{{/\* style/GetMainCSS Process the main css stylesheet and return as resource @author @regisphilibert @context Any (.) @returns Resource @uses - func/style/GetResource \*/}} {{ $main\_style := dict }} {{/\* We prepare a slice of resources to be concatenated as one \*/}} {{ $assets\_to\_concat := slice }} {{/\* We add locale css files to the slice in the proper order \*/}} {{ range slice "\_tachyons.css" "\_code.css" "\_hugo-internal-templates.css" "\_social-icons.css" "\_styles.css" }} {{ with partialCached "func/style/GetResource" . . }} {{ $assets\_to\_concat = $assets\_to\_concat | append . }} {{ end }} {{ end }} {{ with partialCached "func/socials/Get" "socials/Get" }} {{ $socials\_rules := slice }} {{ range $service := . }} {{ with .color }} {{ $rule := printf ` .ananke-socials a.%s:hover { color: %s }` $service.name $service.color }} {{ $socials\_rules = $socials\_rules | append $rule }} {{ end }} {{ end }} {{ with $socials\_rules }} {{ $socials\_rules = delimit . "" }} {{ $socials\_css := $socials\_rules | resources.FromString "ananke/css/generated\_socials.css" }} {{ $assets\_to\_concat = $assets\_to\_concat | append $socials\_css }} {{ end }} {{ end }} {{/\* We look for any custom css files registered by the user under `site.params.custom\_css and if found in the theme's css asset directory we (unless condition below) add to aforementioned slice \*/}} {{ with site.Params.custom\_css }} {{ range . }} {{ with partialCached "func/style/GetResource" . . }} {{ if eq .MediaType.SubType "x-scss" "x-sass" "scss" "sass" }} {{ if hugo.IsExtended }} {{/\* as we cannot concatenate styles of different types, we sass/scss to be transformed to css beforehand \*/}} {{ $assets\_to\_concat = $assets\_to\_concat | append (. | resources.ToCSS) }} {{ else }} {{ partial "func/warn" (printf "Processing of stylesheet %s of type %s has been skipped. You need Hugo Extended to process such files." .Name .MediaType.SubType) }} {{ end }} {{ else }} {{ $assets\_to\_concat = $assets\_to\_concat | append . }} {{ end }} {{ end }} {{ end }} {{ end }} {{ with $assets\_to\_concat }} {{/\* We proceed to concatenate the $assets\_to\_concat \*/}} {{ $style := . | resources.Concat "ananke/css/main.css" }} {{/\* We then use toCSS to add sourceMap and minify \*/}} {{ $options := dict "enableSourceMap" true "precision" 6 }} {{ $style = $style | resources.ToCSS $options | minify }} {{/\* We fingerprint in production for cache busting purposes \*/}} {{ if eq (getenv "HUGO\_ENV") "production" }} {{ $style = $style | fingerprint }} {{ end }} {{/\* We're ready to set returning variable with resulting resource \*/}} {{ $main\_style = $style }} {{ end }} {{ return $main\_style }}