CSScribe 3.0.9 cheatsheet

Command types

CSScribe is a *markup language.* Most of what you'll write is actual text that will be shown in the final document, but there are also commands which let you decide how to format the final file.

We will make a distinction between text (information you type) and commands (short statements you write to define how the text will be formatted).

Text is text, there's really nothing more to say about it. But we have four different types of commands:

- Classes: big sections that are independant of each other and of the rest of the document, like image sections.
- Editors: things that will only be seen inside the source code, like comments, but will not affect the final document.
- Modifiers: commands that define the aspect of words or add elements, like adding a **bold** word.
- Tags: custom tags that are carried over to the *HTML* temporal file and can define how other elements look or interact based on the `style.less` file.

The relationship between them is simple:

- Most classes have text that can include modifiers.
- Some modifiers can also have other modifiers inside to add up the effect. Like combining **bold** and *italic* to get ***bold italic.***
- Editors are completely independant from the document but can help format it. The newline character is compiled into a new line without the need to make a new line inside the editor (useful in large documents, like this one).
- Tags have special functions that can format further the document. An example is the index tag, will will align a number to the right and remove ordered list numeration inside an index.

Now we'll dive deeper into specific commands.

Classes

As of this version, there are four class types.

Code

The first one is the code class. You can insert code into it and will be formatted this way:

```
{
    "variable1": 2,
    "variable2": 3
}
```

To include some code like this one in your `.cssc` file, input

```
/code-json
{
    "variable1": 2,
    "variable2": 3
}
/code
```

You can replace *json* with any programming language, and it's usually written in lowercase. The code will automatically adquire color and syntax highlighting based on the language you choose (not yet for *CSScribe*).

Comment

There's also a side comment class. The side comment is used to specify some information in the document, but formatted in a different way to acknowledge that it's not part of the main content. Here's an example:

> This information is not as relevant, but I choose to include it here anyway and that's why I formatted it differently.

To input something like this, use the following code:

```
/> This information is not ... and that''s why I formatted it ...
```

You might notice I added two single quotes at the same time. It's not a mistake, as you'll see later.

Image

To add an image, first you have to download an image and create a folder named `images/` in the same directory the file is.
Then, rename the file to be `1.png` for the first, `2.png` for the second...
Also make sure your working folder in *Visual Studio Code* is the one that contains the

`images/` folder directly.
To include an image, use the following code:

```
/img /br A picture of a cat. /br Image 4.
```

I also included an *HTML* break tag to make it appear in a new line so I could add a footer and image number.

Math

The math class allows you to input any mathematical formula. It is based on *LaTeX* and thus it's really powerful. I recommend you to input mathematical formulas in this website first so you know how they'll look on paper. Here's an example:

$$3x + 2 = -1$$

To input the previous equation, you'd have to type:

```
/math
3x + 2 = -1
/math
```

Inserting complex mathematical formulas requires extensive *LaTeX* knowledge so I suggest playing with the website provided for a while.

Editors

There are three types of editors.

Comment

The comment editor is different from the class editor. Whilst the comment class outputs a message, the comment editor is like a traditional comment in a programming language, something that isn't shown to the user and it's meant to be in the code for other developers.
Here's how to input a comment editor:

```
#> THIS IS A COMMENT <#
```

It is also a multi-line comment.

Newline

The newline allows you to input a new line in the *Markdown* code without having to input it on yours by replacing it with `/n`. That allows you to write more compact text inside *VSC.*

> Tip: if you're not sure whether to use or not the newline, use it, you won't lose anything.

Scape

The scape character is a character which cancels any command if it's on it's way.
Imagine we wanted to write `/n` in the final document without it being transformed into a new line. We would input the not sign (`U+00AC`, `&not;`, I can't input it since I'm writing the *CSScribe* guide inside *CSScribe*) between the `/` and the `n` to avoid the conversion.

> In case you are curious, I'm inputting CSScribe commands inside CSScribe source code to make the `.pdf` you're reading using the scape character. That's the reason why I can't write it.

Actually, you can type it via math modifier (¬).

Modifiers

There are 15 types of modifiers.

Bold

It's quite self-explanatory, it just makes some text be **bold.**
To do so, input the following code:

```
I want the following sentence to be bold: 'this is bold.'
```

The scape character to input just one single quote is writing two single quotes (as if you wanted to input something bold) together without anything in between.

Brochet

This is kinda not a modifier, but it is colored anyway, I might remove this soon because it offers no functionality. Just literally input ( ) or [ ] or { } and that's it.
When I started coding this markup language, a modifier was anything that was colored and that's why it's still in this list.
And there's no support for scape character because why would you want to add the brochet then?

Checkbox

Adds a small checked checkbox:

☑ Done!

The code is:

```
/X Done!
```

Code

The code modifier is not the same as the code class. The code class creates a separate big area whilst the code modifier adds `small inline code`.
Here's how to use it:

```
This is a normal sentence but the following /cdword/cd will be ...
```

h1

Adds a big title, like the one at the start of this article. The code behind it is:

```
/h1 *CSScribe* 3.0.7 cheatsheet
```

You might also realize that there are some weird dots around *CSScribe.* It's italic.

h2

Literally the same as *h1* but replace the 1 with a 2. You should also include a separator like this

---

after the *h2* with the following code:

```
---/n
```

h3

The same

Italic

The weird dots. There's no scape character either because I haven't seen anyone use those characters, I literally chose the least used character in the keyboard.

Math

The math modifier is not the same as the math class because this one is inline. It's the same but inline.
Code used 4 example.

```
Code user /_4/_ example./n
```

Ordered

Adds an ordered list. Like this:

1. Element 1
2. Element 2

Code behind:

```
    o Element 1
    o Element 2
```

Strikethrough

Literally ~~strikesthrough~~ some text:

```
Literally /-strikethroughs/- some text:/n
```

Subscript

I don't use it a lot, the *LaTeX* subscript and superscript version is better.$_1$

Code behind:

```
... version is better./s1/s/n
```

Superscript

Same [2] things. Code behind:

```
Same /S2/S things.
```

Unordered

Like the ordered but without the numbers:

- Element.
- Another element.

Code behind:

```
    u Element.
    u Another element./n
```

Voidbox

Like the checkbox but unchecked:

☐ It is void.

Code behind:

```
/O It is void./n
```

Tags

There are three types of tags:

Break

It's quite literally the *br* tag in *HTML*.

Code:

```
/br
```

Index

It's used in index sections:

Code behind:

```
    o Chapter 1 /In1/in
    o Chapter 2 /In23/in
    o Chapter 3 /In44/in/n
```

pbba

Obsolete tag, please do not use it unless you really want to.

When you write an *h2* section, the default is to make a page break. *pbba* was designed to prevent this by writing it before the *h2*. I don't think it works anymore but feel free to use it. Code:

```
/pbba
```

That's everything.