**MAIN PYTHON CODE**

from \_\_future\_\_ import unicode\_literals

from flask import Flask,render\_template,url\_for,request

from spacy\_summarization import text\_summarizer

import time

import spacy

nlp = spacy.load('en\_core\_web\_sm')

app = Flask(\_\_name\_\_)

# Web Scraping Pkg

from bs4 import BeautifulSoup

# from urllib.request import urlopen

from urllib.request import urlopen

# Reading Time

def readingTime(mytext):

total\_words = len([ token.text for token in nlp(mytext)])

estimatedTime = total\_words/200.0

return estimatedTime

# Fetch Text From Url

def get\_text(url):

page = urlopen(url)

soup = BeautifulSoup(page)

fetched\_text = ' '.join(map(lambda p:p.text,soup.find\_all('p')))

return fetched\_text

@app.route('/')

def index():

return render\_template('index.html')

@app.route('/analyze',methods=['GET','POST'])

def analyze():

start = time.time()

if request.method == 'POST':

rawtext = request.form['rawtext']

final\_reading\_time = readingTime(rawtext)

final\_summary = text\_summarizer(rawtext)

summary\_reading\_time = readingTime(final\_summary)

end = time.time()

final\_time = end-start

return render\_template('index.html',ctext=rawtext,final\_summary=final\_summary,final\_time=final\_time,final\_reading\_time=final\_reading\_time,summary\_reading\_time=summary\_reading\_time)

@app.route('/analyze\_url',methods=['GET','POST'])

def analyze\_url():

start = time.time()

if request.method == 'POST':

raw\_url = request.form['raw\_url']

rawtext = get\_text(raw\_url)

final\_reading\_time = readingTime(rawtext)

final\_summary = text\_summarizer(rawtext)

summary\_reading\_time = readingTime(final\_summary)

end = time.time()

final\_time = end-start

return render\_template('index.html',ctext=rawtext,final\_summary=final\_summary,final\_time=final\_time,final\_reading\_time=final\_reading\_time,summary\_reading\_time=summary\_reading\_time)

@app.route('/compare\_summary')

def compare\_summary():

return render\_template('compare\_summary.html')

@app.route('/comparer',methods=['GET','POST'])

def comparer():

start = time.time()

if request.method == 'POST':

rawtext = request.form['rawtext']

final\_reading\_time = readingTime(rawtext)

final\_summary\_spacy = text\_summarizer(rawtext)

summary\_reading\_time = readingTime(final\_summary\_spacy)

# Gensim Summarizer

final\_summary\_gensim = summarize(rawtext)

summary\_reading\_time\_gensim = readingTime(final\_summary\_gensim)

# NLTK

final\_summary\_nltk = nltk\_summarizer(rawtext)

summary\_reading\_time\_nltk = readingTime(final\_summary\_nltk)

# Sumy

final\_summary\_sumy = sumy\_summary(rawtext)

summary\_reading\_time\_sumy = readingTime(final\_summary\_sumy)

end = time.time()

final\_time = end-start

return render\_template('compare\_summary.html',ctext=rawtext,final\_summary\_spacy=final\_summary\_spacy,final\_summary\_gensim=final\_summary\_gensim,final\_summary\_nltk=final\_summary\_nltk,final\_time=final\_time,final\_reading\_time=final\_reading\_time,summary\_reading\_time=summary\_reading\_time,summary\_reading\_time\_gensim=summary\_reading\_time\_gensim,final\_summary\_sumy=final\_summary\_sumy,summary\_reading\_time\_sumy=summary\_reading\_time\_sumy,summary\_reading\_time\_nltk=summary\_reading\_time\_nltk)

@app.route('/about')

def about():

return render\_template('index.html')

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**BACKGROUND PYTHON CODE**

**Spacy Summarization**

# NLP Pkgs

import spacy

nlp = spacy.load('en\_core\_web\_sm')

# Pkgs for Normalizing Text

from spacy.lang.en.stop\_words import STOP\_WORDS

from string import punctuation

# Import Heapq for Finding the Top N Sentences

from heapq import nlargest

def text\_summarizer(raw\_docx):

raw\_text = raw\_docx

docx = nlp(raw\_text)

stopwords = list(STOP\_WORDS)

# Build Word Frequency # word.text is tokenization in spacy

word\_frequencies = {}

for word in docx:

if word.text not in stopwords:

if word.text not in word\_frequencies.keys():

word\_frequencies[word.text] = 1

else:

word\_frequencies[word.text] += 1

maximum\_frequncy = max(word\_frequencies.values())

for word in word\_frequencies.keys():

word\_frequencies[word] = (word\_frequencies[word]/maximum\_frequncy)

# Sentence Tokens

sentence\_list = [ sentence for sentence in docx.sents ]

# Sentence Scores

sentence\_scores = {}

for sent in sentence\_list:

for word in sent:

if word.text.lower() in word\_frequencies.keys():

if len(sent.text.split(' ')) < 30:

if sent not in sentence\_scores.keys():

sentence\_scores[sent] = word\_frequencies[word.text.lower()]

else:

sentence\_scores[sent] += word\_frequencies[word.text.lower()]

summarized\_sentences = nlargest(7, sentence\_scores, key=sentence\_scores.get)

final\_sentences = [ w.text for w in summarized\_sentences ]

summary = ' '.join(final\_sentences)

return summary

**FRONT- END CODE**

**Index.html**  
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>  
  <meta name="viewport" content="width=device-width, initial-scale=1"/>  
  <title>Summaryzer</title>  
  
  <!-- CSS  -->  
  <link href="<https://fonts.googleapis.com/icon?family=Material+Icons>" rel="stylesheet">  
  <link href="static/css/materialize.css" type="text/css" rel="stylesheet" media="screen,projection"/>  
  <link href="static/css/style.css" type="text/css" rel="stylesheet" media="screen,projection"/>  
   <link rel="stylesheet" href="<https://use.fontawesome.com/releases/v5.5.0/css/all.css>" integrity="sha384-B4dIYHKNBt8Bc12p+WXckhzcICo0wtJAoU8YZTY5qE0Id1GSseTk6S+L3BlXeVIU" crossorigin="anonymous">  
</head>  
</head>  
<body><div class="navbar-fixed">  
   <nav class="grey darken-4" role="navigation">  
    <div class="nav-wrapper container">  
      <a id="logo-container" href="#" class="brand-logo">Summarizer</a>  
      <ul class="right hide-on-med-and-down">  
          </ul>  
  
       
      <a href="#" data-target="nav-mobile" class="sidenav-trigger"><i class="material-icons">menu</i></a>  
    </div>  
  </nav>  
</div>  
<ul id="nav-mobile" class="sidenav">  
      </ul>  
<!--Slider -->  
 <section class="slider">  
  <div id="index-banner" class="parallax-container">  
     <ul class="slides">  
      <li>  
        <img src="backgroundblue.png" alt="Unsplashed background img 1" alt="">  
        <div class="caption center-align">  
          <h2>Summarization Simplified</h2>  
          <h5 class="light grey-text text-lighten-3 hide-on-small-only">  
           Using SpaCy  
          </h5>  
           
           <div class="row center">  
          <a href="{{url\_for('index')}}" id="download-button" class="btn-large waves-effect waves-light blue lighten-1">Refresh</a>  
        </div>  
        </div>  
      </li>  
      <li>  
       <img src="backgroundpurple.png" alt="Unsplashed background img 1">  
        <div class="caption left-align">  
          <h2>Text Summarizer with SpaCy</h2>  
          <h5 class="light grey-text text-lighten-3 hide-on-small-only">  
           NLP based App  
          </h5>  
          <a href="#aboutapp" class="btn btn-large waves-effect waves-light light-blue lighten-1">Learn More</a>  
        </div>  
      </li>  
     
    </ul>  
  </div>  
</section>  
  
<!-- Start of Main Section -->  
 <div class="container">  
    <div class="section">  
         
      <!--   Icon Section   -->  
      <div class="row">  
        <div class="input-field col s12 m10">  
          <div class="icon-block">  
            <h2 class="center brown-text"><i class="material-icons">chrome\_reader\_mode</i></h2>  
           <form method="POST" action="/analyze">  
         <textarea name="rawtext" cols="3" rows="5" class="form-control" required="true" placeholder="Enter Text Here"></textarea>  
           
         <br/>  
         <button class="btn btn-small waves-effect waves-light light-blue lighten-1" type="reset">Clear</button>  
          <button class="btn btn-small waves-effect waves-light purple lighten-1" type="submit">Summarize</button>  
           
        </form>  
          </div>  
        </div>  
  
      </div>  
  
        <div class="row">  
        <div class="input-field col s12 m10">  
          <div class="icon-block">  
            <h2 class="center brown-text"><i class="material-icons">chrome\_reader\_mode</i></h2>  
           <form method="POST" action="/analyze\_url">  
         <input type="text" name="raw\_url" placeholder="Enter URL Here" required="true">  
         <button class="btn btn-small waves-effect waves-light light-blue lighten-1" type="reset">Clear</button>  
          <button class="btn btn-small waves-effect waves-light purple lighten-1" type="submit">Summarize</button>  
           
        </form>  
          </div>  
        </div>  
  
      </div>  
  
    </div>  
  </div>  
  <!-- End -->  
  
  <!-- Result Display-->  
  <section class="section section-solutions-about grey darken-2">  
 <div class="container white-text">  
   
      <!--   Icon Section   -->  
      <div class="row">  
        <div class="col s12 m6">  
          <div class="icon-block">  
            <h2 class="center brown-text"><i class="material-icons">group</i></h2>  
            <h5 class="center">Your Text</h5>  
             <p>Reading Time: <span style="color:#0091EA;">{{ final\_reading\_time }} min </span></p>  
            <p class="light">{{ctext}}</p>  
            <div class="alert alert-info" role="alert"><p>Time Elapsed: <span style="color:#0091EA;">{{ final\_time }} mins </span></p><br/>  
               
        </div>  
          </div>  
        </div>  
  
        <div class="col s12 m6 grey darken-3">  
          <div class="icon-block">  
            <h2 class="center brown-text"><i class="material-icons">chrome\_reader\_mode</i></h2>  
            <h5 class="center">Summarized</h5>  
              <div class="" role="alert">  
              <p>Reading Time: <span style="color:#0091EA;">{{ summary\_reading\_time }} min </span></p>  
        </div>  
            <p class="light">{{ final\_summary }}</p>  
           
          </div>  
        </div>  
  
      </div>  
  
   
  </div>  
</section>  
  
  </body>  
</html>

**CSS File**

**Custom.css**

nav ul a,

nav .brand-logo {

color: #444;

}

p {

line-height: 2rem;

}

.sidenav-trigger {

color: #26a69a;

}

.parallax-container {

min-height: 380px;

line-height: 0;

height: auto;

color: rgba(255,255,255,.9);

}

.parallax-container .section {

width: 100%;

}

@media only screen and (max-width : 992px) {

.parallax-container .section {

position: absolute;

top: 40%;

}

#index-banner .section {

top: 10%;

}

}

@media only screen and (max-width : 600px) {

#index-banner .section {

top: 0;

}

}

.icon-block {

padding: 0 15px;

}

.icon-block .material-icons {

font-size: inherit;

}

footer.page-footer {

margin: 0;

}

**Materialize.css**

.materialize-red {

background-color: #e51c23 !important;

}

.materialize-red-text {

color: #e51c23 !important;

}

.materialize-red.lighten-5 {

background-color: #fdeaeb !important;

}

.materialize-red-text.text-lighten-5 {

color: #fdeaeb !important;

}

.materialize-red.lighten-4 {

background-color: #f8c1c3 !important;

}

.materialize-red-text.text-lighten-4 {

color: #f8c1c3 !important;

}

.materialize-red.lighten-3 {

background-color: #f3989b !important;

}

.materialize-red-text.text-lighten-3 {

color: #f3989b !important;

}

.materialize-red.lighten-2 {

background-color: #ee6e73 !important;

}

.materialize-red-text.text-lighten-2 {

color: #ee6e73 !important;

}

.materialize-red.lighten-1 {

background-color: #ea454b !important;

}

.materialize-red-text.text-lighten-1 {

color: #ea454b !important;

}

.materialize-red.darken-1 {

background-color: #d0181e !important;

}

.materialize-red-text.text-darken-1 {

color: #d0181e !important;

}

.materialize-red.darken-2 {

background-color: #b9151b !important;

}

.materialize-red-text.text-darken-2 {

color: #b9151b !important;

}

.materialize-red.darken-3 {

background-color: #a21318 !important;

}

.materialize-red-text.text-darken-3 {

color: #a21318 !important;

}

.materialize-red.darken-4 {

background-color: #8b1014 !important;

}

.materialize-red-text.text-darken-4 {

color: #8b1014 !important;

}

.red {

background-color: #F44336 !important;

}

.red-text {

color: #F44336 !important;

}

.red.lighten-5 {

background-color: #FFEBEE !important;

}

.red-text.text-lighten-5 {

color: #FFEBEE !important;

}

.red.lighten-4 {

background-color: #FFCDD2 !important;

}

.red-text.text-lighten-4 {

color: #FFCDD2 !important;

}

.red.lighten-3 {

background-color: #EF9A9A !important;

}

.red-text.text-lighten-3 {

color: #EF9A9A !important;

}

.red.lighten-2 {

background-color: #E57373 !important;

}

.red-text.text-lighten-2 {

color: #E57373 !important;

}

.red.lighten-1 {

background-color: #EF5350 !important;

}

.red-text.text-lighten-1 {

color: #EF5350 !important;

}

.red.darken-1 {

background-color: #E53935 !important;

}

.red-text.text-darken-1 {

color: #E53935 !important;

}

.red.darken-2 {

background-color: #D32F2F !important;

}

.red-text.text-darken-2 {

color: #D32F2F !important;

}

.red.darken-3 {

background-color: #C62828 !important;

}

.red-text.text-darken-3 {

color: #C62828 !important;

}

.red.darken-4 {

background-color: #B71C1C !important;

}

.red-text.text-darken-4 {

color: #B71C1C !important;

}

.red.accent-1 {

background-color: #FF8A80 !important;

}

.red-text.text-accent-1 {

color: #FF8A80 !important;

}

.red.accent-2 {

background-color: #FF5252 !important;

}

.red-text.text-accent-2 {

color: #FF5252 !important;

}

.red.accent-3 {

background-color: #FF1744 !important;

}

.red-text.text-accent-3 {

color: #FF1744 !important;

}

.red.accent-4 {

background-color: #D50000 !important;

}

.red-text.text-accent-4 {

color: #D50000 !important;

}

.pink {

background-color: #e91e63 !important;

}

.pink-text {

color: #e91e63 !important;

}

.pink.lighten-5 {

background-color: #fce4ec !important;

}

.pink-text.text-lighten-5 {

color: #fce4ec !important;

}

.pink.lighten-4 {

background-color: #f8bbd0 !important;

}

.pink-text.text-lighten-4 {

color: #f8bbd0 !important;

}

.pink.lighten-3 {

background-color: #f48fb1 !important;

}

.pink-text.text-lighten-3 {

color: #f48fb1 !important;

}

.pink.lighten-2 {

background-color: #f06292 !important;

}

.pink-text.text-lighten-2 {

color: #f06292 !important;

}

.pink.lighten-1 {

background-color: #ec407a !important;

}

.pink-text.text-lighten-1 {

color: #ec407a !important;

}

.pink.darken-1 {

background-color: #d81b60 !important;

}

.pink-text.text-darken-1 {

color: #d81b60 !important;

}

.pink.darken-2 {

background-color: #c2185b !important;

}

.pink-text.text-darken-2 {

color: #c2185b !important;

}

.pink.darken-3 {

background-color: #ad1457 !important;

}

.pink-text.text-darken-3 {

color: #ad1457 !important;

}

.pink.darken-4 {

background-color: #880e4f !important;

}

.pink-text.text-darken-4 {

color: #880e4f !important;

}

.pink.accent-1 {

background-color: #ff80ab !important;

}

.pink-text.text-accent-1 {

color: #ff80ab !important;

}

.pink.accent-2 {

background-color: #ff4081 !important;

}

.pink-text.text-accent-2 {

color: #ff4081 !important;

}

.pink.accent-3 {

background-color: #f50057 !important;

}

.pink-text.text-accent-3 {

color: #f50057 !important;

}

.pink.accent-4 {

background-color: #c51162 !important;

}

.pink-text.text-accent-4 {

color: #c51162 !important;

}

.purple {

background-color: #9c27b0 !important;

}

.purple-text {

color: #9c27b0 !important;

}

.purple.lighten-5 {

background-color: #f3e5f5 !important;

}

.purple-text.text-lighten-5 {

color: #f3e5f5 !important;

}

.purple.lighten-4 {

background-color: #e1bee7 !important;

}

.purple-text.text-lighten-4 {

color: #e1bee7 !important;

}

.purple.lighten-3 {

background-color: #ce93d8 !important;

}

.purple-text.text-lighten-3 {

color: #ce93d8 !important;

}

.purple.lighten-2 {

background-color: #ba68c8 !important;

}

.purple-text.text-lighten-2 {

color: #ba68c8 !important;

}

.purple.lighten-1 {

background-color: #ab47bc !important;

}

.purple-text.text-lighten-1 {

color: #ab47bc !important;

}

.purple.darken-1 {

background-color: #8e24aa !important;

}

.purple-text.text-darken-1 {

color: #8e24aa !important;

}

.purple.darken-2 {

background-color: #7b1fa2 !important;

}

.purple-text.text-darken-2 {

color: #7b1fa2 !important;

}

.purple.darken-3 {

background-color: #6a1b9a !important;

}

.purple-text.text-darken-3 {

color: #6a1b9a !important;

}

.purple.darken-4 {

background-color: #4a148c !important;

}

.purple-text.text-darken-4 {

color: #4a148c !important;

}

.purple.accent-1 {

background-color: #ea80fc !important;

}

.purple-text.text-accent-1 {

color: #ea80fc !important;

}

.purple.accent-2 {

background-color: #e040fb !important;

}

.purple-text.text-accent-2 {

color: #e040fb !important;

}

.purple.accent-3 {

background-color: #d500f9 !important;

}

.purple-text.text-accent-3 {

color: #d500f9 !important;

}

.purple.accent-4 {

background-color: #aa00ff !important;

}

.purple-text.text-accent-4 {

color: #aa00ff !important;

}

.deep-purple {

background-color: #673ab7 !important;

}

.deep-purple-text {

color: #673ab7 !important;

}

.deep-purple.lighten-5 {

background-color: #ede7f6 !important;

}

.deep-purple-text.text-lighten-5 {

color: #ede7f6 !important;

}

.deep-purple.lighten-4 {

background-color: #d1c4e9 !important;

}

.deep-purple-text.text-lighten-4 {

color: #d1c4e9 !important;

}

.deep-purple.lighten-3 {

background-color: #b39ddb !important;

}

.deep-purple-text.text-lighten-3 {

color: #b39ddb !important;

}

.deep-purple.lighten-2 {

background-color: #9575cd !important;

}

.deep-purple-text.text-lighten-2 {

color: #9575cd !important;

}

.deep-purple.lighten-1 {

background-color: #7e57c2 !important;

}

.deep-purple-text.text-lighten-1 {

color: #7e57c2 !important;

}

.deep-purple.darken-1 {

background-color: #5e35b1 !important;

}

.deep-purple-text.text-darken-1 {

color: #5e35b1 !important;

}

.deep-purple.darken-2 {

background-color: #512da8 !important;

}

.deep-purple-text.text-darken-2 {

color: #512da8 !important;

}

.deep-purple.darken-3 {

background-color: #4527a0 !important;

}

.deep-purple-text.text-darken-3 {

color: #4527a0 !important;

}

.deep-purple.darken-4 {

background-color: #311b92 !important;

}

.deep-purple-text.text-darken-4 {

color: #311b92 !important;

}

.deep-purple.accent-1 {

background-color: #b388ff !important;

}

.deep-purple-text.text-accent-1 {

color: #b388ff !important;

}

.deep-purple.accent-2 {

background-color: #7c4dff !important;

}

.deep-purple-text.text-accent-2 {

color: #7c4dff !important;

}

.deep-purple.accent-3 {

background-color: #651fff !important;

}

.deep-purple-text.text-accent-3 {

color: #651fff !important;

}

.deep-purple.accent-4 {

background-color: #6200ea !important;

}

.deep-purple-text.text-accent-4 {

color: #6200ea !important;

}

.indigo {

background-color: #3f51b5 !important;

}

.indigo-text {

color: #3f51b5 !important;

}

.indigo.lighten-5 {

background-color: #e8eaf6 !important;

}

.indigo-text.text-lighten-5 {

color: #e8eaf6 !important;

}

.indigo.lighten-4 {

background-color: #c5cae9 !important;

}

.indigo-text.text-lighten-4 {

color: #c5cae9 !important;

}

.indigo.lighten-3 {

background-color: #9fa8da !important;

}

.indigo-text.text-lighten-3 {

color: #9fa8da !important;

}

.indigo.lighten-2 {

background-color: #7986cb !important;

}

.indigo-text.text-lighten-2 {

color: #7986cb !important;

}

.indigo.lighten-1 {

background-color: #5c6bc0 !important;

}

.indigo-text.text-lighten-1 {

color: #5c6bc0 !important;

}

.indigo.darken-1 {

background-color: #3949ab !important;

}

.indigo-text.text-darken-1 {

color: #3949ab !important;

}

.indigo.darken-2 {

background-color: #303f9f !important;

}

.indigo-text.text-darken-2 {

color: #303f9f !important;

}

.indigo.darken-3 {

background-color: #283593 !important;

}

.indigo-text.text-darken-3 {

color: #283593 !important;

}

.indigo.darken-4 {

background-color: #1a237e !important;

}

.indigo-text.text-darken-4 {

color: #1a237e !important;

}

.indigo.accent-1 {

background-color: #8c9eff !important;

}

.indigo-text.text-accent-1 {

color: #8c9eff !important;

}

.indigo.accent-2 {

background-color: #536dfe !important;

}

.indigo-text.text-accent-2 {

color: #536dfe !important;

}

.indigo.accent-3 {

background-color: #3d5afe !important;

}

.indigo-text.text-accent-3 {

color: #3d5afe !important;

}

.indigo.accent-4 {

background-color: #304ffe !important;

}

.indigo-text.text-accent-4 {

color: #304ffe !important;

}

.blue {

background-color: #2196F3 !important;

}

.blue-text {

color: #2196F3 !important;

}

.blue.lighten-5 {

background-color: #E3F2FD !important;

}

.blue-text.text-lighten-5 {

color: #E3F2FD !important;

}

.blue.lighten-4 {

background-color: #BBDEFB !important;

}

.blue-text.text-lighten-4 {

color: #BBDEFB !important;

}

.blue.lighten-3 {

background-color: #90CAF9 !important;

}

.blue-text.text-lighten-3 {

color: #90CAF9 !important;

}

.blue.lighten-2 {

background-color: #64B5F6 !important;

}

.blue-text.text-lighten-2 {

color: #64B5F6 !important;

}

.blue.lighten-1 {

background-color: #42A5F5 !important;

}

.blue-text.text-lighten-1 {

color: #42A5F5 !important;

}

.blue.darken-1 {

background-color: #1E88E5 !important;

}

.blue-text.text-darken-1 {

color: #1E88E5 !important;

}

.blue.darken-2 {

background-color: #1976D2 !important;

}

.blue-text.text-darken-2 {

color: #1976D2 !important;

}

.blue.darken-3 {

background-color: #1565C0 !important;

}

.blue-text.text-darken-3 {

color: #1565C0 !important;

}

.blue.darken-4 {

background-color: #0D47A1 !important;

}

.blue-text.text-darken-4 {

color: #0D47A1 !important;

}

.blue.accent-1 {

background-color: #82B1FF !important;

}

.blue-text.text-accent-1 {

color: #82B1FF !important;

}

.blue.accent-2 {

background-color: #448AFF !important;

}

.blue-text.text-accent-2 {

color: #448AFF !important;

}

.blue.accent-3 {

background-color: #2979FF !important;

}

.blue-text.text-accent-3 {

color: #2979FF !important;

}

.blue.accent-4 {

background-color: #2962FF !important;

}

.blue-text.text-accent-4 {

color: #2962FF !important;

}

.light-blue {

background-color: #03a9f4 !important;

}

.light-blue-text {

color: #03a9f4 !important;

}

.light-blue.lighten-5 {

background-color: #e1f5fe !important;

}

.light-blue-text.text-lighten-5 {

color: #e1f5fe !important;

}

.light-blue.lighten-4 {

background-color: #b3e5fc !important;

}

.light-blue-text.text-lighten-4 {

color: #b3e5fc !important;

}

.light-blue.lighten-3 {

background-color: #81d4fa !important;

}

.light-blue-text.text-lighten-3 {

color: #81d4fa !important;

}

.light-blue.lighten-2 {

background-color: #4fc3f7 !important;

}

.light-blue-text.text-lighten-2 {

color: #4fc3f7 !important;

}

.light-blue.lighten-1 {

background-color: #29b6f6 !important;

}

.light-blue-text.text-lighten-1 {

color: #29b6f6 !important;

}

.light-blue.darken-1 {

background-color: #039be5 !important;

}

.light-blue-text.text-darken-1 {

color: #039be5 !important;

}

.light-blue.darken-2 {

background-color: #0288d1 !important;

}

.light-blue-text.text-darken-2 {

color: #0288d1 !important;

}

.light-blue.darken-3 {

background-color: #0277bd !important;

}

.light-blue-text.text-darken-3 {

color: #0277bd !important;

}

.light-blue.darken-4 {

background-color: #01579b !important;

}

.light-blue-text.text-darken-4 {

color: #01579b !important;

}

.light-blue.accent-1 {

background-color: #80d8ff !important;

}

.light-blue-text.text-accent-1 {

color: #80d8ff !important;

}

.light-blue.accent-2 {

background-color: #40c4ff !important;

}

.light-blue-text.text-accent-2 {

color: #40c4ff !important;

}

.light-blue.accent-3 {

background-color: #00b0ff !important;

}

.light-blue-text.text-accent-3 {

color: #00b0ff !important;

}

.light-blue.accent-4 {

background-color: #0091ea !important;

}

.light-blue-text.text-accent-4 {

color: #0091ea !important;

}

.cyan {

background-color: #00bcd4 !important;

}

.cyan-text {

color: #00bcd4 !important;

}

.cyan.lighten-5 {

background-color: #e0f7fa !important;

}

.cyan-text.text-lighten-5 {

color: #e0f7fa !important;

}

.cyan.lighten-4 {

background-color: #b2ebf2 !important;

}

.cyan-text.text-lighten-4 {

color: #b2ebf2 !important;

}

.cyan.lighten-3 {

background-color: #80deea !important;

}

.cyan-text.text-lighten-3 {

color: #80deea !important;

}

.cyan.lighten-2 {

background-color: #4dd0e1 !important;

}

.cyan-text.text-lighten-2 {

color: #4dd0e1 !important;

}

.cyan.lighten-1 {

background-color: #26c6da !important;

}

.cyan-text.text-lighten-1 {

color: #26c6da !important;

}

.cyan.darken-1 {

background-color: #00acc1 !important;

}

.cyan-text.text-darken-1 {

color: #00acc1 !important;

}

.cyan.darken-2 {

background-color: #0097a7 !important;

}

.cyan-text.text-darken-2 {

color: #0097a7 !important;

}

.cyan.darken-3 {

background-color: #00838f !important;

}

.cyan-text.text-darken-3 {

color: #00838f !important;

}

.cyan.darken-4 {

background-color: #006064 !important;

}

.cyan-text.text-darken-4 {

color: #006064 !important;

}

.cyan.accent-1 {

background-color: #84ffff !important;

}

.cyan-text.text-accent-1 {

color: #84ffff !important;

}

.cyan.accent-2 {

background-color: #18ffff !important;

}

.cyan-text.text-accent-2 {

color: #18ffff !important;

}

.cyan.accent-3 {

background-color: #00e5ff !important;

}

.cyan-text.text-accent-3 {

color: #00e5ff !important;

}

.cyan.accent-4 {

background-color: #00b8d4 !important;

}

.cyan-text.text-accent-4 {

color: #00b8d4 !important;

}

.teal {

background-color: #009688 !important;

}

.teal-text {

color: #009688 !important;

}

.teal.lighten-5 {

background-color: #e0f2f1 !important;

}

.teal-text.text-lighten-5 {

color: #e0f2f1 !important;

}

.teal.lighten-4 {

background-color: #b2dfdb !important;

}

.teal-text.text-lighten-4 {

color: #b2dfdb !important;

}

.teal.lighten-3 {

background-color: #80cbc4 !important;

}

.teal-text.text-lighten-3 {

color: #80cbc4 !important;

}

.teal.lighten-2 {

background-color: #4db6ac !important;

}

.teal-text.text-lighten-2 {

color: #4db6ac !important;

}

.teal.lighten-1 {

background-color: #26a69a !important;

}

.teal-text.text-lighten-1 {

color: #26a69a !important;

}

.teal.darken-1 {

background-color: #00897b !important;

}

.teal-text.text-darken-1 {

color: #00897b !important;

}

.teal.darken-2 {

background-color: #00796b !important;

}

.teal-text.text-darken-2 {

color: #00796b !important;

}

.teal.darken-3 {

background-color: #00695c !important;

}

.teal-text.text-darken-3 {

color: #00695c !important;

}

.teal.darken-4 {

background-color: #004d40 !important;

}

.teal-text.text-darken-4 {

color: #004d40 !important;

}

.teal.accent-1 {

background-color: #a7ffeb !important;

}

.teal-text.text-accent-1 {

color: #a7ffeb !important;

}

.teal.accent-2 {

background-color: #64ffda !important;

}

.teal-text.text-accent-2 {

color: #64ffda !important;

}

.teal.accent-3 {

background-color: #1de9b6 !important;

}

.teal-text.text-accent-3 {

color: #1de9b6 !important;

}

.teal.accent-4 {

background-color: #00bfa5 !important;

}

.teal-text.text-accent-4 {

color: #00bfa5 !important;

}

.green {

background-color: #4CAF50 !important;

}

.green-text {

color: #4CAF50 !important;

}

.green.lighten-5 {

background-color: #E8F5E9 !important;

}

.green-text.text-lighten-5 {

color: #E8F5E9 !important;

}

.green.lighten-4 {

background-color: #C8E6C9 !important;

}

.green-text.text-lighten-4 {

color: #C8E6C9 !important;

}

.green.lighten-3 {

background-color: #A5D6A7 !important;

}

.green-text.text-lighten-3 {

color: #A5D6A7 !important;

}

.green.lighten-2 {

background-color: #81C784 !important;

}

.green-text.text-lighten-2 {

color: #81C784 !important;

}

.green.lighten-1 {

background-color: #66BB6A !important;

}

.green-text.text-lighten-1 {

color: #66BB6A !important;

}

.green.darken-1 {

background-color: #43A047 !important;

}

.green-text.text-darken-1 {

color: #43A047 !important;

}

.green.darken-2 {

background-color: #388E3C !important;

}

.green-text.text-darken-2 {

color: #388E3C !important;

}

.green.darken-3 {

background-color: #2E7D32 !important;

}

.green-text.text-darken-3 {

color: #2E7D32 !important;

}

.green.darken-4 {

background-color: #1B5E20 !important;

}

.green-text.text-darken-4 {

color: #1B5E20 !important;

}

.green.accent-1 {

background-color: #B9F6CA !important;

}

.green-text.text-accent-1 {

color: #B9F6CA !important;

}

.green.accent-2 {

background-color: #69F0AE !important;

}

.green-text.text-accent-2 {

color: #69F0AE !important;

}

.green.accent-3 {

background-color: #00E676 !important;

}

.green-text.text-accent-3 {

color: #00E676 !important;

}

.green.accent-4 {

background-color: #00C853 !important;

}

.green-text.text-accent-4 {

color: #00C853 !important;

}

.light-green {

background-color: #8bc34a !important;

}

.light-green-text {

color: #8bc34a !important;

}

.light-green.lighten-5 {

background-color: #f1f8e9 !important;

}

.light-green-text.text-lighten-5 {

color: #f1f8e9 !important;

}

.light-green.lighten-4 {

background-color: #dcedc8 !important;

}

.light-green-text.text-lighten-4 {

color: #dcedc8 !important;

}

.light-green.lighten-3 {

background-color: #c5e1a5 !important;

}

.light-green-text.text-lighten-3 {

color: #c5e1a5 !important;

}

.light-green.lighten-2 {

background-color: #aed581 !important;

}

.light-green-text.text-lighten-2 {

color: #aed581 !important;

}

.light-green.lighten-1 {

background-color: #9ccc65 !important;

}

.light-green-text.text-lighten-1 {

color: #9ccc65 !important;

}

.light-green.darken-1 {

background-color: #7cb342 !important;

}

.light-green-text.text-darken-1 {

color: #7cb342 !important;

}

.light-green.darken-2 {

background-color: #689f38 !important;

}

.light-green-text.text-darken-2 {

color: #689f38 !important;

}

.light-green.darken-3 {

background-color: #558b2f !important;

}

.light-green-text.text-darken-3 {

color: #558b2f !important;

}

.light-green.darken-4 {

background-color: #33691e !important;

}

.light-green-text.text-darken-4 {

color: #33691e !important;

}

.light-green.accent-1 {

background-color: #ccff90 !important;

}

.light-green-text.text-accent-1 {

color: #ccff90 !important;

}

.light-green.accent-2 {

background-color: #b2ff59 !important;

}

.light-green-text.text-accent-2 {

color: #b2ff59 !important;

}

.light-green.accent-3 {

background-color: #76ff03 !important;

}

.light-green-text.text-accent-3 {

color: #76ff03 !important;

}

.light-green.accent-4 {

background-color: #64dd17 !important;

}

.light-green-text.text-accent-4 {

color: #64dd17 !important;

}

.lime {

background-color: #cddc39 !important;

}

.lime-text {

color: #cddc39 !important;

}

.lime.lighten-5 {

background-color: #f9fbe7 !important;

}

.lime-text.text-lighten-5 {

color: #f9fbe7 !important;

}

.lime.lighten-4 {

background-color: #f0f4c3 !important;

}

.lime-text.text-lighten-4 {

color: #f0f4c3 !important;

}

.lime.lighten-3 {

background-color: #e6ee9c !important;

}

.lime-text.text-lighten-3 {

color: #e6ee9c !important;

}

.lime.lighten-2 {

background-color: #dce775 !important;

}

.lime-text.text-lighten-2 {

color: #dce775 !important;

}

.lime.lighten-1 {

background-color: #d4e157 !important;

}

.lime-text.text-lighten-1 {

color: #d4e157 !important;

}

.lime.darken-1 {

background-color: #c0ca33 !important;

}

.lime-text.text-darken-1 {

color: #c0ca33 !important;

}

.lime.darken-2 {

background-color: #afb42b !important;

}

.lime-text.text-darken-2 {

color: #afb42b !important;

}

.lime.darken-3 {

background-color: #9e9d24 !important;

}

.lime-text.text-darken-3 {

color: #9e9d24 !important;

}

.lime.darken-4 {

background-color: #827717 !important;

}

.lime-text.text-darken-4 {

color: #827717 !important;

}

.lime.accent-1 {

background-color: #f4ff81 !important;

}

.lime-text.text-accent-1 {

color: #f4ff81 !important;

}

.lime.accent-2 {

background-color: #eeff41 !important;

}

.lime-text.text-accent-2 {

color: #eeff41 !important;

}

.lime.accent-3 {

background-color: #c6ff00 !important;

}

.lime-text.text-accent-3 {

color: #c6ff00 !important;

}

.lime.accent-4 {

background-color: #aeea00 !important;

}

.lime-text.text-accent-4 {

color: #aeea00 !important;

}

.yellow {

background-color: #ffeb3b !important;

}

.yellow-text {

color: #ffeb3b !important;

}

.yellow.lighten-5 {

background-color: #fffde7 !important;

}

.yellow-text.text-lighten-5 {

color: #fffde7 !important;

}

.yellow.lighten-4 {

background-color: #fff9c4 !important;

}

.yellow-text.text-lighten-4 {

color: #fff9c4 !important;

}

.yellow.lighten-3 {

background-color: #fff59d !important;

}

.yellow-text.text-lighten-3 {

color: #fff59d !important;

}

.yellow.lighten-2 {

background-color: #fff176 !important;

}

.yellow-text.text-lighten-2 {

color: #fff176 !important;

}

.yellow.lighten-1 {

background-color: #ffee58 !important;

}

.yellow-text.text-lighten-1 {

color: #ffee58 !important;

}

.yellow.darken-1 {

background-color: #fdd835 !important;

}

.yellow-text.text-darken-1 {

color: #fdd835 !important;

}

.yellow.darken-2 {

background-color: #fbc02d !important;

}

.yellow-text.text-darken-2 {

color: #fbc02d !important;

}

.yellow.darken-3 {

background-color: #f9a825 !important;

}

.yellow-text.text-darken-3 {

color: #f9a825 !important;

}

.yellow.darken-4 {

background-color: #f57f17 !important;

}

.yellow-text.text-darken-4 {

color: #f57f17 !important;

}

.yellow.accent-1 {

background-color: #ffff8d !important;

}

.yellow-text.text-accent-1 {

color: #ffff8d !important;

}

.yellow.accent-2 {

background-color: #ffff00 !important;

}

.yellow-text.text-accent-2 {

color: #ffff00 !important;

}

.yellow.accent-3 {

background-color: #ffea00 !important;

}

.yellow-text.text-accent-3 {

color: #ffea00 !important;

}

.yellow.accent-4 {

background-color: #ffd600 !important;

}

.yellow-text.text-accent-4 {

color: #ffd600 !important;

}

.amber {

background-color: #ffc107 !important;

}

.amber-text {

color: #ffc107 !important;

}

.amber.lighten-5 {

background-color: #fff8e1 !important;

}

.amber-text.text-lighten-5 {

color: #fff8e1 !important;

}

.amber.lighten-4 {

background-color: #ffecb3 !important;

}

.amber-text.text-lighten-4 {

color: #ffecb3 !important;

}

.amber.lighten-3 {

background-color: #ffe082 !important;

}

.amber-text.text-lighten-3 {

color: #ffe082 !important;

}

.amber.lighten-2 {

background-color: #ffd54f !important;

}

.amber-text.text-lighten-2 {

color: #ffd54f !important;

}

.amber.lighten-1 {

background-color: #ffca28 !important;

}

.amber-text.text-lighten-1 {

color: #ffca28 !important;

}

.amber.darken-1 {

background-color: #ffb300 !important;

}

.amber-text.text-darken-1 {

color: #ffb300 !important;

}

.amber.darken-2 {

background-color: #ffa000 !important;

}

.amber-text.text-darken-2 {

color: #ffa000 !important;

}

.amber.darken-3 {

background-color: #ff8f00 !important;

}

.amber-text.text-darken-3 {

color: #ff8f00 !important;

}

.amber.darken-4 {

background-color: #ff6f00 !important;

}

.amber-text.text-darken-4 {

color: #ff6f00 !important;

}

.amber.accent-1 {

background-color: #ffe57f !important;

}

.amber-text.text-accent-1 {

color: #ffe57f !important;

}

.amber.accent-2 {

background-color: #ffd740 !important;

}

.amber-text.text-accent-2 {

color: #ffd740 !important;

}

.amber.accent-3 {

background-color: #ffc400 !important;

}

.amber-text.text-accent-3 {

color: #ffc400 !important;

}

.amber.accent-4 {

background-color: #ffab00 !important;

}

.amber-text.text-accent-4 {

color: #ffab00 !important;

}

.orange {

background-color: #ff9800 !important;

}

.orange-text {

color: #ff9800 !important;

}

.orange.lighten-5 {

background-color: #fff3e0 !important;

}

.orange-text.text-lighten-5 {

color: #fff3e0 !important;

}

.orange.lighten-4 {

background-color: #ffe0b2 !important;

}

.orange-text.text-lighten-4 {

color: #ffe0b2 !important;

}

.orange.lighten-3 {

background-color: #ffcc80 !important;

}

.orange-text.text-lighten-3 {

color: #ffcc80 !important;

}

.orange.lighten-2 {

background-color: #ffb74d !important;

}

.orange-text.text-lighten-2 {

color: #ffb74d !important;

}

.orange.lighten-1 {

background-color: #ffa726 !important;

}

.orange-text.text-lighten-1 {

color: #ffa726 !important;

}

.orange.darken-1 {

background-color: #fb8c00 !important;

}

.orange-text.text-darken-1 {

color: #fb8c00 !important;

}

.orange.darken-2 {

background-color: #f57c00 !important;

}

.orange-text.text-darken-2 {

color: #f57c00 !important;

}

.orange.darken-3 {

background-color: #ef6c00 !important;

}

.orange-text.text-darken-3 {

color: #ef6c00 !important;

}

.orange.darken-4 {

background-color: #e65100 !important;

}

.orange-text.text-darken-4 {

color: #e65100 !important;

}

.orange.accent-1 {

background-color: #ffd180 !important;

}

.orange-text.text-accent-1 {

color: #ffd180 !important;

}

.orange.accent-2 {

background-color: #ffab40 !important;

}

.orange-text.text-accent-2 {

color: #ffab40 !important;

}

.orange.accent-3 {

background-color: #ff9100 !important;

}

.orange-text.text-accent-3 {

color: #ff9100 !important;

}

.orange.accent-4 {

background-color: #ff6d00 !important;

}

.orange-text.text-accent-4 {

color: #ff6d00 !important;

}

.deep-orange {

background-color: #ff5722 !important;

}

.deep-orange-text {

color: #ff5722 !important;

}

.deep-orange.lighten-5 {

background-color: #fbe9e7 !important;

}

.deep-orange-text.text-lighten-5 {

color: #fbe9e7 !important;

}

.deep-orange.lighten-4 {

background-color: #ffccbc !important;

}

.deep-orange-text.text-lighten-4 {

color: #ffccbc !important;

}

.deep-orange.lighten-3 {

background-color: #ffab91 !important;

}

.deep-orange-text.text-lighten-3 {

color: #ffab91 !important;

}

.deep-orange.lighten-2 {

background-color: #ff8a65 !important;

}

.deep-orange-text.text-lighten-2 {

color: #ff8a65 !important;

}

.deep-orange.lighten-1 {

background-color: #ff7043 !important;

}

.deep-orange-text.text-lighten-1 {

color: #ff7043 !important;

}

.deep-orange.darken-1 {

background-color: #f4511e !important;

}

.deep-orange-text.text-darken-1 {

color: #f4511e !important;

}

.deep-orange.darken-2 {

background-color: #e64a19 !important;

}

.deep-orange-text.text-darken-2 {

color: #e64a19 !important;

}

.deep-orange.darken-3 {

background-color: #d84315 !important;

}

.deep-orange-text.text-darken-3 {

color: #d84315 !important;

}

.deep-orange.darken-4 {

background-color: #bf360c !important;

}

.deep-orange-text.text-darken-4 {

color: #bf360c !important;

}

.deep-orange.accent-1 {

background-color: #ff9e80 !important;

}

.deep-orange-text.text-accent-1 {

color: #ff9e80 !important;

}

.deep-orange.accent-2 {

background-color: #ff6e40 !important;

}

.deep-orange-text.text-accent-2 {

color: #ff6e40 !important;

}

.deep-orange.accent-3 {

background-color: #ff3d00 !important;

}

.deep-orange-text.text-accent-3 {

color: #ff3d00 !important;

}

.deep-orange.accent-4 {

background-color: #dd2c00 !important;

}

.deep-orange-text.text-accent-4 {

color: #dd2c00 !important;

}

.brown {

background-color: #795548 !important;

}

.brown-text {

color: #795548 !important;

}

.brown.lighten-5 {

background-color: #efebe9 !important;

}

.brown-text.text-lighten-5 {

color: #efebe9 !important;

}

.brown.lighten-4 {

background-color: #d7ccc8 !important;

}

.brown-text.text-lighten-4 {

color: #d7ccc8 !important;

}

.brown.lighten-3 {

background-color: #bcaaa4 !important;

}

.brown-text.text-lighten-3 {

color: #bcaaa4 !important;

}

.brown.lighten-2 {

background-color: #a1887f !important;

}

.brown-text.text-lighten-2 {

color: #a1887f !important;

}

.brown.lighten-1 {

background-color: #8d6e63 !important;

}

.brown-text.text-lighten-1 {

color: #8d6e63 !important;

}

.brown.darken-1 {

background-color: #6d4c41 !important;

}

.brown-text.text-darken-1 {

color: #6d4c41 !important;

}

.brown.darken-2 {

background-color: #5d4037 !important;

}

.brown-text.text-darken-2 {

color: #5d4037 !important;

}

.brown.darken-3 {

background-color: #4e342e !important;

}

.brown-text.text-darken-3 {

color: #4e342e !important;

}

.brown.darken-4 {

background-color: #3e2723 !important;

}

.brown-text.text-darken-4 {

color: #3e2723 !important;

}

.blue-grey {

background-color: #607d8b !important;

}

.blue-grey-text {

color: #607d8b !important;

}

.blue-grey.lighten-5 {

background-color: #eceff1 !important;

}

.blue-grey-text.text-lighten-5 {

color: #eceff1 !important;

}

.blue-grey.lighten-4 {

background-color: #cfd8dc !important;

}

.blue-grey-text.text-lighten-4 {

color: #cfd8dc !important;

}

.blue-grey.lighten-3 {

background-color: #b0bec5 !important;

}

.blue-grey-text.text-lighten-3 {

color: #b0bec5 !important;

}

.blue-grey.lighten-2 {

background-color: #90a4ae !important;

}

.blue-grey-text.text-lighten-2 {

color: #90a4ae !important;

}

.blue-grey.lighten-1 {

background-color: #78909c !important;

}

.blue-grey-text.text-lighten-1 {

color: #78909c !important;

}

.blue-grey.darken-1 {

background-color: #546e7a !important;

}

.blue-grey-text.text-darken-1 {

color: #546e7a !important;

}

.blue-grey.darken-2 {

background-color: #455a64 !important;

}

.blue-grey-text.text-darken-2 {

color: #455a64 !important;

}

.blue-grey.darken-3 {

background-color: #37474f !important;

}

.blue-grey-text.text-darken-3 {

color: #37474f !important;

}

.blue-grey.darken-4 {

background-color: #263238 !important;

}

.blue-grey-text.text-darken-4 {

color: #263238 !important;

}

.grey {

background-color: #9e9e9e !important;

}

.grey-text {

color: #9e9e9e !important;

}

.grey.lighten-5 {

background-color: #fafafa !important;

}

.grey-text.text-lighten-5 {

color: #fafafa !important;

}

.grey.lighten-4 {

background-color: #f5f5f5 !important;

}

.grey-text.text-lighten-4 {

color: #f5f5f5 !important;

}

.grey.lighten-3 {

background-color: #eeeeee !important;

}

.grey-text.text-lighten-3 {

color: #eeeeee !important;

}

.grey.lighten-2 {

background-color: #e0e0e0 !important;

}

.grey-text.text-lighten-2 {

color: #e0e0e0 !important;

}

.grey.lighten-1 {

background-color: #bdbdbd !important;

}

.grey-text.text-lighten-1 {

color: #bdbdbd !important;

}

.grey.darken-1 {

background-color: #757575 !important;

}

.grey-text.text-darken-1 {

color: #757575 !important;

}

.grey.darken-2 {

background-color: #616161 !important;

}

.grey-text.text-darken-2 {

color: #616161 !important;

}

.grey.darken-3 {

background-color: #424242 !important;

}

.grey-text.text-darken-3 {

color: #424242 !important;

}

.grey.darken-4 {

background-color: #212121 !important;

}

.grey-text.text-darken-4 {

color: #212121 !important;

}

.black {

background-color: #000000 !important;

}

.black-text {

color: #000000 !important;

}

.white {

background-color: #FFFFFF !important;

}

.white-text {

color: #FFFFFF !important;

}

.transparent {

background-color: transparent !important;

}

.transparent-text {

color: transparent !important;

}

html {

line-height: 1.15;

/\* 1 \*/

-ms-text-size-adjust: 100%;

/\* 2 \*/

-webkit-text-size-adjust: 100%;

/\* 2 \*/

}

/\* Sections

========================================================================== \*/

/\*\*

\* Remove the margin in all browsers (opinionated).

\*/

body {

margin: 0;

}

/\*\*

\* Add the correct display in IE 9-.

\*/

article,

aside,

footer,

header,

nav,

section {

display: block;

}

/\*\*

\* Correct the font size and margin on `h1` elements within `section` and

\* `article` contexts in Chrome, Firefox, and Safari.

\*/

h1 {

font-size: 2em;

margin: 0.67em 0;

}

/\* Grouping content

========================================================================== \*/

/\*\*

\* Add the correct display in IE 9-.

\* 1. Add the correct display in IE.

\*/

figcaption,

figure,

main {

/\* 1 \*/

display: block;

}

/\*\*

\* Add the correct margin in IE 8.

\*/

figure {

margin: 1em 40px;

}

/\*\*

\* 1. Add the correct box sizing in Firefox.

\* 2. Show the overflow in Edge and IE.

\*/

hr {

-webkit-box-sizing: content-box;

box-sizing: content-box;

/\* 1 \*/

height: 0;

/\* 1 \*/

overflow: visible;

/\* 2 \*/

}

/\*\*

\* 1. Correct the inheritance and scaling of font size in all browsers.

\* 2. Correct the odd `em` font sizing in all browsers.

\*/

pre {

font-family: monospace, monospace;

/\* 1 \*/

font-size: 1em;

/\* 2 \*/

}

/\* Text-level semantics

========================================================================== \*/

/\*\*

\* 1. Remove the gray background on active links in IE 10.

\* 2. Remove gaps in links underline in iOS 8+ and Safari 8+.

\*/

a {

background-color: transparent;

/\* 1 \*/

-webkit-text-decoration-skip: objects;

/\* 2 \*/

}

/\*\*

\* 1. Remove the bottom border in Chrome 57- and Firefox 39-.

\* 2. Add the correct text decoration in Chrome, Edge, IE, Opera, and Safari.

\*/

abbr[title] {

border-bottom: none;

/\* 1 \*/

text-decoration: underline;

/\* 2 \*/

-webkit-text-decoration: underline dotted;

-moz-text-decoration: underline dotted;

text-decoration: underline dotted;

/\* 2 \*/

}

/\*\*

\* Prevent the duplicate application of `bolder` by the next rule in Safari 6.

\*/

b,

strong {

font-weight: inherit;

}

/\*\*

\* Add the correct font weight in Chrome, Edge, and Safari.

\*/

b,

strong {

font-weight: bolder;

}

/\*\*

\* 1. Correct the inheritance and scaling of font size in all browsers.

\* 2. Correct the odd `em` font sizing in all browsers.

\*/

code,

kbd,

samp {

font-family: monospace, monospace;

/\* 1 \*/

font-size: 1em;

/\* 2 \*/

}

/\*\*

\* Add the correct font style in Android 4.3-.

\*/

dfn {

font-style: italic;

}

/\*\*

\* Add the correct background and color in IE 9-.

\*/

mark {

background-color: #ff0;

color: #000;

}

/\*\*

\* Add the correct font size in all browsers.

\*/

small {

font-size: 80%;

}

/\*\*

\* Prevent `sub` and `sup` elements from affecting the line height in

\* all browsers.

\*/

sub,

sup {

font-size: 75%;

line-height: 0;

position: relative;

vertical-align: baseline;

}

sub {

bottom: -0.25em;

}

sup {

top: -0.5em;

}

/\* Embedded content

========================================================================== \*/

/\*\*

\* Add the correct display in IE 9-.

\*/

audio,

video {

display: inline-block;

}

/\*\*

\* Add the correct display in iOS 4-7.

\*/

audio:not([controls]) {

display: none;

height: 0;

}

/\*\*

\* Remove the border on images inside links in IE 10-.

\*/

img {

border-style: none;

}

/\*\*

\* Hide the overflow in IE.

\*/

svg:not(:root) {

overflow: hidden;

}

/\* Forms

========================================================================== \*/

/\*\*

\* 1. Change the font styles in all browsers (opinionated).

\* 2. Remove the margin in Firefox and Safari.

\*/

button,

input,

optgroup,

select,

textarea {

font-family: sans-serif;

/\* 1 \*/

font-size: 100%;

/\* 1 \*/

line-height: 1.15;

/\* 1 \*/

margin: 0;

/\* 2 \*/

}

/\*\*

\* Show the overflow in IE.

\* 1. Show the overflow in Edge.

\*/

button,

input {

/\* 1 \*/

overflow: visible;

}

/\*\*

\* Remove the inheritance of text transform in Edge, Firefox, and IE.

\* 1. Remove the inheritance of text transform in Firefox.

\*/

button,

select {

/\* 1 \*/

text-transform: none;

}

/\*\*

\* 1. Prevent a WebKit bug where (2) destroys native `audio` and `video`

\* controls in Android 4.

\* 2. Correct the inability to style clickable types in iOS and Safari.

\*/

button,

html [type="button"],

[type="reset"],

[type="submit"] {

-webkit-appearance: button;

/\* 2 \*/

}

/\*\*

\* Remove the inner border and padding in Firefox.

\*/

button::-moz-focus-inner,

[type="button"]::-moz-focus-inner,

[type="reset"]::-moz-focus-inner,

[type="submit"]::-moz-focus-inner {

border-style: none;

padding: 0;

}

/\*\*

\* Restore the focus styles unset by the previous rule.

\*/

button:-moz-focusring,

[type="button"]:-moz-focusring,

[type="reset"]:-moz-focusring,

[type="submit"]:-moz-focusring {

outline: 1px dotted ButtonText;

}

/\*\*

\* Correct the padding in Firefox.

\*/

fieldset {

padding: 0.35em 0.75em 0.625em;

}

/\*\*

\* 1. Correct the text wrapping in Edge and IE.

\* 2. Correct the color inheritance from `fieldset` elements in IE.

\* 3. Remove the padding so developers are not caught out when they zero out

\* `fieldset` elements in all browsers.

\*/

legend {

-webkit-box-sizing: border-box;

box-sizing: border-box;

/\* 1 \*/

color: inherit;

/\* 2 \*/

display: table;

/\* 1 \*/

max-width: 100%;

/\* 1 \*/

padding: 0;

/\* 3 \*/

white-space: normal;

/\* 1 \*/

}

/\*\*

\* 1. Add the correct display in IE 9-.

\* 2. Add the correct vertical alignment in Chrome, Firefox, and Opera.

\*/

progress {

display: inline-block;

/\* 1 \*/

vertical-align: baseline;

/\* 2 \*/

}

/\*\*

\* Remove the default vertical scrollbar in IE.

\*/

textarea {

overflow: auto;

}

/\*\*

\* 1. Add the correct box sizing in IE 10-.

\* 2. Remove the padding in IE 10-.

\*/

[type="checkbox"],

[type="radio"] {

-webkit-box-sizing: border-box;

box-sizing: border-box;

/\* 1 \*/

padding: 0;

/\* 2 \*/

}

/\*\*

\* Correct the cursor style of increment and decrement buttons in Chrome.

\*/

[type="number"]::-webkit-inner-spin-button,

[type="number"]::-webkit-outer-spin-button {

height: auto;

}

/\*\*

\* 1. Correct the odd appearance in Chrome and Safari.

\* 2. Correct the outline style in Safari.

\*/

[type="search"] {

-webkit-appearance: textfield;

/\* 1 \*/

outline-offset: -2px;

/\* 2 \*/

}

/\*\*

\* Remove the inner padding and cancel buttons in Chrome and Safari on macOS.

\*/

[type="search"]::-webkit-search-cancel-button,

[type="search"]::-webkit-search-decoration {

-webkit-appearance: none;

}

/\*\*

\* 1. Correct the inability to style clickable types in iOS and Safari.

\* 2. Change font properties to `inherit` in Safari.

\*/

::-webkit-file-upload-button {

-webkit-appearance: button;

/\* 1 \*/

font: inherit;

/\* 2 \*/

}

/\* Interactive

========================================================================== \*/

/\*

\* Add the correct display in IE 9-.

\* 1. Add the correct display in Edge, IE, and Firefox.

\*/

details,

menu {

display: block;

}

/\*

\* Add the correct display in all browsers.

\*/

summary {

display: list-item;

}

/\* Scripting

========================================================================== \*/

/\*\*

\* Add the correct display in IE 9-.

\*/

canvas {

display: inline-block;

}

/\*\*

\* Add the correct display in IE.

\*/

template {

display: none;

}

/\* Hidden

========================================================================== \*/

/\*\*

\* Add the correct display in IE 10-.

\*/

[hidden] {

display: none;

}

html {

-webkit-box-sizing: border-box;

box-sizing: border-box;

}

\*, \*:before, \*:after {

-webkit-box-sizing: inherit;

box-sizing: inherit;

}

button,

input,

optgroup,

select,

textarea {

font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, Oxygen-Sans, Ubuntu, Cantarell, "Helvetica Neue", sans-serif;

}

ul:not(.browser-default) {

padding-left: 0;

list-style-type: none;

}

ul:not(.browser-default) > li {

list-style-type: none;

}

a {

color: #039be5;

text-decoration: none;

-webkit-tap-highlight-color: transparent;

}

.valign-wrapper {

display: -webkit-box;

display: -webkit-flex;

display: -ms-flexbox;

display: flex;

-webkit-box-align: center;

-webkit-align-items: center;

-ms-flex-align: center;

align-items: center;

}

.clearfix {

clear: both;

}

.z-depth-0 {

-webkit-box-shadow: none !important;

box-shadow: none !important;

}

/\* 2dp elevation modified\*/

.z-depth-1, nav, .card-panel, .card, .toast, .btn, .btn-large, .btn-small, .btn-floating, .dropdown-content, .collapsible, .sidenav {

-webkit-box-shadow: 0 2px 2px 0 rgba(0, 0, 0, 0.14), 0 3px 1px -2px rgba(0, 0, 0, 0.12), 0 1px 5px 0 rgba(0, 0, 0, 0.2);

box-shadow: 0 2px 2px 0 rgba(0, 0, 0, 0.14), 0 3px 1px -2px rgba(0, 0, 0, 0.12), 0 1px 5px 0 rgba(0, 0, 0, 0.2);

}

.z-depth-1-half, .btn:hover, .btn-large:hover, .btn-small:hover, .btn-floating:hover {

-webkit-box-shadow: 0 3px 3px 0 rgba(0, 0, 0, 0.14), 0 1px 7px 0 rgba(0, 0, 0, 0.12), 0 3px 1px -1px rgba(0, 0, 0, 0.2);

box-shadow: 0 3px 3px 0 rgba(0, 0, 0, 0.14), 0 1px 7px 0 rgba(0, 0, 0, 0.12), 0 3px 1px -1px rgba(0, 0, 0, 0.2);

}

/\* 6dp elevation modified\*/

.z-depth-2 {

-webkit-box-shadow: 0 4px 5px 0 rgba(0, 0, 0, 0.14), 0 1px 10px 0 rgba(0, 0, 0, 0.12), 0 2px 4px -1px rgba(0, 0, 0, 0.3);

box-shadow: 0 4px 5px 0 rgba(0, 0, 0, 0.14), 0 1px 10px 0 rgba(0, 0, 0, 0.12), 0 2px 4px -1px rgba(0, 0, 0, 0.3);

}

/\* 12dp elevation modified\*/

.z-depth-3 {

-webkit-box-shadow: 0 8px 17px 2px rgba(0, 0, 0, 0.14), 0 3px 14px 2px rgba(0, 0, 0, 0.12), 0 5px 5px -3px rgba(0, 0, 0, 0.2);

box-shadow: 0 8px 17px 2px rgba(0, 0, 0, 0.14), 0 3px 14px 2px rgba(0, 0, 0, 0.12), 0 5px 5px -3px rgba(0, 0, 0, 0.2);

}

/\* 16dp elevation \*/

.z-depth-4 {

-webkit-box-shadow: 0 16px 24px 2px rgba(0, 0, 0, 0.14), 0 6px 30px 5px rgba(0, 0, 0, 0.12), 0 8px 10px -7px rgba(0, 0, 0, 0.2);

box-shadow: 0 16px 24px 2px rgba(0, 0, 0, 0.14), 0 6px 30px 5px rgba(0, 0, 0, 0.12), 0 8px 10px -7px rgba(0, 0, 0, 0.2);

}

/\* 24dp elevation \*/

.z-depth-5, .modal {

-webkit-box-shadow: 0 24px 38px 3px rgba(0, 0, 0, 0.14), 0 9px 46px 8px rgba(0, 0, 0, 0.12), 0 11px 15px -7px rgba(0, 0, 0, 0.2);

box-shadow: 0 24px 38px 3px rgba(0, 0, 0, 0.14), 0 9px 46px 8px rgba(0, 0, 0, 0.12), 0 11px 15px -7px rgba(0, 0, 0, 0.2);

}

.hoverable {

-webkit-transition: -webkit-box-shadow .25s;

transition: -webkit-box-shadow .25s;

transition: box-shadow .25s;

transition: box-shadow .25s, -webkit-box-shadow .25s;

}

.hoverable:hover {

-webkit-box-shadow: 0 8px 17px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);

box-shadow: 0 8px 17px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);

}

.divider {

height: 1px;

overflow: hidden;

background-color: #e0e0e0;

}

blockquote {

margin: 20px 0;

padding-left: 1.5rem;

border-left: 5px solid #ee6e73;

}

i {

line-height: inherit;

}

i.left {

float: left;

margin-right: 15px;

}

i.right {

float: right;

margin-left: 15px;

}

i.tiny {

font-size: 1rem;

}

i.small {

font-size: 2rem;

}

i.medium {

font-size: 4rem;

}

i.large {

font-size: 6rem;

}

img.responsive-img,

video.responsive-video {

max-width: 100%;

height: auto;

}

.pagination li {

display: inline-block;

border-radius: 2px;

text-align: center;

vertical-align: top;

height: 30px;

}

.pagination li a {

color: #444;

display: inline-block;

font-size: 1.2rem;

padding: 0 10px;

line-height: 30px;

}

.pagination li.active a {

color: #fff;

}

.pagination li.active {

background-color: #ee6e73;

}

.pagination li.disabled a {

cursor: default;

color: #999;

}

.pagination li i {

font-size: 2rem;

}

.pagination li.pages ul li {

display: inline-block;

float: none;

}

@media only screen and (max-width: 992px) {

.pagination {

width: 100%;

}

.pagination li.prev,

.pagination li.next {

width: 10%;

}

.pagination li.pages {

width: 80%;

overflow: hidden;

white-space: nowrap;

}

}

.breadcrumb {

font-size: 18px;

color: rgba(255, 255, 255, 0.7);

}

.breadcrumb i,

.breadcrumb [class^="mdi-"], .breadcrumb [class\*="mdi-"],

.breadcrumb i.material-icons {

display: inline-block;

float: left;

font-size: 24px;

}

.breadcrumb:before {

content: '\E5CC';

color: rgba(255, 255, 255, 0.7);

vertical-align: top;

display: inline-block;

font-family: 'Material Icons';

font-weight: normal;

font-style: normal;

font-size: 25px;

margin: 0 10px 0 8px;

-webkit-font-smoothing: antialiased;

}

.breadcrumb:first-child:before {

display: none;

}

.breadcrumb:last-child {

color: #fff;

}

.parallax-container {

position: relative;

overflow: hidden;

height: 500px;

}

.parallax-container .parallax {

position: absolute;

top: 0;

left: 0;

right: 0;

bottom: 0;

z-index: -1;

}

.parallax-container .parallax img {

opacity: 0;

position: absolute;

left: 50%;

bottom: 0;

min-width: 100%;

min-height: 100%;

-webkit-transform: translate3d(0, 0, 0);

transform: translate3d(0, 0, 0);

-webkit-transform: translateX(-50%);

transform: translateX(-50%);

}

.pin-top, .pin-bottom {

position: relative;

}

.pinned {

position: fixed !important;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Transition Classes

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

ul.staggered-list li {

opacity: 0;

}

.fade-in {

opacity: 0;

-webkit-transform-origin: 0 50%;

transform-origin: 0 50%;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Media Query Classes

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

@media only screen and (max-width: 600px) {

.hide-on-small-only, .hide-on-small-and-down {

display: none !important;

}

}

@media only screen and (max-width: 992px) {

.hide-on-med-and-down {

display: none !important;

}

}

@media only screen and (min-width: 601px) {

.hide-on-med-and-up {

display: none !important;

}

}

@media only screen and (min-width: 600px) and (max-width: 992px) {

.hide-on-med-only {

display: none !important;

}

}

@media only screen and (min-width: 993px) {

.hide-on-large-only {

display: none !important;

}

}

@media only screen and (min-width: 1201px) {

.hide-on-extra-large-only {

display: none !important;

}

}

@media only screen and (min-width: 1201px) {

.show-on-extra-large {

display: block !important;

}

}

@media only screen and (min-width: 993px) {

.show-on-large {

display: block !important;

}

}

@media only screen and (min-width: 600px) and (max-width: 992px) {

.show-on-medium {

display: block !important;

}

}

@media only screen and (max-width: 600px) {

.show-on-small {

display: block !important;

}

}

@media only screen and (min-width: 601px) {

.show-on-medium-and-up {

display: block !important;

}

}

@media only screen and (max-width: 992px) {

.show-on-medium-and-down {

display: block !important;

}

}

@media only screen and (max-width: 600px) {

.center-on-small-only {

text-align: center;

}

}

.page-footer {

padding-top: 20px;

color: #fff;

background-color: #ee6e73;

}

.page-footer .footer-copyright {

overflow: hidden;

min-height: 50px;

display: -webkit-box;

display: -webkit-flex;

display: -ms-flexbox;

display: flex;

-webkit-box-align: center;

-webkit-align-items: center;

-ms-flex-align: center;

align-items: center;

-webkit-box-pack: justify;

-webkit-justify-content: space-between;

-ms-flex-pack: justify;

justify-content: space-between;

padding: 10px 0px;

color: rgba(255, 255, 255, 0.8);

background-color: rgba(51, 51, 51, 0.08);

}

table, th, td {

border: none;

}

table {

width: 100%;

display: table;

border-collapse: collapse;

border-spacing: 0;

}

table.striped tr {

border-bottom: none;

}

table.striped > tbody > tr:nth-child(odd) {

background-color: rgba(242, 242, 242, 0.5);

}

table.striped > tbody > tr > td {

border-radius: 0;

}

table.highlight > tbody > tr {

-webkit-transition: background-color .25s ease;

transition: background-color .25s ease;

}

table.highlight > tbody > tr:hover {

background-color: rgba(242, 242, 242, 0.5);

}

table.centered thead tr th, table.centered tbody tr td {

text-align: center;

}

tr {

border-bottom: 1px solid rgba(0, 0, 0, 0.12);

}

td, th {

padding: 15px 5px;

display: table-cell;

text-align: left;

vertical-align: middle;

border-radius: 2px;

}

@media only screen and (max-width: 992px) {

table.responsive-table {

width: 100%;

border-collapse: collapse;

border-spacing: 0;

display: block;

position: relative;

/\* sort out borders \*/

}

table.responsive-table td:empty:before {

content: '\00a0';

}

table.responsive-table th,

table.responsive-table td {

margin: 0;

vertical-align: top;

}

table.responsive-table th {

text-align: left;

