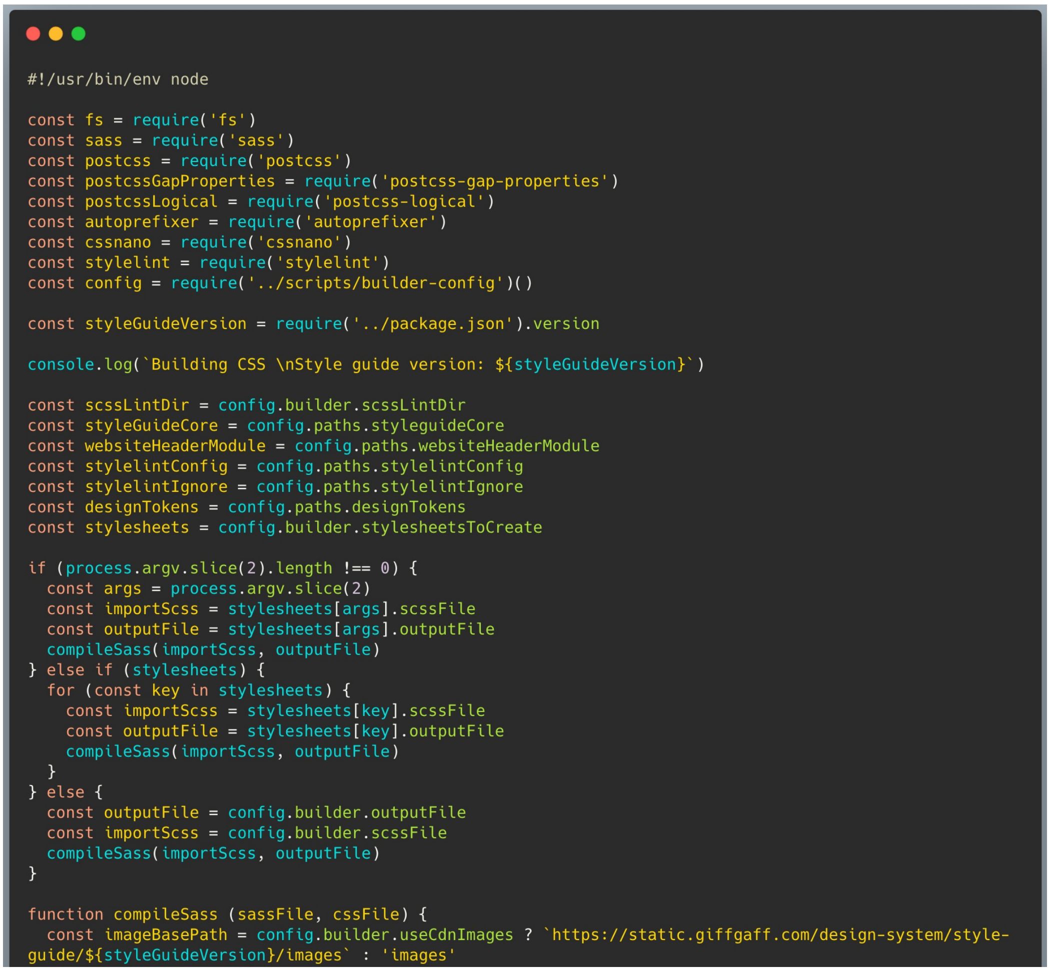
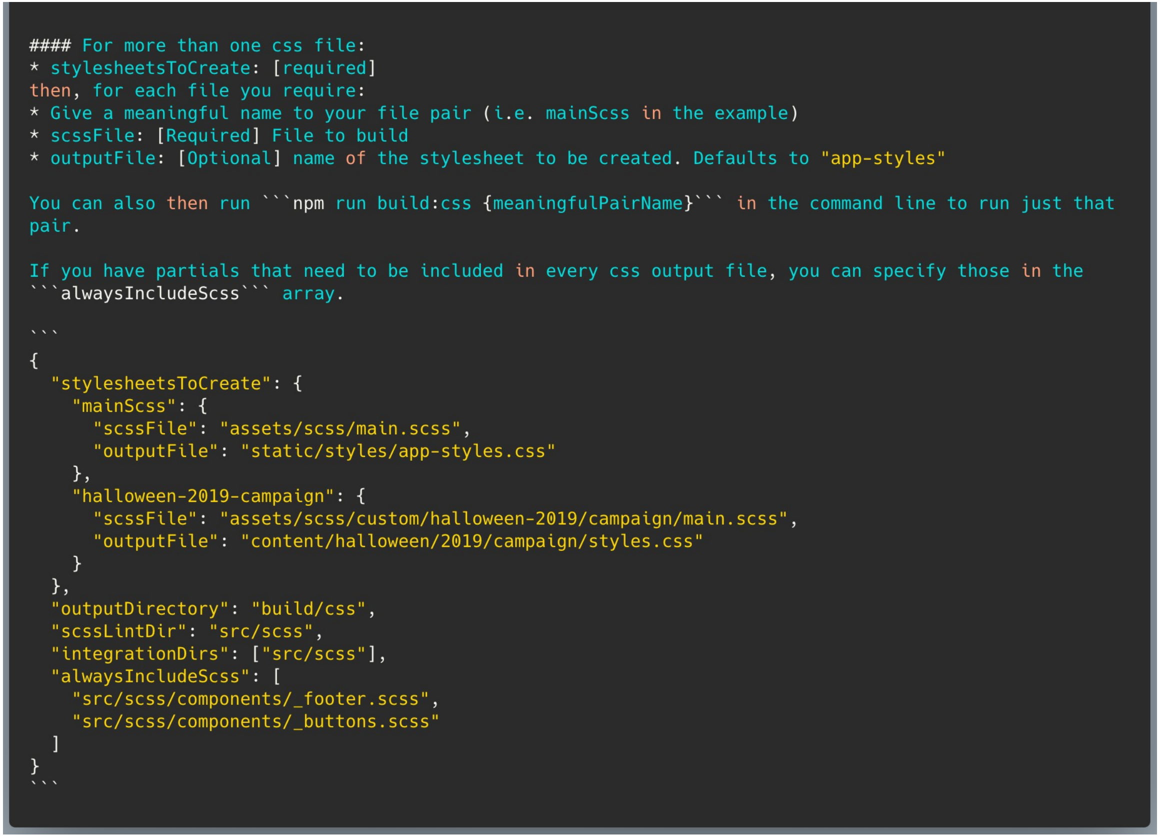


```
#!/usr/bin/env node

const fs = require('fs')
const sass = require('sass')
const postcss = require('postcss')
const postcssGapProperties = require('postcss-gap-properties')
const postcssLogical = require('postcss-logical')
const autoprefixer = require('autoprefixer')
const cssnano = require('cssnano')
const stylelint = require('stylelint')
const config = require('../scripts/builder-config')()

const styleGuideVersion = require('../package.json').version

console.log(`Building CSS \nStyle guide version: ${styleGuideVersion}`)

const scssLintDir = config.builder.scssLintDir
const styleGuideCore = config.paths.styleguideCore
const websiteHeaderModule = config.paths.websiteHeaderModule
const stylelintConfig = config.paths.stylelintConfig
const stylelintIgnore = config.paths.stylelintIgnore
const designTokens = config.paths.designTokens
const stylesheets = config.builder.stylesheetsToCreate

if (process.argv.slice(2).length !== 0) {
  const args = process.argv.slice(2)
  const importScss = stylesheets[args].scssFile
  const outputFile = stylesheets[args].outputFile
  compileSass(importScss, outputFile)
} else if (stylesheets) {
  for (const key in stylesheets) {
    const importScss = stylesheets[key].scssFile
    const outputFile = stylesheets[key].outputFile
    compileSass(importScss, outputFile)
  }
} else {
