foundation.drilldown.js* and *foundation.core.js* plugins. The plugin requires the following libraries:

- **foundation.util.keyboard.js**
- **foundation.util.motion.js**
- **foundation.util.nest.js**

Foundation.Drilldown

It specifies instance of a drilldown menu as defined below:

```
var elem = new Foundation.Drilldown(element);
```

S. No.	Name	Description	Type
1	element	It creates a jQuery object into a drilldown menu.	jQuery
2	options	Default plugin setting is override.	Object

Plugin Options

You can use following plugin to customize the drilldown menu instance. You can set the plugin option as individual data attributes.

S. No.	Name	Description	Example
1	backButton	<i>js-drilldown-back</i> class is required. For JS generated back button, markup is used.	<\li><\a>Back
2	wrapper	<i>is-drilldown</i> class is required for independent styling.	<\div class="is-drilldown"></div>
3	closeOnClick	Clicking on the body it helps the menu to return to root list.	false

Events

The drilldown menus plugin attached to any element can trigger the following event-

S. No.	Name	Description

1	closed.zf.drilldown menu	It triggers an event when the menu is completely closed.
2	hide.zf.drilldown menus	It triggers an event when the open submenus are closed.

Functions

Following are the functions used in drilldown menu.

.hideAll

It closes all elements that are opened, and returns back to root menu.

.back

In each *back* button the event listener is included.

S. No.	Name	Type	Description
1	\$elem	jQuery	<i>back</i> event is included in the current sub-menu.

.show

Submenus are opened.

S. No.	Name	Type	Description
1	\$elem	jQuery	Open the current submenus.

.hide

Submenu is hided.

S. No.	Name	Type	Description
1	\$elem	jQuery	Hides the current submenu

.destroy

The drilldown menu is destroyed.

Foundation – Navigation Accordion Menu

Description

It displays the collapsible menu with accordion effects and provides support for auto collapse using Accordion Menu plugin.

S. No.	Features & Description
1	Basic accordion menu It creates collapsing division in the menu.
2	SASS Reference Change the styles of the components by using SASS variables.
3	JavaScript Reference Foundation provides JavaScript components for a accordion menu

Foundation – Accordion Menu

Description

It displays the collapsible menu with accordion effects and provides support for auto collapse.

Example

The following example demonstrates the use of *accordion menu* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Accordion Menu</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Accordion Menu Example</h2>
  <ul class="vertical menu" data-accordion-menu>
    <li><a href="#">Two Wheeler</a>
      <ul class="menu vertical nested is-active">
        <li><a href="#">Honda</a>
```

```

<ul class="menu vertical nested">
    <li><a href="#">CB Shine</a></li>
    <li><a href="#">Unicorn</a></li>
    <li><a href="#">Dream Yuga</a></li>
</ul>
</li>
<li><a href="#">TVS</a></li>
<li><a href="#">Bajaj</a></li>
</ul>
</li>
<li><a href="#">Four Wheeler</a>
    <ul class="menu vertical nested">
        <li><a href="#">Mahindra</a></li>
        <li><a href="#">Ford</a></li>
    </ul>
</li>
<li><a href="#">Heavy Weight Vehicle</a></li>
</ul>

<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **accordion_menu.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Accordion Menu Example

Two Wheeler

Honda

CB Shine

Unicorn

Dream Yuga

TVS

Bajaj

Four Wheeler

Heavy Weight Vehicle

Foundation – Accordion Menu SASS Reference

Variables

You can change the styles of the components by using the following SASS variables as listed in the table-

S. No.	Name	Description	Type	Default Value
1	\$accordionmenu-arrows	Accordion menus are set if they have default arrow styles.	Boolean	true

Foundation – Accordion Menu JavaScript Reference

Foundation provides JavaScript components for an accordion menu as listed below.

Initializing

You can initialize the accordion menu in JavaScript by using `foundation.accordionmenu.js` and `foundation.core.js` plugins. The plugin requires the following libraries:

- **foundation.util.keyboard.js**
- **foundation.util.motion.js**
- **foundation.util.nest.js**

Foundation.AccordionMenu

It specifies instance of a accordion menu as defined below:

```
var elem = new Foundation.AccordionMenu(element);
```

S. No.	Name	Type	Description
1	element	jQuery	It creates a jQuery object into a accordion menu.
2	options	Object	Default plugin setting is override.

Plugin Options

You can use the following plugin to customize the accordion menu instance. You can set the plugin option as individual data attributes.

S. No.	Name	Description	Example
1	slideSpeed	Set the time in ms(millisecond) to animate the opening of a submenu.	250
2	multiOpen	Multiple open panes are allowed in the menu.	true

Events

The accordion menus plugin attached to any element can trigger the following event.

S. No.	Name	Description
1	down.zf.accordion menu	It triggers an event when the menu is collapsed up.
2	up.zf.accordion menu	It triggers an event when the menu is collapsed down.

Functions

Following are the functions used in accordion menu.

.hideAll

It closes all panes of the menu that are opened.

.toggle

It toggles on submenu open/close state.

S. No.	Name	Type	Description
1	\$target	jQuery	Toggle the submenu.

.down

The submenus are opened defined by **\$target**.

S. No.	Name	Type	Description
1	\$target	jQuery	Opens the submenu.

.up

The submenus are closed defined by **\$target**. It also closes the submenus that are present inside the target.

S. No.	Name	Type	Description
1	\$target	jQuery	Closes the submenus.

.destroy

The accordion menu is destroyed.

Foundation – Navigation Top Bar

Description

Top bar helps to display the complex navigation bar easily on different sizes of screen.

The following table describes the features of top bar along with description.

S. No.	Features & Description
1	Basic Top Bar It is used to create a navigation header in the website.
2	SASS Reference Change the styles of the components by using SASS variables and mixins.

Foundation – Top Bar

Description

It is used to create navigation header in the website.

Example

The following example demonstrates the use of *top bar* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Top Bar</title>
```

249

```

<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Top Bar Example</h2>
<div class="top-bar">
  <div class="top-bar-left">
    <ul class="dropdown menu" data-dropdown-menu>
      <li class="menu-text">Foundation</li>
      <li><a href="#">Home</a></li>
      <li><a href="#">About Us</a></li>
      <li class="has-submenu">
        <a href="#">Services</a>
        <ul class="submenu menu vertical" data-submenu>
          <li><a href="#">Service1</a></li>
          <li><a href="#">Service2</a></li>
          <li><a href="#">Service3</a></li>
        </ul>
      </li>
      <li><a href="#">Contact Us</a></li>
    </ul>
  </div>
  <div class="top-bar-right">
    <ul class="menu">
      <li><input type="search" placeholder="Search Here"></li>
      <li><button type="button" class="button">Search</button></li>
    </ul>
  </div>
</div>
<script>
$(document).ready(function() {
  $(document).foundation();
})
</script>

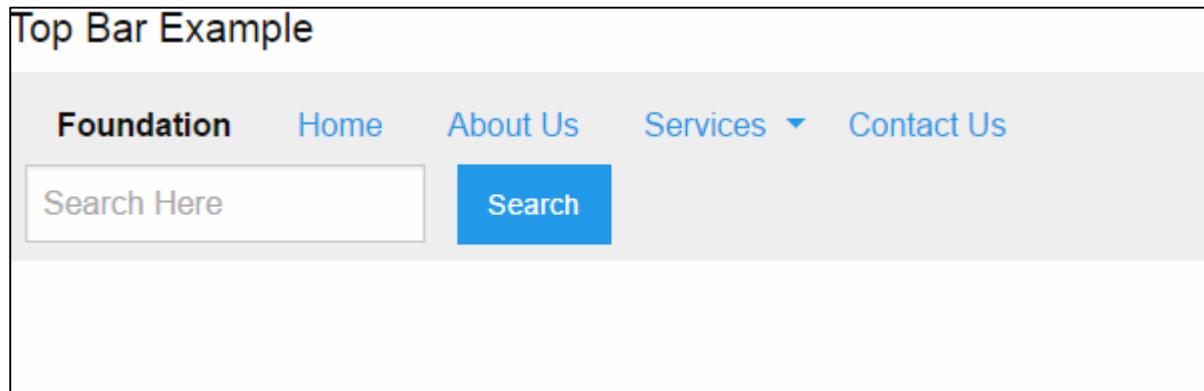
```

```
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **top_bar.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Top Bar SASS Reference

Variables

You can change the styles of the components by using the following SASS variables as listed in the table.

S. No.	Name	Description	Type	Default Value
1	\$topbar-padding	Sets padding for top bar.	Number	0.5em
2	\$topbar-background	Background color is set for top bar.	Color	\$light-gray
3	\$topbar-link-color	Sets color for link that is present inside the top bar menu.	Color	\$primary-color
4	\$topbar-input-width	Set width in the top bar for <input> element.	Number	200px

Mixins

You can use the mixins to build the CSS class structure for your components as listed in the table.

top-bar-container

Includes styles in the top bar container.

```
@include top-bar-container;
```

top-bar-stacked

Makes the top bar menu stacked.

```
@include top-bar-stacked;
```

Foundation – Responsive Navigation

Description

Responsive menu plugin allocates the menu in the different size of screen. *data-responsive-menu* is used to change the drill down menu into dropdown menu for large screen.

The following table lists some of the Responsive navigations:

S. No.	Features & Description
1	Responsive Menu Using responsive menu plugin, a default pattern can be applied to menu.
2	Responsive Toggle The toggle helps to display the menu on the small screen size.

Foundation – Responsive Navigation Menu

Description

To change the menu to different screen size, the menu consists some responsive CSS classes. Using responsive menu plugin, a default pattern can be applied to menu, you can also change the menu pattern for different size of screen.

Example

The following example demonstrates the use of responsive navigation menu in Foundation:

```
<!DOCTYPE html>
<html>
<head>
  <title>Foundation Template</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
```

```

<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.js"></script>
</head>
<body>
<h2> Example of Responsive Navigation menu</h2>
<ul class="menu dropdown" data-responsive-menu="drilldown medium-dropdown">
    <li><a href="#">Bike</a>
        <ul class="menu">
            <li><a href="#">Bajaj</a></li>
            <li><a href="#">Suzuki</a></li>
            <li><a href="#">Honda</a>
                <ul class="menu">
                    <li><a href="#">Activa</a></li>
                    <li><a href="#">Dio</a></li>
                    <li><a href="#">Passion</a></li>
                </ul>
            </li>
        </ul>
    </li>
    <li><a href="#">Car</a></li>
    <li><a href="#">Service</a></li>
    <li><a href="#">Contact</a></li>
</ul>

<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **responsive_menu.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Responsive Navigation menu

Bike Car Service Contact

Foundation – Responsive Navigation Toggle

Description

data-responsive-toggle attribute is used for menu toggling. The toggle helps to display the menu on the small screen size.

Example

The following example demonstrates the use of responsive navigation menu in Foundation:

```
<!DOCTYPE html>
<html>
<head>
    <title>Foundation Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.js">
</script>

</head>
<body>
<h2> Example of Responsive Navigation menu</h2>
<div class="title-bar" data-responsive-toggle="example-menu" data-hide-for="medium">
    <button class="menu-icon" type="button" data-toggle></button>
    <div class="title-bar-title">Menu</div>
</div>
<div id="example-menu">
    <ul class="menu dropdown" data-dropdown-menu>
        <li><a href="#">Bike</a>
            <ul class="menu">
                <li><a href="#">Bajaj</a></li>
                <li><a href="#">Suzuki</a></li>
                <li><a href="#">Honda</a>
                    <ul class="menu">
                        <li><a href="#">Activa</a></li>
                        <li><a href="#">Dio</a></li>
                        <li><a href="#">Passion</a></li>
                    </ul>
                </li>
            </ul>
        </li>
        <li><a href="#">Car</a></li>
        <li><a href="#">Service</a></li>
        <li><a href="#">Contact</a></li>
    </ul>
</div>

<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **responsive_toggle.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Responsive Navigation menu



Foundation – Navigation Magellan

Description

Magellan creates a navigation, which is in a fixed position; it automatically tracks the navigation list on a page based on the scroll position.

The following table describes the features of Magellan along with description.

S. No.	Features & Description
1	Setup <i>data-magellan</i> attribute is included in the container.
2	Sticky Navigation Sticky plugin can be included with magellan to create a constant navigation header.
3	JavaScript Reference Foundation provides JavaScript components for Magellan.

Foundation – Magellan Setup

Description

data-magellan attribute is included in the container. It can be used with any navigation element like menu; unique *id* has to be provided for every section that has to be linked to the navigation of your page.

Example

The following example demonstrates the use of magellan setup in Foundation:

```

<!DOCTYPE html>
<html>
<head>
    <title>Foundation Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.js">
</script>
</head>
<body>
    <h2>Example of Magellan Setup</h2>
    <ul class="horizontal menu" data-magellan>
        <li><a href="#home">Home</a></li>
        <li><a href="#about">About Us</a></li>
        <li><a href="#service">Service</a></li>
    </ul>

    <div class="sections">
        <section id="home">
            <h3>Home</h3>
            <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p>
            <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p>
            <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p>
            <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p>
        </section>
    </div>

```

```

<section id="about">
    <h3>About</h3>
    <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p>
    <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p>
    <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p>
    <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p>
</section>
<section id="service">
    <h3>Service</h3>
    <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p>
    <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p>
    <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p>
</section>
</div>
<script>
    $(document).ready(function() {
        $(document).foundation();
    })
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **magellan_setup.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Magellan Setup

[Home](#) [About Us](#) [Service](#)

Home

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

About

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Service

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Foundation – Magellan Sticky Navigation

Description

Sticky plugin can be included with magellan to create a constant navigation header. The following example works well on the older version of Foundation.

Example

The following example demonstrates the use of magellan sticky navigation in Foundation:

```
<!DOCTYPE html>
<html>
<head>
    <title>Foundation Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/5.5.2/css/foundation.css">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/jquery/2.0.3/jquery.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/5.5.2/js/foundation.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/modernizr/2.8.3/modernizr.js"></script>
</head>
<body>
<div data-magellan-expedition="fixed">
    <section class="top-bar-section">
        <div class="sticky-mag">
            <ul class="horizontal menu" data-magellan>
                <li data-magellan-arrival="home"><a href="#home">Home</a></li>
                <li data-magellan-arrival="about"><a href="#about">About</a></li>
                <li data-magellan-arrival="service"><a href="#service">Service</a></li>
            </ul>
        </div>
    </section>
</div>
<br/><br/>
```

```

<h2>Example of Magellan Sticky Navigation</h2>
<h3 data-magellan-destination="home">Home</h3>
<a name="home"></a>

<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p>

<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p>

<h3 data-magellan-destination="about">About</h3>
<a name="about"></a>

<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p>

<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p>

<h3 data-magellan-destination="service">Service</h3>
<a name="service"></a>

<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p>

<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p>

<script>
$(document).ready(function() {
  $(document).foundation();
})
</script>
</body>
</html>

```

Output

261

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **magellan_sticky_navigation.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

[Home](#)[About](#)[Service](#)

Example of Magellan Sticky Navigation

Home

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

About

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Service

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Foundation – Magellan JavaScript Reference

Foundation provides JavaScript components for Magellan as listed below.

Initializing

You can initialize the Magellan in JavaScript by using *foundation.magellan.js* and *foundation.core.js* plugins.

Foundation.Magellan

It specifies instance of a Magellan as defined below:

```
var elem = new Foundation.Magellan(element);
```

S. No.	Name	Type	Description
1	element	jQuery	It creates a jQuery object into a Magellan.
2	options	Object	Default plugin setting is override.

Plugin Options

You can use the following plugin to customize the Magellan instance. You can set the plugin option as individual data attributes.

S. No.	Name	Description	Example
1	animationDuration	Set the time in ms while scrolling between the locations.	500
2	animationEasing	Use animation style, while scrolling between the locations.	ease-in-out
3	threshold	For location changes, use number of pixels as a marker.	50
4	activeClass	In magellan container, apply the class for active link location.	active
5	deepLinking	Script allows to manipulate the current page URL	true
6	barOffset	Set number of pixels to offset when using sticky nav bar.	25

Events

The Magellan plugin attached to any element can trigger the following event-

S. No.	Name	Description
1	update.zf.magellan	It triggers an event when the magellan is finished updating to the new active element.

Functions

Following are the functions used in Magellan.

.calcPoints

The array of pixel values are calculated lines of distinction between the locations on a page.

.reflow

To update Magellan, necessary functions are called upon DOM change.

.destroyA

The instance of magellan is destroyed.

Foundation – Navigation Pagination

Description

Pagination, an unordered list is handled by Foundation like a lot of other interface elements.

The following table describes the features of Pagination along with description.

S. No.	Features & Description
1	Basic It is a navigation type that divides the content into a series of related pages.
2	Centered Pagination list is created at the center of the page.
3	SASS Reference Change the styles of the components by using SASS variables and mixins.

Foundation – Pagination

Description

It is a navigation type that divides the content into a series of related pages.

Example

The following example demonstrates the use of *pagination* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Pagination</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Pagination Example</h2>
  <ul class="pagination">
    <li class="disabled">Previous</li>
    <li class="current">1</li>
    <li><a href="#">2</a></li>
    <li><a href="#">3</a></li>
    <li><a href="#">4</a></li>
    <li class="ellipsis"></li>
    <li><a href="#">12</a></li>
    <li><a href="#">13</a></li>
    <li><a href="#">Next</a></li>
  </ul>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **pagination.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Pagination Example

[Previous](#) [Next](#)

Foundation – Centered Pagination

Description

Including the class `.text-center` or `text-align: center` in CSS, you can make the pagination list at the center of the page.

Example

The following example demonstrates the use of `pagination` in Foundation:

```
<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Pagination</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
    <h3>Pagination Centered Example</h3>
    <ul class="pagination text-center">
        <li class="disabled">Previous</li>
        <li class="current">1</li>
        <li><a href="#">2</a></li>
        <li><a href="#">3</a></li>
    
```

```

<li><a href="#">4</a></li>
<li class="ellipsis"></li>
<li><a href="#">12</a></li>
<li><a href="#">13</a></li>
<li><a href="#">Next</a></li>
</ul>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **pagination_centered.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Pagination Centered Example

[Previous](#) [Next](#)

Foundation – Pagination SASS Reference

Variables

You can change the styles of the components by using the following SASS variables as listed in the table.

S. No.	Name	Type	Default Value
1	\$pagination-font-size Sets font size of pagination.	Number	rem-calc(14)
2	\$pagination-margin-bottom Sets default margin for bottom of media object.	Number	\$global-margin
3	\$pagination-item-color Sets color of the text of pagination item.	Color	\$black

4	\$pagination-item-padding Sets padding inside the pagination item.	Number	rem-calc(3 10)
5	\$pagination-item-spacing Sets margin at the right to keep space in pagination item.	Number	rem-calc(1)
6	\$pagination-radius Sets Default radius	Number	\$global-radius
7	\$pagination-item-background-hover Sets background color on hover	Color	\$light-gray
8	\$pagination-item-background-current Sets background color for current page of pagination item	Color	\$primary-color
9	\$pagination-item-color-current Sets the text color for the current page	color	foreground(\$pagination-item-background-current)
10	\$pagination-item-color-disabled Sets text color for the disabled pagination item.	Color	\$medium-gray
11	\$pagination-ellipsis-color Sets ellipsis color	Color	\$black
12	\$pagination-mobile-items Displays only previous/next button in mobile screen.	Boolean	false
13	\$pagination-arrows Arrows are included in the previous and next links of pagination	Boolean	true

Mixins

You can use the mixins to build the CSS class structure for your components as listed in the table.

pagination-container

Styles are included in the pagination container. It is applied to

```
@include pagination-container;
```

pagination-item-current

Styles are included for the current pagination item. It is applied to <a>

```
@include pagination-item-current;
```

pagination-item-disabled

Styles are included for a disabled pagination. It is applied to <a>

```
@include pagination-item-disabled;
```

pagination-ellipsis

Styles are included for ellipsis for use in a pagination list.

```
@include pagination-ellipsis;
```

Foundation – Navigation Breadcrumbs

Description

A breadcrumb specifies the current location for a site within navigational hierarchy. It is useful when accessing large number of the pages of a website. A breadcrumb in Foundation is simply an unordered list with a class of *.breadcrumb*. For detailed information check [this chapter](#).

SASS Reference

Variables

You can change the styles of the components by using the following SASS variables as listed in the table.

S. No.	Name & Description	Type	Default Value
1	\$breadcrumbs-margin Around the breadcrumb container, the margin is set.	Number	0 0 \$global-margin 0
2	\$breadcrumbs-item-font-size Sets font size for breadcrumb item.	Number	rem-calc(11)
3	\$breadcrumbs-item-color Sets color for breadcrumb item.	Color	\$primary-color
4	\$breadcrumbs-item-color-current Sets color for current breadcrumb item.	Color	\$black
5	\$breadcrumbs-item-color-disabled The breadcrumb item is disabled.	Number	\$medium-gray

6	\$breadcrumbs-item-margin Sets margin between the breadcrumb items.	Number	0.75rem
7	\$breadcrumbs-item-uppercase Makes the breadcrumb item in uppercase.	Boolean	true
8	\$breadcrumbs-item-slash Includes slash between the breadcrumb items.	Boolean	true

Mixin

You can use the mixins to build the CSS class structure for your components as listed in the table:

breadcrumbs-container

It includes styles for the breadcrumb container with the style for element ` <a>` present inside it.

```
@include breadcrumbs-container;
```

18. Foundation – Containers

Foundation containers are used to make full width of the browsers at all times for the site and wrap the site contents.