}

table.responsive-table thead {

display: block;

float: left;

}

table.responsive-table thead tr {

display: block;

padding: 0 10px 0 0;

}

table.responsive-table thead tr th::before {

content: "\00a0";

}

table.responsive-table tbody {

display: block;

width: auto;

position: relative;

overflow-x: auto;

white-space: nowrap;

}

table.responsive-table tbody tr {

display: inline-block;

vertical-align: top;

}

table.responsive-table th {

display: block;

text-align: right;

}

table.responsive-table td {

display: block;

min-height: 1.25em;

text-align: left;

}

table.responsive-table tr {

border-bottom: none;

padding: 0 10px;

}

table.responsive-table thead {

border: 0;

border-right: 1px solid rgba(0, 0, 0, 0.12);

}

}

.collection {

margin: 0.5rem 0 1rem 0;

border: 1px solid #e0e0e0;

border-radius: 2px;

overflow: hidden;

position: relative;

}

.collection .collection-item {

background-color: #fff;

line-height: 1.5rem;

padding: 10px 20px;

margin: 0;

border-bottom: 1px solid #e0e0e0;

}

.collection .collection-item.avatar {

min-height: 84px;

padding-left: 72px;

position: relative;

}

.collection .collection-item.avatar:not(.circle-clipper) > .circle,

.collection .collection-item.avatar :not(.circle-clipper) > .circle {

position: absolute;

width: 42px;

height: 42px;

overflow: hidden;

left: 15px;

display: inline-block;

vertical-align: middle;

}

.collection .collection-item.avatar i.circle {

font-size: 18px;

line-height: 42px;

color: #fff;

background-color: #999;

text-align: center;

}

.collection .collection-item.avatar .title {

font-size: 16px;

}

.collection .collection-item.avatar p {

margin: 0;

}

.collection .collection-item.avatar .secondary-content {

position: absolute;

top: 16px;

right: 16px;

}

.collection .collection-item:last-child {

border-bottom: none;

}

.collection .collection-item.active {

background-color: #26a69a;

color: #eafaf9;

}

.collection .collection-item.active .secondary-content {

color: #fff;

}

.collection a.collection-item {

display: block;

-webkit-transition: .25s;

transition: .25s;

color: #26a69a;

}

.collection a.collection-item:not(.active):hover {

background-color: #ddd;

}

.collection.with-header .collection-header {

background-color: #fff;

border-bottom: 1px solid #e0e0e0;

padding: 10px 20px;

}

.collection.with-header .collection-item {

padding-left: 30px;

}

.collection.with-header .collection-item.avatar {

padding-left: 72px;

}

.secondary-content {

float: right;

color: #26a69a;

}

.collapsible .collection {

margin: 0;

border: none;

}

.video-container {

position: relative;

padding-bottom: 56.25%;

height: 0;

overflow: hidden;

}

.video-container iframe, .video-container object, .video-container embed {

position: absolute;

top: 0;

left: 0;

width: 100%;

height: 100%;

}

.progress {

position: relative;

height: 4px;

display: block;

width: 100%;

background-color: #acece6;

border-radius: 2px;

margin: 0.5rem 0 1rem 0;

overflow: hidden;

}

.progress .determinate {

position: absolute;

top: 0;

left: 0;

bottom: 0;

background-color: #26a69a;

-webkit-transition: width .3s linear;

transition: width .3s linear;

}

.progress .indeterminate {

background-color: #26a69a;

}

.progress .indeterminate:before {

content: '';

position: absolute;

background-color: inherit;

top: 0;

left: 0;

bottom: 0;

will-change: left, right;

-webkit-animation: indeterminate 2.1s cubic-bezier(0.65, 0.815, 0.735, 0.395) infinite;

animation: indeterminate 2.1s cubic-bezier(0.65, 0.815, 0.735, 0.395) infinite;

}

.progress .indeterminate:after {

content: '';

position: absolute;

background-color: inherit;

top: 0;

left: 0;

bottom: 0;

will-change: left, right;

-webkit-animation: indeterminate-short 2.1s cubic-bezier(0.165, 0.84, 0.44, 1) infinite;

animation: indeterminate-short 2.1s cubic-bezier(0.165, 0.84, 0.44, 1) infinite;

-webkit-animation-delay: 1.15s;

animation-delay: 1.15s;

}

@-webkit-keyframes indeterminate {

0% {

left: -35%;

right: 100%;

}

60% {

left: 100%;

right: -90%;

}

100% {

left: 100%;

right: -90%;

}

}

@keyframes indeterminate {

0% {

left: -35%;

right: 100%;

}

60% {

left: 100%;

right: -90%;

}

100% {

left: 100%;

right: -90%;

}

}

@-webkit-keyframes indeterminate-short {

0% {

left: -200%;

right: 100%;

}

60% {

left: 107%;

right: -8%;

}

100% {

left: 107%;

right: -8%;

}

}

@keyframes indeterminate-short {

0% {

left: -200%;

right: 100%;

}

60% {

left: 107%;

right: -8%;

}

100% {

left: 107%;

right: -8%;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Utility Classes

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

.hide {

display: none !important;

}

.left-align {

text-align: left;

}

.right-align {

text-align: right;

}

.center, .center-align {

text-align: center;

}

.left {

float: left !important;

}

.right {

float: right !important;

}

.no-select, input[type=range],

input[type=range] + .thumb {

-webkit-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

user-select: none;

}

.circle {

border-radius: 50%;

}

.center-block {

display: block;

margin-left: auto;

margin-right: auto;

}

.truncate {

display: block;

white-space: nowrap;

overflow: hidden;

text-overflow: ellipsis;

}

.no-padding {

padding: 0 !important;

}

span.badge {

min-width: 3rem;

padding: 0 6px;

margin-left: 14px;

text-align: center;

font-size: 1rem;

line-height: 22px;

height: 22px;

color: #757575;

float: right;

-webkit-box-sizing: border-box;

box-sizing: border-box;

}

span.badge.new {

font-weight: 300;

font-size: 0.8rem;

color: #fff;

background-color: #26a69a;

border-radius: 2px;

}

span.badge.new:after {

content: " new";

}

span.badge[data-badge-caption]::after {

content: " " attr(data-badge-caption);

}

nav ul a span.badge {

display: inline-block;

float: none;

margin-left: 4px;

line-height: 22px;

height: 22px;

-webkit-font-smoothing: auto;

}

.collection-item span.badge {

margin-top: calc(0.75rem - 11px);

}

.collapsible span.badge {

margin-left: auto;

}

.sidenav span.badge {

margin-top: calc(24px - 11px);

}

table span.badge {

display: inline-block;

float: none;

margin-left: auto;

}

/\* This is needed for some mobile phones to display the Google Icon font properly \*/

.material-icons {

text-rendering: optimizeLegibility;

-webkit-font-feature-settings: 'liga';

-moz-font-feature-settings: 'liga';

font-feature-settings: 'liga';

}

.container {

margin: 0 auto;

max-width: 1280px;

width: 90%;

}

@media only screen and (min-width: 601px) {

.container {

width: 85%;

}

}

@media only screen and (min-width: 993px) {

.container {

width: 70%;

}

}

.col .row {

margin-left: -0.75rem;

margin-right: -0.75rem;

}

.section {

padding-top: 1rem;

padding-bottom: 1rem;

}

.section.no-pad {

padding: 0;

}

.section.no-pad-bot {

padding-bottom: 0;

}

.section.no-pad-top {

padding-top: 0;

}

.row {

margin-left: auto;

margin-right: auto;

margin-bottom: 20px;

}

.row:after {

content: "";

display: table;

clear: both;

}

.row .col {

float: left;

-webkit-box-sizing: border-box;

box-sizing: border-box;

padding: 0 0.75rem;

min-height: 1px;

}

.row .col[class\*="push-"], .row .col[class\*="pull-"] {

position: relative;

}

.row .col.s1 {

width: 8.3333333333%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.s2 {

width: 16.6666666667%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.s3 {

width: 25%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.s4 {

width: 33.3333333333%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.s5 {

width: 41.6666666667%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.s6 {

width: 50%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.s7 {

width: 58.3333333333%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.s8 {

width: 66.6666666667%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.s9 {

width: 75%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.s10 {

width: 83.3333333333%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.s11 {

width: 91.6666666667%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.s12 {

width: 100%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.offset-s1 {

margin-left: 8.3333333333%;

}

.row .col.pull-s1 {

right: 8.3333333333%;

}

.row .col.push-s1 {

left: 8.3333333333%;

}

.row .col.offset-s2 {

margin-left: 16.6666666667%;

}

.row .col.pull-s2 {

right: 16.6666666667%;

}

.row .col.push-s2 {

left: 16.6666666667%;

}

.row .col.offset-s3 {

margin-left: 25%;

}

.row .col.pull-s3 {

right: 25%;

}

.row .col.push-s3 {

left: 25%;

}

.row .col.offset-s4 {

margin-left: 33.3333333333%;

}

.row .col.pull-s4 {

right: 33.3333333333%;

}

.row .col.push-s4 {

left: 33.3333333333%;

}

.row .col.offset-s5 {

margin-left: 41.6666666667%;

}

.row .col.pull-s5 {

right: 41.6666666667%;

}

.row .col.push-s5 {

left: 41.6666666667%;

}

.row .col.offset-s6 {

margin-left: 50%;

}

.row .col.pull-s6 {

right: 50%;

}

.row .col.push-s6 {

left: 50%;

}

.row .col.offset-s7 {

margin-left: 58.3333333333%;

}

.row .col.pull-s7 {

right: 58.3333333333%;

}

.row .col.push-s7 {

left: 58.3333333333%;

}

.row .col.offset-s8 {

margin-left: 66.6666666667%;

}

.row .col.pull-s8 {

right: 66.6666666667%;

}

.row .col.push-s8 {

left: 66.6666666667%;

}

.row .col.offset-s9 {

margin-left: 75%;

}

.row .col.pull-s9 {

right: 75%;

}

.row .col.push-s9 {

left: 75%;

}

.row .col.offset-s10 {

margin-left: 83.3333333333%;

}

.row .col.pull-s10 {

right: 83.3333333333%;

}

.row .col.push-s10 {

left: 83.3333333333%;

}

.row .col.offset-s11 {

margin-left: 91.6666666667%;

}

.row .col.pull-s11 {

right: 91.6666666667%;

}

.row .col.push-s11 {

left: 91.6666666667%;

}

.row .col.offset-s12 {

margin-left: 100%;

}

.row .col.pull-s12 {

right: 100%;

}

.row .col.push-s12 {

left: 100%;

}

@media only screen and (min-width: 601px) {

.row .col.m1 {

width: 8.3333333333%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.m2 {

width: 16.6666666667%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.m3 {

width: 25%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.m4 {

width: 33.3333333333%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.m5 {

width: 41.6666666667%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.m6 {

width: 50%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.m7 {

width: 58.3333333333%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.m8 {

width: 66.6666666667%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.m9 {

width: 75%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.m10 {

width: 83.3333333333%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.m11 {

width: 91.6666666667%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.m12 {

width: 100%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.offset-m1 {

margin-left: 8.3333333333%;

}

.row .col.pull-m1 {

right: 8.3333333333%;

}

.row .col.push-m1 {

left: 8.3333333333%;

}

.row .col.offset-m2 {

margin-left: 16.6666666667%;

}

.row .col.pull-m2 {

right: 16.6666666667%;

}

.row .col.push-m2 {

left: 16.6666666667%;

}

.row .col.offset-m3 {

margin-left: 25%;

}

.row .col.pull-m3 {

right: 25%;

}

.row .col.push-m3 {

left: 25%;

}

.row .col.offset-m4 {

margin-left: 33.3333333333%;

}

.row .col.pull-m4 {

right: 33.3333333333%;

}

.row .col.push-m4 {

left: 33.3333333333%;

}

.row .col.offset-m5 {

margin-left: 41.6666666667%;

}

.row .col.pull-m5 {

right: 41.6666666667%;

}

.row .col.push-m5 {

left: 41.6666666667%;

}

.row .col.offset-m6 {

margin-left: 50%;

}

.row .col.pull-m6 {

right: 50%;

}

.row .col.push-m6 {

left: 50%;

}

.row .col.offset-m7 {

margin-left: 58.3333333333%;

}

.row .col.pull-m7 {

right: 58.3333333333%;

}

.row .col.push-m7 {

left: 58.3333333333%;

}

.row .col.offset-m8 {

margin-left: 66.6666666667%;

}

.row .col.pull-m8 {

right: 66.6666666667%;

}

.row .col.push-m8 {

left: 66.6666666667%;

}

.row .col.offset-m9 {

margin-left: 75%;

}

.row .col.pull-m9 {

right: 75%;

}

.row .col.push-m9 {

left: 75%;

}

.row .col.offset-m10 {

margin-left: 83.3333333333%;

}

.row .col.pull-m10 {

right: 83.3333333333%;

}

.row .col.push-m10 {

left: 83.3333333333%;

}

.row .col.offset-m11 {

margin-left: 91.6666666667%;

}

.row .col.pull-m11 {

right: 91.6666666667%;

}

.row .col.push-m11 {

left: 91.6666666667%;

}

.row .col.offset-m12 {

margin-left: 100%;

}

.row .col.pull-m12 {

right: 100%;

}

.row .col.push-m12 {

left: 100%;

}

}

@media only screen and (min-width: 993px) {

.row .col.l1 {

width: 8.3333333333%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.l2 {

width: 16.6666666667%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.l3 {

width: 25%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.l4 {

width: 33.3333333333%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.l5 {

width: 41.6666666667%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.l6 {

width: 50%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.l7 {

width: 58.3333333333%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.l8 {

width: 66.6666666667%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.l9 {

width: 75%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.l10 {

width: 83.3333333333%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.l11 {

width: 91.6666666667%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.l12 {

width: 100%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.offset-l1 {

margin-left: 8.3333333333%;

}

.row .col.pull-l1 {

right: 8.3333333333%;

}

.row .col.push-l1 {

left: 8.3333333333%;

}

.row .col.offset-l2 {

margin-left: 16.6666666667%;

}

.row .col.pull-l2 {

right: 16.6666666667%;

}

.row .col.push-l2 {

left: 16.6666666667%;

}

.row .col.offset-l3 {

margin-left: 25%;

}

.row .col.pull-l3 {

right: 25%;

}

.row .col.push-l3 {

left: 25%;

}

.row .col.offset-l4 {

margin-left: 33.3333333333%;

}

.row .col.pull-l4 {

right: 33.3333333333%;

}

.row .col.push-l4 {

left: 33.3333333333%;

}

.row .col.offset-l5 {

margin-left: 41.6666666667%;

}

.row .col.pull-l5 {

right: 41.6666666667%;

}

.row .col.push-l5 {

left: 41.6666666667%;

}

.row .col.offset-l6 {

margin-left: 50%;

}

.row .col.pull-l6 {

right: 50%;

}

.row .col.push-l6 {

left: 50%;

}

.row .col.offset-l7 {

margin-left: 58.3333333333%;

}

.row .col.pull-l7 {

right: 58.3333333333%;

}

.row .col.push-l7 {

left: 58.3333333333%;

}

.row .col.offset-l8 {

margin-left: 66.6666666667%;

}

.row .col.pull-l8 {

right: 66.6666666667%;

}

.row .col.push-l8 {

left: 66.6666666667%;

}

.row .col.offset-l9 {

margin-left: 75%;

}

.row .col.pull-l9 {

right: 75%;

}

.row .col.push-l9 {

left: 75%;

}

.row .col.offset-l10 {

margin-left: 83.3333333333%;

}

.row .col.pull-l10 {

right: 83.3333333333%;

}

.row .col.push-l10 {

left: 83.3333333333%;

}

.row .col.offset-l11 {

margin-left: 91.6666666667%;

}

.row .col.pull-l11 {

right: 91.6666666667%;

}

.row .col.push-l11 {

left: 91.6666666667%;

}

.row .col.offset-l12 {

margin-left: 100%;

}

.row .col.pull-l12 {

right: 100%;

}

.row .col.push-l12 {

left: 100%;

}

}

@media only screen and (min-width: 1201px) {

.row .col.xl1 {

width: 8.3333333333%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.xl2 {

width: 16.6666666667%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.xl3 {

width: 25%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.xl4 {

width: 33.3333333333%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.xl5 {

width: 41.6666666667%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.xl6 {

width: 50%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.xl7 {

width: 58.3333333333%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.xl8 {

width: 66.6666666667%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.xl9 {

width: 75%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.xl10 {

width: 83.3333333333%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.xl11 {

width: 91.6666666667%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.xl12 {

width: 100%;

margin-left: auto;

left: auto;

right: auto;

}

.row .col.offset-xl1 {

margin-left: 8.3333333333%;

}

.row .col.pull-xl1 {

right: 8.3333333333%;

}

.row .col.push-xl1 {

left: 8.3333333333%;

}

.row .col.offset-xl2 {

margin-left: 16.6666666667%;

}

.row .col.pull-xl2 {

right: 16.6666666667%;

}

.row .col.push-xl2 {

left: 16.6666666667%;

}

.row .col.offset-xl3 {

margin-left: 25%;

}

.row .col.pull-xl3 {

right: 25%;

}

.row .col.push-xl3 {

left: 25%;

}

.row .col.offset-xl4 {

margin-left: 33.3333333333%;

}

.row .col.pull-xl4 {

right: 33.3333333333%;

}

.row .col.push-xl4 {

left: 33.3333333333%;

}

.row .col.offset-xl5 {

margin-left: 41.6666666667%;

}

.row .col.pull-xl5 {

right: 41.6666666667%;

}

.row .col.push-xl5 {

left: 41.6666666667%;

}

.row .col.offset-xl6 {

margin-left: 50%;

}

.row .col.pull-xl6 {

right: 50%;

}

.row .col.push-xl6 {

left: 50%;

}

.row .col.offset-xl7 {

margin-left: 58.3333333333%;

}

.row .col.pull-xl7 {

right: 58.3333333333%;

}

.row .col.push-xl7 {

left: 58.3333333333%;

}

.row .col.offset-xl8 {

margin-left: 66.6666666667%;

}

.row .col.pull-xl8 {

right: 66.6666666667%;

}

.row .col.push-xl8 {

left: 66.6666666667%;

}

.row .col.offset-xl9 {

margin-left: 75%;

}

.row .col.pull-xl9 {

right: 75%;

}

.row .col.push-xl9 {

left: 75%;

}

.row .col.offset-xl10 {

margin-left: 83.3333333333%;

}

.row .col.pull-xl10 {

right: 83.3333333333%;

}

.row .col.push-xl10 {

left: 83.3333333333%;

}

.row .col.offset-xl11 {

margin-left: 91.6666666667%;

}

.row .col.pull-xl11 {

right: 91.6666666667%;

}

.row .col.push-xl11 {

left: 91.6666666667%;

}

.row .col.offset-xl12 {

margin-left: 100%;

}

.row .col.pull-xl12 {

right: 100%;

}

.row .col.push-xl12 {

left: 100%;

}

}

nav {

color: #fff;

background-color: #ee6e73;

width: 100%;

height: 56px;

line-height: 56px;

}

nav.nav-extended {

height: auto;

}

nav.nav-extended .nav-wrapper {

min-height: 56px;

height: auto;

}

nav.nav-extended .nav-content {

position: relative;

line-height: normal;

}

nav a {

color: #fff;

}

nav i,

nav [class^="mdi-"], nav [class\*="mdi-"],

nav i.material-icons {

display: block;

font-size: 24px;

height: 56px;

line-height: 56px;

}

nav .nav-wrapper {

position: relative;

height: 100%;

}

@media only screen and (min-width: 993px) {

nav a.sidenav-trigger {

display: none;

}

}

nav .sidenav-trigger {

float: left;

position: relative;

z-index: 1;

height: 56px;

margin: 0 18px;

}

nav .sidenav-trigger i {

height: 56px;

line-height: 56px;

}

nav .brand-logo {

position: absolute;

color: #fff;

display: inline-block;

font-size: 2.1rem;

padding: 0;

}

nav .brand-logo.center {

left: 50%;

-webkit-transform: translateX(-50%);

transform: translateX(-50%);

}

@media only screen and (max-width: 992px) {

nav .brand-logo {

left: 50%;

-webkit-transform: translateX(-50%);

transform: translateX(-50%);

}

nav .brand-logo.left, nav .brand-logo.right {

padding: 0;

-webkit-transform: none;

transform: none;

}

nav .brand-logo.left {

left: 0.5rem;

}

nav .brand-logo.right {

right: 0.5rem;

left: auto;

}

}

nav .brand-logo.right {

right: 0.5rem;

padding: 0;

}

nav .brand-logo i,

nav .brand-logo [class^="mdi-"], nav .brand-logo [class\*="mdi-"],

nav .brand-logo i.material-icons {

float: left;

margin-right: 15px;

}

nav .nav-title {

display: inline-block;

font-size: 32px;

padding: 28px 0;

}

nav ul {

margin: 0;

}

nav ul li {

-webkit-transition: background-color .3s;

transition: background-color .3s;

float: left;

padding: 0;

}

nav ul li.active {

background-color: rgba(0, 0, 0, 0.1);

}

nav ul a {

-webkit-transition: background-color .3s;

transition: background-color .3s;

font-size: 1rem;

color: #fff;

display: block;

padding: 0 15px;

cursor: pointer;

}

nav ul a.btn, nav ul a.btn-large, nav ul a.btn-small, nav ul a.btn-large, nav ul a.btn-flat, nav ul a.btn-floating {

margin-top: -2px;

margin-left: 15px;

margin-right: 15px;

}

nav ul a.btn > .material-icons, nav ul a.btn-large > .material-icons, nav ul a.btn-small > .material-icons, nav ul a.btn-large > .material-icons, nav ul a.btn-flat > .material-icons, nav ul a.btn-floating > .material-icons {

height: inherit;

line-height: inherit;

}

nav ul a:hover {

background-color: rgba(0, 0, 0, 0.1);

}

nav ul.left {

float: left;

}

nav form {

height: 100%;

}

nav .input-field {

margin: 0;

height: 100%;

}

nav .input-field input {

height: 100%;

font-size: 1.2rem;

border: none;

padding-left: 2rem;

}

nav .input-field input:focus, nav .input-field input[type=text]:valid, nav .input-field input[type=password]:valid, nav .input-field input[type=email]:valid, nav .input-field input[type=url]:valid, nav .input-field input[type=date]:valid {

border: none;

-webkit-box-shadow: none;

box-shadow: none;

}

nav .input-field label {

top: 0;

left: 0;

}

nav .input-field label i {

color: rgba(255, 255, 255, 0.7);

-webkit-transition: color .3s;

transition: color .3s;

}

nav .input-field label.active i {

color: #fff;

}

.navbar-fixed {

position: relative;

height: 56px;

z-index: 997;

}

.navbar-fixed nav {

position: fixed;

}

@media only screen and (min-width: 601px) {

nav.nav-extended .nav-wrapper {

min-height: 64px;

}

nav, nav .nav-wrapper i, nav a.sidenav-trigger, nav a.sidenav-trigger i {

height: 64px;

line-height: 64px;

}

.navbar-fixed {

height: 64px;

}

}

a {

text-decoration: none;

}

html {

line-height: 1.5;

font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, Oxygen-Sans, Ubuntu, Cantarell, "Helvetica Neue", sans-serif;

font-weight: normal;

color: rgba(0, 0, 0, 0.87);

}

@media only screen and (min-width: 0) {

html {

font-size: 14px;

}

}

@media only screen and (min-width: 992px) {

html {

font-size: 14.5px;

}

}

@media only screen and (min-width: 1200px) {

html {

font-size: 15px;

}

}

h1, h2, h3, h4, h5, h6 {

font-weight: 400;

line-height: 1.3;

}

h1 a, h2 a, h3 a, h4 a, h5 a, h6 a {

font-weight: inherit;

}

h1 {

font-size: 4.2rem;

line-height: 110%;

margin: 2.8rem 0 1.68rem 0;

}

h2 {

font-size: 3.56rem;

line-height: 110%;

margin: 2.3733333333rem 0 1.424rem 0;

}

h3 {

font-size: 2.92rem;

line-height: 110%;

margin: 1.9466666667rem 0 1.168rem 0;

}

h4 {

font-size: 2.28rem;

line-height: 110%;

margin: 1.52rem 0 0.912rem 0;

}

h5 {

font-size: 1.64rem;

line-height: 110%;

margin: 1.0933333333rem 0 0.656rem 0;

}

h6 {

font-size: 1.15rem;

line-height: 110%;

margin: 0.7666666667rem 0 0.46rem 0;

}

em {

font-style: italic;

}

strong {

font-weight: 500;

}

small {

font-size: 75%;

}

.light {

font-weight: 300;

}

.thin {

font-weight: 200;

}

@media only screen and (min-width: 360px) {

.flow-text {

font-size: 1.2rem;

}

}

@media only screen and (min-width: 390px) {

.flow-text {

font-size: 1.224rem;

}

}

@media only screen and (min-width: 420px) {

.flow-text {

font-size: 1.248rem;

}

}

@media only screen and (min-width: 450px) {

.flow-text {

font-size: 1.272rem;

}

}

@media only screen and (min-width: 480px) {

.flow-text {

font-size: 1.296rem;

}

}

@media only screen and (min-width: 510px) {

.flow-text {

font-size: 1.32rem;

}

}

@media only screen and (min-width: 540px) {

.flow-text {

font-size: 1.344rem;

}

}

@media only screen and (min-width: 570px) {

.flow-text {

font-size: 1.368rem;

}

}

@media only screen and (min-width: 600px) {

.flow-text {

font-size: 1.392rem;

}

}

@media only screen and (min-width: 630px) {

.flow-text {

font-size: 1.416rem;

}

}

@media only screen and (min-width: 660px) {

.flow-text {

font-size: 1.44rem;

}

}

@media only screen and (min-width: 690px) {

.flow-text {

font-size: 1.464rem;

}

}

@media only screen and (min-width: 720px) {

.flow-text {

font-size: 1.488rem;

}

}

@media only screen and (min-width: 750px) {

.flow-text {

font-size: 1.512rem;

}

}

@media only screen and (min-width: 780px) {

.flow-text {

font-size: 1.536rem;

}

}

@media only screen and (min-width: 810px) {

.flow-text {

font-size: 1.56rem;

}

}

@media only screen and (min-width: 840px) {

.flow-text {

font-size: 1.584rem;

}

}

@media only screen and (min-width: 870px) {

.flow-text {

font-size: 1.608rem;

}

}

@media only screen and (min-width: 900px) {

.flow-text {

font-size: 1.632rem;

}

}

@media only screen and (min-width: 930px) {

.flow-text {

font-size: 1.656rem;

}

}

@media only screen and (min-width: 960px) {

.flow-text {

font-size: 1.68rem;

}

}

@media only screen and (max-width: 360px) {

.flow-text {

font-size: 1.2rem;

}

}

.scale-transition {

-webkit-transition: -webkit-transform 0.3s cubic-bezier(0.53, 0.01, 0.36, 1.63) !important;

transition: -webkit-transform 0.3s cubic-bezier(0.53, 0.01, 0.36, 1.63) !important;

transition: transform 0.3s cubic-bezier(0.53, 0.01, 0.36, 1.63) !important;

transition: transform 0.3s cubic-bezier(0.53, 0.01, 0.36, 1.63), -webkit-transform 0.3s cubic-bezier(0.53, 0.01, 0.36, 1.63) !important;

}

.scale-transition.scale-out {

-webkit-transform: scale(0);

transform: scale(0);

-webkit-transition: -webkit-transform .2s !important;

transition: -webkit-transform .2s !important;

transition: transform .2s !important;

transition: transform .2s, -webkit-transform .2s !important;

}

.scale-transition.scale-in {

-webkit-transform: scale(1);

transform: scale(1);

}

.card-panel {

-webkit-transition: -webkit-box-shadow .25s;

transition: -webkit-box-shadow .25s;

transition: box-shadow .25s;

transition: box-shadow .25s, -webkit-box-shadow .25s;

padding: 24px;

margin: 0.5rem 0 1rem 0;

border-radius: 2px;

background-color: #fff;

}

.card {

position: relative;

margin: 0.5rem 0 1rem 0;

background-color: #fff;

-webkit-transition: -webkit-box-shadow .25s;

transition: -webkit-box-shadow .25s;

transition: box-shadow .25s;

transition: box-shadow .25s, -webkit-box-shadow .25s;

border-radius: 2px;

}

.card .card-title {

font-size: 24px;

font-weight: 300;

}

.card .card-title.activator {

cursor: pointer;

}

.card.small, .card.medium, .card.large {

position: relative;

}

.card.small .card-image, .card.medium .card-image, .card.large .card-image {

max-height: 60%;

overflow: hidden;

}

.card.small .card-image + .card-content, .card.medium .card-image + .card-content, .card.large .card-image + .card-content {

max-height: 40%;

}

.card.small .card-content, .card.medium .card-content, .card.large .card-content {

max-height: 100%;

overflow: hidden;

}

.card.small .card-action, .card.medium .card-action, .card.large .card-action {

position: absolute;

bottom: 0;

left: 0;

right: 0;

}

.card.small {

height: 300px;

}

.card.medium {

height: 400px;

}

.card.large {

height: 500px;

}

.card.horizontal {

display: -webkit-box;

display: -webkit-flex;

display: -ms-flexbox;

display: flex;

}

.card.horizontal.small .card-image, .card.horizontal.medium .card-image, .card.horizontal.large .card-image {

height: 100%;

max-height: none;

overflow: visible;

}

.card.horizontal.small .card-image img, .card.horizontal.medium .card-image img, .card.horizontal.large .card-image img {

height: 100%;

}

.card.horizontal .card-image {

max-width: 50%;

}

.card.horizontal .card-image img {

border-radius: 2px 0 0 2px;

max-width: 100%;

width: auto;

}

.card.horizontal .card-stacked {

display: -webkit-box;

display: -webkit-flex;

display: -ms-flexbox;

display: flex;

-webkit-box-orient: vertical;

-webkit-box-direction: normal;

-webkit-flex-direction: column;

-ms-flex-direction: column;

flex-direction: column;

-webkit-box-flex: 1;

-webkit-flex: 1;

-ms-flex: 1;

flex: 1;

position: relative;

}

.card.horizontal .card-stacked .card-content {

-webkit-box-flex: 1;

-webkit-flex-grow: 1;

-ms-flex-positive: 1;

flex-grow: 1;

}

.card.sticky-action .card-action {

z-index: 2;

}

.card.sticky-action .card-reveal {

z-index: 1;

padding-bottom: 64px;

}

.card .card-image {

position: relative;

}

.card .card-image img {

display: block;

border-radius: 2px 2px 0 0;

position: relative;

left: 0;

right: 0;

top: 0;

bottom: 0;

width: 100%;

}

.card .card-image .card-title {

color: #fff;

position: absolute;

bottom: 0;

left: 0;

max-width: 100%;

padding: 24px;

}

.card .card-content {

padding: 24px;

border-radius: 0 0 2px 2px;

}

.card .card-content p {

margin: 0;

}

.card .card-content .card-title {

display: block;

line-height: 32px;

margin-bottom: 8px;

}

.card .card-content .card-title i {

line-height: 32px;

}

.card .card-action {

background-color: inherit;

border-top: 1px solid rgba(160, 160, 160, 0.2);

position: relative;

padding: 16px 24px;

}

.card .card-action:last-child {

border-radius: 0 0 2px 2px;

}

.card .card-action a:not(.btn):not(.btn-large):not(.btn-small):not(.btn-large):not(.btn-floating) {

color: #ffab40;

margin-right: 24px;

-webkit-transition: color .3s ease;

transition: color .3s ease;

text-transform: uppercase;

}

.card .card-action a:not(.btn):not(.btn-large):not(.btn-small):not(.btn-large):not(.btn-floating):hover {

color: #ffd8a6;

}

.card .card-reveal {

padding: 24px;

position: absolute;

background-color: #fff;

width: 100%;

overflow-y: auto;

left: 0;

top: 100%;

height: 100%;

z-index: 3;

display: none;

}

.card .card-reveal .card-title {

cursor: pointer;

display: block;

}

#toast-container {

display: block;

position: fixed;

z-index: 10000;

}

@media only screen and (max-width: 600px) {

#toast-container {

min-width: 100%;

bottom: 0%;

}

}

@media only screen and (min-width: 601px) and (max-width: 992px) {

#toast-container {

left: 5%;

bottom: 7%;

max-width: 90%;

}

}

@media only screen and (min-width: 993px) {

#toast-container {

top: 10%;

right: 7%;

max-width: 86%;

}

}

.toast {

border-radius: 2px;

top: 35px;

width: auto;

margin-top: 10px;

position: relative;

max-width: 100%;

height: auto;

min-height: 48px;

line-height: 1.5em;

background-color: #323232;

padding: 10px 25px;

font-size: 1.1rem;

font-weight: 300;

color: #fff;

display: -webkit-box;

display: -webkit-flex;

display: -ms-flexbox;

display: flex;

-webkit-box-align: center;

-webkit-align-items: center;

-ms-flex-align: center;

align-items: center;

-webkit-box-pack: justify;

-webkit-justify-content: space-between;

-ms-flex-pack: justify;

justify-content: space-between;

cursor: default;

}

.toast .toast-action {

color: #eeff41;

font-weight: 500;

margin-right: -25px;

margin-left: 3rem;

}

.toast.rounded {

border-radius: 24px;

}

@media only screen and (max-width: 600px) {

.toast {

width: 100%;

border-radius: 0;

}

}

.tabs {

position: relative;

overflow-x: auto;

overflow-y: hidden;

height: 48px;

width: 100%;

background-color: #fff;

margin: 0 auto;

white-space: nowrap;

}

.tabs.tabs-transparent {

background-color: transparent;

}

.tabs.tabs-transparent .tab a,

.tabs.tabs-transparent .tab.disabled a,

.tabs.tabs-transparent .tab.disabled a:hover {

color: rgba(255, 255, 255, 0.7);

}

.tabs.tabs-transparent .tab a:hover,

.tabs.tabs-transparent .tab a.active {

color: #fff;

}

.tabs.tabs-transparent .indicator {

background-color: #fff;

}

.tabs.tabs-fixed-width {

display: -webkit-box;

display: -webkit-flex;

display: -ms-flexbox;

display: flex;

}

.tabs.tabs-fixed-width .tab {

-webkit-box-flex: 1;

-webkit-flex-grow: 1;

-ms-flex-positive: 1;

flex-grow: 1;

}

.tabs .tab {

display: inline-block;

text-align: center;

line-height: 48px;

height: 48px;

padding: 0;

margin: 0;

text-transform: uppercase;

}

.tabs .tab a {

color: rgba(238, 110, 115, 0.7);

display: block;

width: 100%;

height: 100%;

padding: 0 24px;

font-size: 14px;

text-overflow: ellipsis;

overflow: hidden;

-webkit-transition: color .28s ease, background-color .28s ease;

transition: color .28s ease, background-color .28s ease;

}

.tabs .tab a:focus, .tabs .tab a:focus.active {

background-color: rgba(246, 178, 181, 0.2);

outline: none;

}

.tabs .tab a:hover, .tabs .tab a.active {

background-color: transparent;

color: #ee6e73;

}

.tabs .tab.disabled a,

.tabs .tab.disabled a:hover {

color: rgba(238, 110, 115, 0.4);

cursor: default;

}

.tabs .indicator {

position: absolute;

bottom: 0;

height: 2px;

background-color: #f6b2b5;

will-change: left, right;

}

@media only screen and (max-width: 992px) {

.tabs {

display: -webkit-box;

display: -webkit-flex;

display: -ms-flexbox;

display: flex;

}

.tabs .tab {

-webkit-box-flex: 1;

-webkit-flex-grow: 1;

-ms-flex-positive: 1;

flex-grow: 1;

}

.tabs .tab a {

padding: 0 12px;

}

}

.material-tooltip {

padding: 10px 8px;

font-size: 1rem;

z-index: 2000;

background-color: transparent;

border-radius: 2px;

color: #fff;

min-height: 36px;

line-height: 120%;

opacity: 0;

position: absolute;

text-align: center;

max-width: calc(100% - 4px);

overflow: hidden;

left: 0;

top: 0;

pointer-events: none;

visibility: hidden;

background-color: #323232;

}

.backdrop {

position: absolute;

opacity: 0;

height: 7px;

width: 14px;

border-radius: 0 0 50% 50%;

background-color: #323232;

z-index: -1;

-webkit-transform-origin: 50% 0%;

transform-origin: 50% 0%;

visibility: hidden;

}

.btn, .btn-large, .btn-small,

.btn-flat {

border: none;

border-radius: 2px;

display: inline-block;

height: 36px;

line-height: 36px;

padding: 0 16px;

text-transform: uppercase;

vertical-align: middle;

-webkit-tap-highlight-color: transparent;

}

.btn.disabled, .disabled.btn-large, .disabled.btn-small,

.btn-floating.disabled,

.btn-large.disabled,

.btn-small.disabled,

.btn-flat.disabled,

.btn:disabled,

.btn-large:disabled,

.btn-small:disabled,

.btn-floating:disabled,

.btn-large:disabled,

.btn-small:disabled,

.btn-flat:disabled,

.btn[disabled],

.btn-large[disabled],

.btn-small[disabled],

.btn-floating[disabled],

.btn-large[disabled],

.btn-small[disabled],

.btn-flat[disabled] {

pointer-events: none;

background-color: #DFDFDF !important;

-webkit-box-shadow: none;

box-shadow: none;

color: #9F9F9F !important;

cursor: default;

}

.btn.disabled:hover, .disabled.btn-large:hover, .disabled.btn-small:hover,

.btn-floating.disabled:hover,

.btn-large.disabled:hover,

.btn-small.disabled:hover,

.btn-flat.disabled:hover,

.btn:disabled:hover,

.btn-large:disabled:hover,

.btn-small:disabled:hover,

.btn-floating:disabled:hover,

.btn-large:disabled:hover,

.btn-small:disabled:hover,

.btn-flat:disabled:hover,

.btn[disabled]:hover,

.btn-large[disabled]:hover,

.btn-small[disabled]:hover,

.btn-floating[disabled]:hover,

.btn-large[disabled]:hover,

.btn-small[disabled]:hover,

.btn-flat[disabled]:hover {

background-color: #DFDFDF !important;

color: #9F9F9F !important;

}

.btn, .btn-large, .btn-small,

.btn-floating,

.btn-large,

.btn-small,

.btn-flat {

font-size: 14px;

outline: 0;

}

.btn i, .btn-large i, .btn-small i,

.btn-floating i,

.btn-large i,

.btn-small i,

.btn-flat i {

font-size: 1.3rem;

line-height: inherit;

}

.btn:focus, .btn-large:focus, .btn-small:focus,

.btn-floating:focus {

background-color: #1d7d74;

}

.btn, .btn-large, .btn-small {

text-decoration: none;

color: #fff;

background-color: #26a69a;

text-align: center;

letter-spacing: .5px;

-webkit-transition: background-color .2s ease-out;

transition: background-color .2s ease-out;

cursor: pointer;

}

.btn:hover, .btn-large:hover, .btn-small:hover {

background-color: #2bbbad;

}

.btn-floating {

display: inline-block;

color: #fff;

position: relative;

overflow: hidden;

z-index: 1;

width: 40px;

height: 40px;

line-height: 40px;

padding: 0;

background-color: #26a69a;

border-radius: 50%;

-webkit-transition: background-color .3s;

transition: background-color .3s;

cursor: pointer;

vertical-align: middle;

}

.btn-floating:hover {

background-color: #26a69a;

}

.btn-floating:before {

border-radius: 0;

}

.btn-floating.btn-large {

width: 56px;

height: 56px;

padding: 0;

}

.btn-floating.btn-large.halfway-fab {

bottom: -28px;

}

.btn-floating.btn-large i {

line-height: 56px;

}

.btn-floating.btn-small {

width: 32.4px;

height: 32.4px;

}

.btn-floating.btn-small.halfway-fab {

bottom: -16.2px;

}

.btn-floating.btn-small i {

line-height: 32.4px;

}

.btn-floating.halfway-fab {

position: absolute;

right: 24px;

bottom: -20px;

}

.btn-floating.halfway-fab.left {

right: auto;

left: 24px;

}

.btn-floating i {

width: inherit;

display: inline-block;

text-align: center;

color: #fff;

font-size: 1.6rem;

line-height: 40px;

}

button.btn-floating {

border: none;

}

.fixed-action-btn {

position: fixed;

right: 23px;

bottom: 23px;

padding-top: 15px;

margin-bottom: 0;

z-index: 997;

}

.fixed-action-btn.active ul {

visibility: visible;

}

.fixed-action-btn.direction-left, .fixed-action-btn.direction-right {

padding: 0 0 0 15px;

}

.fixed-action-btn.direction-left ul, .fixed-action-btn.direction-right ul {

text-align: right;

right: 64px;

top: 50%;

-webkit-transform: translateY(-50%);

transform: translateY(-50%);

height: 100%;

left: auto;

/\*width 100% only goes to width of button container \*/

width: 500px;

}

.fixed-action-btn.direction-left ul li, .fixed-action-btn.direction-right ul li {

display: inline-block;

margin: 7.5px 15px 0 0;

}

.fixed-action-btn.direction-right {

padding: 0 15px 0 0;

}

.fixed-action-btn.direction-right ul {

text-align: left;

direction: rtl;

left: 64px;

right: auto;

}

.fixed-action-btn.direction-right ul li {

margin: 7.5px 0 0 15px;

}

.fixed-action-btn.direction-bottom {

padding: 0 0 15px 0;

}

.fixed-action-btn.direction-bottom ul {

top: 64px;

bottom: auto;

display: -webkit-box;

display: -webkit-flex;

display: -ms-flexbox;

display: flex;

-webkit-box-orient: vertical;

-webkit-box-direction: reverse;

-webkit-flex-direction: column-reverse;

-ms-flex-direction: column-reverse;

flex-direction: column-reverse;

}

.fixed-action-btn.direction-bottom ul li {

margin: 15px 0 0 0;

}

.fixed-action-btn.toolbar {

padding: 0;

height: 56px;

}

.fixed-action-btn.toolbar.active > a i {

opacity: 0;

}

.fixed-action-btn.toolbar ul {

display: -webkit-box;

display: -webkit-flex;

display: -ms-flexbox;

display: flex;

top: 0;

bottom: 0;

z-index: 1;

}

.fixed-action-btn.toolbar ul li {

-webkit-box-flex: 1;

-webkit-flex: 1;

-ms-flex: 1;

flex: 1;

display: inline-block;

margin: 0;

height: 100%;

-webkit-transition: none;

transition: none;

}

.fixed-action-btn.toolbar ul li a {

display: block;

overflow: hidden;

position: relative;

width: 100%;

height: 100%;

background-color: transparent;

-webkit-box-shadow: none;

box-shadow: none;

color: #fff;

line-height: 56px;

z-index: 1;

}

.fixed-action-btn.toolbar ul li a i {

line-height: inherit;

}

.fixed-action-btn ul {

left: 0;

right: 0;

text-align: center;

position: absolute;

bottom: 64px;

margin: 0;

visibility: hidden;

}

.fixed-action-btn ul li {

margin-bottom: 15px;

}

.fixed-action-btn ul a.btn-floating {

opacity: 0;

}

.fixed-action-btn .fab-backdrop {

position: absolute;

top: 0;

left: 0;

z-index: -1;

width: 40px;

height: 40px;

background-color: #26a69a;

border-radius: 50%;

-webkit-transform: scale(0);

transform: scale(0);

}

.btn-flat {

-webkit-box-shadow: none;

box-shadow: none;

background-color: transparent;

color: #343434;

cursor: pointer;

-webkit-transition: background-color .2s;

transition: background-color .2s;

}

.btn-flat:focus, .btn-flat:hover {

-webkit-box-shadow: none;

box-shadow: none;

}

.btn-flat:focus {

background-color: rgba(0, 0, 0, 0.1);

}

.btn-flat.disabled, .btn-flat.btn-flat[disabled] {

background-color: transparent !important;

color: #b3b2b2 !important;

cursor: default;

}

.btn-large {

height: 54px;

line-height: 54px;

font-size: 15px;

padding: 0 28px;

}

.btn-large i {

font-size: 1.6rem;

}

.btn-small {

height: 32.4px;

line-height: 32.4px;

font-size: 13px;

}

.btn-small i {

font-size: 1.2rem;

}

.btn-block {

display: block;

}

.dropdown-content {

background-color: #fff;

margin: 0;

display: none;

min-width: 100px;

overflow-y: auto;

opacity: 0;

position: absolute;

left: 0;

top: 0;

z-index: 9999;

-webkit-transform-origin: 0 0;

transform-origin: 0 0;

}

.dropdown-content:focus {

outline: 0;

}

.dropdown-content li {

clear: both;

color: rgba(0, 0, 0, 0.87);

cursor: pointer;

min-height: 50px;

line-height: 1.5rem;

width: 100%;

text-align: left;

}

.dropdown-content li:hover, .dropdown-content li.active {

background-color: #eee;

}

.dropdown-content li:focus {

outline: none;

}

.dropdown-content li.divider {

min-height: 0;

height: 1px;

}

.dropdown-content li > a, .dropdown-content li > span {

font-size: 16px;

color: #26a69a;

display: block;

line-height: 22px;

padding: 14px 16px;

}

.dropdown-content li > span > label {

top: 1px;

left: 0;

height: 18px;

}

.dropdown-content li > a > i {

height: inherit;

line-height: inherit;

float: left;

margin: 0 24px 0 0;

width: 24px;

}

body.keyboard-focused .dropdown-content li:focus {

background-color: #dadada;

}

.input-field.col .dropdown-content [type="checkbox"] + label {

top: 1px;

left: 0;

height: 18px;

-webkit-transform: none;

transform: none;

}

.dropdown-trigger {

cursor: pointer;

}

.waves-effect {

position: relative;

cursor: pointer;

display: inline-block;

overflow: hidden;

-webkit-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

user-select: none;

-webkit-tap-highlight-color: transparent;

vertical-align: middle;

z-index: 1;

-webkit-transition: .3s ease-out;

transition: .3s ease-out;

}

.waves-effect .waves-ripple {

position: absolute;

border-radius: 50%;

width: 20px;

height: 20px;

margin-top: -10px;

margin-left: -10px;

opacity: 0;

background: rgba(0, 0, 0, 0.2);

-webkit-transition: all 0.7s ease-out;

transition: all 0.7s ease-out;

-webkit-transition-property: opacity, -webkit-transform;

transition-property: opacity, -webkit-transform;

transition-property: transform, opacity;

transition-property: transform, opacity, -webkit-transform;

-webkit-transform: scale(0);

transform: scale(0);

pointer-events: none;

}

.waves-effect.waves-light .waves-ripple {

background-color: rgba(255, 255, 255, 0.45);

}

.waves-effect.waves-red .waves-ripple {

background-color: rgba(244, 67, 54, 0.7);

}

.waves-effect.waves-yellow .waves-ripple {

background-color: rgba(255, 235, 59, 0.7);

}

.waves-effect.waves-orange .waves-ripple {

background-color: rgba(255, 152, 0, 0.7);

}

.waves-effect.waves-purple .waves-ripple {

background-color: rgba(156, 39, 176, 0.7);

}

.waves-effect.waves-green .waves-ripple {

background-color: rgba(76, 175, 80, 0.7);

}

.waves-effect.waves-teal .waves-ripple {

background-color: rgba(0, 150, 136, 0.7);

}

.waves-effect input[type="button"], .waves-effect input[type="reset"], .waves-effect input[type="submit"] {

border: 0;

font-style: normal;

font-size: inherit;

text-transform: inherit;

background: none;

}

.waves-effect img {

position: relative;

z-index: -1;

}

.waves-notransition {

-webkit-transition: none !important;

transition: none !important;

}

.waves-circle {

-webkit-transform: translateZ(0);

transform: translateZ(0);

-webkit-mask-image: -webkit-radial-gradient(circle, white 100%, black 100%);

}

.waves-input-wrapper {

border-radius: 0.2em;

vertical-align: bottom;

}

.waves-input-wrapper .waves-button-input {

position: relative;

top: 0;

left: 0;

z-index: 1;

}

.waves-circle {

text-align: center;

width: 2.5em;

height: 2.5em;

line-height: 2.5em;

border-radius: 50%;

-webkit-mask-image: none;

}

.waves-block {

display: block;

}

/\* Firefox Bug: link not triggered \*/

.waves-effect .waves-ripple {

z-index: -1;

}

.modal {

display: none;

position: fixed;

left: 0;

right: 0;

background-color: #fafafa;

padding: 0;

max-height: 70%;

width: 55%;

margin: auto;

overflow-y: auto;

border-radius: 2px;

will-change: top, opacity;

}

.modal:focus {

outline: none;

}

@media only screen and (max-width: 992px) {

.modal {

width: 80%;

}

}

.modal h1, .modal h2, .modal h3, .modal h4 {

margin-top: 0;

}

.modal .modal-content {

padding: 24px;

}

.modal .modal-close {

cursor: pointer;

}

.modal .modal-footer {

border-radius: 0 0 2px 2px;

background-color: #fafafa;

padding: 4px 6px;

height: 56px;

width: 100%;

text-align: right;

}

.modal .modal-footer .btn, .modal .modal-footer .btn-large, .modal .modal-footer .btn-small, .modal .modal-footer .btn-flat {

margin: 6px 0;

}

.modal-overlay {

position: fixed;

z-index: 999;

top: -25%;

left: 0;

bottom: 0;

right: 0;

height: 125%;

width: 100%;

background: #000;

display: none;

will-change: opacity;

}

.modal.modal-fixed-footer {

padding: 0;

height: 70%;

}

.modal.modal-fixed-footer .modal-content {

position: absolute;

height: calc(100% - 56px);

max-height: 100%;

width: 100%;

overflow-y: auto;

}

.modal.modal-fixed-footer .modal-footer {

border-top: 1px solid rgba(0, 0, 0, 0.1);

position: absolute;

bottom: 0;

}

.modal.bottom-sheet {

top: auto;

bottom: -100%;

margin: 0;

width: 100%;

max-height: 45%;

border-radius: 0;

will-change: bottom, opacity;

}

.collapsible {

border-top: 1px solid #ddd;

border-right: 1px solid #ddd;

border-left: 1px solid #ddd;

margin: 0.5rem 0 1rem 0;

}

.collapsible-header {

display: -webkit-box;

display: -webkit-flex;

display: -ms-flexbox;

display: flex;

cursor: pointer;

-webkit-tap-highlight-color: transparent;

line-height: 1.5;

padding: 1rem;

background-color: #fff;

border-bottom: 1px solid #ddd;

}

.collapsible-header:focus {

outline: 0;

}

.collapsible-header i {

width: 2rem;

font-size: 1.6rem;

display: inline-block;

text-align: center;

margin-right: 1rem;

}

.keyboard-focused .collapsible-header:focus {

background-color: #eee;

}

.collapsible-body {

display: none;

border-bottom: 1px solid #ddd;

-webkit-box-sizing: border-box;

box-sizing: border-box;

padding: 2rem;

}

.sidenav .collapsible,

.sidenav.fixed .collapsible {

border: none;

-webkit-box-shadow: none;

box-shadow: none;

}

.sidenav .collapsible li,

.sidenav.fixed .collapsible li {

padding: 0;

}

.sidenav .collapsible-header,

.sidenav.fixed .collapsible-header {

background-color: transparent;

border: none;

line-height: inherit;

height: inherit;

padding: 0 16px;

}

.sidenav .collapsible-header:hover,

.sidenav.fixed .collapsible-header:hover {

background-color: rgba(0, 0, 0, 0.05);

}

.sidenav .collapsible-header i,

.sidenav.fixed .collapsible-header i {

line-height: inherit;

}

.sidenav .collapsible-body,

.sidenav.fixed .collapsible-body {

border: 0;

background-color: #fff;

}

.sidenav .collapsible-body li a,

.sidenav.fixed .collapsible-body li a {

padding: 0 23.5px 0 31px;

}

.collapsible.popout {

border: none;

-webkit-box-shadow: none;

box-shadow: none;

}

.collapsible.popout > li {

-webkit-box-shadow: 0 2px 5px 0 rgba(0, 0, 0, 0.16), 0 2px 10px 0 rgba(0, 0, 0, 0.12);

box-shadow: 0 2px 5px 0 rgba(0, 0, 0, 0.16), 0 2px 10px 0 rgba(0, 0, 0, 0.12);

margin: 0 24px;

-webkit-transition: margin 0.35s cubic-bezier(0.25, 0.46, 0.45, 0.94);

transition: margin 0.35s cubic-bezier(0.25, 0.46, 0.45, 0.94);

}

.collapsible.popout > li.active {

-webkit-box-shadow: 0 5px 11px 0 rgba(0, 0, 0, 0.18), 0 4px 15px 0 rgba(0, 0, 0, 0.15);

box-shadow: 0 5px 11px 0 rgba(0, 0, 0, 0.18), 0 4px 15px 0 rgba(0, 0, 0, 0.15);

margin: 16px 0;

}

.chip {

display: inline-block;

height: 32px;

font-size: 13px;

font-weight: 500;

color: rgba(0, 0, 0, 0.6);

line-height: 32px;

padding: 0 12px;

border-radius: 16px;

background-color: #e4e4e4;

margin-bottom: 5px;

margin-right: 5px;

}

.chip:focus {

outline: none;

background-color: #26a69a;

color: #fff;

}

.chip > img {

float: left;

margin: 0 8px 0 -12px;

height: 32px;

width: 32px;

border-radius: 50%;

}

.chip .close {

cursor: pointer;

float: right;

font-size: 16px;

line-height: 32px;

padding-left: 8px;

}

.chips {

border: none;

border-bottom: 1px solid #9e9e9e;

-webkit-box-shadow: none;

box-shadow: none;

margin: 0 0 8px 0;

min-height: 45px;

outline: none;

-webkit-transition: all .3s;

transition: all .3s;

}

.chips.focus {

border-bottom: 1px solid #26a69a;

-webkit-box-shadow: 0 1px 0 0 #26a69a;

box-shadow: 0 1px 0 0 #26a69a;

}

.chips:hover {

cursor: text;

}

.chips .input {

background: none;

border: 0;

color: rgba(0, 0, 0, 0.6);

display: inline-block;

font-size: 16px;

height: 3rem;

line-height: 32px;

outline: 0;

margin: 0;

padding: 0 !important;

width: 120px !important;

}

.chips .input:focus {

border: 0 !important;

-webkit-box-shadow: none !important;

box-shadow: none !important;

}

.chips .autocomplete-content {

margin-top: 0;

margin-bottom: 0;

}

.prefix ~ .chips {

margin-left: 3rem;

width: 92%;

width: calc(100% - 3rem);

}

.chips:empty ~ label {

font-size: 0.8rem;

-webkit-transform: translateY(-140%);

transform: translateY(-140%);

}

.materialboxed {

display: block;

cursor: -webkit-zoom-in;

cursor: zoom-in;

position: relative;

-webkit-transition: opacity .4s;

transition: opacity .4s;

-webkit-backface-visibility: hidden;

}

.materialboxed:hover:not(.active) {

opacity: .8;

}

.materialboxed.active {

cursor: -webkit-zoom-out;

cursor: zoom-out;

}

#materialbox-overlay {

position: fixed;

top: 0;

right: 0;

bottom: 0;

left: 0;

background-color: #292929;

z-index: 1000;

will-change: opacity;

}

.materialbox-caption {

position: fixed;

display: none;

color: #fff;

line-height: 50px;

bottom: 0;

left: 0;

width: 100%;

text-align: center;

padding: 0% 15%;

height: 50px;

z-index: 1000;

-webkit-font-smoothing: antialiased;

}

select:focus {

outline: 1px solid #c9f3ef;

}

button:focus {

outline: none;

background-color: #2ab7a9;

}

label {

font-size: 0.8rem;

color: #9e9e9e;

}

/\* Text Inputs + Textarea

========================================================================== \*/

/\* Style Placeholders \*/

::-webkit-input-placeholder {

color: #d1d1d1;

}

::-moz-placeholder {

color: #d1d1d1;

}

:-ms-input-placeholder {

color: #d1d1d1;

}

::-ms-input-placeholder {

color: #d1d1d1;

}

::placeholder {

color: #d1d1d1;

}

/\* Text inputs \*/

input:not([type]),

input[type=text]:not(.browser-default),

input[type=password]:not(.browser-default),

input[type=email]:not(.browser-default),

input[type=url]:not(.browser-default),

input[type=time]:not(.browser-default),

input[type=date]:not(.browser-default),

input[type=datetime]:not(.browser-default),

input[type=datetime-local]:not(.browser-default),

input[type=tel]:not(.browser-default),

input[type=number]:not(.browser-default),

input[type=search]:not(.browser-default),

textarea.materialize-textarea {

background-color: transparent;

border: none;

border-bottom: 1px solid #9e9e9e;

border-radius: 0;

outline: none;

height: 3rem;

width: 100%;

font-size: 16px;

margin: 0 0 8px 0;

padding: 0;

-webkit-box-shadow: none;

box-shadow: none;

-webkit-box-sizing: content-box;

box-sizing: content-box;

-webkit-transition: border .3s, -webkit-box-shadow .3s;

transition: border .3s, -webkit-box-shadow .3s;

transition: box-shadow .3s, border .3s;

transition: box-shadow .3s, border .3s, -webkit-box-shadow .3s;

}

input:not([type]):disabled, input:not([type])[readonly="readonly"],

input[type=text]:not(.browser-default):disabled,

input[type=text]:not(.browser-default)[readonly="readonly"],

input[type=password]:not(.browser-default):disabled,

input[type=password]:not(.browser-default)[readonly="readonly"],

input[type=email]:not(.browser-default):disabled,

input[type=email]:not(.browser-default)[readonly="readonly"],

input[type=url]:not(.browser-default):disabled,

input[type=url]:not(.browser-default)[readonly="readonly"],

input[type=time]:not(.browser-default):disabled,

input[type=time]:not(.browser-default)[readonly="readonly"],

input[type=date]:not(.browser-default):disabled,

input[type=date]:not(.browser-default)[readonly="readonly"],

input[type=datetime]:not(.browser-default):disabled,

input[type=datetime]:not(.browser-default)[readonly="readonly"],

input[type=datetime-local]:not(.browser-default):disabled,

input[type=datetime-local]:not(.browser-default)[readonly="readonly"],

input[type=tel]:not(.browser-default):disabled,

input[type=tel]:not(.browser-default)[readonly="readonly"],

input[type=number]:not(.browser-default):disabled,

input[type=number]:not(.browser-default)[readonly="readonly"],

input[type=search]:not(.browser-default):disabled,

input[type=search]:not(.browser-default)[readonly="readonly"],

textarea.materialize-textarea:disabled,

textarea.materialize-textarea[readonly="readonly"] {

color: rgba(0, 0, 0, 0.42);

border-bottom: 1px dotted rgba(0, 0, 0, 0.42);

}

input:not([type]):disabled + label,

input:not([type])[readonly="readonly"] + label,

input[type=text]:not(.browser-default):disabled + label,

input[type=text]:not(.browser-default)[readonly="readonly"] + label,

input[type=password]:not(.browser-default):disabled + label,

input[type=password]:not(.browser-default)[readonly="readonly"] + label,

input[type=email]:not(.browser-default):disabled + label,

input[type=email]:not(.browser-default)[readonly="readonly"] + label,

input[type=url]:not(.browser-default):disabled + label,

input[type=url]:not(.browser-default)[readonly="readonly"] + label,

input[type=time]:not(.browser-default):disabled + label,

input[type=time]:not(.browser-default)[readonly="readonly"] + label,

input[type=date]:not(.browser-default):disabled + label,

input[type=date]:not(.browser-default)[readonly="readonly"] + label,

input[type=datetime]:not(.browser-default):disabled + label,

input[type=datetime]:not(.browser-default)[readonly="readonly"] + label,

input[type=datetime-local]:not(.browser-default):disabled + label,

input[type=datetime-local]:not(.browser-default)[readonly="readonly"] + label,

input[type=tel]:not(.browser-default):disabled + label,

input[type=tel]:not(.browser-default)[readonly="readonly"] + label,

input[type=number]:not(.browser-default):disabled + label,

input[type=number]:not(.browser-default)[readonly="readonly"] + label,

input[type=search]:not(.browser-default):disabled + label,

input[type=search]:not(.browser-default)[readonly="readonly"] + label,

textarea.materialize-textarea:disabled + label,

textarea.materialize-textarea[readonly="readonly"] + label {

color: rgba(0, 0, 0, 0.42);

}

input:not([type]):focus:not([readonly]),

input[type=text]:not(.browser-default):focus:not([readonly]),

input[type=password]:not(.browser-default):focus:not([readonly]),

input[type=email]:not(.browser-default):focus:not([readonly]),

input[type=url]:not(.browser-default):focus:not([readonly]),

input[type=time]:not(.browser-default):focus:not([readonly]),

input[type=date]:not(.browser-default):focus:not([readonly]),

input[type=datetime]:not(.browser-default):focus:not([readonly]),

input[type=datetime-local]:not(.browser-default):focus:not([readonly]),

input[type=tel]:not(.browser-default):focus:not([readonly]),

input[type=number]:not(.browser-default):focus:not([readonly]),

input[type=search]:not(.browser-default):focus:not([readonly]),

textarea.materialize-textarea:focus:not([readonly]) {

border-bottom: 1px solid #26a69a;

-webkit-box-shadow: 0 1px 0 0 #26a69a;

box-shadow: 0 1px 0 0 #26a69a;

}

input:not([type]):focus:not([readonly]) + label,

input[type=text]:not(.browser-default):focus:not([readonly]) + label,

input[type=password]:not(.browser-default):focus:not([readonly]) + label,

input[type=email]:not(.browser-default):focus:not([readonly]) + label,

input[type=url]:not(.browser-default):focus:not([readonly]) + label,

input[type=time]:not(.browser-default):focus:not([readonly]) + label,

input[type=date]:not(.browser-default):focus:not([readonly]) + label,

input[type=datetime]:not(.browser-default):focus:not([readonly]) + label,

input[type=datetime-local]:not(.browser-default):focus:not([readonly]) + label,

input[type=tel]:not(.browser-default):focus:not([readonly]) + label,

input[type=number]:not(.browser-default):focus:not([readonly]) + label,

input[type=search]:not(.browser-default):focus:not([readonly]) + label,

textarea.materialize-textarea:focus:not([readonly]) + label {

color: #26a69a;

}

input:not([type]):focus.valid ~ label,

input[type=text]:not(.browser-default):focus.valid ~ label,

input[type=password]:not(.browser-default):focus.valid ~ label,

input[type=email]:not(.browser-default):focus.valid ~ label,

input[type=url]:not(.browser-default):focus.valid ~ label,

input[type=time]:not(.browser-default):focus.valid ~ label,

input[type=date]:not(.browser-default):focus.valid ~ label,

input[type=datetime]:not(.browser-default):focus.valid ~ label,

input[type=datetime-local]:not(.browser-default):focus.valid ~ label,

input[type=tel]:not(.browser-default):focus.valid ~ label,

input[type=number]:not(.browser-default):focus.valid ~ label,

input[type=search]:not(.browser-default):focus.valid ~ label,

textarea.materialize-textarea:focus.valid ~ label {

color: #4CAF50;

}

input:not([type]):focus.invalid ~ label,

input[type=text]:not(.browser-default):focus.invalid ~ label,

input[type=password]:not(.browser-default):focus.invalid ~ label,

input[type=email]:not(.browser-default):focus.invalid ~ label,

input[type=url]:not(.browser-default):focus.invalid ~ label,

input[type=time]:not(.browser-default):focus.invalid ~ label,

input[type=date]:not(.browser-default):focus.invalid ~ label,

input[type=datetime]:not(.browser-default):focus.invalid ~ label,

input[type=datetime-local]:not(.browser-default):focus.invalid ~ label,

input[type=tel]:not(.browser-default):focus.invalid ~ label,

input[type=number]:not(.browser-default):focus.invalid ~ label,

input[type=search]:not(.browser-default):focus.invalid ~ label,

textarea.materialize-textarea:focus.invalid ~ label {

color: #F44336;

}

input:not([type]).validate + label,

input[type=text]:not(.browser-default).validate + label,

input[type=password]:not(.browser-default).validate + label,

input[type=email]:not(.browser-default).validate + label,

input[type=url]:not(.browser-default).validate + label,

input[type=time]:not(.browser-default).validate + label,

input[type=date]:not(.browser-default).validate + label,

input[type=datetime]:not(.browser-default).validate + label,

input[type=datetime-local]:not(.browser-default).validate + label,

input[type=tel]:not(.browser-default).validate + label,

input[type=number]:not(.browser-default).validate + label,

input[type=search]:not(.browser-default).validate + label,

textarea.materialize-textarea.validate + label {

width: 100%;

}

/\* Validation Sass Placeholders \*/

input.valid:not([type]), input.valid:not([type]):focus,

input.valid[type=text]:not(.browser-default),

input.valid[type=text]:not(.browser-default):focus,

input.valid[type=password]:not(.browser-default),

input.valid[type=password]:not(.browser-default):focus,

input.valid[type=email]:not(.browser-default),

input.valid[type=email]:not(.browser-default):focus,

input.valid[type=url]:not(.browser-default),

input.valid[type=url]:not(.browser-default):focus,

input.valid[type=time]:not(.browser-default),

input.valid[type=time]:not(.browser-default):focus,

input.valid[type=date]:not(.browser-default),

input.valid[type=date]:not(.browser-default):focus,

input.valid[type=datetime]:not(.browser-default),

input.valid[type=datetime]:not(.browser-default):focus,

input.valid[type=datetime-local]:not(.browser-default),

input.valid[type=datetime-local]:not(.browser-default):focus,

input.valid[type=tel]:not(.browser-default),

input.valid[type=tel]:not(.browser-default):focus,

input.valid[type=number]:not(.browser-default),

input.valid[type=number]:not(.browser-default):focus,

input.valid[type=search]:not(.browser-default),

input.valid[type=search]:not(.browser-default):focus,

textarea.materialize-textarea.valid,

textarea.materialize-textarea.valid:focus, .select-wrapper.valid > input.select-dropdown {

border-bottom: 1px solid #4CAF50;

-webkit-box-shadow: 0 1px 0 0 #4CAF50;

box-shadow: 0 1px 0 0 #4CAF50;

}

input.invalid:not([type]), input.invalid:not([type]):focus,

input.invalid[type=text]:not(.browser-default),

input.invalid[type=text]:not(.browser-default):focus,

input.invalid[type=password]:not(.browser-default),

input.invalid[type=password]:not(.browser-default):focus,

input.invalid[type=email]:not(.browser-default),

input.invalid[type=email]:not(.browser-default):focus,

input.invalid[type=url]:not(.browser-default),

input.invalid[type=url]:not(.browser-default):focus,

input.invalid[type=time]:not(.browser-default),

input.invalid[type=time]:not(.browser-default):focus,

input.invalid[type=date]:not(.browser-default),

input.invalid[type=date]:not(.browser-default):focus,

input.invalid[type=datetime]:not(.browser-default),

input.invalid[type=datetime]:not(.browser-default):focus,

input.invalid[type=datetime-local]:not(.browser-default),

input.invalid[type=datetime-local]:not(.browser-default):focus,

input.invalid[type=tel]:not(.browser-default),

input.invalid[type=tel]:not(.browser-default):focus,

input.invalid[type=number]:not(.browser-default),

input.invalid[type=number]:not(.browser-default):focus,

input.invalid[type=search]:not(.browser-default),

input.invalid[type=search]:not(.browser-default):focus,

textarea.materialize-textarea.invalid,

textarea.materialize-textarea.invalid:focus, .select-wrapper.invalid > input.select-dropdown,

