CSS CODE QUALITY

Analyze URL | Analyze File | Analyze CSS input

File to analyze

CHOOSE FILES style.css

style.css **(1.91 kB)**

```
/* Reset css  */
* {
    margin: 0;
```

Prettify CSS?

Prettifying makes inspecting the CSS easier, but very slighty changes the numbers.

ANALYZE CSS

MAINTAINABILITY	COMPLEXITY	PERFORMANCE
100	100	97

0 - 50 50 - 80 80 - 100

Score breakdown

Show only metrics for (•) All ( ) Maintainability ( ) Complexity ( ) Performance

Declaration Duplications — 37.0% Declaration duplication

To keep filesize at a minimum, Declarations should not be repeated too often. A lot of duplicated Declarations are a sign that something could be abstracted away or pieces are possibly obsolete and need a cleanup.

Scoring high here means a lot of Declarations (combinations of the same Property and Value) are duplicated across the CSS.

```
/* The same declaration repeated several times */
.warning {
  font-size: 14px;
}

.small {
  font-size: 14;
}

.footer {
  font-size: 14px;
}
```

Category: performance

Avoid `@import` — 0 Rules

Remove empty RuleSets — 0 empty RuleSets

Avoid many Selector code duplications — 13.6% Selector duplication

Keep filesize low — 1.71 kB

Limit CSS comments — 137 B

Limit embedded content — 0 B

Avoid many Source Lines of Code — 78 Source Lines of Code

Keep average Selectors per RuleSet low — 1 Selectors per RuleSet

Keep average Declarations per RuleSet low — 2.45 Declarations per RuleSet

Avoid many Selectors in a single RuleSet — 1 Selectors at most

Avoid many Declarations in a single RuleSet — 5 Declarations at most

Avoid larger than common SelectorLists — 1 Selectors per RuleSet (most common)

Avoid larger than usual Declaration Blocks — 1.50 Declaration(s) per RuleSet is most common

More than most common Selector Complexity — 0% of Selectors complexities more complex than most common

Avoid higher-than-usual Selector Specificity — 0% of Selectors more specific than most common Specificity

Avoid complex selectors — 1 Complexity points at most

Keep average Selector Complexity low — 1 Complexity points

Avoid ID Selectors — 0% ID Selectors

Keep `!important` usage low — 0% !importants

Raw CSS

Copy CSS

```
/* Reset css  */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {
    font-family: Verdana, Geneva, Tahoma, sans-serif;
    font-size: 10pt;
}
/* css Header  */
.Header {
```

Report JSON

Copy JSON


```
{
  "score": 0,
  "violations": [
    {
      "id": "DeclarationDuplications",
      "score": 3,
      "value": 0.37037037037037035
    }
  ],
  "passes": [
    {
      "id": "Imports",
      "score": 0,
```


The best way to analyze your CSS is to use the CSS analyzer, but if you're in a hurry or if you want an opinionated tool, then you can use this CSS Code Quality analyzer. It will use the output of the CSS analyzer to run a couple of checks and turn that into a set of recommendations for your CSS. It will focus on Performance, Maintainability and Complexity and score each one of them between 0 and 100 points. Think of it like PageSpeed Insights, but for CSS.