The following table lists some of the containers used in Foundation:

S. No.	Containers & Description
1	Accordion Accordions contain vertical tabs, which are used on websites to expand and collapse large amount of data.
2	Callout It places the content inside the component.
3	Dropdown-panes It displays the content when you click the button.
4	Media Object It adds media objects such as images, videos, blogs comments, etc. along with some content.
5	Off-canvas It sets off the navigation menu from the visible area and displays the main content.
6	Reveal-Modal Foundation allows creating modal dialogs or pop-up windows using <i>reveal</i> class.
7	Tables Foundation provides layouts for displaying data in tabular format.
8	Tabs It is a navigation based tab that displays the content into different panes without leaving the page.

Foundation – Accordion

Description

Accordions contain vertical tabs, which are used on websites to expand and collapse large amount of data. It makes the collapsing divisions of the page and the content is broken into logical sections.

The following table lists some of the Accordion features:

S. No.	Features & Description
1	Accordion It creates collapsing division on the webpage.
2	Multi-expand It is used to open multiple panes of an accordion at a time.
3	All Closed It will close all the panes of an accordion.
4	Sass Reference You can change the styles of the components by using SASS variables.
5	JavaScript Reference Foundation provides JavaScript components for an accordion.

Foundation – Basic Accordion

Description

It creates collapsing division on the webpage, which hides and shows large amount of content.

Example

The following example demonstrates the use of *basic accordion* in Foundation:

```
<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Accordion</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
```

```

<body>
  <h2>Basic Accordion Example</h2>
  <ul class="accordion" data-accordion role="tablist">
    <li class="accordion-item is-active">
      <a href="#collapse1" role="tab" class="accordion-title" id="collapse1-heading">Foundation</a>
      <div id="collapse1" class="accordion-content" role="tabpanel" data-tab-content aria-labelledby="panel1d-heading">
        The most advanced responsive front-end framework.
      </div>
    </li>

    <li class="accordion-item">
      <a href="#collapse2" role="tab" class="accordion-title" id="collapse2-heading">Bootstrap</a>
      <div id="collapse2" class="accordion-content" role="tabpanel" data-tab-content aria-labelledby="panel1d-heading">
        Bootstrap is the most popular and powerful front-end framework.
      </div>
    </li>

    <li class="accordion-item">
      <a href="#collapse3" role="tab" class="accordion-title" id="collapse3-heading">HTML</a>
      <div id="collapse3" class="accordion-content" role="tabpanel" data-tab-content aria-labelledby="panel1d-heading">
        HTML is a markup language for describing web documents.
      </div>
    </li>
  </ul>
  <script>
    $(document).ready(function() {
      $(document).foundation();
    })
  </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **basic_accordion.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

The screenshot shows a web page titled "Basic Accordion Example". Inside, there's a list of three items: "Foundation" (which is expanded, showing its content), "Bootstrap" (which has a "+" sign to its right, indicating it can be expanded), and "HTML" (which also has a "+" sign to its right). A vertical scrollbar is present on the right side of the page.

Foundation – Multi Expand Accordion

Description

It is used to open multiple panes of accordion at a time by setting *data-multi-expand* to true.

Example

The following example demonstrates the use of *multi-expand accordion* in Foundation:

```
<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Accordion</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
```

```

</head>
<body>
<h2>Multi Expand Accordion Example</h2>
<ul class="accordion" data-accordion data-multi-expand="true">
    <li class="accordion-item is-active">
        <a href="#collapse1" role="tab" class="accordion-title" id="collapse1-heading">Foundation</a>
        <div id="collapse1" class="accordion-content" role="tabpanel" data-tab-content aria-labelledby="panel1d-heading">
            The most advanced responsive front-end framework.
        </div>
    </li>

    <li class="accordion-item">
        <a href="#collapse2" role="tab" class="accordion-title" id="collapse2-heading">Bootstrap</a>
        <div id="collapse2" class="accordion-content" role="tabpanel" data-tab-content aria-labelledby="panel1d-heading">
            Bootstrap is the most popular and powerful front-end framework.
        </div>
    </li>

    <li class="accordion-item">
        <a href="#collapse3" role="tab" class="accordion-title" id="collapse3-heading">HTML</a>
        <div id="collapse3" class="accordion-content" role="tabpanel" data-tab-content aria-labelledby="panel1d-heading">
            HTML is a markup language for describing web documents.
        </div>
    </li>
</ul>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **multi-expand.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

The screenshot shows a web page titled "Multi Expand Accordion Example". It features three accordions. The first accordion, labeled "Foundation", has the text "The most advanced responsive front-end framework.". The second accordion, labeled "Bootstrap", has the text "Bootstrap is the most popular and powerful front-end framework.". The third accordion, labeled "HTML", has the text "HTML is a markup language for describing web documents.". Each accordion has a minus sign icon to its right, indicating it is currently closed. A vertical scrollbar is visible on the right side of the page.

Foundation – Closed Accordion

Description

It will close all the panes of an accordion by setting *data-allow-all-closed* to true.

Example

The following example demonstrates the use of *all closed accordion* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Accordion</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Closed Accordion Example</h2>
<ul class="accordion" data-accordion data-allow-all-closed="true">
    <li class="accordion-item is-active">
        <a href="#collapse1" role="tab" class="accordion-title" id="collapse1-heading">Foundation</a>
        <div id="collapse1" class="accordion-content" role="tabpanel" data-tab-content aria-labelledby="panel1d-heading">
            The most advanced responsive front-end framework.
        </div>
    </li>

    <li class="accordion-item">
        <a href="#collapse2" role="tab" class="accordion-title" id="collapse2-heading">Bootstrap</a>
        <div id="collapse2" class="accordion-content" role="tabpanel" data-tab-content aria-labelledby="panel1d-heading">
            Bootstrap is the most popular and powerful front-end framework.
        </div>
    </li>

    <li class="accordion-item">
        <a href="#collapse3" role="tab" class="accordion-title" id="collapse3-heading">HTML</a>
        <div id="collapse3" class="accordion-content" role="tabpanel" data-tab-content aria-labelledby="panel1d-heading">
            HTML is a markup language for describing web documents.
        </div>
    </li>
</ul>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **all_closed.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Closed Accordion Example

Foundation -

The most advanced responsive front-end framework.

Bootstrap +

HTML +

Foundation – Sass Reference

Variables

You can change the styles of the components by using the following SASS variables as listed in the table:

S. No.	Name & Description	Type	Default Value
1	\$accordion-background It defines the default background color of an accordion group.	Color	\$white
2	\$accordion-plusminus You can add plus and minus icons to the side of each accordion title by setting it to true.	Boolean	true
3	\$accordion-item-color It displays the default text color for items in a menu.	Color	foreground(\$accordion-background, \$primary-color)
4	\$accordion-item-background-hover Default background color on hover for items in a Menu.	Color	\$light-gray
5	\$accordion-item-padding Default padding of an accordion item.	Number or List	1.25rem 1rem

6	\$accordion-content-background Provides background color of the tab content.	Color	\$white
7	\$accordion-content-border Provides border color of the tab content.	Color	1px solid \$light-gray
8	\$accordion-content-color It specifies the text color of tab content.	Color	foreground(\$accordion-background, \$primary-color)
9	\$accordion-content-padding Displays the padding for tab content.	Number or List	1rem

Mixins

You can use the mixins to build the CSS class structure for your components as listed in the table.

S. No.	Name & Description	Format
1	accordion-container It includes the styles for an accordion container.	@include accordion-container
2	accordion-title It defines the title for an accordion item.	@include accordion-title
3	accordion-content It provides styles for the accordion content.	@include accordion-content

Foundation – JavaScript Reference

Foundation provides JavaScript components for an accordion as listed below.

Initializing

You can initialize the accordion in JavaScript by using `foundation.accordion.js` and `foundation.core.js` plugins. The `foundation.core.js` plugin requires following libraries:

- **foundation.util.keyboard.js**
- **foundation.util.motion.js**

Foundation.Accordion

It specifies instance of an accordion as defined below:

```
var my_element = new Foundation.Accordion(element);
```

The accordion includes one event as specified below:

S. No.	Name & Description	Type
1	element It creates a jQuery object into an accordion.	jQuery

Plugin Options

You can use the following plugin instances to customize the accordion instances.

S. No.	Name & Description	Example
1	slideSpeed It specifies the time taken to open an accordion.	200
2	multiExpand It allows you to open multiple panes in an accordion.	false
3	allowAllClosed It will close all the panes of an accordion.	false

Events

Accordion plugin provides the following events as listed in the table:

S. No.	Name & Description
1	down.zf.accordion It triggers an event when you open the tab.
2	up.zf.accordion It triggers an event when the tab is collapsing up.

Events

Accordion plugin provides the following functions as listed below:

.toggle

It toggles the on and off state of the content pane. It contains the following event:

S. No.	Name & Description	Type
1	\$target It creates jQuery object for toggling of pane.	jQuery

.down

It fires when you open the accordion tab. It includes the following events:

S. No.	Name & Description	Type
1	\$target It opens the accordion pane.	jQuery
2	firstTime It specifies re-execution of the accordion.	Boolean

.up

It triggers when the tab is closed. It contains the following event:

S. No.	Name & Description	Type
1	\$target It closes the accordion tab.	jQuery

.destroy

It ruins the instance of an accordion.

Foundation – Callout

Description

Callouts combine panels and alerts from Foundation 5 into one generic container component.

The following table lists some of the Callout features:

S. No.	Feature & Description
1	Basic Callout It is used to put the content inside.
2	Coloring You can color the callouts by using <code>.secondary</code> , <code>.primary</code> , <code>.success</code> , <code>.warning</code> or <code>.alert</code> classes.
3	Sizing You can make the size of callouts small and large by using <code>.small</code> and <code>.large</code> classes.
4	Making Closable Callouts can be closed by using the <code>data-closable</code> attribute.
5	Sass Reference

You can change the styles of the components by using SASS variables.

Foundation – Basic Callout

Description

It is used to put the content inside by using the `.callout` class.

Example

The following example demonstrates the use of *basic callout* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Basic Callout</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Basic Callout Example</h2>
  <div class="callout">
    <h5>This is basic callout</h5>
    <a href="#">Foundation is a responsive front-end framework.</a>
  </div>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **basic_callout.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Basic Callout Example

This is basic callout

[Foundation is a responsive front-end framework.](#)

Foundation – Callout Coloring

Description

You can color the callouts by using the `.secondary`, `.primary`, `.success`, `.warning` or `.alert` classes.

Example

The following example demonstrates the use of *coloring* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Coloring Callout</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Coloring Callout Example</h2>
  <div class="callout secondary">
    <h5>secondary callout</h5>
    <a href="#">Foundation is a responsive front-end framework.</a>
  </div>
  <div class="callout success">
```

```
<h5>success callout</h5>
<a href="#">Foundation is a responsive front-end framework.</a>
</div>
<div class="callout warning">
  <h5>warning callout</h5>
  <a href="#">Foundation is a responsive front-end framework.</a>
</div>
<div class="callout alert">
  <h5>alert callout</h5>
  <a href="#">Foundation is a responsive front-end framework.</a>
</div>
<div class="callout primary">
  <h5>primary callout</h5>
  <a href="#">Foundation is a responsive front-end framework.</a>
</div>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **coloring.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Coloring Callout Example

secondary callout

Foundation is a responsive front-end framework.

success callout

Foundation is a responsive front-end framework.

warning callout

Foundation is a responsive front-end framework.

alert callout

Foundation is a responsive front-end framework.

primary callout

Foundation is a responsive front-end framework.

Foundation – Callout Sizing

Description

You can make the size of callouts small and large by using `.small` and `.large` classes respectively.

Example

The following example demonstrates the use of *sizing* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Sizing Callout</title>
```

```
<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Sizing Callout Example</h2>
<div class="callout small">
    <h5>This is small callout</h5>
    <a href="#">Foundation is a responsive front-end framework.</a>
</div>
<div class="callout large">
    <h5>This is large callout</h5>
    <a href="#">Foundation is a responsive front-end framework.</a>
</div>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **sizing.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Sizing Callout Example

This is small callout

Foundation is a responsive front-end framework.

This is large callout

Foundation is a responsive front-end framework.

Foundation – Callout Closable

Description

Callout can be closed by using the *data-closable* attribute to an element, which creates the dismissable alert box.

Example

The following example demonstrates the use of *closable callout* in Foundation:

```
<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Closable Callout</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.5/css/foundation.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/motion-
    ui/1.1.1/motion-ui.css"/>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.5/js/vendor/jquery.m
    in.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.5/js/foundation.min.
    js"></script>
</head>
<body>
```

```

<h2>Closable Callout Example</h2>
<div class="alert callout" data-closable>
  <h5>This is Closable Callout</h5>
  <p>Foundation is a responsive front-end framework.</p>
  <button class="close-button" data-close aria-label="Dismiss alert">x</button>
</div>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **making_closable.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Closable Callout Example

This is Closable Callout ×

Foundation is a responsive front-end framework.

Foundation – Callout SASS Reference

Variables

You can change the styles of the components by using the following SASS variables as listed in the table.

S. No.	Name & Description	Type	Default Value
1	\$callout-background It defines the default background color.	Color	\$white
2	\$callout-background-fade It specifies the fade value for callout backgrounds.	Number	85%
3	\$callout-border It displays border style for callouts.	List	1px solid rgba(\$black, 0.25)

4	\$callout-margin It displays callouts bottom margin.	Number	0 0 1rem 0
5	\$callout-padding It provides callouts inner padding.	Number	1rem
6	\$callout-font-color It defines font color of the callouts.	Color	\$body-font-color
7	\$callout-font-color-alt It defines font color of the callouts if there is a dark background for callouts.	Color	\$body-background
8	\$callout-radius It defines border radius for callouts.	Color	\$global-radius
9	\$callout-link-tint It provides tint links for colored panels and can be disable by setting it to false.	Number or Boolean	30%

Mixins

You can use the mixins to build the CSS class structure for your components as described below.

callout-base

It adds styles such as padding, margin to a callout and styles can be included using the following mixin:

```
@include callout-base
```

callout-style

It specifies quick styles by using a single color as a baseline for the callout and you can include styles using the following mixin:

```
@include callout-style($color);
```

It can be specified as defined in the following table:

Parameter & Description	Type	Default Value
\$color It sets the background color for callout.	color	\$callout-background

callout

It specifies styles for callout and it can include styles using the following mixin:

```
@include callout($color);
```

It can be determined as shown in the following table:

Parameter & Description	Type	Default Value
\$color It sets the background color for callout.	color	\$callout-background

Foundation – Dropdown-panes

Description

It displays the content when you click the button or hover on the element.

The following table lists some of the Dropdown pane features:

S. No.	Feature & Description
1	Basic Dropdown Pane It is used to create the dropdown pane to select the values from the list.
2	Positioning It specifies the position of the dropdown pane.
3	Sass Reference You can change the styles of the components by using SASS variables.
4	JavaScript Reference Foundation provides JavaScript components for the dropdown pane.

Foundation – Basic Dropdown Pane

Description

It is used to create the dropdown pane to select the values from the list.

Example

The following example demonstrates the use of *basic dropdown pane* in Foundation:

```
<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Dropdown Pane</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Dropdown Pane Example</h2>
<button class="button" type="button" data-toggle="dropdown_toggle">Basic
Dropdown Pane</button>
<div class="dropdown-pane" id="dropdown_toggle" data-dropdown>
    Foundation is a responsive front-end framework.
</div>
<button class="button" type="button" data-
toggle="dropdown_toggle1">Hoverable Dropdown Pane</button>
<div class="dropdown-pane" id="dropdown_toggle1" data-dropdown data-
hover="true">
    Foundation is a responsive front-end framework.
</div>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **basic_dropdown_pane.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Dropdown Pane Example

Basic Dropdown Pane

Hoverable Dropdown Pane

Foundation – Dropdown Pane Positioning

Description

It specifies the position of the dropdown pane such as top, right, left or bottom by using the `.top`, `.right`, `.left` and `.bottom` classes.

Example

The following example demonstrates the use of *positioning dropdown pane* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Dropdown Pane</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Dropdown Pane Positioning Example</h2>
  <button class="button" type="button" data-toggle="dropdown_toggle">Right</button>
  <div class="dropdown-pane right" id="dropdown_toggle" data-dropdown>
    Foundation is a responsive front-end framework.
  </div>
```

```

<button class="button" type="button" data-
toggle="dropdown_toggle1">Bottom</button>
<div class="dropdown-pane bottom" id="dropdown_toggle1" data-dropdown>
    Foundation is a responsive front-end framework.
</div><br><br><br>

<button class="button" type="button" data-
toggle="dropdown_toggle2">Top</button>
<div class="dropdown-pane top" id="dropdown_toggle2" data-dropdown>
    Foundation is a responsive front-end framework.
</div>

<button class="button" style="float:right" type="button" data-
toggle="dropdown_toggle3">Left</button>
<div class="dropdown-pane left" id="dropdown_toggle3" data-dropdown>
    Foundation is a responsive front-end framework.
</div>

<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **downdown_pane_positioning.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Dropdown Pane Positioning Example



Foundation – Dropdown Pane SASS Reference

Variables

You can change the styles of the components by using the following SASS variables as listed in the table.

S. No.	Name & Description	Type	Default Value
1	\$dropdown-padding It defines the padding for dropdown pane.	List	1rem
2	\$dropdown-border It defines Border for dropdown pane.	List	1px solid \$medium-gray
3	\$dropdown-font-size It displays the font size for dropdown pane.	List	16rem
4	\$dropdown-width It specifies width of the dropdown pane.	Number	300px
5	\$dropdown-radius It determines the border radius of the dropdown panes.	Number	\$global-radius
6	\$dropdown-sizes It defines the sizes of dropdown panes.	Map	tiny: 100px small: 200px large: 400px

Mixins

You can use the mixins to build the CSS class structure for your dropdown pane components. You can include styles for basic dropdown by using the following mixin:

```
@include dropdown-container;
```

Foundation – Dropdown Pane JavaScript Reference

Foundation provides JavaScript components for the Dropdown pane as listed below.

Initializing

You can initialize the pane in JavaScript by using `foundation.dropdown.js` and `foundation.core.js` plugins. The `foundation.core.js` plugin requires the following libraries:

- **`foundation.util.keyboard.js`**
- **`foundation.util.box.js`**

Foundation.Dropdown

It specifies instance of the dropdown as defined below:

```
var my_element = new Foundation.Dropdown(element);
```

The dropdown includes the following events as listed in the table:

S. No.	Name	Type	Description
1	element	jQuery	It creates a jQuery object into an dropdown.
2	options	Object	It overrides the default plugin settings.

Plugin Options

You can use the following plugin options to customize the dropdown instance.

S. No.	Name	Description	Example
1	hoverDelay	It specifies the time taken to open a submenu on hover event.	250
2	hover	It open the submenu on hover events.	false
3	hoverPane	Does not close dropdown when hovering over the dropdown pane.	true
4	vOffset	It provides number of pixels when you open the dropdown pane and the triggering element.	1
5	hOffset	It provides number of pixels when you open the dropdown pane and the triggering element.	1
6	positionClass	It is used to adjust the position when the class is applied to adjust.	'top'
7	trapFocus	This plugin will trap the focus to the dropdown pane if you open the dropdown with keyboard commands.	false

8	autoFocus	It is used to set the focus on the element on the pane.	true
---	-----------	---	------

Events

Dropdown pane provides the following events as listed in the table:

S. No.	Name	Description
1	closeme.zf.dropdown	It is used to close the other open dropdowns.
2	show.zf.dropdown	It shows the dropdown pane when it is visible.

Functions

The Dropdown pane provides the following functions as listed below:

- **.getPositionClass**: It specifies the current position of the dropdown pane.
- **.open**: It shows the dropdown pane and fires to close other dropdowns by using bubbling event.
- **.close**: It closes the open dropdown pane.
- **.toggle**: It toggles the dropdown pane.
- **.destroy**: It destroys the dropdown pane.

Foundation – Media Object

Description

It is used to add media objects such as images, videos, blog comments, etc. which can be placed left or right of the content block.

The following table lists some of the media object features:

S. No.	Feature & Description
1	Basic Media Object It adds media objects along with the content.
2	Section Alignment Align the sections with middle or bottom aligned.
3	Stack on Small It makes media object responsive with width of 100%.
4	Nesting Media Objects

	You can nest the media object inside another media object section.
5	Sass Reference You can change the styles of the components by using SASS variables.

Foundation – Basic Media Object

Description

It adds media objects along with the content by using `.media-object` class and use the `.media-object-section` class for sections.

Example

The following example demonstrates the use of *basic media object* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Basic Media Object</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Basic Media Object Example</h2>
  <div class="media-object">
    <div class="media-object-section middle">
      <div class="thumbnail">
        <img src= "/foundation/images/kitchen_sink/winter.jpg" alt="Media Object">
      </div>
    </div>
    <div class="media-object-section">
      <h4>Foundation</h4>
      <p>Foundation is a family of responsive front-end frameworks that make it easy to design beautiful responsive websites, apps and emails that look amazing on any device.</p>
    </div>
  </div>

```

```

</div>
</div>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **basic_media_object.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Basic Media Object Example

Foundation



Foundation is a family of responsive front-end frameworks that make it easy to design beautiful responsive websites, apps and emails that look amazing on any device.