.select-wrapper.invalid > input.select-dropdown:focus {

border-bottom: 1px solid #F44336;

-webkit-box-shadow: 0 1px 0 0 #F44336;

box-shadow: 0 1px 0 0 #F44336;

}

input:not([type]).valid ~ .helper-text[data-success],

input:not([type]):focus.valid ~ .helper-text[data-success],

input:not([type]).invalid ~ .helper-text[data-error],

input:not([type]):focus.invalid ~ .helper-text[data-error],

input[type=text]:not(.browser-default).valid ~ .helper-text[data-success],

input[type=text]:not(.browser-default):focus.valid ~ .helper-text[data-success],

input[type=text]:not(.browser-default).invalid ~ .helper-text[data-error],

input[type=text]:not(.browser-default):focus.invalid ~ .helper-text[data-error],

input[type=password]:not(.browser-default).valid ~ .helper-text[data-success],

input[type=password]:not(.browser-default):focus.valid ~ .helper-text[data-success],

input[type=password]:not(.browser-default).invalid ~ .helper-text[data-error],

input[type=password]:not(.browser-default):focus.invalid ~ .helper-text[data-error],

input[type=email]:not(.browser-default).valid ~ .helper-text[data-success],

input[type=email]:not(.browser-default):focus.valid ~ .helper-text[data-success],

input[type=email]:not(.browser-default).invalid ~ .helper-text[data-error],

input[type=email]:not(.browser-default):focus.invalid ~ .helper-text[data-error],

input[type=url]:not(.browser-default).valid ~ .helper-text[data-success],

input[type=url]:not(.browser-default):focus.valid ~ .helper-text[data-success],

input[type=url]:not(.browser-default).invalid ~ .helper-text[data-error],

input[type=url]:not(.browser-default):focus.invalid ~ .helper-text[data-error],

input[type=time]:not(.browser-default).valid ~ .helper-text[data-success],

input[type=time]:not(.browser-default):focus.valid ~ .helper-text[data-success],

input[type=time]:not(.browser-default).invalid ~ .helper-text[data-error],

input[type=time]:not(.browser-default):focus.invalid ~ .helper-text[data-error],

input[type=date]:not(.browser-default).valid ~ .helper-text[data-success],

input[type=date]:not(.browser-default):focus.valid ~ .helper-text[data-success],

input[type=date]:not(.browser-default).invalid ~ .helper-text[data-error],

input[type=date]:not(.browser-default):focus.invalid ~ .helper-text[data-error],

input[type=datetime]:not(.browser-default).valid ~ .helper-text[data-success],

input[type=datetime]:not(.browser-default):focus.valid ~ .helper-text[data-success],

input[type=datetime]:not(.browser-default).invalid ~ .helper-text[data-error],

input[type=datetime]:not(.browser-default):focus.invalid ~ .helper-text[data-error],

input[type=datetime-local]:not(.browser-default).valid ~ .helper-text[data-success],

input[type=datetime-local]:not(.browser-default):focus.valid ~ .helper-text[data-success],

input[type=datetime-local]:not(.browser-default).invalid ~ .helper-text[data-error],

input[type=datetime-local]:not(.browser-default):focus.invalid ~ .helper-text[data-error],

input[type=tel]:not(.browser-default).valid ~ .helper-text[data-success],

input[type=tel]:not(.browser-default):focus.valid ~ .helper-text[data-success],

input[type=tel]:not(.browser-default).invalid ~ .helper-text[data-error],

input[type=tel]:not(.browser-default):focus.invalid ~ .helper-text[data-error],

input[type=number]:not(.browser-default).valid ~ .helper-text[data-success],

input[type=number]:not(.browser-default):focus.valid ~ .helper-text[data-success],

input[type=number]:not(.browser-default).invalid ~ .helper-text[data-error],

input[type=number]:not(.browser-default):focus.invalid ~ .helper-text[data-error],

input[type=search]:not(.browser-default).valid ~ .helper-text[data-success],

input[type=search]:not(.browser-default):focus.valid ~ .helper-text[data-success],

input[type=search]:not(.browser-default).invalid ~ .helper-text[data-error],

input[type=search]:not(.browser-default):focus.invalid ~ .helper-text[data-error],

textarea.materialize-textarea.valid ~ .helper-text[data-success],

textarea.materialize-textarea:focus.valid ~ .helper-text[data-success],

textarea.materialize-textarea.invalid ~ .helper-text[data-error],

textarea.materialize-textarea:focus.invalid ~ .helper-text[data-error], .select-wrapper.valid .helper-text[data-success],

.select-wrapper.invalid ~ .helper-text[data-error] {

color: transparent;

-webkit-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

user-select: none;

pointer-events: none;

}

input:not([type]).valid ~ .helper-text:after,

input:not([type]):focus.valid ~ .helper-text:after,

input[type=text]:not(.browser-default).valid ~ .helper-text:after,

input[type=text]:not(.browser-default):focus.valid ~ .helper-text:after,

input[type=password]:not(.browser-default).valid ~ .helper-text:after,

input[type=password]:not(.browser-default):focus.valid ~ .helper-text:after,

input[type=email]:not(.browser-default).valid ~ .helper-text:after,

input[type=email]:not(.browser-default):focus.valid ~ .helper-text:after,

input[type=url]:not(.browser-default).valid ~ .helper-text:after,

input[type=url]:not(.browser-default):focus.valid ~ .helper-text:after,

input[type=time]:not(.browser-default).valid ~ .helper-text:after,

input[type=time]:not(.browser-default):focus.valid ~ .helper-text:after,

input[type=date]:not(.browser-default).valid ~ .helper-text:after,

input[type=date]:not(.browser-default):focus.valid ~ .helper-text:after,

input[type=datetime]:not(.browser-default).valid ~ .helper-text:after,

input[type=datetime]:not(.browser-default):focus.valid ~ .helper-text:after,

input[type=datetime-local]:not(.browser-default).valid ~ .helper-text:after,

input[type=datetime-local]:not(.browser-default):focus.valid ~ .helper-text:after,

input[type=tel]:not(.browser-default).valid ~ .helper-text:after,

input[type=tel]:not(.browser-default):focus.valid ~ .helper-text:after,

input[type=number]:not(.browser-default).valid ~ .helper-text:after,

input[type=number]:not(.browser-default):focus.valid ~ .helper-text:after,

input[type=search]:not(.browser-default).valid ~ .helper-text:after,

input[type=search]:not(.browser-default):focus.valid ~ .helper-text:after,

textarea.materialize-textarea.valid ~ .helper-text:after,

textarea.materialize-textarea:focus.valid ~ .helper-text:after, .select-wrapper.valid ~ .helper-text:after {

content: attr(data-success);

color: #4CAF50;

}

input:not([type]).invalid ~ .helper-text:after,

input:not([type]):focus.invalid ~ .helper-text:after,

input[type=text]:not(.browser-default).invalid ~ .helper-text:after,

input[type=text]:not(.browser-default):focus.invalid ~ .helper-text:after,

input[type=password]:not(.browser-default).invalid ~ .helper-text:after,

input[type=password]:not(.browser-default):focus.invalid ~ .helper-text:after,

input[type=email]:not(.browser-default).invalid ~ .helper-text:after,

input[type=email]:not(.browser-default):focus.invalid ~ .helper-text:after,

input[type=url]:not(.browser-default).invalid ~ .helper-text:after,

input[type=url]:not(.browser-default):focus.invalid ~ .helper-text:after,

input[type=time]:not(.browser-default).invalid ~ .helper-text:after,

input[type=time]:not(.browser-default):focus.invalid ~ .helper-text:after,

input[type=date]:not(.browser-default).invalid ~ .helper-text:after,

input[type=date]:not(.browser-default):focus.invalid ~ .helper-text:after,

input[type=datetime]:not(.browser-default).invalid ~ .helper-text:after,

input[type=datetime]:not(.browser-default):focus.invalid ~ .helper-text:after,

input[type=datetime-local]:not(.browser-default).invalid ~ .helper-text:after,

input[type=datetime-local]:not(.browser-default):focus.invalid ~ .helper-text:after,

input[type=tel]:not(.browser-default).invalid ~ .helper-text:after,

input[type=tel]:not(.browser-default):focus.invalid ~ .helper-text:after,

input[type=number]:not(.browser-default).invalid ~ .helper-text:after,

input[type=number]:not(.browser-default):focus.invalid ~ .helper-text:after,

input[type=search]:not(.browser-default).invalid ~ .helper-text:after,

input[type=search]:not(.browser-default):focus.invalid ~ .helper-text:after,

textarea.materialize-textarea.invalid ~ .helper-text:after,

textarea.materialize-textarea:focus.invalid ~ .helper-text:after, .select-wrapper.invalid ~ .helper-text:after {

content: attr(data-error);

color: #F44336;

}

input:not([type]) + label:after,

input[type=text]:not(.browser-default) + label:after,

input[type=password]:not(.browser-default) + label:after,

input[type=email]:not(.browser-default) + label:after,

input[type=url]:not(.browser-default) + label:after,

input[type=time]:not(.browser-default) + label:after,

input[type=date]:not(.browser-default) + label:after,

input[type=datetime]:not(.browser-default) + label:after,

input[type=datetime-local]:not(.browser-default) + label:after,

input[type=tel]:not(.browser-default) + label:after,

input[type=number]:not(.browser-default) + label:after,

input[type=search]:not(.browser-default) + label:after,

textarea.materialize-textarea + label:after, .select-wrapper + label:after {

display: block;

content: "";

position: absolute;

top: 100%;

left: 0;

opacity: 0;

-webkit-transition: .2s opacity ease-out, .2s color ease-out;

transition: .2s opacity ease-out, .2s color ease-out;

}

.input-field {

position: relative;

margin-top: 1rem;

margin-bottom: 1rem;

}

.input-field.inline {

display: inline-block;

vertical-align: middle;

margin-left: 5px;

}

.input-field.inline input,

.input-field.inline .select-dropdown {

margin-bottom: 1rem;

}

.input-field.col label {

left: 0.75rem;

}

.input-field.col .prefix ~ label,

.input-field.col .prefix ~ .validate ~ label {

width: calc(100% - 3rem - 1.5rem);

}

.input-field > label {

color: #9e9e9e;

position: absolute;

top: 0;

left: 0;

font-size: 1rem;

cursor: text;

-webkit-transition: color .2s ease-out, -webkit-transform .2s ease-out;

transition: color .2s ease-out, -webkit-transform .2s ease-out;

transition: transform .2s ease-out, color .2s ease-out;

transition: transform .2s ease-out, color .2s ease-out, -webkit-transform .2s ease-out;

-webkit-transform-origin: 0% 100%;

transform-origin: 0% 100%;

text-align: initial;

-webkit-transform: translateY(12px);

transform: translateY(12px);

}

.input-field > label:not(.label-icon).active {

-webkit-transform: translateY(-14px) scale(0.8);

transform: translateY(-14px) scale(0.8);

-webkit-transform-origin: 0 0;

transform-origin: 0 0;

}

.input-field > input[type]:-webkit-autofill:not(.browser-default):not([type="search"]) + label,

.input-field > input[type=date]:not(.browser-default) + label,

.input-field > input[type=time]:not(.browser-default) + label {

-webkit-transform: translateY(-14px) scale(0.8);

transform: translateY(-14px) scale(0.8);

-webkit-transform-origin: 0 0;

transform-origin: 0 0;

}

.input-field .helper-text {

position: relative;

min-height: 18px;

display: block;

font-size: 12px;

color: rgba(0, 0, 0, 0.54);

}

.input-field .helper-text::after {

opacity: 1;

position: absolute;

top: 0;

left: 0;

}

.input-field .prefix {

position: absolute;

width: 3rem;

font-size: 2rem;

-webkit-transition: color .2s;

transition: color .2s;

top: 0.5rem;

}

.input-field .prefix.active {

color: #26a69a;

}

.input-field .prefix ~ input,

.input-field .prefix ~ textarea,

.input-field .prefix ~ label,

.input-field .prefix ~ .validate ~ label,

.input-field .prefix ~ .helper-text,

.input-field .prefix ~ .autocomplete-content {

margin-left: 3rem;

width: 92%;

width: calc(100% - 3rem);

}

.input-field .prefix ~ label {

margin-left: 3rem;

}

@media only screen and (max-width: 992px) {

.input-field .prefix ~ input {

width: 86%;

width: calc(100% - 3rem);

}

}

@media only screen and (max-width: 600px) {

.input-field .prefix ~ input {

width: 80%;

width: calc(100% - 3rem);

}

}

/\* Search Field \*/

.input-field input[type=search] {

display: block;

line-height: inherit;

-webkit-transition: .3s background-color;

transition: .3s background-color;

}

.nav-wrapper .input-field input[type=search] {

height: inherit;

padding-left: 4rem;

width: calc(100% - 4rem);

border: 0;

-webkit-box-shadow: none;

box-shadow: none;

}

.input-field input[type=search]:focus:not(.browser-default) {

background-color: #fff;

border: 0;

-webkit-box-shadow: none;

box-shadow: none;

color: #444;

}

.input-field input[type=search]:focus:not(.browser-default) + label i,

.input-field input[type=search]:focus:not(.browser-default) ~ .mdi-navigation-close,

.input-field input[type=search]:focus:not(.browser-default) ~ .material-icons {

color: #444;

}

.input-field input[type=search] + .label-icon {

-webkit-transform: none;

transform: none;

left: 1rem;

}

.input-field input[type=search] ~ .mdi-navigation-close,

.input-field input[type=search] ~ .material-icons {

position: absolute;

top: 0;

right: 1rem;

color: transparent;

cursor: pointer;

font-size: 2rem;

-webkit-transition: .3s color;

transition: .3s color;

}

/\* Textarea \*/

textarea {

width: 100%;

height: 3rem;

background-color: transparent;

}

textarea.materialize-textarea {

line-height: normal;

overflow-y: hidden;

/\* prevents scroll bar flash \*/

padding: .8rem 0 .8rem 0;

/\* prevents text jump on Enter keypress \*/

resize: none;

min-height: 3rem;

-webkit-box-sizing: border-box;

box-sizing: border-box;

}

.hiddendiv {

visibility: hidden;

white-space: pre-wrap;

word-wrap: break-word;

overflow-wrap: break-word;

/\* future version of deprecated 'word-wrap' \*/

padding-top: 1.2rem;

/\* prevents text jump on Enter keypress \*/

position: absolute;

top: 0;

z-index: -1;

}

/\* Autocomplete \*/

.autocomplete-content li .highlight {

color: #444;

}

.autocomplete-content li img {

height: 40px;

width: 40px;

margin: 5px 15px;

}

/\* Character Counter \*/

.character-counter {

min-height: 18px;

}

/\* Radio Buttons

========================================================================== \*/

[type="radio"]:not(:checked),

[type="radio"]:checked {

position: absolute;

opacity: 0;

pointer-events: none;

}

[type="radio"]:not(:checked) + span,

[type="radio"]:checked + span {

position: relative;

padding-left: 35px;

cursor: pointer;

display: inline-block;

height: 25px;

line-height: 25px;

font-size: 1rem;

-webkit-transition: .28s ease;

transition: .28s ease;

-webkit-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

user-select: none;

}

[type="radio"] + span:before,

[type="radio"] + span:after {

content: '';

position: absolute;

left: 0;

top: 0;

margin: 4px;

width: 16px;

height: 16px;

z-index: 0;

-webkit-transition: .28s ease;

transition: .28s ease;

}

/\* Unchecked styles \*/

[type="radio"]:not(:checked) + span:before,

[type="radio"]:not(:checked) + span:after,

[type="radio"]:checked + span:before,

[type="radio"]:checked + span:after,

[type="radio"].with-gap:checked + span:before,

[type="radio"].with-gap:checked + span:after {

border-radius: 50%;

}

[type="radio"]:not(:checked) + span:before,

[type="radio"]:not(:checked) + span:after {

border: 2px solid #5a5a5a;

}

[type="radio"]:not(:checked) + span:after {

-webkit-transform: scale(0);

transform: scale(0);

}

/\* Checked styles \*/

[type="radio"]:checked + span:before {

border: 2px solid transparent;

}

[type="radio"]:checked + span:after,

[type="radio"].with-gap:checked + span:before,

[type="radio"].with-gap:checked + span:after {

border: 2px solid #26a69a;

}

[type="radio"]:checked + span:after,

[type="radio"].with-gap:checked + span:after {

background-color: #26a69a;

}

[type="radio"]:checked + span:after {

-webkit-transform: scale(1.02);

transform: scale(1.02);

}

/\* Radio With gap \*/

[type="radio"].with-gap:checked + span:after {

-webkit-transform: scale(0.5);

transform: scale(0.5);

}

/\* Focused styles \*/

[type="radio"].tabbed:focus + span:before {

-webkit-box-shadow: 0 0 0 10px rgba(0, 0, 0, 0.1);

box-shadow: 0 0 0 10px rgba(0, 0, 0, 0.1);

}

/\* Disabled Radio With gap \*/

[type="radio"].with-gap:disabled:checked + span:before {

border: 2px solid rgba(0, 0, 0, 0.42);

}

[type="radio"].with-gap:disabled:checked + span:after {

border: none;

background-color: rgba(0, 0, 0, 0.42);

}

/\* Disabled style \*/

[type="radio"]:disabled:not(:checked) + span:before,

[type="radio"]:disabled:checked + span:before {

background-color: transparent;

border-color: rgba(0, 0, 0, 0.42);

}

[type="radio"]:disabled + span {

color: rgba(0, 0, 0, 0.42);

}

[type="radio"]:disabled:not(:checked) + span:before {

border-color: rgba(0, 0, 0, 0.42);

}

[type="radio"]:disabled:checked + span:after {

background-color: rgba(0, 0, 0, 0.42);

border-color: #949494;

}

/\* Checkboxes

========================================================================== \*/

/\* Remove default checkbox \*/

[type="checkbox"]:not(:checked),

[type="checkbox"]:checked {

position: absolute;

opacity: 0;

pointer-events: none;

}

[type="checkbox"] {

/\* checkbox aspect \*/

}

[type="checkbox"] + span:not(.lever) {

position: relative;

padding-left: 35px;

cursor: pointer;

display: inline-block;

height: 25px;

line-height: 25px;

font-size: 1rem;

-webkit-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

user-select: none;

}

[type="checkbox"] + span:not(.lever):before,

[type="checkbox"]:not(.filled-in) + span:not(.lever):after {

content: '';

position: absolute;

top: 0;

left: 0;

width: 18px;

height: 18px;

z-index: 0;

border: 2px solid #5a5a5a;

border-radius: 1px;

margin-top: 3px;

-webkit-transition: .2s;

transition: .2s;

}

[type="checkbox"]:not(.filled-in) + span:not(.lever):after {

border: 0;

-webkit-transform: scale(0);

transform: scale(0);

}

[type="checkbox"]:not(:checked):disabled + span:not(.lever):before {

border: none;

background-color: rgba(0, 0, 0, 0.42);

}

[type="checkbox"].tabbed:focus + span:not(.lever):after {

-webkit-transform: scale(1);

transform: scale(1);

border: 0;

border-radius: 50%;

-webkit-box-shadow: 0 0 0 10px rgba(0, 0, 0, 0.1);

box-shadow: 0 0 0 10px rgba(0, 0, 0, 0.1);

background-color: rgba(0, 0, 0, 0.1);

}

[type="checkbox"]:checked + span:not(.lever):before {

top: -4px;

left: -5px;

width: 12px;

height: 22px;

border-top: 2px solid transparent;

border-left: 2px solid transparent;

border-right: 2px solid #26a69a;

border-bottom: 2px solid #26a69a;

-webkit-transform: rotate(40deg);

transform: rotate(40deg);

-webkit-backface-visibility: hidden;

backface-visibility: hidden;

-webkit-transform-origin: 100% 100%;

transform-origin: 100% 100%;

}

[type="checkbox"]:checked:disabled + span:before {

border-right: 2px solid rgba(0, 0, 0, 0.42);

border-bottom: 2px solid rgba(0, 0, 0, 0.42);

}

/\* Indeterminate checkbox \*/

[type="checkbox"]:indeterminate + span:not(.lever):before {

top: -11px;

left: -12px;

width: 10px;

height: 22px;

border-top: none;

border-left: none;

border-right: 2px solid #26a69a;

border-bottom: none;

-webkit-transform: rotate(90deg);

transform: rotate(90deg);

-webkit-backface-visibility: hidden;

backface-visibility: hidden;

-webkit-transform-origin: 100% 100%;

transform-origin: 100% 100%;

}

[type="checkbox"]:indeterminate:disabled + span:not(.lever):before {

border-right: 2px solid rgba(0, 0, 0, 0.42);

background-color: transparent;

}

[type="checkbox"].filled-in + span:not(.lever):after {

border-radius: 2px;

}

[type="checkbox"].filled-in + span:not(.lever):before,

[type="checkbox"].filled-in + span:not(.lever):after {

content: '';

left: 0;

position: absolute;

/\* .1s delay is for check animation \*/

-webkit-transition: border .25s, background-color .25s, width .20s .1s, height .20s .1s, top .20s .1s, left .20s .1s;

transition: border .25s, background-color .25s, width .20s .1s, height .20s .1s, top .20s .1s, left .20s .1s;

z-index: 1;

}

[type="checkbox"].filled-in:not(:checked) + span:not(.lever):before {

width: 0;

height: 0;

border: 3px solid transparent;

left: 6px;

top: 10px;

-webkit-transform: rotateZ(37deg);

transform: rotateZ(37deg);

-webkit-transform-origin: 100% 100%;

transform-origin: 100% 100%;

}

[type="checkbox"].filled-in:not(:checked) + span:not(.lever):after {

height: 20px;

width: 20px;

background-color: transparent;

border: 2px solid #5a5a5a;

top: 0px;

z-index: 0;

}

[type="checkbox"].filled-in:checked + span:not(.lever):before {

top: 0;

left: 1px;

width: 8px;

height: 13px;

border-top: 2px solid transparent;

border-left: 2px solid transparent;

border-right: 2px solid #fff;

border-bottom: 2px solid #fff;

-webkit-transform: rotateZ(37deg);

transform: rotateZ(37deg);

-webkit-transform-origin: 100% 100%;

transform-origin: 100% 100%;

}

[type="checkbox"].filled-in:checked + span:not(.lever):after {

top: 0;

width: 20px;

height: 20px;

border: 2px solid #26a69a;

background-color: #26a69a;

z-index: 0;

}

[type="checkbox"].filled-in.tabbed:focus + span:not(.lever):after {

border-radius: 2px;

border-color: #5a5a5a;

background-color: rgba(0, 0, 0, 0.1);

}

[type="checkbox"].filled-in.tabbed:checked:focus + span:not(.lever):after {

border-radius: 2px;

background-color: #26a69a;

border-color: #26a69a;

}

[type="checkbox"].filled-in:disabled:not(:checked) + span:not(.lever):before {

background-color: transparent;

border: 2px solid transparent;

}

[type="checkbox"].filled-in:disabled:not(:checked) + span:not(.lever):after {

border-color: transparent;

background-color: #949494;

}

[type="checkbox"].filled-in:disabled:checked + span:not(.lever):before {

background-color: transparent;

}

[type="checkbox"].filled-in:disabled:checked + span:not(.lever):after {

background-color: #949494;

border-color: #949494;

}

/\* Switch

========================================================================== \*/

.switch,

.switch \* {

-webkit-tap-highlight-color: transparent;

-webkit-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

user-select: none;

}

.switch label {

cursor: pointer;

}

.switch label input[type=checkbox] {

opacity: 0;

width: 0;

height: 0;

}

.switch label input[type=checkbox]:checked + .lever {

background-color: #84c7c1;

}

.switch label input[type=checkbox]:checked + .lever:before, .switch label input[type=checkbox]:checked + .lever:after {

left: 18px;

}

.switch label input[type=checkbox]:checked + .lever:after {

background-color: #26a69a;

}

.switch label .lever {

content: "";

display: inline-block;

position: relative;

width: 36px;

height: 14px;

background-color: rgba(0, 0, 0, 0.38);

border-radius: 15px;

margin-right: 10px;

-webkit-transition: background 0.3s ease;

transition: background 0.3s ease;

vertical-align: middle;

margin: 0 16px;

}

.switch label .lever:before, .switch label .lever:after {

content: "";

position: absolute;

display: inline-block;

width: 20px;

height: 20px;

border-radius: 50%;

left: 0;

top: -3px;

-webkit-transition: left 0.3s ease, background .3s ease, -webkit-box-shadow 0.1s ease, -webkit-transform .1s ease;

transition: left 0.3s ease, background .3s ease, -webkit-box-shadow 0.1s ease, -webkit-transform .1s ease;

transition: left 0.3s ease, background .3s ease, box-shadow 0.1s ease, transform .1s ease;

transition: left 0.3s ease, background .3s ease, box-shadow 0.1s ease, transform .1s ease, -webkit-box-shadow 0.1s ease, -webkit-transform .1s ease;

}

.switch label .lever:before {

background-color: rgba(38, 166, 154, 0.15);

}

.switch label .lever:after {

background-color: #F1F1F1;

-webkit-box-shadow: 0px 3px 1px -2px rgba(0, 0, 0, 0.2), 0px 2px 2px 0px rgba(0, 0, 0, 0.14), 0px 1px 5px 0px rgba(0, 0, 0, 0.12);

box-shadow: 0px 3px 1px -2px rgba(0, 0, 0, 0.2), 0px 2px 2px 0px rgba(0, 0, 0, 0.14), 0px 1px 5px 0px rgba(0, 0, 0, 0.12);

}

input[type=checkbox]:checked:not(:disabled) ~ .lever:active::before,

input[type=checkbox]:checked:not(:disabled).tabbed:focus ~ .lever::before {

-webkit-transform: scale(2.4);

transform: scale(2.4);

background-color: rgba(38, 166, 154, 0.15);

}

input[type=checkbox]:not(:disabled) ~ .lever:active:before,

input[type=checkbox]:not(:disabled).tabbed:focus ~ .lever::before {

-webkit-transform: scale(2.4);

transform: scale(2.4);

background-color: rgba(0, 0, 0, 0.08);

}

.switch input[type=checkbox][disabled] + .lever {

cursor: default;

background-color: rgba(0, 0, 0, 0.12);

}

.switch label input[type=checkbox][disabled] + .lever:after,

.switch label input[type=checkbox][disabled]:checked + .lever:after {

background-color: #949494;

}

/\* Select Field

========================================================================== \*/

select {

display: none;

}

select.browser-default {

display: block;

}

select {

background-color: rgba(255, 255, 255, 0.9);

width: 100%;

padding: 5px;

border: 1px solid #f2f2f2;

border-radius: 2px;

height: 3rem;

}

.select-label {

position: absolute;

}

.select-wrapper {

position: relative;

}

.select-wrapper.valid + label,

.select-wrapper.invalid + label {

width: 100%;

pointer-events: none;

}

.select-wrapper input.select-dropdown {

position: relative;

cursor: pointer;

background-color: transparent;

border: none;

border-bottom: 1px solid #9e9e9e;

outline: none;

height: 3rem;

line-height: 3rem;

width: 100%;

font-size: 16px;

margin: 0 0 8px 0;

padding: 0;

display: block;

-webkit-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

user-select: none;

z-index: 1;

}

.select-wrapper input.select-dropdown:focus {

border-bottom: 1px solid #26a69a;

}

.select-wrapper .caret {

position: absolute;

right: 0;

top: 0;

bottom: 0;

margin: auto 0;

z-index: 0;

fill: rgba(0, 0, 0, 0.87);

}

.select-wrapper + label {

position: absolute;

top: -26px;

font-size: 0.8rem;

}

select:disabled {

color: rgba(0, 0, 0, 0.42);

}

.select-wrapper.disabled + label {

color: rgba(0, 0, 0, 0.42);

}

.select-wrapper.disabled .caret {

fill: rgba(0, 0, 0, 0.42);

}

.select-wrapper input.select-dropdown:disabled {

color: rgba(0, 0, 0, 0.42);

cursor: default;

-webkit-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

user-select: none;

}

.select-wrapper i {

color: rgba(0, 0, 0, 0.3);

}

.select-dropdown li.disabled,

.select-dropdown li.disabled > span,

.select-dropdown li.optgroup {

color: rgba(0, 0, 0, 0.3);

background-color: transparent;

}

body.keyboard-focused .select-dropdown.dropdown-content li:focus {

background-color: rgba(0, 0, 0, 0.08);

}

.select-dropdown.dropdown-content li:hover {

background-color: rgba(0, 0, 0, 0.08);

}

.select-dropdown.dropdown-content li.selected {

background-color: rgba(0, 0, 0, 0.03);

}

.prefix ~ .select-wrapper {

margin-left: 3rem;

width: 92%;

width: calc(100% - 3rem);

}

.prefix ~ label {

margin-left: 3rem;

}

.select-dropdown li img {

height: 40px;

width: 40px;

margin: 5px 15px;

float: right;

}

.select-dropdown li.optgroup {

border-top: 1px solid #eee;

}

.select-dropdown li.optgroup.selected > span {

color: rgba(0, 0, 0, 0.7);

}

.select-dropdown li.optgroup > span {

color: rgba(0, 0, 0, 0.4);

}

.select-dropdown li.optgroup ~ li.optgroup-option {

padding-left: 1rem;

}

/\* File Input

========================================================================== \*/

.file-field {

position: relative;

}

.file-field .file-path-wrapper {

overflow: hidden;

padding-left: 10px;

}

.file-field input.file-path {

width: 100%;

}

.file-field .btn, .file-field .btn-large, .file-field .btn-small {

float: left;

height: 3rem;

line-height: 3rem;

}

.file-field span {

cursor: pointer;

}

.file-field input[type=file] {

position: absolute;

top: 0;

right: 0;

left: 0;

bottom: 0;

width: 100%;

margin: 0;

padding: 0;

font-size: 20px;

cursor: pointer;

opacity: 0;

filter: alpha(opacity=0);

}

.file-field input[type=file]::-webkit-file-upload-button {

display: none;

}

/\* Range

========================================================================== \*/

.range-field {

position: relative;

}

input[type=range],

input[type=range] + .thumb {

cursor: pointer;

}

input[type=range] {

position: relative;

background-color: transparent;

border: none;

outline: none;

width: 100%;

margin: 15px 0;

padding: 0;

}

input[type=range]:focus {

outline: none;

}

input[type=range] + .thumb {

position: absolute;

top: 10px;

left: 0;

border: none;

height: 0;

width: 0;

border-radius: 50%;

background-color: #26a69a;

margin-left: 7px;

-webkit-transform-origin: 50% 50%;

transform-origin: 50% 50%;

-webkit-transform: rotate(-45deg);

transform: rotate(-45deg);

}

input[type=range] + .thumb .value {

display: block;

width: 30px;

text-align: center;

color: #26a69a;

font-size: 0;

-webkit-transform: rotate(45deg);

transform: rotate(45deg);

}

input[type=range] + .thumb.active {

border-radius: 50% 50% 50% 0;

}

input[type=range] + .thumb.active .value {

color: #fff;

margin-left: -1px;

margin-top: 8px;

font-size: 10px;

}

input[type=range] {

-webkit-appearance: none;

}

input[type=range]::-webkit-slider-runnable-track {

height: 3px;

background: #c2c0c2;

border: none;

}

input[type=range]::-webkit-slider-thumb {

border: none;

height: 14px;

width: 14px;

border-radius: 50%;

background: #26a69a;

-webkit-transition: -webkit-box-shadow .3s;

transition: -webkit-box-shadow .3s;

transition: box-shadow .3s;

transition: box-shadow .3s, -webkit-box-shadow .3s;

-webkit-appearance: none;

background-color: #26a69a;

-webkit-transform-origin: 50% 50%;

transform-origin: 50% 50%;

margin: -5px 0 0 0;

}

.keyboard-focused input[type=range]:focus:not(.active)::-webkit-slider-thumb {

-webkit-box-shadow: 0 0 0 10px rgba(38, 166, 154, 0.26);

box-shadow: 0 0 0 10px rgba(38, 166, 154, 0.26);

}

input[type=range] {

/\* fix for FF unable to apply focus style bug \*/

border: 1px solid white;

/\*required for proper track sizing in FF\*/

}

input[type=range]::-moz-range-track {

height: 3px;

background: #c2c0c2;

border: none;

}

input[type=range]::-moz-focus-inner {

border: 0;

}

input[type=range]::-moz-range-thumb {

border: none;

height: 14px;

width: 14px;

border-radius: 50%;

background: #26a69a;

-webkit-transition: -webkit-box-shadow .3s;

transition: -webkit-box-shadow .3s;

transition: box-shadow .3s;

transition: box-shadow .3s, -webkit-box-shadow .3s;

margin-top: -5px;

}

input[type=range]:-moz-focusring {

outline: 1px solid #fff;

outline-offset: -1px;

}

.keyboard-focused input[type=range]:focus:not(.active)::-moz-range-thumb {

box-shadow: 0 0 0 10px rgba(38, 166, 154, 0.26);

}

input[type=range]::-ms-track {

height: 3px;

background: transparent;

border-color: transparent;

border-width: 6px 0;

/\*remove default tick marks\*/

color: transparent;

}

input[type=range]::-ms-fill-lower {

background: #777;

}

input[type=range]::-ms-fill-upper {

background: #ddd;

}

input[type=range]::-ms-thumb {

border: none;

height: 14px;

width: 14px;

border-radius: 50%;

background: #26a69a;

-webkit-transition: -webkit-box-shadow .3s;

transition: -webkit-box-shadow .3s;

transition: box-shadow .3s;

transition: box-shadow .3s, -webkit-box-shadow .3s;

}

.keyboard-focused input[type=range]:focus:not(.active)::-ms-thumb {

box-shadow: 0 0 0 10px rgba(38, 166, 154, 0.26);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Nav List

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

.table-of-contents.fixed {

position: fixed;

}

.table-of-contents li {

padding: 2px 0;

}

.table-of-contents a {

display: inline-block;

font-weight: 300;

color: #757575;

padding-left: 16px;

height: 1.5rem;

line-height: 1.5rem;

letter-spacing: .4;

display: inline-block;

}

.table-of-contents a:hover {

color: #a8a8a8;

padding-left: 15px;

border-left: 1px solid #ee6e73;

}

.table-of-contents a.active {

font-weight: 500;

padding-left: 14px;

border-left: 2px solid #ee6e73;

}

.sidenav {

position: fixed;

width: 300px;

left: 0;

top: 0;

margin: 0;

-webkit-transform: translateX(-100%);

transform: translateX(-100%);

height: 100%;

height: calc(100% + 60px);

height: -moz-calc(100%);

padding-bottom: 60px;

background-color: #fff;

z-index: 999;

overflow-y: auto;

will-change: transform;

-webkit-backface-visibility: hidden;

backface-visibility: hidden;

-webkit-transform: translateX(-105%);

transform: translateX(-105%);

}

.sidenav.right-aligned {

right: 0;

-webkit-transform: translateX(105%);

transform: translateX(105%);

left: auto;

-webkit-transform: translateX(100%);

transform: translateX(100%);

}

.sidenav .collapsible {

margin: 0;

}

.sidenav li {

float: none;

line-height: 48px;

}

.sidenav li.active {

background-color: rgba(0, 0, 0, 0.05);

}

.sidenav li > a {

color: rgba(0, 0, 0, 0.87);

display: block;

font-size: 14px;

font-weight: 500;

height: 48px;

line-height: 48px;

padding: 0 32px;

}

.sidenav li > a:hover {

background-color: rgba(0, 0, 0, 0.05);

}

.sidenav li > a.btn, .sidenav li > a.btn-large, .sidenav li > a.btn-small, .sidenav li > a.btn-large, .sidenav li > a.btn-flat, .sidenav li > a.btn-floating {

margin: 10px 15px;

}

.sidenav li > a.btn, .sidenav li > a.btn-large, .sidenav li > a.btn-small, .sidenav li > a.btn-large, .sidenav li > a.btn-floating {

color: #fff;

}

.sidenav li > a.btn-flat {

color: #343434;

}

.sidenav li > a.btn:hover, .sidenav li > a.btn-large:hover, .sidenav li > a.btn-small:hover, .sidenav li > a.btn-large:hover {

background-color: #2bbbad;

}

.sidenav li > a.btn-floating:hover {

background-color: #26a69a;

}

.sidenav li > a > i,

.sidenav li > a > [class^="mdi-"], .sidenav li > a li > a > [class\*="mdi-"],

.sidenav li > a > i.material-icons {

float: left;

height: 48px;

line-height: 48px;

margin: 0 32px 0 0;

width: 24px;

color: rgba(0, 0, 0, 0.54);

}

.sidenav .divider {

margin: 8px 0 0 0;

}

.sidenav .subheader {

cursor: initial;

pointer-events: none;

color: rgba(0, 0, 0, 0.54);

font-size: 14px;

font-weight: 500;

line-height: 48px;

}

.sidenav .subheader:hover {

background-color: transparent;

}

.sidenav .user-view {

position: relative;

padding: 32px 32px 0;

margin-bottom: 8px;

}

.sidenav .user-view > a {

height: auto;

padding: 0;

}

.sidenav .user-view > a:hover {

background-color: transparent;

}

.sidenav .user-view .background {

overflow: hidden;

position: absolute;

top: 0;3

right: 0;

bottom: 0;

left: 0;

z-index: -1;

}

.sidenav .user-view .circle, .sidenav .user-view .name, .sidenav .user-view .email {

display: block;

}

.sidenav .user-view .circle {

height: 64px;

width: 64px;

}

.sidenav .user-view .name,

.sidenav .user-view .email {

font-size: 14px;

line-height: 24px;

}

.sidenav .user-view .name {

margin-top: 16px;

font-weight: 500;

}

.sidenav .user-view .email {

padding-bottom: 16px;

font-weight: 400;

}

.drag-target {

height: 100%;

width: 10px;

position: fixed;

top: 0;

z-index: 998;

}

.drag-target.right-aligned {

right: 0;

}

.sidenav.sidenav-fixed {

left: 0;

-webkit-transform: translateX(0);

transform: translateX(0);

position: fixed;

}

.sidenav.sidenav-fixed.right-aligned {

right: 0;

left: auto;

}

@media only screen and (max-width: 992px) {

.sidenav.sidenav-fixed {

-webkit-transform: translateX(-105%);

transform: translateX(-105%);

}

.sidenav.sidenav-fixed.right-aligned {

-webkit-transform: translateX(105%);

transform: translateX(105%);

}

.sidenav > a {

padding: 0 16px;

}

.sidenav .user-view {

padding: 16px 16px 0;

}

}

.sidenav .collapsible-body > ul:not(.collapsible) > li.active,

.sidenav.sidenav-fixed .collapsible-body > ul:not(.collapsible) > li.active {

background-color: #ee6e73;

}

.sidenav .collapsible-body > ul:not(.collapsible) > li.active a,

.sidenav.sidenav-fixed .collapsible-body > ul:not(.collapsible) > li.active a {

color: #fff;

}

.sidenav .collapsible-body {

padding: 0;

}

.sidenav-overlay {

position: fixed;

top: 0;

left: 0;

right: 0;

opacity: 0;

height: 120vh;

background-color: rgba(0, 0, 0, 0.5);

z-index: 997;

display: none;

}

.preloader-wrapper {

display: inline-block;

position: relative;

width: 50px;

height: 50px;

}

.preloader-wrapper.small {

width: 36px;

height: 36px;

}

.preloader-wrapper.big {

width: 64px;

height: 64px;

}

.preloader-wrapper.active {

/\* duration: 360 \* ARCTIME / (ARCSTARTROT + (360-ARCSIZE)) \*/

-webkit-animation: container-rotate 1568ms linear infinite;

animation: container-rotate 1568ms linear infinite;

}

@-webkit-keyframes container-rotate {

to {

-webkit-transform: rotate(360deg);

}

}

@keyframes container-rotate {

to {

-webkit-transform: rotate(360deg);

transform: rotate(360deg);

}

}

.spinner-layer {

position: absolute;

width: 100%;

height: 100%;

opacity: 0;

border-color: #26a69a;

}

.spinner-blue,

.spinner-blue-only {

border-color: #4285f4;

}

.spinner-red,

.spinner-red-only {

border-color: #db4437;

}

.spinner-yellow,

.spinner-yellow-only {

border-color: #f4b400;

}

.spinner-green,

.spinner-green-only {

border-color: #0f9d58;

}

.active .spinner-layer.spinner-blue {

/\* durations: 4 \* ARCTIME \*/

-webkit-animation: fill-unfill-rotate 5332ms cubic-bezier(0.4, 0, 0.2, 1) infinite both, blue-fade-in-out 5332ms cubic-bezier(0.4, 0, 0.2, 1) infinite both;

animation: fill-unfill-rotate 5332ms cubic-bezier(0.4, 0, 0.2, 1) infinite both, blue-fade-in-out 5332ms cubic-bezier(0.4, 0, 0.2, 1) infinite both;

}

.active .spinner-layer.spinner-red {

/\* durations: 4 \* ARCTIME \*/

-webkit-animation: fill-unfill-rotate 5332ms cubic-bezier(0.4, 0, 0.2, 1) infinite both, red-fade-in-out 5332ms cubic-bezier(0.4, 0, 0.2, 1) infinite both;

animation: fill-unfill-rotate 5332ms cubic-bezier(0.4, 0, 0.2, 1) infinite both, red-fade-in-out 5332ms cubic-bezier(0.4, 0, 0.2, 1) infinite both;

}

.active .spinner-layer.spinner-yellow {

/\* durations: 4 \* ARCTIME \*/

-webkit-animation: fill-unfill-rotate 5332ms cubic-bezier(0.4, 0, 0.2, 1) infinite both, yellow-fade-in-out 5332ms cubic-bezier(0.4, 0, 0.2, 1) infinite both;

animation: fill-unfill-rotate 5332ms cubic-bezier(0.4, 0, 0.2, 1) infinite both, yellow-fade-in-out 5332ms cubic-bezier(0.4, 0, 0.2, 1) infinite both;

}

.active .spinner-layer.spinner-green {

/\* durations: 4 \* ARCTIME \*/

-webkit-animation: fill-unfill-rotate 5332ms cubic-bezier(0.4, 0, 0.2, 1) infinite both, green-fade-in-out 5332ms cubic-bezier(0.4, 0, 0.2, 1) infinite both;

animation: fill-unfill-rotate 5332ms cubic-bezier(0.4, 0, 0.2, 1) infinite both, green-fade-in-out 5332ms cubic-bezier(0.4, 0, 0.2, 1) infinite both;

}

.active .spinner-layer,

.active .spinner-layer.spinner-blue-only,

.active .spinner-layer.spinner-red-only,

.active .spinner-layer.spinner-yellow-only,

.active .spinner-layer.spinner-green-only {

/\* durations: 4 \* ARCTIME \*/

opacity: 1;

-webkit-animation: fill-unfill-rotate 5332ms cubic-bezier(0.4, 0, 0.2, 1) infinite both;

animation: fill-unfill-rotate 5332ms cubic-bezier(0.4, 0, 0.2, 1) infinite both;

}

@-webkit-keyframes fill-unfill-rotate {

12.5% {

-webkit-transform: rotate(135deg);

}

/\* 0.5 \* ARCSIZE \*/

25% {

-webkit-transform: rotate(270deg);

}

/\* 1 \* ARCSIZE \*/

37.5% {

-webkit-transform: rotate(405deg);

}

/\* 1.5 \* ARCSIZE \*/

50% {

-webkit-transform: rotate(540deg);

}

/\* 2 \* ARCSIZE \*/

62.5% {

-webkit-transform: rotate(675deg);

}

/\* 2.5 \* ARCSIZE \*/

75% {

-webkit-transform: rotate(810deg);

}

/\* 3 \* ARCSIZE \*/

87.5% {

-webkit-transform: rotate(945deg);

}

/\* 3.5 \* ARCSIZE \*/

to {

-webkit-transform: rotate(1080deg);

}

/\* 4 \* ARCSIZE \*/

}

@keyframes fill-unfill-rotate {

12.5% {

-webkit-transform: rotate(135deg);

transform: rotate(135deg);

}

/\* 0.5 \* ARCSIZE \*/

25% {

-webkit-transform: rotate(270deg);

transform: rotate(270deg);

}

/\* 1 \* ARCSIZE \*/

37.5% {

-webkit-transform: rotate(405deg);

transform: rotate(405deg);

}

/\* 1.5 \* ARCSIZE \*/

50% {

-webkit-transform: rotate(540deg);

transform: rotate(540deg);

}

/\* 2 \* ARCSIZE \*/

62.5% {

-webkit-transform: rotate(675deg);

transform: rotate(675deg);

}

/\* 2.5 \* ARCSIZE \*/

75% {

-webkit-transform: rotate(810deg);

transform: rotate(810deg);

}

/\* 3 \* ARCSIZE \*/

87.5% {

-webkit-transform: rotate(945deg);

transform: rotate(945deg);

}

/\* 3.5 \* ARCSIZE \*/

to {

-webkit-transform: rotate(1080deg);

transform: rotate(1080deg);

}

/\* 4 \* ARCSIZE \*/

}

@-webkit-keyframes blue-fade-in-out {

from {

opacity: 1;

}

25% {

opacity: 1;

}

26% {

opacity: 0;

}

89% {

opacity: 0;

}

90% {

opacity: 1;

}

100% {

opacity: 1;

}

}

@keyframes blue-fade-in-out {

from {

opacity: 1;

}

25% {

opacity: 1;

}

26% {

opacity: 0;

}

89% {

opacity: 0;

}

90% {

opacity: 1;

}

100% {

opacity: 1;

}

}

@-webkit-keyframes red-fade-in-out {

from {

opacity: 0;

}

15% {

opacity: 0;

}

25% {

opacity: 1;

}

50% {

opacity: 1;

}

51% {

opacity: 0;

}

}

@keyframes red-fade-in-out {

from {

opacity: 0;

}

15% {

opacity: 0;

}

25% {

opacity: 1;

}

50% {

opacity: 1;

}

51% {

opacity: 0;

}

}

@-webkit-keyframes yellow-fade-in-out {

from {

opacity: 0;

}

40% {

opacity: 0;

}

50% {

opacity: 1;

}

75% {

opacity: 1;

}

76% {

opacity: 0;

}

}

@keyframes yellow-fade-in-out {

from {

opacity: 0;

}

40% {

opacity: 0;

}

50% {

opacity: 1;

}

75% {

opacity: 1;

}

76% {

opacity: 0;

}

}

@-webkit-keyframes green-fade-in-out {

from {

opacity: 0;

}

65% {

opacity: 0;

}

75% {

opacity: 1;

}

90% {

opacity: 1;

}

100% {

opacity: 0;

}

}

@keyframes green-fade-in-out {

from {

opacity: 0;

}

65% {

opacity: 0;

}

75% {

opacity: 1;

}

90% {

opacity: 1;

}

100% {

opacity: 0;

}

}

/\*\*

\* Patch the gap that appear between the two adjacent div.circle-clipper while the

\* spinner is rotating (appears on Chrome 38, Safari 7.1, and IE 11).

\*/

.gap-patch {

position: absolute;

top: 0;

left: 45%;

width: 10%;

height: 100%;

overflow: hidden;

border-color: inherit;

}

.gap-patch .circle {

width: 1000%;

left: -450%;

}

.circle-clipper {

display: inline-block;

position: relative;

width: 50%;

height: 100%;

overflow: hidden;

border-color: inherit;

}

.circle-clipper .circle {

width: 200%;

height: 100%;

border-width: 3px;

/\* STROKEWIDTH \*/

border-style: solid;

border-color: inherit;

border-bottom-color: transparent !important;

border-radius: 50%;

-webkit-animation: none;

animation: none;

position: absolute;

top: 0;

right: 0;

bottom: 0;

}

.circle-clipper.left .circle {

left: 0;

border-right-color: transparent !important;

-webkit-transform: rotate(129deg);

transform: rotate(129deg);

}

.circle-clipper.right .circle {

left: -100%;

border-left-color: transparent !important;

-webkit-transform: rotate(-129deg);

transform: rotate(-129deg);

}

.active .circle-clipper.left .circle {

/\* duration: ARCTIME \*/

-webkit-animation: left-spin 1333ms cubic-bezier(0.4, 0, 0.2, 1) infinite both;

animation: left-spin 1333ms cubic-bezier(0.4, 0, 0.2, 1) infinite both;

}

.active .circle-clipper.right .circle {

/\* duration: ARCTIME \*/

-webkit-animation: right-spin 1333ms cubic-bezier(0.4, 0, 0.2, 1) infinite both;

animation: right-spin 1333ms cubic-bezier(0.4, 0, 0.2, 1) infinite both;

}

@-webkit-keyframes left-spin {

from {

-webkit-transform: rotate(130deg);

}

50% {

-webkit-transform: rotate(-5deg);

}

to {

-webkit-transform: rotate(130deg);

}

}

@keyframes left-spin {

from {

-webkit-transform: rotate(130deg);

transform: rotate(130deg);

}

50% {

-webkit-transform: rotate(-5deg);

transform: rotate(-5deg);

}

to {

-webkit-transform: rotate(130deg);

transform: rotate(130deg);

}

}

@-webkit-keyframes right-spin {

from {

-webkit-transform: rotate(-130deg);

}

50% {

-webkit-transform: rotate(5deg);

}

to {

-webkit-transform: rotate(-130deg);

}

}

@keyframes right-spin {

from {

-webkit-transform: rotate(-130deg);

transform: rotate(-130deg);

}

50% {

-webkit-transform: rotate(5deg);

transform: rotate(5deg);

}

to {

-webkit-transform: rotate(-130deg);

transform: rotate(-130deg);

}

}

#spinnerContainer.cooldown {

/\* duration: SHRINK\_TIME \*/

-webkit-animation: container-rotate 1568ms linear infinite, fade-out 400ms cubic-bezier(0.4, 0, 0.2, 1);

animation: container-rotate 1568ms linear infinite, fade-out 400ms cubic-bezier(0.4, 0, 0.2, 1);

}

@-webkit-keyframes fade-out {

from {

opacity: 1;

}

to {

opacity: 0;

}

}

@keyframes fade-out {

from {

opacity: 1;

}

to {

opacity: 0;

}

}

.slider {

position: relative;

height: 400px;

width: 100%;

}

.slider.fullscreen {

height: 100%;

width: 100%;

position: absolute;

top: 0;

left: 0;

right: 0;

bottom: 0;

}

.slider.fullscreen ul.slides {

height: 100%;

}

.slider.fullscreen ul.indicators {

z-index: 2;

bottom: 30px;

}

.slider .slides {

background-color: #9e9e9e;

margin: 0;

height: 400px;

}

.slider .slides li {

opacity: 0;

position: absolute;

top: 0;

left: 0;

z-index: 1;

width: 100%;

height: inherit;

overflow: hidden;

}

.slider .slides li img {

height: 100%;

width: 100%;

background-size: cover;

background-position: center;

}

.slider .slides li .caption {

color: #fff;

position: absolute;

top: 15%;

left: 15%;

width: 70%;

opacity: 0;

}

.slider .slides li .caption p {

color: #e0e0e0;

}

.slider .slides li.active {

z-index: 2;

}

.slider .indicators {

position: absolute;

text-align: center;

left: 0;

right: 0;

bottom: 0;

margin: 0;

}

.slider .indicators .indicator-item {

display: inline-block;

position: relative;

cursor: pointer;

height: 16px;

width: 16px;

margin: 0 12px;

background-color: #e0e0e0;

-webkit-transition: background-color .3s;

transition: background-color .3s;

border-radius: 50%;

}

.slider .indicators .indicator-item.active {

background-color: #4CAF50;

}

.carousel {

overflow: hidden;

position: relative;

width: 100%;

height: 400px;

-webkit-perspective: 500px;

perspective: 500px;

-webkit-transform-style: preserve-3d;

transform-style: preserve-3d;

-webkit-transform-origin: 0% 50%;

transform-origin: 0% 50%;

}

.carousel.carousel-slider {

top: 0;

left: 0;

}

.carousel.carousel-slider .carousel-fixed-item {

position: absolute;

left: 0;

right: 0;

bottom: 20px;

z-index: 1;

}

.carousel.carousel-slider .carousel-fixed-item.with-indicators {

bottom: 68px;

}

.carousel.carousel-slider .carousel-item {

width: 100%;

height: 100%;

min-height: 400px;

position: absolute;

top: 0;

left: 0;

}

.carousel.carousel-slider .carousel-item h2 {

font-size: 24px;

font-weight: 500;

line-height: 32px;

}

.carousel.carousel-slider .carousel-item p {

font-size: 15px;

}

.carousel .carousel-item {

visibility: hidden;

width: 200px;

height: 200px;

position: absolute;

top: 0;

left: 0;

}

.carousel .carousel-item > img {

width: 100%;

}

.carousel .indicators {

position: absolute;

text-align: center;

left: 0;

right: 0;

bottom: 0;

margin: 0;

}

.carousel .indicators .indicator-item {

display: inline-block;

position: relative;

cursor: pointer;

height: 8px;

width: 8px;

margin: 24px 4px;

background-color: rgba(255, 255, 255, 0.5);

-webkit-transition: background-color .3s;

transition: background-color .3s;

border-radius: 50%;

}

.carousel .indicators .indicator-item.active {

background-color: #fff;

}

.carousel.scrolling .carousel-item .materialboxed,

.carousel .carousel-item:not(.active) .materialboxed {

pointer-events: none;

}

.tap-target-wrapper {

width: 800px;

height: 800px;

position: fixed;

z-index: 1000;

visibility: hidden;

-webkit-transition: visibility 0s .3s;

transition: visibility 0s .3s;

}

.tap-target-wrapper.open {

visibility: visible;

-webkit-transition: visibility 0s;

transition: visibility 0s;

}

.tap-target-wrapper.open .tap-target {

-webkit-transform: scale(1);

transform: scale(1);

opacity: .95;

-webkit-transition: opacity 0.3s cubic-bezier(0.42, 0, 0.58, 1), -webkit-transform 0.3s cubic-bezier(0.42, 0, 0.58, 1);

transition: opacity 0.3s cubic-bezier(0.42, 0, 0.58, 1), -webkit-transform 0.3s cubic-bezier(0.42, 0, 0.58, 1);

transition: transform 0.3s cubic-bezier(0.42, 0, 0.58, 1), opacity 0.3s cubic-bezier(0.42, 0, 0.58, 1);

transition: transform 0.3s cubic-bezier(0.42, 0, 0.58, 1), opacity 0.3s cubic-bezier(0.42, 0, 0.58, 1), -webkit-transform 0.3s cubic-bezier(0.42, 0, 0.58, 1);

}

.tap-target-wrapper.open .tap-target-wave::before {

-webkit-transform: scale(1);

transform: scale(1);

}

.tap-target-wrapper.open .tap-target-wave::after {

visibility: visible;

-webkit-animation: pulse-animation 1s cubic-bezier(0.24, 0, 0.38, 1) infinite;

animation: pulse-animation 1s cubic-bezier(0.24, 0, 0.38, 1) infinite;

-webkit-transition: opacity .3s,

visibility 0s 1s,

-webkit-transform .3s;

transition: opacity .3s,

visibility 0s 1s,

-webkit-transform .3s;

transition: opacity .3s,

transform .3s,

visibility 0s 1s;

transition: opacity .3s,

transform .3s,

visibility 0s 1s,

-webkit-transform .3s;

}

.tap-target {

position: absolute;

font-size: 1rem;

border-radius: 50%;

background-color: #ee6e73;

-webkit-box-shadow: 0 20px 20px 0 rgba(0, 0, 0, 0.14), 0 10px 50px 0 rgba(0, 0, 0, 0.12), 0 30px 10px -20px rgba(0, 0, 0, 0.2);

box-shadow: 0 20px 20px 0 rgba(0, 0, 0, 0.14), 0 10px 50px 0 rgba(0, 0, 0, 0.12), 0 30px 10px -20px rgba(0, 0, 0, 0.2);

width: 100%;

height: 100%;

opacity: 0;

-webkit-transform: scale(0);

transform: scale(0);

-webkit-transition: opacity 0.3s cubic-bezier(0.42, 0, 0.58, 1), -webkit-transform 0.3s cubic-bezier(0.42, 0, 0.58, 1);

transition: opacity 0.3s cubic-bezier(0.42, 0, 0.58, 1), -webkit-transform 0.3s cubic-bezier(0.42, 0, 0.58, 1);

transition: transform 0.3s cubic-bezier(0.42, 0, 0.58, 1), opacity 0.3s cubic-bezier(0.42, 0, 0.58, 1);

transition: transform 0.3s cubic-bezier(0.42, 0, 0.58, 1), opacity 0.3s cubic-bezier(0.42, 0, 0.58, 1), -webkit-transform 0.3s cubic-bezier(0.42, 0, 0.58, 1);

}

.tap-target-content {

position: relative;

display: table-cell;

}

.tap-target-wave {

position: absolute;

border-radius: 50%;

z-index: 10001;

}

.tap-target-wave::before, .tap-target-wave::after {

content: '';

display: block;

position: absolute;

width: 100%;

height: 100%;

border-radius: 50%;

background-color: #ffffff;

}

.tap-target-wave::before {

-webkit-transform: scale(0);

transform: scale(0);

-webkit-transition: -webkit-transform .3s;

transition: -webkit-transform .3s;

transition: transform .3s;

transition: transform .3s, -webkit-transform .3s;

}

.tap-target-wave::after {

visibility: hidden;

-webkit-transition: opacity .3s,

visibility 0s,

-webkit-transform .3s;

transition: opacity .3s,

visibility 0s,

-webkit-transform .3s;

transition: opacity .3s,

transform .3s,

visibility 0s;

transition: opacity .3s,

transform .3s,

visibility 0s,

-webkit-transform .3s;

z-index: -1;

}

.tap-target-origin {

top: 50%;

left: 50%;

-webkit-transform: translate(-50%, -50%);

transform: translate(-50%, -50%);

z-index: 10002;

position: absolute !important;

}

.tap-target-origin:not(.btn):not(.btn-large):not(.btn-small), .tap-target-origin:not(.btn):not(.btn-large):not(.btn-small):hover {

background: none;

}

@media only screen and (max-width: 600px) {

.tap-target, .tap-target-wrapper {

width: 600px;

height: 600px;

}

}

.pulse {

overflow: visible;

position: relative;

}

.pulse::before {

content: '';

display: block;

position: absolute;

width: 100%;

height: 100%;

top: 0;

left: 0;

background-color: inherit;

border-radius: inherit;

-webkit-transition: opacity .3s, -webkit-transform .3s;

transition: opacity .3s, -webkit-transform .3s;

transition: opacity .3s, transform .3s;

transition: opacity .3s, transform .3s, -webkit-transform .3s;

-webkit-animation: pulse-animation 1s cubic-bezier(0.24, 0, 0.38, 1) infinite;

animation: pulse-animation 1s cubic-bezier(0.24, 0, 0.38, 1) infinite;

z-index: -1;

}

@-webkit-keyframes pulse-animation {

0% {

opacity: 1;

-webkit-transform: scale(1);

transform: scale(1);

}

50% {

opacity: 0;

-webkit-transform: scale(1.5);

transform: scale(1.5);

}

100% {

opacity: 0;

-webkit-transform: scale(1.5);

transform: scale(1.5);

}

}

@keyframes pulse-animation {

0% {

opacity: 1;

-webkit-transform: scale(1);

transform: scale(1);

}

50% {

opacity: 0;

-webkit-transform: scale(1.5);

transform: scale(1.5);

}

100% {

opacity: 0;

-webkit-transform: scale(1.5);

transform: scale(1.5);

}

}

/\* Modal \*/

.datepicker-modal {

max-width: 325px;

min-width: 300px;

max-height: none;

}

.datepicker-container.modal-content {

display: -webkit-box;

display: -webkit-flex;

display: -ms-flexbox;

display: flex;

-webkit-box-orient: vertical;

-webkit-box-direction: normal;

-webkit-flex-direction: column;

-ms-flex-direction: column;

flex-direction: column;

padding: 0;

}

.datepicker-controls {

display: -webkit-box;

display: -webkit-flex;

display: -ms-flexbox;

display: flex;

-webkit-box-pack: justify;

-webkit-justify-content: space-between;

-ms-flex-pack: justify;

justify-content: space-between;

width: 280px;

margin: 0 auto;

}

.datepicker-controls .selects-container {

display: -webkit-box;

display: -webkit-flex;

display: -ms-flexbox;

display: flex;

}

.datepicker-controls .select-wrapper input {

border-bottom: none;

text-align: center;

margin: 0;

}

.datepicker-controls .select-wrapper input:focus {

border-bottom: none;

}

.datepicker-controls .select-wrapper .caret {

display: none;

}

.datepicker-controls .select-year input {

width: 50px;

}

.datepicker-controls .select-month input {

width: 70px;

}

.month-prev, .month-next {

margin-top: 4px;

cursor: pointer;

background-color: transparent;

border: none;

}

/\* Date Display \*/

.datepicker-date-display {

-webkit-box-flex: 1;

-webkit-flex: 1 auto;

-ms-flex: 1 auto;

flex: 1 auto;

background-color: #26a69a;

color: #fff;

padding: 20px 22px;

font-weight: 500;

}

.datepicker-date-display .year-text {

display: block;

font-size: 1.5rem;

line-height: 25px;

color: rgba(255, 255, 255, 0.7);

}

.datepicker-date-display .date-text {

display: block;

font-size: 2.8rem;

line-height: 47px;

font-weight: 500;

}

/\* Calendar \*/

.datepicker-calendar-container {

-webkit-box-flex: 2.5;

-webkit-flex: 2.5 auto;

-ms-flex: 2.5 auto;

flex: 2.5 auto;

}

.datepicker-table {

width: 280px;

font-size: 1rem;

margin: 0 auto;

}

.datepicker-table thead {

border-bottom: none;

}

.datepicker-table th {

padding: 10px 5px;

text-align: center;

}

.datepicker-table tr {

border: none;

}

.datepicker-table abbr {

text-decoration: none;

color: #999;

}

.datepicker-table td {

border-radius: 50%;

padding: 0;

}

.datepicker-table td.is-today {

color: #26a69a;

}

.datepicker-table td.is-selected {

background-color: #26a69a;

color: #fff;

}

.datepicker-table td.is-outside-current-month, .datepicker-table td.is-disabled {

color: rgba(0, 0, 0, 0.3);

pointer-events: none;

}

.datepicker-day-button {

background-color: transparent;

border: none;

line-height: 38px;

display: block;

width: 100%;

border-radius: 50%;

padding: 0 5px;

cursor: pointer;

color: inherit;

}

.datepicker-day-button:focus {

background-color: rgba(43, 161, 150, 0.25);

}

/\* Footer \*/

.datepicker-footer {

width: 280px;

margin: 0 auto;

padding-bottom: 5px;

display: -webkit-box;

display: -webkit-flex;

display: -ms-flexbox;

display: flex;

-webkit-box-pack: justify;

-webkit-justify-content: space-between;

-ms-flex-pack: justify;

justify-content: space-between;

}

.datepicker-cancel,

.datepicker-clear,

.datepicker-today,

.datepicker-done {

color: #26a69a;

padding: 0 1rem;

}

.datepicker-clear {

color: #F44336;

}

/\* Media Queries \*/

@media only screen and (min-width: 601px) {

.datepicker-modal {

max-width: 625px;

}

.datepicker-container.modal-content {

-webkit-box-orient: horizontal;

-webkit-box-direction: normal;

-webkit-flex-direction: row;

-ms-flex-direction: row;

flex-direction: row;

}

.datepicker-date-display {

-webkit-box-flex: 0;

-webkit-flex: 0 1 270px;

-ms-flex: 0 1 270px;

flex: 0 1 270px;

}

.datepicker-controls,

.datepicker-table,

.datepicker-footer {

width: 320px;

}

.datepicker-day-button {

line-height: 44px;

}

}

/\* Timepicker Containers \*/

.timepicker-modal {

max-width: 325px;

max-height: none;

}

.timepicker-container.modal-content {

display: -webkit-box;

display: -webkit-flex;

display: -ms-flexbox;

display: flex;

-webkit-box-orient: vertical;

-webkit-box-direction: normal;

-webkit-flex-direction: column;

-ms-flex-direction: column;

flex-direction: column;

padding: 0;

}

.text-primary {

color: white;

}

/\* Clock Digital Display \*/

.timepicker-digital-display {

-webkit-box-flex: 1;

-webkit-flex: 1 auto;

-ms-flex: 1 auto;

flex: 1 auto;

background-color: #26a69a;

padding: 10px;

font-weight: 300;

}

.timepicker-text-container {

font-size: 4rem;

font-weight: bold;

text-align: center;

color: rgba(255, 255, 255, 0.6);

font-weight: 400;

position: relative;

-webkit-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

user-select: none;

}

.timepicker-span-hours,

.timepicker-span-minutes,

.timepicker-span-am-pm div {

cursor: pointer;

}

.timepicker-span-hours {

margin-right: 3px;

}

.timepicker-span-minutes {

margin-left: 3px;

}

.timepicker-display-am-pm {

font-size: 1.3rem;

position: absolute;

right: 1rem;

bottom: 1rem;

font-weight: 400;

}

/\* Analog Clock Display \*/

.timepicker-analog-display {

-webkit-box-flex: 2.5;

-webkit-flex: 2.5 auto;

-ms-flex: 2.5 auto;

flex: 2.5 auto;

}

.timepicker-plate {

background-color: #eee;

border-radius: 50%;

width: 270px;

height: 270px;

overflow: visible;

position: relative;

margin: auto;

margin-top: 25px;

margin-bottom: 5px;

-webkit-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

user-select: none;

}

.timepicker-canvas,

.timepicker-dial {

position: absolute;

left: 0;

right: 0;

top: 0;

bottom: 0;

}

.timepicker-minutes {

visibility: hidden;

}

.timepicker-tick {

border-radius: 50%;

color: rgba(0, 0, 0, 0.87);

line-height: 40px;

text-align: center;

width: 40px;

height: 40px;

position: absolute;

cursor: pointer;

font-size: 15px;

}

.timepicker-tick.active,

.timepicker-tick:hover {

background-color: rgba(38, 166, 154, 0.25);

}

.timepicker-dial {

-webkit-transition: opacity 350ms, -webkit-transform 350ms;

transition: opacity 350ms, -webkit-transform 350ms;

transition: transform 350ms, opacity 350ms;

transition: transform 350ms, opacity 350ms, -webkit-transform 350ms;

}

.timepicker-dial-out {

opacity: 0;

}

.timepicker-dial-out.timepicker-hours {

-webkit-transform: scale(1.1, 1.1);

transform: scale(1.1, 1.1);

}

.timepicker-dial-out.timepicker-minutes {

-webkit-transform: scale(0.8, 0.8);

transform: scale(0.8, 0.8);

}

.timepicker-canvas {

-webkit-transition: opacity 175ms;

transition: opacity 175ms;

}

.timepicker-canvas line {

stroke: #26a69a;

stroke-width: 4;

stroke-linecap: round;

}

.timepicker-canvas-out {

opacity: 0.25;

}

.timepicker-canvas-bearing {

stroke: none;

fill: #26a69a;

}

.timepicker-canvas-bg {

stroke: none;

fill: #26a69a;

}

/\* Footer \*/

.timepicker-footer {

margin: 0 auto;

padding: 5px 1rem;

display: -webkit-box;

display: -webkit-flex;

display: -ms-flexbox;

display: flex;

-webkit-box-pack: justify;

-webkit-justify-content: space-between;

-ms-flex-pack: justify;

justify-content: space-between;

}

.timepicker-clear {

color: #F44336;

}

.timepicker-close {

color: #26a69a;

}

.timepicker-clear,

.timepicker-close {

padding: 0 20px;

}

/\* Media Queries \*/

@media only screen and (min-width: 601px) {

.timepicker-modal {

max-width: 600px;

}

.timepicker-container.modal-content {

-webkit-box-orient: horizontal;

-webkit-box-direction: normal;

-webkit-flex-direction: row;

-ms-flex-direction: row;

flex-direction: row;

}

.timepicker-text-container {

top: 32%;

}

.timepicker-display-am-pm {

position: relative;

right: auto;

bottom: auto;

text-align: center;

margin-top: 1.2rem;

}

}

**Java Script code**

**Init.js**

(function($){

$(function(){

$('.sidenav').sidenav();

$('.parallax').parallax();

// TABS INIT

$('.tabs').tabs();

// CAROUSEL INIT

$('.carousel.carousel-slider').carousel({ fullWidth: true });

// SLIDER INIT

$('.slider').slider({

indicators: false,

// we don't want the little dots to show

height: 500,

transition: 500,

interval: 6000

// how long the slide stays for

});

}); // end of document ready

})(jQuery); // end of jQuery name space

**materialize.js**

var \_get = function get(object, property, receiver) { if (object === null) object = Function.prototype; var desc = Object.getOwnPropertyDescriptor(object, property); if (desc === undefined) { var parent = Object.getPrototypeOf(object); if (parent === null) { return undefined; } else { return get(parent, property, receiver); } } else if ("value" in desc) { return desc.value; } else { var getter = desc.get; if (getter === undefined) { return undefined; } return getter.call(receiver); } };

var \_createClass = function () { function defineProperties(target, props) { for (var i = 0; i < props.length; i++) { var descriptor = props[i]; descriptor.enumerable = descriptor.enumerable || false; descriptor.configurable = true; if ("value" in descriptor) descriptor.writable = true; Object.defineProperty(target, descriptor.key, descriptor); } } return function (Constructor, protoProps, staticProps) { if (protoProps) defineProperties(Constructor.prototype, protoProps); if (staticProps) defineProperties(Constructor, staticProps); return Constructor; }; }();

function \_possibleConstructorReturn(self, call) { if (!self) { throw new ReferenceError("this hasn't been initialised - super() hasn't been called"); } return call && (typeof call === "object" || typeof call === "function") ? call : self; }

function \_inherits(subClass, superClass) { if (typeof superClass !== "function" && superClass !== null) { throw new TypeError("Super expression must either be null or a function, not " + typeof superClass); } subClass.prototype = Object.create(superClass && superClass.prototype, { constructor: { value: subClass, enumerable: false, writable: true, configurable: true } }); if (superClass) Object.setPrototypeOf ? Object.setPrototypeOf(subClass, superClass) : subClass.\_\_proto\_\_ = superClass; }

function \_classCallCheck(instance, Constructor) { if (!(instance instanceof Constructor)) { throw new TypeError("Cannot call a class as a function"); } }

(function (factory) {

window.cash = factory();

})(function () {

var doc = document,

win = window,

ArrayProto = Array.prototype,

slice = ArrayProto.slice,

filter = ArrayProto.filter,

push = ArrayProto.push;

var noop = function () {},

isFunction = function (item) {

return typeof item === typeof noop && item.call;

},

isString = function (item) {

return typeof item === typeof "";

};

var idMatch = /^#[\w-]\*$/,

classMatch = /^\.[\w-]\*$/,

htmlMatch = /<.+>/,

singlet = /^\w+$/;

function find(selector, context) {

context = context || doc;

var elems = classMatch.test(selector) ? context.getElementsByClassName(selector.slice(1)) : singlet.test(selector) ? context.getElementsByTagName(selector) : context.querySelectorAll(selector);

return elems;

}

var frag;

function parseHTML(str) {

if (!frag) {

frag = doc.implementation.createHTMLDocument(null);

var base = frag.createElement("base");

base.href = doc.location.href;

frag.head.appendChild(base);

}

frag.body.innerHTML = str;

return frag.body.childNodes;

}

function onReady(fn) {

if (doc.readyState !== "loading") {

fn();

} else {

doc.addEventListener("DOMContentLoaded", fn);

}

}

function Init(selector, context) {

if (!selector) {

return this;

}

// If already a cash collection, don't do any further processing

if (selector.cash && selector !== win) {

return selector;

}

var elems = selector,

i = 0,

length;

if (isString(selector)) {

elems = idMatch.test(selector) ?

