

Theory: Comments

 5 minutes

0 / 5 problems solved

Skip this topic

Start practicing

1670 users solved this topic. Latest completion was 15 minutes ago.

In CSS files it is often necessary to structure code sections or explain what the written styles do. A great way to do that are **comments**. A **comment** in CSS is an arbitrary text that is purely informative and does not affect the visual display of the site. Comments are not displayed on web pages in the browser.

Well-added comments can make your code more understandable. This is especially important for projects developed by large groups of people.

§1. Syntax

The syntax of CSS comments is quite simple:

```
1  /* Any text */
```

Any text placed between `/*` and `*/` will be considered a comment. The text of the comment can be placed either on one line or several.

Let's look at specific examples of how and when comments are used.

§2. One-line comments

One-line comments, as follows from the name, occupy just one line. Please take a close look at the syntactic features of one-line comments:

```
1  p {
2    color: blue; /* the color of the text paragraph is blue */
3  }
```

The text of the comment will not be seen by the visitors of your web page. With the aid of such notes, you will quickly remember what a certain code fragment is intended for – even if it's been a while since you last saw it! Comments can also serve as hints for other developers who will work with your code.

§3. Multi-line comments

Comments that take up several lines are called **multi-line**. Consider their syntax:

```
1  /* The styles given
2    to the paragraph
3    with the text
4  */
5  p {
6    color: blue;
7    background-color: black;
8  }
```

Such comments are useful for explaining particularly complex pieces of code or temporarily disabling large code sections. Also, multi-line comments located at the beginning of a file are quite useful when you need to specify copyright information.

§4. Disabling code fragments

Another important role of comments is temporary disabling parts of code. Consider the following example:

Current topic:

Comments

Stage 2

Topic depends on:

✗ Basic syntax

Stage 2

Topic is required for:

Google Fonts

Stage 2

Colors

Stage 2

Border

Stage 2

List style

...

Opacity

...

Absolute units

Stage 2

Relative units

Stage 2

Visibility

Stage 3

Table of contents:

[↑ Comments](#)

[§1. Syntax](#)

[§2. One-line comments](#)

[§3. Multi-line comments](#)

[§4. Disabling code fragments](#)

[§5. Conclusion](#)

[Feedback & Comments](#)

```
1  /*
2  h1 {
3      color: #33A1D9;
4      border-top: 2px solid;
5      font-size: 3em;
6  }
7  */
```

In this case, no styles will be applied to the `h1` element, but we can make them "visible" to the browser again if we delete the comment symbols. This process is called **uncommenting**.

Temporary code disabling is a great strategy when it is difficult to locate an error. It's inconvenient to just remove the code because you may need to restore it, so comment/uncomment is a great solution.

\$5. Conclusion

Comments are really useful and hence they shouldn't be neglected. However, keep in mind that the more comments you add to your code, the bigger it gets, which affects the loading speed. Use comments wisely, and you will write the most understandable and readable code!

Oh, and by the way, on Hyperskill we often use comments to explain our examples to beginners!

 Report a typo

159 users liked this theory. **1** didn't like it. What about you?



Start practicing

[Comments \(3\)](#)

[Hints \(0\)](#)

[Useful links \(0\)](#)

[Show discussion](#)