Foundation – Media Object Section Alignment

Description

You can align the sections in the middle or bottom by using the *.middle* and *.bottom* classes respectively.

Example

The following example demonstrates the use of *section alignment* in Foundation:

```

<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Media Object Section Alignment</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
  s">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
  in.js"></script>
```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
js"></script>
</head>
<body>
<h2>Media Object Section Alignment Example</h2>
<div class="media-object">
  <div class="media-object-section middle">
    <div class="thumbnail">
      <img src= "/foundation/images/kitchen_sink/winter.jpg" alt="Section
Alignment">
    </div>
  </div>
  <div class="media-object-section">
    <h4>Foundation</h4>
    <p>Foundation is a family of responsive front-end frameworks that make
it easy to design beautiful responsive websites, apps and emails that look amazing
on any device.</p>
    <p>Foundation is semantic, readable, flexible, and completely
customizable. Foundation for Sites is jam-packed with features to help people
rapidly build content-focused websites.</p>
  </div>
  <div class="media-object-section bottom">
    <div class="thumbnail">
      <img src= "/foundation/images/kitchen_sink/winter.jpg" alt="Section
Alignment">
    </div>
  </div>
</div>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **media_section_alignment.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Media Object Section Alignment Example

Foundation



Foundation is a family of responsive front-end frameworks that make it easy to design beautiful responsive websites, apps and emails that look amazing on any device.

Foundation is semantic, readable, flexible, and completely customizable. Foundation for Sites is jam-packed with features to help people rapidly build content-focused websites.



Foundation – Media Object Stack on Small

Description

It makes the media object responsive with width of 100% by using the `.stack-for-small` class.

Example

The following example demonstrates the use of `stack on small` in Foundation:

```
<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Media Object Stack on Small</title>
    <link rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
        src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
        src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
    <h2>Media Object Stack on Small Example</h2>
    <div class="media-object stack-for-small">
        <div class="media-object-section">
```

```

<div class="thumbnail">
    <img src= "/foundation/images/kitchen_sink/winter.jpg">
</div>
</div>
<div class="media-object-section">
    <h4>Foundation</h4>
    <p>Foundation is a family of responsive front-end frameworks that make it easy to design beautiful responsive websites, apps and emails that look amazing on any device.</p>
</div>
</div>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **media_stack_on_small.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Media Object Stack on Small Example



Foundation

Foundation is a family of responsive front-end frameworks that make it easy to design beautiful responsive websites, apps and emails that look amazing on any device.

Foundation – Nesting Media Objects

Description

You can nest the media object inside another media object section.

Example

The following example demonstrates the use of *nesting media objects* in Foundation:

```

<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Nesting Media Objects</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Nesting Media Objects Example</h2>
<div class="media-object">
  <div class="media-object-section">
    <div class="thumbnail">
      <img src= "/foundation/images/kitchen_sink/winter.jpg" alt="Nesting Media Objects">
    </div>
  </div>
  <div class="media-object-section">
    <h4>Foundation</h4>
    <p>Foundation is a family of responsive front-end frameworks that make it easy to design beautiful responsive websites, apps and emails that look amazing on any device.</p>
  <div class="media-object">
    <div class="media-object-section">
      <div class="thumbnail">
        <img src= "/foundation/images/kitchen_sink/winter.jpg" alt="Nesting Media Objects">
      </div>
    </div>
    <div class="media-object-section">
      <h4>Foundation</h4>
      <p>Foundation is a family of responsive front-end frameworks that make it easy to design beautiful responsive websites, apps and emails that look amazing on any device.</p>
    </div>
  </div>
</div>

```

```
<div class="media-object">
  <div class="media-object-section">
    <div class="thumbnail">
      <img src= "/foundation/images/kitchen_sink/winter.jpg" alt="Nesting Media Objects">
    </div>
  </div>
  <div class="media-object-section">
    <h4>Foundation</h4>
    <p>Foundation is a family of responsive front-end frameworks that make it easy to design beautiful responsive websites, apps and emails that look amazing on any device.</p>
  </div>
</div>
</div>
</div>
</div>
</div>
</div>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **nesting_media_objects.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Nesting Media Objects Example



Foundation

Foundation is a family of responsive front-end frameworks that make it easy to design beautiful responsive websites, apps and emails that look amazing on any device.



Foundation

Foundation is a family of responsive front-end frameworks that make it easy to design beautiful responsive websites, apps and emails that look amazing on any device.



Foundation

Foundation is a family of responsive front-end frameworks that make it easy to design beautiful responsive websites, apps and emails that look amazing on any device.

Foundation – Media Objects Sass Reference

Variables

You can change the styles of the components by using the following SASS variables as listed in the table.

S. No.	Name & Description	Type	Default Value
1	\$mediaobject-margin-bottom It sets the bottom margin of a media object.	Number	\$global-margin
2	\$mediaobject-section-padding It defines the left and right padding within a media object.	Number	\$global-padding
3	\$mediaobject-image-width-stacked When the object is stacked vertically, images will get a width of 100% and keeps the natural width of image by setting it to 'auto'.	Number	100%

Mixins

You can use the mixins to build the CSS class structure for your media object components. You can use the following mixins to build your own class structure:

media-object-container

You can add styles for media object by using the following mixin:

```
@include media-object-container;
```

media-object-section

You can add styles for media object section by using the following mixin:

```
@include media-object-section($padding);
```

It can be defined using the parameter as shown in the table given below:

Parameter & Description	Type	Default Value
\$padding It defines padding between sections.	Number	\$mediaobject-section-padding

media-object-stack

You can add styles for stack sections of media object by using the following mixin:

```
@include media-object-stack;
```

Foundation – Off-canvas

Description

It sets off the navigation menu from the visible area and displays the main content.

The following table lists some of the Off-canvas features:

S. No.	Feature & Description
1	Off Canvas Setup You can create off canvas menu using <i>.off-canvas</i> class and <i>data-off canvas</i> attribute.
2	Multiple Menus Foundation off-canvas creates multiple menus on the left and right side.
3	Title Bar You can toggle the off-canvas by using the title bar on both left and right sides.
4	Reveal on Larger Screens

	Sets the left and right hand off canvas on larger and medium screens using <code>.reveal-for-medium</code> or <code>.reveal-for-large</code> classes.
5	Sass Reference You can change the styles of the components by using SASS variables.
6	JavaScript Reference Foundation provides JavaScript components for the Off-canvas.

Foundation – Off-canvas Setup

Description

You can create off canvas menu using the `.off-canvas` class and `data-off-canvas` attribute.

Example

The following example demonstrates the use of `.off canvas` class in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Off Canvas Setup</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <div class="off-canvas-wrapper">
    <div class="off-canvas-wrapper-inner" data-off-canvas-wrapper>
      <div class="off-canvas position-left" id="offCanvas" data-off-canvas data-position="left">
        Welcome to Foundation
      </div>
      <div class="off-canvas-content" data-off-canvas-content>
        <button type="button" class="button" data-toggle="offCanvas">Click here to open content</button>
      </div>
    </div>
  </div>

```

```

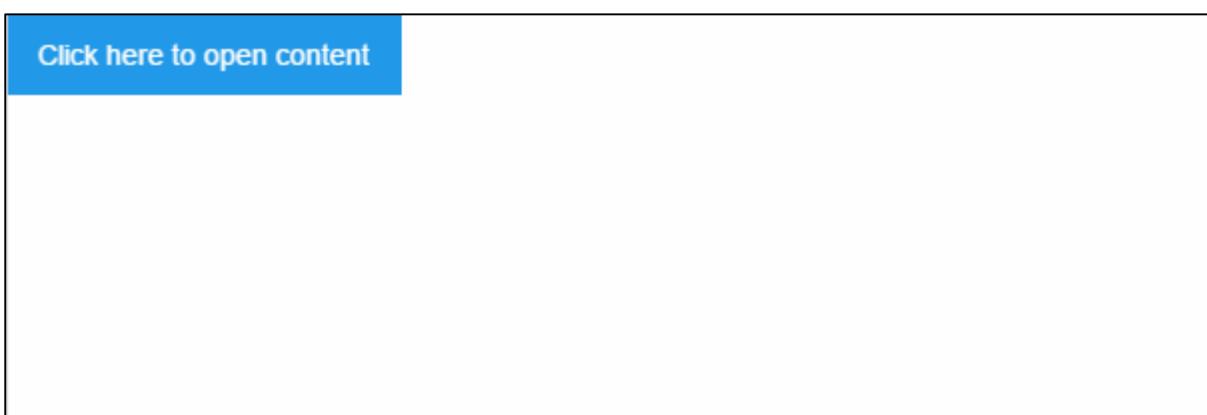
    </div>
</div>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **off_canvas_setup.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Click here to open content

Foundation – Off-canvas Multiple Menus

Description

Foundation off-canvas creates multiple menus on the left and right side by using *.position-left* and *.position-right* classes respectively.

Example

The following example demonstrates the use of *off canvas multiple menus* in Foundation:

```

<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Off Canvas Multiple Menus</title>

```

```

<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Off-canvas Multiple Menus Example</h2>
<div class="off-canvas-wrapper">
    <div class="off-canvas-wrapper-inner" data-off-canvas-wrapper>
        <div class="off-canvas position-left" id="off_canvas_left" data-off-canvas></div>
        <div class="off-canvas position-right" id="off_canvas_right" data-off-canvas data-position="right"></div>
        <div class="off-canvas-content" data-off-canvas-content>
            <p style="text-align:center;">Welcome to Foundation</p>
        </div>
        <button type="button" style="display:block; margin: 0 auto;" class="button" data-toggle="off_canvas_left">Click here to move left</button><br>
        <button type="button" style="display:block; margin: 0 auto;" class="button" data-toggle="off_canvas_right">Click here to move right</button>
    </div>
</div>
<script>
    $(document).ready(function() {
        $(document).foundation();
    })
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **off_canvas_multiple_menus.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Off-canvas Multiple Menus Example

Welcome to Foundation

[Click here to move left](#)

[Click here to move right](#)

Foundation – Off-canvas Title Bar

Description

You can toggle the off-canvas by using the title bar on both left and right sides by using the `.title-bar` class.

Example

The following example demonstrates the use of `title bar` in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Off Canvas Title Bar</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Off-canvas Title Bar Example</h2>
  <div class="off-canvas-wrapper">
    <div class="off-canvas-wrapper-inner" data-off-canvas-wrapper>
      <div class="off-canvas position-right" id="off_canvas_right" data-off-canvas data-position="right">
        Right side content
      </div>
    </div>
  </div>
```

```

<div class="off-canvas position-left" id="off_canvas_left" data-off-canvas data-position="left">
    Left side content
</div>
<div class="main-content" data-off-canvas-content>
    <div class="title-bar">
        <div class="title-bar-left">
            <button class="menu-icon" type="button" data-open="off_canvas_right"></button>
            <span class="title-bar-title">Welcome to Foundation</span>
        </div>
        <div class="title-bar-right">
            <button class="menu-icon" type="button" data-open="off_canvas_left"></button>
        </div>
    </div>
</div>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

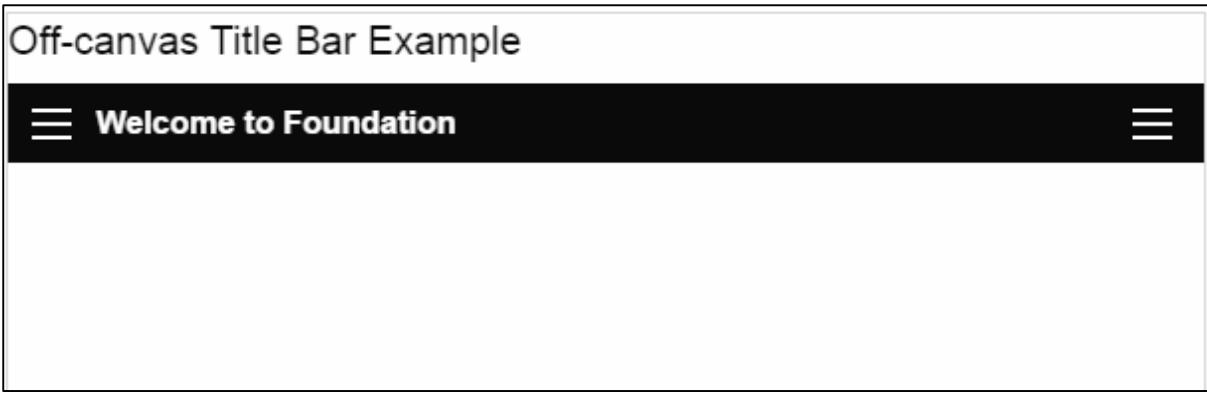
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **off_canvas_title_bar.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Off-canvas Title Bar Example



☰ Welcome to Foundation ☰

Foundation – Off-canvas Larger Screen Reveal

Description

Set the left and right hand off-canvas on larger and medium screens using the `.reveal-for-medium` or `.reveal-for-large` classes.

Example

The following example demonstrates the use of *reveal on larger screens* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Off Canvas Reveal on Larger Screens</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Off-canvas Reveal on Larger Screen Example</h2>
  <div class="off-canvas-wrapper">
    <div class="off-canvas-wrapper-inner" data-off-canvas-wrapper>
      <div class="off-canvas position-left reveal-for-large"
        id="off_canvas_left" data-off-canvas></div>
      <div class="off-canvas-content" data-off-canvas-content>
```

```

        Welcome to Foundation
    </div>
    <button type="button" class="button" data-
    toggle="off_canvas_left">Click Here</button>
    </div>
    </div>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **off_canvas_larger_screen_reveal.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Off-canvas Reveal on Larger Screen Example

Welcome to Foundation

Click Here

Foundation – Off-canvas SASS Reference

Variables

You can change the styles of the components by using the following SASS variables as listed in the table:

S. No.	Name & Description	Type	Default Value
1	\$offcanvas-size It specifies the off-canvas menu width.	Number	250px

2	\$offcanvas-background It displays the background color of an off-canvas menu.	Color	\$light-gray
3	\$offcanvas-zindex It specifies Z-index of an off-canvas menu.	Number	-1
4	\$offcanvas-transition-length It displays the length of the animation on an off-canvas menu.	Number	0.5s
5	\$offcanvas-transition-timing It specifies the off-canvas menu's timing function of the animation.	Keyword	ease
6	\$offcanvas-fixed-reveal If true, a revealed off-canvas can fix the position of revealed off canvas by setting it to true.		true
7	\$offcanvas-exit-background It displays the background color for the off-canvas menu's overlay.	Color	rgba(\$white, 0.25)
8	\$maincontent-class It uses CSS class for the main content area.		'off-canvas-content'
9	\$maincontent-shadow It displays the box shadow for the main content area.	Shadow	0 0 10px rgba(\$black, 0.5)
10	\$titlebar-background It displays the background color of a title bar.	Color	\$black
11	\$titlebar-color It defines the text color for the text inside a title bar.	Color	\$white
12	\$titlebar-padding It defines padding inside a title bar.	Length	0.5rem
13	\$titlebar-text-font-weight It specifies the font weight of the text.	Weight	bold
14	\$titlebar-icon-color It defines the title bar icon color of the menu.	Color	\$white
15	\$titlebar-icon-color-hover It defines the title bar icon color of the menu on hover.	Color	\$medium-gray
16	\$titlebar-icon-spacing It specifies the spacing between the menu icon and the text inside a title bar.	Length	0.25rem

Mixins

You can use the mixins to build the CSS class structure for your components as described below.

off-canvas-basics

It adds styles for off-canvas and styles can be included using the following mixin:

```
@include off-canvas-basics;
```

off-canvas-base

It specifies basic styles for the off-canvas menu and you can include styles using the following mixin:

```
@include off-canvas-base;
```

off-canvas-reveal

It specifies styles that reveal the off-canvas menu.

```
@include off-canvas-reveal($position);
```

It can be specified as defined in the following table:

Parameter & Description	Type	Default Value
\$position It sets the position of the off-canvas menu.	Keyword	left

Foundation – Off-canvas JavaScript Reference

Foundation provides JavaScript components for the Off-canvas as listed below.

Initializing

You can initialize the pane in the JavaScript by using the *foundation.offcanvas.js* and *foundation.core.js* plugins. The *foundation.core.js* plugin requires the following libraries:

- **foundation.util.triggers.js**
- **foundation.util.motion.js**

Foundation.OffCanvas

It specifies instance of the off-canvas wrapper as defined below:

```
var my_element = new Foundation.OffCanvas(element);
```

The Off-canvas includes the following events as listed in the table:

S. No.	Name & Description	Type
1	element It initializes the jQuery object.	Object
2	options It overrides the default plugin settings.	Object

Plugin Options

You can use the following plugin options to customize the off-canvas instance.

S. No.	Name & Description	Example
1	closeOnClick It closes the menu when the user clicks outside of the menu.	true
2	transitionTime It specifies the time taken to open and close transition in ms.	500
3	position It defines the direction from where the off-canvas should open.	left
4	forceTop It forces the page to scroll to the top as soon as it is opened.	
5	isRevealed It will open the off-canvas, until it reaches the remaining breakpoints.	false
6	revealOn It defines the breakpoint at which to reveal the Off-canvas.	reveal-for-large
7	autoFocus It focuses on off-canvas when it will get open.	true
8	revealClass This class is used to force an off-canvas to remain open.	reveal-for-large

Events

Off-canvas provides the following events as listed in the table:

S. No.	Name & Description
1	opened.zf.off-canvas It triggers the event when it opens the off-canvas menu.
2	closed.zf.off-canvas It triggers the event when it opens the off-canvas menu.

Functions

Off-canvas provides the following functions as defined below:

.reveal

It will open the off-canvas until it reaches the remaining breakpoints. It has the following function as specified in the table:

Name & Description	Type
isRevealed It reveals the element if this function set to true.	Boolean

.open

It will open the off-canvas menu. It has following functions as specified in the table:

Name & Description	Type
event It passes the event object from listener.	Object
trigger It triggers an element to open the off-canvas.	jQuery

.close

It closes the off-canvas menu.

.toggle

It toggles the off-canvas menu. It has the following functions as specified in the table:

Name & Description	Type
event It passes the event object from listener.	Object
trigger It triggers an element to open the off-canvas.	jQuery

.destroy

It destroys the off-canvas plugin.

Foundation – Reveal

Description

Foundation allows creating modal dialogs or pop-up window using *reveal* class.

The following table lists some of the Reveal features:

S. No.	Feature & Description
1	Reveal Basics Create modal dialogs or pop-up window using the <i>reveal</i> class.
2	Modals in Modals Foundation allow us to create modal within modal.
3	Reveal Video It creates a modal to display the video.
4	Reveal Sizing It creates different types of modals such as tiny, small and large modals.
5	Reveal Full Screen You can create full screen modal according to width and height of the window.
6	Reveal No Overlay Foundation removes the overlay by setting <i>data-overlay</i> to false.
7	Reveal Animations It creates the animations on the modal.
8	Sass Reference You can change the styles of the components by using SASS variables.
9	JavaScript Reference Foundation provides JavaScript components for the modal.

Foundation – Reveal Basics

Description

You can create modal dialogs or pop-up window using the *reveal* class.

Example

The following example demonstrates the use of *basic reveal* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Reveal Basics</title>
```

```

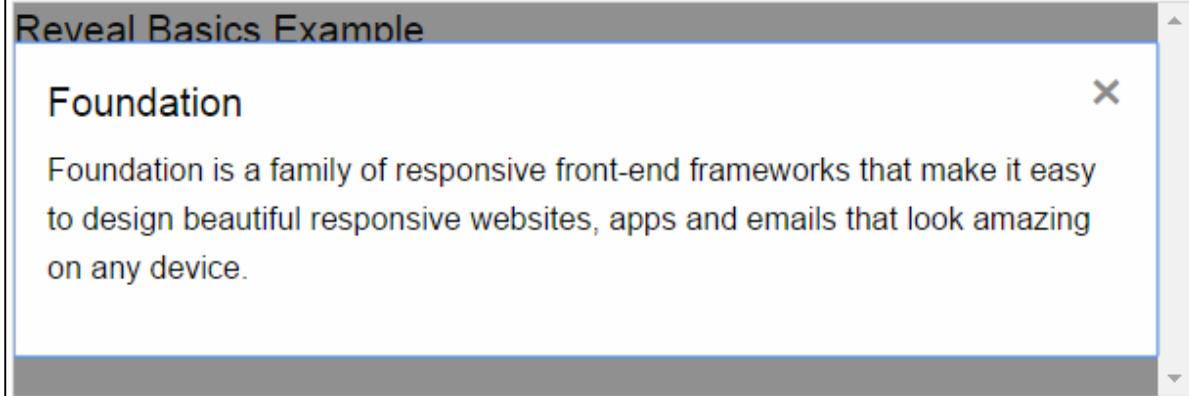
<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Reveal Basics Example</h2>
<p><a data-open="reveal_modal">Click here to open modal</a></p>
<div class="reveal" id="reveal_modal" data-reveal>
    <h2>Foundation</h2>
    <p>Foundation is a family of responsive front-end frameworks that make it
       easy to design beautiful responsive websites, apps and emails that look amazing
       on any device.</p>
    <button class="close-button" data-close aria-label="Close reveal"
           type="button">
        <span aria-hidden="true">x</span>
    </button>
</div>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **reveal_basics.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Reveal Modals in Modals

Description

Foundation allows us to create a modal within a modal.

Example

The following example demonstrates the use of *modals in modals* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Reveal Modal in Modal</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Reveal Modal in Modal Example</h2>
  <p><a data-open="reveal_modal">Click here to open modal</a></p>
  <div class="reveal" id="reveal_modal" data-reveal>
    <h2>Foundation</h2>
  </div>
</body>
```

```

<p>Foundation is a family of responsive front-end frameworks that make it
easy to design beautiful responsive websites, apps and emails that look amazing
on any device.</p>

<a class="button" data-toggle="reveal_modal1">Click here to open another
modal!</a>

<button class="close-button" data-close aria-label="Close reveal"
type="button">
    <span aria-hidden="true">x</span>
</button>
</div>

<div class="reveal" id="reveal_modal1" data-reveal>
    <h2>This is another modal!!!</h2>
    <p>Foundation is semantic, readable, flexible, and completely
customizable.</p>
    <button class="close-button" data-close aria-label="Close reveal"
type="button">
        <span aria-hidden="true">x</span>
    </button>
</div>

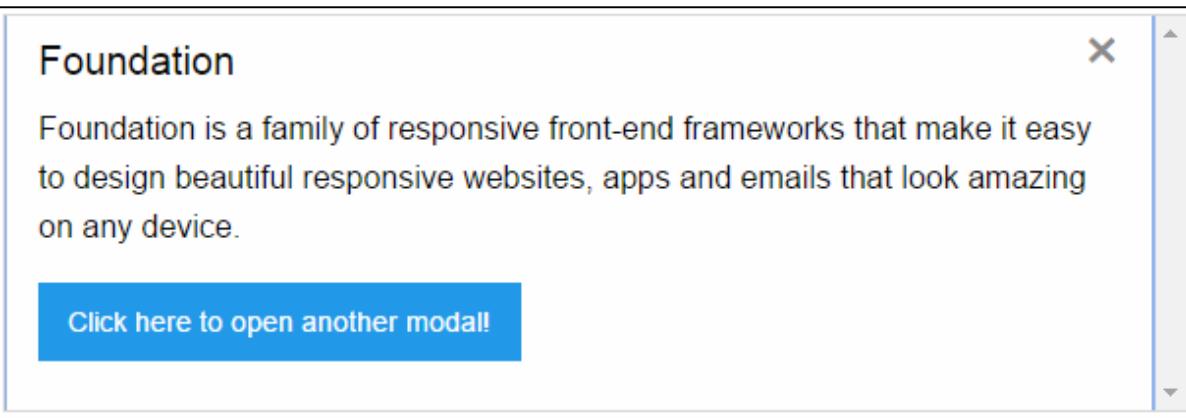
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **reveal_modals_in_modals.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Reveal Video

Description

It creates a modal to display the video.

Example

The following example demonstrates the use of *video in modal* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Reveal Video in Modal</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Reveal Video in Modal Example</h2>
  <p><a data-toggle="reveal_modal">Click here to open modal</a></p>
  <div class="reveal" id="reveal_modal" data-reveal>
    <h2>Video in Modal</h2>
    <div class="flex-video widescreen vimeo">
      <iframe src="/foundation/images/kitchen_sink/video.mp4"
      allowfullscreen></iframe>
    </div>
  </div>
</body>
```

```

    </div>
    <button class="close-button" data-close aria-label="Close reveal"
type="button">
        <span aria-hidden="true">x</span>
    </button>
</div>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **reveal_video.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Reveal Sizing

Description

It creates different types of modals such as tiny, small and large modals by adding *tiny*, *small* and *large* classes as described below:

Tiny Size Modal

It creates a tiny size of modal.

Example

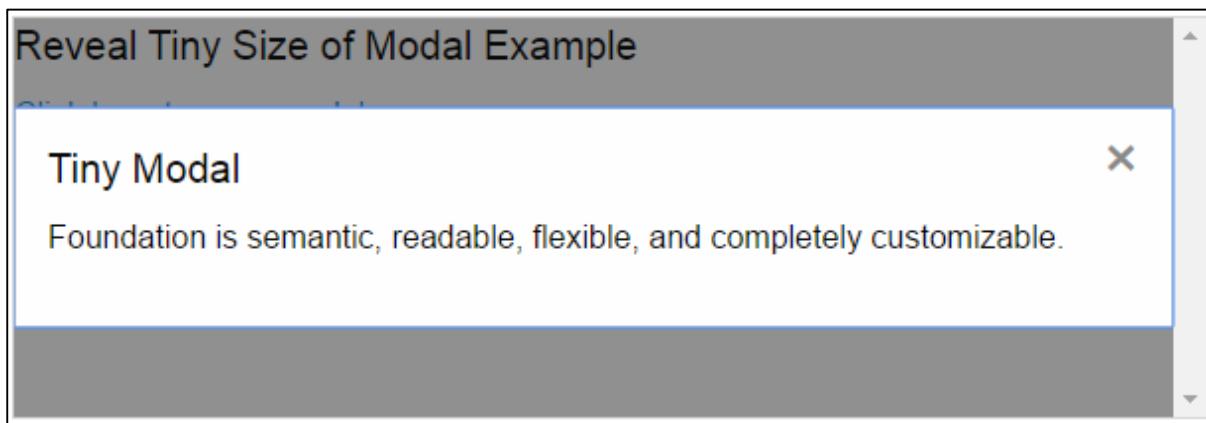
The following example demonstrates the use of *tiny size modal* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Reveal Tiny Size Modal</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Reveal Tiny Size of Modal Example</h2>
  <p><a data-toggle="reveal_modal">Click here to open modal</a></p>
  <div class="tiny reveal" id="reveal_modal" data-reveal>
    <h2>Tiny Modal</h2>
    <p>Foundation is semantic, readable, flexible, and completely customizable.</p>
    <button class="close-button" data-close aria-label="Close reveal"
      type="button">
      <span aria-hidden="true">x</span>
    </button>
  </div>
<script>
  $(document).ready(function() {
    $(document).foundation();
  })
</script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **reveal_tiny_modal.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Small Size Modal

It creates a small size of modal.

Example

The following example demonstrates the use of *small size modal* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Reveal Small Size Modal</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Reveal Small Size of Modal Example</h2>
  <p><a data-toggle="reveal_modal">Click here to open modal</a></p>
  <div class="small reveal" id="reveal_modal" data-reveal>
    <h2>Small Modal</h2>
  </div>
</body>
</html>
```

```

<p>Foundation is semantic, readable, flexible, and completely
customizable.</p>

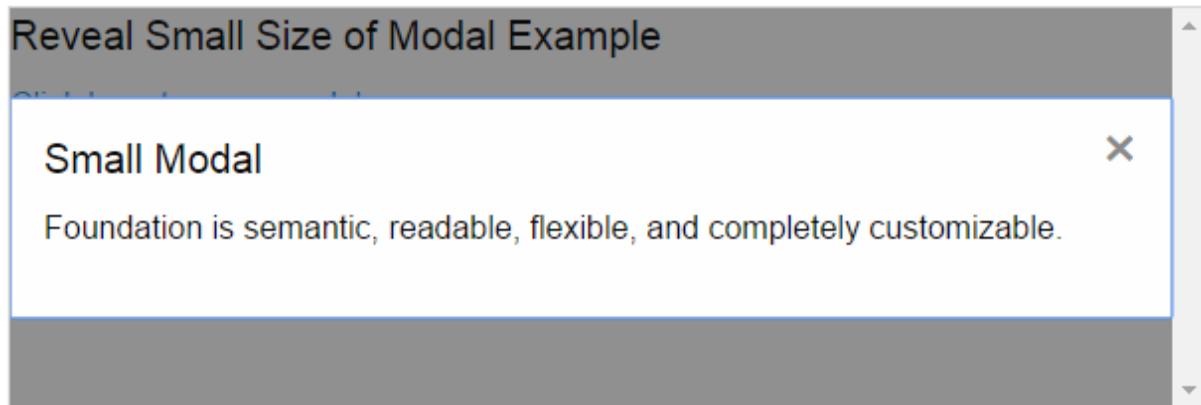
<button class="close-button" data-close aria-label="Close reveal"
type="button">
    <span aria-hidden="true">x</span>
</button>
</div>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **reveal_small_modal.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Large Size Modal

It creates a large size of modal.

Example

The following example demonstrates the use of *large size modal* in Foundation:

```

<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">

```

```

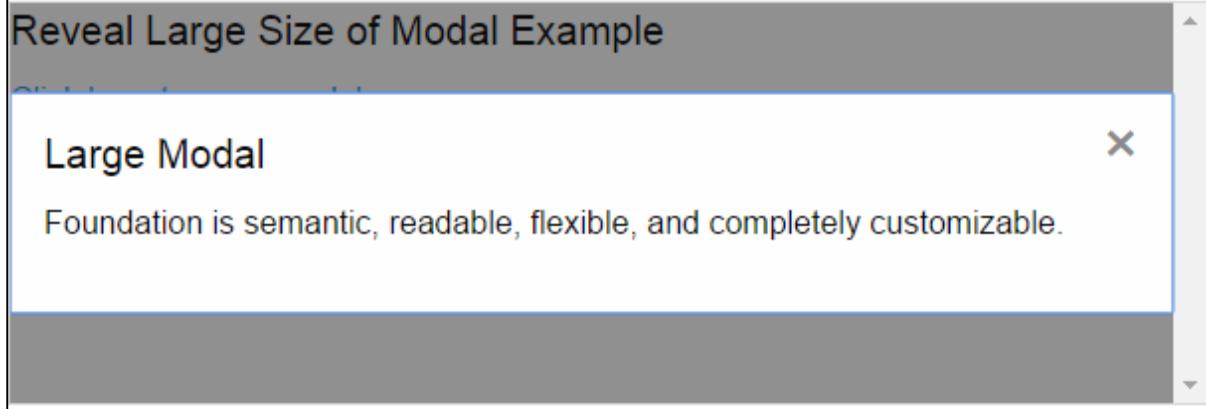
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Reveal Large Size Modal</title>
<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Reveal Large Size of Modal Example</h2>
<p><a data-toggle="reveal_modal">Click here to open modal</a></p>
<div class="large reveal" id="reveal_modal" data-reveal>
    <h2>Large Modal</h2>
    <p>Foundation is semantic, readable, flexible, and completely
    customizable.</p>
    <button class="close-button" data-close aria-label="Close reveal"
           type="button">
        <span aria-hidden="true">x</span>
    </button>
</div>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **reveal_large_modal.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Reveal Full-screen

Description

You can create a full screen modal according to the width and height of the window.

Example

The following example demonstrates the use of *full screen modal* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Reveal Full Screen Modal</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Reveal Full Screen Modal Example</h2>
  <p><a data-toggle="reveal_modal">Click here to open modal</a></p>
  <div class="full reveal" id="reveal_modal" data-reveal>
    <h2>This is Full Screen Modal</h2>
    <p>Foundation is semantic, readable, flexible, and completely
    customizable.</p>

```

```

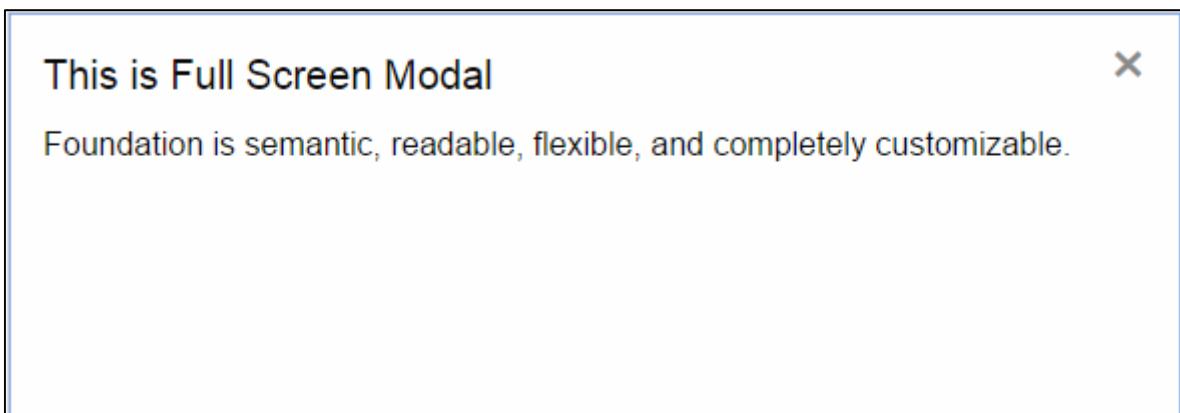
        <button class="close-button" data-close aria-label="Close reveal"
type="button">
    <span aria-hidden="true">x</span>
</button>
</div>
</script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **reveal_full_screen_modal.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Reveal No Overlay

Description

Foundation removes the overlay by setting *data-overlay* to false.

Example

The following example demonstrates the use of *no overlay* in Foundation:

```

<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">

```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Reveal No Overlay</title>
<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Reveal No Overlay Example</h2>
<p><a data-toggle="reveal_modal">Click here to open modal</a></p>
<div class="reveal" id="reveal_modal" data-reveal data-overlay="false">
    <h2>This is No Overlay Modal</h2>
    <p>Foundation is semantic, readable, flexible, and completely
    customizable.</p>
    <button class="close-button" data-close aria-label="Close reveal"
           type="button">
        <span aria-hidden="true">x</span>
    </button>
</div>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **reveal_no_overlay.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Reveal No Overlay Example

This is No Overlay Modal



Foundation is semantic, readable, flexible, and completely customizable.

Foundation – Reveal Animations

Description

It creates the animations on the modal.

Example

The following example demonstrates the use of *animations* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Reveal Animations</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/motion-ui/1.1.1/motion-ui.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Reveal Animations Example</h2>
  <p><a data-toggle="reveal_modal">Click here to open modal</a></p>
  <div class="reveal" id="reveal_modal" data-reveal data-animation-in="spin-in"
    data-animation-out="spin-out">
    <h2>This is Animations Modal</h2>
  </div>
</body>

```

```

<p>Foundation is semantic, readable, flexible, and completely
customizable.</p>

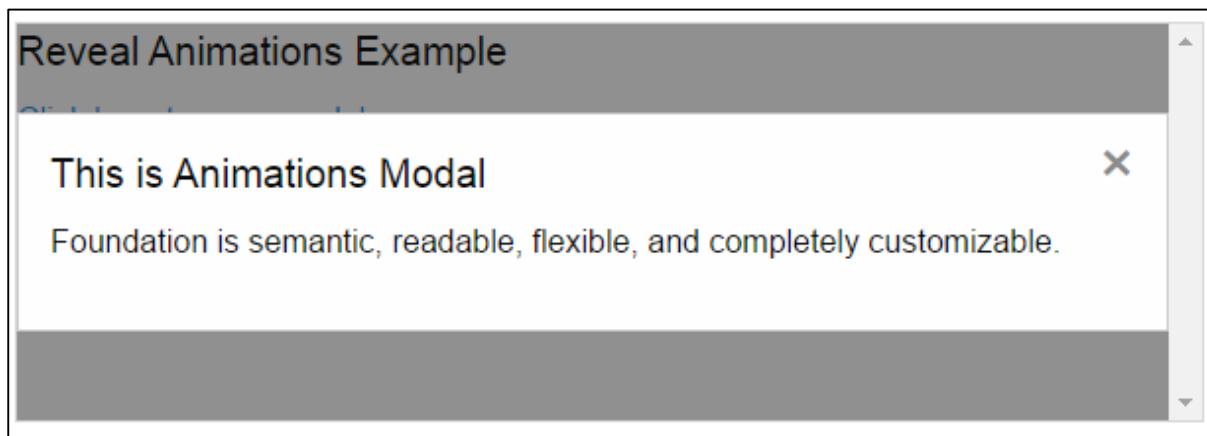
<button class="close-button" data-close aria-label="Close reveal"
type="button">
    <span aria-hidden="true">x</span>
</button>
</div>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **reveal_animations.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Reveal SASS Reference

Variables

You can change the styles of the components by using SASS variables as listed in the table given below.

S. No.	Name & Description	Type	Default Value
1	\$reveal-background It displays the background color of a modal.	Color	\$white

2	\$reveal-width It defines the width of the modal.	Number	600px
3	\$reveal-max-width It specifies the maximum width of a modal.	Number	\$global-width
4	\$reveal-offset When you add the modal, it will offset from the top of the window.	Number	rem-calc(100)
5	\$reveal-padding It defines the padding inside a modal.	Number	\$global-padding
6	\$reveal-border It sets the border for a modal.	Number	1px solid \$medium-gray
7	\$reveal-radius It specifies radius for the modal.	Number	\$global-radius
8	\$reveal-zindex The overlay uses z-index value for modals.	Number	1005
9	\$reveal-overlay-background It displays the background color of modal overlays.	Color	rgba(\$black, 0.45)

Mixins

You can use the mixins to build the CSS class structure for your reveal modal components.

reveal-overlay

You can include styles for modal overlay using the following mixin:

```
@include reveal-overlay($background);
```

It contains one parameter as specified in the following table:

Parameter & Description	Type	Default Value
\$background It displays the background color of overlay.	Color	\$reveal-overlay-background

reveal-modal-base

You can add base styles for a modal using the following mixin:

```
@include reveal-modal-base;
```

reveal-modal-width

It is used to create width of a modal using the following mixin:

```
@include reveal-modal-width($width, $max-width);
```

It contains the following parameters as specified in the table:

Parameter & Description	Type	Default Value
\$width It specifies the width of the modal.	Number	None
\$max-width It specifies the maximum width of the modal.	Number	\$reveal-max-width

reveal-modal-fullscreen

You can create full screen modal according to the width and height of the window by using the following mixin:

```
@include reveal-modal-fullscreen;
```

Foundation – Reveal JavaScript Reference

Foundation provides JavaScript components for the reveal modal as listed below:

Initializing

You can initialize the pane in JavaScript by using the *foundation.reveal.js* and *foundation.core.js* plugins. The *foundation.core.js* plugin requires the following libraries:

- **foundation.util.keyboard.js**
- **foundation.util.box.js**
- **foundation.util.triggers.js**
- **foundation.util.mediaQuery.js**
- **foundation.util.motion.js**

Foundation.Reveal

It specifies instance of the reveal as defined below:

```
var my_element = new Foundation.Reveal(element);
```

The reveal includes the following values as listed in the table:

Name & Description	Type
element It uses the jQuery object for the modal.	Number
options It specifies the optional parameters for the modal.	Object

Plugin Options

You can use the following plugin options to customize the reveal instance.

S. No.	Name & Description	Example
1	animationIn It is used for animated elements.	'slide-in-left'
2	animationOut It is used for animated elements.	'slide-out-right'
3	showDelay It displays the time taken to open the modal in ms.	10
4	hideDelay It displays the time taken to close the modal in ms.	10
5	closeOnClick It closes the modal when you click on the body or overlay.	true
6	closeOnEsc It closes the modal when you click the 'ESCAPE' key.	true
7	multipleOpened It displays the multiple modals at once.	false
8	vOffset It specifies the distance in pixels when modal should push down from the top of the screen.	100
9	hOffset It specifies the distance in pixels when modal should push down from the top of the screen.	0
10	fullScreen You can create full screen modal according to width and height of the window.	false
11	btmOffsetPct When the modal should push up from the bottom of the view, it specifies the percentage of screen height.	10
12	overlay When modal opens, it will generate an overlay div.	true
13	resetOnClose It specifies that modal should be reset when you close it.	false

Events

Reveal provides the following events as listed in the table:

S. No.	Event & Description
1	closeAll.zf.reveal It closes all the current modals before opening the new modal.
2	open.zf.reveal It triggers the event when you open the modal successfully.
3	closed.zf.reveal It triggers the event when it closes the modal.

Functions

Reveal provides the following functions:

S. No.	Function & Description
1	.open It will open the modal, which is controlled by <i>this.\$anchor</i> and closes the other modals.
2	.close It will close the modal.
3	.toggle It toggles the state of a modal.
4	.destroy It destroys the modal instance.

Foundation – Basic Table

Description

It displays the basic table style with padding and horizontal dividers.