// If an ID use the faster getElementById check

doc.getElementById(selector.slice(1)) : htmlMatch.test(selector) ?

// If HTML, parse it into real elements

parseHTML(selector) :

// else use `find`

find(selector, context);

// If function, use as shortcut for DOM ready

} else if (isFunction(selector)) {

onReady(selector);return this;

}

if (!elems) {

return this;

}

// If a single DOM element is passed in or received via ID, return the single element

if (elems.nodeType || elems === win) {

this[0] = elems;

this.length = 1;

} else {

// Treat like an array and loop through each item.

length = this.length = elems.length;

for (; i < length; i++) {

this[i] = elems[i];

}

}

return this;

}

function cash(selector, context) {

return new Init(selector, context);

}

var fn = cash.fn = cash.prototype = Init.prototype = { // jshint ignore:line

cash: true,

length: 0,

push: push,

splice: ArrayProto.splice,

map: ArrayProto.map,

init: Init

};

Object.defineProperty(fn, "constructor", { value: cash });

cash.parseHTML = parseHTML;

cash.noop = noop;

cash.isFunction = isFunction;

cash.isString = isString;

cash.extend = fn.extend = function (target) {

target = target || {};

var args = slice.call(arguments),

length = args.length,

i = 1;

if (args.length === 1) {

target = this;

i = 0;

}

for (; i < length; i++) {

if (!args[i]) {

continue;

}

for (var key in args[i]) {

if (args[i].hasOwnProperty(key)) {

target[key] = args[i][key];

}

}

}

return target;

};

function each(collection, callback) {

var l = collection.length,

i = 0;

for (; i < l; i++) {

if (callback.call(collection[i], collection[i], i, collection) === false) {

break;

}

}

}

function matches(el, selector) {

var m = el && (el.matches || el.webkitMatchesSelector || el.mozMatchesSelector || el.msMatchesSelector || el.oMatchesSelector);

return !!m && m.call(el, selector);

}

function getCompareFunction(selector) {

return (

/\* Use browser's `matches` function if string \*/

isString(selector) ? matches :

/\* Match a cash element \*/

selector.cash ? function (el) {

return selector.is(el);

} :

/\* Direct comparison \*/

function (el, selector) {

return el === selector;

}

);

}

function unique(collection) {

return cash(slice.call(collection).filter(function (item, index, self) {

return self.indexOf(item) === index;

}));

}

cash.extend({

merge: function (first, second) {

var len = +second.length,

i = first.length,

j = 0;

for (; j < len; i++, j++) {

first[i] = second[j];

}

first.length = i;

return first;

},

each: each,

matches: matches,

unique: unique,

isArray: Array.isArray,

isNumeric: function (n) {

return !isNaN(parseFloat(n)) && isFinite(n);

}

});

var uid = cash.uid = "\_cash" + Date.now();

function getDataCache(node) {

return node[uid] = node[uid] || {};

}

function setData(node, key, value) {

return getDataCache(node)[key] = value;

}

function getData(node, key) {

var c = getDataCache(node);

if (c[key] === undefined) {

c[key] = node.dataset ? node.dataset[key] : cash(node).attr("data-" + key);

}

return c[key];

}

function removeData(node, key) {

var c = getDataCache(node);

if (c) {

delete c[key];

} else if (node.dataset) {

delete node.dataset[key];

} else {

cash(node).removeAttr("data-" + name);

}

}

fn.extend({

data: function (name, value) {

if (isString(name)) {

return value === undefined ? getData(this[0], name) : this.each(function (v) {

return setData(v, name, value);

});

}

for (var key in name) {

this.data(key, name[key]);

}

return this;

},

removeData: function (key) {

return this.each(function (v) {

return removeData(v, key);

});

}

});

var notWhiteMatch = /\S+/g;

function getClasses(c) {

return isString(c) && c.match(notWhiteMatch);

}

function hasClass(v, c) {

return v.classList ? v.classList.contains(c) : new RegExp("(^| )" + c + "( |$)", "gi").test(v.className);

}

function addClass(v, c, spacedName) {

if (v.classList) {

v.classList.add(c);

} else if (spacedName.indexOf(" " + c + " ")) {

v.className += " " + c;

}

}

function removeClass(v, c) {

if (v.classList) {

v.classList.remove(c);

} else {

v.className = v.className.replace(c, "");

}

}

fn.extend({

addClass: function (c) {

var classes = getClasses(c);

return classes ? this.each(function (v) {

var spacedName = " " + v.className + " ";

each(classes, function (c) {

addClass(v, c, spacedName);

});

}) : this;

},

attr: function (name, value) {

if (!name) {

return undefined;

}

if (isString(name)) {

if (value === undefined) {

return this[0] ? this[0].getAttribute ? this[0].getAttribute(name) : this[0][name] : undefined;

}

return this.each(function (v) {

if (v.setAttribute) {

v.setAttribute(name, value);

} else {

v[name] = value;

}

});

}

for (var key in name) {

this.attr(key, name[key]);

}

return this;

},

hasClass: function (c) {

var check = false,

classes = getClasses(c);

if (classes && classes.length) {

this.each(function (v) {

check = hasClass(v, classes[0]);

return !check;

});

}

return check;

},

prop: function (name, value) {

if (isString(name)) {

return value === undefined ? this[0][name] : this.each(function (v) {

v[name] = value;

});

}

for (var key in name) {

this.prop(key, name[key]);

}

return this;

},

removeAttr: function (name) {

return this.each(function (v) {

if (v.removeAttribute) {

v.removeAttribute(name);

} else {

delete v[name];

}

});

},

removeClass: function (c) {

if (!arguments.length) {

return this.attr("class", "");

}

var classes = getClasses(c);

return classes ? this.each(function (v) {

each(classes, function (c) {

removeClass(v, c);

});

}) : this;

},

removeProp: function (name) {

return this.each(function (v) {

delete v[name];

});

},

toggleClass: function (c, state) {

if (state !== undefined) {

return this[state ? "addClass" : "removeClass"](c);

}

var classes = getClasses(c);

return classes ? this.each(function (v) {

var spacedName = " " + v.className + " ";

each(classes, function (c) {

if (hasClass(v, c)) {

removeClass(v, c);

} else {

addClass(v, c, spacedName);

}

});

}) : this;

} });

fn.extend({

add: function (selector, context) {

return unique(cash.merge(this, cash(selector, context)));

},

each: function (callback) {

each(this, callback);

return this;

},

eq: function (index) {

return cash(this.get(index));

},

filter: function (selector) {

if (!selector) {

return this;

}

var comparator = isFunction(selector) ? selector : getCompareFunction(selector);

return cash(filter.call(this, function (e) {

return comparator(e, selector);

}));

},

first: function () {

return this.eq(0);

},

get: function (index) {

if (index === undefined) {

return slice.call(this);

}

return index < 0 ? this[index + this.length] : this[index];

},

index: function (elem) {

var child = elem ? cash(elem)[0] : this[0],

collection = elem ? this : cash(child).parent().children();

return slice.call(collection).indexOf(child);

},

last: function () {

return this.eq(-1);

}

});

var camelCase = function () {

var camelRegex = /(?:^\w|[A-Z]|\b\w)/g,

whiteSpace = /[\s-\_]+/g;

return function (str) {

return str.replace(camelRegex, function (letter, index) {

return letter[index === 0 ? "toLowerCase" : "toUpperCase"]();

}).replace(whiteSpace, "");

};

}();

var getPrefixedProp = function () {

var cache = {},

doc = document,

div = doc.createElement("div"),

style = div.style;

return function (prop) {

prop = camelCase(prop);

if (cache[prop]) {

return cache[prop];

}

var ucProp = prop.charAt(0).toUpperCase() + prop.slice(1),

prefixes = ["webkit", "moz", "ms", "o"],

props = (prop + " " + prefixes.join(ucProp + " ") + ucProp).split(" ");

each(props, function (p) {

if (p in style) {

cache[p] = prop = cache[prop] = p;

return false;

}

});

return cache[prop];

};

}();

cash.prefixedProp = getPrefixedProp;

cash.camelCase = camelCase;

fn.extend({

css: function (prop, value) {

if (isString(prop)) {

prop = getPrefixedProp(prop);

return arguments.length > 1 ? this.each(function (v) {

return v.style[prop] = value;

}) : win.getComputedStyle(this[0])[prop];

}

for (var key in prop) {

this.css(key, prop[key]);

}

return this;

}

});

function compute(el, prop) {

return parseInt(win.getComputedStyle(el[0], null)[prop], 10) || 0;

}

each(["Width", "Height"], function (v) {

var lower = v.toLowerCase();

fn[lower] = function () {

return this[0].getBoundingClientRect()[lower];

};

fn["inner" + v] = function () {

return this[0]["client" + v];

};

fn["outer" + v] = function (margins) {

return this[0]["offset" + v] + (margins ? compute(this, "margin" + (v === "Width" ? "Left" : "Top")) + compute(this, "margin" + (v === "Width" ? "Right" : "Bottom")) : 0);

};

});

function registerEvent(node, eventName, callback) {

var eventCache = getData(node, "\_cashEvents") || setData(node, "\_cashEvents", {});

eventCache[eventName] = eventCache[eventName] || [];

eventCache[eventName].push(callback);

node.addEventListener(eventName, callback);

}

function removeEvent(node, eventName, callback) {

var events = getData(node, "\_cashEvents"),

eventCache = events && events[eventName],

index;

if (!eventCache) {

return;

}

if (callback) {

node.removeEventListener(eventName, callback);

index = eventCache.indexOf(callback);

if (index >= 0) {

eventCache.splice(index, 1);

}

} else {

each(eventCache, function (event) {

node.removeEventListener(eventName, event);

});

eventCache = [];

}

}

fn.extend({

off: function (eventName, callback) {

return this.each(function (v) {

return removeEvent(v, eventName, callback);

});

},

on: function (eventName, delegate, callback, runOnce) {

// jshint ignore:line

var originalCallback;

if (!isString(eventName)) {

for (var key in eventName) {

this.on(key, delegate, eventName[key]);

}

return this;

}

if (isFunction(delegate)) {

callback = delegate;

delegate = null;

}

if (eventName === "ready") {

onReady(callback);

return this;

}

if (delegate) {

originalCallback = callback;

callback = function (e) {

var t = e.target;

while (!matches(t, delegate)) {

if (t === this || t === null) {

return t = false;

}

t = t.parentNode;

}

if (t) {

originalCallback.call(t, e);

}

};

}

return this.each(function (v) {

var finalCallback = callback;

if (runOnce) {

finalCallback = function () {

callback.apply(this, arguments);

removeEvent(v, eventName, finalCallback);

};

}

registerEvent(v, eventName, finalCallback);

});

},

one: function (eventName, delegate, callback) {

return this.on(eventName, delegate, callback, true);

},

ready: onReady,

/\*\*

\* Modified

\* Triggers browser event

\* @param String eventName

\* @param Object data - Add properties to event object

\*/

trigger: function (eventName, data) {

if (document.createEvent) {

var evt = document.createEvent('HTMLEvents');

evt.initEvent(eventName, true, false);

evt = this.extend(evt, data);

return this.each(function (v) {

return v.dispatchEvent(evt);

});

}

}

});

function encode(name, value) {

return "&" + encodeURIComponent(name) + "=" + encodeURIComponent(value).replace(/%20/g, "+");

}

function getSelectMultiple\_(el) {

var values = [];

each(el.options, function (o) {

if (o.selected) {

values.push(o.value);

}

});

return values.length ? values : null;

}

function getSelectSingle\_(el) {

var selectedIndex = el.selectedIndex;

return selectedIndex >= 0 ? el.options[selectedIndex].value : null;

}

function getValue(el) {

var type = el.type;

if (!type) {

return null;

}

switch (type.toLowerCase()) {

case "select-one":

return getSelectSingle\_(el);

case "select-multiple":

return getSelectMultiple\_(el);

case "radio":

return el.checked ? el.value : null;

case "checkbox":

return el.checked ? el.value : null;

default:

return el.value ? el.value : null;

}

}

fn.extend({

serialize: function () {

var query = "";

each(this[0].elements || this, function (el) {

if (el.disabled || el.tagName === "FIELDSET") {

return;

}

var name = el.name;

switch (el.type.toLowerCase()) {

case "file":

case "reset":

case "submit":

case "button":

break;

case "select-multiple":

var values = getValue(el);

if (values !== null) {

each(values, function (value) {

query += encode(name, value);

});

}

break;

default:

var value = getValue(el);

if (value !== null) {

query += encode(name, value);

}

}

});

return query.substr(1);

},

val: function (value) {

if (value === undefined) {

return getValue(this[0]);

}

return this.each(function (v) {

return v.value = value;

});

}

});

function insertElement(el, child, prepend) {

if (prepend) {

var first = el.childNodes[0];

el.insertBefore(child, first);

} else {

el.appendChild(child);

}

}

function insertContent(parent, child, prepend) {

var str = isString(child);

if (!str && child.length) {

each(child, function (v) {

return insertContent(parent, v, prepend);

});

return;

}

each(parent, str ? function (v) {

return v.insertAdjacentHTML(prepend ? "afterbegin" : "beforeend", child);

} : function (v, i) {

return insertElement(v, i === 0 ? child : child.cloneNode(true), prepend);

});

}

fn.extend({

after: function (selector) {

cash(selector).insertAfter(this);

return this;

},

append: function (content) {

insertContent(this, content);

return this;

},

appendTo: function (parent) {

insertContent(cash(parent), this);

return this;

},

before: function (selector) {

cash(selector).insertBefore(this);

return this;

},

clone: function () {

return cash(this.map(function (v) {

return v.cloneNode(true);

}));

},

empty: function () {

this.html("");

return this;

},

html: function (content) {

if (content === undefined) {

return this[0].innerHTML;

}

var source = content.nodeType ? content[0].outerHTML : content;

return this.each(function (v) {

return v.innerHTML = source;

});

},

insertAfter: function (selector) {

var \_this = this;

cash(selector).each(function (el, i) {

var parent = el.parentNode,

sibling = el.nextSibling;

\_this.each(function (v) {

parent.insertBefore(i === 0 ? v : v.cloneNode(true), sibling);

});

});

return this;

},

insertBefore: function (selector) {

var \_this2 = this;

cash(selector).each(function (el, i) {

var parent = el.parentNode;

\_this2.each(function (v) {

parent.insertBefore(i === 0 ? v : v.cloneNode(true), el);

});

});

return this;

},

prepend: function (content) {

insertContent(this, content, true);

return this;

},

prependTo: function (parent) {

insertContent(cash(parent), this, true);

return this;

},

remove: function () {

return this.each(function (v) {

if (!!v.parentNode) {

return v.parentNode.removeChild(v);

}

});

},

text: function (content) {

if (content === undefined) {

return this[0].textContent;

}

return this.each(function (v) {

return v.textContent = content;

});

}

});

var docEl = doc.documentElement;

fn.extend({

position: function () {

var el = this[0];

return {

left: el.offsetLeft,

top: el.offsetTop

};

},

offset: function () {

var rect = this[0].getBoundingClientRect();

return {

top: rect.top + win.pageYOffset - docEl.clientTop,

left: rect.left + win.pageXOffset - docEl.clientLeft

};

},

offsetParent: function () {

return cash(this[0].offsetParent);

}

});

fn.extend({

children: function (selector) {

var elems = [];

this.each(function (el) {

push.apply(elems, el.children);

});

elems = unique(elems);

return !selector ? elems : elems.filter(function (v) {

return matches(v, selector);

});

},

closest: function (selector) {

if (!selector || this.length < 1) {

return cash();

}

if (this.is(selector)) {

return this.filter(selector);

}

return this.parent().closest(selector);

},

is: function (selector) {

if (!selector) {

return false;

}

var match = false,

comparator = getCompareFunction(selector);

this.each(function (el) {

match = comparator(el, selector);

return !match;

});

return match;

},

find: function (selector) {

if (!selector || selector.nodeType) {

return cash(selector && this.has(selector).length ? selector : null);

}

var elems = [];

this.each(function (el) {

push.apply(elems, find(selector, el));

});

return unique(elems);

},

has: function (selector) {

var comparator = isString(selector) ? function (el) {

return find(selector, el).length !== 0;

} : function (el) {

return el.contains(selector);

};

return this.filter(comparator);

},

next: function () {

return cash(this[0].nextElementSibling);

},

not: function (selector) {

if (!selector) {

return this;

}

var comparator = getCompareFunction(selector);

return this.filter(function (el) {

return !comparator(el, selector);

});

},

parent: function () {

var result = [];

this.each(function (item) {

if (item && item.parentNode) {

result.push(item.parentNode);

}

});

return unique(result);

},

parents: function (selector) {

var last,

result = [];

this.each(function (item) {

last = item;

while (last && last.parentNode && last !== doc.body.parentNode) {

last = last.parentNode;

if (!selector || selector && matches(last, selector)) {

result.push(last);

}

}

});

return unique(result);

},

prev: function () {

return cash(this[0].previousElementSibling);

},

siblings: function (selector) {

var collection = this.parent().children(selector),

el = this[0];

return collection.filter(function (i) {

return i !== el;

});

}

});

return cash;

});

;

var Component = function () {

/\*\*

\* Generic constructor for all components

\* @constructor

\* @param {Element} el

\* @param {Object} options

\*/

function Component(classDef, el, options) {

\_classCallCheck(this, Component);

// Display error if el is valid HTML Element

if (!(el instanceof Element)) {

console.error(Error(el + ' is not an HTML Element'));

}

// If exists, destroy and reinitialize in child

var ins = classDef.getInstance(el);

if (!!ins) {

ins.destroy();

}

this.el = el;

this.$el = cash(el);

}

/\*\*

\* Initializes components

\* @param {class} classDef

\* @param {Element | NodeList | jQuery} els

\* @param {Object} options

\*/

\_createClass(Component, null, [{

key: "init",

value: function init(classDef, els, options) {

var instances = null;

if (els instanceof Element) {

instances = new classDef(els, options);

} else if (!!els && (els.jquery || els.cash || els instanceof NodeList)) {

var instancesArr = [];

for (var i = 0; i < els.length; i++) {

instancesArr.push(new classDef(els[i], options));

}

instances = instancesArr;

}

return instances;

}

}]);

return Component;

}();

; // Required for Meteor package, the use of window prevents export by Meteor

(function (window) {

if (window.Package) {

M = {};

} else {

window.M = {};

}

// Check for jQuery

M.jQueryLoaded = !!window.jQuery;

})(window);

// AMD

if (typeof define === 'function' && define.amd) {

define('M', [], function () {

return M;

});

// Common JS

} else if (typeof exports !== 'undefined' && !exports.nodeType) {

if (typeof module !== 'undefined' && !module.nodeType && module.exports) {

exports = module.exports = M;

}

exports.default = M;

}

M.version = '1.0.0';

M.keys = {

TAB: 9,

ENTER: 13,

ESC: 27,

ARROW\_UP: 38,

ARROW\_DOWN: 40

};

/\*\*

\* TabPress Keydown handler

\*/

M.tabPressed = false;

M.keyDown = false;

var docHandleKeydown = function (e) {

M.keyDown = true;

if (e.which === M.keys.TAB || e.which === M.keys.ARROW\_DOWN || e.which === M.keys.ARROW\_UP) {

M.tabPressed = true;

}

};

var docHandleKeyup = function (e) {

M.keyDown = false;

if (e.which === M.keys.TAB || e.which === M.keys.ARROW\_DOWN || e.which === M.keys.ARROW\_UP) {

M.tabPressed = false;

}

};

var docHandleFocus = function (e) {

if (M.keyDown) {

document.body.classList.add('keyboard-focused');

}

};

var docHandleBlur = function (e) {

document.body.classList.remove('keyboard-focused');

};

document.addEventListener('keydown', docHandleKeydown, true);

document.addEventListener('keyup', docHandleKeyup, true);

document.addEventListener('focus', docHandleFocus, true);

document.addEventListener('blur', docHandleBlur, true);

/\*\*

\* Initialize jQuery wrapper for plugin

\* @param {Class} plugin javascript class

\* @param {string} pluginName jQuery plugin name

\* @param {string} classRef Class reference name

\*/

M.initializeJqueryWrapper = function (plugin, pluginName, classRef) {

jQuery.fn[pluginName] = function (methodOrOptions) {

// Call plugin method if valid method name is passed in

if (plugin.prototype[methodOrOptions]) {

var params = Array.prototype.slice.call(arguments, 1);

// Getter methods

if (methodOrOptions.slice(0, 3) === 'get') {

var instance = this.first()[0][classRef];

return instance[methodOrOptions].apply(instance, params);

}

// Void methods

return this.each(function () {

var instance = this[classRef];

instance[methodOrOptions].apply(instance, params);

});

// Initialize plugin if options or no argument is passed in

} else if (typeof methodOrOptions === 'object' || !methodOrOptions) {

plugin.init(this, arguments[0]);

return this;

}

// Return error if an unrecognized method name is passed in

jQuery.error("Method " + methodOrOptions + " does not exist on jQuery." + pluginName);

};

};

/\*\*

\* Automatically initialize components

\* @param {Element} context DOM Element to search within for components

\*/

M.AutoInit = function (context) {

// Use document.body if no context is given

var root = !!context ? context : document.body;

var registry = {

Autocomplete: root.querySelectorAll('.autocomplete:not(.no-autoinit)'),

Carousel: root.querySelectorAll('.carousel:not(.no-autoinit)'),

Chips: root.querySelectorAll('.chips:not(.no-autoinit)'),

Collapsible: root.querySelectorAll('.collapsible:not(.no-autoinit)'),

Datepicker: root.querySelectorAll('.datepicker:not(.no-autoinit)'),

Dropdown: root.querySelectorAll('.dropdown-trigger:not(.no-autoinit)'),

Materialbox: root.querySelectorAll('.materialboxed:not(.no-autoinit)'),

Modal: root.querySelectorAll('.modal:not(.no-autoinit)'),

Parallax: root.querySelectorAll('.parallax:not(.no-autoinit)'),

Pushpin: root.querySelectorAll('.pushpin:not(.no-autoinit)'),

ScrollSpy: root.querySelectorAll('.scrollspy:not(.no-autoinit)'),

FormSelect: root.querySelectorAll('select:not(.no-autoinit)'),

Sidenav: root.querySelectorAll('.sidenav:not(.no-autoinit)'),

Tabs: root.querySelectorAll('.tabs:not(.no-autoinit)'),

TapTarget: root.querySelectorAll('.tap-target:not(.no-autoinit)'),

Timepicker: root.querySelectorAll('.timepicker:not(.no-autoinit)'),

Tooltip: root.querySelectorAll('.tooltipped:not(.no-autoinit)'),

FloatingActionButton: root.querySelectorAll('.fixed-action-btn:not(.no-autoinit)')

};

for (var pluginName in registry) {

var plugin = M[pluginName];

plugin.init(registry[pluginName]);

}

};

/\*\*

\* Generate approximated selector string for a jQuery object

\* @param {jQuery} obj jQuery object to be parsed

\* @returns {string}

\*/

M.objectSelectorString = function (obj) {

var tagStr = obj.prop('tagName') || '';

var idStr = obj.attr('id') || '';

var classStr = obj.attr('class') || '';

return (tagStr + idStr + classStr).replace(/\s/g, '');

};

// Unique Random ID

M.guid = function () {

function s4() {

return Math.floor((1 + Math.random()) \* 0x10000).toString(16).substring(1);

}

return function () {

return s4() + s4() + '-' + s4() + '-' + s4() + '-' + s4() + '-' + s4() + s4() + s4();

};

}();

/\*\*

\* Escapes hash from special characters

\* @param {string} hash String returned from this.hash

\* @returns {string}

\*/

M.escapeHash = function (hash) {

return hash.replace(/(:|\.|\[|\]|,|=|\/)/g, '\\$1');

};

M.elementOrParentIsFixed = function (element) {

var $element = $(element);

var $checkElements = $element.add($element.parents());

var isFixed = false;

$checkElements.each(function () {

if ($(this).css('position') === 'fixed') {

isFixed = true;

return false;

}

});

return isFixed;

};

/\*\*

\* @typedef {Object} Edges

\* @property {Boolean} top If the top edge was exceeded

\* @property {Boolean} right If the right edge was exceeded

\* @property {Boolean} bottom If the bottom edge was exceeded

\* @property {Boolean} left If the left edge was exceeded

\*/

/\*\*

\* @typedef {Object} Bounding

\* @property {Number} left left offset coordinate

\* @property {Number} top top offset coordinate

\* @property {Number} width

\* @property {Number} height

\*/

/\*\*

\* Escapes hash from special characters

\* @param {Element} container Container element that acts as the boundary

\* @param {Bounding} bounding element bounding that is being checked

\* @param {Number} offset offset from edge that counts as exceeding

\* @returns {Edges}

\*/

M.checkWithinContainer = function (container, bounding, offset) {

var edges = {

top: false,

right: false,

bottom: false,

left: false

};

var containerRect = container.getBoundingClientRect();

// If body element is smaller than viewport, use viewport height instead.

var containerBottom = container === document.body ? Math.max(containerRect.bottom, window.innerHeight) : containerRect.bottom;

var scrollLeft = container.scrollLeft;

var scrollTop = container.scrollTop;

var scrolledX = bounding.left - scrollLeft;

var scrolledY = bounding.top - scrollTop;

// Check for container and viewport for each edge

if (scrolledX < containerRect.left + offset || scrolledX < offset) {

edges.left = true;

}

if (scrolledX + bounding.width > containerRect.right - offset || scrolledX + bounding.width > window.innerWidth - offset) {

edges.right = true;

}

if (scrolledY < containerRect.top + offset || scrolledY < offset) {

edges.top = true;

}

if (scrolledY + bounding.height > containerBottom - offset || scrolledY + bounding.height > window.innerHeight - offset) {

edges.bottom = true;

}

return edges;

};

M.checkPossibleAlignments = function (el, container, bounding, offset) {

var canAlign = {

top: true,

right: true,

bottom: true,

left: true,

spaceOnTop: null,

spaceOnRight: null,

spaceOnBottom: null,

spaceOnLeft: null

};

var containerAllowsOverflow = getComputedStyle(container).overflow === 'visible';

var containerRect = container.getBoundingClientRect();

var containerHeight = Math.min(containerRect.height, window.innerHeight);

var containerWidth = Math.min(containerRect.width, window.innerWidth);

var elOffsetRect = el.getBoundingClientRect();

var scrollLeft = container.scrollLeft;

var scrollTop = container.scrollTop;

var scrolledX = bounding.left - scrollLeft;

var scrolledYTopEdge = bounding.top - scrollTop;

var scrolledYBottomEdge = bounding.top + elOffsetRect.height - scrollTop;

// Check for container and viewport for left

canAlign.spaceOnRight = !containerAllowsOverflow ? containerWidth - (scrolledX + bounding.width) : window.innerWidth - (elOffsetRect.left + bounding.width);

if (canAlign.spaceOnRight < 0) {

canAlign.left = false;

}

// Check for container and viewport for Right

canAlign.spaceOnLeft = !containerAllowsOverflow ? scrolledX - bounding.width + elOffsetRect.width : elOffsetRect.right - bounding.width;

if (canAlign.spaceOnLeft < 0) {

canAlign.right = false;

}

// Check for container and viewport for Top

canAlign.spaceOnBottom = !containerAllowsOverflow ? containerHeight - (scrolledYTopEdge + bounding.height + offset) : window.innerHeight - (elOffsetRect.top + bounding.height + offset);

if (canAlign.spaceOnBottom < 0) {

canAlign.top = false;

}

// Check for container and viewport for Bottom

canAlign.spaceOnTop = !containerAllowsOverflow ? scrolledYBottomEdge - (bounding.height - offset) : elOffsetRect.bottom - (bounding.height + offset);

if (canAlign.spaceOnTop < 0) {

canAlign.bottom = false;

}

return canAlign;

};

M.getOverflowParent = function (element) {

if (element == null) {

return null;

}

if (element === document.body || getComputedStyle(element).overflow !== 'visible') {

return element;

}

return M.getOverflowParent(element.parentElement);

};

/\*\*

\* Gets id of component from a trigger

\* @param {Element} trigger trigger

\* @returns {string}

\*/

M.getIdFromTrigger = function (trigger) {

var id = trigger.getAttribute('data-target');

if (!id) {

id = trigger.getAttribute('href');

if (id) {

id = id.slice(1);

} else {

id = '';

}

}

return id;

};

/\*\*

\* Multi browser support for document scroll top

\* @returns {Number}

\*/

M.getDocumentScrollTop = function () {

return window.pageYOffset || document.documentElement.scrollTop || document.body.scrollTop || 0;

};

/\*\*

\* Multi browser support for document scroll left

\* @returns {Number}

\*/

M.getDocumentScrollLeft = function () {

return window.pageXOffset || document.documentElement.scrollLeft || document.body.scrollLeft || 0;

};

/\*\*

\* @typedef {Object} Edges

\* @property {Boolean} top If the top edge was exceeded

\* @property {Boolean} right If the right edge was exceeded

\* @property {Boolean} bottom If the bottom edge was exceeded

\* @property {Boolean} left If the left edge was exceeded

\*/

/\*\*

\* @typedef {Object} Bounding

\* @property {Number} left left offset coordinate

\* @property {Number} top top offset coordinate

\* @property {Number} width

\* @property {Number} height

\*/

var getTime = Date.now || function () {

return new Date().getTime();

};

M.throttle = function (func, wait, options) {

var context = void 0,

args = void 0,

result = void 0;

var timeout = null;

var previous = 0;

options || (options = {});

var later = function () {

previous = options.leading === false ? 0 : getTime();

timeout = null;

result = func.apply(context, args);

context = args = null;

};

return function () {

var now = getTime();

if (!previous && options.leading === false) previous = now;

var remaining = wait - (now - previous);

context = this;

args = arguments;

if (remaining <= 0) {

clearTimeout(timeout);

timeout = null;

previous = now;

result = func.apply(context, args);

context = args = null;

} else if (!timeout && options.trailing !== false) {

timeout = setTimeout(later, remaining);

}

return result;

};

};

;

var $jscomp = { scope: {} };$jscomp.defineProperty = "function" == typeof Object.defineProperties ? Object.defineProperty : function (e, r, p) {

if (p.get || p.set) throw new TypeError("ES3 does not support getters and setters.");e != Array.prototype && e != Object.prototype && (e[r] = p.value);

};$jscomp.getGlobal = function (e) {

return "undefined" != typeof window && window === e ? e : "undefined" != typeof global && null != global ? global : e;

};$jscomp.global = $jscomp.getGlobal(this);$jscomp.SYMBOL\_PREFIX = "jscomp\_symbol\_";

$jscomp.initSymbol = function () {

$jscomp.initSymbol = function () {};$jscomp.global.Symbol || ($jscomp.global.Symbol = $jscomp.Symbol);

};$jscomp.symbolCounter\_ = 0;$jscomp.Symbol = function (e) {

return $jscomp.SYMBOL\_PREFIX + (e || "") + $jscomp.symbolCounter\_++;

};

$jscomp.initSymbolIterator = function () {

$jscomp.initSymbol();var e = $jscomp.global.Symbol.iterator;e || (e = $jscomp.global.Symbol.iterator = $jscomp.global.Symbol("iterator"));"function" != typeof Array.prototype[e] && $jscomp.defineProperty(Array.prototype, e, { configurable: !0, writable: !0, value: function () {

return $jscomp.arrayIterator(this);

} });$jscomp.initSymbolIterator = function () {};

};$jscomp.arrayIterator = function (e) {

var r = 0;return $jscomp.iteratorPrototype(function () {

return r < e.length ? { done: !1, value: e[r++] } : { done: !0 };

});

};

$jscomp.iteratorPrototype = function (e) {

$jscomp.initSymbolIterator();e = { next: e };e[$jscomp.global.Symbol.iterator] = function () {

return this;

};return e;

};$jscomp.array = $jscomp.array || {};$jscomp.iteratorFromArray = function (e, r) {

$jscomp.initSymbolIterator();e instanceof String && (e += "");var p = 0,

m = { next: function () {

if (p < e.length) {

var u = p++;return { value: r(u, e[u]), done: !1 };

}m.next = function () {

return { done: !0, value: void 0 };

};return m.next();

} };m[Symbol.iterator] = function () {

return m;

};return m;

};

$jscomp.polyfill = function (e, r, p, m) {

if (r) {

p = $jscomp.global;e = e.split(".");for (m = 0; m < e.length - 1; m++) {

var u = e[m];u in p || (p[u] = {});p = p[u];

}e = e[e.length - 1];m = p[e];r = r(m);r != m && null != r && $jscomp.defineProperty(p, e, { configurable: !0, writable: !0, value: r });

}

};$jscomp.polyfill("Array.prototype.keys", function (e) {

return e ? e : function () {

return $jscomp.iteratorFromArray(this, function (e) {

return e;

});

};

}, "es6-impl", "es3");var $jscomp$this = this;

(function (r) {

M.anime = r();

})(function () {

function e(a) {

if (!h.col(a)) try {

return document.querySelectorAll(a);

} catch (c) {}

}function r(a, c) {

for (var d = a.length, b = 2 <= arguments.length ? arguments[1] : void 0, f = [], n = 0; n < d; n++) {

if (n in a) {

var k = a[n];c.call(b, k, n, a) && f.push(k);

}

}return f;

}function p(a) {

return a.reduce(function (a, d) {

return a.concat(h.arr(d) ? p(d) : d);

}, []);

}function m(a) {

if (h.arr(a)) return a;

h.str(a) && (a = e(a) || a);return a instanceof NodeList || a instanceof HTMLCollection ? [].slice.call(a) : [a];

}function u(a, c) {

return a.some(function (a) {

return a === c;

});

}function C(a) {

var c = {},

d;for (d in a) {

c[d] = a[d];

}return c;

}function D(a, c) {

var d = C(a),

b;for (b in a) {

d[b] = c.hasOwnProperty(b) ? c[b] : a[b];

}return d;

}function z(a, c) {

var d = C(a),

b;for (b in c) {

d[b] = h.und(a[b]) ? c[b] : a[b];

}return d;

}function T(a) {

a = a.replace(/^#?([a-f\d])([a-f\d])([a-f\d])$/i, function (a, c, d, k) {

return c + c + d + d + k + k;

});var c = /^#?([a-f\d]{2})([a-f\d]{2})([a-f\d]{2})$/i.exec(a);

a = parseInt(c[1], 16);var d = parseInt(c[2], 16),

c = parseInt(c[3], 16);return "rgba(" + a + "," + d + "," + c + ",1)";

}function U(a) {

function c(a, c, b) {

0 > b && (b += 1);1 < b && --b;return b < 1 / 6 ? a + 6 \* (c - a) \* b : .5 > b ? c : b < 2 / 3 ? a + (c - a) \* (2 / 3 - b) \* 6 : a;

}var d = /hsl\((\d+),\s\*([\d.]+)%,\s\*([\d.]+)%\)/g.exec(a) || /hsla\((\d+),\s\*([\d.]+)%,\s\*([\d.]+)%,\s\*([\d.]+)\)/g.exec(a);a = parseInt(d[1]) / 360;var b = parseInt(d[2]) / 100,

f = parseInt(d[3]) / 100,

d = d[4] || 1;if (0 == b) f = b = a = f;else {

var n = .5 > f ? f \* (1 + b) : f + b - f \* b,

k = 2 \* f - n,

f = c(k, n, a + 1 / 3),

b = c(k, n, a);a = c(k, n, a - 1 / 3);

}return "rgba(" + 255 \* f + "," + 255 \* b + "," + 255 \* a + "," + d + ")";

}function y(a) {

if (a = /([\+\-]?[0-9#\.]+)(%|px|pt|em|rem|in|cm|mm|ex|ch|pc|vw|vh|vmin|vmax|deg|rad|turn)?$/.exec(a)) return a[2];

}function V(a) {

if (-1 < a.indexOf("translate") || "perspective" === a) return "px";if (-1 < a.indexOf("rotate") || -1 < a.indexOf("skew")) return "deg";

}function I(a, c) {

return h.fnc(a) ? a(c.target, c.id, c.total) : a;

}function E(a, c) {

if (c in a.style) return getComputedStyle(a).getPropertyValue(c.replace(/([a-z])([A-Z])/g, "$1-$2").toLowerCase()) || "0";

}function J(a, c) {

if (h.dom(a) && u(W, c)) return "transform";if (h.dom(a) && (a.getAttribute(c) || h.svg(a) && a[c])) return "attribute";if (h.dom(a) && "transform" !== c && E(a, c)) return "css";if (null != a[c]) return "object";

}function X(a, c) {

var d = V(c),

d = -1 < c.indexOf("scale") ? 1 : 0 + d;a = a.style.transform;if (!a) return d;for (var b = [], f = [], n = [], k = /(\w+)\((.+?)\)/g; b = k.exec(a);) {

f.push(b[1]), n.push(b[2]);

}a = r(n, function (a, b) {

return f[b] === c;

});return a.length ? a[0] : d;

}function K(a, c) {

switch (J(a, c)) {case "transform":

return X(a, c);case "css":

return E(a, c);case "attribute":

return a.getAttribute(c);}return a[c] || 0;

}function L(a, c) {

var d = /^(\\*=|\+=|-=)/.exec(a);if (!d) return a;var b = y(a) || 0;c = parseFloat(c);a = parseFloat(a.replace(d[0], ""));switch (d[0][0]) {case "+":

return c + a + b;case "-":

return c - a + b;case "\*":

return c \* a + b;}

}function F(a, c) {

return Math.sqrt(Math.pow(c.x - a.x, 2) + Math.pow(c.y - a.y, 2));

}function M(a) {

a = a.points;for (var c = 0, d, b = 0; b < a.numberOfItems; b++) {

var f = a.getItem(b);0 < b && (c += F(d, f));d = f;

}return c;

}function N(a) {

if (a.getTotalLength) return a.getTotalLength();switch (a.tagName.toLowerCase()) {case "circle":

return 2 \* Math.PI \* a.getAttribute("r");case "rect":

return 2 \* a.getAttribute("width") + 2 \* a.getAttribute("height");case "line":

return F({ x: a.getAttribute("x1"), y: a.getAttribute("y1") }, { x: a.getAttribute("x2"), y: a.getAttribute("y2") });case "polyline":

return M(a);case "polygon":

var c = a.points;return M(a) + F(c.getItem(c.numberOfItems - 1), c.getItem(0));}

}function Y(a, c) {

function d(b) {

b = void 0 === b ? 0 : b;return a.el.getPointAtLength(1 <= c + b ? c + b : 0);

}var b = d(),

f = d(-1),

n = d(1);switch (a.property) {case "x":

return b.x;case "y":

return b.y;

case "angle":

return 180 \* Math.atan2(n.y - f.y, n.x - f.x) / Math.PI;}

}function O(a, c) {

var d = /-?\d\*\.?\d+/g,

b;b = h.pth(a) ? a.totalLength : a;if (h.col(b)) {

if (h.rgb(b)) {

var f = /rgb\((\d+,\s\*[\d]+,\s\*[\d]+)\)/g.exec(b);b = f ? "rgba(" + f[1] + ",1)" : b;

} else b = h.hex(b) ? T(b) : h.hsl(b) ? U(b) : void 0;

} else f = (f = y(b)) ? b.substr(0, b.length - f.length) : b, b = c && !/\s/g.test(b) ? f + c : f;b += "";return { original: b, numbers: b.match(d) ? b.match(d).map(Number) : [0], strings: h.str(a) || c ? b.split(d) : [] };

}function P(a) {

a = a ? p(h.arr(a) ? a.map(m) : m(a)) : [];return r(a, function (a, d, b) {

return b.indexOf(a) === d;

});

}function Z(a) {

var c = P(a);return c.map(function (a, b) {

return { target: a, id: b, total: c.length };

});

}function aa(a, c) {

var d = C(c);if (h.arr(a)) {

var b = a.length;2 !== b || h.obj(a[0]) ? h.fnc(c.duration) || (d.duration = c.duration / b) : a = { value: a };

}return m(a).map(function (a, b) {

b = b ? 0 : c.delay;a = h.obj(a) && !h.pth(a) ? a : { value: a };h.und(a.delay) && (a.delay = b);return a;

}).map(function (a) {

return z(a, d);

});

}function ba(a, c) {

var d = {},

b;for (b in a) {

var f = I(a[b], c);h.arr(f) && (f = f.map(function (a) {

return I(a, c);

}), 1 === f.length && (f = f[0]));d[b] = f;

}d.duration = parseFloat(d.duration);d.delay = parseFloat(d.delay);return d;

}function ca(a) {

return h.arr(a) ? A.apply(this, a) : Q[a];

}function da(a, c) {

var d;return a.tweens.map(function (b) {

b = ba(b, c);var f = b.value,

e = K(c.target, a.name),

k = d ? d.to.original : e,

k = h.arr(f) ? f[0] : k,

w = L(h.arr(f) ? f[1] : f, k),

e = y(w) || y(k) || y(e);b.from = O(k, e);b.to = O(w, e);b.start = d ? d.end : a.offset;b.end = b.start + b.delay + b.duration;b.easing = ca(b.easing);b.elasticity = (1E3 - Math.min(Math.max(b.elasticity, 1), 999)) / 1E3;b.isPath = h.pth(f);b.isColor = h.col(b.from.original);b.isColor && (b.round = 1);return d = b;

});

}function ea(a, c) {

return r(p(a.map(function (a) {

return c.map(function (b) {

var c = J(a.target, b.name);if (c) {

var d = da(b, a);b = { type: c, property: b.name, animatable: a, tweens: d, duration: d[d.length - 1].end, delay: d[0].delay };

} else b = void 0;return b;

});

})), function (a) {

return !h.und(a);

});

}function R(a, c, d, b) {

var f = "delay" === a;return c.length ? (f ? Math.min : Math.max).apply(Math, c.map(function (b) {

return b[a];

})) : f ? b.delay : d.offset + b.delay + b.duration;

}function fa(a) {

var c = D(ga, a),

d = D(S, a),

b = Z(a.targets),

f = [],

e = z(c, d),

k;for (k in a) {

e.hasOwnProperty(k) || "targets" === k || f.push({ name: k, offset: e.offset, tweens: aa(a[k], d) });

}a = ea(b, f);return z(c, { children: [], animatables: b, animations: a, duration: R("duration", a, c, d), delay: R("delay", a, c, d) });

}function q(a) {

function c() {

return window.Promise && new Promise(function (a) {

return p = a;

});

}function d(a) {

return g.reversed ? g.duration - a : a;

}function b(a) {

for (var b = 0, c = {}, d = g.animations, f = d.length; b < f;) {

var e = d[b],

k = e.animatable,

h = e.tweens,

n = h.length - 1,

l = h[n];n && (l = r(h, function (b) {

return a < b.end;

})[0] || l);for (var h = Math.min(Math.max(a - l.start - l.delay, 0), l.duration) / l.duration, w = isNaN(h) ? 1 : l.easing(h, l.elasticity), h = l.to.strings, p = l.round, n = [], m = void 0, m = l.to.numbers.length, t = 0; t < m; t++) {

var x = void 0,

x = l.to.numbers[t],

q = l.from.numbers[t],

x = l.isPath ? Y(l.value, w \* x) : q + w \* (x - q);p && (l.isColor && 2 < t || (x = Math.round(x \* p) / p));n.push(x);

}if (l = h.length) for (m = h[0], w = 0; w < l; w++) {

p = h[w + 1], t = n[w], isNaN(t) || (m = p ? m + (t + p) : m + (t + " "));

} else m = n[0];ha[e.type](k.target, e.property, m, c, k.id);e.currentValue = m;b++;

}if (b = Object.keys(c).length) for (d = 0; d < b; d++) {

H || (H = E(document.body, "transform") ? "transform" : "-webkit-transform"), g.animatables[d].target.style[H] = c[d].join(" ");

}g.currentTime = a;g.progress = a / g.duration \* 100;

}function f(a) {

if (g[a]) g[a](g);

}function e() {

g.remaining && !0 !== g.remaining && g.remaining--;

}function k(a) {

var k = g.duration,

n = g.offset,

w = n + g.delay,

r = g.currentTime,

x = g.reversed,

q = d(a);if (g.children.length) {

var u = g.children,

v = u.length;

if (q >= g.currentTime) for (var G = 0; G < v; G++) {

u[G].seek(q);

} else for (; v--;) {

u[v].seek(q);

}

}if (q >= w || !k) g.began || (g.began = !0, f("begin")), f("run");if (q > n && q < k) b(q);else if (q <= n && 0 !== r && (b(0), x && e()), q >= k && r !== k || !k) b(k), x || e();f("update");a >= k && (g.remaining ? (t = h, "alternate" === g.direction && (g.reversed = !g.reversed)) : (g.pause(), g.completed || (g.completed = !0, f("complete"), "Promise" in window && (p(), m = c()))), l = 0);

}a = void 0 === a ? {} : a;var h,

t,

l = 0,

p = null,

m = c(),

g = fa(a);g.reset = function () {

var a = g.direction,

c = g.loop;g.currentTime = 0;g.progress = 0;g.paused = !0;g.began = !1;g.completed = !1;g.reversed = "reverse" === a;g.remaining = "alternate" === a && 1 === c ? 2 : c;b(0);for (a = g.children.length; a--;) {

g.children[a].reset();

}

};g.tick = function (a) {

h = a;t || (t = h);k((l + h - t) \* q.speed);

};g.seek = function (a) {

k(d(a));

};g.pause = function () {

var a = v.indexOf(g);-1 < a && v.splice(a, 1);g.paused = !0;

};g.play = function () {

g.paused && (g.paused = !1, t = 0, l = d(g.currentTime), v.push(g), B || ia());

};g.reverse = function () {

g.reversed = !g.reversed;t = 0;l = d(g.currentTime);

};g.restart = function () {

g.pause();

g.reset();g.play();

};g.finished = m;g.reset();g.autoplay && g.play();return g;

}var ga = { update: void 0, begin: void 0, run: void 0, complete: void 0, loop: 1, direction: "normal", autoplay: !0, offset: 0 },

S = { duration: 1E3, delay: 0, easing: "easeOutElastic", elasticity: 500, round: 0 },

W = "translateX translateY translateZ rotate rotateX rotateY rotateZ scale scaleX scaleY scaleZ skewX skewY perspective".split(" "),

H,

h = { arr: function (a) {

return Array.isArray(a);

}, obj: function (a) {

return -1 < Object.prototype.toString.call(a).indexOf("Object");

},

pth: function (a) {

return h.obj(a) && a.hasOwnProperty("totalLength");

}, svg: function (a) {

return a instanceof SVGElement;

}, dom: function (a) {

return a.nodeType || h.svg(a);

}, str: function (a) {

return "string" === typeof a;

}, fnc: function (a) {

return "function" === typeof a;

}, und: function (a) {

return "undefined" === typeof a;

}, hex: function (a) {

return (/(^#[0-9A-F]{6}$)|(^#[0-9A-F]{3}$)/i.test(a)

);

}, rgb: function (a) {

return (/^rgb/.test(a)

);

}, hsl: function (a) {

return (/^hsl/.test(a)

);

}, col: function (a) {

return h.hex(a) || h.rgb(a) || h.hsl(a);

} },

A = function () {

function a(a, d, b) {

return (((1 - 3 \* b + 3 \* d) \* a + (3 \* b - 6 \* d)) \* a + 3 \* d) \* a;

}return function (c, d, b, f) {

if (0 <= c && 1 >= c && 0 <= b && 1 >= b) {

var e = new Float32Array(11);if (c !== d || b !== f) for (var k = 0; 11 > k; ++k) {

e[k] = a(.1 \* k, c, b);

}return function (k) {

if (c === d && b === f) return k;if (0 === k) return 0;if (1 === k) return 1;for (var h = 0, l = 1; 10 !== l && e[l] <= k; ++l) {

h += .1;

}--l;var l = h + (k - e[l]) / (e[l + 1] - e[l]) \* .1,

n = 3 \* (1 - 3 \* b + 3 \* c) \* l \* l + 2 \* (3 \* b - 6 \* c) \* l + 3 \* c;if (.001 <= n) {

for (h = 0; 4 > h; ++h) {

n = 3 \* (1 - 3 \* b + 3 \* c) \* l \* l + 2 \* (3 \* b - 6 \* c) \* l + 3 \* c;if (0 === n) break;var m = a(l, c, b) - k,

l = l - m / n;

}k = l;

} else if (0 === n) k = l;else {

var l = h,

h = h + .1,

g = 0;do {

m = l + (h - l) / 2, n = a(m, c, b) - k, 0 < n ? h = m : l = m;

} while (1e-7 < Math.abs(n) && 10 > ++g);k = m;

}return a(k, d, f);

};

}

};

}(),

Q = function () {

function a(a, b) {

return 0 === a || 1 === a ? a : -Math.pow(2, 10 \* (a - 1)) \* Math.sin(2 \* (a - 1 - b / (2 \* Math.PI) \* Math.asin(1)) \* Math.PI / b);

}var c = "Quad Cubic Quart Quint Sine Expo Circ Back Elastic".split(" "),

d = { In: [[.55, .085, .68, .53], [.55, .055, .675, .19], [.895, .03, .685, .22], [.755, .05, .855, .06], [.47, 0, .745, .715], [.95, .05, .795, .035], [.6, .04, .98, .335], [.6, -.28, .735, .045], a], Out: [[.25, .46, .45, .94], [.215, .61, .355, 1], [.165, .84, .44, 1], [.23, 1, .32, 1], [.39, .575, .565, 1], [.19, 1, .22, 1], [.075, .82, .165, 1], [.175, .885, .32, 1.275], function (b, c) {

return 1 - a(1 - b, c);

}], InOut: [[.455, .03, .515, .955], [.645, .045, .355, 1], [.77, 0, .175, 1], [.86, 0, .07, 1], [.445, .05, .55, .95], [1, 0, 0, 1], [.785, .135, .15, .86], [.68, -.55, .265, 1.55], function (b, c) {

return .5 > b ? a(2 \* b, c) / 2 : 1 - a(-2 \* b + 2, c) / 2;

}] },

b = { linear: A(.25, .25, .75, .75) },

f = {},

e;for (e in d) {

f.type = e, d[f.type].forEach(function (a) {

return function (d, f) {

b["ease" + a.type + c[f]] = h.fnc(d) ? d : A.apply($jscomp$this, d);

};

}(f)), f = { type: f.type };

}return b;

}(),

ha = { css: function (a, c, d) {

return a.style[c] = d;

}, attribute: function (a, c, d) {

return a.setAttribute(c, d);

}, object: function (a, c, d) {

return a[c] = d;

}, transform: function (a, c, d, b, f) {

b[f] || (b[f] = []);b[f].push(c + "(" + d + ")");

} },

v = [],

B = 0,

ia = function () {

function a() {

B = requestAnimationFrame(c);

}function c(c) {

var b = v.length;if (b) {

for (var d = 0; d < b;) {

v[d] && v[d].tick(c), d++;

}a();

} else cancelAnimationFrame(B), B = 0;

}return a;

}();q.version = "2.2.0";q.speed = 1;q.running = v;q.remove = function (a) {

a = P(a);for (var c = v.length; c--;) {

for (var d = v[c], b = d.animations, f = b.length; f--;) {

u(a, b[f].animatable.target) && (b.splice(f, 1), b.length || d.pause());

}

}

};q.getValue = K;q.path = function (a, c) {

var d = h.str(a) ? e(a)[0] : a,

b = c || 100;return function (a) {

return { el: d, property: a, totalLength: N(d) \* (b / 100) };

};

};q.setDashoffset = function (a) {

var c = N(a);a.setAttribute("stroke-dasharray", c);return c;

};q.bezier = A;q.easings = Q;q.timeline = function (a) {

var c = q(a);c.pause();c.duration = 0;c.add = function (d) {

c.children.forEach(function (a) {

a.began = !0;a.completed = !0;

});m(d).forEach(function (b) {

var d = z(b, D(S, a || {}));d.targets = d.targets || a.targets;b = c.duration;var e = d.offset;d.autoplay = !1;d.direction = c.direction;d.offset = h.und(e) ? b : L(e, b);c.began = !0;c.completed = !0;c.seek(d.offset);d = q(d);d.began = !0;d.completed = !0;d.duration > b && (c.duration = d.duration);c.children.push(d);

});c.seek(0);c.reset();c.autoplay && c.restart();return c;

};return c;

};q.random = function (a, c) {

return Math.floor(Math.random() \* (c - a + 1)) + a;

};return q;

});

;(function ($, anim) {

'use strict';

var \_defaults = {

accordion: true,

onOpenStart: undefined,

onOpenEnd: undefined,

onCloseStart: undefined,

onCloseEnd: undefined,

inDuration: 300,

outDuration: 300

};

/\*\*

\* @class

\*

\*/

var Collapsible = function (\_Component) {

\_inherits(Collapsible, \_Component);

/\*\*

\* Construct Collapsible instance

\* @constructor

\* @param {Element} el

\* @param {Object} options

\*/

function Collapsible(el, options) {

\_classCallCheck(this, Collapsible);

var \_this3 = \_possibleConstructorReturn(this, (Collapsible.\_\_proto\_\_ || Object.getPrototypeOf(Collapsible)).call(this, Collapsible, el, options));

\_this3.el.M\_Collapsible = \_this3;

/\*\*

\* Options for the collapsible

\* @member Collapsible#options

\* @prop {Boolean} [accordion=false] - Type of the collapsible

\* @prop {Function} onOpenStart - Callback function called before collapsible is opened

\* @prop {Function} onOpenEnd - Callback function called after collapsible is opened

\* @prop {Function} onCloseStart - Callback function called before collapsible is closed

\* @prop {Function} onCloseEnd - Callback function called after collapsible is closed

\* @prop {Number} inDuration - Transition in duration in milliseconds.

\* @prop {Number} outDuration - Transition duration in milliseconds.