  const outputFile = config.builder.outputFile
  const importScss = config.builder.scssFile
  compileSass(importScss, outputFile)
}

function compileSass (sassFile, cssFile) {
  const imageBasePath = config.builder.useCdnImages ? `https://static.giffgaff.com/design-system/style-
guide/${styleGuideVersion}/images` : 'images'
```



~~~
#### For more than one css file:
* stylesheetsToCreate: [required]
then, for each file you require:
* Give a meaningful name to your file pair (i.e. mainScss in the example)
* scssFile: [Required] File to build
* outputFile: [Optional] name of the stylesheet to be created. Defaults to "app-styles"

You can also then run ```npm run build:css {meaningfulPairName}``` in the command line to run just that
pair.

If you have partials that need to be included in every css output file, you can specify those in the
```alwaysIncludeScss``` array.

```
{
  "stylesheetsToCreate": {
    "mainScss": {
      "scssFile": "assets/scss/main.scss",
      "outputFile": "static/styles/app-styles.css"
    },
    "halloween-2019-campaign": {
      "scssFile": "assets/scss/custom/halloween-2019/campaign/main.scss",
      "outputFile": "content/halloween/2019/campaign/styles.css"
    }
  },
  "outputDirectory": "build/css",
  "scssLintDir": "src/scss",
  "integrationDirs": ["src/scss"],
  "alwaysIncludeScss": [
    "src/scss/components/_footer.scss",
    "src/scss/components/_buttons.scss"
  ]
}
```
~~~