Example

The following example demonstrates the use of *basic table* in Foundation:

```
<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Basic Table</title>
```

```

<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Basic Table Example</h2>
<table>
  <thead>
    <tr><th>Team</th><th>Player Name</th><th>Batsman/Bowler</th></tr>
  </thead>
  <tbody>
    <tr>
      <td>India</td>
      <td>Sachin Tendulkar</td>
      <td>Batsman</td>
    </tr>
    <tr>
      <td>South Africa</td>
      <td>Shaun Pollock</td>
      <td>Bowler</td>
    </tr>
    <tr>
      <td>West Indies</td>
      <td>Brian Lara</td>
      <td>Batsman</td>
    </tr>
    <tr>
      <td>Australia</td>
      <td>Shane Watson</td>
      <td>Bowler</td>
    </tr>
    <tr>
      <td>Ross Taylor</td>
      <td>New Zealand</td>
      <td>Batsman</td>
    </tr>
  </tbody>
</table>

```

```

</tr>
</tbody>
</table>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **basic_table.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Basic Table Example

Team	Player Name	Batsman/Bowler
India	Sachin Tendulkar	Batsman
South Africa	Shaun Pollock	Bowler
West Indies	Brian Lara	Batsman
Australia	Shane Watson	Bowler
Ross Taylor	New Zealand	Batsman

Foundation – Hover Table

Description

Foundation allows you to hover on table rows by adding the `.hover` class.

Example

The following example demonstrates the use of *hover table* in Foundation:

```

<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Hover Table</title>

```

```
<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Hover Table Example</h2>
<table class="hover">
  <thead>
    <tr><th>Team</th><th>Player Name</th><th>Batsman/Bowler</th></tr>
  </thead>
  <tbody>
    <tr>
      <td>India</td>
      <td>Sachin Tendulkar</td>
      <td>Batsman</td>
    </tr>
    <tr>
      <td>South Africa</td>
      <td>Shaun Pollock</td>
      <td>Bowler</td>
    </tr>
    <tr>
      <td>West Indies</td>
      <td>Brian Lara</td>
      <td>Batsman</td>
    </tr>
    <tr>
      <td>Australia</td>
      <td>Shane Watson</td>
      <td>Bowler</td>
    </tr>
    <tr>
      <td>Ross Taylor</td>
      <td>New Zealand</td>
      <td>Batsman</td>
    </tr>
  </tbody>
</table>
```

```

</tr>
</tbody>
</table>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **hover_table.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Hover Table Example

Team	Player Name	Batsman/Bowler
India	Sachin Tendulkar	Batsman
South Africa	Shaun Pollock	Bowler
West Indies	Brian Lara	Batsman
Australia	Shane Watson	Bowler
Ross Taylor	New Zealand	Batsman

Foundation – Stacked Table

Description

You can stack the table by adding the *.stack* class.

Example

The following example demonstrates the use of *stacked table* in Foundation:

```

<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Stacked Table</title>

```

```

        <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
        <script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
in.js"></script>
        <script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
js"></script>
</head>
<body>
<h2>Stacked Table Example</h2>
<table class="stack">
    <thead>
        <tr><th>Team</th><th>Player Name</th><th>Batsman/Bowler</th></tr>
    </thead>
    <tbody>
        <tr>
            <td>India</td>
            <td>Sachin Tendulkar</td>
            <td>Batsman</td>
        </tr>
        <tr>
            <td>South Africa</td>
            <td>Shaun Pollock</td>
            <td>Bowler</td>
        </tr>
        <tr>
            <td>West Indies</td>
            <td>Brian Lara</td>
            <td>Batsman</td>
        </tr>
        <tr>
            <td>Australia</td>
            <td>Shane Watson</td>
            <td>Bowler</td>
        </tr>
        <tr>
            <td>Ross Taylor</td>
            <td>New Zealand</td>
            <td>Batsman</td>
        </tr>
    </tbody>
</table>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **stacked_table.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Stacked Table Example

India	
Sachin Tendulkar	
Batsman	
South Africa	
Shaun Pollock	
Bowler	
West Indies	
Brian Lara	
Batsman	
Australia	
Shane Watson	
Bowler	
Ross Taylor	
New Zealand	
Batsman	

Foundation – Scroll Table

Description

342

You can scroll the table horizontally by adding the `.scroll` class.

Example

The following example demonstrates the use of `scroll table` in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Scroll Table</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Scroll Table Example</h2>
  <table class="scroll">
    <thead>
      <tr>
        <th>Team</th>
        <th>Player One</th>
        <th>Player Two</th>
        <th>Player Three</th>
        <th>Player Four</th>
        <th>Player Five</th>
        <th>Player Six</th>
        <th>Player Seven</th>
        <th>Player Eight</th>
        <th>Player Nine</th>
        <th>Player Ten</th>
        <th>Player Eleven</th>
      </tr>
    </thead>
    <tbody>
      <tr>
```

```
<td>India</td>
<td>Mahendra Singh Dhoni</td>
<td>Virat Kohli</td>
<td>Ajinkya Rahane</td>
<td>Shikhar Dhawan</td>
<td>Rohit Sharma</td>
<td>Suresh Raina</td>
<td>Ravindra Jadeja</td>
<td>Ravichandran Ashwin</td>
<td>Bhuvneshwar Kumar</td>
<td>Mohammed Shami</td>
<td>Umesh Yadav</td>
</tr>
<tr>
<td>Srilanka</td>
<td>Angelo Mathews</td>
<td>Thirimanne</td>
<td>Tillakaratne Dilshan</td>
<td>Mahela Jayawardene</td>
<td>Tharindu Kaushal</td>
<td>Kulasekara</td>
<td>Suranga Lakmal</td>
<td>Lasith Malinga</td>
<td>Kusal Perera</td>
<td>Thisara Perera</td>
<td>Sangakkara</td>
</tr>
<tr>
<td>New Zealand</td>
<td>Brendon McCullum</td>
<td>Corey Anderson</td>
<td>Trent Boult</td>
<td>Grant Elliott</td>
<td>Martin Guptill</td>
<td>Nathan McCullum</td>
<td>Kyle Mills</td>
<td>Luke Ronchi</td>
<td>Tim Southee</td>
<td>Ross Taylor</td>
```

```

<td>Dan Vettori</td>
</tr>
</tbody>
</table>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **scroll_table.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Scroll Table Example					
Team	Player One	Player Two	Player Three	Player Four	Player Five
India	Mahendra Singh Dhoni	Virat Kohli	Ajinkya Rahane	Shikhar Dhawan	Rohit Sharma
Srilanka	Angelo Mathews	Thirimanne	Tillakaratne Dilshan	Mahela Jayawardene	Tharindu Kaushal
New Zealand	Brendon McCullum	Corey Anderson	Trent Boult	Grant Elliott	Martin Guptill

Foundation – Table SASS Reference

Variables

You can change the styles of the components by using the following SASS variables as listed in the table.

S. No.	Name & Description	Type	Default Value
1	\$table-background Applies the background color for the table.	Color	\$white

2	\$table-color-scale It specifies the scale for darkening the striped table rows and border.	Number	5%
3	\$table-border It defines the style for the table border.	List	1px solid smart-scale(\$table-background, \$table-color-scale)
4	\$table-padding It provides padding of the table.	Number	rem-calc(8 10 10)
5	\$table-hover-scale It specifies the scale for darkening the striped table rows on hover.	Number	2%
6	\$table-row-hover Applies the color for standard rows on hover.	List	darker(\$table-background, \$table-hover-scale)
7	\$table-row-stripe-hover Applies the color for striped rows on hover.	List	darker(\$table-background, \$table-color-scale + \$table-hover-scale)
8	\$table-striped-background It provides the background color for striped rows.	Color	smart-scale(\$table-background, \$table-color-scale)
9	\$table-stripe It shows the stripe on table rows. If a row is even, then even rows will have a background color and if a row is odd, then odd rows will have a background color. If row is empty or any other value, then the table rows will have no stripes.	Keyword	even
10	\$table-head-background It specifies the color of header background.	Color	smart-scale(\$table-background, \$table-color-scale / 2)
11	\$table-foot-background It specifies the color of footer background.	Color	smart-scale(\$table-background, \$table-color-scale)
12	\$table-head-font-color It defines the font color of a header.	Color	\$body-font-color
13	\$show-header-for-stacked It shows the default value for the header when using stacked tables.	Boolean	false

Mixins

You can use the mixins to build the CSS class structure for your components as described below:

table

It adds styles for tables and styles can be included using the following mixin:

```
@include table($stripe);
```

It uses the parameter as defined in the following table:

Parameter & Description	Type	Default Value
\$stripe It shows the stripe on table rows using even, odd or none value. Even is the default value.	Keyword	\$table-stripe

table-scroll

It scrolls the table horizontally by using the following mixin:

```
@include table-scroll;
```

table-hover

It hovers on the table rows by using the following mixin:

```
@include table-hover;
```

table-stack

It adds styles for the stacked table.

```
@include table-stack($header);
```

It uses the parameter as defined in the following table:

Parameter & Description	Type	Default Value
\$header It shows the header when table is stacked.	Boolean	\$show-header-for-stacked

Foundation – Tabs

Description

It is a navigation based tab that display the content into different panes without leaving the page.

The following table lists some of the types of foundation tabs:

S. No.	Type & Description
1	Basic Tab It provides navigation menu that displays the content into different panes.
2	Vertical Tabs It displays the tab vertically.
3	Sass Reference You can change the styles of the table components by using the SASS variables.
4	JavaScript Reference Foundation provides JavaScript components for the tabs.

Foundation – Basic Tabs

Description

It provides navigation menu that displays the content into different panes and each pane can be viewed at a time.

Example

The following example demonstrates the use of *basic tabs* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Basic Tabs</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Basic Tabs Example</h2>
  <ul class="tabs" data-tabs id="tabs_example">
    <li class="tabs-title is-active"><a href="#tab1">Player 1</a></li>
    <li class="tabs-title"><a href="#tab2">Player 2</a></li>
```

```

<li class="tabs-title"><a href="#tab3">Player 3</a></li>
<li class="tabs-title"><a href="#tab4">Player 4</a></li>
<li class="tabs-title"><a href="#tab5">Player 5</a></li>
</ul>
<div class="tabs-content" data-tabs-content="tabs_example">
<div class="tabs-panel is-active" id="tab1">
<p>First Player</p>
<p>Sachin Tendulkar</p>
</div>
<div class="tabs-panel" id="tab2">
<p>Second Player</p>
<p>M S Dhoni</p>
</div>
<div class="tabs-panel" id="tab3">
<p>Third Player</p>
<p>Shane Warne</p>
</div>
<div class="tabs-panel" id="tab4">
<p>Fourth Player</p>
<p>Shaun Pollock</p>
</div>
<div class="tabs-panel" id="tab5">
<p>Five Player</p>
<p>Adam Gilchrist</p>
</div>
</div>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **basic_tabs.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Basic Tabs Example

The screenshot shows a horizontal navigation bar with five tabs labeled "Player 1", "Player 2", "Player 3", "Player 4", and "Player 5". The tab "Player 3" is highlighted with a blue background and white text. Below the tabs, the content area displays the text "First Player" and "Sachin Tendulkar".

Foundation – Vertical Tabs

Description

It displays the tab vertically.

Example

The following example demonstrates the use of *vertical tabs* in Foundation:

```
<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Vertical Tabs</title>
    <link rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
        src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
        src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
    <h2>Vertical Tabs Example</h2>
    <div class="row collapse">
        <div class="medium-3 columns">
            <ul class="tabs vertical" data-tabs id="tabs_example">
                <li class="tabs-title is-active"><a href="#tab1">Player 1</a></li>
                <li class="tabs-title"><a href="#tab2">Player 2</a></li>
                <li class="tabs-title"><a href="#tab3">Player 3</a></li>
```

```

<li class="tabs-title"><a href="#tab4">Player 4</a></li>
<li class="tabs-title"><a href="#tab5">Player 5</a></li>
</ul>
</div>
<div class="medium-9 columns">
<div class="tabs-content vertical" data-tabs-content="tabs_example">
<div class="tabs-panel is-active" id="tab1">
<p>First Player</p>
<p>Sachin Tendulkar</p>
</div>
<div class="tabs-panel" id="tab2">
<p>Second Player</p>
<p>M S Dhoni</p>
</div>
<div class="tabs-panel" id="tab3">
<p>Third Player</p>
<p>Shane Warne</p>
</div>
<div class="tabs-panel" id="tab4">
<p>Fourth Player</p>
<p>Shaun Pollock</p>
</div>
<div class="tabs-panel" id="tab5">
<p>Five Player</p>
<p>Adam Gilchrist</p>
</div>
</div>
</div>
</script>
$(document).ready(function() {
  $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **vertical_tabs.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Vertical Tabs Example

Player 1	First Player
Player 2	Sachin Tendulkar
Player 3	
Player 4	
Player 5	

Foundation – Tabs SASS Reference

Variables

You can change the styles of the components by using the following SASS variables as listed in the table.

S. No.	Name & Description	Type	Default Value
1	\$tab-margin It sets the margin for the tab bar.	Number	0
2	\$tab-background It displays the background color of a tab bar.	Color	\$white
3	\$tab-background-active It displays the active background color of a tab bar.	Color	\$light-gray
4	\$tab-border It specifies the border color for a tab.	Color	\$light-gray
5	\$tab-item-color It displays the text color for menu items.	Color	foreground(\$tab-background, \$primary-color)

6	\$tab-item-background-hover It displays the background color for items on hover.		\$white
7	\$tab-item-padding It defines the padding of a tab item.	Number	1.25rem 1.5rem
8	\$tab-expand-max It includes maximum number of expand-n classes in the CSS.	Number	6
9	\$tab-content-background It displays the background color of the tab content.	Color	\$white
10	\$tab-content-border It displays the border color of the tab content.	Color	\$light-gray
11	\$tab-content-color It provides the text color of the tab content.	Color	foreground(\$tab-background, \$primary-color)
12	\$tab-content-padding It defines the padding for the tab content.	Number or List	1rem

Mixins

You can use the mixins to build the CSS class structure for your reveal modal components.

tabs-container

You can include styles for a tab container using the following mixin:

```
@include tabs-container;
```

tabs-container-vertical

It displays the tabs in vertical manner by using the following mixin:

```
@include tabs-container-vertical;
```

tabs-title

It applies the styles for the tab links by using the following mixin:

```
@include tabs-title;
```

tabs-content

It applies the styles for tab content panes by using the following mixin:

```
@include tabs-content;
```

tabs-content-vertical

It specifies the tab, content must display, in a vertical manner by using the following mixin:

```
@include tabs-content-vertical;
```

tabs-panel

You can include styles for a tab content panel by using the following mixin:

```
@include tabs-panel;
```

Foundation – Tabs JavaScript Reference

Foundation provides JavaScript components for the tabs as listed below:

Initializing

You can initialize the pane in JavaScript by using the *foundation.tabs.js* and *foundation.core.js* plugins. The *foundation.core.js* plugin requires the following libraries:

- **foundation.util.keyboard.js**
- **foundation.util.timerAndImageLoader.js**

Foundation.Tabs

It specifies the instance of tabs as defined below:

```
var my_element = new Foundation.Tabs(element);
```

The tabs include the following values:

Name & Description	Type
element It uses the jQuery object for the tabs.	Number
options It overrides to the default plugin settings for the tabs.	Object

Plugin Options

You can use the following plugin options to customize the tabs instance.

S. No.	Name & Description	Example
1	autoFocus If you set this plugin to true, window will scroll to the content of active pane on load.	false
2	wrapOnKeys It wraps the content around tab links using keyboard input.	true
3	matchHeight It matches the height of tab content panes by setting it to true.	false
4	linkClass It is applied to the 'li's in tab link list.	'tabs-title'

5	panelClass It is applied to the content containers.	'tabs-panel'
---	---	--------------

Events

Tabs provide the following event as listed in the table:

Event	Description
change.zf.tabs	It triggers when the plugin changes the tab successfully.

Functions

Tabs provide the following functions as defined below:

.handleTabChange

It handles the `$targetContent` tab specified by `$target` function as shown in the following table:

S. No.	Name & Description	Type
1	\$target It will open the tab.	jQuery

.selectTab

It is used to select the content pane for displaying the content and can be specified as shown in the following table:

S. No.	Name & Description	Type
1	elem It uses jQuery object or id of the pane for displaying content pane.	jQuery

.destroy

It is used to destroy the tab instance.

19. Foundation – Media

In this chapter we will study about *media* in Foundation. The Foundation consists of several media types like *Flex Video*, *Label*, *orbit*, *progress bar* and *tooltip*. The following table lists all the media types.

S. No.	Media types & Description
1	Flex Video It is used to embed videos in a flex video container to keep the right aspect ratio despite the screen size.
2	Label It can be used for inline styling which can be put into body to invoke specific section or to attach metadata.
3	Orbit It is a powerful and responsive slider, allowing a user to swipe on touch-screen devices.
4	Progress Bar It is used to display your progress and can be added to your layouts.
5	Tooltips It is used to display extra information for a term or action on a page.

Foundation – Flex Video

The flex video is used to embed videos in a container to keep the correct aspect ratio despite the screen size.

The following table lists the classes used to embed the videos.

S. No.	Flex Video Details & Description
1	flex-video As the width of the screen changes, this class can be used to maintain the aspect ratio of the embedded videos.
2	widescreen By using this class you can set the embedded video's ratio to 16:9.
3	SASS Reference You can change the styles of the components by using SASS Reference.

The *Vimeo* videos are distinctive snowflakes of their own. You can add *.vimeo* class, to wrap *vimeo* embed in flex video container.

Foundation – Flex Video Class

Description

As the width of the screen changes, `.flex-video` class can be used to maintain aspect ratio of the embedded videos.

Example

The following example demonstrates how to embed video in Foundation.

```
<html>
<head>
    <title>Flex Video</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
</head>
<body>
    <div class="flex-video">
        <iframe width="420" height="315"
src="/foundation/images/kitchen_sink/video.mp4" frameborder="0"
allowfullscreen></iframe>
    </div>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script>
        $(document).ready(function() {
            $(document).foundation();
        })
    </script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **flex-video-class.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Flex Video Class

Description

By default, the aspect ratio is 4:3; Using *widescreen* class you can set the embedded video's ratio to 16:9.

Example

The following example demonstrates how to use the *widescreen* class to embed video in Foundation.

```
<html>
<head>
    <title>Flex Video Widescreen</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
</head>
<body>
    <div class="flex-video widescreen">
```

```

<iframe width="420" height="315"
src="/foundation/images/kitchen_sink/video.mp4" frameborder="0"
allowfullscreen></iframe>
</div>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **widescreen-class.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Flex Video SASS Reference

Description

You can change the styles of the components by using SASS Reference.

Variables

The following table lists the SASS variables in a project's settings file that make the component's default styles to get customized.

S. No.	Name & Description	Type	Default Value
1	\$flexvideo-padding-top It represents the padding above the container of flex video.	Number	rem-calc(25)
2	\$flexvideo-margin-bottom It represents the margin below the container of flex video.	Number	rem-calc(16)
3	\$flexvideo-ratio It represents the padding used to create an aspect ratio of 4:3.	Number	4 by 3
4	\$flexvideo-ratio-widescreen It represents the padding used to create an aspect ratio of 16:9.	Number	16 by 9

Mixins

To build this component's final CSS output, the following mixins can be used. To build your own class structure using Foundation components, you can use the mixins yourself.

flex-video

```
@include flex-video($ratio);
```

It is used to create a flex video container. It accepts `$ratio` parameter which is explained below.

S. No.	Parameter & Description	Type	Default Value
1	\$ratio It represents the container's ratio and formatted as <code>x by y</code> .	List	<code>\$flexvideo-ratio</code>

Functions

flex-video

```
flex-video($ratio)
```

It creates a height in percentage, which is used as padding in container of flex video. It accepts the parameter `$ratio`, which is of type *List*. The parameter `$ratio` represents the container's ratio and formatted as x by y .

Foundation – Media Label

Labels can be used for inline styling, which can be put into body to invoke specific section or to attach metadata. For instance, label can intimate you when something gets updated.

The following table shows how these labels can be used.

S. No.	Label details & Description
1	Basics To create a label, you have to add <code>.label</code> class to an element.
2	Coloring Coloring the labels can be done by using the classes used for buttons and with icons as well.
3	SASS Reference You can change the styles of the components by using SASS Reference.

Foundation – Label Basics

Description

To create a label, you have to add the `.label` class to an element. For instance, you can use the `` tag or even other tags work good. Labels can be used to describe another element. Binding two elements together can be done by providing a label, a unique ID and on the main element, reference the ID in `aria-describedby` attribute.

Example

The following example demonstrates how to use `label` in Foundation.

```

<html>
  <head>
    <title>Label Basics</title>
    <link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  </head>
  <body>
    <span class="label">Default</span>
    <p aria-describedby="email">Re: re: A Christmas gift for you!</p>
    <span class="label" id="email">Maximum Priority</span>
  </body>
</html>

```

```

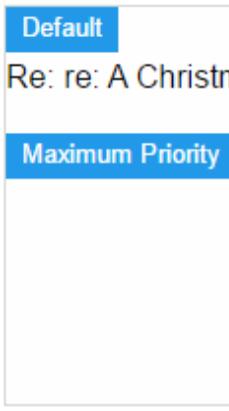
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **label_basics.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



When multiple labels describe an element, you have to put multiple IDs within *aria-describedby*. For more information, go through Label Multiple IDs.

Foundation – Label Multiple IDs

Description

When multiple labels describe an element, you have to put multiple IDs within *aria-describedby* attribute.

Example

The following example demonstrates how to use *multiple labels* in Foundation.

```

<html>
<head>
    <title>Label Multiple IDs</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
</head>
<body>
    <p aria-describedby="email1 email2">Re: re: A Christmas gift for you!</p>
    <span class="label" id="emaill">Maximum Priority<span>
    <span class="label" id="email2">Unread<span>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
in.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
js"></script>
    <script>
        $(document).ready(function() {
            $(document).foundation();
        })
    </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **label_multipleID.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Re: re: A Christmas gift for you!

Maximum Priority Unread

Foundation – Label Coloring

Description

Coloring the labels can be done by using the classes used for buttons and with icons as well. Any icon fonts or image-based icons can be used.

Example

The following example demonstrates how to use *label coloring* in Foundation.