\*/

\_this3.options = $.extend({}, Collapsible.defaults, options);

// Setup tab indices

\_this3.$headers = \_this3.$el.children('li').children('.collapsible-header');

\_this3.$headers.attr('tabindex', 0);

\_this3.\_setupEventHandlers();

// Open first active

var $activeBodies = \_this3.$el.children('li.active').children('.collapsible-body');

if (\_this3.options.accordion) {

// Handle Accordion

$activeBodies.first().css('display', 'block');

} else {

// Handle Expandables

$activeBodies.css('display', 'block');

}

return \_this3;

}

\_createClass(Collapsible, [{

key: "destroy",

/\*\*

\* Teardown component

\*/

value: function destroy() {

this.\_removeEventHandlers();

this.el.M\_Collapsible = undefined;

}

/\*\*

\* Setup Event Handlers

\*/

}, {

key: "\_setupEventHandlers",

value: function \_setupEventHandlers() {

var \_this4 = this;

this.\_handleCollapsibleClickBound = this.\_handleCollapsibleClick.bind(this);

this.\_handleCollapsibleKeydownBound = this.\_handleCollapsibleKeydown.bind(this);

this.el.addEventListener('click', this.\_handleCollapsibleClickBound);

this.$headers.each(function (header) {

header.addEventListener('keydown', \_this4.\_handleCollapsibleKeydownBound);

});

}

/\*\*

\* Remove Event Handlers

\*/

}, {

key: "\_removeEventHandlers",

value: function \_removeEventHandlers() {

var \_this5 = this;

this.el.removeEventListener('click', this.\_handleCollapsibleClickBound);

this.$headers.each(function (header) {

header.removeEventListener('keydown', \_this5.\_handleCollapsibleKeydownBound);

});

}

/\*\*

\* Handle Collapsible Click

\* @param {Event} e

\*/

}, {

key: "\_handleCollapsibleClick",

value: function \_handleCollapsibleClick(e) {

var $header = $(e.target).closest('.collapsible-header');

if (e.target && $header.length) {

var $collapsible = $header.closest('.collapsible');

if ($collapsible[0] === this.el) {

var $collapsibleLi = $header.closest('li');

var $collapsibleLis = $collapsible.children('li');

var isActive = $collapsibleLi[0].classList.contains('active');

var index = $collapsibleLis.index($collapsibleLi);

if (isActive) {

this.close(index);

} else {

this.open(index);

}

}

}

}

/\*\*

\* Handle Collapsible Keydown

\* @param {Event} e

\*/

}, {

key: "\_handleCollapsibleKeydown",

value: function \_handleCollapsibleKeydown(e) {

if (e.keyCode === 13) {

this.\_handleCollapsibleClickBound(e);

}

}

/\*\*

\* Animate in collapsible slide

\* @param {Number} index - 0th index of slide

\*/

}, {

key: "\_animateIn",

value: function \_animateIn(index) {

var \_this6 = this;

var $collapsibleLi = this.$el.children('li').eq(index);

if ($collapsibleLi.length) {

var $body = $collapsibleLi.children('.collapsible-body');

anim.remove($body[0]);

$body.css({

display: 'block',

overflow: 'hidden',

height: 0,

paddingTop: '',

paddingBottom: ''

});

var pTop = $body.css('padding-top');

var pBottom = $body.css('padding-bottom');

var finalHeight = $body[0].scrollHeight;

$body.css({

paddingTop: 0,

paddingBottom: 0

});

anim({

targets: $body[0],

height: finalHeight,

paddingTop: pTop,

paddingBottom: pBottom,

duration: this.options.inDuration,

easing: 'easeInOutCubic',

complete: function (anim) {

$body.css({

overflow: '',

paddingTop: '',

paddingBottom: '',

height: ''

});

// onOpenEnd callback

if (typeof \_this6.options.onOpenEnd === 'function') {

\_this6.options.onOpenEnd.call(\_this6, $collapsibleLi[0]);

}

}

});

}

}

/\*\*

\* Animate out collapsible slide

\* @param {Number} index - 0th index of slide to open

\*/

}, {

key: "\_animateOut",

value: function \_animateOut(index) {

var \_this7 = this;

var $collapsibleLi = this.$el.children('li').eq(index);

if ($collapsibleLi.length) {

var $body = $collapsibleLi.children('.collapsible-body');

anim.remove($body[0]);

$body.css('overflow', 'hidden');

anim({

targets: $body[0],

height: 0,

paddingTop: 0,

paddingBottom: 0,

duration: this.options.outDuration,

easing: 'easeInOutCubic',

complete: function () {

$body.css({

height: '',

overflow: '',

padding: '',

display: ''

});

// onCloseEnd callback

if (typeof \_this7.options.onCloseEnd === 'function') {

\_this7.options.onCloseEnd.call(\_this7, $collapsibleLi[0]);

}

}

});

}

}

/\*\*

\* Open Collapsible

\* @param {Number} index - 0th index of slide

\*/

}, {

key: "open",

value: function open(index) {

var \_this8 = this;

var $collapsibleLi = this.$el.children('li').eq(index);

if ($collapsibleLi.length && !$collapsibleLi[0].classList.contains('active')) {

// onOpenStart callback

if (typeof this.options.onOpenStart === 'function') {

this.options.onOpenStart.call(this, $collapsibleLi[0]);

}

// Handle accordion behavior

if (this.options.accordion) {

var $collapsibleLis = this.$el.children('li');

var $activeLis = this.$el.children('li.active');

$activeLis.each(function (el) {

var index = $collapsibleLis.index($(el));

\_this8.close(index);

});

}

// Animate in

$collapsibleLi[0].classList.add('active');

this.\_animateIn(index);

}

}

/\*\*

\* Close Collapsible

\* @param {Number} index - 0th index of slide

\*/

}, {

key: "close",

value: function close(index) {

var $collapsibleLi = this.$el.children('li').eq(index);

if ($collapsibleLi.length && $collapsibleLi[0].classList.contains('active')) {

// onCloseStart callback

if (typeof this.options.onCloseStart === 'function') {

this.options.onCloseStart.call(this, $collapsibleLi[0]);

}

// Animate out

$collapsibleLi[0].classList.remove('active');

this.\_animateOut(index);

}

}

}], [{

key: "init",

value: function init(els, options) {

return \_get(Collapsible.\_\_proto\_\_ || Object.getPrototypeOf(Collapsible), "init", this).call(this, this, els, options);

}

/\*\*

\* Get Instance

\*/

}, {

key: "getInstance",

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_Collapsible;

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return Collapsible;

}(Component);

M.Collapsible = Collapsible;

if (M.jQueryLoaded) {

M.initializeJqueryWrapper(Collapsible, 'collapsible', 'M\_Collapsible');

}

})(cash, M.anime);

;(function ($, anim) {

'use strict';

var \_defaults = {

alignment: 'left',

autoFocus: true,

constrainWidth: true,

container: null,

coverTrigger: true,

closeOnClick: true,

hover: false,

inDuration: 150,

outDuration: 250,

onOpenStart: null,

onOpenEnd: null,

onCloseStart: null,

onCloseEnd: null,

onItemClick: null

};

/\*\*

\* @class

\*/

var Dropdown = function (\_Component2) {

\_inherits(Dropdown, \_Component2);

function Dropdown(el, options) {

\_classCallCheck(this, Dropdown);

var \_this9 = \_possibleConstructorReturn(this, (Dropdown.\_\_proto\_\_ || Object.getPrototypeOf(Dropdown)).call(this, Dropdown, el, options));

\_this9.el.M\_Dropdown = \_this9;

Dropdown.\_dropdowns.push(\_this9);

\_this9.id = M.getIdFromTrigger(el);

\_this9.dropdownEl = document.getElementById(\_this9.id);

\_this9.$dropdownEl = $(\_this9.dropdownEl);

/\*\*

\* Options for the dropdown

\* @member Dropdown#options

\* @prop {String} [alignment='left'] - Edge which the dropdown is aligned to

\* @prop {Boolean} [autoFocus=true] - Automatically focus dropdown el for keyboard

\* @prop {Boolean} [constrainWidth=true] - Constrain width to width of the button

\* @prop {Element} container - Container element to attach dropdown to (optional)

\* @prop {Boolean} [coverTrigger=true] - Place dropdown over trigger

\* @prop {Boolean} [closeOnClick=true] - Close on click of dropdown item

\* @prop {Boolean} [hover=false] - Open dropdown on hover

\* @prop {Number} [inDuration=150] - Duration of open animation in ms

\* @prop {Number} [outDuration=250] - Duration of close animation in ms

\* @prop {Function} onOpenStart - Function called when dropdown starts opening

\* @prop {Function} onOpenEnd - Function called when dropdown finishes opening

\* @prop {Function} onCloseStart - Function called when dropdown starts closing

\* @prop {Function} onCloseEnd - Function called when dropdown finishes closing

\*/

\_this9.options = $.extend({}, Dropdown.defaults, options);

/\*\*

\* Describes open/close state of dropdown

\* @type {Boolean}

\*/

\_this9.isOpen = false;

/\*\*

\* Describes if dropdown content is scrollable

\* @type {Boolean}

\*/

\_this9.isScrollable = false;

/\*\*

\* Describes if touch moving on dropdown content

\* @type {Boolean}

\*/

\_this9.isTouchMoving = false;

\_this9.focusedIndex = -1;

\_this9.filterQuery = [];

// Move dropdown-content after dropdown-trigger

if (!!\_this9.options.container) {

$(\_this9.options.container).append(\_this9.dropdownEl);

} else {

\_this9.$el.after(\_this9.dropdownEl);

}

\_this9.\_makeDropdownFocusable();

\_this9.\_resetFilterQueryBound = \_this9.\_resetFilterQuery.bind(\_this9);

\_this9.\_handleDocumentClickBound = \_this9.\_handleDocumentClick.bind(\_this9);

\_this9.\_handleDocumentTouchmoveBound = \_this9.\_handleDocumentTouchmove.bind(\_this9);

\_this9.\_handleDropdownClickBound = \_this9.\_handleDropdownClick.bind(\_this9);

\_this9.\_handleDropdownKeydownBound = \_this9.\_handleDropdownKeydown.bind(\_this9);

\_this9.\_handleTriggerKeydownBound = \_this9.\_handleTriggerKeydown.bind(\_this9);

\_this9.\_setupEventHandlers();

return \_this9;

}

\_createClass(Dropdown, [{

key: "destroy",

/\*\*

\* Teardown component

\*/

value: function destroy() {

this.\_resetDropdownStyles();

this.\_removeEventHandlers();

Dropdown.\_dropdowns.splice(Dropdown.\_dropdowns.indexOf(this), 1);

this.el.M\_Dropdown = undefined;

}

/\*\*

\* Setup Event Handlers

\*/

}, {

key: "\_setupEventHandlers",

value: function \_setupEventHandlers() {

// Trigger keydown handler

this.el.addEventListener('keydown', this.\_handleTriggerKeydownBound);

// Item click handler

this.dropdownEl.addEventListener('click', this.\_handleDropdownClickBound);

// Hover event handlers

if (this.options.hover) {

this.\_handleMouseEnterBound = this.\_handleMouseEnter.bind(this);

this.el.addEventListener('mouseenter', this.\_handleMouseEnterBound);

this.\_handleMouseLeaveBound = this.\_handleMouseLeave.bind(this);

this.el.addEventListener('mouseleave', this.\_handleMouseLeaveBound);

this.dropdownEl.addEventListener('mouseleave', this.\_handleMouseLeaveBound);

// Click event handlers

} else {

this.\_handleClickBound = this.\_handleClick.bind(this);

this.el.addEventListener('click', this.\_handleClickBound);

}

}

/\*\*

\* Remove Event Handlers

\*/

}, {

key: "\_removeEventHandlers",

value: function \_removeEventHandlers() {

this.el.removeEventListener('keydown', this.\_handleTriggerKeydownBound);

this.dropdownEl.removeEventListener('click', this.\_handleDropdownClickBound);

if (this.options.hover) {

this.el.removeEventListener('mouseenter', this.\_handleMouseEnterBound);

this.el.removeEventListener('mouseleave', this.\_handleMouseLeaveBound);

this.dropdownEl.removeEventListener('mouseleave', this.\_handleMouseLeaveBound);

} else {

this.el.removeEventListener('click', this.\_handleClickBound);

}

}

}, {

key: "\_setupTemporaryEventHandlers",

value: function \_setupTemporaryEventHandlers() {

// Use capture phase event handler to prevent click

document.body.addEventListener('click', this.\_handleDocumentClickBound, true);

document.body.addEventListener('touchend', this.\_handleDocumentClickBound);

document.body.addEventListener('touchmove', this.\_handleDocumentTouchmoveBound);

this.dropdownEl.addEventListener('keydown', this.\_handleDropdownKeydownBound);

}

}, {

key: "\_removeTemporaryEventHandlers",

value: function \_removeTemporaryEventHandlers() {

// Use capture phase event handler to prevent click

document.body.removeEventListener('click', this.\_handleDocumentClickBound, true);

document.body.removeEventListener('touchend', this.\_handleDocumentClickBound);

document.body.removeEventListener('touchmove', this.\_handleDocumentTouchmoveBound);

this.dropdownEl.removeEventListener('keydown', this.\_handleDropdownKeydownBound);

}

}, {

key: "\_handleClick",

value: function \_handleClick(e) {

e.preventDefault();

this.open();

}

}, {

key: "\_handleMouseEnter",

value: function \_handleMouseEnter() {

this.open();

}

}, {

key: "\_handleMouseLeave",

value: function \_handleMouseLeave(e) {

var toEl = e.toElement || e.relatedTarget;

var leaveToDropdownContent = !!$(toEl).closest('.dropdown-content').length;

var leaveToActiveDropdownTrigger = false;

var $closestTrigger = $(toEl).closest('.dropdown-trigger');

if ($closestTrigger.length && !!$closestTrigger[0].M\_Dropdown && $closestTrigger[0].M\_Dropdown.isOpen) {

leaveToActiveDropdownTrigger = true;

}

// Close hover dropdown if mouse did not leave to either active dropdown-trigger or dropdown-content

if (!leaveToActiveDropdownTrigger && !leaveToDropdownContent) {

this.close();

}

}

}, {

key: "\_handleDocumentClick",

value: function \_handleDocumentClick(e) {

var \_this10 = this;

var $target = $(e.target);

if (this.options.closeOnClick && $target.closest('.dropdown-content').length && !this.isTouchMoving) {

// isTouchMoving to check if scrolling on mobile.

setTimeout(function () {

\_this10.close();

}, 0);

} else if ($target.closest('.dropdown-trigger').length || !$target.closest('.dropdown-content').length) {

setTimeout(function () {

\_this10.close();

}, 0);

}

this.isTouchMoving = false;

}

}, {

key: "\_handleTriggerKeydown",

value: function \_handleTriggerKeydown(e) {

// ARROW DOWN OR ENTER WHEN SELECT IS CLOSED - open Dropdown

if ((e.which === M.keys.ARROW\_DOWN || e.which === M.keys.ENTER) && !this.isOpen) {

e.preventDefault();

this.open();

}

}

/\*\*

\* Handle Document Touchmove

\* @param {Event} e

\*/

}, {

key: "\_handleDocumentTouchmove",

value: function \_handleDocumentTouchmove(e) {

var $target = $(e.target);

if ($target.closest('.dropdown-content').length) {

this.isTouchMoving = true;

}

}

/\*\*

\* Handle Dropdown Click

\* @param {Event} e

\*/

}, {

key: "\_handleDropdownClick",

value: function \_handleDropdownClick(e) {

// onItemClick callback

if (typeof this.options.onItemClick === 'function') {

var itemEl = $(e.target).closest('li')[0];

this.options.onItemClick.call(this, itemEl);

}

}

/\*\*

\* Handle Dropdown Keydown

\* @param {Event} e

\*/

}, {

key: "\_handleDropdownKeydown",

value: function \_handleDropdownKeydown(e) {

if (e.which === M.keys.TAB) {

e.preventDefault();

this.close();

// Navigate down dropdown list

} else if ((e.which === M.keys.ARROW\_DOWN || e.which === M.keys.ARROW\_UP) && this.isOpen) {

e.preventDefault();

var direction = e.which === M.keys.ARROW\_DOWN ? 1 : -1;

var newFocusedIndex = this.focusedIndex;

var foundNewIndex = false;

do {

newFocusedIndex = newFocusedIndex + direction;

if (!!this.dropdownEl.children[newFocusedIndex] && this.dropdownEl.children[newFocusedIndex].tabIndex !== -1) {

foundNewIndex = true;

break;

}

} while (newFocusedIndex < this.dropdownEl.children.length && newFocusedIndex >= 0);

if (foundNewIndex) {

this.focusedIndex = newFocusedIndex;

this.\_focusFocusedItem();

}

// ENTER selects choice on focused item

} else if (e.which === M.keys.ENTER && this.isOpen) {

// Search for <a> and <button>

var focusedElement = this.dropdownEl.children[this.focusedIndex];

var $activatableElement = $(focusedElement).find('a, button').first();

// Click a or button tag if exists, otherwise click li tag

if (!!$activatableElement.length) {

$activatableElement[0].click();

} else if (!!focusedElement) {

focusedElement.click();

}

// Close dropdown on ESC

} else if (e.which === M.keys.ESC && this.isOpen) {

e.preventDefault();

this.close();

}

// CASE WHEN USER TYPE LETTERS

var letter = String.fromCharCode(e.which).toLowerCase(),

nonLetters = [9, 13, 27, 38, 40];

if (letter && nonLetters.indexOf(e.which) === -1) {

this.filterQuery.push(letter);

var string = this.filterQuery.join(''),

newOptionEl = $(this.dropdownEl).find('li').filter(function (el) {

return $(el).text().toLowerCase().indexOf(string) === 0;

})[0];

if (newOptionEl) {

this.focusedIndex = $(newOptionEl).index();

this.\_focusFocusedItem();

}

}

this.filterTimeout = setTimeout(this.\_resetFilterQueryBound, 1000);

}

/\*\*

\* Setup dropdown

\*/

}, {

key: "\_resetFilterQuery",

value: function \_resetFilterQuery() {

this.filterQuery = [];

}

}, {

key: "\_resetDropdownStyles",

value: function \_resetDropdownStyles() {

this.$dropdownEl.css({

display: '',

width: '',

height: '',

left: '',

top: '',

'transform-origin': '',

transform: '',

opacity: ''

});

}

}, {

key: "\_makeDropdownFocusable",

value: function \_makeDropdownFocusable() {

// Needed for arrow key navigation

this.dropdownEl.tabIndex = 0;

// Only set tabindex if it hasn't been set by user

$(this.dropdownEl).children().each(function (el) {

if (!el.getAttribute('tabindex')) {

el.setAttribute('tabindex', 0);

}

});

}

}, {

key: "\_focusFocusedItem",

value: function \_focusFocusedItem() {

if (this.focusedIndex >= 0 && this.focusedIndex < this.dropdownEl.children.length && this.options.autoFocus) {

this.dropdownEl.children[this.focusedIndex].focus();

}

}

}, {

key: "\_getDropdownPosition",

value: function \_getDropdownPosition() {

var offsetParentBRect = this.el.offsetParent.getBoundingClientRect();

var triggerBRect = this.el.getBoundingClientRect();

var dropdownBRect = this.dropdownEl.getBoundingClientRect();

var idealHeight = dropdownBRect.height;

var idealWidth = dropdownBRect.width;

var idealXPos = triggerBRect.left - dropdownBRect.left;

var idealYPos = triggerBRect.top - dropdownBRect.top;

var dropdownBounds = {

left: idealXPos,

top: idealYPos,

height: idealHeight,

width: idealWidth

};

// Countainer here will be closest ancestor with overflow: hidden

var closestOverflowParent = !!this.dropdownEl.offsetParent ? this.dropdownEl.offsetParent : this.dropdownEl.parentNode;

var alignments = M.checkPossibleAlignments(this.el, closestOverflowParent, dropdownBounds, this.options.coverTrigger ? 0 : triggerBRect.height);

var verticalAlignment = 'top';

var horizontalAlignment = this.options.alignment;

idealYPos += this.options.coverTrigger ? 0 : triggerBRect.height;

// Reset isScrollable

this.isScrollable = false;

if (!alignments.top) {

if (alignments.bottom) {

verticalAlignment = 'bottom';

} else {

this.isScrollable = true;

// Determine which side has most space and cutoff at correct height

if (alignments.spaceOnTop > alignments.spaceOnBottom) {

verticalAlignment = 'bottom';

idealHeight += alignments.spaceOnTop;

idealYPos -= alignments.spaceOnTop;

} else {

idealHeight += alignments.spaceOnBottom;

}

}

}

// If preferred horizontal alignment is possible

if (!alignments[horizontalAlignment]) {

var oppositeAlignment = horizontalAlignment === 'left' ? 'right' : 'left';

if (alignments[oppositeAlignment]) {

horizontalAlignment = oppositeAlignment;

} else {

// Determine which side has most space and cutoff at correct height

if (alignments.spaceOnLeft > alignments.spaceOnRight) {

horizontalAlignment = 'right';

idealWidth += alignments.spaceOnLeft;

idealXPos -= alignments.spaceOnLeft;

} else {

horizontalAlignment = 'left';

idealWidth += alignments.spaceOnRight;

}

}

}

if (verticalAlignment === 'bottom') {

idealYPos = idealYPos - dropdownBRect.height + (this.options.coverTrigger ? triggerBRect.height : 0);

}

if (horizontalAlignment === 'right') {

idealXPos = idealXPos - dropdownBRect.width + triggerBRect.width;

}

return {

x: idealXPos,

y: idealYPos,

verticalAlignment: verticalAlignment,

horizontalAlignment: horizontalAlignment,

height: idealHeight,

width: idealWidth

};

}

/\*\*

\* Animate in dropdown

\*/

}, {

key: "\_animateIn",

value: function \_animateIn() {

var \_this11 = this;

anim.remove(this.dropdownEl);

anim({

targets: this.dropdownEl,

opacity: {

value: [0, 1],

easing: 'easeOutQuad'

},

scaleX: [0.3, 1],

scaleY: [0.3, 1],

duration: this.options.inDuration,

easing: 'easeOutQuint',

complete: function (anim) {

if (\_this11.options.autoFocus) {

\_this11.dropdownEl.focus();

}

// onOpenEnd callback

if (typeof \_this11.options.onOpenEnd === 'function') {

\_this11.options.onOpenEnd.call(\_this11, \_this11.el);

}

}

});

}

/\*\*

\* Animate out dropdown

\*/

}, {

key: "\_animateOut",

value: function \_animateOut() {

var \_this12 = this;

anim.remove(this.dropdownEl);

anim({

targets: this.dropdownEl,

opacity: {

value: 0,

easing: 'easeOutQuint'

},

scaleX: 0.3,

scaleY: 0.3,

duration: this.options.outDuration,

easing: 'easeOutQuint',

complete: function (anim) {

\_this12.\_resetDropdownStyles();

// onCloseEnd callback

if (typeof \_this12.options.onCloseEnd === 'function') {

\_this12.options.onCloseEnd.call(\_this12, \_this12.el);

}

}

});

}

/\*\*

\* Place dropdown

\*/

}, {

key: "\_placeDropdown",

value: function \_placeDropdown() {

// Set width before calculating positionInfo

var idealWidth = this.options.constrainWidth ? this.el.getBoundingClientRect().width : this.dropdownEl.getBoundingClientRect().width;

this.dropdownEl.style.width = idealWidth + 'px';

var positionInfo = this.\_getDropdownPosition();

this.dropdownEl.style.left = positionInfo.x + 'px';

this.dropdownEl.style.top = positionInfo.y + 'px';

this.dropdownEl.style.height = positionInfo.height + 'px';

this.dropdownEl.style.width = positionInfo.width + 'px';

this.dropdownEl.style.transformOrigin = (positionInfo.horizontalAlignment === 'left' ? '0' : '100%') + " " + (positionInfo.verticalAlignment === 'top' ? '0' : '100%');

}

/\*\*

\* Open Dropdown

\*/

}, {

key: "open",

value: function open() {

if (this.isOpen) {

return;

}

this.isOpen = true;

// onOpenStart callback

if (typeof this.options.onOpenStart === 'function') {

this.options.onOpenStart.call(this, this.el);

}

// Reset styles

this.\_resetDropdownStyles();

this.dropdownEl.style.display = 'block';

this.\_placeDropdown();

this.\_animateIn();

this.\_setupTemporaryEventHandlers();

}

/\*\*

\* Close Dropdown

\*/

}, {

key: "close",

value: function close() {

if (!this.isOpen) {

return;

}

this.isOpen = false;

this.focusedIndex = -1;

// onCloseStart callback

if (typeof this.options.onCloseStart === 'function') {

this.options.onCloseStart.call(this, this.el);

}

this.\_animateOut();

this.\_removeTemporaryEventHandlers();

if (this.options.autoFocus) {

this.el.focus();

}

}

/\*\*

\* Recalculate dimensions

\*/

}, {

key: "recalculateDimensions",

value: function recalculateDimensions() {

if (this.isOpen) {

this.$dropdownEl.css({

width: '',

height: '',

left: '',

top: '',

'transform-origin': ''

});

this.\_placeDropdown();

}

}

}], [{

key: "init",

value: function init(els, options) {

return \_get(Dropdown.\_\_proto\_\_ || Object.getPrototypeOf(Dropdown), "init", this).call(this, this, els, options);

}

/\*\*

\* Get Instance

\*/

}, {

key: "getInstance",

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_Dropdown;

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return Dropdown;

}(Component);

/\*\*

\* @static

\* @memberof Dropdown

\*/

Dropdown.\_dropdowns = [];

M.Dropdown = Dropdown;

if (M.jQueryLoaded) {

M.initializeJqueryWrapper(Dropdown, 'dropdown', 'M\_Dropdown');

}

})(cash, M.anime);

;(function ($, anim) {

'use strict';

var \_defaults = {

opacity: 0.5,

inDuration: 250,

outDuration: 250,

onOpenStart: null,

onOpenEnd: null,

onCloseStart: null,

onCloseEnd: null,

preventScrolling: true,

dismissible: true,

startingTop: '4%',

endingTop: '10%'

};

/\*\*

\* @class

\*

\*/

var Modal = function (\_Component3) {

\_inherits(Modal, \_Component3);

/\*\*

\* Construct Modal instance and set up overlay

\* @constructor

\* @param {Element} el

\* @param {Object} options

\*/

function Modal(el, options) {

\_classCallCheck(this, Modal);

var \_this13 = \_possibleConstructorReturn(this, (Modal.\_\_proto\_\_ || Object.getPrototypeOf(Modal)).call(this, Modal, el, options));

\_this13.el.M\_Modal = \_this13;

/\*\*

\* Options for the modal

\* @member Modal#options

\* @prop {Number} [opacity=0.5] - Opacity of the modal overlay

\* @prop {Number} [inDuration=250] - Length in ms of enter transition

\* @prop {Number} [outDuration=250] - Length in ms of exit transition

\* @prop {Function} onOpenStart - Callback function called before modal is opened

\* @prop {Function} onOpenEnd - Callback function called after modal is opened

\* @prop {Function} onCloseStart - Callback function called before modal is closed

\* @prop {Function} onCloseEnd - Callback function called after modal is closed

\* @prop {Boolean} [dismissible=true] - Allow modal to be dismissed by keyboard or overlay click

\* @prop {String} [startingTop='4%'] - startingTop

\* @prop {String} [endingTop='10%'] - endingTop

\*/

\_this13.options = $.extend({}, Modal.defaults, options);

/\*\*

\* Describes open/close state of modal

\* @type {Boolean}

\*/

\_this13.isOpen = false;

\_this13.id = \_this13.$el.attr('id');

\_this13.\_openingTrigger = undefined;

\_this13.$overlay = $('<div class="modal-overlay"></div>');

\_this13.el.tabIndex = 0;

\_this13.\_nthModalOpened = 0;

Modal.\_count++;

\_this13.\_setupEventHandlers();

return \_this13;

}

\_createClass(Modal, [{

key: "destroy",

/\*\*

\* Teardown component

\*/

value: function destroy() {

Modal.\_count--;

this.\_removeEventHandlers();

this.el.removeAttribute('style');

this.$overlay.remove();

this.el.M\_Modal = undefined;

}

/\*\*

\* Setup Event Handlers

\*/

}, {

key: "\_setupEventHandlers",

value: function \_setupEventHandlers() {

this.\_handleOverlayClickBound = this.\_handleOverlayClick.bind(this);

this.\_handleModalCloseClickBound = this.\_handleModalCloseClick.bind(this);

if (Modal.\_count === 1) {

document.body.addEventListener('click', this.\_handleTriggerClick);

}

this.$overlay[0].addEventListener('click', this.\_handleOverlayClickBound);

this.el.addEventListener('click', this.\_handleModalCloseClickBound);

}

/\*\*

\* Remove Event Handlers

\*/

}, {

key: "\_removeEventHandlers",

value: function \_removeEventHandlers() {

if (Modal.\_count === 0) {

document.body.removeEventListener('click', this.\_handleTriggerClick);

}

this.$overlay[0].removeEventListener('click', this.\_handleOverlayClickBound);

this.el.removeEventListener('click', this.\_handleModalCloseClickBound);

}

/\*\*

\* Handle Trigger Click

\* @param {Event} e

\*/

}, {

key: "\_handleTriggerClick",

value: function \_handleTriggerClick(e) {

var $trigger = $(e.target).closest('.modal-trigger');

if ($trigger.length) {

var modalId = M.getIdFromTrigger($trigger[0]);

var modalInstance = document.getElementById(modalId).M\_Modal;

if (modalInstance) {

modalInstance.open($trigger);

}

e.preventDefault();

}

}

/\*\*

\* Handle Overlay Click

\*/

}, {

key: "\_handleOverlayClick",

value: function \_handleOverlayClick() {

if (this.options.dismissible) {

this.close();

}

}

/\*\*

\* Handle Modal Close Click

\* @param {Event} e

\*/

}, {

key: "\_handleModalCloseClick",

value: function \_handleModalCloseClick(e) {

var $closeTrigger = $(e.target).closest('.modal-close');

if ($closeTrigger.length) {

this.close();

}

}

/\*\*

\* Handle Keydown

\* @param {Event} e

\*/

}, {

key: "\_handleKeydown",

value: function \_handleKeydown(e) {

// ESC key

if (e.keyCode === 27 && this.options.dismissible) {

this.close();

}

}

/\*\*

\* Handle Focus

\* @param {Event} e

\*/

}, {

key: "\_handleFocus",

value: function \_handleFocus(e) {

// Only trap focus if this modal is the last model opened (prevents loops in nested modals).

if (!this.el.contains(e.target) && this.\_nthModalOpened === Modal.\_modalsOpen) {

this.el.focus();

}

}

/\*\*

\* Animate in modal

\*/

}, {

key: "\_animateIn",

value: function \_animateIn() {

var \_this14 = this;

// Set initial styles

$.extend(this.el.style, {

display: 'block',

opacity: 0

});

$.extend(this.$overlay[0].style, {

display: 'block',

opacity: 0

});

// Animate overlay

anim({

targets: this.$overlay[0],

opacity: this.options.opacity,

duration: this.options.inDuration,

easing: 'easeOutQuad'

});

// Define modal animation options

var enterAnimOptions = {

targets: this.el,

duration: this.options.inDuration,

easing: 'easeOutCubic',

// Handle modal onOpenEnd callback

complete: function () {

if (typeof \_this14.options.onOpenEnd === 'function') {

\_this14.options.onOpenEnd.call(\_this14, \_this14.el, \_this14.\_openingTrigger);

}

}

};

// Bottom sheet animation

if (this.el.classList.contains('bottom-sheet')) {

$.extend(enterAnimOptions, {

bottom: 0,

opacity: 1

});

anim(enterAnimOptions);

// Normal modal animation

} else {

$.extend(enterAnimOptions, {

top: [this.options.startingTop, this.options.endingTop],

opacity: 1,

scaleX: [0.8, 1],

scaleY: [0.8, 1]

});

anim(enterAnimOptions);

}

}

/\*\*

\* Animate out modal

\*/

}, {

key: "\_animateOut",

value: function \_animateOut() {

var \_this15 = this;

// Animate overlay

anim({

targets: this.$overlay[0],

opacity: 0,

duration: this.options.outDuration,

easing: 'easeOutQuart'

});

// Define modal animation options

var exitAnimOptions = {

targets: this.el,

duration: this.options.outDuration,

easing: 'easeOutCubic',

// Handle modal ready callback

complete: function () {

\_this15.el.style.display = 'none';

\_this15.$overlay.remove();

// Call onCloseEnd callback

if (typeof \_this15.options.onCloseEnd === 'function') {

\_this15.options.onCloseEnd.call(\_this15, \_this15.el);

}

}

};

// Bottom sheet animation

if (this.el.classList.contains('bottom-sheet')) {

$.extend(exitAnimOptions, {

bottom: '-100%',

opacity: 0

});

anim(exitAnimOptions);

// Normal modal animation

} else {

$.extend(exitAnimOptions, {

top: [this.options.endingTop, this.options.startingTop],

opacity: 0,

scaleX: 0.8,

scaleY: 0.8

});

anim(exitAnimOptions);

}

}

/\*\*

\* Open Modal

\* @param {cash} [$trigger]

\*/

}, {

key: "open",

value: function open($trigger) {

if (this.isOpen) {

return;

}

this.isOpen = true;

Modal.\_modalsOpen++;

this.\_nthModalOpened = Modal.\_modalsOpen;

// Set Z-Index based on number of currently open modals

this.$overlay[0].style.zIndex = 1000 + Modal.\_modalsOpen \* 2;

this.el.style.zIndex = 1000 + Modal.\_modalsOpen \* 2 + 1;

// Set opening trigger, undefined indicates modal was opened by javascript

this.\_openingTrigger = !!$trigger ? $trigger[0] : undefined;

// onOpenStart callback

if (typeof this.options.onOpenStart === 'function') {

this.options.onOpenStart.call(this, this.el, this.\_openingTrigger);

}

if (this.options.preventScrolling) {

document.body.style.overflow = 'hidden';

}

this.el.classList.add('open');

this.el.insertAdjacentElement('afterend', this.$overlay[0]);

if (this.options.dismissible) {

this.\_handleKeydownBound = this.\_handleKeydown.bind(this);

this.\_handleFocusBound = this.\_handleFocus.bind(this);

document.addEventListener('keydown', this.\_handleKeydownBound);

document.addEventListener('focus', this.\_handleFocusBound, true);

}

anim.remove(this.el);

anim.remove(this.$overlay[0]);

this.\_animateIn();

// Focus modal

this.el.focus();

return this;

}

/\*\*

\* Close Modal

\*/

}, {

key: "close",

value: function close() {

if (!this.isOpen) {

return;

}

this.isOpen = false;

Modal.\_modalsOpen--;

this.\_nthModalOpened = 0;

// Call onCloseStart callback

if (typeof this.options.onCloseStart === 'function') {

this.options.onCloseStart.call(this, this.el);

}

this.el.classList.remove('open');

// Enable body scrolling only if there are no more modals open.

if (Modal.\_modalsOpen === 0) {

document.body.style.overflow = '';

}

if (this.options.dismissible) {

document.removeEventListener('keydown', this.\_handleKeydownBound);

document.removeEventListener('focus', this.\_handleFocusBound, true);

}

anim.remove(this.el);

anim.remove(this.$overlay[0]);

this.\_animateOut();

return this;

}

}], [{

key: "init",

value: function init(els, options) {

return \_get(Modal.\_\_proto\_\_ || Object.getPrototypeOf(Modal), "init", this).call(this, this, els, options);

}

/\*\*

\* Get Instance

\*/

}, {

key: "getInstance",

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_Modal;

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return Modal;

}(Component);

/\*\*

\* @static

\* @memberof Modal

\*/

Modal.\_modalsOpen = 0;

/\*\*

\* @static

\* @memberof Modal

\*/

Modal.\_count = 0;

M.Modal = Modal;

if (M.jQueryLoaded) {

M.initializeJqueryWrapper(Modal, 'modal', 'M\_Modal');

}

})(cash, M.anime);

;(function ($, anim) {

'use strict';

var \_defaults = {

inDuration: 275,

outDuration: 200,

onOpenStart: null,

onOpenEnd: null,

onCloseStart: null,

onCloseEnd: null

};

/\*\*

\* @class

\*

\*/

var Materialbox = function (\_Component4) {

\_inherits(Materialbox, \_Component4);

/\*\*

\* Construct Materialbox instance

\* @constructor

\* @param {Element} el

\* @param {Object} options

\*/

function Materialbox(el, options) {

\_classCallCheck(this, Materialbox);

var \_this16 = \_possibleConstructorReturn(this, (Materialbox.\_\_proto\_\_ || Object.getPrototypeOf(Materialbox)).call(this, Materialbox, el, options));

\_this16.el.M\_Materialbox = \_this16;

/\*\*

\* Options for the modal

\* @member Materialbox#options

\* @prop {Number} [inDuration=275] - Length in ms of enter transition

\* @prop {Number} [outDuration=200] - Length in ms of exit transition

\* @prop {Function} onOpenStart - Callback function called before materialbox is opened

\* @prop {Function} onOpenEnd - Callback function called after materialbox is opened

\* @prop {Function} onCloseStart - Callback function called before materialbox is closed

\* @prop {Function} onCloseEnd - Callback function called after materialbox is closed

\*/

\_this16.options = $.extend({}, Materialbox.defaults, options);

\_this16.overlayActive = false;

\_this16.doneAnimating = true;

\_this16.placeholder = $('<div></div>').addClass('material-placeholder');

\_this16.originalWidth = 0;

\_this16.originalHeight = 0;

\_this16.originInlineStyles = \_this16.$el.attr('style');

\_this16.caption = \_this16.el.getAttribute('data-caption') || '';

// Wrap

\_this16.$el.before(\_this16.placeholder);

\_this16.placeholder.append(\_this16.$el);

\_this16.\_setupEventHandlers();

return \_this16;

}

\_createClass(Materialbox, [{

key: "destroy",

/\*\*

\* Teardown component

\*/

value: function destroy() {

this.\_removeEventHandlers();

this.el.M\_Materialbox = undefined;

// Unwrap image

$(this.placeholder).after(this.el).remove();

this.$el.removeAttr('style');

}

/\*\*

\* Setup Event Handlers

\*/

}, {

key: "\_setupEventHandlers",

value: function \_setupEventHandlers() {

this.\_handleMaterialboxClickBound = this.\_handleMaterialboxClick.bind(this);

this.el.addEventListener('click', this.\_handleMaterialboxClickBound);

}

/\*\*

\* Remove Event Handlers

\*/

}, {

key: "\_removeEventHandlers",

value: function \_removeEventHandlers() {

this.el.removeEventListener('click', this.\_handleMaterialboxClickBound);

}

/\*\*

\* Handle Materialbox Click

\* @param {Event} e

\*/

}, {

key: "\_handleMaterialboxClick",

value: function \_handleMaterialboxClick(e) {

// If already modal, return to original

if (this.doneAnimating === false || this.overlayActive && this.doneAnimating) {

this.close();

} else {

this.open();

}

}

/\*\*

\* Handle Window Scroll

\*/

}, {

key: "\_handleWindowScroll",

value: function \_handleWindowScroll() {

if (this.overlayActive) {

this.close();

}

}

/\*\*

\* Handle Window Resize

\*/

}, {

key: "\_handleWindowResize",

value: function \_handleWindowResize() {

if (this.overlayActive) {

this.close();

}

}

/\*\*

\* Handle Window Resize

\* @param {Event} e

\*/

}, {

key: "\_handleWindowEscape",

value: function \_handleWindowEscape(e) {

// ESC key

if (e.keyCode === 27 && this.doneAnimating && this.overlayActive) {

this.close();

}

}

/\*\*

\* Find ancestors with overflow: hidden; and make visible

\*/

}, {

key: "\_makeAncestorsOverflowVisible",

value: function \_makeAncestorsOverflowVisible() {

this.ancestorsChanged = $();

var ancestor = this.placeholder[0].parentNode;

while (ancestor !== null && !$(ancestor).is(document)) {

var curr = $(ancestor);

if (curr.css('overflow') !== 'visible') {

curr.css('overflow', 'visible');

if (this.ancestorsChanged === undefined) {

this.ancestorsChanged = curr;

} else {

this.ancestorsChanged = this.ancestorsChanged.add(curr);

}

}

ancestor = ancestor.parentNode;

}

}

/\*\*

\* Animate image in

\*/

}, {

key: "\_animateImageIn",

value: function \_animateImageIn() {

var \_this17 = this;

var animOptions = {

targets: this.el,

height: [this.originalHeight, this.newHeight],

width: [this.originalWidth, this.newWidth],

left: M.getDocumentScrollLeft() + this.windowWidth / 2 - this.placeholder.offset().left - this.newWidth / 2,

top: M.getDocumentScrollTop() + this.windowHeight / 2 - this.placeholder.offset().top - this.newHeight / 2,

duration: this.options.inDuration,

easing: 'easeOutQuad',

complete: function () {

\_this17.doneAnimating = true;

// onOpenEnd callback

if (typeof \_this17.options.onOpenEnd === 'function') {

\_this17.options.onOpenEnd.call(\_this17, \_this17.el);

}

}

};

// Override max-width or max-height if needed

this.maxWidth = this.$el.css('max-width');

this.maxHeight = this.$el.css('max-height');

if (this.maxWidth !== 'none') {

animOptions.maxWidth = this.newWidth;

}

if (this.maxHeight !== 'none') {

animOptions.maxHeight = this.newHeight;

}

anim(animOptions);

}

/\*\*

\* Animate image out

\*/

}, {

key: "\_animateImageOut",

value: function \_animateImageOut() {

var \_this18 = this;

var animOptions = {

targets: this.el,

width: this.originalWidth,

height: this.originalHeight,

left: 0,

top: 0,

duration: this.options.outDuration,

easing: 'easeOutQuad',

complete: function () {

\_this18.placeholder.css({

height: '',

width: '',

position: '',

top: '',

left: ''

});

// Revert to width or height attribute

if (\_this18.attrWidth) {

\_this18.$el.attr('width', \_this18.attrWidth);

}

if (\_this18.attrHeight) {

\_this18.$el.attr('height', \_this18.attrHeight);

}

\_this18.$el.removeAttr('style');

\_this18.originInlineStyles && \_this18.$el.attr('style', \_this18.originInlineStyles);

// Remove class

\_this18.$el.removeClass('active');

\_this18.doneAnimating = true;

// Remove overflow overrides on ancestors

if (\_this18.ancestorsChanged.length) {

\_this18.ancestorsChanged.css('overflow', '');

}

// onCloseEnd callback

if (typeof \_this18.options.onCloseEnd === 'function') {

\_this18.options.onCloseEnd.call(\_this18, \_this18.el);

}

}

};

anim(animOptions);

}

/\*\*

\* Update open and close vars

\*/

}, {

key: "\_updateVars",

value: function \_updateVars() {

this.windowWidth = window.innerWidth;

this.windowHeight = window.innerHeight;

this.caption = this.el.getAttribute('data-caption') || '';

}

/\*\*

\* Open Materialbox

\*/

}, {

key: "open",

value: function open() {

var \_this19 = this;

this.\_updateVars();

this.originalWidth = this.el.getBoundingClientRect().width;

this.originalHeight = this.el.getBoundingClientRect().height;

// Set states

this.doneAnimating = false;

this.$el.addClass('active');

this.overlayActive = true;

// onOpenStart callback

if (typeof this.options.onOpenStart === 'function') {

this.options.onOpenStart.call(this, this.el);

}

// Set positioning for placeholder

this.placeholder.css({

width: this.placeholder[0].getBoundingClientRect().width + 'px',

height: this.placeholder[0].getBoundingClientRect().height + 'px',

position: 'relative',

top: 0,

left: 0

});

this.\_makeAncestorsOverflowVisible();

// Set css on origin

this.$el.css({

position: 'absolute',

'z-index': 1000,

'will-change': 'left, top, width, height'

});

// Change from width or height attribute to css

this.attrWidth = this.$el.attr('width');

this.attrHeight = this.$el.attr('height');

if (this.attrWidth) {

this.$el.css('width', this.attrWidth + 'px');

this.$el.removeAttr('width');

}

if (this.attrHeight) {

this.$el.css('width', this.attrHeight + 'px');

this.$el.removeAttr('height');

}

// Add overlay

this.$overlay = $('<div id="materialbox-overlay"></div>').css({

opacity: 0

}).one('click', function () {

if (\_this19.doneAnimating) {

\_this19.close();

}

});

// Put before in origin image to preserve z-index layering.

this.$el.before(this.$overlay);

// Set dimensions if needed

var overlayOffset = this.$overlay[0].getBoundingClientRect();

this.$overlay.css({

width: this.windowWidth + 'px',

height: this.windowHeight + 'px',

left: -1 \* overlayOffset.left + 'px',

top: -1 \* overlayOffset.top + 'px'

});

anim.remove(this.el);

anim.remove(this.$overlay[0]);

// Animate Overlay

anim({

targets: this.$overlay[0],

opacity: 1,

duration: this.options.inDuration,

easing: 'easeOutQuad'

});

// Add and animate caption if it exists

if (this.caption !== '') {

if (this.$photocaption) {

anim.remove(this.$photoCaption[0]);

}

this.$photoCaption = $('<div class="materialbox-caption"></div>');

this.$photoCaption.text(this.caption);

$('body').append(this.$photoCaption);

this.$photoCaption.css({ display: 'inline' });

anim({

targets: this.$photoCaption[0],

opacity: 1,

duration: this.options.inDuration,

easing: 'easeOutQuad'

});

}

// Resize Image

var ratio = 0;

var widthPercent = this.originalWidth / this.windowWidth;

var heightPercent = this.originalHeight / this.windowHeight;

this.newWidth = 0;

this.newHeight = 0;

if (widthPercent > heightPercent) {

ratio = this.originalHeight / this.originalWidth;

this.newWidth = this.windowWidth \* 0.9;

this.newHeight = this.windowWidth \* 0.9 \* ratio;

} else {

ratio = this.originalWidth / this.originalHeight;

this.newWidth = this.windowHeight \* 0.9 \* ratio;

this.newHeight = this.windowHeight \* 0.9;

}

this.\_animateImageIn();

// Handle Exit triggers

this.\_handleWindowScrollBound = this.\_handleWindowScroll.bind(this);

this.\_handleWindowResizeBound = this.\_handleWindowResize.bind(this);

this.\_handleWindowEscapeBound = this.\_handleWindowEscape.bind(this);

window.addEventListener('scroll', this.\_handleWindowScrollBound);

window.addEventListener('resize', this.\_handleWindowResizeBound);

window.addEventListener('keyup', this.\_handleWindowEscapeBound);

}

/\*\*

\* Close Materialbox

\*/

}, {

key: "close",

value: function close() {

var \_this20 = this;

this.\_updateVars();

this.doneAnimating = false;

// onCloseStart callback

if (typeof this.options.onCloseStart === 'function') {

this.options.onCloseStart.call(this, this.el);

}

anim.remove(this.el);

anim.remove(this.$overlay[0]);

if (this.caption !== '') {

anim.remove(this.$photoCaption[0]);

}

// disable exit handlers

window.removeEventListener('scroll', this.\_handleWindowScrollBound);

window.removeEventListener('resize', this.\_handleWindowResizeBound);

window.removeEventListener('keyup', this.\_handleWindowEscapeBound);

anim({

targets: this.$overlay[0],

opacity: 0,

duration: this.options.outDuration,

easing: 'easeOutQuad',

complete: function () {

\_this20.overlayActive = false;

\_this20.$overlay.remove();

}

});

this.\_animateImageOut();

// Remove Caption + reset css settings on image

if (this.caption !== '') {

anim({

targets: this.$photoCaption[0],

opacity: 0,

duration: this.options.outDuration,

easing: 'easeOutQuad',

complete: function () {

\_this20.$photoCaption.remove();

}

});

}

}

}], [{

key: "init",

value: function init(els, options) {

return \_get(Materialbox.\_\_proto\_\_ || Object.getPrototypeOf(Materialbox), "init", this).call(this, this, els, options);

}

/\*\*

\* Get Instance

\*/

}, {

key: "getInstance",

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_Materialbox;

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return Materialbox;

}(Component);

M.Materialbox = Materialbox;

if (M.jQueryLoaded) {

M.initializeJqueryWrapper(Materialbox, 'materialbox', 'M\_Materialbox');

}

})(cash, M.anime);

;(function ($) {

'use strict';

var \_defaults = {

responsiveThreshold: 0 // breakpoint for swipeable

};

var Parallax = function (\_Component5) {

\_inherits(Parallax, \_Component5);

function Parallax(el, options) {

\_classCallCheck(this, Parallax);

var \_this21 = \_possibleConstructorReturn(this, (Parallax.\_\_proto\_\_ || Object.getPrototypeOf(Parallax)).call(this, Parallax, el, options));

\_this21.el.M\_Parallax = \_this21;

/\*\*

\* Options for the Parallax

\* @member Parallax#options

\* @prop {Number} responsiveThreshold

\*/

\_this21.options = $.extend({}, Parallax.defaults, options);

\_this21.\_enabled = window.innerWidth > \_this21.options.responsiveThreshold;

\_this21.$img = \_this21.$el.find('img').first();

\_this21.$img.each(function () {

var el = this;

if (el.complete) $(el).trigger('load');

});

\_this21.\_updateParallax();

\_this21.\_setupEventHandlers();

\_this21.\_setupStyles();

Parallax.\_parallaxes.push(\_this21);

return \_this21;

}

\_createClass(Parallax, [{

key: "destroy",

/\*\*

\* Teardown component

\*/

value: function destroy() {

Parallax.\_parallaxes.splice(Parallax.\_parallaxes.indexOf(this), 1);

this.$img[0].style.transform = '';

this.\_removeEventHandlers();

this.$el[0].M\_Parallax = undefined;

}

}, {

key: "\_setupEventHandlers",

value: function \_setupEventHandlers() {

this.\_handleImageLoadBound = this.\_handleImageLoad.bind(this);

this.$img[0].addEventListener('load', this.\_handleImageLoadBound);

if (Parallax.\_parallaxes.length === 0) {

Parallax.\_handleScrollThrottled = M.throttle(Parallax.\_handleScroll, 5);

window.addEventListener('scroll', Parallax.\_handleScrollThrottled);

Parallax.\_handleWindowResizeThrottled = M.throttle(Parallax.\_handleWindowResize, 5);

window.addEventListener('resize', Parallax.\_handleWindowResizeThrottled);

}

}

}, {

key: "\_removeEventHandlers",

value: function \_removeEventHandlers() {

this.$img[0].removeEventListener('load', this.\_handleImageLoadBound);

if (Parallax.\_parallaxes.length === 0) {

window.removeEventListener('scroll', Parallax.\_handleScrollThrottled);

window.removeEventListener('resize', Parallax.\_handleWindowResizeThrottled);

}

}

}, {

key: "\_setupStyles",

value: function \_setupStyles() {

this.$img[0].style.opacity = 1;

}

}, {

key: "\_handleImageLoad",

value: function \_handleImageLoad() {

this.\_updateParallax();

}

}, {

key: "\_updateParallax",

value: function \_updateParallax() {

var containerHeight = this.$el.height() > 0 ? this.el.parentNode.offsetHeight : 500;

var imgHeight = this.$img[0].offsetHeight;

var parallaxDist = imgHeight - containerHeight;

var bottom = this.$el.offset().top + containerHeight;

var top = this.$el.offset().top;

var scrollTop = M.getDocumentScrollTop();

var windowHeight = window.innerHeight;

var windowBottom = scrollTop + windowHeight;

var percentScrolled = (windowBottom - top) / (containerHeight + windowHeight);

var parallax = parallaxDist \* percentScrolled;

if (!this.\_enabled) {

this.$img[0].style.transform = '';

} else if (bottom > scrollTop && top < scrollTop + windowHeight) {

this.$img[0].style.transform = "translate3D(-50%, " + parallax + "px, 0)";

}

}

}], [{

key: "init",

value: function init(els, options) {

return \_get(Parallax.\_\_proto\_\_ || Object.getPrototypeOf(Parallax), "init", this).call(this, this, els, options);

}

/\*\*

\* Get Instance

\*/

}, {

key: "getInstance",

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_Parallax;

}

}, {

key: "\_handleScroll",

value: function \_handleScroll() {

for (var i = 0; i < Parallax.\_parallaxes.length; i++) {

var parallaxInstance = Parallax.\_parallaxes[i];

parallaxInstance.\_updateParallax.call(parallaxInstance);

}

}

}, {

key: "\_handleWindowResize",

value: function \_handleWindowResize() {

for (var i = 0; i < Parallax.\_parallaxes.length; i++) {

var parallaxInstance = Parallax.\_parallaxes[i];

parallaxInstance.\_enabled = window.innerWidth > parallaxInstance.options.responsiveThreshold;

}

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return Parallax;

}(Component);

/\*\*

\* @static

\* @memberof Parallax

\*/

Parallax.\_parallaxes = [];

M.Parallax = Parallax;

if (M.jQueryLoaded) {

M.initializeJqueryWrapper(Parallax, 'parallax', 'M\_Parallax');

}

})(cash);

;(function ($, anim) {

'use strict';

var \_defaults = {

duration: 300,

onShow: null,

swipeable: false,

responsiveThreshold: Infinity // breakpoint for swipeable

};

/\*\*

\* @class

\*

\*/

var Tabs = function (\_Component6) {

\_inherits(Tabs, \_Component6);

/\*\*

\* Construct Tabs instance

\* @constructor

\* @param {Element} el

\* @param {Object} options

\*/

function Tabs(el, options) {

\_classCallCheck(this, Tabs);

var \_this22 = \_possibleConstructorReturn(this, (Tabs.\_\_proto\_\_ || Object.getPrototypeOf(Tabs)).call(this, Tabs, el, options));

\_this22.el.M\_Tabs = \_this22;

/\*\*

\* Options for the Tabs

\* @member Tabs#options

\* @prop {Number} duration

\* @prop {Function} onShow

\* @prop {Boolean} swipeable

\* @prop {Number} responsiveThreshold

\*/

\_this22.options = $.extend({}, Tabs.defaults, options);

// Setup

\_this22.$tabLinks = \_this22.$el.children('li.tab').children('a');

\_this22.index = 0;

\_this22.\_setupActiveTabLink();

// Setup tabs content

if (\_this22.options.swipeable) {

\_this22.\_setupSwipeableTabs();

} else {

\_this22.\_setupNormalTabs();

}

// Setup tabs indicator after content to ensure accurate widths

\_this22.\_setTabsAndTabWidth();

\_this22.\_createIndicator();

\_this22.\_setupEventHandlers();

return \_this22;

}

\_createClass(Tabs, [{

key: "destroy",

/\*\*

\* Teardown component

\*/

value: function destroy() {

this.\_removeEventHandlers();

this.\_indicator.parentNode.removeChild(this.\_indicator);

if (this.options.swipeable) {

this.\_teardownSwipeableTabs();

} else {

this.\_teardownNormalTabs();

}

this.$el[0].M\_Tabs = undefined;

}

/\*\*

\* Setup Event Handlers

\*/

}, {

key: "\_setupEventHandlers",

value: function \_setupEventHandlers() {

this.\_handleWindowResizeBound = this.\_handleWindowResize.bind(this);

window.addEventListener('resize', this.\_handleWindowResizeBound);

this.\_handleTabClickBound = this.\_handleTabClick.bind(this);

this.el.addEventListener('click', this.\_handleTabClickBound);

}

/\*\*

\* Remove Event Handlers

\*/

}, {

key: "\_removeEventHandlers",

value: function \_removeEventHandlers() {

window.removeEventListener('resize', this.\_handleWindowResizeBound);

this.el.removeEventListener('click', this.\_handleTabClickBound);

}

/\*\*

\* Handle window Resize

\*/

}, {

key: "\_handleWindowResize",

value: function \_handleWindowResize() {

this.\_setTabsAndTabWidth();

if (this.tabWidth !== 0 && this.tabsWidth !== 0) {

this.\_indicator.style.left = this.\_calcLeftPos(this.$activeTabLink) + 'px';

this.\_indicator.style.right = this.\_calcRightPos(this.$activeTabLink) + 'px';

}

}

/\*\*

\* Handle tab click

\* @param {Event} e

\*/

}, {

key: "\_handleTabClick",

value: function \_handleTabClick(e) {

var \_this23 = this;

var tab = $(e.target).closest('li.tab');

var tabLink = $(e.target).closest('a');

// Handle click on tab link only

if (!tabLink.length || !tabLink.parent().hasClass('tab')) {

return;

}

if (tab.hasClass('disabled')) {

e.preventDefault();

return;

}

// Act as regular link if target attribute is specified.

if (!!tabLink.attr('target')) {

return;

}

// Make the old tab inactive.

this.$activeTabLink.removeClass('active');

var $oldContent = this.$content;

// Update the variables with the new link and content

this.$activeTabLink = tabLink;

this.$content = $(M.escapeHash(tabLink[0].hash));

this.$tabLinks = this.$el.children('li.tab').children('a');

// Make the tab active.

this.$activeTabLink.addClass('active');

var prevIndex = this.index;

this.index = Math.max(this.$tabLinks.index(tabLink), 0);

// Swap content

if (this.options.swipeable) {

if (this.\_tabsCarousel) {

this.\_tabsCarousel.set(this.index, function () {

if (typeof \_this23.options.onShow === 'function') {

\_this23.options.onShow.call(\_this23, \_this23.$content[0]);

}

});

}

} else {

if (this.$content.length) {

this.$content[0].style.display = 'block';

this.$content.addClass('active');

if (typeof this.options.onShow === 'function') {

this.options.onShow.call(this, this.$content[0]);

}

if ($oldContent.length && !$oldContent.is(this.$content)) {

$oldContent[0].style.display = 'none';

$oldContent.removeClass('active');

}

}

}

// Update widths after content is swapped (scrollbar bugfix)

this.\_setTabsAndTabWidth();

// Update indicator

this.\_animateIndicator(prevIndex);

// Prevent the anchor's default click action

e.preventDefault();

}

/\*\*

\* Generate elements for tab indicator.

\*/

}, {

key: "\_createIndicator",

value: function \_createIndicator() {

var \_this24 = this;

var indicator = document.createElement('li');

indicator.classList.add('indicator');

this.el.appendChild(indicator);

this.\_indicator = indicator;

setTimeout(function () {

\_this24.\_indicator.style.left = \_this24.\_calcLeftPos(\_this24.$activeTabLink) + 'px';

\_this24.\_indicator.style.right = \_this24.\_calcRightPos(\_this24.$activeTabLink) + 'px';

}, 0);

}

/\*\*

\* Setup first active tab link.

\*/

}, {

key: "\_setupActiveTabLink",

value: function \_setupActiveTabLink() {

// If the location.hash matches one of the links, use that as the active tab.

this.$activeTabLink = $(this.$tabLinks.filter('[href="' + location.hash + '"]'));

// If no match is found, use the first link or any with class 'active' as the initial active tab.

if (this.$activeTabLink.length === 0) {

this.$activeTabLink = this.$el.children('li.tab').children('a.active').first();

}

if (this.$activeTabLink.length === 0) {

this.$activeTabLink = this.$el.children('li.tab').children('a').first();

}

this.$tabLinks.removeClass('active');

this.$activeTabLink[0].classList.add('active');

this.index = Math.max(this.$tabLinks.index(this.$activeTabLink), 0);

if (this.$activeTabLink.length) {

this.$content = $(M.escapeHash(this.$activeTabLink[0].hash));

this.$content.addClass('active');

}

}

/\*\*

\* Setup swipeable tabs

\*/

}, {

key: "\_setupSwipeableTabs",

value: function \_setupSwipeableTabs() {

var \_this25 = this;

// Change swipeable according to responsive threshold

if (window.innerWidth > this.options.responsiveThreshold) {

this.options.swipeable = false;

}

var $tabsContent = $();

this.$tabLinks.each(function (link) {

var $currContent = $(M.escapeHash(link.hash));

$currContent.addClass('carousel-item');

$tabsContent = $tabsContent.add($currContent);

});

var $tabsWrapper = $('<div class="tabs-content carousel carousel-slider"></div>');

$tabsContent.first().before($tabsWrapper);

$tabsWrapper.append($tabsContent);

$tabsContent[0].style.display = '';

// Keep active tab index to set initial carousel slide

var activeTabIndex = this.$activeTabLink.closest('.tab').index();

this.\_tabsCarousel = M.Carousel.init($tabsWrapper[0], {

fullWidth: true,

noWrap: true,

onCycleTo: function (item) {

var prevIndex = \_this25.index;

\_this25.index = $(item).index();

\_this25.$activeTabLink.removeClass('active');

\_this25.$activeTabLink = \_this25.$tabLinks.eq(\_this25.index);

\_this25.$activeTabLink.addClass('active');

\_this25.\_animateIndicator(prevIndex);

if (typeof \_this25.options.onShow === 'function') {

\_this25.options.onShow.call(\_this25, \_this25.$content[0]);

}

}

});

// Set initial carousel slide to active tab

this.\_tabsCarousel.set(activeTabIndex);

}

/\*\*

\* Teardown normal tabs.

\*/

}, {

key: "\_teardownSwipeableTabs",

value: function \_teardownSwipeableTabs() {

var $tabsWrapper = this.\_tabsCarousel.$el;

this.\_tabsCarousel.destroy();

// Unwrap

$tabsWrapper.after($tabsWrapper.children());

$tabsWrapper.remove();

}

/\*\*

\* Setup normal tabs.

\*/

}, {

key: "\_setupNormalTabs",

value: function \_setupNormalTabs() {

// Hide Tabs Content

this.$tabLinks.not(this.$activeTabLink).each(function (link) {

if (!!link.hash) {

var $currContent = $(M.escapeHash(link.hash));

if ($currContent.length) {

$currContent[0].style.display = 'none';

}

}

});

}

/\*\*

\* Teardown normal tabs.

\*/

}, {

key: "\_teardownNormalTabs",

value: function \_teardownNormalTabs() {

// show Tabs Content

this.$tabLinks.each(function (link) {

if (!!link.hash) {

var $currContent = $(M.escapeHash(link.hash));

if ($currContent.length) {

$currContent[0].style.display = '';

}

}

});

}

/\*\*

\* set tabs and tab width

\*/

}, {

key: "\_setTabsAndTabWidth",

value: function \_setTabsAndTabWidth() {

this.tabsWidth = this.$el.width();

this.tabWidth = Math.max(this.tabsWidth, this.el.scrollWidth) / this.$tabLinks.length;

}

/\*\*

\* Finds right attribute for indicator based on active tab.

\* @param {cash} el

\*/

}, {

key: "\_calcRightPos",

value: function \_calcRightPos(el) {

return Math.ceil(this.tabsWidth - el.position().left - el[0].getBoundingClientRect().width);

}

/\*\*

\* Finds left attribute for indicator based on active tab.

\* @param {cash} el

\*/

}, {

key: "\_calcLeftPos",

value: function \_calcLeftPos(el) {

return Math.floor(el.position().left);

}

}, {

key: "updateTabIndicator",

value: function updateTabIndicator() {

this.\_setTabsAndTabWidth();

this.\_animateIndicator(this.index);

}

/\*\*

\* Animates Indicator to active tab.

\* @param {Number} prevIndex

\*/

}, {

key: "\_animateIndicator",

value: function \_animateIndicator(prevIndex) {

var leftDelay = 0,

rightDelay = 0;

if (this.index - prevIndex >= 0) {

leftDelay = 90;

} else {

rightDelay = 90;

}

// Animate

var animOptions = {

targets: this.\_indicator,

left: {

value: this.\_calcLeftPos(this.$activeTabLink),

delay: leftDelay

},

right: {

value: this.\_calcRightPos(this.$activeTabLink),

delay: rightDelay

},

duration: this.options.duration,

easing: 'easeOutQuad'

};

anim.remove(this.\_indicator);

anim(animOptions);

}

/\*\*

\* Select tab.

\* @param {String} tabId

\*/

}, {

key: "select",

value: function select(tabId) {

var tab = this.$tabLinks.filter('[href="#' + tabId + '"]');

if (tab.length) {

tab.trigger('click');

}

}

}], [{

key: "init",

value: function init(els, options) {

return \_get(Tabs.\_\_proto\_\_ || Object.getPrototypeOf(Tabs), "init", this).call(this, this, els, options);

}

/\*\*

\* Get Instance

\*/

}, {

key: "getInstance",

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_Tabs;

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return Tabs;

}(Component);

M.Tabs = Tabs;

if (M.jQueryLoaded) {

M.initializeJqueryWrapper(Tabs, 'tabs', 'M\_Tabs');

}

})(cash, M.anime);

;(function ($, anim) {

'use strict';

var \_defaults = {

exitDelay: 200,

enterDelay: 0,

html: null,

margin: 5,

inDuration: 250,

outDuration: 200,

position: 'bottom',

transitionMovement: 10

};

/\*\*

\* @class

\*

\*/

var Tooltip = function (\_Component7) {

\_inherits(Tooltip, \_Component7);

/\*\*

\* Construct Tooltip instance

\* @constructor

\* @param {Element} el

\* @param {Object} options

\*/

function Tooltip(el, options) {

\_classCallCheck(this, Tooltip);

var \_this26 = \_possibleConstructorReturn(this, (Tooltip.\_\_proto\_\_ || Object.getPrototypeOf(Tooltip)).call(this, Tooltip, el, options));

\_this26.el.M\_Tooltip = \_this26;

\_this26.options = $.extend({}, Tooltip.defaults, options);

\_this26.isOpen = false;

\_this26.isHovered = false;

\_this26.isFocused = false;

\_this26.\_appendTooltipEl();

\_this26.\_setupEventHandlers();

return \_this26;

}

\_createClass(Tooltip, [{

key: "destroy",

/\*\*

\* Teardown component

\*/

value: function destroy() {

$(this.tooltipEl).remove();

this.\_removeEventHandlers();

this.el.M\_Tooltip = undefined;

}

}, {

key: "\_appendTooltipEl",

value: function \_appendTooltipEl() {

var tooltipEl = document.createElement('div');

tooltipEl.classList.add('material-tooltip');

this.tooltipEl = tooltipEl;

var tooltipContentEl = document.createElement('div');

tooltipContentEl.classList.add('tooltip-content');

tooltipContentEl.innerHTML = this.options.html;

tooltipEl.appendChild(tooltipContentEl);

document.body.appendChild(tooltipEl);

}

}, {

key: "\_updateTooltipContent",

value: function \_updateTooltipContent() {

this.tooltipEl.querySelector('.tooltip-content').innerHTML = this.options.html;

}

}, {

key: "\_setupEventHandlers",

value: function \_setupEventHandlers() {

this.\_handleMouseEnterBound = this.\_handleMouseEnter.bind(this);

this.\_handleMouseLeaveBound = this.\_handleMouseLeave.bind(this);

this.\_handleFocusBound = this.\_handleFocus.bind(this);

this.\_handleBlurBound = this.\_handleBlur.bind(this);

this.el.addEventListener('mouseenter', this.\_handleMouseEnterBound);

this.el.addEventListener('mouseleave', this.\_handleMouseLeaveBound);

this.el.addEventListener('focus', this.\_handleFocusBound, true);

this.el.addEventListener('blur', this.\_handleBlurBound, true);

}

}, {

key: "\_removeEventHandlers",

value: function \_removeEventHandlers() {

this.el.removeEventListener('mouseenter', this.\_handleMouseEnterBound);

this.el.removeEventListener('mouseleave', this.\_handleMouseLeaveBound);

this.el.removeEventListener('focus', this.\_handleFocusBound, true);

this.el.removeEventListener('blur', this.\_handleBlurBound, true);

}

}, {

key: "open",

value: function open(isManual) {

if (this.isOpen) {

return;

}

isManual = isManual === undefined ? true : undefined; // Default value true

this.isOpen = true;

// Update tooltip content with HTML attribute options

this.options = $.extend({}, this.options, this.\_getAttributeOptions());

this.\_updateTooltipContent();

this.\_setEnterDelayTimeout(isManual);

}

}, {

key: "close",

value: function close() {

if (!this.isOpen) {

return;

}

this.isHovered = false;

this.isFocused = false;

this.isOpen = false;

this.\_setExitDelayTimeout();

}

/\*\*

\* Create timeout which delays when the tooltip closes

\*/

}, {

key: "\_setExitDelayTimeout",

value: function \_setExitDelayTimeout() {

var \_this27 = this;

clearTimeout(this.\_exitDelayTimeout);

this.\_exitDelayTimeout = setTimeout(function () {

if (\_this27.isHovered || \_this27.isFocused) {

return;

}

\_this27.\_animateOut();

}, this.options.exitDelay);

}

/\*\*

\* Create timeout which delays when the toast closes

\*/

}, {

key: "\_setEnterDelayTimeout",

value: function \_setEnterDelayTimeout(isManual) {

var \_this28 = this;

clearTimeout(this.\_enterDelayTimeout);

this.\_enterDelayTimeout = setTimeout(function () {

if (!\_this28.isHovered && !\_this28.isFocused && !isManual) {

return;

}

\_this28.\_animateIn();

}, this.options.enterDelay);

}

}, {

key: "\_positionTooltip",

value: function \_positionTooltip() {

var origin = this.el,

tooltip = this.tooltipEl,

originHeight = origin.offsetHeight,

originWidth = origin.offsetWidth,

tooltipHeight = tooltip.offsetHeight,

tooltipWidth = tooltip.offsetWidth,

newCoordinates = void 0,

margin = this.options.margin,

targetTop = void 0,

targetLeft = void 0;

this.xMovement = 0, this.yMovement = 0;

targetTop = origin.getBoundingClientRect().top + M.getDocumentScrollTop();

targetLeft = origin.getBoundingClientRect().left + M.getDocumentScrollLeft();

if (this.options.position === 'top') {

targetTop += -tooltipHeight - margin;

targetLeft += originWidth / 2 - tooltipWidth / 2;

this.yMovement = -this.options.transitionMovement;

} else if (this.options.position === 'right') {

targetTop += originHeight / 2 - tooltipHeight / 2;

targetLeft += originWidth + margin;

this.xMovement = this.options.transitionMovement;

} else if (this.options.position === 'left') {

targetTop += originHeight / 2 - tooltipHeight / 2;

targetLeft += -tooltipWidth - margin;

this.xMovement = -this.options.transitionMovement;

} else {

targetTop += originHeight + margin;

targetLeft += originWidth / 2 - tooltipWidth / 2;

this.yMovement = this.options.transitionMovement;

}

newCoordinates = this.\_repositionWithinScreen(targetLeft, targetTop, tooltipWidth, tooltipHeight);

$(tooltip).css({

top: newCoordinates.y + 'px',

left: newCoordinates.x + 'px'

});

}

}, {

key: "\_repositionWithinScreen",

value: function \_repositionWithinScreen(x, y, width, height) {

var scrollLeft = M.getDocumentScrollLeft();

var scrollTop = M.getDocumentScrollTop();

var newX = x - scrollLeft;

var newY = y - scrollTop;

var bounding = {

left: newX,

top: newY,

width: width,

height: height

};

var offset = this.options.margin + this.options.transitionMovement;

var edges = M.checkWithinContainer(document.body, bounding, offset);

if (edges.left) {

newX = offset;

} else if (edges.right) {

newX -= newX + width - window.innerWidth;

}

if (edges.top) {

newY = offset;

} else if (edges.bottom) {

newY -= newY + height - window.innerHeight;

}

return {

x: newX + scrollLeft,

y: newY + scrollTop

};

}

}, {

key: "\_animateIn",

value: function \_animateIn() {

this.\_positionTooltip();

this.tooltipEl.style.visibility = 'visible';

anim.remove(this.tooltipEl);

anim({

targets: this.tooltipEl,

opacity: 1,

translateX: this.xMovement,

translateY: this.yMovement,

duration: this.options.inDuration,

easing: 'easeOutCubic'

});

}

}, {

key: "\_animateOut",

value: function \_animateOut() {

anim.remove(this.tooltipEl);

anim({

targets: this.tooltipEl,

opacity: 0,

translateX: 0,

translateY: 0,

duration: this.options.outDuration,

easing: 'easeOutCubic'

});

}

}, {

key: "\_handleMouseEnter",

value: function \_handleMouseEnter() {

this.isHovered = true;

this.isFocused = false; // Allows close of tooltip when opened by focus.

this.open(false);

}

}, {

key: "\_handleMouseLeave",

value: function \_handleMouseLeave() {

this.isHovered = false;

this.isFocused = false; // Allows close of tooltip when opened by focus.

this.close();

}

}, {

key: "\_handleFocus",

value: function \_handleFocus() {

if (M.tabPressed) {

this.isFocused = true;

this.open(false);

}

}

}, {

key: "\_handleBlur",

value: function \_handleBlur() {

this.isFocused = false;

this.close();

}

}, {

key: "\_getAttributeOptions",

value: function \_getAttributeOptions() {

var attributeOptions = {};

var tooltipTextOption = this.el.getAttribute('data-tooltip');

var positionOption = this.el.getAttribute('data-position');

if (tooltipTextOption) {

attributeOptions.html = tooltipTextOption;

}

if (positionOption) {

attributeOptions.position = positionOption;

}

return attributeOptions;

}

}], [{

key: "init",

value: function init(els, options) {

return \_get(Tooltip.\_\_proto\_\_ || Object.getPrototypeOf(Tooltip), "init", this).call(this, this, els, options);

}

/\*\*

\* Get Instance

\*/

}, {

key: "getInstance",

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_Tooltip;

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return Tooltip;

}(Component);

M.Tooltip = Tooltip;

if (M.jQueryLoaded) {

M.initializeJqueryWrapper(Tooltip, 'tooltip', 'M\_Tooltip');

}

})(cash, M.anime);

;

;(function (window) {

'use strict';

var Waves = Waves || {};

var $$ = document.querySelectorAll.bind(document);

// Find exact position of element

function isWindow(obj) {

return obj !== null && obj === obj.window;

}

function getWindow(elem) {

return isWindow(elem) ? elem : elem.nodeType === 9 && elem.defaultView;

}

function offset(elem) {

var docElem,

win,

box = { top: 0, left: 0 },

doc = elem && elem.ownerDocument;

docElem = doc.documentElement;

if (typeof elem.getBoundingClientRect !== typeof undefined) {

box = elem.getBoundingClientRect();

}

win = getWindow(doc);

return {

top: box.top + win.pageYOffset - docElem.clientTop,

left: box.left + win.pageXOffset - docElem.clientLeft

};

}

function convertStyle(obj) {

var style = '';

for (var a in obj) {

if (obj.hasOwnProperty(a)) {

style += a + ':' + obj[a] + ';';

}

}

return style;

}

var Effect = {

// Effect delay

duration: 750,

show: function (e, element) {

// Disable right click

if (e.button === 2) {

return false;

}

var el = element || this;

// Create ripple

var ripple = document.createElement('div');

ripple.className = 'waves-ripple';

el.appendChild(ripple);

// Get click coordinate and element witdh

var pos = offset(el);

var relativeY = e.pageY - pos.top;

var relativeX = e.pageX - pos.left;

var scale = 'scale(' + el.clientWidth / 100 \* 10 + ')';

// Support for touch devices

if ('touches' in e) {

relativeY = e.touches[0].pageY - pos.top;

relativeX = e.touches[0].pageX - pos.left;

}

// Attach data to element

ripple.setAttribute('data-hold', Date.now());

ripple.setAttribute('data-scale', scale);

ripple.setAttribute('data-x', relativeX);

ripple.setAttribute('data-y', relativeY);

// Set ripple position

var rippleStyle = {

'top': relativeY + 'px',

'left': relativeX + 'px'

};

ripple.className = ripple.className + ' waves-notransition';

ripple.setAttribute('style', convertStyle(rippleStyle));

ripple.className = ripple.className.replace('waves-notransition', '');

// Scale the ripple

rippleStyle['-webkit-transform'] = scale;

rippleStyle['-moz-transform'] = scale;

rippleStyle['-ms-transform'] = scale;

rippleStyle['-o-transform'] = scale;

rippleStyle.transform = scale;

rippleStyle.opacity = '1';

rippleStyle['-webkit-transition-duration'] = Effect.duration + 'ms';

rippleStyle['-moz-transition-duration'] = Effect.duration + 'ms';

rippleStyle['-o-transition-duration'] = Effect.duration + 'ms';

rippleStyle['transition-duration'] = Effect.duration + 'ms';

rippleStyle['-webkit-transition-timing-function'] = 'cubic-bezier(0.250, 0.460, 0.450, 0.940)';

rippleStyle['-moz-transition-timing-function'] = 'cubic-bezier(0.250, 0.460, 0.450, 0.940)';

rippleStyle['-o-transition-timing-function'] = 'cubic-bezier(0.250, 0.460, 0.450, 0.940)';

rippleStyle['transition-timing-function'] = 'cubic-bezier(0.250, 0.460, 0.450, 0.940)';

ripple.setAttribute('style', convertStyle(rippleStyle));

},

hide: function (e) {

TouchHandler.touchup(e);

var el = this;

var width = el.clientWidth \* 1.4;

// Get first ripple

var ripple = null;

var ripples = el.getElementsByClassName('waves-ripple');

if (ripples.length > 0) {

ripple = ripples[ripples.length - 1];

} else {

return false;

}

var relativeX = ripple.getAttribute('data-x');

var relativeY = ripple.getAttribute('data-y');

var scale = ripple.getAttribute('data-scale');

// Get delay beetween mousedown and mouse leave

var diff = Date.now() - Number(ripple.getAttribute('data-hold'));

var delay = 350 - diff;

if (delay < 0) {

delay = 0;

}

// Fade out ripple after delay

setTimeout(function () {

var style = {

'top': relativeY + 'px',

'left': relativeX + 'px',

'opacity': '0',

// Duration

'-webkit-transition-duration': Effect.duration + 'ms',

'-moz-transition-duration': Effect.duration + 'ms',

'-o-transition-duration': Effect.duration + 'ms',

'transition-duration': Effect.duration + 'ms',

'-webkit-transform': scale,

'-moz-transform': scale,

'-ms-transform': scale,

'-o-transform': scale,

'transform': scale

};

ripple.setAttribute('style', convertStyle(style));

setTimeout(function () {

try {

el.removeChild(ripple);

} catch (e) {

return false;

}

}, Effect.duration);

}, delay);

},

// Little hack to make <input> can perform waves effect

wrapInput: function (elements) {

for (var a = 0; a < elements.length; a++) {

var el = elements[a];

if (el.tagName.toLowerCase() === 'input') {

var parent = el.parentNode;

// If input already have parent just pass through

if (parent.tagName.toLowerCase() === 'i' && parent.className.indexOf('waves-effect') !== -1) {

continue;

}

// Put element class and style to the specified parent

var wrapper = document.createElement('i');

wrapper.className = el.className + ' waves-input-wrapper';

var elementStyle = el.getAttribute('style');

if (!elementStyle) {

elementStyle = '';

}

wrapper.setAttribute('style', elementStyle);

el.className = 'waves-button-input';

el.removeAttribute('style');

// Put element as child

parent.replaceChild(wrapper, el);

wrapper.appendChild(el);

}

}

}

};

/\*\*

\* Disable mousedown event for 500ms during and after touch

\*/

var TouchHandler = {

/\* uses an integer rather than bool so there's no issues with

\* needing to clear timeouts if another touch event occurred

\* within the 500ms. Cannot mouseup between touchstart and

\* touchend, nor in the 500ms after touchend. \*/

touches: 0,

allowEvent: function (e) {

var allow = true;

if (e.type === 'touchstart') {

TouchHandler.touches += 1; //push

} else if (e.type === 'touchend' || e.type === 'touchcancel') {

setTimeout(function () {

if (TouchHandler.touches > 0) {

TouchHandler.touches -= 1; //pop after 500ms

}

}, 500);

} else if (e.type === 'mousedown' && TouchHandler.touches > 0) {

allow = false;

}

return allow;

},

touchup: function (e) {

TouchHandler.allowEvent(e);

}

};

/\*\*

\* Delegated click handler for .waves-effect element.

