

CSSDOC First Public Draft

Author: Tom Klingenberg - www.lastflood.com

Version: 0.2.5

Date: 2007-11-03

Table of Contents

CSSDOC First Public Draft.....	1
Introduction.....	2
Main Elements.....	3
DocBlock.....	3
DocBlock Examples.....	3
Comment Types.....	3
File Comment.....	3
Section Comment.....	4
Standard CSS Comment.....	4
Tags.....	4
Tags.....	5
Tag Values.....	5
File Comment Tags.....	5
Section Comment Tags.....	5
Tags related to CSS Hacks.....	5
Application defined Tags.....	5
Parser and Control Tags.....	5
Tags Reference.....	7
Appendix.....	12
Selectors that look like Tags.....	12
Tags used by Third Parties.....	12
Todo.....	13

Introduction

CSSDOC is an adoption of the well known JavaDoc / DocBlock based way of commenting source code. CSS writer can use it to beautify their cascading style sheet (CSS) files and to find a convention they can refer to while sharing files with other writers or designers. There are many arguments for commenting source code especially in larger projects. These arguments are not part of this draft.

This Draft will describe the elements CSSDOC consists of and acts as a reference to a first set of tags that were developed in summer 2007. Feedback to this Draft is welcome.

Tom Klingenberg <tklingenberg@lastflood.com>

Main Elements

The Main Elements CSSDOC consists of DocBlock Comment-Blocks defining a CSSDOC Comment. A CSSDOC Comment can be of one of two types: file comment and section comment. Within each DocBlock certain Tags can be used to put structured data into a DocBlock. See Section %% 3.) for more information about Tags.

DocBlock

These are comments spanning multiple lines written in a specific convention. They start with a "/*" character sequence and are followed by one or more nicely intended lines beginning with a "*". The DocBlock is closed on the last line with the "*/" character which has to be intended as well.

Inside this "Block marker" there is space for a Title (normally the first line) and a human readable Description, which can follow separated by an empty line in the DocBlock. The *Larger DocBlock example* shows this.

DocBlock Examples

```
/**
 * DocBlock Comment
 */
```

Example: Small DocBlock example

```
/**
 * Larger Comment
 *
 * This DocBlock comment is even longer an contains
 * even more Text.
 */
```

Example: Larger DocBlock example

Comment Types

CSSDOC uses DocBlock Comment Blocks for certain types. Each CSS File must have at least one comment Block on top which is the *File Comment* Block. It can be followed by any amount of *Section Comment* Blocks.

File Comment

The *File Comment* DocBlock is for the main description of a CSS-File. It is human readably written down in plain text as well as with structured Tags for File-Metadata. This can be something as simple as the Authors Name as well as some more practicall stuff like Version, a certain Style the File is part of etc. .

The File Comment DocBlock should be placed on top of the file. It can be preceded by the

@charset rule¹ which needs to be on the very top of each CSS file.

Section Comment

A *Section Comment* opens a Sections in a CSS File that can be used to structure CSS code. The meaning of section is quite broad, there is no specified meaning of what a section is by this specification at all. Common sections in CSS are: Reset, Layout, Typography etc. but each CSS author does this mainly on its own.

%%Illustration

Standard CSS Comment

A *standard CSS Comment* is - as the name says - a standard CSS Comment by the CSS specification. These comments can be used to put additional comments into CSS files, the CSSDOC specification does not limit its usage.

Tags

Tags are one-word-identifiers that start with the @-symbol. They are placed inside DocBlock comments.

```
/**
 * Comment with Tags
 *
 * This DocBlock comment has tags.
 *
 * @author Mina Margin
 * @version 1.0
 * @see http://mina.blogsaroundthe.net/
 */
```

Example: DocBlock example with Tags

Tags are used at the end of a DocBlock Comment after the Title and the human readable description. A Tag can have a value which should be indented properly if the preceding or following line contains a Tag and Value pair as well.

Tags have a specific meaning and are machine-readable. Some Tags must be used on a per File basis, some Tags are optional and some Tags can only be used inside the one or other Comment-Type (File or Section). The name, meaning and location of each Tag is written down in the Tags Reference.

¹ See <http://www.w3.org/International/questions/qa-css-charset>

Tags

Tags can be used for various meanings. General purpose tags mainly go inside the *File Comment* DocBlock. Those related to parts of the document into sections and sometimes a whole section is defined only to separate a complicated CSS-Hack from well written and easy readable layout rules.

There is a set of tags that is designed for CSSDOC related parsers and applications as well as a set that can be used by Authors of web applications to place meta data into CSS-Files.

Tag Values

A value of a Tag is generally Text encoded in the encoding of the file. To let parsers do a better work, a subset between all theoretical files is a must. A De-Facto Standard for this is might be 7-bit ASCII. But even this set is too broad, so Identifier Values should only contain [0-9a-zA-Z] and the special characters \$-_.+!*'() , and SPACE (asc(32)). This is mainly compatible with the encoding of URLs² which can be used as well including their special characters. Another Idea is to take the specs of a CSS ID because any CSS-Author knows what's that. ID, IDREF or Name: [a-zA-Z][a-zA-Z0-9\-_\:\.]*³ [Edit+: This section needs a more clear specification, maybe a charset section is needed as well and take a look at value-types (URL; The misc. Identifiers and Dates etc.) which can be listed as well.]

