
CDK Documentation

Release v0.0.1

Simeon Franklin

Sep 27, 2017

Contents

1	About	3
2	Installing	5
2.1	For users	5
2.2	For contributors	5
3	Using <i>cdk</i>	7
4	Installing/customizing Themes	9
5	Selecting Filters, tweaking sizes	11
6	Other Resources	13
7	Design Justifications	15
8	Indices and tables	17

Contents:

CHAPTER 1

About

tl;dr - using *cdk* write documents in asciidoc and produce elegant single-file slidedecks as html.

cdk is a Python program to more easily stitch together [Asciidoc](#) and [asciidoc-deckjs](#) which allows asciidoc to create [deck.js](#) slide decks. This means you can author presentations in plain text in a markdown-like format and get well behaved attractive presentations in a single html file.

In addition cdk is opinionated. It uses many plugins from the deck.js ecosystem and provides a few custom plugins as well. It provides theming support. In general features are oriented around the kinds of things technical instructors need to do:

- showing and explaining lots of code
- organizing and navigating large slide decks
- minimizing the effort needed to make things look good

CHAPTER 2

Installing

For users

```
$ sudo pip cdk
```

Or if you don't have pip

```
$ sudo easy_install cdk
```

For contributors

Check the source out of github. Notice the *cdk/data* directory is empty - run

```
make getdata
```

To download the many (non pip-installable) external dependencies.

cdk/__init__.py is the setuptools entry point. You can run

```
python cdk/__init__.py --generate=sample.asc
```

to get the same command.

CHAPTER 3

Using *cdk*

cdk transforms asciidoc documents into single file html slide decks. You can generate a simple slide deck with

```
$ cdk --generate=sample.asc
```

The sample file might look something like:

```
= My Presentation

Author: @somebody

== My First Slide

Presentation software for engineers

[options="incremental"]
* Content should be simple but presentation stylish.
* Author slides in plain text (asciidoc)
* And play them in the browser.

== My Second Slide

Use the source!

[source,python]
----
>>> x = list("Simeon")
>>> x.lower()
"simeon"
----
```

Now run *cdk* on your input file to produce a single html file you can open in your browser.

```
$ cdk sample.asc
```

Open the resulting *sample.html* file in your browser! Hit the space bar to advance a slide or hit the “h” key to see what other commands are available to navigate through the deck.

Installing/customizing Themes

cdk has a themes directory that contains the default theme named “plain”.

A theme is just a directory with a .css file of the same name. For example a theme “foo” is just a directory named *foo* with a *foo.css* file in it.

cdk comes with a command to install a theme. All this means is putting a directory in a zip file - the contents will be extracted to the *cdk* theme directory. If you name your theme “plain” it will overwrite the default (but don’t do that!)

cdk also comes with a flag to pick a theme for a build or to set the default to always use a theme when building a slide deck.

So basically - you need to make a .zip with a theme. Let’s say you called the theme “newcircle”. You could install it and make it the default with

```
$ cdk --install-theme=newcircle.zip $ cdk --default-theme=newcircle
```

Note: In practice I may use *lessc* and some helper scripts to make editing .css easier and to allow packing of images into the .css. See the included twitter theme at <https://github.com/twitter/cdk/tree/master/cdk/custom/deck.js/themes/twitter>

For another example theme contributed by a third-party check out <https://github.com/thenewcircle/cdk-theme>

Selecting Filters, tweaking sizes

TODO

CHAPTER 6

Other Resources

Slides are asciidoc so be sure to read the [docs](#) or at least a [cheatsheet](#).

The resulting slide deck is provided by *deck.js* using the *asciidoc-deckjs* backend so read up you might want to read the [asciidoc-deckjs tutorial](#)

Design Justifications

I'm having some challenges writing a utility that mainly stitches together other programs. In particular my main goals are:

- ease of installation. I want to be able to tell people to just *sudo easy_install cdk*
- My dependencies are not dependable: *deck.js* isn't a python package at all and *asciidoc* isn't setup to be installed.
- I don't want to have a source tree full of thousands of files I'm not authoring.

Complicating things is the difficulty of getting *setuptools* to include non-python data files in a known good location. Currently I am:

- including a couple of package data directories *cdk/data* and *cdk/custom*. I'm not checking *cdk/data* into source control, the contents can be recreated by running *make getdata*.
- I'm using the *MANIFEST.in* file to include all the contents of *cdk/data* and *cdk/custom* in the *sdist* I'm building with *python setup.py sdist*.

CHAPTER 8

Indices and tables

- search