\* returns null when .waves-effect element not in "click tree"

\*/

function getWavesEffectElement(e) {

if (TouchHandler.allowEvent(e) === false) {

return null;

}

var element = null;

var target = e.target || e.srcElement;

while (target.parentNode !== null) {

if (!(target instanceof SVGElement) && target.className.indexOf('waves-effect') !== -1) {

element = target;

break;

}

target = target.parentNode;

}

return element;

}

/\*\*

\* Bubble the click and show effect if .waves-effect elem was found

\*/

function showEffect(e) {

var element = getWavesEffectElement(e);

if (element !== null) {

Effect.show(e, element);

if ('ontouchstart' in window) {

element.addEventListener('touchend', Effect.hide, false);

element.addEventListener('touchcancel', Effect.hide, false);

}

element.addEventListener('mouseup', Effect.hide, false);

element.addEventListener('mouseleave', Effect.hide, false);

element.addEventListener('dragend', Effect.hide, false);

}

}

Waves.displayEffect = function (options) {

options = options || {};

if ('duration' in options) {

Effect.duration = options.duration;

}

//Wrap input inside <i> tag

Effect.wrapInput($$('.waves-effect'));

if ('ontouchstart' in window) {

document.body.addEventListener('touchstart', showEffect, false);

}

document.body.addEventListener('mousedown', showEffect, false);

};

/\*\*

\* Attach Waves to an input element (or any element which doesn't

\* bubble mouseup/mousedown events).

\* Intended to be used with dynamically loaded forms/inputs, or

\* where the user doesn't want a delegated click handler.

\*/

Waves.attach = function (element) {

//FUTURE: automatically add waves classes and allow users

// to specify them with an options param? Eg. light/classic/button

if (element.tagName.toLowerCase() === 'input') {

Effect.wrapInput([element]);

element = element.parentNode;

}

if ('ontouchstart' in window) {

element.addEventListener('touchstart', showEffect, false);

}

element.addEventListener('mousedown', showEffect, false);

};

window.Waves = Waves;

document.addEventListener('DOMContentLoaded', function () {

Waves.displayEffect();

}, false);

})(window);

;(function ($, anim) {

'use strict';

var \_defaults = {

html: '',

displayLength: 4000,

inDuration: 300,

outDuration: 375,

classes: '',

completeCallback: null,

activationPercent: 0.8

};

var Toast = function () {

function Toast(options) {

\_classCallCheck(this, Toast);

/\*\*

\* Options for the toast

\* @member Toast#options

\*/

this.options = $.extend({}, Toast.defaults, options);

this.message = this.options.html;

/\*\*

\* Describes current pan state toast

\* @type {Boolean}

\*/

this.panning = false;

/\*\*

\* Time remaining until toast is removed

\*/

this.timeRemaining = this.options.displayLength;

if (Toast.\_toasts.length === 0) {

Toast.\_createContainer();

}

// Create new toast

Toast.\_toasts.push(this);

var toastElement = this.\_createToast();

toastElement.M\_Toast = this;

this.el = toastElement;

this.$el = $(toastElement);

this.\_animateIn();

this.\_setTimer();

}

\_createClass(Toast, [{

key: "\_createToast",

/\*\*

\* Create toast and append it to toast container

\*/

value: function \_createToast() {

var toast = document.createElement('div');

toast.classList.add('toast');

// Add custom classes onto toast

if (!!this.options.classes.length) {

$(toast).addClass(this.options.classes);

}

// Set content

if (typeof HTMLElement === 'object' ? this.message instanceof HTMLElement : this.message && typeof this.message === 'object' && this.message !== null && this.message.nodeType === 1 && typeof this.message.nodeName === 'string') {

toast.appendChild(this.message);

// Check if it is jQuery object

} else if (!!this.message.jquery) {

$(toast).append(this.message[0]);

// Insert as html;

} else {

toast.innerHTML = this.message;

}

// Append toasft

Toast.\_container.appendChild(toast);

return toast;

}

/\*\*

\* Animate in toast

\*/

}, {

key: "\_animateIn",

value: function \_animateIn() {

// Animate toast in

anim({

targets: this.el,

top: 0,

opacity: 1,

duration: this.options.inDuration,

easing: 'easeOutCubic'

});

}

/\*\*

\* Create setInterval which automatically removes toast when timeRemaining >= 0

\* has been reached

\*/

}, {

key: "\_setTimer",

value: function \_setTimer() {

var \_this29 = this;

if (this.timeRemaining !== Infinity) {

this.counterInterval = setInterval(function () {

// If toast is not being dragged, decrease its time remaining

if (!\_this29.panning) {

\_this29.timeRemaining -= 20;

}

// Animate toast out

if (\_this29.timeRemaining <= 0) {

\_this29.dismiss();

}

}, 20);

}

}

/\*\*

\* Dismiss toast with animation

\*/

}, {

key: "dismiss",

value: function dismiss() {

var \_this30 = this;

window.clearInterval(this.counterInterval);

var activationDistance = this.el.offsetWidth \* this.options.activationPercent;

if (this.wasSwiped) {

this.el.style.transition = 'transform .05s, opacity .05s';

this.el.style.transform = "translateX(" + activationDistance + "px)";

this.el.style.opacity = 0;

}

anim({

targets: this.el,

opacity: 0,

marginTop: -40,

duration: this.options.outDuration,

easing: 'easeOutExpo',

complete: function () {

// Call the optional callback

if (typeof \_this30.options.completeCallback === 'function') {

\_this30.options.completeCallback();

}

// Remove toast from DOM

\_this30.$el.remove();

Toast.\_toasts.splice(Toast.\_toasts.indexOf(\_this30), 1);

if (Toast.\_toasts.length === 0) {

Toast.\_removeContainer();

}

}

});

}

}], [{

key: "getInstance",

/\*\*

\* Get Instance

\*/

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_Toast;

}

/\*\*

\* Append toast container and add event handlers

\*/

}, {

key: "\_createContainer",

value: function \_createContainer() {

var container = document.createElement('div');

container.setAttribute('id', 'toast-container');

// Add event handler

container.addEventListener('touchstart', Toast.\_onDragStart);

container.addEventListener('touchmove', Toast.\_onDragMove);

container.addEventListener('touchend', Toast.\_onDragEnd);

container.addEventListener('mousedown', Toast.\_onDragStart);

document.addEventListener('mousemove', Toast.\_onDragMove);

document.addEventListener('mouseup', Toast.\_onDragEnd);

document.body.appendChild(container);

Toast.\_container = container;

}

/\*\*

\* Remove toast container and event handlers

\*/

}, {

key: "\_removeContainer",

value: function \_removeContainer() {

// Add event handler

document.removeEventListener('mousemove', Toast.\_onDragMove);

document.removeEventListener('mouseup', Toast.\_onDragEnd);

$(Toast.\_container).remove();

Toast.\_container = null;

}

/\*\*

\* Begin drag handler

\* @param {Event} e

\*/

}, {

key: "\_onDragStart",

value: function \_onDragStart(e) {

if (e.target && $(e.target).closest('.toast').length) {

var $toast = $(e.target).closest('.toast');

var toast = $toast[0].M\_Toast;

toast.panning = true;

Toast.\_draggedToast = toast;

toast.el.classList.add('panning');

toast.el.style.transition = '';

toast.startingXPos = Toast.\_xPos(e);

toast.time = Date.now();

toast.xPos = Toast.\_xPos(e);

}

}

/\*\*

\* Drag move handler

\* @param {Event} e

\*/

}, {

key: "\_onDragMove",

value: function \_onDragMove(e) {

if (!!Toast.\_draggedToast) {

e.preventDefault();

var toast = Toast.\_draggedToast;

toast.deltaX = Math.abs(toast.xPos - Toast.\_xPos(e));

toast.xPos = Toast.\_xPos(e);

toast.velocityX = toast.deltaX / (Date.now() - toast.time);

toast.time = Date.now();

var totalDeltaX = toast.xPos - toast.startingXPos;

var activationDistance = toast.el.offsetWidth \* toast.options.activationPercent;

toast.el.style.transform = "translateX(" + totalDeltaX + "px)";

toast.el.style.opacity = 1 - Math.abs(totalDeltaX / activationDistance);

}

}

/\*\*

\* End drag handler

\*/

}, {

key: "\_onDragEnd",

value: function \_onDragEnd() {

if (!!Toast.\_draggedToast) {

var toast = Toast.\_draggedToast;

toast.panning = false;

toast.el.classList.remove('panning');

var totalDeltaX = toast.xPos - toast.startingXPos;

var activationDistance = toast.el.offsetWidth \* toast.options.activationPercent;

var shouldBeDismissed = Math.abs(totalDeltaX) > activationDistance || toast.velocityX > 1;

// Remove toast

if (shouldBeDismissed) {

toast.wasSwiped = true;

toast.dismiss();

// Animate toast back to original position

} else {

toast.el.style.transition = 'transform .2s, opacity .2s';

toast.el.style.transform = '';

toast.el.style.opacity = '';

}

Toast.\_draggedToast = null;

}

}

/\*\*

\* Get x position of mouse or touch event

\* @param {Event} e

\*/

}, {

key: "\_xPos",

value: function \_xPos(e) {

if (e.targetTouches && e.targetTouches.length >= 1) {

return e.targetTouches[0].clientX;

}

// mouse event

return e.clientX;

}

/\*\*

\* Remove all toasts

\*/

}, {

key: "dismissAll",

value: function dismissAll() {

for (var toastIndex in Toast.\_toasts) {

Toast.\_toasts[toastIndex].dismiss();

}

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return Toast;

}();

/\*\*

\* @static

\* @memberof Toast

\* @type {Array.<Toast>}

\*/

Toast.\_toasts = [];

/\*\*

\* @static

\* @memberof Toast

\*/

Toast.\_container = null;

/\*\*

\* @static

\* @memberof Toast

\* @type {Toast}

\*/

Toast.\_draggedToast = null;

M.Toast = Toast;

M.toast = function (options) {

return new Toast(options);

};

})(cash, M.anime);

;(function ($, anim) {

'use strict';

var \_defaults = {

edge: 'left',

draggable: true,

inDuration: 250,

outDuration: 200,

onOpenStart: null,

onOpenEnd: null,

onCloseStart: null,

onCloseEnd: null,

preventScrolling: true

};

/\*\*

\* @class

\*/

var Sidenav = function (\_Component8) {

\_inherits(Sidenav, \_Component8);

/\*\*

\* Construct Sidenav instance and set up overlay

\* @constructor

\* @param {Element} el

\* @param {Object} options

\*/

function Sidenav(el, options) {

\_classCallCheck(this, Sidenav);

var \_this31 = \_possibleConstructorReturn(this, (Sidenav.\_\_proto\_\_ || Object.getPrototypeOf(Sidenav)).call(this, Sidenav, el, options));

\_this31.el.M\_Sidenav = \_this31;

\_this31.id = \_this31.$el.attr('id');

/\*\*

\* Options for the Sidenav

\* @member Sidenav#options

\* @prop {String} [edge='left'] - Side of screen on which Sidenav appears

\* @prop {Boolean} [draggable=true] - Allow swipe gestures to open/close Sidenav

\* @prop {Number} [inDuration=250] - Length in ms of enter transition

\* @prop {Number} [outDuration=200] - Length in ms of exit transition

\* @prop {Function} onOpenStart - Function called when sidenav starts entering

\* @prop {Function} onOpenEnd - Function called when sidenav finishes entering

\* @prop {Function} onCloseStart - Function called when sidenav starts exiting

\* @prop {Function} onCloseEnd - Function called when sidenav finishes exiting

\*/

\_this31.options = $.extend({}, Sidenav.defaults, options);

/\*\*

\* Describes open/close state of Sidenav

\* @type {Boolean}

\*/

\_this31.isOpen = false;

/\*\*

\* Describes if Sidenav is fixed

\* @type {Boolean}

\*/

\_this31.isFixed = \_this31.el.classList.contains('sidenav-fixed');

/\*\*

\* Describes if Sidenav is being draggeed

\* @type {Boolean}

\*/

\_this31.isDragged = false;

// Window size variables for window resize checks

\_this31.lastWindowWidth = window.innerWidth;

\_this31.lastWindowHeight = window.innerHeight;

\_this31.\_createOverlay();

\_this31.\_createDragTarget();

\_this31.\_setupEventHandlers();

\_this31.\_setupClasses();

\_this31.\_setupFixed();

Sidenav.\_sidenavs.push(\_this31);

return \_this31;

}

\_createClass(Sidenav, [{

key: "destroy",

/\*\*

\* Teardown component

\*/

value: function destroy() {

this.\_removeEventHandlers();

this.\_enableBodyScrolling();

this.\_overlay.parentNode.removeChild(this.\_overlay);

this.dragTarget.parentNode.removeChild(this.dragTarget);

this.el.M\_Sidenav = undefined;

this.el.style.transform = '';

var index = Sidenav.\_sidenavs.indexOf(this);

if (index >= 0) {

Sidenav.\_sidenavs.splice(index, 1);

}

}

}, {

key: "\_createOverlay",

value: function \_createOverlay() {

var overlay = document.createElement('div');

this.\_closeBound = this.close.bind(this);

overlay.classList.add('sidenav-overlay');

overlay.addEventListener('click', this.\_closeBound);

document.body.appendChild(overlay);

this.\_overlay = overlay;

}

}, {

key: "\_setupEventHandlers",

value: function \_setupEventHandlers() {

if (Sidenav.\_sidenavs.length === 0) {

document.body.addEventListener('click', this.\_handleTriggerClick);

}

this.\_handleDragTargetDragBound = this.\_handleDragTargetDrag.bind(this);

this.\_handleDragTargetReleaseBound = this.\_handleDragTargetRelease.bind(this);

this.\_handleCloseDragBound = this.\_handleCloseDrag.bind(this);

this.\_handleCloseReleaseBound = this.\_handleCloseRelease.bind(this);

this.\_handleCloseTriggerClickBound = this.\_handleCloseTriggerClick.bind(this);

this.dragTarget.addEventListener('touchmove', this.\_handleDragTargetDragBound);

this.dragTarget.addEventListener('touchend', this.\_handleDragTargetReleaseBound);

this.\_overlay.addEventListener('touchmove', this.\_handleCloseDragBound);

this.\_overlay.addEventListener('touchend', this.\_handleCloseReleaseBound);

this.el.addEventListener('touchmove', this.\_handleCloseDragBound);

this.el.addEventListener('touchend', this.\_handleCloseReleaseBound);

this.el.addEventListener('click', this.\_handleCloseTriggerClickBound);

// Add resize for side nav fixed

if (this.isFixed) {

this.\_handleWindowResizeBound = this.\_handleWindowResize.bind(this);

window.addEventListener('resize', this.\_handleWindowResizeBound);

}

}

}, {

key: "\_removeEventHandlers",

value: function \_removeEventHandlers() {

if (Sidenav.\_sidenavs.length === 1) {

document.body.removeEventListener('click', this.\_handleTriggerClick);

}

this.dragTarget.removeEventListener('touchmove', this.\_handleDragTargetDragBound);

this.dragTarget.removeEventListener('touchend', this.\_handleDragTargetReleaseBound);

this.\_overlay.removeEventListener('touchmove', this.\_handleCloseDragBound);

this.\_overlay.removeEventListener('touchend', this.\_handleCloseReleaseBound);

this.el.removeEventListener('touchmove', this.\_handleCloseDragBound);

this.el.removeEventListener('touchend', this.\_handleCloseReleaseBound);

this.el.removeEventListener('click', this.\_handleCloseTriggerClickBound);

// Remove resize for side nav fixed

if (this.isFixed) {

window.removeEventListener('resize', this.\_handleWindowResizeBound);

}

}

/\*\*

\* Handle Trigger Click

\* @param {Event} e

\*/

}, {

key: "\_handleTriggerClick",

value: function \_handleTriggerClick(e) {

var $trigger = $(e.target).closest('.sidenav-trigger');

if (e.target && $trigger.length) {

var sidenavId = M.getIdFromTrigger($trigger[0]);

var sidenavInstance = document.getElementById(sidenavId).M\_Sidenav;

if (sidenavInstance) {

sidenavInstance.open($trigger);

}

e.preventDefault();

}

}

/\*\*

\* Set variables needed at the beggining of drag

\* and stop any current transition.

\* @param {Event} e

\*/

}, {

key: "\_startDrag",

value: function \_startDrag(e) {

var clientX = e.targetTouches[0].clientX;

this.isDragged = true;

this.\_startingXpos = clientX;

this.\_xPos = this.\_startingXpos;

this.\_time = Date.now();

this.\_width = this.el.getBoundingClientRect().width;

this.\_overlay.style.display = 'block';

this.\_initialScrollTop = this.isOpen ? this.el.scrollTop : M.getDocumentScrollTop();

this.\_verticallyScrolling = false;

anim.remove(this.el);

anim.remove(this.\_overlay);

}

/\*\*

\* Set variables needed at each drag move update tick

\* @param {Event} e

\*/

}, {

key: "\_dragMoveUpdate",

value: function \_dragMoveUpdate(e) {

var clientX = e.targetTouches[0].clientX;

var currentScrollTop = this.isOpen ? this.el.scrollTop : M.getDocumentScrollTop();

this.deltaX = Math.abs(this.\_xPos - clientX);

this.\_xPos = clientX;

this.velocityX = this.deltaX / (Date.now() - this.\_time);

this.\_time = Date.now();

if (this.\_initialScrollTop !== currentScrollTop) {

this.\_verticallyScrolling = true;

}

}

/\*\*

\* Handles Dragging of Sidenav

\* @param {Event} e

\*/

}, {

key: "\_handleDragTargetDrag",

value: function \_handleDragTargetDrag(e) {

// Check if draggable

if (!this.options.draggable || this.\_isCurrentlyFixed() || this.\_verticallyScrolling) {

return;

}

// If not being dragged, set initial drag start variables

if (!this.isDragged) {

this.\_startDrag(e);

}

// Run touchmove updates

this.\_dragMoveUpdate(e);

// Calculate raw deltaX

var totalDeltaX = this.\_xPos - this.\_startingXpos;

// dragDirection is the attempted user drag direction

var dragDirection = totalDeltaX > 0 ? 'right' : 'left';

// Don't allow totalDeltaX to exceed Sidenav width or be dragged in the opposite direction

totalDeltaX = Math.min(this.\_width, Math.abs(totalDeltaX));

if (this.options.edge === dragDirection) {

totalDeltaX = 0;

}

/\*\*

\* transformX is the drag displacement

\* transformPrefix is the initial transform placement

\* Invert values if Sidenav is right edge

\*/

var transformX = totalDeltaX;

var transformPrefix = 'translateX(-100%)';

if (this.options.edge === 'right') {

transformPrefix = 'translateX(100%)';

transformX = -transformX;

}

// Calculate open/close percentage of sidenav, with open = 1 and close = 0

this.percentOpen = Math.min(1, totalDeltaX / this.\_width);

// Set transform and opacity styles

this.el.style.transform = transformPrefix + " translateX(" + transformX + "px)";

this.\_overlay.style.opacity = this.percentOpen;

}

/\*\*

\* Handle Drag Target Release

\*/

}, {

key: "\_handleDragTargetRelease",

value: function \_handleDragTargetRelease() {

if (this.isDragged) {

if (this.percentOpen > 0.2) {

this.open();

} else {

this.\_animateOut();

}

this.isDragged = false;

this.\_verticallyScrolling = false;

}

}

/\*\*

\* Handle Close Drag

\* @param {Event} e

\*/

}, {

key: "\_handleCloseDrag",

value: function \_handleCloseDrag(e) {

if (this.isOpen) {

// Check if draggable

if (!this.options.draggable || this.\_isCurrentlyFixed() || this.\_verticallyScrolling) {

return;

}

// If not being dragged, set initial drag start variables

if (!this.isDragged) {

this.\_startDrag(e);

}

// Run touchmove updates

this.\_dragMoveUpdate(e);

// Calculate raw deltaX

var totalDeltaX = this.\_xPos - this.\_startingXpos;

// dragDirection is the attempted user drag direction

var dragDirection = totalDeltaX > 0 ? 'right' : 'left';

// Don't allow totalDeltaX to exceed Sidenav width or be dragged in the opposite direction

totalDeltaX = Math.min(this.\_width, Math.abs(totalDeltaX));

if (this.options.edge !== dragDirection) {

totalDeltaX = 0;

}

var transformX = -totalDeltaX;

if (this.options.edge === 'right') {

transformX = -transformX;

}

// Calculate open/close percentage of sidenav, with open = 1 and close = 0

this.percentOpen = Math.min(1, 1 - totalDeltaX / this.\_width);

// Set transform and opacity styles

this.el.style.transform = "translateX(" + transformX + "px)";

this.\_overlay.style.opacity = this.percentOpen;

}

}

/\*\*

\* Handle Close Release

\*/

}, {

key: "\_handleCloseRelease",

value: function \_handleCloseRelease() {

if (this.isOpen && this.isDragged) {

if (this.percentOpen > 0.8) {

this.\_animateIn();

} else {

this.close();

}

this.isDragged = false;

this.\_verticallyScrolling = false;

}

}

/\*\*

\* Handles closing of Sidenav when element with class .sidenav-close

\*/

}, {

key: "\_handleCloseTriggerClick",

value: function \_handleCloseTriggerClick(e) {

var $closeTrigger = $(e.target).closest('.sidenav-close');

if ($closeTrigger.length && !this.\_isCurrentlyFixed()) {

this.close();

}

}

/\*\*

\* Handle Window Resize

\*/

}, {

key: "\_handleWindowResize",

value: function \_handleWindowResize() {

// Only handle horizontal resizes

if (this.lastWindowWidth !== window.innerWidth) {

if (window.innerWidth > 992) {

this.open();

} else {

this.close();

}

}

this.lastWindowWidth = window.innerWidth;

this.lastWindowHeight = window.innerHeight;

}

}, {

key: "\_setupClasses",

value: function \_setupClasses() {

if (this.options.edge === 'right') {

this.el.classList.add('right-aligned');

this.dragTarget.classList.add('right-aligned');

}

}

}, {

key: "\_removeClasses",

value: function \_removeClasses() {

this.el.classList.remove('right-aligned');

this.dragTarget.classList.remove('right-aligned');

}

}, {

key: "\_setupFixed",

value: function \_setupFixed() {

if (this.\_isCurrentlyFixed()) {

this.open();

}

}

}, {

key: "\_isCurrentlyFixed",

value: function \_isCurrentlyFixed() {

return this.isFixed && window.innerWidth > 992;

}

}, {

key: "\_createDragTarget",

value: function \_createDragTarget() {

var dragTarget = document.createElement('div');

dragTarget.classList.add('drag-target');

document.body.appendChild(dragTarget);

this.dragTarget = dragTarget;

}

}, {

key: "\_preventBodyScrolling",

value: function \_preventBodyScrolling() {

var body = document.body;

body.style.overflow = 'hidden';

}

}, {

key: "\_enableBodyScrolling",

value: function \_enableBodyScrolling() {

var body = document.body;

body.style.overflow = '';

}

}, {

key: "open",

value: function open() {

if (this.isOpen === true) {

return;

}

this.isOpen = true;

// Run onOpenStart callback

if (typeof this.options.onOpenStart === 'function') {

this.options.onOpenStart.call(this, this.el);

}

// Handle fixed Sidenav

if (this.\_isCurrentlyFixed()) {

anim.remove(this.el);

anim({

targets: this.el,

translateX: 0,

duration: 0,

easing: 'easeOutQuad'

});

this.\_enableBodyScrolling();

this.\_overlay.style.display = 'none';

// Handle non-fixed Sidenav

} else {

if (this.options.preventScrolling) {

this.\_preventBodyScrolling();

}

if (!this.isDragged || this.percentOpen != 1) {

this.\_animateIn();

}

}

}

}, {

key: "close",

value: function close() {

if (this.isOpen === false) {

return;

}

this.isOpen = false;

// Run onCloseStart callback

if (typeof this.options.onCloseStart === 'function') {

this.options.onCloseStart.call(this, this.el);

}

// Handle fixed Sidenav

if (this.\_isCurrentlyFixed()) {

var transformX = this.options.edge === 'left' ? '-105%' : '105%';

this.el.style.transform = "translateX(" + transformX + ")";

// Handle non-fixed Sidenav

} else {

this.\_enableBodyScrolling();

if (!this.isDragged || this.percentOpen != 0) {

this.\_animateOut();

} else {

this.\_overlay.style.display = 'none';

}

}

}

}, {

key: "\_animateIn",

value: function \_animateIn() {

this.\_animateSidenavIn();

this.\_animateOverlayIn();

}

}, {

key: "\_animateSidenavIn",

value: function \_animateSidenavIn() {

var \_this32 = this;

var slideOutPercent = this.options.edge === 'left' ? -1 : 1;

if (this.isDragged) {

slideOutPercent = this.options.edge === 'left' ? slideOutPercent + this.percentOpen : slideOutPercent - this.percentOpen;

}

anim.remove(this.el);

anim({

targets: this.el,

translateX: [slideOutPercent \* 100 + "%", 0],

duration: this.options.inDuration,

easing: 'easeOutQuad',

complete: function () {

// Run onOpenEnd callback

if (typeof \_this32.options.onOpenEnd === 'function') {

\_this32.options.onOpenEnd.call(\_this32, \_this32.el);

}

}

});

}

}, {

key: "\_animateOverlayIn",

value: function \_animateOverlayIn() {

var start = 0;

if (this.isDragged) {

start = this.percentOpen;

} else {

$(this.\_overlay).css({

display: 'block'

});

}

anim.remove(this.\_overlay);

anim({

targets: this.\_overlay,

opacity: [start, 1],

duration: this.options.inDuration,

easing: 'easeOutQuad'

});

}

}, {

key: "\_animateOut",

value: function \_animateOut() {

this.\_animateSidenavOut();

this.\_animateOverlayOut();

}

}, {

key: "\_animateSidenavOut",

value: function \_animateSidenavOut() {

var \_this33 = this;

var endPercent = this.options.edge === 'left' ? -1 : 1;

var slideOutPercent = 0;

if (this.isDragged) {

slideOutPercent = this.options.edge === 'left' ? endPercent + this.percentOpen : endPercent - this.percentOpen;

}

anim.remove(this.el);

anim({

targets: this.el,

translateX: [slideOutPercent \* 100 + "%", endPercent \* 105 + "%"],

duration: this.options.outDuration,

easing: 'easeOutQuad',

complete: function () {

// Run onOpenEnd callback

if (typeof \_this33.options.onCloseEnd === 'function') {

\_this33.options.onCloseEnd.call(\_this33, \_this33.el);

}

}

});

}

}, {

key: "\_animateOverlayOut",

value: function \_animateOverlayOut() {

var \_this34 = this;

anim.remove(this.\_overlay);

anim({

targets: this.\_overlay,

opacity: 0,

duration: this.options.outDuration,

easing: 'easeOutQuad',

complete: function () {

$(\_this34.\_overlay).css('display', 'none');

}

});

}

}], [{

key: "init",

value: function init(els, options) {

return \_get(Sidenav.\_\_proto\_\_ || Object.getPrototypeOf(Sidenav), "init", this).call(this, this, els, options);

}

/\*\*

\* Get Instance

\*/

}, {

key: "getInstance",

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_Sidenav;

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return Sidenav;

}(Component);

/\*\*

\* @static

\* @memberof Sidenav

\* @type {Array.<Sidenav>}

\*/

Sidenav.\_sidenavs = [];

M.Sidenav = Sidenav;

if (M.jQueryLoaded) {

M.initializeJqueryWrapper(Sidenav, 'sidenav', 'M\_Sidenav');

}

})(cash, M.anime);

;(function ($, anim) {

'use strict';

var \_defaults = {

throttle: 100,

scrollOffset: 200, // offset - 200 allows elements near bottom of page to scroll

activeClass: 'active',

getActiveElement: function (id) {

return 'a[href="#' + id + '"]';

}

};

/\*\*

\* @class

\*

\*/

var ScrollSpy = function (\_Component9) {

\_inherits(ScrollSpy, \_Component9);

/\*\*

\* Construct ScrollSpy instance

\* @constructor

\* @param {Element} el

\* @param {Object} options

\*/

function ScrollSpy(el, options) {

\_classCallCheck(this, ScrollSpy);

var \_this35 = \_possibleConstructorReturn(this, (ScrollSpy.\_\_proto\_\_ || Object.getPrototypeOf(ScrollSpy)).call(this, ScrollSpy, el, options));

\_this35.el.M\_ScrollSpy = \_this35;

/\*\*

\* Options for the modal

\* @member Modal#options

\* @prop {Number} [throttle=100] - Throttle of scroll handler

\* @prop {Number} [scrollOffset=200] - Offset for centering element when scrolled to

\* @prop {String} [activeClass='active'] - Class applied to active elements

\* @prop {Function} [getActiveElement] - Used to find active element

\*/

\_this35.options = $.extend({}, ScrollSpy.defaults, options);

// setup

ScrollSpy.\_elements.push(\_this35);

ScrollSpy.\_count++;

ScrollSpy.\_increment++;

\_this35.tickId = -1;

\_this35.id = ScrollSpy.\_increment;

\_this35.\_setupEventHandlers();

\_this35.\_handleWindowScroll();

return \_this35;

}

\_createClass(ScrollSpy, [{

key: "destroy",

/\*\*

\* Teardown component

\*/

value: function destroy() {

ScrollSpy.\_elements.splice(ScrollSpy.\_elements.indexOf(this), 1);

ScrollSpy.\_elementsInView.splice(ScrollSpy.\_elementsInView.indexOf(this), 1);

ScrollSpy.\_visibleElements.splice(ScrollSpy.\_visibleElements.indexOf(this.$el), 1);

ScrollSpy.\_count--;

this.\_removeEventHandlers();

$(this.options.getActiveElement(this.$el.attr('id'))).removeClass(this.options.activeClass);

this.el.M\_ScrollSpy = undefined;

}

/\*\*

\* Setup Event Handlers

\*/

}, {

key: "\_setupEventHandlers",

value: function \_setupEventHandlers() {

var throttledResize = M.throttle(this.\_handleWindowScroll, 200);

this.\_handleThrottledResizeBound = throttledResize.bind(this);

this.\_handleWindowScrollBound = this.\_handleWindowScroll.bind(this);

if (ScrollSpy.\_count === 1) {

window.addEventListener('scroll', this.\_handleWindowScrollBound);

window.addEventListener('resize', this.\_handleThrottledResizeBound);

document.body.addEventListener('click', this.\_handleTriggerClick);

}

}

/\*\*

\* Remove Event Handlers

\*/

}, {

key: "\_removeEventHandlers",

value: function \_removeEventHandlers() {

if (ScrollSpy.\_count === 0) {

window.removeEventListener('scroll', this.\_handleWindowScrollBound);

window.removeEventListener('resize', this.\_handleThrottledResizeBound);

document.body.removeEventListener('click', this.\_handleTriggerClick);

}

}

/\*\*

\* Handle Trigger Click

\* @param {Event} e

\*/

}, {

key: "\_handleTriggerClick",

value: function \_handleTriggerClick(e) {

var $trigger = $(e.target);

for (var i = ScrollSpy.\_elements.length - 1; i >= 0; i--) {

var scrollspy = ScrollSpy.\_elements[i];

if ($trigger.is('a[href="#' + scrollspy.$el.attr('id') + '"]')) {

e.preventDefault();

var offset = scrollspy.$el.offset().top + 1;

anim({

targets: [document.documentElement, document.body],

scrollTop: offset - scrollspy.options.scrollOffset,

duration: 400,

easing: 'easeOutCubic'

});

break;

}

}

}

/\*\*

\* Handle Window Scroll

\*/

}, {

key: "\_handleWindowScroll",

value: function \_handleWindowScroll() {

// unique tick id

ScrollSpy.\_ticks++;

// viewport rectangle

var top = M.getDocumentScrollTop(),

left = M.getDocumentScrollLeft(),

right = left + window.innerWidth,

bottom = top + window.innerHeight;

// determine which elements are in view

var intersections = ScrollSpy.\_findElements(top, right, bottom, left);

for (var i = 0; i < intersections.length; i++) {

var scrollspy = intersections[i];

var lastTick = scrollspy.tickId;

if (lastTick < 0) {

// entered into view

scrollspy.\_enter();

}

// update tick id

scrollspy.tickId = ScrollSpy.\_ticks;

}

for (var \_i = 0; \_i < ScrollSpy.\_elementsInView.length; \_i++) {

var \_scrollspy = ScrollSpy.\_elementsInView[\_i];

var \_lastTick = \_scrollspy.tickId;

if (\_lastTick >= 0 && \_lastTick !== ScrollSpy.\_ticks) {

// exited from view

\_scrollspy.\_exit();

\_scrollspy.tickId = -1;

}

}

// remember elements in view for next tick

ScrollSpy.\_elementsInView = intersections;

}

/\*\*

\* Find elements that are within the boundary

\* @param {number} top

\* @param {number} right

\* @param {number} bottom

\* @param {number} left

\* @return {Array.<ScrollSpy>} A collection of elements

\*/

}, {

key: "\_enter",

value: function \_enter() {

ScrollSpy.\_visibleElements = ScrollSpy.\_visibleElements.filter(function (value) {

return value.height() != 0;

});

if (ScrollSpy.\_visibleElements[0]) {

$(this.options.getActiveElement(ScrollSpy.\_visibleElements[0].attr('id'))).removeClass(this.options.activeClass);

if (ScrollSpy.\_visibleElements[0][0].M\_ScrollSpy && this.id < ScrollSpy.\_visibleElements[0][0].M\_ScrollSpy.id) {

ScrollSpy.\_visibleElements.unshift(this.$el);

} else {

ScrollSpy.\_visibleElements.push(this.$el);

}

} else {

ScrollSpy.\_visibleElements.push(this.$el);

}

$(this.options.getActiveElement(ScrollSpy.\_visibleElements[0].attr('id'))).addClass(this.options.activeClass);

}

}, {

key: "\_exit",

value: function \_exit() {

var \_this36 = this;

ScrollSpy.\_visibleElements = ScrollSpy.\_visibleElements.filter(function (value) {

return value.height() != 0;

});

if (ScrollSpy.\_visibleElements[0]) {

$(this.options.getActiveElement(ScrollSpy.\_visibleElements[0].attr('id'))).removeClass(this.options.activeClass);

ScrollSpy.\_visibleElements = ScrollSpy.\_visibleElements.filter(function (el) {

return el.attr('id') != \_this36.$el.attr('id');

});

if (ScrollSpy.\_visibleElements[0]) {

// Check if empty

$(this.options.getActiveElement(ScrollSpy.\_visibleElements[0].attr('id'))).addClass(this.options.activeClass);

}

}

}

}], [{

key: "init",

value: function init(els, options) {

return \_get(ScrollSpy.\_\_proto\_\_ || Object.getPrototypeOf(ScrollSpy), "init", this).call(this, this, els, options);

}

/\*\*

\* Get Instance

\*/

}, {

key: "getInstance",

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_ScrollSpy;

}

}, {

key: "\_findElements",

value: function \_findElements(top, right, bottom, left) {

var hits = [];

for (var i = 0; i < ScrollSpy.\_elements.length; i++) {

var scrollspy = ScrollSpy.\_elements[i];

var currTop = top + scrollspy.options.scrollOffset || 200;

if (scrollspy.$el.height() > 0) {

var elTop = scrollspy.$el.offset().top,

elLeft = scrollspy.$el.offset().left,

elRight = elLeft + scrollspy.$el.width(),

elBottom = elTop + scrollspy.$el.height();

var isIntersect = !(elLeft > right || elRight < left || elTop > bottom || elBottom < currTop);

if (isIntersect) {

hits.push(scrollspy);

}

}

}

return hits;

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return ScrollSpy;

}(Component);

/\*\*

\* @static

\* @memberof ScrollSpy

\* @type {Array.<ScrollSpy>}

\*/

ScrollSpy.\_elements = [];

/\*\*

\* @static

\* @memberof ScrollSpy

\* @type {Array.<ScrollSpy>}

\*/

ScrollSpy.\_elementsInView = [];

/\*\*

\* @static

\* @memberof ScrollSpy

\* @type {Array.<cash>}

\*/

ScrollSpy.\_visibleElements = [];

/\*\*

\* @static

\* @memberof ScrollSpy

\*/

ScrollSpy.\_count = 0;

/\*\*

\* @static

\* @memberof ScrollSpy

\*/

ScrollSpy.\_increment = 0;

/\*\*

\* @static

\* @memberof ScrollSpy

\*/

ScrollSpy.\_ticks = 0;

M.ScrollSpy = ScrollSpy;

if (M.jQueryLoaded) {

M.initializeJqueryWrapper(ScrollSpy, 'scrollSpy', 'M\_ScrollSpy');

}

})(cash, M.anime);

;(function ($) {

'use strict';

var \_defaults = {

data: {}, // Autocomplete data set

limit: Infinity, // Limit of results the autocomplete shows

onAutocomplete: null, // Callback for when autocompleted

minLength: 1, // Min characters before autocomplete starts

sortFunction: function (a, b, inputString) {

// Sort function for sorting autocomplete results

return a.indexOf(inputString) - b.indexOf(inputString);

}

};

/\*\*

\* @class

\*

\*/

var Autocomplete = function (\_Component10) {

\_inherits(Autocomplete, \_Component10);

/\*\*

\* Construct Autocomplete instance

\* @constructor

\* @param {Element} el

\* @param {Object} options

\*/

function Autocomplete(el, options) {

\_classCallCheck(this, Autocomplete);

var \_this37 = \_possibleConstructorReturn(this, (Autocomplete.\_\_proto\_\_ || Object.getPrototypeOf(Autocomplete)).call(this, Autocomplete, el, options));

\_this37.el.M\_Autocomplete = \_this37;

/\*\*

\* Options for the autocomplete

\* @member Autocomplete#options

\* @prop {Number} duration

\* @prop {Number} dist

\* @prop {number} shift

\* @prop {number} padding

\* @prop {Boolean} fullWidth

\* @prop {Boolean} indicators

\* @prop {Boolean} noWrap

\* @prop {Function} onCycleTo

\*/

\_this37.options = $.extend({}, Autocomplete.defaults, options);

// Setup

\_this37.isOpen = false;

\_this37.count = 0;

\_this37.activeIndex = -1;

\_this37.oldVal;

\_this37.$inputField = \_this37.$el.closest('.input-field');

\_this37.$active = $();

\_this37.\_mousedown = false;

\_this37.\_setupDropdown();

\_this37.\_setupEventHandlers();

return \_this37;

}

\_createClass(Autocomplete, [{

key: "destroy",

/\*\*

\* Teardown component

\*/

value: function destroy() {

this.\_removeEventHandlers();

this.\_removeDropdown();

this.el.M\_Autocomplete = undefined;

}

/\*\*

\* Setup Event Handlers

\*/

}, {

key: "\_setupEventHandlers",

value: function \_setupEventHandlers() {

this.\_handleInputBlurBound = this.\_handleInputBlur.bind(this);

this.\_handleInputKeyupAndFocusBound = this.\_handleInputKeyupAndFocus.bind(this);

this.\_handleInputKeydownBound = this.\_handleInputKeydown.bind(this);

this.\_handleInputClickBound = this.\_handleInputClick.bind(this);

this.\_handleContainerMousedownAndTouchstartBound = this.\_handleContainerMousedownAndTouchstart.bind(this);

this.\_handleContainerMouseupAndTouchendBound = this.\_handleContainerMouseupAndTouchend.bind(this);

this.el.addEventListener('blur', this.\_handleInputBlurBound);

this.el.addEventListener('keyup', this.\_handleInputKeyupAndFocusBound);

this.el.addEventListener('focus', this.\_handleInputKeyupAndFocusBound);

this.el.addEventListener('keydown', this.\_handleInputKeydownBound);

this.el.addEventListener('click', this.\_handleInputClickBound);

this.container.addEventListener('mousedown', this.\_handleContainerMousedownAndTouchstartBound);

this.container.addEventListener('mouseup', this.\_handleContainerMouseupAndTouchendBound);

if (typeof window.ontouchstart !== 'undefined') {

this.container.addEventListener('touchstart', this.\_handleContainerMousedownAndTouchstartBound);

this.container.addEventListener('touchend', this.\_handleContainerMouseupAndTouchendBound);

}

}

/\*\*

\* Remove Event Handlers

\*/

}, {

key: "\_removeEventHandlers",

value: function \_removeEventHandlers() {

this.el.removeEventListener('blur', this.\_handleInputBlurBound);

this.el.removeEventListener('keyup', this.\_handleInputKeyupAndFocusBound);

this.el.removeEventListener('focus', this.\_handleInputKeyupAndFocusBound);

this.el.removeEventListener('keydown', this.\_handleInputKeydownBound);

this.el.removeEventListener('click', this.\_handleInputClickBound);

this.container.removeEventListener('mousedown', this.\_handleContainerMousedownAndTouchstartBound);

this.container.removeEventListener('mouseup', this.\_handleContainerMouseupAndTouchendBound);

if (typeof window.ontouchstart !== 'undefined') {

this.container.removeEventListener('touchstart', this.\_handleContainerMousedownAndTouchstartBound);

this.container.removeEventListener('touchend', this.\_handleContainerMouseupAndTouchendBound);

}

}

/\*\*

\* Setup dropdown

\*/

}, {

key: "\_setupDropdown",

value: function \_setupDropdown() {

var \_this38 = this;

this.container = document.createElement('ul');

this.container.id = "autocomplete-options-" + M.guid();

$(this.container).addClass('autocomplete-content dropdown-content');

this.$inputField.append(this.container);

this.el.setAttribute('data-target', this.container.id);

this.dropdown = M.Dropdown.init(this.el, {

autoFocus: false,

closeOnClick: false,

coverTrigger: false,

onItemClick: function (itemEl) {

\_this38.selectOption($(itemEl));

}

});

// Sketchy removal of dropdown click handler

this.el.removeEventListener('click', this.dropdown.\_handleClickBound);

}

/\*\*

\* Remove dropdown

\*/

}, {

key: "\_removeDropdown",

value: function \_removeDropdown() {

this.container.parentNode.removeChild(this.container);

}

/\*\*

\* Handle Input Blur

\*/

}, {

key: "\_handleInputBlur",

value: function \_handleInputBlur() {

if (!this.\_mousedown) {

this.close();

this.\_resetAutocomplete();

}

}

/\*\*

\* Handle Input Keyup and Focus

\* @param {Event} e

\*/

}, {

key: "\_handleInputKeyupAndFocus",

value: function \_handleInputKeyupAndFocus(e) {

if (e.type === 'keyup') {

Autocomplete.\_keydown = false;

}

this.count = 0;

var val = this.el.value.toLowerCase();

// Don't capture enter or arrow key usage.

if (e.keyCode === 13 || e.keyCode === 38 || e.keyCode === 40) {

return;

}

// Check if the input isn't empty

// Check if focus triggered by tab

if (this.oldVal !== val && (M.tabPressed || e.type !== 'focus')) {

this.open();

}

// Update oldVal

this.oldVal = val;

}

/\*\*

\* Handle Input Keydown

\* @param {Event} e

\*/

}, {

key: "\_handleInputKeydown",

value: function \_handleInputKeydown(e) {

Autocomplete.\_keydown = true;

// Arrow keys and enter key usage

var keyCode = e.keyCode,

liElement = void 0,

numItems = $(this.container).children('li').length;

// select element on Enter

if (keyCode === M.keys.ENTER && this.activeIndex >= 0) {

liElement = $(this.container).children('li').eq(this.activeIndex);

if (liElement.length) {

this.selectOption(liElement);

e.preventDefault();

}

return;

}

// Capture up and down key

if (keyCode === M.keys.ARROW\_UP || keyCode === M.keys.ARROW\_DOWN) {

e.preventDefault();

if (keyCode === M.keys.ARROW\_UP && this.activeIndex > 0) {

this.activeIndex--;

}

if (keyCode === M.keys.ARROW\_DOWN && this.activeIndex < numItems - 1) {

this.activeIndex++;

}

this.$active.removeClass('active');

if (this.activeIndex >= 0) {

this.$active = $(this.container).children('li').eq(this.activeIndex);

this.$active.addClass('active');

}

}

}

/\*\*

\* Handle Input Click

\* @param {Event} e

\*/

}, {

key: "\_handleInputClick",

value: function \_handleInputClick(e) {

this.open();

}

/\*\*

\* Handle Container Mousedown and Touchstart

\* @param {Event} e

\*/

}, {

key: "\_handleContainerMousedownAndTouchstart",

value: function \_handleContainerMousedownAndTouchstart(e) {

this.\_mousedown = true;

}

/\*\*

\* Handle Container Mouseup and Touchend

\* @param {Event} e

\*/

}, {

key: "\_handleContainerMouseupAndTouchend",

value: function \_handleContainerMouseupAndTouchend(e) {

this.\_mousedown = false;

}

/\*\*

\* Highlight partial match

\*/

}, {

key: "\_highlight",

value: function \_highlight(string, $el) {

var img = $el.find('img');

var matchStart = $el.text().toLowerCase().indexOf('' + string.toLowerCase() + ''),

matchEnd = matchStart + string.length - 1,

beforeMatch = $el.text().slice(0, matchStart),

matchText = $el.text().slice(matchStart, matchEnd + 1),

afterMatch = $el.text().slice(matchEnd + 1);

$el.html("<span>" + beforeMatch + "<span class='highlight'>" + matchText + "</span>" + afterMatch + "</span>");

if (img.length) {

$el.prepend(img);

}

}

/\*\*

\* Reset current element position

\*/

}, {

key: "\_resetCurrentElement",

value: function \_resetCurrentElement() {

this.activeIndex = -1;

this.$active.removeClass('active');

}

/\*\*

\* Reset autocomplete elements

\*/

}, {

key: "\_resetAutocomplete",

value: function \_resetAutocomplete() {

$(this.container).empty();

this.\_resetCurrentElement();

this.oldVal = null;

this.isOpen = false;

this.\_mousedown = false;

}

/\*\*

\* Select autocomplete option

\* @param {Element} el Autocomplete option list item element

\*/

}, {

key: "selectOption",

value: function selectOption(el) {

var text = el.text().trim();

this.el.value = text;

this.$el.trigger('change');

this.\_resetAutocomplete();

this.close();

// Handle onAutocomplete callback.

if (typeof this.options.onAutocomplete === 'function') {

this.options.onAutocomplete.call(this, text);

}

}

/\*\*

\* Render dropdown content

\* @param {Object} data data set

\* @param {String} val current input value

\*/

}, {

key: "\_renderDropdown",

value: function \_renderDropdown(data, val) {

var \_this39 = this;

this.\_resetAutocomplete();

var matchingData = [];

// Gather all matching data

for (var key in data) {

if (data.hasOwnProperty(key) && key.toLowerCase().indexOf(val) !== -1) {

// Break if past limit

if (this.count >= this.options.limit) {

break;

}

var entry = {

data: data[key],

key: key

};

matchingData.push(entry);

this.count++;

}

}

// Sort

if (this.options.sortFunction) {

var sortFunctionBound = function (a, b) {

return \_this39.options.sortFunction(a.key.toLowerCase(), b.key.toLowerCase(), val.toLowerCase());

};

matchingData.sort(sortFunctionBound);

}

// Render

for (var i = 0; i < matchingData.length; i++) {

var \_entry = matchingData[i];

var $autocompleteOption = $('<li></li>');

if (!!\_entry.data) {

$autocompleteOption.append("<img src=\"" + \_entry.data + "\" class=\"right circle\"><span>" + \_entry.key + "</span>");

} else {

$autocompleteOption.append('<span>' + \_entry.key + '</span>');

}

$(this.container).append($autocompleteOption);

this.\_highlight(val, $autocompleteOption);

}

}

/\*\*

\* Open Autocomplete Dropdown

\*/

}, {

key: "open",

value: function open() {

var val = this.el.value.toLowerCase();

this.\_resetAutocomplete();

if (val.length >= this.options.minLength) {

this.isOpen = true;

this.\_renderDropdown(this.options.data, val);

}

// Open dropdown

if (!this.dropdown.isOpen) {

this.dropdown.open();

} else {

// Recalculate dropdown when its already open

this.dropdown.recalculateDimensions();

}

}

/\*\*

\* Close Autocomplete Dropdown

\*/

}, {

key: "close",

value: function close() {

this.dropdown.close();

}

/\*\*

\* Update Data

\* @param {Object} data

\*/

}, {

key: "updateData",

value: function updateData(data) {

var val = this.el.value.toLowerCase();

this.options.data = data;

if (this.isOpen) {

this.\_renderDropdown(data, val);

}

}

}], [{

key: "init",

value: function init(els, options) {

return \_get(Autocomplete.\_\_proto\_\_ || Object.getPrototypeOf(Autocomplete), "init", this).call(this, this, els, options);

}

/\*\*

\* Get Instance

\*/

}, {

key: "getInstance",

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_Autocomplete;

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return Autocomplete;

}(Component);

/\*\*

\* @static

\* @memberof Autocomplete

\*/

Autocomplete.\_keydown = false;

M.Autocomplete = Autocomplete;

if (M.jQueryLoaded) {

M.initializeJqueryWrapper(Autocomplete, 'autocomplete', 'M\_Autocomplete');

}

})(cash);

;(function ($) {

// Function to update labels of text fields

M.updateTextFields = function () {

var input\_selector = 'input[type=text], input[type=password], input[type=email], input[type=url], input[type=tel], input[type=number], input[type=search], input[type=date], input[type=time], textarea';

$(input\_selector).each(function (element, index) {

var $this = $(this);

if (element.value.length > 0 || $(element).is(':focus') || element.autofocus || $this.attr('placeholder') !== null) {

$this.siblings('label').addClass('active');

} else if (element.validity) {

$this.siblings('label').toggleClass('active', element.validity.badInput === true);

} else {

$this.siblings('label').removeClass('active');

}

});

};

M.validate\_field = function (object) {

var hasLength = object.attr('data-length') !== null;

var lenAttr = parseInt(object.attr('data-length'));

var len = object[0].value.length;

if (len === 0 && object[0].validity.badInput === false && !object.is(':required')) {

if (object.hasClass('validate')) {

object.removeClass('valid');

object.removeClass('invalid');

}

} else {

if (object.hasClass('validate')) {

// Check for character counter attributes

if (object.is(':valid') && hasLength && len <= lenAttr || object.is(':valid') && !hasLength) {

object.removeClass('invalid');

object.addClass('valid');

} else {

object.removeClass('valid');

object.addClass('invalid');

}

}

}

};

M.textareaAutoResize = function ($textarea) {

// Wrap if native element

if ($textarea instanceof Element) {

$textarea = $($textarea);

}

if (!$textarea.length) {

console.error('No textarea element found');

return;

}

// Textarea Auto Resize

var hiddenDiv = $('.hiddendiv').first();

if (!hiddenDiv.length) {

hiddenDiv = $('<div class="hiddendiv common"></div>');

$('body').append(hiddenDiv);

}

// Set font properties of hiddenDiv

var fontFamily = $textarea.css('font-family');

var fontSize = $textarea.css('font-size');

var lineHeight = $textarea.css('line-height');

// Firefox can't handle padding shorthand.

var paddingTop = $textarea.css('padding-top');

var paddingRight = $textarea.css('padding-right');

var paddingBottom = $textarea.css('padding-bottom');

var paddingLeft = $textarea.css('padding-left');

if (fontSize) {

hiddenDiv.css('font-size', fontSize);

}

if (fontFamily) {

hiddenDiv.css('font-family', fontFamily);

}

if (lineHeight) {

hiddenDiv.css('line-height', lineHeight);

}

if (paddingTop) {

hiddenDiv.css('padding-top', paddingTop);

}

if (paddingRight) {

hiddenDiv.css('padding-right', paddingRight);

}

if (paddingBottom) {

hiddenDiv.css('padding-bottom', paddingBottom);

}

if (paddingLeft) {

hiddenDiv.css('padding-left', paddingLeft);

}

// Set original-height, if none

if (!$textarea.data('original-height')) {

$textarea.data('original-height', $textarea.height());

}

if ($textarea.attr('wrap') === 'off') {

hiddenDiv.css('overflow-wrap', 'normal').css('white-space', 'pre');

}

hiddenDiv.text($textarea[0].value + '\n');

var content = hiddenDiv.html().replace(/\n/g, '<br>');

hiddenDiv.html(content);

// When textarea is hidden, width goes crazy.

// Approximate with half of window size

if ($textarea[0].offsetWidth > 0 && $textarea[0].offsetHeight > 0) {

hiddenDiv.css('width', $textarea.width() + 'px');

} else {

hiddenDiv.css('width', window.innerWidth / 2 + 'px');

}

/\*\*

\* Resize if the new height is greater than the

\* original height of the textarea

\*/

if ($textarea.data('original-height') <= hiddenDiv.innerHeight()) {

$textarea.css('height', hiddenDiv.innerHeight() + 'px');

} else if ($textarea[0].value.length < $textarea.data('previous-length')) {

/\*\*

\* In case the new height is less than original height, it

\* means the textarea has less text than before

\* So we set the height to the original one

\*/

$textarea.css('height', $textarea.data('original-height') + 'px');

}

$textarea.data('previous-length', $textarea[0].value.length);

};

$(document).ready(function () {

// Text based inputs

var input\_selector = 'input[type=text], input[type=password], input[type=email], input[type=url], input[type=tel], input[type=number], input[type=search], input[type=date], input[type=time], textarea';

// Add active if form auto complete

$(document).on('change', input\_selector, function () {

if (this.value.length !== 0 || $(this).attr('placeholder') !== null) {

$(this).siblings('label').addClass('active');

}

M.validate\_field($(this));

});

// Add active if input element has been pre-populated on document ready

$(document).ready(function () {

M.updateTextFields();

});

// HTML DOM FORM RESET handling

$(document).on('reset', function (e) {

var formReset = $(e.target);

if (formReset.is('form')) {

formReset.find(input\_selector).removeClass('valid').removeClass('invalid');

formReset.find(input\_selector).each(function (e) {

if (this.value.length) {

$(this).siblings('label').removeClass('active');

}

});

// Reset select (after native reset)

setTimeout(function () {

formReset.find('select').each(function () {

// check if initialized

if (this.M\_FormSelect) {

$(this).trigger('change');

}

});

}, 0);

}

});

/\*\*

\* Add active when element has focus

\* @param {Event} e

\*/

document.addEventListener('focus', function (e) {

if ($(e.target).is(input\_selector)) {

$(e.target).siblings('label, .prefix').addClass('active');

}

}, true);

/\*\*

\* Remove active when element is blurred

\* @param {Event} e

\*/

document.addEventListener('blur', function (e) {

var $inputElement = $(e.target);

if ($inputElement.is(input\_selector)) {

var selector = '.prefix';

if ($inputElement[0].value.length === 0 && $inputElement[0].validity.badInput !== true && $inputElement.attr('placeholder') === null) {

selector += ', label';

}

$inputElement.siblings(selector).removeClass('active');

M.validate\_field($inputElement);

}

}, true);

// Radio and Checkbox focus class

var radio\_checkbox = 'input[type=radio], input[type=checkbox]';

$(document).on('keyup', radio\_checkbox, function (e) {

// TAB, check if tabbing to radio or checkbox.

if (e.which === M.keys.TAB) {

$(this).addClass('tabbed');

var $this = $(this);

$this.one('blur', function (e) {

$(this).removeClass('tabbed');

});

return;

}

});

var text\_area\_selector = '.materialize-textarea';

$(text\_area\_selector).each(function () {

var $textarea = $(this);

/\*\*

\* Resize textarea on document load after storing

\* the original height and the original length

\*/

$textarea.data('original-height', $textarea.height());

$textarea.data('previous-length', this.value.length);

M.textareaAutoResize($textarea);

});

$(document).on('keyup', text\_area\_selector, function () {

M.textareaAutoResize($(this));

});

$(document).on('keydown', text\_area\_selector, function () {

M.textareaAutoResize($(this));

});

// File Input Path

$(document).on('change', '.file-field input[type="file"]', function () {

var file\_field = $(this).closest('.file-field');

var path\_input = file\_field.find('input.file-path');

var files = $(this)[0].files;

var file\_names = [];

for (var i = 0; i < files.length; i++) {

file\_names.push(files[i].name);

}

path\_input[0].value = file\_names.join(', ');

path\_input.trigger('change');

});

}); // End of $(document).ready

})(cash);

;(function ($, anim) {

'use strict';

var \_defaults = {

indicators: true,

height: 400,

duration: 500,

interval: 6000

};

/\*\*

\* @class

\*

\*/

var Slider = function (\_Component11) {

\_inherits(Slider, \_Component11);

/\*\*

\* Construct Slider instance and set up overlay

\* @constructor

\* @param {Element} el

\* @param {Object} options

\*/

function Slider(el, options) {

\_classCallCheck(this, Slider);

var \_this40 = \_possibleConstructorReturn(this, (Slider.\_\_proto\_\_ || Object.getPrototypeOf(Slider)).call(this, Slider, el, options));

\_this40.el.M\_Slider = \_this40;

/\*\*

\* Options for the modal

\* @member Slider#options

\* @prop {Boolean} [indicators=true] - Show indicators

\* @prop {Number} [height=400] - height of slider

\* @prop {Number} [duration=500] - Length in ms of slide transition

\* @prop {Number} [interval=6000] - Length in ms of slide interval

\*/

\_this40.options = $.extend({}, Slider.defaults, options);

// setup

\_this40.$slider = \_this40.$el.find('.slides');

\_this40.$slides = \_this40.$slider.children('li');

\_this40.activeIndex = \_this40.$slides.filter(function (item) {

return $(item).hasClass('active');

}).first().index();

if (\_this40.activeIndex != -1) {

\_this40.$active = \_this40.$slides.eq(\_this40.activeIndex);

}

\_this40.\_setSliderHeight();

// Set initial positions of captions

\_this40.$slides.find('.caption').each(function (el) {

\_this40.\_animateCaptionIn(el, 0);

});

// Move img src into background-image

\_this40.$slides.find('img').each(function (el) {

var placeholderBase64 = '';

if ($(el).attr('src') !== placeholderBase64) {

$(el).css('background-image', 'url("' + $(el).attr('src') + '")');

$(el).attr('src', placeholderBase64);

}

});

\_this40.\_setupIndicators();

// Show active slide

if (\_this40.$active) {

\_this40.$active.css('display', 'block');

} else {

\_this40.$slides.first().addClass('active');

anim({

targets: \_this40.$slides.first()[0],

opacity: 1,

duration: \_this40.options.duration,

easing: 'easeOutQuad'

});

\_this40.activeIndex = 0;

\_this40.$active = \_this40.$slides.eq(\_this40.activeIndex);

// Update indicators

if (\_this40.options.indicators) {

\_this40.$indicators.eq(\_this40.activeIndex).addClass('active');

}

}

// Adjust height to current slide

\_this40.$active.find('img').each(function (el) {

anim({

targets: \_this40.$active.find('.caption')[0],

opacity: 1,

translateX: 0,

translateY: 0,

duration: \_this40.options.duration,

easing: 'easeOutQuad'

});

});

\_this40.\_setupEventHandlers();

// auto scroll

\_this40.start();

return \_this40;

}

\_createClass(Slider, [{

key: "destroy",

/\*\*

\* Teardown component

\*/

value: function destroy() {

this.pause();

this.\_removeIndicators();

this.\_removeEventHandlers();

this.el.M\_Slider = undefined;

}

/\*\*

\* Setup Event Handlers

\*/

}, {

key: "\_setupEventHandlers",

value: function \_setupEventHandlers() {

var \_this41 = this;

this.\_handleIntervalBound = this.\_handleInterval.bind(this);

this.\_handleIndicatorClickBound = this.\_handleIndicatorClick.bind(this);

if (this.options.indicators) {

this.$indicators.each(function (el) {

el.addEventListener('click', \_this41.\_handleIndicatorClickBound);

});

}

}

/\*\*

\* Remove Event Handlers

\*/

}, {

key: "\_removeEventHandlers",

value: function \_removeEventHandlers() {

var \_this42 = this;

if (this.options.indicators) {

this.$indicators.each(function (el) {

el.removeEventListener('click', \_this42.\_handleIndicatorClickBound);

});

}

}

/\*\*

\* Handle indicator click

\* @param {Event} e

\*/

}, {

key: "\_handleIndicatorClick",

value: function \_handleIndicatorClick(e) {

var currIndex = $(e.target).index();

this.set(currIndex);

}

/\*\*

\* Handle Interval

\*/

}, {

key: "\_handleInterval",

value: function \_handleInterval() {

var newActiveIndex = this.$slider.find('.active').index();

if (this.$slides.length === newActiveIndex + 1) newActiveIndex = 0;

// loop to start

else newActiveIndex += 1;

this.set(newActiveIndex);

}

/\*\*

\* Animate in caption

\* @param {Element} caption

\* @param {Number} duration

\*/

}, {

key: "\_animateCaptionIn",

value: function \_animateCaptionIn(caption, duration) {

var animOptions = {

targets: caption,

opacity: 0,

duration: duration,

easing: 'easeOutQuad'

};

if ($(caption).hasClass('center-align')) {

animOptions.translateY = -100;

} else if ($(caption).hasClass('right-align')) {

animOptions.translateX = 100;

} else if ($(caption).hasClass('left-align')) {

animOptions.translateX = -100;

}

anim(animOptions);

}

/\*\*

\* Set height of slider

\*/

}, {

key: "\_setSliderHeight",

value: function \_setSliderHeight() {

// If fullscreen, do nothing

if (!this.$el.hasClass('fullscreen')) {

if (this.options.indicators) {

// Add height if indicators are present

this.$el.css('height', this.options.height + 40 + 'px');

} else {

this.$el.css('height', this.options.height + 'px');

}

this.$slider.css('height', this.options.height + 'px');

}

}

/\*\*

\* Setup indicators

\*/

}, {

key: "\_setupIndicators",

value: function \_setupIndicators() {

var \_this43 = this;

if (this.options.indicators) {

this.$indicators = $('<ul class="indicators"></ul>');

this.$slides.each(function (el, index) {

var $indicator = $('<li class="indicator-item"></li>');

\_this43.$indicators.append($indicator[0]);

});

this.$el.append(this.$indicators[0]);

this.$indicators = this.$indicators.children('li.indicator-item');

}

}

/\*\*

\* Remove indicators

\*/

}, {

key: "\_removeIndicators",

value: function \_removeIndicators() {

this.$el.find('ul.indicators').remove();

}

/\*\*

\* Cycle to nth item

\* @param {Number} index

\*/

}, {

key: "set",

value: function set(index) {

var \_this44 = this;

// Wrap around indices.

if (index >= this.$slides.length) index = 0;else if (index < 0) index = this.$slides.length - 1;

// Only do if index changes

if (this.activeIndex != index) {

this.$active = this.$slides.eq(this.activeIndex);

var $caption = this.$active.find('.caption');

this.$active.removeClass('active');

anim({

targets: this.$active[0],

opacity: 0,

duration: this.options.duration,

easing: 'easeOutQuad',

complete: function () {

\_this44.$slides.not('.active').each(function (el) {

anim({

targets: el,

opacity: 0,

translateX: 0,

translateY: 0,

duration: 0,

easing: 'easeOutQuad'

