Gulpfile: Sass to CSS

The Sass section in the Gulpfile is delimited by a `BEGIN APP:SASS` and `END APP:SASS` and consists of 6 tasks. Lets begin by understanding the flow of each task. While describing the following tasks we assume the project name to be ***app***.

Tasks	Flow
Task 1 (sass:app)	The first sass gulp task called "sass:app" compiles all the `scss` files located in `src/sass/app` by piping the files through the **sass** gulp plugin, whose result is inturn piped through **autoprefixer**, whose result is piped through **insert** plugin (which inserts a banner and a charset statement) and the final modified output is written to the destination folder `public/css/app/raw/ltr` We have included autoprefixer so that you can keep your sass files clean by not using mixins or specific browser/vendor prefixes. Autoprefixer takes care of that for you and you should generally avoid hardcoding prefixes which are bound to get deprecated in the future.
Task 2 (sass:app:rtl) *Depends on Task (1)*	The second gulp task called "sass:app:rtl" depends on "sass:app". This task is only called if the argument **--rtl** is passed to gulp command (See: **Gulpfile.js > Basics** (/app/docs/gulpfile/basics) for more info). This task collects all the files generated from Task 1 above and flips them to RTL format and the final modified output is written to the destination folder `public/css/app/raw/rtl`
Task 3 (minifycss:app) and Task 4 (minifycss:app:rtl)	The third and fourth gulp task are run during production (if argument **--production** is passed to gulp command). The fourth task also depends on **--rtl** argument being passed to gulp command. These tasks minify the CSS generated from Task 1 and Task 2.
Task 5 (bless:app) and Task 6 (bless:app:rtl)	The fifth and sixth gulp task are run during production (if argument **--production** is passed to gulp command). The sixth task also depends on **--rtl** argument being passed to gulp command. These tasks take care of a very specific and lesser known IE9 related stylesheet bug. IE9 has hard limits on the number of selectors allowed in a CSS file. Once the limit is reached, IE silently fails and just ignores any further CSS leaving parts of your site totally unstyled. To fix this issue we use the awesome blesscss (http://blesscss.com/) library. The blessed files are written to `public/css/app/blessed/ltr` and < `public/css/app/blessed/rtl` . The blessed files are ordered (ex: main-blessed1.css, main.css). These stylesheets should be placed, in a descending order, before the closing of the `<head>` tag. Here is an example snippet:

```
  <link rel='stylesheet' type='text/css' media='screen,print' href='/css/app/blessed/lt
r/main-blessed1.css' />
  <link rel='stylesheet' type='text/css' media='screen,print' href='/css/app/blessed/lt
r/main.css' />
  <link rel='stylesheet' type='text/css' media='screen' href='/css/app/blessed/ltr/them
e.css' />
  <link rel='stylesheet' type='text/css' media='screen' href='/css/app/blessed/ltr/color
s-blessed1.css' />
  <link rel='stylesheet' type='text/css' media='screen' href='/css/app/blessed/ltr/color
s.css' />
  <link rel='stylesheet' type='text/css' media='screen' href='/css/app/blessed/ltr/font-f
aces.css' />
```