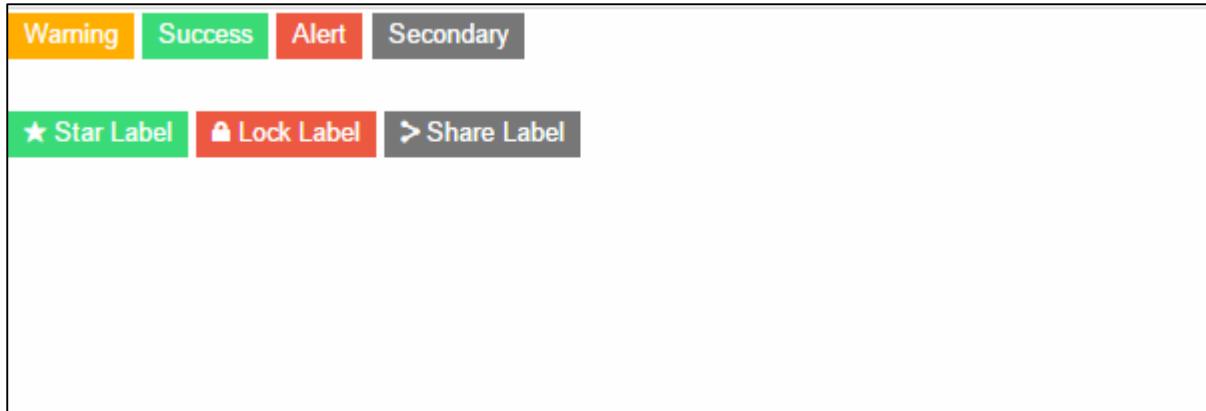
```
<html>
<head>
    <title>Label Coloring</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundicons/3.0.0/foundation-
icons.css">
</head>
<body>
    <div>
        <span class="warning label">Warning</span>
        <span class="success label">Success</span>
        <span class="alert label">Alert</span>
        <span class="secondary label">Secondary</span>
    </div><br>
    <div>
        <span class="success label"><i class="fi-star"></i> Star Label</span>
        <span class="alert label"><i class="fi-lock"></i> Lock Label</span>
        <span class="secondary label"><i class="fi-share"></i> Share Label</span>
    </div>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
in.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
js"></script>
    <script>
        $(document).ready(function() {
            $(document).foundation();
        })
    </script>
</body>
```

```
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **label_coloring.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Label SASS Reference

Description

You can change the styles of the components by using SASS Reference.

Variables

The following table lists the SASS variables in the project's settings file that makes this component's default styles to get customized.

S. No.	Name & Description	Type	Default Value
1	\$label-background It represents the label's default background color.	Color	\$primary-color
2	\$label-color It represents the label's default text color.	Color	foreground(\$label-background)
3	\$label-font-size It represents the label's default font size.	Number	0.8rem
4	\$label-padding It represents the label's default internal padding.	Number	0.33333rem 0.5rem

5	\$label-radius It represents the label's default radius.	Number	\$global-radius
---	--	--------	-----------------

Mixins

To build this component's final CSS output, the following mixins can be used. To build your own class structure using Foundation components, you can use the mixins yourself.

label

```
@include label;
```

It is used to generate a label's base styles.

Foundation – Media Orbit

Orbit is a powerful and responsive slider, allowing a user to swipe on touch-screen devices.

The following table shows how the orbit can be used.

S. No.	Orbit Details & Description
1	Basics The orbit created by adding <i>data-orbit</i> to your <i></i> tag.
2	SASS Reference You can change the styles of the components by using SASS Reference.
3	JavaScript Reference Foundation provides JavaScript components for the orbit.

Foundation – Media Orbit Basics

Description

The orbit is created by adding *data-orbit* to your ** tag. If animations are added to the slider, then you need to use the *motion-ui* plugin. You can change the default animations by setting the animation options with *data-options*.

Example

The following example demonstrates how to use *orbit* in Foundation.

```
<html>
<head>
  <title>Orbit Basics</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
```

```

<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.5/css/app.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/motion-
ui/1.1.1/motion-ui.css">
</head>
<body>
<div class="orbit" role="region" aria-label="Space Images" data-orbit>
  <ul class="orbit-container">
    <button class="orbit-previous" aria-label="previous"><span class="show-
for-sr">Previous Slide</span><</button>
    <button class="orbit-next" aria-label="next"><span class="show-for-
sr">Next Slide</span>></button>
    <li class="is-active orbit-slide">
      
      <figcaption class="orbit-caption">Milkyway</figcaption>
    </li>
    <li class="orbit-slide">
      
      <figcaption class="orbit-caption">Satellite</figcaption>
    </li>
    <li class="orbit-slide">
      
      <figcaption class="orbit-caption">Rocket</figcaption>
    </li>
    <li class="orbit-slide">
      
      <figcaption class="orbit-caption">Moon</figcaption>
    </li>
  </ul>
  <nav class="orbit-bullets">
    <button class="is-active" data-slide="0"><span class="show-for-
sr">First slide.</span><span class="show-for-sr">Current Slide</span></button>
    <button data-slide="1"><span class="show-for-sr">Second
slide.</span></button>
    <button data-slide="2"><span class="show-for-sr">Third
slide.</span></button>
    <button data-slide="3"><span class="show-for-sr">Fourth
slide.</span></button>
  </nav>
</div>

```

```

        </div>

        <div class="orbit" role="region" aria-label="Favorite Space Pictures" data-
orbit data-use-m-u-i="false">
            <ul class="orbit-container">
                <button class="orbit-previous" aria-label="previous"><span class="show-
for-sr">Previous Slide</span><</button>
                <button class="orbit-next" aria-label="next"><span class="show-for-
sr">Next Slide</span>></button>
                <li class="is-active orbit-slide">
                    <div>
                        <h3 class="text-center">Hi! Here is your first slider.</h3>
                    </div>
                </li>
                <li class="orbit-slide">
                    <div>
                        <h3 class="text-center">Hi! Here is your second slider.</h3>
                    </div>
                </li>
                <li class="orbit-slide">
                    <div>
                        <h3 class="text-center">Hi! Here is your third slider.</h3>
                    </div>
                </li>
                <li class="orbit-slide">
                    <div>
                        <h3 class="text-center">Hi! Here is your fourth slider.</h3>
                    </div>
                </li>
            </ul>
            <nav class="orbit-bullets">
                <button class="is-active" data-slide="0"><span class="show-for-
sr">First slide.</span><span class="show-for-sr">Current Slide</span></button>
                <button data-slide="1"><span class="show-for-sr">Second
slide.</span></button>
                <button data-slide="2"><span class="show-for-sr">Third
slide.</span></button>
                <button data-slide="3"><span class="show-for-sr">Fourth
slide.</span></button>
            </nav>
        </div>
    
```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/5.5.2/js/foundation/foun
dation.orbit.min.js"></script>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
in.js"></script>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
js"></script>

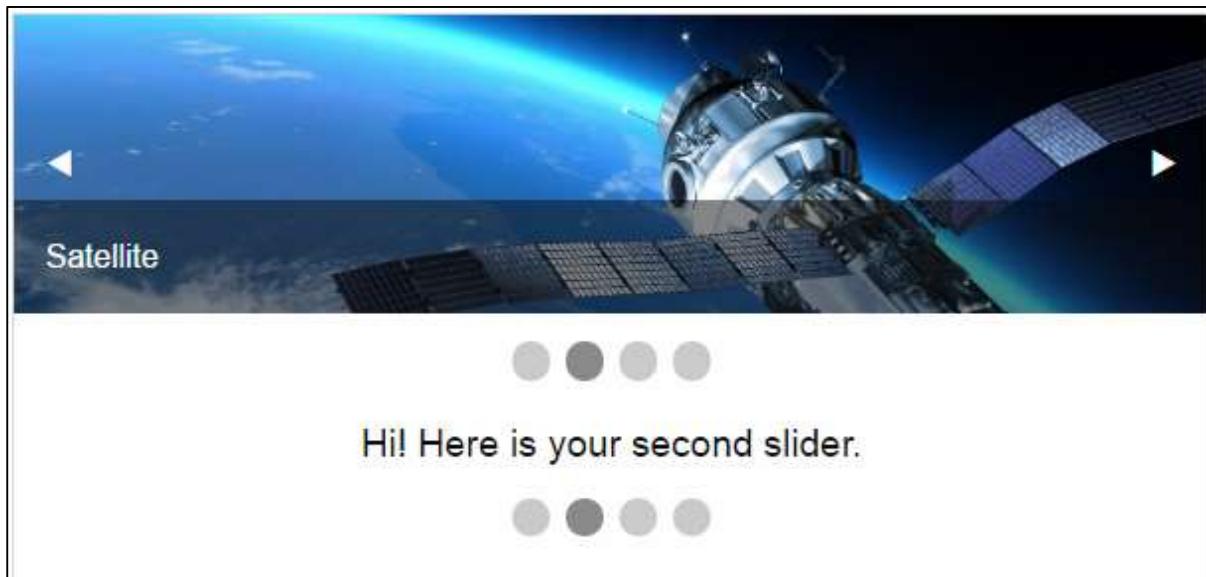
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **orbit_basics.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Media SASS Reference

Description

You can change the styles of the components by using SASS Reference.

369

Variables

The following table lists SASS variables in a project's settings file that makes this component's default styles to get customized.

S. No.	Name & Description	Type	Default Value
1	\$orbit-bullet-background It represents the default color for bullets of the orbit.	Color	\$medium-gray
2	\$orbit-bullet-background-active It represents the default active color for bullets of the orbit.	Color	\$dark-gray
3	\$orbit-bullet-diameter It represents the default diameter for bullets of the orbit.	Number	1.2rem
4	\$orbit-bullet-margin It represents the default margin between bullets of the orbit.	Number	0.1rem
5	\$orbit-bullet-margin-top It represents the default distance from slide region for bullets of the orbit.	Number	0.8rem
6	\$orbit-bullet-margin-bottom It represents the default bottom margin from bullets to the content below it.	Number	0.8rem
7	\$orbit-caption-background It represents the default background color for caption of the orbit.	Color	rgba(\$black, 0.5)
8	\$orbit-caption-padding It represents the default padding for caption of the orbit.	Number	1rem
9	\$orbit-control-background-hover It represents the default background color for controls when hovered.	Color	rgba(\$black, 0.5)
10	\$orbit-control-padding It represents the default padding for controls of the orbit.	Number	1rem
11	\$orbit-control-zindex It represents the default z-index for controls of the orbit.	Number	10

Mixins

To build this component's final CSS output, the following mixins can be used. To build your own class structure using Foundation components, you can use the mixins yourself.

orbit-wrapper

```
@include orbit-wrapper;
```

It adds styles to the wrapper of the orbit.

orbit-container

```
@include orbit-container;
```

It adds styles to the internal orbit slide container. It is used on class *.orbit-container*.

orbit-slide

```
@include orbit-slide;
```

It adds styles to the separate slides of a slider. It is used on class *.orbit-slide*.

orbit-caption

```
@include orbit-caption;
```

It adds styles to the slide caption.

orbit-control

```
@include orbit-control;
```

It adds base styles to the next and previous buttons. Styles are divided between the *.orbit-next* and *.orbit-previous* classes in default CSS.

orbit-previous

```
@include orbit-previous;
```

It adds styles to the previous button. These are used on class *.orbit-previous*.

orbit-next

```
@include orbit-next;
```

It adds styles to the next button. These are used on class *.orbit-next*.

orbit-bullets

```
@include orbit-bullets;
```

It adds styles to a container of bullets of the orbit and add styles to the previous button and are used on *.orbit-bullets* class.

Foundation – Media JavaScript Reference

Description

Foundation provides JavaScript components for the orbit as listed below.

Initializing

To use the orbit plugin, the `foundation.core.js`, `foundation.orbit.js` files should be included in your JavaScript. This plugin requires the following utility libraries:

- **`foundation.util.keyboard.js`**
- **`foundation.util.motion.js`**
- **`foundation.util.timerAndImageLoader.js`**
- **`foundation.util.touch.js`**

Foundation.Orbit

It is used to create new instance of a carousel of the orbit.

```
var elem = new Foundation.Orbit(element);
```

S. No.	Name & Description	Type
1	element It is a jQuery object to make into a carousel.	jQuery
2	options It overrides the default settings of the plugin.	Object

Plugin Options

You can use these to customize an orbit's instance. It can be set as separate data attributes, one merged `data-options` attribute or as an object passed to the constructor of the plugin. The following table lists the plugin options used in Foundation.

S. No.	Name & Description	Example
1	bullets It tells the JS to load bullets.	true
2	navButtons It tells the JS to add event listeners to nav buttons.	true
3	animInFromRight motion-ui animation class to be applied.	'slide-in-right'
4	animOutToRight motion-ui animation class to be applied.	'slide-out-right'
5	animInFromLeft motion-ui animation class to be applied.	'slide-in-left'

6	animOutToLeft motion-ui animation class to be applied.	'slide-out-left'
7	autoPlay It automatically allows orbit to animate on page load.	true
8	timerDelay It represents time, in ms, which will be applied for slide transition.	5000
9	infiniteWrap It infinitely loops via slides.	true
10	swipe It allows orbit slides to bind the swipe events for mobile.	true
11	pauseOnHover It allows the timing functions to pause the animation on hovering on it.	true
12	accessible It binds the keyboard events to the slider.	true
13	containerClass Class applied to the orbit's container.	'orbit-container'
14	slideClass Class applied to individual slides of the orbit.	'orbit-slide'
15	boxOfBullets Class applied to the container of the bullet.	'orbit-bullets'
16	nextClass Class applied to the next button.	'orbit-next'
17	prevClass Class applied to the previous button.	'orbit-previous'
18	useMUI It sets Boolean to flag the JS to use motion ui classes or not. For backwards compatibility defaults to true.	true

Events

These events will trigger from any element with a plugin attached. The `slidechange.zf.orbit` will fire when the animation is finished by the slide.

Functions

The `.geoSync` sets a `timer` object and starts the counter for the next slide.

.changeSlide

It changes the slide into a new one. It triggers `Orbit#event` events for slide change. The following table lists the events, which will be triggered.

S. No.	Name & Description	Type
1	isLTR It flags when the slide must move from left to right.	Boolean
2	chosenSlide jQuery elements of the slide to display next, when one is selected.	jQuery
3	idx It represents the new slide index in its collection, when one is chosen.	Number

The `.destroy` destroys the carousel and hides the element.

Foundation – Progress Bar Basics

Description

Progress bar contains two elements- container, `.progress` and a meter, `.progress-meter`. The status of the progress bar is simplified by `role` and `aria-` attributes.

The following list clarifies the purpose of the attributes:

- **aria-valuemin:** It represents the minimum value for the progress bar.
- **aria-valuemax:** It represents the maximum value for the progress bar.
- **aria-valuenow:** It represents the current value for the progress bar.
- **aria-valuetext:** If the progress bar's value is not numeric, add this attribute, which must contain a human readable version of value of the progress bar.

To fill the progress bar, you have to add the `width` CSS property to the inner meter.

Example

The following example demonstrates how to use `progress bar` in Foundation.

```
<html>
<head>
  <title>Progress Bar</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
</head>
<body>
  <h2>Progress Bar</h2>
  <div class="progress" role="progressbar" tabindex="0" aria-valuenow="0"
  aria-valuemin="0" aria-valuemax="100">
    <div class="progress-meter"></div>
  </div>
  <div class="progress" role="progressbar" tabindex="0" aria-valuenow="80"
  aria-valuemin="0" aria-valuetext="80 percent" aria-valuemax="100">
```

```

<div class="progress-meter" style="width: 80%"></div>
</div>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
<script>
$(document).ready(function() {
  $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **progress_basics.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Progress Bar



Foundation – Progress Bar Basics

Description

Styling the progress bar can be done by using the color classes such as `.success`, `.warning` and `.alert`

Example

The following example demonstrates how to *color the progress bar* in Foundation.

```
<html>
```

```

<head>
    <title>Progress Bar Coloring</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
</head>
<body>
    <h2>Progress Bar</h2>
    <div class="warning progress" role="progressbar" tabindex="0" aria-
    valuenow="25" aria-valuemin="0" aria-valuetext="25 percent" aria-
    valuemax="100">
        <div class="progress-meter" style="width: 25%"></div>
    </div>
    <div class="alert progress">
        <div class="progress-meter" style="width: 50%"></div>
    </div>
    <div class="success progress">
        <div class="progress-meter" style="width: 75%"></div>
    </div>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
in.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
js"></script>
    <script>
        $(document).ready(function() {
            $(document).foundation();
        })
    </script>
</body>
</html>

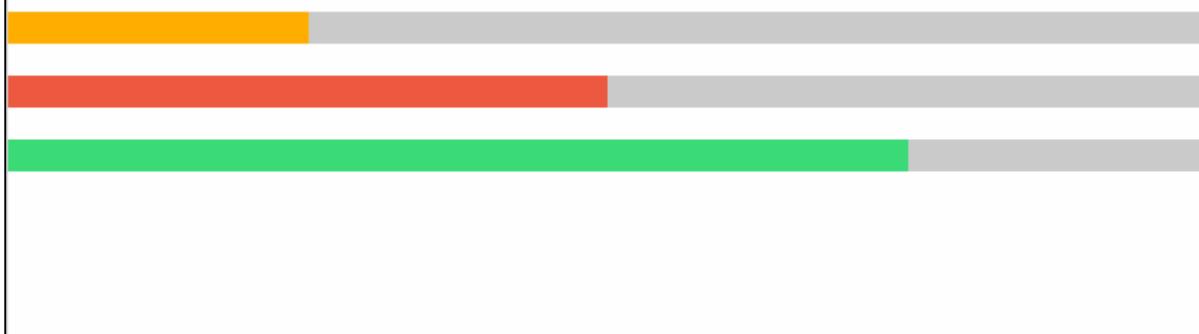
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **progress_colors.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Progress Bar



Foundation – Progress Bar With Text

Description

Progress bar can be used with text by using the *progress-meter-text* class.

Example

The following example demonstrates the use of *progress bar with text* in Foundation.

```
<html>
<head>
    <title>Progress Bar With Text</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
    s">
</head>
<body>
    <h2>Progress Bar</h2>
    <div class="progress" role="progressbar" tabindex="0" aria-valuenow="50"
    aria-valuemin="0" aria-valuetext="50 percent" aria-valuemax="100">
        <span class="progress-meter" style="width: 50%">
            <p class="progress-meter-text">50%</p>
        </span>
    </div>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
    in.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
    js"></script>
    <script>
        $(document).ready(function() {
            $(document).foundation();
    
```

```

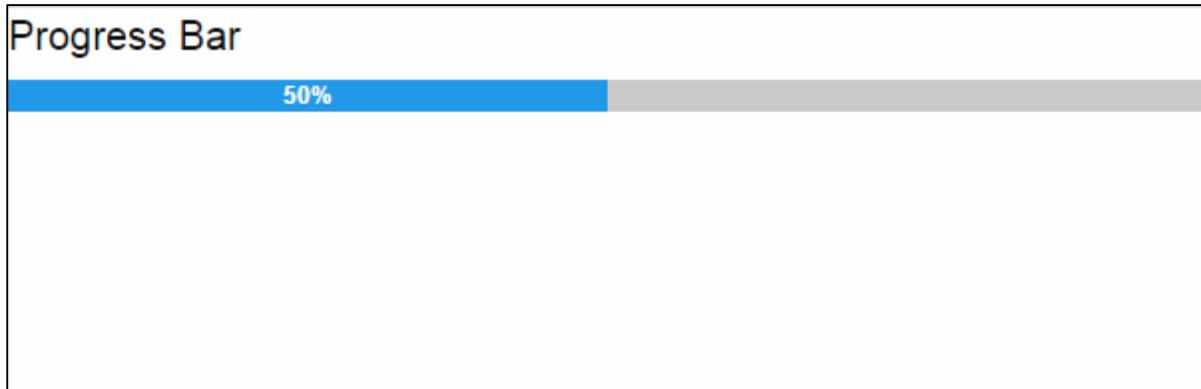
        })
    </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **progress_with_text.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Progress Bar SASS Reference

Description

You can change the styles of the components by using SASS Reference.

Variables

The following table lists the SASS variables in a project's settings file that makes this component's default styles to get customized.

S. No.	Name & Description	Type	Default Value
1	\$progress-height It represents the progress bar's height.	Number	1rem
2	\$progress-background It represents the progress bar's background color.	Color	\$medium-gray
3	\$progress-margin-bottom It represents the progress bar's bottom margin.	Number	\$global-margin
4	\$progress-meter-background It represents the default color of the meter of the progress bar.	Color	\$primary-color

5	\$progress-radius It represents the default progress bar's radius.	Number	\$global-radius
---	--	--------	-----------------

Mixins

To build this component's final CSS output, the following mixins can be used. To build your own class structure using Foundation components, you can use the mixins yourself.

progress-container

```
@include progress-container;
```

It adds styles to the container of the progress bar.

progress-meter

```
@include progress-meter;
```

It adds styles to the progress bar's inner meter.

progress-meter-text

```
@include progress-meter-text;
```

It adds styles to the text in meter of the progress bar.

Foundation – Media Tooltips

Tooltip is used to display an extra information for a term or action on a page. The following table shows how to use *tooltips* in Foundation.

S. No.	Tooltip Details & Description
1	Tooltip Tooltip can be created by using <i>data-tooltip</i> in your <i></i> tag.
2	SASS Reference You can change the styles of the components by using SASS Reference.
3	JavaScript Reference Foundation provides JavaScript components for the orbit.

Foundation – Media Tooltips Basics

Description

Tooltip can be created by using *data-tooltip* in your ** tag. By default, tooltip appears below the definition but you can make your tooltip to appear on *left*, *right* or *top* of the definition.