});

});

}

});

this.\_animateCaptionIn($caption[0], this.options.duration);

// Update indicators

if (this.options.indicators) {

this.$indicators.eq(this.activeIndex).removeClass('active');

this.$indicators.eq(index).addClass('active');

}

anim({

targets: this.$slides.eq(index)[0],

opacity: 1,

duration: this.options.duration,

easing: 'easeOutQuad'

});

anim({

targets: this.$slides.eq(index).find('.caption')[0],

opacity: 1,

translateX: 0,

translateY: 0,

duration: this.options.duration,

delay: this.options.duration,

easing: 'easeOutQuad'

});

this.$slides.eq(index).addClass('active');

this.activeIndex = index;

// Reset interval

this.start();

}

}

/\*\*

\* Pause slider interval

\*/

}, {

key: "pause",

value: function pause() {

clearInterval(this.interval);

}

/\*\*

\* Start slider interval

\*/

}, {

key: "start",

value: function start() {

clearInterval(this.interval);

this.interval = setInterval(this.\_handleIntervalBound, this.options.duration + this.options.interval);

}

/\*\*

\* Move to next slide

\*/

}, {

key: "next",

value: function next() {

var newIndex = this.activeIndex + 1;

// Wrap around indices.

if (newIndex >= this.$slides.length) newIndex = 0;else if (newIndex < 0) newIndex = this.$slides.length - 1;

this.set(newIndex);

}

/\*\*

\* Move to previous slide

\*/

}, {

key: "prev",

value: function prev() {

var newIndex = this.activeIndex - 1;

// Wrap around indices.

if (newIndex >= this.$slides.length) newIndex = 0;else if (newIndex < 0) newIndex = this.$slides.length - 1;

this.set(newIndex);

}

}], [{

key: "init",

value: function init(els, options) {

return \_get(Slider.\_\_proto\_\_ || Object.getPrototypeOf(Slider), "init", this).call(this, this, els, options);

}

/\*\*

\* Get Instance

\*/

}, {

key: "getInstance",

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_Slider;

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return Slider;

}(Component);

M.Slider = Slider;

if (M.jQueryLoaded) {

M.initializeJqueryWrapper(Slider, 'slider', 'M\_Slider');

}

})(cash, M.anime);

;(function ($, anim) {

$(document).on('click', '.card', function (e) {

if ($(this).children('.card-reveal').length) {

var $card = $(e.target).closest('.card');

if ($card.data('initialOverflow') === undefined) {

$card.data('initialOverflow', $card.css('overflow') === undefined ? '' : $card.css('overflow'));

}

var $cardReveal = $(this).find('.card-reveal');

if ($(e.target).is($('.card-reveal .card-title')) || $(e.target).is($('.card-reveal .card-title i'))) {

// Make Reveal animate down and display none

anim({

targets: $cardReveal[0],

translateY: 0,

duration: 225,

easing: 'easeInOutQuad',

complete: function (anim) {

var el = anim.animatables[0].target;

$(el).css({ display: 'none' });

$card.css('overflow', $card.data('initialOverflow'));

}

});

} else if ($(e.target).is($('.card .activator')) || $(e.target).is($('.card .activator i'))) {

$card.css('overflow', 'hidden');

$cardReveal.css({ display: 'block' });

anim({

targets: $cardReveal[0],

translateY: '-100%',

duration: 300,

easing: 'easeInOutQuad'

});

}

}

});

})(cash, M.anime);

;(function ($) {

'use strict';

var \_defaults = {

data: [],

placeholder: '',

secondaryPlaceholder: '',

autocompleteOptions: {},

limit: Infinity,

onChipAdd: null,

onChipSelect: null,

onChipDelete: null

};

/\*\*

\* @typedef {Object} chip

\* @property {String} tag chip tag string

\* @property {String} [image] chip avatar image string

\*/

/\*\*

\* @class

\*

\*/

var Chips = function (\_Component12) {

\_inherits(Chips, \_Component12);

/\*\*

\* Construct Chips instance and set up overlay

\* @constructor

\* @param {Element} el

\* @param {Object} options

\*/

function Chips(el, options) {

\_classCallCheck(this, Chips);

var \_this45 = \_possibleConstructorReturn(this, (Chips.\_\_proto\_\_ || Object.getPrototypeOf(Chips)).call(this, Chips, el, options));

\_this45.el.M\_Chips = \_this45;

/\*\*

\* Options for the modal

\* @member Chips#options

\* @prop {Array} data

\* @prop {String} placeholder

\* @prop {String} secondaryPlaceholder

\* @prop {Object} autocompleteOptions

\*/

\_this45.options = $.extend({}, Chips.defaults, options);

\_this45.$el.addClass('chips input-field');

\_this45.chipsData = [];

\_this45.$chips = $();

\_this45.\_setupInput();

\_this45.hasAutocomplete = Object.keys(\_this45.options.autocompleteOptions).length > 0;

// Set input id

if (!\_this45.$input.attr('id')) {

\_this45.$input.attr('id', M.guid());

}

// Render initial chips

if (\_this45.options.data.length) {

\_this45.chipsData = \_this45.options.data;

\_this45.\_renderChips(\_this45.chipsData);

}

// Setup autocomplete if needed

if (\_this45.hasAutocomplete) {

\_this45.\_setupAutocomplete();

}

\_this45.\_setPlaceholder();

\_this45.\_setupLabel();

\_this45.\_setupEventHandlers();

return \_this45;

}

\_createClass(Chips, [{

key: "getData",

/\*\*

\* Get Chips Data

\*/

value: function getData() {

return this.chipsData;

}

/\*\*

\* Teardown component

\*/

}, {

key: "destroy",

value: function destroy() {

this.\_removeEventHandlers();

this.$chips.remove();

this.el.M\_Chips = undefined;

}

/\*\*

\* Setup Event Handlers

\*/

}, {

key: "\_setupEventHandlers",

value: function \_setupEventHandlers() {

this.\_handleChipClickBound = this.\_handleChipClick.bind(this);

this.\_handleInputKeydownBound = this.\_handleInputKeydown.bind(this);

this.\_handleInputFocusBound = this.\_handleInputFocus.bind(this);

this.\_handleInputBlurBound = this.\_handleInputBlur.bind(this);

this.el.addEventListener('click', this.\_handleChipClickBound);

document.addEventListener('keydown', Chips.\_handleChipsKeydown);

document.addEventListener('keyup', Chips.\_handleChipsKeyup);

this.el.addEventListener('blur', Chips.\_handleChipsBlur, true);

this.$input[0].addEventListener('focus', this.\_handleInputFocusBound);

this.$input[0].addEventListener('blur', this.\_handleInputBlurBound);

this.$input[0].addEventListener('keydown', this.\_handleInputKeydownBound);

}

/\*\*

\* Remove Event Handlers

\*/

}, {

key: "\_removeEventHandlers",

value: function \_removeEventHandlers() {

this.el.removeEventListener('click', this.\_handleChipClickBound);

document.removeEventListener('keydown', Chips.\_handleChipsKeydown);

document.removeEventListener('keyup', Chips.\_handleChipsKeyup);

this.el.removeEventListener('blur', Chips.\_handleChipsBlur, true);

this.$input[0].removeEventListener('focus', this.\_handleInputFocusBound);

this.$input[0].removeEventListener('blur', this.\_handleInputBlurBound);

this.$input[0].removeEventListener('keydown', this.\_handleInputKeydownBound);

}

/\*\*

\* Handle Chip Click

\* @param {Event} e

\*/

}, {

key: "\_handleChipClick",

value: function \_handleChipClick(e) {

var $chip = $(e.target).closest('.chip');

var clickedClose = $(e.target).is('.close');

if ($chip.length) {

var index = $chip.index();

if (clickedClose) {

// delete chip

this.deleteChip(index);

this.$input[0].focus();

} else {

// select chip

this.selectChip(index);

}

// Default handle click to focus on input

} else {

this.$input[0].focus();

}

}

/\*\*

\* Handle Chips Keydown

\* @param {Event} e

\*/

}, {

key: "\_handleInputFocus",

/\*\*

\* Handle Input Focus

\*/

value: function \_handleInputFocus() {

this.$el.addClass('focus');

}

/\*\*

\* Handle Input Blur

\*/

}, {

key: "\_handleInputBlur",

value: function \_handleInputBlur() {

this.$el.removeClass('focus');

}

/\*\*

\* Handle Input Keydown

\* @param {Event} e

\*/

}, {

key: "\_handleInputKeydown",

value: function \_handleInputKeydown(e) {

Chips.\_keydown = true;

// enter

if (e.keyCode === 13) {

// Override enter if autocompleting.

if (this.hasAutocomplete && this.autocomplete && this.autocomplete.isOpen) {

return;

}

e.preventDefault();

this.addChip({

tag: this.$input[0].value

});

this.$input[0].value = '';

// delete or left

} else if ((e.keyCode === 8 || e.keyCode === 37) && this.$input[0].value === '' && this.chipsData.length) {

e.preventDefault();

this.selectChip(this.chipsData.length - 1);

}

}

/\*\*

\* Render Chip

\* @param {chip} chip

\* @return {Element}

\*/

}, {

key: "\_renderChip",

value: function \_renderChip(chip) {

if (!chip.tag) {

return;

}

var renderedChip = document.createElement('div');

var closeIcon = document.createElement('i');

renderedChip.classList.add('chip');

renderedChip.textContent = chip.tag;

renderedChip.setAttribute('tabindex', 0);

$(closeIcon).addClass('material-icons close');

closeIcon.textContent = 'close';

// attach image if needed

if (chip.image) {

var img = document.createElement('img');

img.setAttribute('src', chip.image);

renderedChip.insertBefore(img, renderedChip.firstChild);

}

renderedChip.appendChild(closeIcon);

return renderedChip;

}

/\*\*

\* Render Chips

\*/

}, {

key: "\_renderChips",

value: function \_renderChips() {

this.$chips.remove();

for (var i = 0; i < this.chipsData.length; i++) {

var chipEl = this.\_renderChip(this.chipsData[i]);

this.$el.append(chipEl);

this.$chips.add(chipEl);

}

// move input to end

this.$el.append(this.$input[0]);

}

/\*\*

\* Setup Autocomplete

\*/

}, {

key: "\_setupAutocomplete",

value: function \_setupAutocomplete() {

var \_this46 = this;

this.options.autocompleteOptions.onAutocomplete = function (val) {

\_this46.addChip({

tag: val

});

\_this46.$input[0].value = '';

\_this46.$input[0].focus();

};

this.autocomplete = M.Autocomplete.init(this.$input[0], this.options.autocompleteOptions);

}

/\*\*

\* Setup Input

\*/

}, {

key: "\_setupInput",

value: function \_setupInput() {

this.$input = this.$el.find('input');

if (!this.$input.length) {

this.$input = $('<input></input>');

this.$el.append(this.$input);

}

this.$input.addClass('input');

}

/\*\*

\* Setup Label

\*/

}, {

key: "\_setupLabel",

value: function \_setupLabel() {

this.$label = this.$el.find('label');

if (this.$label.length) {

this.$label.setAttribute('for', this.$input.attr('id'));

}

}

/\*\*

\* Set placeholder

\*/

}, {

key: "\_setPlaceholder",

value: function \_setPlaceholder() {

if (this.chipsData !== undefined && !this.chipsData.length && this.options.placeholder) {

$(this.$input).prop('placeholder', this.options.placeholder);

} else if ((this.chipsData === undefined || !!this.chipsData.length) && this.options.secondaryPlaceholder) {

$(this.$input).prop('placeholder', this.options.secondaryPlaceholder);

}

}

/\*\*

\* Check if chip is valid

\* @param {chip} chip

\*/

}, {

key: "\_isValid",

value: function \_isValid(chip) {

if (chip.hasOwnProperty('tag') && chip.tag !== '') {

var exists = false;

for (var i = 0; i < this.chipsData.length; i++) {

if (this.chipsData[i].tag === chip.tag) {

exists = true;

break;

}

}

return !exists;

}

return false;

}

/\*\*

\* Add chip

\* @param {chip} chip

\*/

}, {

key: "addChip",

value: function addChip(chip) {

if (!this.\_isValid(chip) || this.chipsData.length >= this.options.limit) {

return;

}

var renderedChip = this.\_renderChip(chip);

this.$chips.add(renderedChip);

this.chipsData.push(chip);

$(this.$input).before(renderedChip);

this.\_setPlaceholder();

// fire chipAdd callback

if (typeof this.options.onChipAdd === 'function') {

this.options.onChipAdd.call(this, this.$el, renderedChip);

}

}

/\*\*

\* Delete chip

\* @param {Number} chip

\*/

}, {

key: "deleteChip",

value: function deleteChip(chipIndex) {

var $chip = this.$chips.eq(chipIndex);

this.$chips.eq(chipIndex).remove();

this.$chips = this.$chips.filter(function (el) {

return $(el).index() >= 0;

});

this.chipsData.splice(chipIndex, 1);

this.\_setPlaceholder();

// fire chipDelete callback

if (typeof this.options.onChipDelete === 'function') {

this.options.onChipDelete.call(this, this.$el, $chip[0]);

}

}

/\*\*

\* Select chip

\* @param {Number} chip

\*/

}, {

key: "selectChip",

value: function selectChip(chipIndex) {

var $chip = this.$chips.eq(chipIndex);

this.\_selectedChip = $chip;

$chip[0].focus();

// fire chipSelect callback

if (typeof this.options.onChipSelect === 'function') {

this.options.onChipSelect.call(this, this.$el, $chip[0]);

}

}

}], [{

key: "init",

value: function init(els, options) {

return \_get(Chips.\_\_proto\_\_ || Object.getPrototypeOf(Chips), "init", this).call(this, this, els, options);

}

/\*\*

\* Get Instance

\*/

}, {

key: "getInstance",

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_Chips;

}

}, {

key: "\_handleChipsKeydown",

value: function \_handleChipsKeydown(e) {

Chips.\_keydown = true;

var $chips = $(e.target).closest('.chips');

var chipsKeydown = e.target && $chips.length;

// Don't handle keydown inputs on input and textarea

if ($(e.target).is('input, textarea') || !chipsKeydown) {

return;

}

var currChips = $chips[0].M\_Chips;

// backspace and delete

if (e.keyCode === 8 || e.keyCode === 46) {

e.preventDefault();

var selectIndex = currChips.chipsData.length;

if (currChips.\_selectedChip) {

var index = currChips.\_selectedChip.index();

currChips.deleteChip(index);

currChips.\_selectedChip = null;

// Make sure selectIndex doesn't go negative

selectIndex = Math.max(index - 1, 0);

}

if (currChips.chipsData.length) {

currChips.selectChip(selectIndex);

}

// left arrow key

} else if (e.keyCode === 37) {

if (currChips.\_selectedChip) {

var \_selectIndex = currChips.\_selectedChip.index() - 1;

if (\_selectIndex < 0) {

return;

}

currChips.selectChip(\_selectIndex);

}

// right arrow key

} else if (e.keyCode === 39) {

if (currChips.\_selectedChip) {

var \_selectIndex2 = currChips.\_selectedChip.index() + 1;

if (\_selectIndex2 >= currChips.chipsData.length) {

currChips.$input[0].focus();

} else {

currChips.selectChip(\_selectIndex2);

}

}

}

}

/\*\*

\* Handle Chips Keyup

\* @param {Event} e

\*/

}, {

key: "\_handleChipsKeyup",

value: function \_handleChipsKeyup(e) {

Chips.\_keydown = false;

}

/\*\*

\* Handle Chips Blur

\* @param {Event} e

\*/

}, {

key: "\_handleChipsBlur",

value: function \_handleChipsBlur(e) {

if (!Chips.\_keydown) {

var $chips = $(e.target).closest('.chips');

var currChips = $chips[0].M\_Chips;

currChips.\_selectedChip = null;

}

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return Chips;

}(Component);

/\*\*

\* @static

\* @memberof Chips

\*/

Chips.\_keydown = false;

M.Chips = Chips;

if (M.jQueryLoaded) {

M.initializeJqueryWrapper(Chips, 'chips', 'M\_Chips');

}

$(document).ready(function () {

// Handle removal of static chips.

$(document.body).on('click', '.chip .close', function () {

var $chips = $(this).closest('.chips');

if ($chips.length && $chips[0].M\_Chips) {

return;

}

$(this).closest('.chip').remove();

});

});

})(cash);

;(function ($) {

'use strict';

var \_defaults = {

top: 0,

bottom: Infinity,

offset: 0,

onPositionChange: null

};

/\*\*

\* @class

\*

\*/

var Pushpin = function (\_Component13) {

\_inherits(Pushpin, \_Component13);

/\*\*

\* Construct Pushpin instance

\* @constructor

\* @param {Element} el

\* @param {Object} options

\*/

function Pushpin(el, options) {

\_classCallCheck(this, Pushpin);

var \_this47 = \_possibleConstructorReturn(this, (Pushpin.\_\_proto\_\_ || Object.getPrototypeOf(Pushpin)).call(this, Pushpin, el, options));

\_this47.el.M\_Pushpin = \_this47;

/\*\*

\* Options for the modal

\* @member Pushpin#options

\*/

\_this47.options = $.extend({}, Pushpin.defaults, options);

\_this47.originalOffset = \_this47.el.offsetTop;

Pushpin.\_pushpins.push(\_this47);

\_this47.\_setupEventHandlers();

\_this47.\_updatePosition();

return \_this47;

}

\_createClass(Pushpin, [{

key: "destroy",

/\*\*

\* Teardown component

\*/

value: function destroy() {

this.el.style.top = null;

this.\_removePinClasses();

this.\_removeEventHandlers();

// Remove pushpin Inst

var index = Pushpin.\_pushpins.indexOf(this);

Pushpin.\_pushpins.splice(index, 1);

}

}, {

key: "\_setupEventHandlers",

value: function \_setupEventHandlers() {

document.addEventListener('scroll', Pushpin.\_updateElements);

}

}, {

key: "\_removeEventHandlers",

value: function \_removeEventHandlers() {

document.removeEventListener('scroll', Pushpin.\_updateElements);

}

}, {

key: "\_updatePosition",

value: function \_updatePosition() {

var scrolled = M.getDocumentScrollTop() + this.options.offset;

if (this.options.top <= scrolled && this.options.bottom >= scrolled && !this.el.classList.contains('pinned')) {

this.\_removePinClasses();

this.el.style.top = this.options.offset + "px";

this.el.classList.add('pinned');

// onPositionChange callback

if (typeof this.options.onPositionChange === 'function') {

this.options.onPositionChange.call(this, 'pinned');

}

}

// Add pin-top (when scrolled position is above top)

if (scrolled < this.options.top && !this.el.classList.contains('pin-top')) {

this.\_removePinClasses();

this.el.style.top = 0;

this.el.classList.add('pin-top');

// onPositionChange callback

if (typeof this.options.onPositionChange === 'function') {

this.options.onPositionChange.call(this, 'pin-top');

}

}

// Add pin-bottom (when scrolled position is below bottom)

if (scrolled > this.options.bottom && !this.el.classList.contains('pin-bottom')) {

this.\_removePinClasses();

this.el.classList.add('pin-bottom');

this.el.style.top = this.options.bottom - this.originalOffset + "px";

// onPositionChange callback

if (typeof this.options.onPositionChange === 'function') {

this.options.onPositionChange.call(this, 'pin-bottom');

}

}

}

}, {

key: "\_removePinClasses",

value: function \_removePinClasses() {

// IE 11 bug (can't remove multiple classes in one line)

this.el.classList.remove('pin-top');

this.el.classList.remove('pinned');

this.el.classList.remove('pin-bottom');

}

}], [{

key: "init",

value: function init(els, options) {

return \_get(Pushpin.\_\_proto\_\_ || Object.getPrototypeOf(Pushpin), "init", this).call(this, this, els, options);

}

/\*\*

\* Get Instance

\*/

}, {

key: "getInstance",

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_Pushpin;

}

}, {

key: "\_updateElements",

value: function \_updateElements() {

for (var elIndex in Pushpin.\_pushpins) {

var pInstance = Pushpin.\_pushpins[elIndex];

pInstance.\_updatePosition();

}

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return Pushpin;

}(Component);

/\*\*

\* @static

\* @memberof Pushpin

\*/

Pushpin.\_pushpins = [];

M.Pushpin = Pushpin;

if (M.jQueryLoaded) {

M.initializeJqueryWrapper(Pushpin, 'pushpin', 'M\_Pushpin');

}

})(cash);

;(function ($, anim) {

'use strict';

var \_defaults = {

direction: 'top',

hoverEnabled: true,

toolbarEnabled: false

};

$.fn.reverse = [].reverse;

/\*\*

\* @class

\*

\*/

var FloatingActionButton = function (\_Component14) {

\_inherits(FloatingActionButton, \_Component14);

/\*\*

\* Construct FloatingActionButton instance

\* @constructor

\* @param {Element} el

\* @param {Object} options

\*/

function FloatingActionButton(el, options) {

\_classCallCheck(this, FloatingActionButton);

var \_this48 = \_possibleConstructorReturn(this, (FloatingActionButton.\_\_proto\_\_ || Object.getPrototypeOf(FloatingActionButton)).call(this, FloatingActionButton, el, options));

\_this48.el.M\_FloatingActionButton = \_this48;

/\*\*

\* Options for the fab

\* @member FloatingActionButton#options

\* @prop {Boolean} [direction] - Direction fab menu opens

\* @prop {Boolean} [hoverEnabled=true] - Enable hover vs click

\* @prop {Boolean} [toolbarEnabled=false] - Enable toolbar transition

\*/

\_this48.options = $.extend({}, FloatingActionButton.defaults, options);

\_this48.isOpen = false;

\_this48.$anchor = \_this48.$el.children('a').first();

\_this48.$menu = \_this48.$el.children('ul').first();

\_this48.$floatingBtns = \_this48.$el.find('ul .btn-floating');

\_this48.$floatingBtnsReverse = \_this48.$el.find('ul .btn-floating').reverse();

\_this48.offsetY = 0;

\_this48.offsetX = 0;

\_this48.$el.addClass("direction-" + \_this48.options.direction);

if (\_this48.options.direction === 'top') {

\_this48.offsetY = 40;

} else if (\_this48.options.direction === 'right') {

\_this48.offsetX = -40;

} else if (\_this48.options.direction === 'bottom') {

\_this48.offsetY = -40;

} else {

\_this48.offsetX = 40;

}

\_this48.\_setupEventHandlers();

return \_this48;

}

\_createClass(FloatingActionButton, [{

key: "destroy",

/\*\*

\* Teardown component

\*/

value: function destroy() {

this.\_removeEventHandlers();

this.el.M\_FloatingActionButton = undefined;

}

/\*\*

\* Setup Event Handlers

\*/

}, {

key: "\_setupEventHandlers",

value: function \_setupEventHandlers() {

this.\_handleFABClickBound = this.\_handleFABClick.bind(this);

this.\_handleOpenBound = this.open.bind(this);

this.\_handleCloseBound = this.close.bind(this);

if (this.options.hoverEnabled && !this.options.toolbarEnabled) {

this.el.addEventListener('mouseenter', this.\_handleOpenBound);

this.el.addEventListener('mouseleave', this.\_handleCloseBound);

} else {

this.el.addEventListener('click', this.\_handleFABClickBound);

}

}

/\*\*

\* Remove Event Handlers

\*/

}, {

key: "\_removeEventHandlers",

value: function \_removeEventHandlers() {

if (this.options.hoverEnabled && !this.options.toolbarEnabled) {

this.el.removeEventListener('mouseenter', this.\_handleOpenBound);

this.el.removeEventListener('mouseleave', this.\_handleCloseBound);

} else {

this.el.removeEventListener('click', this.\_handleFABClickBound);

}

}

/\*\*

\* Handle FAB Click

\*/

}, {

key: "\_handleFABClick",

value: function \_handleFABClick() {

if (this.isOpen) {

this.close();

} else {

this.open();

}

}

/\*\*

\* Handle Document Click

\* @param {Event} e

\*/

}, {

key: "\_handleDocumentClick",

value: function \_handleDocumentClick(e) {

if (!$(e.target).closest(this.$menu).length) {

this.close();

}

}

/\*\*

\* Open FAB

\*/

}, {

key: "open",

value: function open() {

if (this.isOpen) {

return;

}

if (this.options.toolbarEnabled) {

this.\_animateInToolbar();

} else {

this.\_animateInFAB();

}

this.isOpen = true;

}

/\*\*

\* Close FAB

\*/

}, {

key: "close",

value: function close() {

if (!this.isOpen) {

return;

}

if (this.options.toolbarEnabled) {

window.removeEventListener('scroll', this.\_handleCloseBound, true);

document.body.removeEventListener('click', this.\_handleDocumentClickBound, true);

this.\_animateOutToolbar();

} else {

this.\_animateOutFAB();

}

this.isOpen = false;

}

/\*\*

\* Classic FAB Menu open

\*/

}, {

key: "\_animateInFAB",

value: function \_animateInFAB() {

var \_this49 = this;

this.$el.addClass('active');

var time = 0;

this.$floatingBtnsReverse.each(function (el) {

anim({

targets: el,

opacity: 1,

scale: [0.4, 1],

translateY: [\_this49.offsetY, 0],

translateX: [\_this49.offsetX, 0],

duration: 275,

delay: time,

easing: 'easeInOutQuad'

});

time += 40;

});

}

/\*\*

\* Classic FAB Menu close

\*/

}, {

key: "\_animateOutFAB",

value: function \_animateOutFAB() {

var \_this50 = this;

this.$floatingBtnsReverse.each(function (el) {

anim.remove(el);

anim({

targets: el,

opacity: 0,

scale: 0.4,

translateY: \_this50.offsetY,

translateX: \_this50.offsetX,

duration: 175,

easing: 'easeOutQuad',

complete: function () {

\_this50.$el.removeClass('active');

}

});

});

}

/\*\*

\* Toolbar transition Menu open

\*/

}, {

key: "\_animateInToolbar",

value: function \_animateInToolbar() {

var \_this51 = this;

var scaleFactor = void 0;

var windowWidth = window.innerWidth;

var windowHeight = window.innerHeight;

var btnRect = this.el.getBoundingClientRect();

var backdrop = $('<div class="fab-backdrop"></div>');

var fabColor = this.$anchor.css('background-color');

this.$anchor.append(backdrop);

this.offsetX = btnRect.left - windowWidth / 2 + btnRect.width / 2;

this.offsetY = windowHeight - btnRect.bottom;

scaleFactor = windowWidth / backdrop[0].clientWidth;

this.btnBottom = btnRect.bottom;

this.btnLeft = btnRect.left;

this.btnWidth = btnRect.width;

// Set initial state

this.$el.addClass('active');

this.$el.css({

'text-align': 'center',

width: '100%',

bottom: 0,

left: 0,

transform: 'translateX(' + this.offsetX + 'px)',

transition: 'none'

});

this.$anchor.css({

transform: 'translateY(' + -this.offsetY + 'px)',

transition: 'none'

});

backdrop.css({

'background-color': fabColor

});

setTimeout(function () {

\_this51.$el.css({

transform: '',

transition: 'transform .2s cubic-bezier(0.550, 0.085, 0.680, 0.530), background-color 0s linear .2s'

});

\_this51.$anchor.css({

overflow: 'visible',

transform: '',

transition: 'transform .2s'

});

setTimeout(function () {

\_this51.$el.css({

overflow: 'hidden',

'background-color': fabColor

});

backdrop.css({

transform: 'scale(' + scaleFactor + ')',

transition: 'transform .2s cubic-bezier(0.550, 0.055, 0.675, 0.190)'

});

\_this51.$menu.children('li').children('a').css({

opacity: 1

});

// Scroll to close.

\_this51.\_handleDocumentClickBound = \_this51.\_handleDocumentClick.bind(\_this51);

window.addEventListener('scroll', \_this51.\_handleCloseBound, true);

document.body.addEventListener('click', \_this51.\_handleDocumentClickBound, true);

}, 100);

}, 0);

}

/\*\*

\* Toolbar transition Menu close

\*/

}, {

key: "\_animateOutToolbar",

value: function \_animateOutToolbar() {

var \_this52 = this;

var windowWidth = window.innerWidth;

var windowHeight = window.innerHeight;

var backdrop = this.$el.find('.fab-backdrop');

var fabColor = this.$anchor.css('background-color');

this.offsetX = this.btnLeft - windowWidth / 2 + this.btnWidth / 2;

this.offsetY = windowHeight - this.btnBottom;

// Hide backdrop

this.$el.removeClass('active');

this.$el.css({

'background-color': 'transparent',

transition: 'none'

});

this.$anchor.css({

transition: 'none'

});

backdrop.css({

transform: 'scale(0)',

'background-color': fabColor

});

this.$menu.children('li').children('a').css({

opacity: ''

});

setTimeout(function () {

backdrop.remove();

// Set initial state.

\_this52.$el.css({

'text-align': '',

width: '',

bottom: '',

left: '',

overflow: '',

'background-color': '',

transform: 'translate3d(' + -\_this52.offsetX + 'px,0,0)'

});

\_this52.$anchor.css({

overflow: '',

transform: 'translate3d(0,' + \_this52.offsetY + 'px,0)'

});

setTimeout(function () {

\_this52.$el.css({

transform: 'translate3d(0,0,0)',

transition: 'transform .2s'

});

\_this52.$anchor.css({

transform: 'translate3d(0,0,0)',

transition: 'transform .2s cubic-bezier(0.550, 0.055, 0.675, 0.190)'

});

}, 20);

}, 200);

}

}], [{

key: "init",

value: function init(els, options) {

return \_get(FloatingActionButton.\_\_proto\_\_ || Object.getPrototypeOf(FloatingActionButton), "init", this).call(this, this, els, options);

}

/\*\*

\* Get Instance

\*/

}, {

key: "getInstance",

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_FloatingActionButton;

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return FloatingActionButton;

}(Component);

M.FloatingActionButton = FloatingActionButton;

if (M.jQueryLoaded) {

M.initializeJqueryWrapper(FloatingActionButton, 'floatingActionButton', 'M\_FloatingActionButton');

}

})(cash, M.anime);

;(function ($) {

'use strict';

var \_defaults = {

// Close when date is selected

autoClose: false,

// the default output format for the input field value

format: 'mmm dd, yyyy',

// Used to create date object from current input string

parse: null,

// The initial date to view when first opened

defaultDate: null,

// Make the `defaultDate` the initial selected value

setDefaultDate: false,

disableWeekends: false,

disableDayFn: null,

// First day of week (0: Sunday, 1: Monday etc)

firstDay: 0,

// The earliest date that can be selected

minDate: null,

// Thelatest date that can be selected

maxDate: null,

// Number of years either side, or array of upper/lower range

yearRange: 10,

// used internally (don't config outside)

minYear: 0,

maxYear: 9999,

minMonth: undefined,

maxMonth: undefined,

startRange: null,

endRange: null,

isRTL: false,

// Render the month after year in the calendar title

showMonthAfterYear: false,

// Render days of the calendar grid that fall in the next or previous month

showDaysInNextAndPreviousMonths: false,

// Specify a DOM element to render the calendar in

container: null,

// Show clear button

showClearBtn: false,

// internationalization

i18n: {

cancel: 'Cancel',

clear: 'Clear',

done: 'Ok',

previousMonth: 'â€¹',

nextMonth: 'â€º',

months: ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December'],

monthsShort: ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'],

weekdays: ['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday'],

weekdaysShort: ['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat'],

weekdaysAbbrev: ['S', 'M', 'T', 'W', 'T', 'F', 'S']

},

// events array

events: [],

// callback function

onSelect: null,

onOpen: null,

onClose: null,

onDraw: null

};

/\*\*

\* @class

\*

\*/

var Datepicker = function (\_Component15) {

\_inherits(Datepicker, \_Component15);

/\*\*

\* Construct Datepicker instance and set up overlay

\* @constructor

\* @param {Element} el

\* @param {Object} options

\*/

function Datepicker(el, options) {

\_classCallCheck(this, Datepicker);

var \_this53 = \_possibleConstructorReturn(this, (Datepicker.\_\_proto\_\_ || Object.getPrototypeOf(Datepicker)).call(this, Datepicker, el, options));

\_this53.el.M\_Datepicker = \_this53;

\_this53.options = $.extend({}, Datepicker.defaults, options);

// make sure i18n defaults are not lost when only few i18n option properties are passed

if (!!options && options.hasOwnProperty('i18n') && typeof options.i18n === 'object') {

\_this53.options.i18n = $.extend({}, Datepicker.defaults.i18n, options.i18n);

}

// Remove time component from minDate and maxDate options

if (\_this53.options.minDate) \_this53.options.minDate.setHours(0, 0, 0, 0);

if (\_this53.options.maxDate) \_this53.options.maxDate.setHours(0, 0, 0, 0);

\_this53.id = M.guid();

\_this53.\_setupVariables();

\_this53.\_insertHTMLIntoDOM();

\_this53.\_setupModal();

\_this53.\_setupEventHandlers();

if (!\_this53.options.defaultDate) {

\_this53.options.defaultDate = new Date(Date.parse(\_this53.el.value));

}

var defDate = \_this53.options.defaultDate;

if (Datepicker.\_isDate(defDate)) {

if (\_this53.options.setDefaultDate) {

\_this53.setDate(defDate, true);

\_this53.setInputValue();

} else {

\_this53.gotoDate(defDate);

}

} else {

\_this53.gotoDate(new Date());

}

/\*\*

\* Describes open/close state of datepicker

\* @type {Boolean}

\*/

\_this53.isOpen = false;

return \_this53;

}

\_createClass(Datepicker, [{

key: "destroy",

/\*\*

\* Teardown component

\*/

value: function destroy() {

this.\_removeEventHandlers();

this.modal.destroy();

$(this.modalEl).remove();

this.destroySelects();

this.el.M\_Datepicker = undefined;

}

}, {

key: "destroySelects",

value: function destroySelects() {

var oldYearSelect = this.calendarEl.querySelector('.orig-select-year');

if (oldYearSelect) {

M.FormSelect.getInstance(oldYearSelect).destroy();

}

var oldMonthSelect = this.calendarEl.querySelector('.orig-select-month');

if (oldMonthSelect) {

M.FormSelect.getInstance(oldMonthSelect).destroy();

}

}

}, {

key: "\_insertHTMLIntoDOM",

value: function \_insertHTMLIntoDOM() {

if (this.options.showClearBtn) {

$(this.clearBtn).css({ visibility: '' });

this.clearBtn.innerHTML = this.options.i18n.clear;

}

this.doneBtn.innerHTML = this.options.i18n.done;

this.cancelBtn.innerHTML = this.options.i18n.cancel;

if (this.options.container) {

this.$modalEl.appendTo(this.options.container);

} else {

this.$modalEl.insertBefore(this.el);

}

}

}, {

key: "\_setupModal",

value: function \_setupModal() {

var \_this54 = this;

this.modalEl.id = 'modal-' + this.id;

this.modal = M.Modal.init(this.modalEl, {

onCloseEnd: function () {

\_this54.isOpen = false;

}

});

}

}, {

key: "toString",

value: function toString(format) {

var \_this55 = this;

format = format || this.options.format;

if (!Datepicker.\_isDate(this.date)) {

return '';

}

var formatArray = format.split(/(d{1,4}|m{1,4}|y{4}|yy|!.)/g);

var formattedDate = formatArray.map(function (label) {

if (\_this55.formats[label]) {

return \_this55.formats[label]();

}

return label;

}).join('');

return formattedDate;

}

}, {

key: "setDate",

value: function setDate(date, preventOnSelect) {

if (!date) {

this.date = null;

this.\_renderDateDisplay();

return this.draw();

}

if (typeof date === 'string') {

date = new Date(Date.parse(date));

}

if (!Datepicker.\_isDate(date)) {

return;

}

var min = this.options.minDate,

max = this.options.maxDate;

if (Datepicker.\_isDate(min) && date < min) {

date = min;

} else if (Datepicker.\_isDate(max) && date > max) {

date = max;

}

this.date = new Date(date.getTime());

this.\_renderDateDisplay();

Datepicker.\_setToStartOfDay(this.date);

this.gotoDate(this.date);

if (!preventOnSelect && typeof this.options.onSelect === 'function') {

this.options.onSelect.call(this, this.date);

}

}

}, {

key: "setInputValue",

value: function setInputValue() {

this.el.value = this.toString();

this.$el.trigger('change', { firedBy: this });

}

}, {

key: "\_renderDateDisplay",

value: function \_renderDateDisplay() {

var displayDate = Datepicker.\_isDate(this.date) ? this.date : new Date();

var i18n = this.options.i18n;

var day = i18n.weekdaysShort[displayDate.getDay()];

var month = i18n.monthsShort[displayDate.getMonth()];

var date = displayDate.getDate();

this.yearTextEl.innerHTML = displayDate.getFullYear();

this.dateTextEl.innerHTML = day + ", " + month + " " + date;

}

/\*\*

\* change view to a specific date

\*/

}, {

key: "gotoDate",

value: function gotoDate(date) {

var newCalendar = true;

if (!Datepicker.\_isDate(date)) {

return;

}

if (this.calendars) {

var firstVisibleDate = new Date(this.calendars[0].year, this.calendars[0].month, 1),

lastVisibleDate = new Date(this.calendars[this.calendars.length - 1].year, this.calendars[this.calendars.length - 1].month, 1),

visibleDate = date.getTime();

// get the end of the month

lastVisibleDate.setMonth(lastVisibleDate.getMonth() + 1);

lastVisibleDate.setDate(lastVisibleDate.getDate() - 1);

newCalendar = visibleDate < firstVisibleDate.getTime() || lastVisibleDate.getTime() < visibleDate;

}

if (newCalendar) {

this.calendars = [{

month: date.getMonth(),

year: date.getFullYear()

}];

}

this.adjustCalendars();

}

}, {

key: "adjustCalendars",

value: function adjustCalendars() {

this.calendars[0] = this.adjustCalendar(this.calendars[0]);

this.draw();

}

}, {

key: "adjustCalendar",

value: function adjustCalendar(calendar) {

if (calendar.month < 0) {

calendar.year -= Math.ceil(Math.abs(calendar.month) / 12);

calendar.month += 12;

}

if (calendar.month > 11) {

calendar.year += Math.floor(Math.abs(calendar.month) / 12);

calendar.month -= 12;

}

return calendar;

}

}, {

key: "nextMonth",

value: function nextMonth() {

this.calendars[0].month++;

this.adjustCalendars();

}

}, {

key: "prevMonth",

value: function prevMonth() {

this.calendars[0].month--;

this.adjustCalendars();

}

}, {

key: "render",

value: function render(year, month, randId) {

var opts = this.options,

now = new Date(),

days = Datepicker.\_getDaysInMonth(year, month),

before = new Date(year, month, 1).getDay(),

data = [],

row = [];

Datepicker.\_setToStartOfDay(now);

if (opts.firstDay > 0) {

before -= opts.firstDay;

if (before < 0) {

before += 7;

}

}

var previousMonth = month === 0 ? 11 : month - 1,

nextMonth = month === 11 ? 0 : month + 1,

yearOfPreviousMonth = month === 0 ? year - 1 : year,

yearOfNextMonth = month === 11 ? year + 1 : year,

daysInPreviousMonth = Datepicker.\_getDaysInMonth(yearOfPreviousMonth, previousMonth);

var cells = days + before,

after = cells;

while (after > 7) {

after -= 7;

}

cells += 7 - after;

var isWeekSelected = false;

for (var i = 0, r = 0; i < cells; i++) {

var day = new Date(year, month, 1 + (i - before)),

isSelected = Datepicker.\_isDate(this.date) ? Datepicker.\_compareDates(day, this.date) : false,

isToday = Datepicker.\_compareDates(day, now),

hasEvent = opts.events.indexOf(day.toDateString()) !== -1 ? true : false,

isEmpty = i < before || i >= days + before,

dayNumber = 1 + (i - before),

monthNumber = month,

yearNumber = year,

isStartRange = opts.startRange && Datepicker.\_compareDates(opts.startRange, day),

isEndRange = opts.endRange && Datepicker.\_compareDates(opts.endRange, day),

isInRange = opts.startRange && opts.endRange && opts.startRange < day && day < opts.endRange,

isDisabled = opts.minDate && day < opts.minDate || opts.maxDate && day > opts.maxDate || opts.disableWeekends && Datepicker.\_isWeekend(day) || opts.disableDayFn && opts.disableDayFn(day);

if (isEmpty) {

if (i < before) {

dayNumber = daysInPreviousMonth + dayNumber;

monthNumber = previousMonth;

yearNumber = yearOfPreviousMonth;

} else {

dayNumber = dayNumber - days;

monthNumber = nextMonth;

yearNumber = yearOfNextMonth;

}

}

var dayConfig = {

day: dayNumber,

month: monthNumber,

year: yearNumber,

hasEvent: hasEvent,

isSelected: isSelected,

isToday: isToday,

isDisabled: isDisabled,

isEmpty: isEmpty,

isStartRange: isStartRange,

isEndRange: isEndRange,

isInRange: isInRange,

showDaysInNextAndPreviousMonths: opts.showDaysInNextAndPreviousMonths

};

row.push(this.renderDay(dayConfig));

if (++r === 7) {

data.push(this.renderRow(row, opts.isRTL, isWeekSelected));

row = [];

r = 0;

isWeekSelected = false;

}

}

return this.renderTable(opts, data, randId);

}

}, {

key: "renderDay",

value: function renderDay(opts) {

var arr = [];

var ariaSelected = 'false';

if (opts.isEmpty) {

if (opts.showDaysInNextAndPreviousMonths) {

arr.push('is-outside-current-month');

arr.push('is-selection-disabled');

} else {

return '<td class="is-empty"></td>';

}

}

if (opts.isDisabled) {

arr.push('is-disabled');

}

if (opts.isToday) {

arr.push('is-today');

}

if (opts.isSelected) {

arr.push('is-selected');

ariaSelected = 'true';

}

if (opts.hasEvent) {

arr.push('has-event');

}

if (opts.isInRange) {

arr.push('is-inrange');

}

if (opts.isStartRange) {

arr.push('is-startrange');

}

if (opts.isEndRange) {

arr.push('is-endrange');

}

return "<td data-day=\"" + opts.day + "\" class=\"" + arr.join(' ') + "\" aria-selected=\"" + ariaSelected + "\">" + ("<button class=\"datepicker-day-button\" type=\"button\" data-year=\"" + opts.year + "\" data-month=\"" + opts.month + "\" data-day=\"" + opts.day + "\">" + opts.day + "</button>") + '</td>';

}

}, {

key: "renderRow",

value: function renderRow(days, isRTL, isRowSelected) {

return '<tr class="datepicker-row' + (isRowSelected ? ' is-selected' : '') + '">' + (isRTL ? days.reverse() : days).join('') + '</tr>';

}

}, {

key: "renderTable",

value: function renderTable(opts, data, randId) {

return '<div class="datepicker-table-wrapper"><table cellpadding="0" cellspacing="0" class="datepicker-table" role="grid" aria-labelledby="' + randId + '">' + this.renderHead(opts) + this.renderBody(data) + '</table></div>';

}

}, {

key: "renderHead",

value: function renderHead(opts) {

var i = void 0,

arr = [];

for (i = 0; i < 7; i++) {

arr.push("<th scope=\"col\"><abbr title=\"" + this.renderDayName(opts, i) + "\">" + this.renderDayName(opts, i, true) + "</abbr></th>");

}

return '<thead><tr>' + (opts.isRTL ? arr.reverse() : arr).join('') + '</tr></thead>';

}

}, {

key: "renderBody",

value: function renderBody(rows) {

return '<tbody>' + rows.join('') + '</tbody>';

}

}, {

key: "renderTitle",

value: function renderTitle(instance, c, year, month, refYear, randId) {

var i = void 0,

j = void 0,

arr = void 0,

opts = this.options,

isMinYear = year === opts.minYear,

isMaxYear = year === opts.maxYear,

html = '<div id="' + randId + '" class="datepicker-controls" role="heading" aria-live="assertive">',

monthHtml = void 0,

yearHtml = void 0,

prev = true,

next = true;

for (arr = [], i = 0; i < 12; i++) {

arr.push('<option value="' + (year === refYear ? i - c : 12 + i - c) + '"' + (i === month ? ' selected="selected"' : '') + (isMinYear && i < opts.minMonth || isMaxYear && i > opts.maxMonth ? 'disabled="disabled"' : '') + '>' + opts.i18n.months[i] + '</option>');

}

monthHtml = '<select class="datepicker-select orig-select-month" tabindex="-1">' + arr.join('') + '</select>';

if ($.isArray(opts.yearRange)) {

i = opts.yearRange[0];

j = opts.yearRange[1] + 1;

} else {

i = year - opts.yearRange;

j = 1 + year + opts.yearRange;

}

for (arr = []; i < j && i <= opts.maxYear; i++) {

if (i >= opts.minYear) {

arr.push("<option value=\"" + i + "\" " + (i === year ? 'selected="selected"' : '') + ">" + i + "</option>");

}

}

yearHtml = "<select class=\"datepicker-select orig-select-year\" tabindex=\"-1\">" + arr.join('') + "</select>";

var leftArrow = '<svg fill="#000000" height="24" viewBox="0 0 24 24" width="24" xmlns="http://www.w3.org/2000/svg"><path d="M15.41 16.09l-4.58-4.59 4.58-4.59L14 5.5l-6 6 6 6z"/><path d="M0-.5h24v24H0z" fill="none"/></svg>';

html += "<button class=\"month-prev" + (prev ? '' : ' is-disabled') + "\" type=\"button\">" + leftArrow + "</button>";

html += '<div class="selects-container">';

if (opts.showMonthAfterYear) {

html += yearHtml + monthHtml;

} else {

html += monthHtml + yearHtml;

}

html += '</div>';

if (isMinYear && (month === 0 || opts.minMonth >= month)) {

prev = false;

}

if (isMaxYear && (month === 11 || opts.maxMonth <= month)) {

next = false;

}

var rightArrow = '<svg fill="#000000" height="24" viewBox="0 0 24 24" width="24" xmlns="http://www.w3.org/2000/svg"><path d="M8.59 16.34l4.58-4.59-4.58-4.59L10 5.75l6 6-6 6z"/><path d="M0-.25h24v24H0z" fill="none"/></svg>';

html += "<button class=\"month-next" + (next ? '' : ' is-disabled') + "\" type=\"button\">" + rightArrow + "</button>";

return html += '</div>';

}

/\*\*

\* refresh the HTML

\*/

}, {

key: "draw",

value: function draw(force) {

if (!this.isOpen && !force) {

return;

}

var opts = this.options,

minYear = opts.minYear,

maxYear = opts.maxYear,

minMonth = opts.minMonth,

maxMonth = opts.maxMonth,

html = '',

randId = void 0;

if (this.\_y <= minYear) {

this.\_y = minYear;

if (!isNaN(minMonth) && this.\_m < minMonth) {

this.\_m = minMonth;

}

}

if (this.\_y >= maxYear) {

this.\_y = maxYear;

if (!isNaN(maxMonth) && this.\_m > maxMonth) {

this.\_m = maxMonth;

}

}

randId = 'datepicker-title-' + Math.random().toString(36).replace(/[^a-z]+/g, '').substr(0, 2);

for (var c = 0; c < 1; c++) {

this.\_renderDateDisplay();

html += this.renderTitle(this, c, this.calendars[c].year, this.calendars[c].month, this.calendars[0].year, randId) + this.render(this.calendars[c].year, this.calendars[c].month, randId);

}

this.destroySelects();

this.calendarEl.innerHTML = html;

// Init Materialize Select

var yearSelect = this.calendarEl.querySelector('.orig-select-year');

var monthSelect = this.calendarEl.querySelector('.orig-select-month');

M.FormSelect.init(yearSelect, {

classes: 'select-year',

dropdownOptions: { container: document.body, constrainWidth: false }

});

M.FormSelect.init(monthSelect, {

classes: 'select-month',

dropdownOptions: { container: document.body, constrainWidth: false }

});

// Add change handlers for select

yearSelect.addEventListener('change', this.\_handleYearChange.bind(this));

monthSelect.addEventListener('change', this.\_handleMonthChange.bind(this));

if (typeof this.options.onDraw === 'function') {

this.options.onDraw(this);

}

}

/\*\*

\* Setup Event Handlers

\*/

}, {

key: "\_setupEventHandlers",

value: function \_setupEventHandlers() {

this.\_handleInputKeydownBound = this.\_handleInputKeydown.bind(this);

this.\_handleInputClickBound = this.\_handleInputClick.bind(this);

this.\_handleInputChangeBound = this.\_handleInputChange.bind(this);

this.\_handleCalendarClickBound = this.\_handleCalendarClick.bind(this);

this.\_finishSelectionBound = this.\_finishSelection.bind(this);

this.\_handleMonthChange = this.\_handleMonthChange.bind(this);

this.\_closeBound = this.close.bind(this);

this.el.addEventListener('click', this.\_handleInputClickBound);

this.el.addEventListener('keydown', this.\_handleInputKeydownBound);

this.el.addEventListener('change', this.\_handleInputChangeBound);

this.calendarEl.addEventListener('click', this.\_handleCalendarClickBound);

this.doneBtn.addEventListener('click', this.\_finishSelectionBound);

this.cancelBtn.addEventListener('click', this.\_closeBound);

if (this.options.showClearBtn) {

this.\_handleClearClickBound = this.\_handleClearClick.bind(this);

this.clearBtn.addEventListener('click', this.\_handleClearClickBound);

}

}

}, {

key: "\_setupVariables",

value: function \_setupVariables() {

var \_this56 = this;

this.$modalEl = $(Datepicker.\_template);

this.modalEl = this.$modalEl[0];

this.calendarEl = this.modalEl.querySelector('.datepicker-calendar');

this.yearTextEl = this.modalEl.querySelector('.year-text');

this.dateTextEl = this.modalEl.querySelector('.date-text');

if (this.options.showClearBtn) {

this.clearBtn = this.modalEl.querySelector('.datepicker-clear');

}

this.doneBtn = this.modalEl.querySelector('.datepicker-done');

this.cancelBtn = this.modalEl.querySelector('.datepicker-cancel');

this.formats = {

d: function () {

return \_this56.date.getDate();

},

dd: function () {

var d = \_this56.date.getDate();

return (d < 10 ? '0' : '') + d;

},

ddd: function () {

return \_this56.options.i18n.weekdaysShort[\_this56.date.getDay()];

},

dddd: function () {

return \_this56.options.i18n.weekdays[\_this56.date.getDay()];

},

m: function () {

return \_this56.date.getMonth() + 1;

},

mm: function () {

var m = \_this56.date.getMonth() + 1;

return (m < 10 ? '0' : '') + m;

},

mmm: function () {

return \_this56.options.i18n.monthsShort[\_this56.date.getMonth()];

},

mmmm: function () {

return \_this56.options.i18n.months[\_this56.date.getMonth()];

},

yy: function () {

return ('' + \_this56.date.getFullYear()).slice(2);

},

yyyy: function () {

return \_this56.date.getFullYear();

}

};

}

/\*\*

\* Remove Event Handlers

\*/

}, {

key: "\_removeEventHandlers",

value: function \_removeEventHandlers() {

this.el.removeEventListener('click', this.\_handleInputClickBound);

this.el.removeEventListener('keydown', this.\_handleInputKeydownBound);

this.el.removeEventListener('change', this.\_handleInputChangeBound);

this.calendarEl.removeEventListener('click', this.\_handleCalendarClickBound);

}

}, {

key: "\_handleInputClick",

value: function \_handleInputClick() {

this.open();

}

}, {

key: "\_handleInputKeydown",

value: function \_handleInputKeydown(e) {

if (e.which === M.keys.ENTER) {

e.preventDefault();

this.open();

}

}

}, {

key: "\_handleCalendarClick",

value: function \_handleCalendarClick(e) {

if (!this.isOpen) {

return;

}

var $target = $(e.target);

if (!$target.hasClass('is-disabled')) {

if ($target.hasClass('datepicker-day-button') && !$target.hasClass('is-empty') && !$target.parent().hasClass('is-disabled')) {

this.setDate(new Date(e.target.getAttribute('data-year'), e.target.getAttribute('data-month'), e.target.getAttribute('data-day')));

if (this.options.autoClose) {

this.\_finishSelection();

}

} else if ($target.closest('.month-prev').length) {

this.prevMonth();

} else if ($target.closest('.month-next').length) {

this.nextMonth();

}

}

}

}, {

key: "\_handleClearClick",

value: function \_handleClearClick() {

this.date = null;

this.setInputValue();

this.close();

}

}, {

key: "\_handleMonthChange",

value: function \_handleMonthChange(e) {

this.gotoMonth(e.target.value);

}

}, {

key: "\_handleYearChange",

value: function \_handleYearChange(e) {

this.gotoYear(e.target.value);

}

/\*\*

\* change view to a specific month (zero-index, e.g. 0: January)

\*/

}, {

key: "gotoMonth",

value: function gotoMonth(month) {

if (!isNaN(month)) {

this.calendars[0].month = parseInt(month, 10);

this.adjustCalendars();

}

}

/\*\*

\* change view to a specific full year (e.g. "2012")

\*/

}, {

key: "gotoYear",

value: function gotoYear(year) {

if (!isNaN(year)) {

this.calendars[0].year = parseInt(year, 10);

this.adjustCalendars();

}

}

}, {

key: "\_handleInputChange",

value: function \_handleInputChange(e) {

var date = void 0;

// Prevent change event from being fired when triggered by the plugin

if (e.firedBy === this) {

return;

}

if (this.options.parse) {

date = this.options.parse(this.el.value, this.options.format);

} else {

date = new Date(Date.parse(this.el.value));

}

if (Datepicker.\_isDate(date)) {

this.setDate(date);

}

}

}, {

key: "renderDayName",

value: function renderDayName(opts, day, abbr) {

day += opts.firstDay;

while (day >= 7) {

day -= 7;

}

return abbr ? opts.i18n.weekdaysAbbrev[day] : opts.i18n.weekdays[day];

}

/\*\*

\* Set input value to the selected date and close Datepicker

\*/

}, {

key: "\_finishSelection",

value: function \_finishSelection() {

this.setInputValue();

this.close();

}

/\*\*

\* Open Datepicker

\*/

}, {

key: "open",

value: function open() {

if (this.isOpen) {

return;

}

this.isOpen = true;

if (typeof this.options.onOpen === 'function') {

this.options.onOpen.call(this);

}

this.draw();

this.modal.open();

return this;

}

/\*\*

\* Close Datepicker

\*/

}, {

key: "close",

value: function close() {

if (!this.isOpen) {

return;

}

this.isOpen = false;

if (typeof this.options.onClose === 'function') {

this.options.onClose.call(this);

}

this.modal.close();

return this;

}

}], [{

key: "init",

value: function init(els, options) {

return \_get(Datepicker.\_\_proto\_\_ || Object.getPrototypeOf(Datepicker), "init", this).call(this, this, els, options);

}

}, {

key: "\_isDate",

value: function \_isDate(obj) {

return (/Date/.test(Object.prototype.toString.call(obj)) && !isNaN(obj.getTime())

);

}

}, {

key: "\_isWeekend",

value: function \_isWeekend(date) {

var day = date.getDay();

return day === 0 || day === 6;

}

}, {

key: "\_setToStartOfDay",

value: function \_setToStartOfDay(date) {

if (Datepicker.\_isDate(date)) date.setHours(0, 0, 0, 0);

}

}, {

key: "\_getDaysInMonth",

value: function \_getDaysInMonth(year, month) {

return [31, Datepicker.\_isLeapYear(year) ? 29 : 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31][month];

}

}, {

key: "\_isLeapYear",

value: function \_isLeapYear(year) {

// solution by Matti Virkkunen: http://stackoverflow.com/a/4881951

return year % 4 === 0 && year % 100 !== 0 || year % 400 === 0;

}

}, {

key: "\_compareDates",

value: function \_compareDates(a, b) {

// weak date comparison (use setToStartOfDay(date) to ensure correct result)

return a.getTime() === b.getTime();

}

}, {

key: "\_setToStartOfDay",

value: function \_setToStartOfDay(date) {

if (Datepicker.\_isDate(date)) date.setHours(0, 0, 0, 0);

}

/\*\*

\* Get Instance

\*/

}, {

key: "getInstance",

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_Datepicker;

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return Datepicker;

}(Component);

Datepicker.\_template = ['<div class= "modal datepicker-modal">', '<div class="modal-content datepicker-container">', '<div class="datepicker-date-display">', '<span class="year-text"></span>', '<span class="date-text"></span>', '</div>', '<div class="datepicker-calendar-container">', '<div class="datepicker-calendar"></div>', '<div class="datepicker-footer">', '<button class="btn-flat datepicker-clear waves-effect" style="visibility: hidden;" type="button"></button>', '<div class="confirmation-btns">', '<button class="btn-flat datepicker-cancel waves-effect" type="button"></button>', '<button class="btn-flat datepicker-done waves-effect" type="button"></button>', '</div>', '</div>', '</div>', '</div>', '</div>'].join('');

M.Datepicker = Datepicker;

if (M.jQueryLoaded) {

M.initializeJqueryWrapper(Datepicker, 'datepicker', 'M\_Datepicker');

}

})(cash);

;(function ($) {

'use strict';

var \_defaults = {

dialRadius: 135,

outerRadius: 105,

innerRadius: 70,

tickRadius: 20,

duration: 350,

container: null,

defaultTime: 'now', // default time, 'now' or '13:14' e.g.

fromNow: 0, // Millisecond offset from the defaultTime

showClearBtn: false,

// internationalization

i18n: {

cancel: 'Cancel',

clear: 'Clear',

done: 'Ok'

},

autoClose: false, // auto close when minute is selected

twelveHour: true, // change to 12 hour AM/PM clock from 24 hour

vibrate: true, // vibrate the device when dragging clock hand

// Callbacks

onOpenStart: null,

onOpenEnd: null,

onCloseStart: null,

onCloseEnd: null,

onSelect: null

};

/\*\*

\* @class

\*

\*/

var Timepicker = function (\_Component16) {

\_inherits(Timepicker, \_Component16);

function Timepicker(el, options) {

\_classCallCheck(this, Timepicker);

var \_this57 = \_possibleConstructorReturn(this, (Timepicker.\_\_proto\_\_ || Object.getPrototypeOf(Timepicker)).call(this, Timepicker, el, options));

\_this57.el.M\_Timepicker = \_this57;

\_this57.options = $.extend({}, Timepicker.defaults, options);

\_this57.id = M.guid();

\_this57.\_insertHTMLIntoDOM();

\_this57.\_setupModal();

\_this57.\_setupVariables();

\_this57.\_setupEventHandlers();

\_this57.\_clockSetup();

\_this57.\_pickerSetup();

return \_this57;

}

\_createClass(Timepicker, [{

key: "destroy",

/\*\*

\* Teardown component

\*/

value: function destroy() {

this.\_removeEventHandlers();

this.modal.destroy();

$(this.modalEl).remove();

this.el.M\_Timepicker = undefined;

}

/\*\*

\* Setup Event Handlers

\*/

}, {

key: "\_setupEventHandlers",

value: function \_setupEventHandlers() {

this.\_handleInputKeydownBound = this.\_handleInputKeydown.bind(this);

this.\_handleInputClickBound = this.\_handleInputClick.bind(this);

this.\_handleClockClickStartBound = this.\_handleClockClickStart.bind(this);

this.\_handleDocumentClickMoveBound = this.\_handleDocumentClickMove.bind(this);

this.\_handleDocumentClickEndBound = this.\_handleDocumentClickEnd.bind(this);

this.el.addEventListener('click', this.\_handleInputClickBound);

this.el.addEventListener('keydown', this.\_handleInputKeydownBound);

this.plate.addEventListener('mousedown', this.\_handleClockClickStartBound);

this.plate.addEventListener('touchstart', this.\_handleClockClickStartBound);

$(this.spanHours).on('click', this.showView.bind(this, 'hours'));

$(this.spanMinutes).on('click', this.showView.bind(this, 'minutes'));

}

}, {

key: "\_removeEventHandlers",

value: function \_removeEventHandlers() {

this.el.removeEventListener('click', this.\_handleInputClickBound);

this.el.removeEventListener('keydown', this.\_handleInputKeydownBound);

}

}, {

key: "\_handleInputClick",

value: function \_handleInputClick() {

this.open();

}

}, {

key: "\_handleInputKeydown",

value: function \_handleInputKeydown(e) {

if (e.which === M.keys.ENTER) {

e.preventDefault();

this.open();

}

}

}, {

key: "\_handleClockClickStart",

value: function \_handleClockClickStart(e) {

e.preventDefault();

var clockPlateBR = this.plate.getBoundingClientRect();

var offset = { x: clockPlateBR.left, y: clockPlateBR.top };

this.x0 = offset.x + this.options.dialRadius;

this.y0 = offset.y + this.options.dialRadius;

this.moved = false;

var clickPos = Timepicker.\_Pos(e);

this.dx = clickPos.x - this.x0;

this.dy = clickPos.y - this.y0;

// Set clock hands

this.setHand(this.dx, this.dy, false);

// Mousemove on document

document.addEventListener('mousemove', this.\_handleDocumentClickMoveBound);

document.addEventListener('touchmove', this.\_handleDocumentClickMoveBound);

// Mouseup on document

document.addEventListener('mouseup', this.\_handleDocumentClickEndBound);

document.addEventListener('touchend', this.\_handleDocumentClickEndBound);

}

}, {

key: "\_handleDocumentClickMove",

value: function \_handleDocumentClickMove(e) {

e.preventDefault();

var clickPos = Timepicker.\_Pos(e);

var x = clickPos.x - this.x0;

var y = clickPos.y - this.y0;

this.moved = true;

this.setHand(x, y, false, true);

}

}, {

key: "\_handleDocumentClickEnd",

value: function \_handleDocumentClickEnd(e) {

var \_this58 = this;

e.preventDefault();

document.removeEventListener('mouseup', this.\_handleDocumentClickEndBound);

document.removeEventListener('touchend', this.\_handleDocumentClickEndBound);

var clickPos = Timepicker.\_Pos(e);

var x = clickPos.x - this.x0;

var y = clickPos.y - this.y0;

if (this.moved && x === this.dx && y === this.dy) {

this.setHand(x, y);

}

if (this.currentView === 'hours') {

this.showView('minutes', this.options.duration / 2);

} else if (this.options.autoClose) {

$(this.minutesView).addClass('timepicker-dial-out');

setTimeout(function () {

\_this58.done();

}, this.options.duration / 2);

}

if (typeof this.options.onSelect === 'function') {

this.options.onSelect.call(this, this.hours, this.minutes);

}

// Unbind mousemove event

document.removeEventListener('mousemove', this.\_handleDocumentClickMoveBound);

document.removeEventListener('touchmove', this.\_handleDocumentClickMoveBound);

}

}, {

key: "\_insertHTMLIntoDOM",

value: function \_insertHTMLIntoDOM() {

this.$modalEl = $(Timepicker.\_template);

this.modalEl = this.$modalEl[0];

this.modalEl.id = 'modal-' + this.id;

// Append popover to input by default

var containerEl = document.querySelector(this.options.container);

if (this.options.container && !!containerEl) {

this.$modalEl.appendTo(containerEl);

} else {

this.$modalEl.insertBefore(this.el);

}

}

}, {

key: "\_setupModal",

value: function \_setupModal() {

var \_this59 = this;

this.modal = M.Modal.init(this.modalEl, {

onOpenStart: this.options.onOpenStart,

onOpenEnd: this.options.onOpenEnd,

onCloseStart: this.options.onCloseStart,

onCloseEnd: function () {

if (typeof \_this59.options.onCloseEnd === 'function') {

\_this59.options.onCloseEnd.call(\_this59);

}

\_this59.isOpen = false;

}

});

}

}, {

key: "\_setupVariables",

value: function \_setupVariables() {

this.currentView = 'hours';

this.vibrate = navigator.vibrate ? 'vibrate' : navigator.webkitVibrate ? 'webkitVibrate' : null;

this.\_canvas = this.modalEl.querySelector('.timepicker-canvas');

this.plate = this.modalEl.querySelector('.timepicker-plate');

this.hoursView = this.modalEl.querySelector('.timepicker-hours');

this.minutesView = this.modalEl.querySelector('.timepicker-minutes');

this.spanHours = this.modalEl.querySelector('.timepicker-span-hours');

this.spanMinutes = this.modalEl.querySelector('.timepicker-span-minutes');

this.spanAmPm = this.modalEl.querySelector('.timepicker-span-am-pm');

this.footer = this.modalEl.querySelector('.timepicker-footer');

this.amOrPm = 'PM';

}

}, {

key: "\_pickerSetup",

value: function \_pickerSetup() {

var $clearBtn = $("<button class=\"btn-flat timepicker-clear waves-effect\" style=\"visibility: hidden;\" type=\"button\" tabindex=\"" + (this.options.twelveHour ? '3' : '1') + "\">" + this.options.i18n.clear + "</button>").appendTo(this.footer).on('click', this.clear.bind(this));

if (this.options.showClearBtn) {

$clearBtn.css({ visibility: '' });

}

var confirmationBtnsContainer = $('<div class="confirmation-btns"></div>');

$('<button class="btn-flat timepicker-close waves-effect" type="button" tabindex="' + (this.options.twelveHour ? '3' : '1') + '">' + this.options.i18n.cancel + '</button>').appendTo(confirmationBtnsContainer).on('click', this.close.bind(this));

$('<button class="btn-flat timepicker-close waves-effect" type="button" tabindex="' + (this.options.twelveHour ? '3' : '1') + '">' + this.options.i18n.done + '</button>').appendTo(confirmationBtnsContainer).on('click', this.done.bind(this));

confirmationBtnsContainer.appendTo(this.footer);

}

}, {

key: "\_clockSetup",

value: function \_clockSetup() {

if (this.options.twelveHour) {

this.$amBtn = $('<div class="am-btn">AM</div>');

this.$pmBtn = $('<div class="pm-btn">PM</div>');

this.$amBtn.on('click', this.\_handleAmPmClick.bind(this)).appendTo(this.spanAmPm);

this.$pmBtn.on('click', this.\_handleAmPmClick.bind(this)).appendTo(this.spanAmPm);

}

this.\_buildHoursView();

this.\_buildMinutesView();

this.\_buildSVGClock();

}

}, {

key: "\_buildSVGClock",

value: function \_buildSVGClock() {

// Draw clock hands and others

var dialRadius = this.options.dialRadius;

var tickRadius = this.options.tickRadius;

var diameter = dialRadius \* 2;

var svg = Timepicker.\_createSVGEl('svg');

svg.setAttribute('class', 'timepicker-svg');

svg.setAttribute('width', diameter);

svg.setAttribute('height', diameter);

var g = Timepicker.\_createSVGEl('g');

g.setAttribute('transform', 'translate(' + dialRadius + ',' + dialRadius + ')');

var bearing = Timepicker.\_createSVGEl('circle');

bearing.setAttribute('class', 'timepicker-canvas-bearing');

bearing.setAttribute('cx', 0);

bearing.setAttribute('cy', 0);

bearing.setAttribute('r', 4);

var hand = Timepicker.\_createSVGEl('line');

hand.setAttribute('x1', 0);

hand.setAttribute('y1', 0);

var bg = Timepicker.\_createSVGEl('circle');

bg.setAttribute('class', 'timepicker-canvas-bg');

bg.setAttribute('r', tickRadius);

g.appendChild(hand);

g.appendChild(bg);

g.appendChild(bearing);

svg.appendChild(g);

this.\_canvas.appendChild(svg);

this.hand = hand;

this.bg = bg;

this.bearing = bearing;

this.g = g;

}

}, {

key: "\_buildHoursView",

value: function \_buildHoursView() {

var $tick = $('<div class="timepicker-tick"></div>');

// Hours view

if (this.options.twelveHour) {

for (var i = 1; i < 13; i += 1) {

var tick = $tick.clone();

var radian = i / 6 \* Math.PI;

var radius = this.options.outerRadius;

tick.css({

left: this.options.dialRadius + Math.sin(radian) \* radius - this.options.tickRadius + 'px',

top: this.options.dialRadius - Math.cos(radian) \* radius - this.options.tickRadius + 'px'

});

tick.html(i === 0 ? '00' : i);

this.hoursView.appendChild(tick[0]);

// tick.on(mousedownEvent, mousedown);

}

} else {

for (var \_i2 = 0; \_i2 < 24; \_i2 += 1) {

var \_tick = $tick.clone();

var \_radian = \_i2 / 6 \* Math.PI;

var inner = \_i2 > 0 && \_i2 < 13;

var \_radius = inner ? this.options.innerRadius : this.options.outerRadius;

\_tick.css({

left: this.options.dialRadius + Math.sin(\_radian) \* \_radius - this.options.tickRadius + 'px',

top: this.options.dialRadius - Math.cos(\_radian) \* \_radius - this.options.tickRadius + 'px'

});

\_tick.html(\_i2 === 0 ? '00' : \_i2);

this.hoursView.appendChild(\_tick[0]);

// tick.on(mousedownEvent, mousedown);

}

}

}

}, {

key: "\_buildMinutesView",

value: function \_buildMinutesView() {

var $tick = $('<div class="timepicker-tick"></div>');

// Minutes view

for (var i = 0; i < 60; i += 5) {

var tick = $tick.clone();

var radian = i / 30 \* Math.PI;

tick.css({

left: this.options.dialRadius + Math.sin(radian) \* this.options.outerRadius - this.options.tickRadius + 'px',

top: this.options.dialRadius - Math.cos(radian) \* this.options.outerRadius - this.options.tickRadius + 'px'

});

tick.html(Timepicker.\_addLeadingZero(i));

this.minutesView.appendChild(tick[0]);

}

}

}, {

key: "\_handleAmPmClick",

value: function \_handleAmPmClick(e) {

var $btnClicked = $(e.target);

this.amOrPm = $btnClicked.hasClass('am-btn') ? 'AM' : 'PM';

this.\_updateAmPmView();

}

}, {

key: "\_updateAmPmView",

value: function \_updateAmPmView() {

if (this.options.twelveHour) {

this.$amBtn.toggleClass('text-primary', this.amOrPm === 'AM');

this.$pmBtn.toggleClass('text-primary', this.amOrPm === 'PM');

}

}

}, {

key: "\_updateTimeFromInput",

value: function \_updateTimeFromInput() {

// Get the time

var value = ((this.el.value || this.options.defaultTime || '') + '').split(':');

if (this.options.twelveHour && !(typeof value[1] === 'undefined')) {

if (value[1].toUpperCase().indexOf('AM') > 0) {

this.amOrPm = 'AM';

} else {

this.amOrPm = 'PM';

}

value[1] = value[1].replace('AM', '').replace('PM', '');

}

if (value[0] === 'now') {

var now = new Date(+new Date() + this.options.fromNow);

value = [now.getHours(), now.getMinutes()];

if (this.options.twelveHour) {

this.amOrPm = value[0] >= 12 && value[0] < 24 ? 'PM' : 'AM';

}

}

this.hours = +value[0] || 0;

this.minutes = +value[1] || 0;

this.spanHours.innerHTML = this.hours;

this.spanMinutes.innerHTML = Timepicker.\_addLeadingZero(this.minutes);

this.\_updateAmPmView();

}

}, {

key: "showView",

value: function showView(view, delay) {

if (view === 'minutes' && $(this.hoursView).css('visibility') === 'visible') {

// raiseCallback(this.options.beforeHourSelect);

}

var isHours = view === 'hours',

nextView = isHours ? this.hoursView : this.minutesView,

hideView = isHours ? this.minutesView : this.hoursView;

this.currentView = view;

$(this.spanHours).toggleClass('text-primary', isHours);

$(this.spanMinutes).toggleClass('text-primary', !isHours);

// Transition view

hideView.classList.add('timepicker-dial-out');

$(nextView).css('visibility', 'visible').removeClass('timepicker-dial-out');

// Reset clock hand

this.resetClock(delay);

// After transitions ended

clearTimeout(this.toggleViewTimer);

this.toggleViewTimer = setTimeout(function () {

$(hideView).css('visibility', 'hidden');

}, this.options.duration);

}

}, {

key: "resetClock",

value: function resetClock(delay) {

var view = this.currentView,

value = this[view],

isHours = view === 'hours',

unit = Math.PI / (isHours ? 6 : 30),

radian = value \* unit,

radius = isHours && value > 0 && value < 13 ? this.options.innerRadius : this.options.outerRadius,

x = Math.sin(radian) \* radius,

y = -Math.cos(radian) \* radius,

self = this;

if (delay) {

$(this.canvas).addClass('timepicker-canvas-out');

setTimeout(function () {

$(self.canvas).removeClass('timepicker-canvas-out');

self.setHand(x, y);

}, delay);

} else {

this.setHand(x, y);

}

}

}, {

key: "setHand",

value: function setHand(x, y, roundBy5) {

var \_this60 = this;

var radian = Math.atan2(x, -y),

isHours = this.currentView === 'hours',

unit = Math.PI / (isHours || roundBy5 ? 6 : 30),

z = Math.sqrt(x \* x + y \* y),

inner = isHours && z < (this.options.outerRadius + this.options.innerRadius) / 2,

radius = inner ? this.options.innerRadius : this.options.outerRadius;

if (this.options.twelveHour) {

radius = this.options.outerRadius;

}

// Radian should in range [0, 2PI]

if (radian < 0) {

radian = Math.PI \* 2 + radian;

}

// Get the round value

var value = Math.round(radian / unit);

// Get the round radian

radian = value \* unit;

// Correct the hours or minutes

if (this.options.twelveHour) {

if (isHours) {

if (value === 0) value = 12;

} else {

if (roundBy5) value \*= 5;

if (value === 60) value = 0;

}

} else {

if (isHours) {

if (value === 12) {

value = 0;

}

value = inner ? value === 0 ? 12 : value : value === 0 ? 0 : value + 12;

} else {

if (roundBy5) {

value \*= 5;

}

if (value === 60) {

value = 0;

}

}

}

// Once hours or minutes changed, vibrate the device

if (this[this.currentView] !== value) {

if (this.vibrate && this.options.vibrate) {

// Do not vibrate too frequently

if (!this.vibrateTimer) {

navigator[this.vibrate](10);

this.vibrateTimer = setTimeout(function () {

\_this60.vibrateTimer = null;

}, 100);

}

}

}

this[this.currentView] = value;

if (isHours) {

this['spanHours'].innerHTML = value;

} else {

this['spanMinutes'].innerHTML = Timepicker.\_addLeadingZero(value);

}

// Set clock hand and others' position

var cx1 = Math.sin(radian) \* (radius - this.options.tickRadius),

cy1 = -Math.cos(radian) \* (radius - this.options.tickRadius),

cx2 = Math.sin(radian) \* radius,

cy2 = -Math.cos(radian) \* radius;

this.hand.setAttribute('x2', cx1);

this.hand.setAttribute('y2', cy1);

this.bg.setAttribute('cx', cx2);

this.bg.setAttribute('cy', cy2);

}

}, {

key: "open",

value: function open() {

if (this.isOpen) {

return;

}

this.isOpen = true;

this.\_updateTimeFromInput();

this.showView('hours');

this.modal.open();

}

}, {

key: "close",

value: function close() {

if (!this.isOpen) {

return;

}

this.isOpen = false;

this.modal.close();

}

/\*\*

\* Finish timepicker selection.