File Comment Tags

Tags that are specially designed for the *File Comment* DocBlock do offer information related to the file. Basically this is information about the author (@author) as well as version or licensing information (@version, @revision, @license).

Section Comment Tags

Tags for the *Section Comment* DocBlock are mainly to define the identifier of the section (@section) that can be used once per file.

Tags related to CSS Hacks

Hacks and Workarounds is CSS written in such a way that only specific browsers will render it or that will circumvent parsing or display bugs of specific browsers. It's a complex topic in CSS and CSSDOC will reflect it with some tags designed for this CSS mastery.

Those Tags are @workaround and @bugfix.

Application defined Tags

Application developers can use CSSDOC to put data into CSS-Files. To not affect other applications, an application defines it's namespace with the @appdef Tag first. That name can be used in own @app-appname-name Tags later on.

Parser and Control Tags

Parser and Control Tags will offer the possibility to leave information for a CSSDOC parser to skip

² See <http://www.rfc-editor.org/rfc/rfc1738.txt>

³ See <http://www.w3.org/TR/html401/types.html#h-6.2>

specific parts of a CSS-File because complicated hacks are used that will affect strict-comment-parsing. This is a section that is currently worked on and there isn't a specific concept to offer right now. This will become important and a first Draft should contain such Tags and explain their usage.

Idea: A simple `@cssdoc` tag can fulfill this job:

```
/* @cssdoc parsingoff */  
...  
/* @cssdoc parsingon */
```

This tag could be used for other controljobs as well. *parsingoff* signals a CSSDOC parser to stop parsing right after the current comment. The parser will then stop parsing CSS and comments will skip all text until the next `@cssdoc parsingon` tag. The parser will then backtrack the beginning of the comment which contains that character sequence and will switch back to CSSDOC parsemode at the Beginning of the comment or commentblock.

```
/**  
 * @css-for      Internet Explorer 5  
 * @workaround  Faux Box Modell  
 * @valid       false; This is a IE parser bug for comments  
 * @cssdoc      parsingoff  
 */  
.box {width:280px; padding:20px;} /* * /.box {width:220px;  
padding:20px;} */
```

```
/**  
 * @section     plain, errorneous texts  
 * @cssdoc      parsingon  
 * @cssdoc      parsingoff  
 */
```

This is not valid CSS at all but it will never be a problem for a browser to handle this file nor for a strict CSSDOC parser.

```
/* @cssdoc parsingon */  
...
```

The Example above displays the flexibility of this Idea of CSSDOC parser control. Two Non-Standard CSS-Parts can follow each other and having Metadata. Looks reasonable to me, but before continuing here, it should make sense to take a look at TidyCSS or similar tools for a better precision of this.

Tags Reference

- **@affected**

- Location: *File Comment, Section Comment*
- Usage: Name a Browser version for a Bugfix or Workaround.
- Value: Browser and Version, Or a Group of them just written down so that it's readable.
- Example:

```
1. /**
   * My CSS Workaround
   *
   * @workaround wrong border display
   * @affected Opera v3 to v6.1 (Windows)
   */
```

- **@appdef**

- Location: *File Comment*
- Usage: Define application based Tag-Namespace
- Value: Lowercase-String of a new Tag-Namespace Definition.
- Example:

```
1. /**
   * My CSS File
   *
   * @appdef myapp
   */
```

- **@app-**

- Location: *File Comment*
- Usage: Tag with a previously defined Tag-Namespace. Can be used by applications to place Metadata into CSS-Files. After the first Space you can leave the Name of the Application.
- Value: Defined by Application
- Example:

```
1. /**
   * My CSS File
   *
   * @appdef myapp My Application
   * @app-myapp-theme Kubrik 3D
   * @app-myapp-used True
   */
```

- **@author**

- Location: *File Comment*
- Usage: Document the main Author(s) of a file
- Value: String containing the Name of the Author. Probably Email as well. For multiple Authors use multiple Tags.
- Example:

```
1. /**
   * My CSS File
   *
```

```
* @author Mina Margin <mm@example.com>
* @version 1
*/
```

- **@bugfix**

- Location: *File Comment, Section Comment, Standard CSS Comment*
- Usage: Documentation of CSS Bugfixes for Browsers
- Value: *None* or a short Description of the Bugfix. A Bugfix is CSS to circumvent Display- or Rendering-Bugs of a Browser.

- Example:

1.

```
/* @bugfix Global fix of the Italics bugs in IE 5.5
and IE 6 */
```
2. Edit+ Full Example see Appdef @vaild yes no

- See: **@css-for**

- **@colordef**

- Location: *File Comment*
- Usage: Definition of a Color.
- Value: Colordef String consisting of a standard CSS color-value and the name of a color (defined by the author) and a more generic usage identifier followed by a semicolon – if applicable.