Example

The following example demonstrates the use of *tooltip* in Foundation.

```

<html>
<head>
    <title>Tooltip</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
</head>
<body>
    <p>This an example for <span data-tooltip aria-haspopup="true" class="has-tip" data-disable-hover='false' tabindex=1 title="Below">Tooltip</span> which displays below.</p>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
    <p>This an example for <span data-tooltip aria-haspopup="true" class="has-tip top" data-disable-hover="false" tabindex=2 title="Top">Tooltip</span> which displays on top.</p>
    <p>This an example for <span data-tooltip aria-haspopup="true" class="has-tip right" data-disable-hover="false" tabindex="3" title="Right">Tooltip</span> which displays right. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. This an example for <span data-tooltip aria-haspopup="true" class="has-tip left" data-disable-hover="false" tabindex="4" title="Left">Tooltip</span> which displays left.</p>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script>
        $(document).ready(function() {
            $(document).foundation();
        })
    </script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **media_tooltips.html** file.

- Open this HTML file in a browser, an output is displayed as shown below.

This an example for **Tooltip** which displays below.

.....
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

This an example for **Tooltip** which displays on top.

.....
This an example for **Tooltip** which displays right. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. This an example for **Tooltip** which displays left.

Foundation – Tooltip SASS Reference

Description

You can change the styles of the components by using SASS Reference.

Variables

The following table lists the SASS variables in a project's settings file that makes this component's default styles to get customized.

S. No.	Name & Description	Type	Default Value
1	\$tooltip-background-color It represents the default background color of the tooltip.	Color	\$black
2	\$tooltip-color It represents the default font color of the tooltip.	Color	\$white
3	\$tooltip-padding It represents the default padding of the tooltip background.	Number	0.75rem
4	\$tooltip-font-size It represents the tooltip text's default font size.	Number	\$small-font-size
5	\$tooltip-pip-width It represents the default pip width.	Number	0.75rem

6	\$tooltip-pip-height It represents the default pip height which is useful for calculating the distance of tooltip from its word.	Number	\$tooltip-pip-width * 0.866
7	\$tooltip-pip-offset It represents the default pip tooltip's offset. This shows at what distance pip is indented from the tooltip's left edge.	Number	1.25rem
8	\$tooltip-radius It represents the tooltip's default radius.	Number	\$global-radius

Foundation – Tooltip JavaScript Reference

Description

Foundation provides JavaScript components for the tooltip as listed below.

Initializing

To use this plugin, the `foundation.core.js`, `foundation.tooltip.js` files should be included in your JavaScript. This plugin requires the following utility libraries:

- **foundation.util.box.js**
- **foundation.util.triggers.js**

Foundation.Tooltip

It creates a new tooltip's instance.

```
var elem = new Foundation.Tooltip(element);
```

It triggers the `Tooltip#event:init` events and accepts the following parameters.

S. No.	Name & Description	Type
1	element It is the jQuery object used to attach a tooltip to.	jQuery
2	options To extend the default configuration, this object is used.	Object

Plugin Options

You can use these to customize a tooltip's instance. It can be set as separate data attributes, one merged `data-options` attribute or as an object passed to the constructor of the plugin. The following table lists the plugin options used in Foundation.

S. No.	Name & Description	Example

1	hoverDelay It represents time in ms, to be taken before a tooltip must open on hover.	200
2	fadeInDuration150 It represents the time to fade into view.	150
3	fadeOutDuration It represents the time to fade out of view.	150
4	disableHover It disables the hover events from opening the tooltip when set to true.	false
5	templateClasses Additional classes to apply to the tooltip template, which is optional.	'my-cool-tip-class'
6	tooltipClass Non-optional class added to templates of the tooltip.	'tooltip'
7	triggerClass Class applied to the anchor element of the tooltip.	'has-tip'
8	showOn Minimum size of the breakpoint at which to open the tooltip.	'small'
9	template Custom template to be used to create tooltip markup.	"
10	tipText Text outputted in tooltip template on opening.	'Some cool space fact here.'
11	clickOpen It opens the tooltip when fired with a click or touch event.	true
12	positionClass Additional classes for positioning, set by JS.	'top'
13	vOffset It represents the distance (pixels) in which the template must push from the anchor on Y axis.	10
14	hOffset It represents the distance (pixels) in which the template must push from the anchor on X axis, when aligned to a side.	12

Events

These events will be triggered from any element with the plugin of the tooltips attached. The following table lists the events with a brief description.

S. No.	Name & Description
--------	--------------------

1	tooltip.zf.tooltips Triggers to close all other tooltips that are opened on the page.
2	show.zf.tooltips Triggers when tooltip is displayed.
3	hide.zf.tooltips Triggers when tooltip is hidden.

Functions

.show

It reveals the tooltip and triggers an event to close other tooltip. It triggers the *Closeeme#event:tooltip* and *Tooltip#event:show* events.

.hide

It hides the present tooltip and resets the positioning class when it changes because of the collision. It triggers the *Tooltip#event:hide* events.

.toggle

In addition to the static show() and hide() functions, .*toggle* adds a toggle method.

.destroy

It destroys a tooltip instance and the template element is removed from the view.

20. Foundation – Plugins

Description

Plugin is a software that provides additional functionality which was not originally completed by Foundation core functionality. Foundation Plugins can be uploaded to expand the functionality of the site. Plugins are used to make your work easier.

The following table describes the different types of Plugins along with the description.

S. No.	Types & Description
1	Abide Abide is used in the HTML5 form validation library with native API using the required attributes and patterns.
2	Equalizer Equalizer is a way to create multiple content with equal heights on your page.
3	Interchange It is used to load the responsive content according to the user's device.
4	Toggler Toggle is used to switch from one setting to another.
5	Sticky Sticky plugin is used to create a constant content or image in the website.

Foundation – Plugins Abide

Description

Abide is used in the HTML5 form validation library with native API using required attributes and patterns.

The following table describes the different types of abide along with the description.

S. No.	Types & Description
1	Abide Demo Abide is used in HTML5 form validation library.
2	SASS Reference Change the styles of the components by using SASS variables and mixins.
3	JavaScript Reference Foundation provides JavaScript components for abide.

Foundation – Plugins Abide Demo

Description

Abide is used in the HTML5 form validation library with native API using required attributes and patterns.

Example

The following example demonstrates the use of abide plugin in the Foundation:

```
<!DOCTYPE html>
<html>
<head>
    <title>Foundation Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.js">
    </script>
</head>
<body>
<h2>Example of Abide</h2>
<form data-abide novalidate>
    <div data-abide-error class="alert callout" style="display: none;">
        <p><i class="fi-alert"></i> There are some errors in your form.</p>
    </div>
    <div class="row">
        <div class="small-12 columns">
            <label>Name
                <input type="text" placeholder="Name" required pattern="[a-zA-Z]+>
                <span class="form-error">
                    Fill the Correct box
                </span>
            </label>
        </div>
    </div>
</form>

```

```

<label>Email
    <input type="text" placeholder="abd@email.com" pattern="email">
</label>
</div>
</div>
<div class="row">
    <div class="medium-4 columns">
        <fieldset>
            <label>Gender</label>
            <input type="radio" name="pockets" value="Male"><label>Male</label>
            <input type="radio" name="pockets" value="Female"><label>Female</label>
            <input type="radio" name="pockets" value="Other"><label>Other</label>
        </fieldset>
        <label>Input Label
            <select id="select" required>
                <option value=""></option>
                <option value="volvo">Volvo</option>
                <option value="saab">Saab</option>
                <option value="mercedes">Mercedes</option>
                <option value="audi">Audi</option>
            </select>
        </label>
    </div>
</div>
<div class="row">
    <fieldset class="large-6 columns">
        <button class="button" type="submit" value="Submit">Submit</button>
        <button class="button" type="reset" value="Reset">Reset</button>
    </fieldset>
</div>
</form>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>

```

```
</body>
</html>
```

Details of the code are as follows:

- The following code is written initially for creating the form. `<form data-abide novalidate>` is the element written initially.

```
<form data-abide novalidate>

<div data-abide-error class="alert callout" style="display: none;">
    <p><i class="fi-alert"></i> There are some errors in your form.</p>
</div>

<div class="row">
    <div class="small-12 columns">
        <label>Name
            <input type="text" placeholder="Name" required pattern="[a-zA-Z]+>
            <span class="form-error">
                Fill the Correct box
            </span>
        </label>
    </div>
</div>
</form>
```

- In the form we have included the error state to be displayed for particular fields while filling the form. The `data-abide-error class="alert callout"` element gives the error state for the complete form.
- The `form-error` class displays the error for the particular field when the entered input is wrong.

```
<form data-abide novalidate>

<div data-abide-error class="alert callout" style="display: none;">
    <p><i class="fi-alert"></i> There are some errors in your form.</p>
</div>

<div class="row">
    <div class="small-12 columns">
        <label>Name
            <input type="text" placeholder="Name" required pattern="[a-zA-Z]+>
            <span class="form-error">
                Fill the Correct box
            </span>
        </label>
    </div>
</div>
</form>
```

```

        </span>
    </label>
</div>
</div>
</form>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **abide_demo.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Abide

Name

Email

Gender

Male Female Other

Input Label

Submit **Reset**

Foundation – Plugins Abide SASS Reference

Variables

You can change the styles of the components by using the following SASS variables as listed in the table.

S. No.	Name & Description	Type	Default Value
1	\$abide-inputs Error styles included in the inputs.	Boolean	true

2	\$abide-labels Error styles included in the labels.	Boolean	true
3	\$input-background-invalid Sets background color of the invalid text inputs.	color	\$alert-color
4	\$form-label-color-invalid Set colors of the labels of invalid input.	color	\$alert-color
5	\$input-error-color Sets the default font color of the error text in the form.	color	\$alert-color
6	\$input-error-font-size Sets the default font size of the error text in the form.	Number	rem-calc(12)
7	\$input-error-font-weight Sets the default font weight of the error text in the form.	Keyword	\$global-weight-bold

Mixin

You can use the mixins to build the CSS class structure for your components as listed in the table.

form-input-error

For error state, the styles are included for the background and border of an input field.

```
@include form-input-error($background);
```

Parameter & Description	Type	Default Value
\$background Color set for the border and background.	Color	\$alert-color

form-error

Using values in the setting file, error styles are included to a form element.

```
@include form-error;
```

Foundation – Plugins Abide JavaScript Reference

Foundation provides JavaScript components for an abide plugin as listed below.

Foundation.Abide

It specifies the instance of abide as defined below-

```
var elem = new Foundation.Abide(element);
```

S. No.	Name & Description	Type
1	element It creates a jQuery object into abide.	jQuery
2	options Default plugin setting is override.	Object

Functions

Following are the functions used in abide.

.requiredCheck

It checks whether the form element has the required attribute or not.

S. No.	Name & Description	Type
1	element The jquery object is used to check the required attribute.	Object

.findLabel

Finds the label in the form element.

S. No.	Name & Description	Type
1	element The jquery object is used to check the required attribute.	Object

.addErrorClasses

CSS error class is included as mentioned by the abide settings to label, form and input.

S. No.	Name & Description	Type
1	element The jquery object is used to check the required attribute.	Object

.removeErrorClasses

CSS error class is removed as mentioned by the abide settings from the label, form and input.

S. No.	Name	Type
1	element The jquery object is used to check the required attribute.	Object

.validateInput

Validates the input through the form according to the type specified.

S. No.	Name & Description	Type
1	element The jquery object is used to check the required attribute.	Object
2	form Find various input element in the entire form.	Object

.validateForm

Helps in validating the form, if invalid inputs are found then it displays a form error element.

S. No.	Name & Description	Type
1	element The jquery object is used to check the required attribute.	Object

.validateText

Validates the text depending upon the patterns specified in the attribute.

S. No.	Name & Description	Type
1	element The jquery object is used to check the required attribute.	Object

.validateRadio

Validates the radio input depending upon whether it is required or not.

S. No.	Name & Description	Type
1	group The jquery object is used to check the required attribute.	String

.resetForm

The styles and inputs are reset.

S. No.	Name & Description	Type
1	\$form The jquery object is used to check the required attribute.	Object

Foundation – Plugins Equalizer

Description

Equalizer is a way to create multiple content of equal heights on your page.

The following table describes the different types of Equalizers along with the description.

S. No.	Types & Description
1	Basic Equalizer The content can be made equal by including the attribute <i>data-equalizer</i> .
2	Nesting We can nest the equalizer inside the other one by providing a unique id to the attribute <i>data-equalizer</i> .
3	JavaScript Reference Foundation provides JavaScript components for an equalizer

Foundation – Plugins Basic Equalizer

Description

The height of the content can be made equal to the other content's height by including the attribute *data-equalizer*. To make the height of the child element same as that of the height of the parent, use *data-equalizer-watch* attribute.

Example

The following example demonstrates the use of the equalizer plugin in Foundation:

```
<!DOCTYPE html>
<html>
<head>
  <title>Foundation Template</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.js">
  </script>
</head>
```

```

<body>
  <h2>Example of Equalizer</h2>
  <div class="row" data-equalizer>
    <div class="medium-4 columns">
      <div class="callout" data-equalizer-watch style="background-color:#C0B0F0;">
        <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p>
      </div>
    </div>
    <div class="medium-4 columns">
      <div class="callout" data-equalizer-watch style="background-color:#8BD6EE;">
        <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry.</p>
      </div>
    </div>
    <div class="medium-4 columns">
      <div class="callout" data-equalizer-watch style="background-color:#C0B0F0;">
        <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s</p>
      </div>
    </div>
  </div>
  <script>
    $(document).ready(function() {
      $(document).foundation();
    })
  </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **equalizer_basic.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Equalizer

Equalizer

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Equalizer

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

Equalizer

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s

Foundation – Plugins Nesting Equalizer

Description

We can nest an equalizer inside the other by providing a unique id to the attribute *data-equalizer*. The *data-equalizer-watch* value should match its parent value.

Example

The following example demonstrates the use of nesting equalizer in Foundation:

```
<!DOCTYPE html>
<html>
<head>
  <title>Foundation Template</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.cs
s">
```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.js">
</script>

</head>

<body>

<h2>Example of Equalizer</h2>

<div class="row" data-equalizer="first">
  <div class="medium-4 columns" data-equalizer-watch="first">
    <div class="callout" data-equalizer="second">
      <h3>Nested panel</h3>
      <div class="callout" data-equalizer-watch="second" style="background-color:#C0B0F0;">
        <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy.</p>
      </div>
      <div class="callout" data-equalizer-watch="second" style="background-color:#8BD6EE;">
        <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy.</p>
      </div>
      <div class="callout" data-equalizer-watch="second" style="background-color:#C0B0F0;">
        <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy.</p>
      </div>
    </div>
  </div>
</div>

<div class="medium-4 columns">
  <div class="callout panel" data-equalizer-watch="first" style="background-color:#808000;">
    <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s.</p>
  </div>
</div>

<div class="medium-4 columns">
  <div class="callout" data-equalizer-watch="first" style="background-color:#7B68EE;">

```

```
<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s.</p>
</div>
</div>
</div>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **equalizer_nesting.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Equalizer

Nested panel

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s.

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s.

Foundation – Equalizer JavaScript Reference

Foundation provides JavaScript components for an equalizer as listed below.

Foundation.Equalizer

It specifies the instance of a equalizer as defined below:

```
var elem = new Foundation.Equalizer(element);
```

S. No.	Name & Description	Type
1	element It creates a jQuery object into equalizer.	jQuery
2	options Default plugin setting is override.	Object

Plugin Options

You can use the following plugins to customize the equalizer instance. You can set the plugin options as individual data attributes.

S. No.	Name & Description	Example
1	equalizeOnStack The height equalization is enabled on small screen, when stacked.	true
2	throttleInterval The time is set in ms to debounce the equalization size. The transitions performance become smoother on mobile in lower times.	50

Events

The equalizer plugins attached to any element can trigger the following event.

S. No.	Name & Description
1	preEqualized.zf.equalizer It triggers an event before the heights are applied.
2	postEqualized.zf.equalizer It triggers an event when the heights have been applied.

Functions

The following are the functions used in accordion menu.

.getHeights

Gets the height of the children container within an equalizer parent. An array is returned.

S. No.	Name & Description	Type
1	\$eqParent It creates jQuery object in equalizer container.	Object

.applyHeight

Applies same height to all the children to match the tallest equalizer in the equalizer parent.

S. No.	Name & Description	Type
1	\$eqParent It creates jQuery object in equalizer container.	Object
2	heights Sets the array of height of children in the equalizer container.	array

.destroy

The equalizer is destroyed.

Foundation – Plugins Interchange

Description

It is used to load the responsive content according to the users device. Interchange example works on the older version of Foundation.

The following table describes the different types of interchange along with description.

S. No.	Types & Description
1	Use with Images Using interchange for images, you can set the image size as per the screen size.
2	Use with HTML It exchanges the entire block of HTML
3	Use with Background Images Passes the image path to set the background-image property of the matching rule.
4	Named Media Queries Interchange supports as shorthand for named queries for full CSS media queries.
5	JavaScript Reference Foundation provides JavaScript components for Interchange

Foundation – Interchange Use with Images

Description

Using interchange for images, you can set the image size as per the screen size i.e. large, medium or small. The image gets displayed smaller on the small screen device. To determine the users device for which image to serve, the CSS media queries are used.

Example

The following example demonstrates the use of interchange with images in the Foundation:

```
<!DOCTYPE html>
<html>
<head>
  <title>Foundation Template</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/5.5.2/css/foundation.css">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/jquery/2.0.3/jquery.min.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/5.5.2/js/foundation.min.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/modernizr/2.8.3/modernizr.js"></script>
</head>
<body>
  <h2>Example of use with images</h2>
  <img data-interchange="[/foundation/images/foundation-plugin/small.jpg,
  (default)], [/foundation/images/foundation-plugin/small.jpg, (small)],
  [/foundation/images/foundation-plugin/medium.jpg, (medium)]">
  <script>
    $(document).ready(function() {
      $(document).foundation();
    })
  </script>
</body>
</html>
```

Output

401

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **use_images.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of use with images

Foundation – Interchange Use with HTML

Description

Interchange can also be used for html files. It can exchange the entire block of HTML, it gets loaded in all types of screen size.

Example

The following example demonstrates the use of interchange with HTML in Foundation:

```
<!DOCTYPE html>
<html>
<head>
  <title>Foundation Template</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/5.5.2/css/foundation.css">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/jquery/2.0.3/jquery.min.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/5.5.2/js/foundation.min.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/modernizr/2.8.3/modernizr.js"></script>
</head>
<body>
  <h2>Example of use with HTML</h2>
```

```

<div data-interchange="[http://www.tutorialspoint.com, (small)],
[http://www.tutorialspoint.com, (medium)], [http://www.tutorialspoint.com,
(large)]"></div>

<script>
$(document).ready(function() {
  $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **use_html.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of use of html

Foundation – Interchange Use with Background Images

Description

To set the background image, use the *data-interchange* attribute inside the `<div>` tag. You can pass the image path to set the *background-image* property of the matching rule.

Example

The following example demonstrates the use of interchange with background image in Foundation:

```

<!DOCTYPE html>
<html>
<head>
  <title>Foundation Template</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">

```

```

<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/foundation/5.5.2/css/foundation.css">
<script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/2.0.3/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/5.5.2/js/foundation.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/modernizr/2.8.3/modernizr.js"></script>

</head>
<body>
<div data-interchange="[/foundation/images/foundation-plugin/smallback.jpg,
(small)], [/foundation/images/foundation-plugin/mediumback.jpg, (medium)],
[/foundation/images/foundation-plugin/background.jpg, (large)]">
    <h2>Welcome to TutorialsPoint</h2>
    <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p>
</div>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **background_image.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Welcome to TutorialsPoint

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Foundation – Interchange Named Media Queries

Description

In `$breakpoints` variable in SASS, you can define any breakpoint along with few keywords. It supports as a shorthand for the named queries for full CSS media queries.

You can include your own named media queries and can add them as properties to **Foundation.Interchange.SPECIAL_QUERIES**

```
Foundation.Interchange.SPECIAL_QUERIES['square'] = 'screen and (aspect-ratio: 1/1)';
```

S. No.	Query Name	Media Query
1	small	screen and (min-width: 0em)
2	medium	only screen and (min-width: 40em)
3	large	only screen and (min-width: 64em)
4	xlarge	only screen and (min-width: 75em)
5	xxlarge	only screen and (min-width: 90em)
6	portrait	screen and (orientation: portrait)
7	landscape	screen and (orientation: landscape)
8	retina	only screen and (-webkit-min-device-pixel-ratio: 2), only screen and (-moz-device-pixel-ratio: 2), only screen and (-o-min-device-pixel-ratio: 2/1), only screen and (min-device-pixel-ratio: 2), only screen and (min-resolution: 192dpi), only screen and (min-resolution: 2dppx)

Foundation – Interchange Javascript Reference

Foundation provides JavaScript components for Interchange as listed below.

Initializing

You can initialize the interchange in JavaScript by using the *foundation.interchange.js* and *foundation.core.js* plugins. The plugin requires the following libraries:

- **foundation.util.mediaQuery.js**
- **foundation.util.timerAndImageLoader.js**

Foundation.Interchange

It specifies instance of an Interchange as defined below:

```
var elem = new Foundation.Interchange(element);
```

S. No.	Name & Description	Type
1	element It creates a jQuery object into a Interchange.	jQuery
2	options Default plugin setting is override.	Object

Plugin Options

You can use the following plugin to customize the Interchange instance. You can set the plugin option as individual data attributes.

S. No.	Name & Description
1	rules <i>data-interchange</i> array notation is set. It is applied to interchange elements.

Functions

Following are the functions used in interchange.

.replace

Replaces the image or HTML *src* property to the specified path.

S. No.	Name & Description	Type
1	path Sets Image or HTML path.	string

.destroy

The instance of interchange is destroyed.

Foundation – Plugins Toggler

Description

Toggle is used to switch from one setting to another. It is easy to toggle the animate element with a click.

The following table describes the different types of toggler along with the description.

S. No.	Types & Description
1	Toggle a CSS class Includes the <i>data-toggler</i> attribute to the element that you want to toggle.
2	Toggle with Animation The animation effect can be applied to toggle by using the attribute <i>data-toggler data-animate</i> in the element.
3	Mark as Closable Include <i>data-closable</i> attributes to close the element.
4	Javascript Reference Foundation provides JavaScript components for toggler.

Foundation – Plugins Toggle a CSS class

Description

Includes the *data-toggler* attribute to the element that you want to toggle. Also include a unique id to the element to target.

Example

The following example demonstrates the use of toggler plugin in Foundation:

```
<!DOCTYPE html>
<html>
<head>
  <title>Foundation Template</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.js">
</script>
</head>
<body>
<h2>Example of Toggle</h2>
<p><a data-toggle="toggle">Toggler</a></p>
<ul class="menu" id="toggle" data-toggler=".expanded">
<li><a href="#">Home</a></li>
<li><a href="#">About</a></li>
<li><a href="#">Service</a></li>
<li><a href="#">Contact</a></li>
</ul>
<script>
$(document).ready(function() {
  $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **toggle_css.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Toggle

Toggler

[Home](#) [About](#) [Service](#) [Contact](#)

Foundation – Plugins Toggle with Animation

Description

The animation effect can be applied to toggle by using the attribute data-toggler data-animate in the element. Using the Motion UI class, the elements come in and out of the view.

