COWTCHOOX

Getting started with Cowtchoox

This document will provides basic information on Cowtchoox.

The features evolve quickly, and the documentation doesn't, and there is a lot of bugs everywhere, so if something doesn't work, feel free to complain here.

Installation

Windows

Go here: https://github.com/Sergueille/cowtchoox/tags, and download the latest release. You should get a `.zip` file. Unzip it where you want Cowtchoox to be installed. You now have to add this directory to your PATH. To do this, you can follow these instructions .

Other OS

I have no idea how to build this for you. You should consider tell me how to do it or building it yourself form the source code.

Compiling a file

Create a text file with your favorite editor, and save it as `hello_world.cow`. It should contain the following:

```
<document>
    <head>
        <title>Hello world!</title>
    </head>
    <body>
        Hello world!
    </body>
</document>
```

Now, to compile the file, just call `cowtchoox hello_world.cow` in your command prompt. Cowtchoox will produce `out.pdf` and `out.html`. You can open `out.pdf` to see your beautiful document.

Basic syntax

First, learn how to write basic HTML. You just need a basic understanding. As you can see above, Cowtchoox looks a lot like HTML.

Document structure

```
<document>
    <head>
        <title>The title of the document</title>
    </head>
    <body>
        Contents of the document
    </body>
</document>
```

- The `<head>` tag contains metadata about your document, such as title, style, paper format...
- The `<body>` tag contains the contents of the document.

Differences with HTML

Comments

You can use HTML comments in Cowtchoox. (actually this is not yet implemented, please use other methods). Cowtchoox also supports *C-style comments*:

- Everything after `//` is ignored until the end of the line
- Everything between `/*` and `*/` is ignored, even across multiple lines

Escaping with `\`

To escape characters (tell the compiler that they are regular character and have no special meaning), use a backslash: `\`. It escapes any character. HTML escaping like `<` isn't supported.

Using CSS

To change the style of documents, we will use *CSS*. If you have no idea of what it is, `click here`. To tell the compiler to use a CSS file, include a `<css>` tag in the `<head>` of the document.

```
<document>
    <head>
        <title>The title of the document</title>
        <css>path to your file here</css>
    </head>
    <body>
        Bla, bla, bla ...
    </body>
</document>
```

You need to specify the path from the `.cow` file to the `css` file. For instance, if it is located in the same directory, just put the name of the file. You have to include the file extension.

Math

Cowtchoox have some special syntax to write math. There are several ways to create math:

- Use dollars for *inline* math: `$This is math$`
- Use double dollars for a big, centered block of math: `$$This is math$$`

Inside math, comments, HTML tags and everything else is still supported. All spaces in math are not displayed, you can add them manually by escaping them: `\` (yes there is a space after the backslash).

In math you can use **math operators**. The syntax is:

```
?operator_name {first argument} {second argument} ...
```

For instance, `?frac{1}{2}` gives $\frac{1}{2}$. If an argument is only one objet, you can omit the braces: `?frac 1 2`

Some operators have an **alias**. It allow them to be called in a shorted way. For instance, to show $\in$, you can use `?belongsto` or `€`. Aliases takes arguments exactly in the same way.
Some aliases are **infix**. That means they have two arguments, and that the first should be placed *before* the alias. Here for a fraction: `1 / 2` or an exponent: `2 ^ n`.

You can add braces (`{}`) to group things. (It will create additional `<div>`s)

Code blocks

To add a code block:

- `` `your code here` `` for an inline code block
- ``` ``your code here`` ``` for a big code block

Everything inside these will be *escaped*, that means that comments, backslash, etc will be displayed in the document.
To make a code block, but with regular parsing inside:

- `<code>your code here</code>` for an inline code block
- `<pre>your code here</pre>` for a big code block

Tips

To edit your documents quickly, you can open `out.html` in the browser. When recompiling, Cowtchoox creates `out.html` in less than a second, and then takes some time to produce `out.pdf`, so you just have to reload the page in your browser to see instant changes in the document. Note that some things may differ from the final pdf file. You will also be able to use the devtools to debug your CSS.