Skin Demo

Asciidoctor custom skin with dynamic dark/light switching

Introduction

This is a simple project that builds on the default Asciidoctor `.css` file, appending select customizations at the very end.

It also adds a button to the Table of Contents for dynamic switching between Light and Dark mode, using Javascript.

To see this skin in action with a lot more text, see the documentation for LEAF, a lightweight error-handling library for C++11.

Customizing the CSS

The first thing I wanted to do after extracting the default Asciidoctor `.css` was change the fonts. This requires a line to be inserted near the very top of the file, in this case:

```
@import
"https://fonts.googleapis.com/css?family=Anonymous+Pro|Istok+Web|Quicksand|Poiret+One"
;
```

This imports the four Google fonts used: Anonymous Pro, Istok Web, Quicksand and Poiret One.

Next I proceeded with hacking the CSS itself, but rather than just making random changes throughout the text, I appended all the changes at the very end. This is possible, because anything set in CSS can be overridden by targeting the same class or element later on.

Dynamic Switching between Dark and Light Mode

This is actually very simple to do with Javascript. Once I created the two `.css` files, `zajo-dark.css` and `zajo-light.css`, I used the `:stylesheet:` attribute at the top of the `.adoc` file to specify that I want `zajo-dark.css` used by default:

```
:stylesheet: zajo-dark.css
```

Adding the button to the Table of Contents requires a tricky hack. By default, Asciidoctor will use "Table of Contents" as the default title for the table of contents, but it allows us to customize it:

```
:toc-title: My Table of Contents
```

I have always thought that this is redundant, and used to turn it off like so:

```
:toc-title:
```

It occurred to me that I can use `:toc-title:` to inject the HTML I need to put up the button. This can be done like so:

```
ifndef::backend-pdf[] ①
:toc-title: pass:[<div style="float:right"><input width="32" height="32" type="image"
alt="Skin" src="./skin.png" onclick="this.blur();switch_style();return
false;"/></div>]
endif::[] ②
```

① We use `ifndef` in order to prevent the HTML being injected when Asciidoctor is producing a PDF.

② `pass` will pass everything between the `[` and `]` and inject it directly in the generated HTML output. We have a floating `div`, and a simple script that calls the `switch_style` function (see below).

Using `docinfo.html` to Inject Javascript

In Asciidoctor it is very easy to add anything to the HEAD section of the generated HTML file. All you have to do is create a file called `docinfo.html`. In this case, the contents of this file is as follows:

```
<link rel="stylesheet" href="./zajo-light.css" disabled=true> ①
<script>
function switch_style() ②
{
    var i, tag;
    for( i=0, tag=document.getElementsByTagName("link"); i<tag.length; i++ )
        if( tag[i].rel.indexOf("stylesheet")!=-1 && tag[i].href.includes("zajo-") ) ③
            tag[i].disabled = !tag[i].disabled; ④
}
</script>
```

① Add the `zajo-light.css` as a link. Note the `disabled=true`, this ensures that the `zajo-dark.css` will be used by default (it is introduced by the `:stylesheet:` attribute, see above).

② This is the Javascript function invoked by the injected button (see the previous section).

③ Enumerate all CSS links in the HTML file, and if we see one with a name that starts with `zajo-`...

④ ...toggle its `disabled` state.

Building

I've included a simple Makefile to generate the HTML using Asciidoctor and to copy the necessary files in the `out` directory (which then can be published). Of course, you must have `asciidoctor` installed on your system.

That’s All!