\*/

}, {

key: "done",

value: function done(e, clearValue) {

// Set input value

var last = this.el.value;

var value = clearValue ? '' : Timepicker.\_addLeadingZero(this.hours) + ':' + Timepicker.\_addLeadingZero(this.minutes);

this.time = value;

if (!clearValue && this.options.twelveHour) {

value = value + " " + this.amOrPm;

}

this.el.value = value;

// Trigger change event

if (value !== last) {

this.$el.trigger('change');

}

this.close();

this.el.focus();

}

}, {

key: "clear",

value: function clear() {

this.done(null, true);

}

}], [{

key: "init",

value: function init(els, options) {

return \_get(Timepicker.\_\_proto\_\_ || Object.getPrototypeOf(Timepicker), "init", this).call(this, this, els, options);

}

}, {

key: "\_addLeadingZero",

value: function \_addLeadingZero(num) {

return (num < 10 ? '0' : '') + num;

}

}, {

key: "\_createSVGEl",

value: function \_createSVGEl(name) {

var svgNS = 'http://www.w3.org/2000/svg';

return document.createElementNS(svgNS, name);

}

/\*\*

\* @typedef {Object} Point

\* @property {number} x The X Coordinate

\* @property {number} y The Y Coordinate

\*/

/\*\*

\* Get x position of mouse or touch event

\* @param {Event} e

\* @return {Point} x and y location

\*/

}, {

key: "\_Pos",

value: function \_Pos(e) {

if (e.targetTouches && e.targetTouches.length >= 1) {

return { x: e.targetTouches[0].clientX, y: e.targetTouches[0].clientY };

}

// mouse event

return { x: e.clientX, y: e.clientY };

}

/\*\*

\* Get Instance

\*/

}, {

key: "getInstance",

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_Timepicker;

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return Timepicker;

}(Component);

Timepicker.\_template = ['<div class= "modal timepicker-modal">', '<div class="modal-content timepicker-container">', '<div class="timepicker-digital-display">', '<div class="timepicker-text-container">', '<div class="timepicker-display-column">', '<span class="timepicker-span-hours text-primary"></span>', ':', '<span class="timepicker-span-minutes"></span>', '</div>', '<div class="timepicker-display-column timepicker-display-am-pm">', '<div class="timepicker-span-am-pm"></div>', '</div>', '</div>', '</div>', '<div class="timepicker-analog-display">', '<div class="timepicker-plate">', '<div class="timepicker-canvas"></div>', '<div class="timepicker-dial timepicker-hours"></div>', '<div class="timepicker-dial timepicker-minutes timepicker-dial-out"></div>', '</div>', '<div class="timepicker-footer"></div>', '</div>', '</div>', '</div>'].join('');

M.Timepicker = Timepicker;

if (M.jQueryLoaded) {

M.initializeJqueryWrapper(Timepicker, 'timepicker', 'M\_Timepicker');

}

})(cash);

;(function ($) {

'use strict';

var \_defaults = {};

/\*\*

\* @class

\*

\*/

var CharacterCounter = function (\_Component17) {

\_inherits(CharacterCounter, \_Component17);

/\*\*

\* Construct CharacterCounter instance

\* @constructor

\* @param {Element} el

\* @param {Object} options

\*/

function CharacterCounter(el, options) {

\_classCallCheck(this, CharacterCounter);

var \_this61 = \_possibleConstructorReturn(this, (CharacterCounter.\_\_proto\_\_ || Object.getPrototypeOf(CharacterCounter)).call(this, CharacterCounter, el, options));

\_this61.el.M\_CharacterCounter = \_this61;

/\*\*

\* Options for the character counter

\*/

\_this61.options = $.extend({}, CharacterCounter.defaults, options);

\_this61.isInvalid = false;

\_this61.isValidLength = false;

\_this61.\_setupCounter();

\_this61.\_setupEventHandlers();

return \_this61;

}

\_createClass(CharacterCounter, [{

key: "destroy",

/\*\*

\* Teardown component

\*/

value: function destroy() {

this.\_removeEventHandlers();

this.el.CharacterCounter = undefined;

this.\_removeCounter();

}

/\*\*

\* Setup Event Handlers

\*/

}, {

key: "\_setupEventHandlers",

value: function \_setupEventHandlers() {

this.\_handleUpdateCounterBound = this.updateCounter.bind(this);

this.el.addEventListener('focus', this.\_handleUpdateCounterBound, true);

this.el.addEventListener('input', this.\_handleUpdateCounterBound, true);

}

/\*\*

\* Remove Event Handlers

\*/

}, {

key: "\_removeEventHandlers",

value: function \_removeEventHandlers() {

this.el.removeEventListener('focus', this.\_handleUpdateCounterBound, true);

this.el.removeEventListener('input', this.\_handleUpdateCounterBound, true);

}

/\*\*

\* Setup counter element

\*/

}, {

key: "\_setupCounter",

value: function \_setupCounter() {

this.counterEl = document.createElement('span');

$(this.counterEl).addClass('character-counter').css({

float: 'right',

'font-size': '12px',

height: 1

});

this.$el.parent().append(this.counterEl);

}

/\*\*

\* Remove counter element

\*/

}, {

key: "\_removeCounter",

value: function \_removeCounter() {

$(this.counterEl).remove();

}

/\*\*

\* Update counter

\*/

}, {

key: "updateCounter",

value: function updateCounter() {

var maxLength = +this.$el.attr('data-length'),

actualLength = this.el.value.length;

this.isValidLength = actualLength <= maxLength;

var counterString = actualLength;

if (maxLength) {

counterString += '/' + maxLength;

this.\_validateInput();

}

$(this.counterEl).html(counterString);

}

/\*\*

\* Add validation classes

\*/

}, {

key: "\_validateInput",

value: function \_validateInput() {

if (this.isValidLength && this.isInvalid) {

this.isInvalid = false;

this.$el.removeClass('invalid');

} else if (!this.isValidLength && !this.isInvalid) {

this.isInvalid = true;

this.$el.removeClass('valid');

this.$el.addClass('invalid');

}

}

}], [{

key: "init",

value: function init(els, options) {

return \_get(CharacterCounter.\_\_proto\_\_ || Object.getPrototypeOf(CharacterCounter), "init", this).call(this, this, els, options);

}

/\*\*

\* Get Instance

\*/

}, {

key: "getInstance",

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_CharacterCounter;

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return CharacterCounter;

}(Component);

M.CharacterCounter = CharacterCounter;

if (M.jQueryLoaded) {

M.initializeJqueryWrapper(CharacterCounter, 'characterCounter', 'M\_CharacterCounter');

}

})(cash);

;(function ($) {

'use strict';

var \_defaults = {

duration: 200, // ms

dist: -100, // zoom scale TODO: make this more intuitive as an option

shift: 0, // spacing for center image

padding: 0, // Padding between non center items

numVisible: 5, // Number of visible items in carousel

fullWidth: false, // Change to full width styles

indicators: false, // Toggle indicators

noWrap: false, // Don't wrap around and cycle through items.

onCycleTo: null // Callback for when a new slide is cycled to.

};

/\*\*

\* @class

\*

\*/

var Carousel = function (\_Component18) {

\_inherits(Carousel, \_Component18);

/\*\*

\* Construct Carousel instance

\* @constructor

\* @param {Element} el

\* @param {Object} options

\*/

function Carousel(el, options) {

\_classCallCheck(this, Carousel);

var \_this62 = \_possibleConstructorReturn(this, (Carousel.\_\_proto\_\_ || Object.getPrototypeOf(Carousel)).call(this, Carousel, el, options));

\_this62.el.M\_Carousel = \_this62;

/\*\*

\* Options for the carousel

\* @member Carousel#options

\* @prop {Number} duration

\* @prop {Number} dist

\* @prop {Number} shift

\* @prop {Number} padding

\* @prop {Number} numVisible

\* @prop {Boolean} fullWidth

\* @prop {Boolean} indicators

\* @prop {Boolean} noWrap

\* @prop {Function} onCycleTo

\*/

\_this62.options = $.extend({}, Carousel.defaults, options);

// Setup

\_this62.hasMultipleSlides = \_this62.$el.find('.carousel-item').length > 1;

\_this62.showIndicators = \_this62.options.indicators && \_this62.hasMultipleSlides;

\_this62.noWrap = \_this62.options.noWrap || !\_this62.hasMultipleSlides;

\_this62.pressed = false;

\_this62.dragged = false;

\_this62.offset = \_this62.target = 0;

\_this62.images = [];

\_this62.itemWidth = \_this62.$el.find('.carousel-item').first().innerWidth();

\_this62.itemHeight = \_this62.$el.find('.carousel-item').first().innerHeight();

\_this62.dim = \_this62.itemWidth \* 2 + \_this62.options.padding || 1; // Make sure dim is non zero for divisions.

\_this62.\_autoScrollBound = \_this62.\_autoScroll.bind(\_this62);

\_this62.\_trackBound = \_this62.\_track.bind(\_this62);

// Full Width carousel setup

if (\_this62.options.fullWidth) {

\_this62.options.dist = 0;

\_this62.\_setCarouselHeight();

// Offset fixed items when indicators.

if (\_this62.showIndicators) {

\_this62.$el.find('.carousel-fixed-item').addClass('with-indicators');

}

}

// Iterate through slides

\_this62.$indicators = $('<ul class="indicators"></ul>');

\_this62.$el.find('.carousel-item').each(function (el, i) {

\_this62.images.push(el);

if (\_this62.showIndicators) {

var $indicator = $('<li class="indicator-item"></li>');

// Add active to first by default.

if (i === 0) {

$indicator[0].classList.add('active');

}

\_this62.$indicators.append($indicator);

}

});

if (\_this62.showIndicators) {

\_this62.$el.append(\_this62.$indicators);

}

\_this62.count = \_this62.images.length;

// Cap numVisible at count

\_this62.options.numVisible = Math.min(\_this62.count, \_this62.options.numVisible);

// Setup cross browser string

\_this62.xform = 'transform';

['webkit', 'Moz', 'O', 'ms'].every(function (prefix) {

var e = prefix + 'Transform';

if (typeof document.body.style[e] !== 'undefined') {

\_this62.xform = e;

return false;

}

return true;

});

\_this62.\_setupEventHandlers();

\_this62.\_scroll(\_this62.offset);

return \_this62;

}

\_createClass(Carousel, [{

key: "destroy",

/\*\*

\* Teardown component

\*/

value: function destroy() {

this.\_removeEventHandlers();

this.el.M\_Carousel = undefined;

}

/\*\*

\* Setup Event Handlers

\*/

}, {

key: "\_setupEventHandlers",

value: function \_setupEventHandlers() {

var \_this63 = this;

this.\_handleCarouselTapBound = this.\_handleCarouselTap.bind(this);

this.\_handleCarouselDragBound = this.\_handleCarouselDrag.bind(this);

this.\_handleCarouselReleaseBound = this.\_handleCarouselRelease.bind(this);

this.\_handleCarouselClickBound = this.\_handleCarouselClick.bind(this);

if (typeof window.ontouchstart !== 'undefined') {

this.el.addEventListener('touchstart', this.\_handleCarouselTapBound);

this.el.addEventListener('touchmove', this.\_handleCarouselDragBound);

this.el.addEventListener('touchend', this.\_handleCarouselReleaseBound);

}

this.el.addEventListener('mousedown', this.\_handleCarouselTapBound);

this.el.addEventListener('mousemove', this.\_handleCarouselDragBound);

this.el.addEventListener('mouseup', this.\_handleCarouselReleaseBound);

this.el.addEventListener('mouseleave', this.\_handleCarouselReleaseBound);

this.el.addEventListener('click', this.\_handleCarouselClickBound);

if (this.showIndicators && this.$indicators) {

this.\_handleIndicatorClickBound = this.\_handleIndicatorClick.bind(this);

this.$indicators.find('.indicator-item').each(function (el, i) {

el.addEventListener('click', \_this63.\_handleIndicatorClickBound);

});

}

// Resize

var throttledResize = M.throttle(this.\_handleResize, 200);

this.\_handleThrottledResizeBound = throttledResize.bind(this);

window.addEventListener('resize', this.\_handleThrottledResizeBound);

}

/\*\*

\* Remove Event Handlers

\*/

}, {

key: "\_removeEventHandlers",

value: function \_removeEventHandlers() {

var \_this64 = this;

if (typeof window.ontouchstart !== 'undefined') {

this.el.removeEventListener('touchstart', this.\_handleCarouselTapBound);

this.el.removeEventListener('touchmove', this.\_handleCarouselDragBound);

this.el.removeEventListener('touchend', this.\_handleCarouselReleaseBound);

}

this.el.removeEventListener('mousedown', this.\_handleCarouselTapBound);

this.el.removeEventListener('mousemove', this.\_handleCarouselDragBound);

this.el.removeEventListener('mouseup', this.\_handleCarouselReleaseBound);

this.el.removeEventListener('mouseleave', this.\_handleCarouselReleaseBound);

this.el.removeEventListener('click', this.\_handleCarouselClickBound);

if (this.showIndicators && this.$indicators) {

this.$indicators.find('.indicator-item').each(function (el, i) {

el.removeEventListener('click', \_this64.\_handleIndicatorClickBound);

});

}

window.removeEventListener('resize', this.\_handleThrottledResizeBound);

}

/\*\*

\* Handle Carousel Tap

\* @param {Event} e

\*/

}, {

key: "\_handleCarouselTap",

value: function \_handleCarouselTap(e) {

// Fixes firefox draggable image bug

if (e.type === 'mousedown' && $(e.target).is('img')) {

e.preventDefault();

}

this.pressed = true;

this.dragged = false;

this.verticalDragged = false;

this.reference = this.\_xpos(e);

this.referenceY = this.\_ypos(e);

this.velocity = this.amplitude = 0;

this.frame = this.offset;

this.timestamp = Date.now();

clearInterval(this.ticker);

this.ticker = setInterval(this.\_trackBound, 100);

}

/\*\*

\* Handle Carousel Drag

\* @param {Event} e

\*/

}, {

key: "\_handleCarouselDrag",

value: function \_handleCarouselDrag(e) {

var x = void 0,

y = void 0,

delta = void 0,

deltaY = void 0;

if (this.pressed) {

x = this.\_xpos(e);

y = this.\_ypos(e);

delta = this.reference - x;

deltaY = Math.abs(this.referenceY - y);

if (deltaY < 30 && !this.verticalDragged) {

// If vertical scrolling don't allow dragging.

if (delta > 2 || delta < -2) {

this.dragged = true;

this.reference = x;

this.\_scroll(this.offset + delta);

}

} else if (this.dragged) {

// If dragging don't allow vertical scroll.

e.preventDefault();

e.stopPropagation();

return false;

} else {

// Vertical scrolling.

this.verticalDragged = true;

}

}

if (this.dragged) {

// If dragging don't allow vertical scroll.

e.preventDefault();

e.stopPropagation();

return false;

}

}

/\*\*

\* Handle Carousel Release

\* @param {Event} e

\*/

}, {

key: "\_handleCarouselRelease",

value: function \_handleCarouselRelease(e) {

if (this.pressed) {

this.pressed = false;

} else {

return;

}

clearInterval(this.ticker);

this.target = this.offset;

if (this.velocity > 10 || this.velocity < -10) {

this.amplitude = 0.9 \* this.velocity;

this.target = this.offset + this.amplitude;

}

this.target = Math.round(this.target / this.dim) \* this.dim;

// No wrap of items.

if (this.noWrap) {

if (this.target >= this.dim \* (this.count - 1)) {

this.target = this.dim \* (this.count - 1);

} else if (this.target < 0) {

this.target = 0;

}

}

this.amplitude = this.target - this.offset;

this.timestamp = Date.now();

requestAnimationFrame(this.\_autoScrollBound);

if (this.dragged) {

e.preventDefault();

e.stopPropagation();

}

return false;

}

/\*\*

\* Handle Carousel CLick

\* @param {Event} e

\*/

}, {

key: "\_handleCarouselClick",

value: function \_handleCarouselClick(e) {

// Disable clicks if carousel was dragged.

if (this.dragged) {

e.preventDefault();

e.stopPropagation();

return false;

} else if (!this.options.fullWidth) {

var clickedIndex = $(e.target).closest('.carousel-item').index();

var diff = this.\_wrap(this.center) - clickedIndex;

// Disable clicks if carousel was shifted by click

if (diff !== 0) {

e.preventDefault();

e.stopPropagation();

}

this.\_cycleTo(clickedIndex);

}

}

/\*\*

\* Handle Indicator CLick

\* @param {Event} e

\*/

}, {

key: "\_handleIndicatorClick",

value: function \_handleIndicatorClick(e) {

e.stopPropagation();

var indicator = $(e.target).closest('.indicator-item');

if (indicator.length) {

this.\_cycleTo(indicator.index());

}

}

/\*\*

\* Handle Throttle Resize

\* @param {Event} e

\*/

}, {

key: "\_handleResize",

value: function \_handleResize(e) {

if (this.options.fullWidth) {

this.itemWidth = this.$el.find('.carousel-item').first().innerWidth();

this.imageHeight = this.$el.find('.carousel-item.active').height();

this.dim = this.itemWidth \* 2 + this.options.padding;

this.offset = this.center \* 2 \* this.itemWidth;

this.target = this.offset;

this.\_setCarouselHeight(true);

} else {

this.\_scroll();

}

}

/\*\*

\* Set carousel height based on first slide

\* @param {Booleam} imageOnly - true for image slides

\*/

}, {

key: "\_setCarouselHeight",

value: function \_setCarouselHeight(imageOnly) {

var \_this65 = this;

var firstSlide = this.$el.find('.carousel-item.active').length ? this.$el.find('.carousel-item.active').first() : this.$el.find('.carousel-item').first();

var firstImage = firstSlide.find('img').first();

if (firstImage.length) {

if (firstImage[0].complete) {

// If image won't trigger the load event

var imageHeight = firstImage.height();

if (imageHeight > 0) {

this.$el.css('height', imageHeight + 'px');

} else {

// If image still has no height, use the natural dimensions to calculate

var naturalWidth = firstImage[0].naturalWidth;

var naturalHeight = firstImage[0].naturalHeight;

var adjustedHeight = this.$el.width() / naturalWidth \* naturalHeight;

this.$el.css('height', adjustedHeight + 'px');

}

} else {

// Get height when image is loaded normally

firstImage.one('load', function (el, i) {

\_this65.$el.css('height', el.offsetHeight + 'px');

});

}

} else if (!imageOnly) {

var slideHeight = firstSlide.height();

this.$el.css('height', slideHeight + 'px');

}

}

/\*\*

\* Get x position from event

\* @param {Event} e

\*/

}, {

key: "\_xpos",

value: function \_xpos(e) {

// touch event

if (e.targetTouches && e.targetTouches.length >= 1) {

return e.targetTouches[0].clientX;

}

// mouse event

return e.clientX;

}

/\*\*

\* Get y position from event

\* @param {Event} e

\*/

}, {

key: "\_ypos",

value: function \_ypos(e) {

// touch event

if (e.targetTouches && e.targetTouches.length >= 1) {

return e.targetTouches[0].clientY;

}

// mouse event

return e.clientY;

}

/\*\*

\* Wrap index

\* @param {Number} x

\*/

}, {

key: "\_wrap",

value: function \_wrap(x) {

return x >= this.count ? x % this.count : x < 0 ? this.\_wrap(this.count + x % this.count) : x;

}

/\*\*

\* Tracks scrolling information

\*/

}, {

key: "\_track",

value: function \_track() {

var now = void 0,

elapsed = void 0,

delta = void 0,

v = void 0;

now = Date.now();

elapsed = now - this.timestamp;

this.timestamp = now;

delta = this.offset - this.frame;

this.frame = this.offset;

v = 1000 \* delta / (1 + elapsed);

this.velocity = 0.8 \* v + 0.2 \* this.velocity;

}

/\*\*

\* Auto scrolls to nearest carousel item.

\*/

}, {

key: "\_autoScroll",

value: function \_autoScroll() {

var elapsed = void 0,

delta = void 0;

if (this.amplitude) {

elapsed = Date.now() - this.timestamp;

delta = this.amplitude \* Math.exp(-elapsed / this.options.duration);

if (delta > 2 || delta < -2) {

this.\_scroll(this.target - delta);

requestAnimationFrame(this.\_autoScrollBound);

} else {

this.\_scroll(this.target);

}

}

}

/\*\*

\* Scroll to target

\* @param {Number} x

\*/

}, {

key: "\_scroll",

value: function \_scroll(x) {

var \_this66 = this;

// Track scrolling state

if (!this.$el.hasClass('scrolling')) {

this.el.classList.add('scrolling');

}

if (this.scrollingTimeout != null) {

window.clearTimeout(this.scrollingTimeout);

}

this.scrollingTimeout = window.setTimeout(function () {

\_this66.$el.removeClass('scrolling');

}, this.options.duration);

// Start actual scroll

var i = void 0,

half = void 0,

delta = void 0,

dir = void 0,

tween = void 0,

el = void 0,

alignment = void 0,

zTranslation = void 0,

tweenedOpacity = void 0,

centerTweenedOpacity = void 0;

var lastCenter = this.center;

var numVisibleOffset = 1 / this.options.numVisible;

this.offset = typeof x === 'number' ? x : this.offset;

this.center = Math.floor((this.offset + this.dim / 2) / this.dim);

delta = this.offset - this.center \* this.dim;

dir = delta < 0 ? 1 : -1;

tween = -dir \* delta \* 2 / this.dim;

half = this.count >> 1;

if (this.options.fullWidth) {

alignment = 'translateX(0)';

centerTweenedOpacity = 1;

} else {

alignment = 'translateX(' + (this.el.clientWidth - this.itemWidth) / 2 + 'px) ';

alignment += 'translateY(' + (this.el.clientHeight - this.itemHeight) / 2 + 'px)';

centerTweenedOpacity = 1 - numVisibleOffset \* tween;

}

// Set indicator active

if (this.showIndicators) {

var diff = this.center % this.count;

var activeIndicator = this.$indicators.find('.indicator-item.active');

if (activeIndicator.index() !== diff) {

activeIndicator.removeClass('active');

this.$indicators.find('.indicator-item').eq(diff)[0].classList.add('active');

}

}

// center

// Don't show wrapped items.

if (!this.noWrap || this.center >= 0 && this.center < this.count) {

el = this.images[this.\_wrap(this.center)];

// Add active class to center item.

if (!$(el).hasClass('active')) {

this.$el.find('.carousel-item').removeClass('active');

el.classList.add('active');

}

var transformString = alignment + " translateX(" + -delta / 2 + "px) translateX(" + dir \* this.options.shift \* tween \* i + "px) translateZ(" + this.options.dist \* tween + "px)";

this.\_updateItemStyle(el, centerTweenedOpacity, 0, transformString);

}

for (i = 1; i <= half; ++i) {

// right side

if (this.options.fullWidth) {

zTranslation = this.options.dist;

tweenedOpacity = i === half && delta < 0 ? 1 - tween : 1;

} else {

zTranslation = this.options.dist \* (i \* 2 + tween \* dir);

tweenedOpacity = 1 - numVisibleOffset \* (i \* 2 + tween \* dir);

}

// Don't show wrapped items.

if (!this.noWrap || this.center + i < this.count) {

el = this.images[this.\_wrap(this.center + i)];

var \_transformString = alignment + " translateX(" + (this.options.shift + (this.dim \* i - delta) / 2) + "px) translateZ(" + zTranslation + "px)";

this.\_updateItemStyle(el, tweenedOpacity, -i, \_transformString);

}

// left side

if (this.options.fullWidth) {

zTranslation = this.options.dist;

tweenedOpacity = i === half && delta > 0 ? 1 - tween : 1;

} else {

zTranslation = this.options.dist \* (i \* 2 - tween \* dir);

tweenedOpacity = 1 - numVisibleOffset \* (i \* 2 - tween \* dir);

}

// Don't show wrapped items.

if (!this.noWrap || this.center - i >= 0) {

el = this.images[this.\_wrap(this.center - i)];

var \_transformString2 = alignment + " translateX(" + (-this.options.shift + (-this.dim \* i - delta) / 2) + "px) translateZ(" + zTranslation + "px)";

this.\_updateItemStyle(el, tweenedOpacity, -i, \_transformString2);

}

}

// center

// Don't show wrapped items.

if (!this.noWrap || this.center >= 0 && this.center < this.count) {

el = this.images[this.\_wrap(this.center)];

var \_transformString3 = alignment + " translateX(" + -delta / 2 + "px) translateX(" + dir \* this.options.shift \* tween + "px) translateZ(" + this.options.dist \* tween + "px)";

this.\_updateItemStyle(el, centerTweenedOpacity, 0, \_transformString3);

}

// onCycleTo callback

var $currItem = this.$el.find('.carousel-item').eq(this.\_wrap(this.center));

if (lastCenter !== this.center && typeof this.options.onCycleTo === 'function') {

this.options.onCycleTo.call(this, $currItem[0], this.dragged);

}

// One time callback

if (typeof this.oneTimeCallback === 'function') {

this.oneTimeCallback.call(this, $currItem[0], this.dragged);

this.oneTimeCallback = null;

}

}

/\*\*

\* Cycle to target

\* @param {Element} el

\* @param {Number} opacity

\* @param {Number} zIndex

\* @param {String} transform

\*/

}, {

key: "\_updateItemStyle",

value: function \_updateItemStyle(el, opacity, zIndex, transform) {

el.style[this.xform] = transform;

el.style.zIndex = zIndex;

el.style.opacity = opacity;

el.style.visibility = 'visible';

}

/\*\*

\* Cycle to target

\* @param {Number} n

\* @param {Function} callback

\*/

}, {

key: "\_cycleTo",

value: function \_cycleTo(n, callback) {

var diff = this.center % this.count - n;

// Account for wraparound.

if (!this.noWrap) {

if (diff < 0) {

if (Math.abs(diff + this.count) < Math.abs(diff)) {

diff += this.count;

}

} else if (diff > 0) {

if (Math.abs(diff - this.count) < diff) {

diff -= this.count;

}

}

}

this.target = this.dim \* Math.round(this.offset / this.dim);

// Next

if (diff < 0) {

this.target += this.dim \* Math.abs(diff);

// Prev

} else if (diff > 0) {

this.target -= this.dim \* diff;

}

// Set one time callback

if (typeof callback === 'function') {

this.oneTimeCallback = callback;

}

// Scroll

if (this.offset !== this.target) {

this.amplitude = this.target - this.offset;

this.timestamp = Date.now();

requestAnimationFrame(this.\_autoScrollBound);

}

}

/\*\*

\* Cycle to next item

\* @param {Number} [n]

\*/

}, {

key: "next",

value: function next(n) {

if (n === undefined || isNaN(n)) {

n = 1;

}

var index = this.center + n;

if (index >= this.count || index < 0) {

if (this.noWrap) {

return;

}

index = this.\_wrap(index);

}

this.\_cycleTo(index);

}

/\*\*

\* Cycle to previous item

\* @param {Number} [n]

\*/

}, {

key: "prev",

value: function prev(n) {

if (n === undefined || isNaN(n)) {

n = 1;

}

var index = this.center - n;

if (index >= this.count || index < 0) {

if (this.noWrap) {

return;

}

index = this.\_wrap(index);

}

this.\_cycleTo(index);

}

/\*\*

\* Cycle to nth item

\* @param {Number} [n]

\* @param {Function} callback

\*/

}, {

key: "set",

value: function set(n, callback) {

if (n === undefined || isNaN(n)) {

n = 0;

}

if (n > this.count || n < 0) {

if (this.noWrap) {

return;

}

n = this.\_wrap(n);

}

this.\_cycleTo(n, callback);

}

}], [{

key: "init",

value: function init(els, options) {

return \_get(Carousel.\_\_proto\_\_ || Object.getPrototypeOf(Carousel), "init", this).call(this, this, els, options);

}

/\*\*

\* Get Instance

\*/

}, {

key: "getInstance",

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_Carousel;

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return Carousel;

}(Component);

M.Carousel = Carousel;

if (M.jQueryLoaded) {

M.initializeJqueryWrapper(Carousel, 'carousel', 'M\_Carousel');

}

})(cash);

;(function ($) {

'use strict';

var \_defaults = {

onOpen: undefined,

onClose: undefined

};

/\*\*

\* @class

\*

\*/

var TapTarget = function (\_Component19) {

\_inherits(TapTarget, \_Component19);

/\*\*

\* Construct TapTarget instance

\* @constructor

\* @param {Element} el

\* @param {Object} options

\*/

function TapTarget(el, options) {

\_classCallCheck(this, TapTarget);

var \_this67 = \_possibleConstructorReturn(this, (TapTarget.\_\_proto\_\_ || Object.getPrototypeOf(TapTarget)).call(this, TapTarget, el, options));

\_this67.el.M\_TapTarget = \_this67;

/\*\*

\* Options for the select

\* @member TapTarget#options

\* @prop {Function} onOpen - Callback function called when feature discovery is opened

\* @prop {Function} onClose - Callback function called when feature discovery is closed

\*/

\_this67.options = $.extend({}, TapTarget.defaults, options);

\_this67.isOpen = false;

// setup

\_this67.$origin = $('#' + \_this67.$el.attr('data-target'));

\_this67.\_setup();

\_this67.\_calculatePositioning();

\_this67.\_setupEventHandlers();

return \_this67;

}

\_createClass(TapTarget, [{

key: "destroy",

/\*\*

\* Teardown component

\*/

value: function destroy() {

this.\_removeEventHandlers();

this.el.TapTarget = undefined;

}

/\*\*

\* Setup Event Handlers

\*/

}, {

key: "\_setupEventHandlers",

value: function \_setupEventHandlers() {

this.\_handleDocumentClickBound = this.\_handleDocumentClick.bind(this);

this.\_handleTargetClickBound = this.\_handleTargetClick.bind(this);

this.\_handleOriginClickBound = this.\_handleOriginClick.bind(this);

this.el.addEventListener('click', this.\_handleTargetClickBound);

this.originEl.addEventListener('click', this.\_handleOriginClickBound);

// Resize

var throttledResize = M.throttle(this.\_handleResize, 200);

this.\_handleThrottledResizeBound = throttledResize.bind(this);

window.addEventListener('resize', this.\_handleThrottledResizeBound);

}

/\*\*

\* Remove Event Handlers

\*/

}, {

key: "\_removeEventHandlers",

value: function \_removeEventHandlers() {

this.el.removeEventListener('click', this.\_handleTargetClickBound);

this.originEl.removeEventListener('click', this.\_handleOriginClickBound);

window.removeEventListener('resize', this.\_handleThrottledResizeBound);

}

/\*\*

\* Handle Target Click

\* @param {Event} e

\*/

}, {

key: "\_handleTargetClick",

value: function \_handleTargetClick(e) {

this.open();

}

/\*\*

\* Handle Origin Click

\* @param {Event} e

\*/

}, {

key: "\_handleOriginClick",

value: function \_handleOriginClick(e) {

this.close();

}

/\*\*

\* Handle Resize

\* @param {Event} e

\*/

}, {

key: "\_handleResize",

value: function \_handleResize(e) {

this.\_calculatePositioning();

}

/\*\*

\* Handle Resize

\* @param {Event} e

\*/

}, {

key: "\_handleDocumentClick",

value: function \_handleDocumentClick(e) {

if (!$(e.target).closest('.tap-target-wrapper').length) {

this.close();

e.preventDefault();

e.stopPropagation();

}

}

/\*\*

\* Setup Tap Target

\*/

}, {

key: "\_setup",

value: function \_setup() {

// Creating tap target

this.wrapper = this.$el.parent()[0];

this.waveEl = $(this.wrapper).find('.tap-target-wave')[0];

this.originEl = $(this.wrapper).find('.tap-target-origin')[0];

this.contentEl = this.$el.find('.tap-target-content')[0];

// Creating wrapper

if (!$(this.wrapper).hasClass('.tap-target-wrapper')) {

this.wrapper = document.createElement('div');

this.wrapper.classList.add('tap-target-wrapper');

this.$el.before($(this.wrapper));

this.wrapper.append(this.el);

}

// Creating content

if (!this.contentEl) {

this.contentEl = document.createElement('div');

this.contentEl.classList.add('tap-target-content');

this.$el.append(this.contentEl);

}

// Creating foreground wave

if (!this.waveEl) {

this.waveEl = document.createElement('div');

this.waveEl.classList.add('tap-target-wave');

// Creating origin

if (!this.originEl) {

this.originEl = this.$origin.clone(true, true);

this.originEl.addClass('tap-target-origin');

this.originEl.removeAttr('id');

this.originEl.removeAttr('style');

this.originEl = this.originEl[0];

this.waveEl.append(this.originEl);

}

this.wrapper.append(this.waveEl);

}

}

/\*\*

\* Calculate positioning

\*/

}, {

key: "\_calculatePositioning",

value: function \_calculatePositioning() {

// Element or parent is fixed position?

var isFixed = this.$origin.css('position') === 'fixed';

if (!isFixed) {

var parents = this.$origin.parents();

for (var i = 0; i < parents.length; i++) {

isFixed = $(parents[i]).css('position') == 'fixed';

if (isFixed) {

break;

}

}

}

// Calculating origin

var originWidth = this.$origin.outerWidth();

var originHeight = this.$origin.outerHeight();

var originTop = isFixed ? this.$origin.offset().top - M.getDocumentScrollTop() : this.$origin.offset().top;

var originLeft = isFixed ? this.$origin.offset().left - M.getDocumentScrollLeft() : this.$origin.offset().left;

// Calculating screen

var windowWidth = window.innerWidth;

var windowHeight = window.innerHeight;

var centerX = windowWidth / 2;

var centerY = windowHeight / 2;

var isLeft = originLeft <= centerX;

var isRight = originLeft > centerX;

var isTop = originTop <= centerY;

var isBottom = originTop > centerY;

var isCenterX = originLeft >= windowWidth \* 0.25 && originLeft <= windowWidth \* 0.75;

// Calculating tap target

var tapTargetWidth = this.$el.outerWidth();

var tapTargetHeight = this.$el.outerHeight();

var tapTargetTop = originTop + originHeight / 2 - tapTargetHeight / 2;

var tapTargetLeft = originLeft + originWidth / 2 - tapTargetWidth / 2;

var tapTargetPosition = isFixed ? 'fixed' : 'absolute';

// Calculating content

var tapTargetTextWidth = isCenterX ? tapTargetWidth : tapTargetWidth / 2 + originWidth;

var tapTargetTextHeight = tapTargetHeight / 2;

var tapTargetTextTop = isTop ? tapTargetHeight / 2 : 0;

var tapTargetTextBottom = 0;

var tapTargetTextLeft = isLeft && !isCenterX ? tapTargetWidth / 2 - originWidth : 0;

var tapTargetTextRight = 0;

var tapTargetTextPadding = originWidth;

var tapTargetTextAlign = isBottom ? 'bottom' : 'top';

// Calculating wave

var tapTargetWaveWidth = originWidth > originHeight ? originWidth \* 2 : originWidth \* 2;

var tapTargetWaveHeight = tapTargetWaveWidth;

var tapTargetWaveTop = tapTargetHeight / 2 - tapTargetWaveHeight / 2;

var tapTargetWaveLeft = tapTargetWidth / 2 - tapTargetWaveWidth / 2;

// Setting tap target

var tapTargetWrapperCssObj = {};

tapTargetWrapperCssObj.top = isTop ? tapTargetTop + 'px' : '';

tapTargetWrapperCssObj.right = isRight ? windowWidth - tapTargetLeft - tapTargetWidth + 'px' : '';

tapTargetWrapperCssObj.bottom = isBottom ? windowHeight - tapTargetTop - tapTargetHeight + 'px' : '';

tapTargetWrapperCssObj.left = isLeft ? tapTargetLeft + 'px' : '';

tapTargetWrapperCssObj.position = tapTargetPosition;

$(this.wrapper).css(tapTargetWrapperCssObj);

// Setting content

$(this.contentEl).css({

width: tapTargetTextWidth + 'px',

height: tapTargetTextHeight + 'px',

top: tapTargetTextTop + 'px',

right: tapTargetTextRight + 'px',

bottom: tapTargetTextBottom + 'px',

left: tapTargetTextLeft + 'px',

padding: tapTargetTextPadding + 'px',

verticalAlign: tapTargetTextAlign

});

// Setting wave

$(this.waveEl).css({

top: tapTargetWaveTop + 'px',

left: tapTargetWaveLeft + 'px',

width: tapTargetWaveWidth + 'px',

height: tapTargetWaveHeight + 'px'

});

}

/\*\*

\* Open TapTarget

\*/

}, {

key: "open",

value: function open() {

if (this.isOpen) {

return;

}

// onOpen callback

if (typeof this.options.onOpen === 'function') {

this.options.onOpen.call(this, this.$origin[0]);

}

this.isOpen = true;

this.wrapper.classList.add('open');

document.body.addEventListener('click', this.\_handleDocumentClickBound, true);

document.body.addEventListener('touchend', this.\_handleDocumentClickBound);

}

/\*\*

\* Close Tap Target

\*/

}, {

key: "close",

value: function close() {

if (!this.isOpen) {

return;

}

// onClose callback

if (typeof this.options.onClose === 'function') {

this.options.onClose.call(this, this.$origin[0]);

}

this.isOpen = false;

this.wrapper.classList.remove('open');

document.body.removeEventListener('click', this.\_handleDocumentClickBound, true);

document.body.removeEventListener('touchend', this.\_handleDocumentClickBound);

}

}], [{

key: "init",

value: function init(els, options) {

return \_get(TapTarget.\_\_proto\_\_ || Object.getPrototypeOf(TapTarget), "init", this).call(this, this, els, options);

}

/\*\*

\* Get Instance

\*/

}, {

key: "getInstance",

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_TapTarget;

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return TapTarget;

}(Component);

M.TapTarget = TapTarget;

if (M.jQueryLoaded) {

M.initializeJqueryWrapper(TapTarget, 'tapTarget', 'M\_TapTarget');

}

})(cash);

;(function ($) {

'use strict';

var \_defaults = {

classes: '',

dropdownOptions: {}

};

/\*\*

\* @class

\*

\*/

var FormSelect = function (\_Component20) {

\_inherits(FormSelect, \_Component20);

/\*\*

\* Construct FormSelect instance

\* @constructor

\* @param {Element} el

\* @param {Object} options

\*/

function FormSelect(el, options) {

\_classCallCheck(this, FormSelect);

// Don't init if browser default version

var \_this68 = \_possibleConstructorReturn(this, (FormSelect.\_\_proto\_\_ || Object.getPrototypeOf(FormSelect)).call(this, FormSelect, el, options));

if (\_this68.$el.hasClass('browser-default')) {

return \_possibleConstructorReturn(\_this68);

}

\_this68.el.M\_FormSelect = \_this68;

/\*\*

\* Options for the select

\* @member FormSelect#options

\*/

\_this68.options = $.extend({}, FormSelect.defaults, options);

\_this68.isMultiple = \_this68.$el.prop('multiple');

// Setup

\_this68.el.tabIndex = -1;

\_this68.\_keysSelected = {};

\_this68.\_valueDict = {}; // Maps key to original and generated option element.

\_this68.\_setupDropdown();

\_this68.\_setupEventHandlers();

return \_this68;

}

\_createClass(FormSelect, [{

key: "destroy",

/\*\*

\* Teardown component

\*/

value: function destroy() {

this.\_removeEventHandlers();

this.\_removeDropdown();

this.el.M\_FormSelect = undefined;

}

/\*\*

\* Setup Event Handlers

\*/

}, {

key: "\_setupEventHandlers",

value: function \_setupEventHandlers() {

var \_this69 = this;

this.\_handleSelectChangeBound = this.\_handleSelectChange.bind(this);

this.\_handleOptionClickBound = this.\_handleOptionClick.bind(this);

this.\_handleInputClickBound = this.\_handleInputClick.bind(this);

$(this.dropdownOptions).find('li:not(.optgroup)').each(function (el) {

el.addEventListener('click', \_this69.\_handleOptionClickBound);

});

this.el.addEventListener('change', this.\_handleSelectChangeBound);

this.input.addEventListener('click', this.\_handleInputClickBound);

}

/\*\*

\* Remove Event Handlers

\*/

}, {

key: "\_removeEventHandlers",

value: function \_removeEventHandlers() {

var \_this70 = this;

$(this.dropdownOptions).find('li:not(.optgroup)').each(function (el) {

el.removeEventListener('click', \_this70.\_handleOptionClickBound);

});

this.el.removeEventListener('change', this.\_handleSelectChangeBound);

this.input.removeEventListener('click', this.\_handleInputClickBound);

}

/\*\*

\* Handle Select Change

\* @param {Event} e

\*/

}, {

key: "\_handleSelectChange",

value: function \_handleSelectChange(e) {

this.\_setValueToInput();

}

/\*\*

\* Handle Option Click

\* @param {Event} e

\*/

}, {

key: "\_handleOptionClick",

value: function \_handleOptionClick(e) {

e.preventDefault();

var option = $(e.target).closest('li')[0];

var key = option.id;

if (!$(option).hasClass('disabled') && !$(option).hasClass('optgroup') && key.length) {

var selected = true;

if (this.isMultiple) {

// Deselect placeholder option if still selected.

var placeholderOption = $(this.dropdownOptions).find('li.disabled.selected');

if (placeholderOption.length) {

placeholderOption.removeClass('selected');

placeholderOption.find('input[type="checkbox"]').prop('checked', false);

this.\_toggleEntryFromArray(placeholderOption[0].id);

}

selected = this.\_toggleEntryFromArray(key);

} else {

$(this.dropdownOptions).find('li').removeClass('selected');

$(option).toggleClass('selected', selected);

}

// Set selected on original select option

// Only trigger if selected state changed

var prevSelected = $(this.\_valueDict[key].el).prop('selected');

if (prevSelected !== selected) {

$(this.\_valueDict[key].el).prop('selected', selected);

this.$el.trigger('change');

}

}

e.stopPropagation();

}

/\*\*

\* Handle Input Click

\*/

}, {

key: "\_handleInputClick",

value: function \_handleInputClick() {

if (this.dropdown && this.dropdown.isOpen) {

this.\_setValueToInput();

this.\_setSelectedStates();

}

}

/\*\*

\* Setup dropdown

\*/

}, {

key: "\_setupDropdown",

value: function \_setupDropdown() {

var \_this71 = this;

this.wrapper = document.createElement('div');

$(this.wrapper).addClass('select-wrapper ' + this.options.classes);

this.$el.before($(this.wrapper));

this.wrapper.appendChild(this.el);

if (this.el.disabled) {

this.wrapper.classList.add('disabled');

}

// Create dropdown

this.$selectOptions = this.$el.children('option, optgroup');

this.dropdownOptions = document.createElement('ul');

this.dropdownOptions.id = "select-options-" + M.guid();

$(this.dropdownOptions).addClass('dropdown-content select-dropdown ' + (this.isMultiple ? 'multiple-select-dropdown' : ''));

// Create dropdown structure.

if (this.$selectOptions.length) {

this.$selectOptions.each(function (el) {

if ($(el).is('option')) {

// Direct descendant option.

var optionEl = void 0;

if (\_this71.isMultiple) {

optionEl = \_this71.\_appendOptionWithIcon(\_this71.$el, el, 'multiple');

} else {

optionEl = \_this71.\_appendOptionWithIcon(\_this71.$el, el);

}

\_this71.\_addOptionToValueDict(el, optionEl);

} else if ($(el).is('optgroup')) {

// Optgroup.

var selectOptions = $(el).children('option');

$(\_this71.dropdownOptions).append($('<li class="optgroup"><span>' + el.getAttribute('label') + '</span></li>')[0]);

selectOptions.each(function (el) {

var optionEl = \_this71.\_appendOptionWithIcon(\_this71.$el, el, 'optgroup-option');

\_this71.\_addOptionToValueDict(el, optionEl);

});

}

});

}

this.$el.after(this.dropdownOptions);

// Add input dropdown

this.input = document.createElement('input');

$(this.input).addClass('select-dropdown dropdown-trigger');

this.input.setAttribute('type', 'text');

this.input.setAttribute('readonly', 'true');

this.input.setAttribute('data-target', this.dropdownOptions.id);

if (this.el.disabled) {

$(this.input).prop('disabled', 'true');

}

this.$el.before(this.input);

this.\_setValueToInput();

// Add caret

var dropdownIcon = $('<svg class="caret" height="24" viewBox="0 0 24 24" width="24" xmlns="http://www.w3.org/2000/svg"><path d="M7 10l5 5 5-5z"/><path d="M0 0h24v24H0z" fill="none"/></svg>');

this.$el.before(dropdownIcon[0]);

// Initialize dropdown

if (!this.el.disabled) {

var dropdownOptions = $.extend({}, this.options.dropdownOptions);

// Add callback for centering selected option when dropdown content is scrollable

dropdownOptions.onOpenEnd = function (el) {

var selectedOption = $(\_this71.dropdownOptions).find('.selected').first();

if (selectedOption.length) {

// Focus selected option in dropdown

M.keyDown = true;

\_this71.dropdown.focusedIndex = selectedOption.index();

\_this71.dropdown.\_focusFocusedItem();

M.keyDown = false;

// Handle scrolling to selected option

if (\_this71.dropdown.isScrollable) {

var scrollOffset = selectedOption[0].getBoundingClientRect().top - \_this71.dropdownOptions.getBoundingClientRect().top; // scroll to selected option

scrollOffset -= \_this71.dropdownOptions.clientHeight / 2; // center in dropdown

\_this71.dropdownOptions.scrollTop = scrollOffset;

}

}

};

if (this.isMultiple) {

dropdownOptions.closeOnClick = false;

}

this.dropdown = M.Dropdown.init(this.input, dropdownOptions);

}

// Add initial selections

this.\_setSelectedStates();

}

/\*\*

\* Add option to value dict

\* @param {Element} el original option element

\* @param {Element} optionEl generated option element

\*/

}, {

key: "\_addOptionToValueDict",

value: function \_addOptionToValueDict(el, optionEl) {

var index = Object.keys(this.\_valueDict).length;

var key = this.dropdownOptions.id + index;

var obj = {};

optionEl.id = key;

obj.el = el;

obj.optionEl = optionEl;

this.\_valueDict[key] = obj;

}

/\*\*

\* Remove dropdown

\*/

}, {

key: "\_removeDropdown",

value: function \_removeDropdown() {

$(this.wrapper).find('.caret').remove();

$(this.input).remove();

$(this.dropdownOptions).remove();

$(this.wrapper).before(this.$el);

$(this.wrapper).remove();

}

/\*\*

\* Setup dropdown

\* @param {Element} select select element

\* @param {Element} option option element from select

\* @param {String} type

\* @return {Element} option element added

\*/

}, {

key: "\_appendOptionWithIcon",

value: function \_appendOptionWithIcon(select, option, type) {

// Add disabled attr if disabled

var disabledClass = option.disabled ? 'disabled ' : '';

var optgroupClass = type === 'optgroup-option' ? 'optgroup-option ' : '';

var multipleCheckbox = this.isMultiple ? "<label><input type=\"checkbox\"" + disabledClass + "\"/><span>" + option.innerHTML + "</span></label>" : option.innerHTML;

var liEl = $('<li></li>');

var spanEl = $('<span></span>');

spanEl.html(multipleCheckbox);

liEl.addClass(disabledClass + " " + optgroupClass);

liEl.append(spanEl);

// add icons

var iconUrl = option.getAttribute('data-icon');

if (!!iconUrl) {

var imgEl = $("<img alt=\"\" src=\"" + iconUrl + "\">");

liEl.prepend(imgEl);

}

// Check for multiple type.

$(this.dropdownOptions).append(liEl[0]);

return liEl[0];

}

/\*\*

\* Toggle entry from option

\* @param {String} key Option key

\* @return {Boolean} if entry was added or removed

\*/

}, {

key: "\_toggleEntryFromArray",

value: function \_toggleEntryFromArray(key) {

var notAdded = !this.\_keysSelected.hasOwnProperty(key);

var $optionLi = $(this.\_valueDict[key].optionEl);

if (notAdded) {

this.\_keysSelected[key] = true;

} else {

delete this.\_keysSelected[key];

}

$optionLi.toggleClass('selected', notAdded);

// Set checkbox checked value

$optionLi.find('input[type="checkbox"]').prop('checked', notAdded);

// use notAdded instead of true (to detect if the option is selected or not)

$optionLi.prop('selected', notAdded);

return notAdded;

}

/\*\*

\* Set text value to input

\*/

}, {

key: "\_setValueToInput",

value: function \_setValueToInput() {

var values = [];

var options = this.$el.find('option');

options.each(function (el) {

if ($(el).prop('selected')) {

var text = $(el).text();

values.push(text);

}

});

if (!values.length) {

var firstDisabled = this.$el.find('option:disabled').eq(0);

if (firstDisabled.length && firstDisabled[0].value === '') {

values.push(firstDisabled.text());

}

}

this.input.value = values.join(', ');

}

/\*\*

\* Set selected state of dropdown to match actual select element

\*/

}, {

key: "\_setSelectedStates",

value: function \_setSelectedStates() {

this.\_keysSelected = {};

for (var key in this.\_valueDict) {

var option = this.\_valueDict[key];

var optionIsSelected = $(option.el).prop('selected');

$(option.optionEl).find('input[type="checkbox"]').prop('checked', optionIsSelected);

if (optionIsSelected) {

this.\_activateOption($(this.dropdownOptions), $(option.optionEl));

this.\_keysSelected[key] = true;

} else {

$(option.optionEl).removeClass('selected');

}

}

}

/\*\*

\* Make option as selected and scroll to selected position

\* @param {jQuery} collection Select options jQuery element

\* @param {Element} newOption element of the new option

\*/

}, {

key: "\_activateOption",

value: function \_activateOption(collection, newOption) {

if (newOption) {

if (!this.isMultiple) {

collection.find('li.selected').removeClass('selected');

}

var option = $(newOption);

option.addClass('selected');

}

}

/\*\*

\* Get Selected Values

\* @return {Array} Array of selected values

\*/

}, {

key: "getSelectedValues",

value: function getSelectedValues() {

var selectedValues = [];

for (var key in this.\_keysSelected) {

selectedValues.push(this.\_valueDict[key].el.value);

}

return selectedValues;

}

}], [{

key: "init",

value: function init(els, options) {

return \_get(FormSelect.\_\_proto\_\_ || Object.getPrototypeOf(FormSelect), "init", this).call(this, this, els, options);

}

/\*\*

\* Get Instance

\*/

}, {

key: "getInstance",

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_FormSelect;

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return FormSelect;

}(Component);

M.FormSelect = FormSelect;

if (M.jQueryLoaded) {

M.initializeJqueryWrapper(FormSelect, 'formSelect', 'M\_FormSelect');

}

})(cash);

;(function ($, anim) {

'use strict';

var \_defaults = {};

/\*\*

\* @class

\*

\*/

var Range = function (\_Component21) {

\_inherits(Range, \_Component21);

/\*\*

\* Construct Range instance

\* @constructor

\* @param {Element} el

\* @param {Object} options

\*/

function Range(el, options) {

\_classCallCheck(this, Range);

var \_this72 = \_possibleConstructorReturn(this, (Range.\_\_proto\_\_ || Object.getPrototypeOf(Range)).call(this, Range, el, options));

\_this72.el.M\_Range = \_this72;

/\*\*

\* Options for the range

\* @member Range#options

\*/

\_this72.options = $.extend({}, Range.defaults, options);

\_this72.\_mousedown = false;

// Setup

\_this72.\_setupThumb();

\_this72.\_setupEventHandlers();

return \_this72;

}

\_createClass(Range, [{

key: "destroy",

/\*\*

\* Teardown component

\*/

value: function destroy() {

this.\_removeEventHandlers();

this.\_removeThumb();

this.el.M\_Range = undefined;

}

/\*\*

\* Setup Event Handlers

\*/

}, {

key: "\_setupEventHandlers",

value: function \_setupEventHandlers() {

this.\_handleRangeChangeBound = this.\_handleRangeChange.bind(this);

this.\_handleRangeMousedownTouchstartBound = this.\_handleRangeMousedownTouchstart.bind(this);

this.\_handleRangeInputMousemoveTouchmoveBound = this.\_handleRangeInputMousemoveTouchmove.bind(this);

this.\_handleRangeMouseupTouchendBound = this.\_handleRangeMouseupTouchend.bind(this);

this.\_handleRangeBlurMouseoutTouchleaveBound = this.\_handleRangeBlurMouseoutTouchleave.bind(this);

this.el.addEventListener('change', this.\_handleRangeChangeBound);

this.el.addEventListener('mousedown', this.\_handleRangeMousedownTouchstartBound);

this.el.addEventListener('touchstart', this.\_handleRangeMousedownTouchstartBound);

this.el.addEventListener('input', this.\_handleRangeInputMousemoveTouchmoveBound);

this.el.addEventListener('mousemove', this.\_handleRangeInputMousemoveTouchmoveBound);

this.el.addEventListener('touchmove', this.\_handleRangeInputMousemoveTouchmoveBound);

this.el.addEventListener('mouseup', this.\_handleRangeMouseupTouchendBound);

this.el.addEventListener('touchend', this.\_handleRangeMouseupTouchendBound);

this.el.addEventListener('blur', this.\_handleRangeBlurMouseoutTouchleaveBound);

this.el.addEventListener('mouseout', this.\_handleRangeBlurMouseoutTouchleaveBound);

this.el.addEventListener('touchleave', this.\_handleRangeBlurMouseoutTouchleaveBound);

}

/\*\*

\* Remove Event Handlers

\*/

}, {

key: "\_removeEventHandlers",

value: function \_removeEventHandlers() {

this.el.removeEventListener('change', this.\_handleRangeChangeBound);

this.el.removeEventListener('mousedown', this.\_handleRangeMousedownTouchstartBound);

this.el.removeEventListener('touchstart', this.\_handleRangeMousedownTouchstartBound);

this.el.removeEventListener('input', this.\_handleRangeInputMousemoveTouchmoveBound);

this.el.removeEventListener('mousemove', this.\_handleRangeInputMousemoveTouchmoveBound);

this.el.removeEventListener('touchmove', this.\_handleRangeInputMousemoveTouchmoveBound);

this.el.removeEventListener('mouseup', this.\_handleRangeMouseupTouchendBound);

this.el.removeEventListener('touchend', this.\_handleRangeMouseupTouchendBound);

this.el.removeEventListener('blur', this.\_handleRangeBlurMouseoutTouchleaveBound);

this.el.removeEventListener('mouseout', this.\_handleRangeBlurMouseoutTouchleaveBound);

this.el.removeEventListener('touchleave', this.\_handleRangeBlurMouseoutTouchleaveBound);

}

/\*\*

\* Handle Range Change

\* @param {Event} e

\*/

}, {

key: "\_handleRangeChange",

value: function \_handleRangeChange() {

$(this.value).html(this.$el.val());

if (!$(this.thumb).hasClass('active')) {

this.\_showRangeBubble();

}

var offsetLeft = this.\_calcRangeOffset();

$(this.thumb).addClass('active').css('left', offsetLeft + 'px');

}

/\*\*

\* Handle Range Mousedown and Touchstart

\* @param {Event} e

\*/

}, {

key: "\_handleRangeMousedownTouchstart",

value: function \_handleRangeMousedownTouchstart(e) {

// Set indicator value

$(this.value).html(this.$el.val());

this.\_mousedown = true;

this.$el.addClass('active');

if (!$(this.thumb).hasClass('active')) {

this.\_showRangeBubble();

}

if (e.type !== 'input') {

var offsetLeft = this.\_calcRangeOffset();

$(this.thumb).addClass('active').css('left', offsetLeft + 'px');

}

}

/\*\*

\* Handle Range Input, Mousemove and Touchmove

\*/

}, {

key: "\_handleRangeInputMousemoveTouchmove",

value: function \_handleRangeInputMousemoveTouchmove() {

if (this.\_mousedown) {

if (!$(this.thumb).hasClass('active')) {

this.\_showRangeBubble();

}

var offsetLeft = this.\_calcRangeOffset();

$(this.thumb).addClass('active').css('left', offsetLeft + 'px');

$(this.value).html(this.$el.val());

}

}

/\*\*

\* Handle Range Mouseup and Touchend

\*/

}, {

key: "\_handleRangeMouseupTouchend",

value: function \_handleRangeMouseupTouchend() {

this.\_mousedown = false;

this.$el.removeClass('active');

}

/\*\*

\* Handle Range Blur, Mouseout and Touchleave

\*/

}, {

key: "\_handleRangeBlurMouseoutTouchleave",

value: function \_handleRangeBlurMouseoutTouchleave() {

if (!this.\_mousedown) {

var paddingLeft = parseInt(this.$el.css('padding-left'));

var marginLeft = 7 + paddingLeft + 'px';

if ($(this.thumb).hasClass('active')) {

anim.remove(this.thumb);

anim({

targets: this.thumb,

height: 0,

width: 0,

top: 10,

easing: 'easeOutQuad',

marginLeft: marginLeft,

duration: 100

});

}

$(this.thumb).removeClass('active');

}

}

/\*\*

\* Setup dropdown

\*/

}, {

key: "\_setupThumb",

value: function \_setupThumb() {

this.thumb = document.createElement('span');

this.value = document.createElement('span');

$(this.thumb).addClass('thumb');

$(this.value).addClass('value');

$(this.thumb).append(this.value);

this.$el.after(this.thumb);

}

/\*\*

\* Remove dropdown

\*/

}, {

key: "\_removeThumb",

value: function \_removeThumb() {

$(this.thumb).remove();

}

/\*\*

\* morph thumb into bubble

\*/

}, {

key: "\_showRangeBubble",

value: function \_showRangeBubble() {

var paddingLeft = parseInt($(this.thumb).parent().css('padding-left'));

var marginLeft = -7 + paddingLeft + 'px'; // TODO: fix magic number?

anim.remove(this.thumb);

anim({

targets: this.thumb,

height: 30,

width: 30,

top: -30,

marginLeft: marginLeft,

duration: 300,

easing: 'easeOutQuint'

});

}

/\*\*

\* Calculate the offset of the thumb

\* @return {Number} offset in pixels

\*/

}, {

key: "\_calcRangeOffset",

value: function \_calcRangeOffset() {

var width = this.$el.width() - 15;

var max = parseFloat(this.$el.attr('max')) || 100; // Range default max

var min = parseFloat(this.$el.attr('min')) || 0; // Range default min

var percent = (parseFloat(this.$el.val()) - min) / (max - min);

return percent \* width;

}

}], [{

key: "init",

value: function init(els, options) {

return \_get(Range.\_\_proto\_\_ || Object.getPrototypeOf(Range), "init", this).call(this, this, els, options);

}

/\*\*

\* Get Instance

\*/

}, {

key: "getInstance",

value: function getInstance(el) {

var domElem = !!el.jquery ? el[0] : el;

return domElem.M\_Range;

}

}, {

key: "defaults",

get: function () {

return \_defaults;

}

}]);

return Range;

}(Component);

M.Range = Range;

if (M.jQueryLoaded) {

M.initializeJqueryWrapper(Range, 'range', 'M\_Range');

}

Range.init($('input[type=range]'));

})(cash, M.anime);