

Optimize the size of no js websites

Yann Esposito

[2019-12-06 Fri]

One of the major problem with CSS and HTML is that they are highly dependent from each other. For example, if you want to minimize your CSS, you are still forced to use the same class names even if they are long. Because the HTML uses them. And the same problem arise when you want to minimize the size of your HTML files.

It means that if you want to minimize a full website you must take care at the same time of HTML pages as well as CSS pages. And this is totally impossible to achieve if JS is involved because there is always the risk the JS code generate class names to manipulate the DOM.

So here is a small script I wanted to write from a long time that do the following:

1. retrieve all class names in the HTML and in the CSS
2. create a map from those long names to shorter names
3. replace the class names in the HTML and CSS files.

So if you have multiple HTML files with:

```
<div class="long-org-class-generated-by-org-mode">...</div>
```

and CSS files with:

```
pre .long-org-class-generated-by-org-mode { ... }
```

Those will be replaced by something like:

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

and CSS files with:

```
pre .av { ... }
```

And thus removing many superfluous bytes.

In my personal website, I run this script after minifying my HTML and CSS with classical tools. And I still get up to 32% smaller HTML and 22% smaller CSS.

Many 25% smaller HTML if there are a lot of code, because org-mode use very long class names when generating the code.

Not bad for a very basic solution.

If you want to try it; here is the quick and dirty script I use:

```
#!/usr/bin/env zsh

webdir="_site"

retrieve_classes_in_html () {
    cat $webdir/**/*.html(N) | \
        perl -pe 's/class="?([a-zA-Z0-9_-]*)/\nCLASS: $1\n/g'
}

retrieve_classes_in_css () {
    cat $webdir/**/*.css(N) | \
        perl -pe 's/\.([a-zA-Z_-][a-zA-Z0-9_-]*)/\nCLASS: $1\n/g'
}

classes=( $( {retrieve_classes_in_html; retrieve_classes_in_css} | \
    egrep "^CLASS: [^]*$" | \
    sort -u | \
    awk 'length($2)>2 {print length($2),$2}' | \
    sort -rn | \
    awk '{print $2}' )

chr() {
    [ "$1" -lt 26 ] || return 1
    printf "\\$(printf '%03o' $(( 97 + $1 )))"
}

shortName() {
    if [ "$1" -gt 25 ]; then
        print -- $(shortName $(( ( $1 / 26 ) - 1 ))$(shortName $(( $1 % 26 )))
    else
        chr $1
    fi
}

i=0;
typeset -A assoc
for c in $classes; do
    sn=$(shortName $i)
    print -- "$c -> $sn"
    assoc[$c]=$sn
    ((i++))
done
```

```

htmlreplacer=''
cssreplacer=''
for long in $classes; do
    htmlreplacer=$htmlreplacer's#class=("${?}"${long})'#class=$1'${assoc[$long]}'#g;'
    cssreplacer=$cssreplacer's#\.'${long}'#.'${assoc[$long]}'#g;'
done

sizeof() {
    stat --format="%s" "$*"
}

for fic in $webdir/**/*.{html,xml}(N); do
    before=$(sizeof $fic)
    print -n -- "$fic ($before)"
    perl -pi -e $htmlreplacer $fic
    after=$(sizeof $fic)
    print -- " => $after [$( ((before - after) * 100) / before )]"
done
for fic in $webdir/**/*.css(N); do
    before=$(sizeof $fic)
    print -n -- "$fic ($before)"
    perl -pi -e $cssreplacer $fic
    after=$(sizeof $fic)
    print -- " => $after [$( ((before - after) * 100) / before )]"
done

```

A few remarks:

- to prevent doing the work twice, the script only takes care for classe names longer or equal to 3 chars. (`awk 'length($2)>2 {print length($2),$2}'`). As consequence take care that your website does not use class name shorter than 3 chars otherwise it could mess with your css.
- The script do not change ids because those can be used for anchors and thus can be part of public URLs.
- The script replace the classes with the longest name first to prevent bug if one class name is a prefix of another one.
- We generate a long perl script to launch perl just once, this make the full find and replace way faster.

Of course this could be improved by providing the shortest name to the most used classes, and also by using a better `shortName` function that could use more chars. But just this quick and dirty script already does a better work than existing methods that do not take into account all the CSS and HTML files.