Example

The following example demonstrates the use of the toggler plugin in Foundation:

```
<!doctype html>
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Reveal Animations</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/motion-
    ui/1.1.1/motion-ui.css">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
    in.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
    js"></script>
</head>
<body>
    <h2>Example of Toggle with animation</h2>
    <button class="button" data-toggle="toggle">Toggle Panel</button>
    <div class="callout" id="toggle" data-toggler data-animate="hinge-in-from-top
    slide-out-right" style="background-color:#8BD6EE;">
        <h4>Welcome to TutorialsPoint</h4>
        <p>Lorem Ipsum is simply dummy text of the printing and typesetting
        industry. Lorem Ipsum has been the industry's standard dummy text.</p>
    </div>
    <script>
        $(document).ready(function() {
            $(document).foundation();
        })
    </script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **toggle_animation.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Example of Toggle with animation

Toggle Panel

Welcome to TutorialsPoint

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text.

Foundation – Plugins Mark as Closable

Description

Includes the *data-closable* attribute to close the element; animation effect can also be applied when the element is exiting. The *data-close* attribute is included when you want to add *click trigger* inside the element.

Example

The following example demonstrates the use of toggler plugin in Foundation:

```
<!doctype html>
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Foundation Template</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/motion-
    ui/1.1.1/motion-ui.css">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
    in.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
    js"></script>
</head>
```

```
<body>

<h2>Example of Toggle with closable</h2>

<div class="callout" data-closable="spin-out" style="background-color:#8BD6EE;">
    <button class="close-button" data-close>x</button>
    <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text.</p>
</div>

<script>
    $(document).ready(function() {
        $(document).foundation();
    })
</script>
</body>
</html>
```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **closable.html** file.
 - Open this HTML file in a browser, an output is displayed as shown below.

Example of Toggle with closable

Text Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text.

Foundation – Toggler Javascript Reference

Foundation provides JavaScript components for toggler as listed below.

Initializing

You can initialize toggle in JavaScript by using the `foundation.toggler.js` and `foundation.core.js` plugins. The plugin requires the following libraries:

- **foundation.util.motion.js**

Foundation.Toggler

It specifies instance of a Toggler as defined below:

```
var elem = new Foundation.Toggler(element);
```

S. No.	Name & Description	Type
1	element It creates a jQuery object into a Toggler.	jQuery
2	options Default plugin setting is override.	Object

Plugin Options

You can use the following plugin to customize the Toggler instance. You can set the plugin option as individual data attributes.

S. No.	Name & Description	Example
1	animate Tells the plugin to animate the element when toggled.	false

Events

The toggler plugin attached to any element can trigger the following event.

S. No.	Name & Description
1	on.zf.toggler It triggers an event when the target element consists of a class after toggle.
2	off.zf.toggler It triggers an event when the target element does not have a class after toggle.

Functions

Following are the functions used in toggler.

.toggle

Depending upon the state- on or off, the event is fired from the original trigger. It toggles the target class onto the target element.

.destroy

The instance of toggler is destroyed.

Foundation – Plugins Sticky

Description

Sticky plugin is used to create a constant content or image in the website.

The following table describes the different types of sticky plugins along with the description.

S. No.	Types & Description
1	Basics To create a sticky type container, include class <code>.sticky</code> and <code>[data-sticky]</code> to an element.
2	Advanced <code>data-top-anchor="idOfSomething"</code> is used to set the image or content, which is sticky to the particular height.
3	JavaScript Reference Foundation provides JavaScript components for Sticky plugins

Foundation – Plugins Sticky Basic

Description

To create a sticky type container include class `.sticky` and `[data-sticky]` to an element. In a container, the sticky element should be wrapped as it determines the grid layout and sizing.

Example

The following example demonstrates the use of toggler plugin in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Sticky Basics</title>
  <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
  <h2>Sticky Basics Example</h2>
```

```
<div class="row">
  <div class="columns small-12">
    <div class="columns small-6" id="sticky_example">
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
      <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.</p>
      <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.</p>
      <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.</p>
    </div>
  </div>
</div>
```

<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.</p>

```

<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing</p>

</div>
<div class="columns small-6 right" data-sticky-container>
  <div class="sticky" data-sticky data-anchor="sticky_example">
    
  </div>
</div>
</div>
</div>

<div>
  <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.</p>
  <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.</p>
</div>
<script>
$(document).ready(function() {
  $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **sticky_basics.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.



Foundation – Plugins Sticky Advanced

Description

`data-top-anchor="idOfSomething", data-btm-anchor="idOfSomething:[top/bottom]",` or a set pixel number `data-top-anchor="150"` is used to set the image or sticky content to a particular height.

Example

The following example demonstrates the use of toggler plugin in Foundation:

```
<!doctype html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Reveal Animations</title>
  <link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/motion-
ui/1.1.1/motion-ui.css">
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.m
in.js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.
js"></script>
  <script
  src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.js">
</script>
</head>
<body>
<h2>Advanced Sticky</h2>
<div class="columns small-6" id="advanced">
  <p>Lorem Ipsum is simply dummy text of the printing and typesetting
industry. Lorem Ipsum has been the industry's</p>
  <p>Lorem Ipsum is simply dummy text of the printing and typesetting
industry. Lorem Ipsum has been the industry's</p>
  <p>Lorem Ipsum is simply dummy text of the printing and typesetting
industry. Lorem Ipsum has been the industry's</p>
  <p>Lorem Ipsum is simply dummy text of the printing and typesetting
industry. Lorem Ipsum has been the industry's</p>
  <p>Lorem Ipsum is simply dummy text of the printing and typesetting
industry. Lorem Ipsum has been the industry's</p>
  <p>Lorem Ipsum is simply dummy text of the printing and typesetting
industry. Lorem Ipsum has been the industry's</p>
  <p>Lorem Ipsum is simply dummy text of the printing and typesetting
industry. Lorem Ipsum has been the industry's</p>
```

```

<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's</p>

<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's</p>

<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's</p>

<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's</p>

<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's</p>

<p id="foo">Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's</p>

<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's</p>

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>

</div>

<div class="columns small-6 right sticky-container" data-sticky-container>
    <div class="sticky" data-sticky data-top-anchor="advanced:top" data-btm-anchor="foo:bottom">
        
    </div>
</div>
<script>
$(document).ready(function() {
    $(document).foundation();
})
</script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **sticky_advanced.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Advanced Sticky

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's



Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Foundation – Plugins Sticky JavaScript Reference

Foundation provides JavaScript components for Sticky plugins as listed below.

Initializing

You can initialize sticky in JavaScript by using the `foundation.sticky.js` and `foundation.core.js` plugins. The plugin requires the following libraries:

- **`foundation.util.triggers.js`**
- **`foundation.util.mediaQuery.js`**

Foundation.Sticky

It specifies instance of sticky as defined below:

```
var elem = new Foundation.Sticky(element);
```

S. No.	Name & Description	Type
1	element It creates a jQuery object into a sticky.	jQuery
2	options Default plugin setting is override.	Object

Plugin Options

You can use the following plugins to customize the sticky instance. You can set the plugin option as individual data attributes.

S. No.	Name & Description	Example
1	container For styling and sizing, own classes can be included.	' '
2	stickTo Sets the location, where the element has to stick.	'top'
3	anchor Includes the id of that element, which is anchored to the single element.	'exampleId'
4	topAnchor Considers the top anchor id if more elements are used as anchor points.	'exampleId:top'
5	btmAnchor Considers the bottom anchor id if more elements are used as anchor points.	'exampleId:bottom'

6	marginTop Sets the margin in em for the top element when it becomes sticky.	1
7	marginBottom Sets the margin in em for the bottom element when it becomes sticky.	1
8	stickyOn Breakpoint string should become sticky.	'medium'
9	stickyClass Applies class to the sticky element and removes on destruction.	'sticky'
11	containerClass Applies class to the sticky container. By default, it is <i>sticky-container</i> .	'sticky-container'
12	checkEvery Sticky points are recalculated by the number of scroll events between the plugins.	50

Events

The sticky plugins attached to any element can trigger the following events.

S. No.	Name & Description
1	stuckto.zf.sticky It triggers an event when \$element has become <i>position: fixed;</i> and Namespaced to <i>top</i> or <i>bottom</i> .
2	unstuckfrom.zf.sticky It triggers an event when \$element has become anchored and Namespaced to <i>top</i> or <i>bottom</i> .

Functions

Following are the functions used in sticky.

.pauseListeners

For scrolling the event, handler is removed and event is changed into anchor.

S. No.	Name & Description	Type
1	scrollListener Window is attached with unique, namespaced scroll listener.	string

.calc

On every *scroll* event, the calc is called and depending upon the cached and booleans value the *_init* fires the functions.

S. No.	Name & Description	Type
1	checkSizes The plugin recalculates the size and breakpoints if it sets to true.	Boolean
2	scroll Scrolls the current position of the event, which is passed from the cb function. Otherwise, by default, it sets to <code>window.pageYOffset</code>	Number

.destroy

The current sticky element is destroyed; the element is reset to the top position. It removes the JS, which includes classes and css properties and `$element` is unwrapped when JS includes `$container`.

.emCalc

Helper function is used to calculate the em values.

S. No.	Name & Description	Type
1	Number Number of ems are calculated in the pixels.	em

Foundation SASS

21. Foundation – SASS Functions

Foundation provides a set of SASS utility functions, which can be used with *util*, *color*, *selector*, *unit*, *value* and many more.

You can import all utility files at a time by using the following line of code:

```
@import 'util/util';
```

You can also import individual utility files as shown below:

```
@import 'util/color';
@import 'util/selector';
@import 'util/unit';
@import 'util/value';
```

SASS Reference

You can change the styles of the components by using the following SASS functions.

foreground

It provides the foreground color to the elements based on the background color. It uses the following format for assigning different types of parameters:

```
foreground($color, $yes, $no, $threshold)
```

The above parameters are specified in the following table:

S. No.	Parameter & Description	Type	Default Value
1	\$color It checks the lightness of the color.	Color	None
2	\$yes If the color is light, then it returns \$yes color.	Color	\$black
3	\$no If the color is dark, then it returns \$no color.	Color	\$white
4	\$threshold It represents the threshold of the lightness.	Percentage	60%

smart-scale

It provides appropriate color for the elements according to its lightness. It uses the following format for specifying the appropriate color:

```
smart-scale($color, $scale, $threshold)
```

The above given parameters are specified in the following table:

S. No.	Parameter & Description	Type	Default Value
1	\$color It is used to scale the color.	Color	None
2	\$scale It specifies the percentage to scale up or down.	Percentage	5%
3	\$threshold It represents the threshold of the lightness.	Percentage	40%

text-inputs

It creates a selector while using the text input type. It uses the following format for specifying the input types:

```
text-inputs($types)
```

It uses the parameter as specified in the following table:

S. No.	Parameter & Description	Type	Default Value
1	\$types It provides a number of text input types for generating a selector.	Color	-

strip-unit

It removes the unit from the value and returns only the number. It uses the following format for removing the unit from value:

```
strip-unit($num)
```

It uses the parameter as specified in the following table:

S. No.	Parameter & Description	Type	Default Value
1	\$num It specifies the number when you remove the unit from the value.	Color	None

rem-calc

It changes the pixel value to match the rem values. It uses the following format for converting pixel values to rem values:

```
rem-calc($values, $base)
```

It uses the following parameters as specified in the table:

S. No.	Parameter & Description	Type	Default Value
1	\$values It converts the pixel values to rem values and separate them using spaces. If you are converting comma separated list, then wrap list in parentheses.	Number or List	None
2	\$base It provides the base value while converting pixel to rem value. If there is null value for the base, then function uses the <code>\$base-font-size</code> variable as the base.	Number	null

has-value

It specifies the value if it is not false. The false values include null, none, 0 or an empty list. It uses the following format for specifying the value:

```
has-value($val)
```

It uses the parameter as specified in the following table:

S. No.	Parameter & Description	Type	Default Value
1	\$val It checks the specified value.	Mixed	None

get-side

It specifies the side of a value and defines the top/right/bottom/left values on padding, margin etc. It uses the following format for specifying the side of a value:

```
has-value($val)
```

It uses the following parameters as specified in the table:

S. No.	Parameter & Description	Type	Default Value
1	\$val It specifies the side of a value.	List or Number	None

2	\$side It determines on which side the (top/right/bottom/left) value should return.	Keyword	None
---	---	---------	------

get-border-value

It determines the border value of an element. It uses the following format for specifying the border value:

```
get-border-value($val, $elem)
```

It uses the following parameters as specified in the table:

S. No.	Parameter & Description	Type	Default Value
1	\$val It finds a specific value of the border.	List	None
2	\$elem It is used to extract the border component.	Keyword	None

22. Foundation – SASS Mixins

Importing

It imports the contents of the SASS mixins, which are placed under the `scss/util/_mixins.scss` file. You can import the SASS mixins by using the following line of code:

```
@import 'util/mixins';
```

SASS Reference

You can change the styles of the components by using the SASS functions.

Mixins

You can use the following mixins to build the CSS class structure for your components.

CSS-TRIANGLE

It is used for creating dropdown arrows, dropdown pips and many more. It uses `&::before` or `&::after` selector for attaching a triangle to an existing element. It uses the following format:

```
@include css-triangle($triangle-size, $triangle-color, $triangle-direction);
```

It uses the following parameters as specified in the table:

S. No.	Parameter & Description	Type	Default Value
1	\$triangle-size It defines the width of the triangle.	Number	None
2	\$triangle-color It defines the color of the triangle.	Color	None
3	\$triangle-direction It defines the direction of the triangle such as up, right, down or left.	Keyword	None

HAMBURGER

It is used for creating menu icon with width, height, number of bars and colors. It uses the following format:

```
@include hamburger($color, $color-hover, $width, $height, $weight, $bars);
```

It uses the following parameters as specified in the table:

S. No.	Parameter & Description	Type	Default Value
1	\$color It defines the color for the icon.	Color	None
2	\$color-hover It defines the color when you hover on the icon.	Color	None
3	\$width It defines the width of the icon.	Number	None
4	\$height It defines the height of the icon.	Number	None
5	\$weight It defines the weight of individual bars in the icon.	Number	None
6	\$bars It defines the number of bars in the icon.	Number	None

BACKGROUND-TRIANGLE

It is used for specifying the background image to an element. It uses the following format:

```
@include background-triangle($color);
```

It uses the parameter as specified in the table:

S. No.	Parameter & Description	Type	Default Value
1	\$color It defines the color for the triangle.	Color	\$black

CLEARFIX

This mixin automatically clears the children elements, so that there is no need of additional markup. It uses the following format:

```
@include clearfix;
```

AUTO-WIDTH

It automatically sizes the elements based on the number of elements present in the container. It uses the following format:

```
@include auto-width($max, $elem);
```

It uses the following parameters as specified in the table:

S. No.	Parameter & Description	Type	Default Value
1	\$max It identifies the maximum number of items in the container.	Number	None
2	\$elem It uses a tag for sibling selectors.	Keyword	li

DISABLE-MOUSE-OUTLINE

It is used to disable the outline around the element, when it identifies the mouse input action. It uses the following format:

```
@include disable-mouse-outline;
```

ELEMENT-INVISIBLE

It is used to hide the elements and can be available to keyboards and other devices. It uses the following format:

```
@include element-invisible;
```

ELEMENT-INVISIBLE-OFF

It is used to remove the invisible elements and reverses the CSS output by using the *element-invisible()* mixin. It uses the following format:

```
@include element-invisible-off;
```

VERTICAL-CENTER

It is used to place the elements vertically-centered inside the non-static parent element by using the following format:

```
@include vertical-center;
```

HORIZONTAL-CENTER

It is used to place the elements horizontally-centered inside the non-static parent element by using the below format:

```
@include horizontal-center;
```

ABSOLUTE-CENTER

It is used to place the elements absolutely-centered inside the non-static parent element by using the following format:

```
@include absolute-center;
```

Foundation Libraries

23. Foundation – Motion UI

Foundation provides Motion UI library for creating UI transitions and animations and is used by Foundation components such as *Toggler*, *Reveal* and *Orbit*.

Installing Motion UI

You can install Motion UI library in your project by using npm or bower as shown in the following line of code:

```
$ npm install motion-ui --save-dev  
bower install motion-ui --save-dev
```

You can add a path for Motion UI library in the *Compass* by using *config.rb* as shown in the below line of code:

```
add_import_path 'node_modules/motion-ui/src'
```

You can include the path in the *gulp-sass* using the following lines of code:

```
gulp.src('./src/scss/app.scss')  
.pipe(sass({  
  includePaths: ['node_modules/motion-ui/src']  
}));
```

Import the Motion UI library in the SASS file using the following code:

```
@import 'motion-ui'
```

Built-in Transitions

Foundation provides transition effects by using transition classes, which are created by Motion UI library. Let us create one simple example using transition effects.

Foundation – Motion UI Transitions

Description

Foundation provides transition effects by using transition classes, which are created by Motion UI library.

Example

The following example demonstrates the use of *transitions* in Foundation:

```

<!doctype html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Motion UI Transition</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/motion-ui/1.1.1/motion-ui.css">
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
    <h2>Motion UI Transition Example</h2>
    <p><a data-toggle="toggle_panel">Click here to toggle the panel</a></p>
    <div class="callout" id="toggle_panel" data-toggler data-animate="hinge-in-from-bottom spin-out">
        <h2>Foundation</h2>
        <p>Foundation is semantic, readable, flexible, and completely customizable.</p>
    </div>
    <script>
        $(document).ready(function() {
            $(document).foundation();
        })
    </script>
</body>
</html>

```

Output

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **foundation_transitions.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

Motion UI Transition Example

[Click here to toggle the panel](#)

Foundation

Foundation is semantic, readable, flexible, and completely customizable.

Custom Transitions

You can set the custom transition classes using Motion UI library. For instance, we will set custom classes for *mui-hinge()* transition, which rotates the element:

```
@include mui-hinge(
  $state: in,
  $from: right,
  $turn-origin: from-back,
  $duration: 0.5s,
  $timing: easeInOut
);
```

Animation

You can use Motion UI transition effects for creating CSS animations. Click this [link](#) to check how animation works on the modal using *data-animation* class.

Foundation – Reveal Animations

Description

It creates the animations on the modal.

Example

The following example demonstrates the use of *animations* in Foundation:

```
<!doctype html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Reveal Animations</title>
```

```

<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/css/foundation.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/motion-ui/1.1.1/motion-ui.css">
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/vendor/jquery.min.js"></script>
<script
      src="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.0.1/js/foundation.min.js"></script>
</head>
<body>
<h2>Reveal Animations Example</h2>
<p><a data-toggle="reveal_modal">Click here to open modal</a></p>
<div class="reveal" id="reveal_modal" data-reveal data-animation-in="spin-in"
      data-animation-out="spin-out">
    <h2>This is Animations Modal</h2>
    <p>Foundation is semantic, readable, flexible, and completely
      customizable.</p>
    <button class="close-button" data-close aria-label="Close reveal"
           type="button">
      <span aria-hidden="true">x</span>
    </button>
  </div>
<script>
  $(document).ready(function() {
    $(document).foundation();
  })
</script>
</body>
</html>

```

Output

437

Let us carry out the following steps to see how the above given code works:

- Save the above given html code in **reveal_animations.html** file.
- Open this HTML file in a browser, an output is displayed as shown below.