- Status: Unstable

- Example:

1.

```
/**
 * ...
 * @colordef rgb(24,59,68); Dark Satin; Lighter Shadow
 * @colordef rgb(12,47,56); Darker Satin; Shadow
 * @colordef rgb(255,0,0); Red
 * ...
 */
```

- Discussion: The colordef tag was created to enable a CSS writer to systematically document used colors. If consistently used, a script or tool should be able to verify used colors and to change colorsets.

- **@copyright**

- Location: *File Comment*
- Usage: Document Copyright Owner of a File.
- Value: Copyright String
- Example:

1.

```
/**
 * ...
 * @copyright Copyright(c) 1992, 2006-2007 by My
Little Big Corp Inc.
 * ...
 */
```

- **@css-for**

- Location: *File Comment, Section Comment*
- Usage: Related to **@bugfix** naming which Browser reads this Bugfix-CSS
- Value: Name of fixed Browsers

- Example:


```
1. /**
   * ...
   * @css-for IE 6 Win32.
   * ...
   */
```

- See: `@bugfix`

- `@date`

- Location: *File Comment*
- Usage: To Specify the File-Date.
- Value: Date of File, ISO 8601⁴ should be preferred. If this is unfitting in bigger, international Projects, Coordinated Universal Time (UTC)⁵ is suggested. Might contain a placeholder for versioning systems as well (Subversion etc.). *[Edit+: Section about Date Values?!]*

- Example:


```
1. /**
   * ...
   * @date 2007-07-24
   * ...
   */
```

- See: `@lastmodified`, `@version`

- `@debug`

- Location: *File Comment, Section Comment*
- Status: Obsolete, was once Bugfix related
- See: `@bugfix`

- `@group`

- Location: *File Comment, Section Comment, Standard CSS Comment*
- Status: Third Party Tag, used by CSSEdit
- See: Appendix: Tags used by Third Parties

- `@lastmodified`

- Location: *File Comment*
- Usage: To Specify the File-Date.
- Value: Timestamp of the last Modification. See `@date` for format.
- Example:

```
1. /**
   * ...
   * @lastmodified 2007-07-24 12:34
   * ...
   */
```

- See: `@date`

- `@license`

- Location: *File Comment*
- Usage: To Specify the Licence the File is released under.

4 See http://en.wikipedia.org/wiki/ISO_8601

5 See <http://en.wikipedia.org/wiki/UTC>

- Value: License Identification String. This might be a URL followed by a human readable description.
- Example:


```
1. /**
   * ...
   * @license
   http://creativecommons.org/licenses/by/2.0/ Creative
   Commons Attribution 2.0 Licence
   * ...
   */
```

- @link

- Location: *File Comment*
- Usage: To Name a Hyperlink that stands for this File. Example: If it is part of a Theme, this could be the Homepage of the Theme etc..
- Value: URI
- Example:


```
1. /**
   * ...
   * @link http://www.example.com/
   * ...
   */
```

- See: @author

- @media

- Location: *File Comment, Section Comment*
- Usage: Specify a Media Descriptor for a File or Section
- Value: CSS Media Descriptor
- Status: Obsolete; CSS Selector should be used instead; Maybe this Tag will be renamed to @formedia
- Example:


```
1. /**
   * ...
   * @media print
   * ...
   */
```

- See: Appendix: Selectors that look like Tags
- Discussion: This conflicts with a CSS Selector

- @note

- Location: *File Comment, Section Comment*
- Usage: Drop a Note
- Value: Text
- Example:


```
1. /**
   * ...
   * @note this is not for production use
   * ...
   */
```

- See: @todo

- **@package**
 - Location: *File Comment*
 - Usage: Associate one or multiple files to a package
 - Value: Package Identifier.
 - Example:

```

1. /**
   * ...
   * @package      my-super-package
   .* @subpackage  my-super-interactive-package
   * ...
   */

```
 - See: **@version**
- **@ref**
 - Location: *File Comment, Section Comment*
 - Status: Unstable; Might become Obsolete, see **@see**
 - Usage: Reference something: A File, a Website, another section
 - Value: Reference
 - See: **@see**
- **@revision**
 - Location: *File Comment*
 - Usage: Documentation of the File Revision
 - Value: Revision Number
 - See: **@version**
- **@section**
- **@see**
- **@since**
- **@site**
- **@style**
- **@subpackage**
- **@tested**
- **@todo**
- **@valid**
- **@version**
- **@workaround**

Appendix

Selectors that look like Tags

CSS has some selectors that look like tags because they start with the @-symbol as well. @media or @charset are the two most well known ones. Those are not used within comments so that a parser won't run accidental over them if it parses the source code correctly. This is important because standard CSS comments might be used to comment CSS out. A well made CSSDOC parser should consider this case as well.

Tags used by Third Parties

The @group Tag is used by a CSS editing software CSSEdit⁶. It uses the @group Tag to create a kind of section that is reflected in an outline afterwards.

⁶ CSSEdit for Macintosh by MacRabbit software; <http://macrabbit.com/cssedit/>

Todo

This first Draft is in progress, some parts of the document need some more work.

- Complete Tags Reference
- Include Illustration
- Concept for Parser and Control Tags has to be defined and written down
- Each Section in Chapter Tags should contain at least one Example