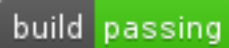

Copydoc

Release 1.0.9

Jun 12, 2018

Contents

1	copydoc	1
1.1	Table of contents	1



Like copytext, but for docs

1.1 Table of contents

1.1.1 Using Copydoc

Basic usage

Copydoc cleans and parses HTML from Google docs. Download the HTML version of a Google document and pass it as a string to the CopyDoc constructor:

```
with open('path/to/html') as f:
    html = f.read()
doc = CopyDoc(html)
```

Now you can print the parsed document:

```
print(str(doc))
```

Access parsed, BeautifulSoup object:

```
soup = doc.soup
```

Using named tokens

You can define simple key/value pairs in your docs, for example:

HEADLINE: Independent candidates gain in polls

FEATURED_GIF: <https://media.giphy.com/media/l3nWl5bhBoim7glNu/giphy.gif>

These key/values can be parsed out by passing a list to the Copydoc constructor:

```
tokens = (
    ('HEADLINE', 'headline'),
    ('FEATURED_GIF', 'featured_gif'),
)
doc = CopyDoc(html, tokens)
```

Now you can access the key/value pairs as attributes on the Copydoc object.

```
print(doc.HEADLINE)
```

This will print “Independent candidates gain in polls”.

Using with Jinja

The behavior of Copydoc has been designed to work nicely with Jinja.

Here’s a sample template snippet based on the doc from above:

```
<h1>{{ doc.headline }}</h1>



{{ doc }}
```

1.1.2 Changelog

1.0.9 - June 11th, 2018

Update setup.py to fix Pip 10 support

1.0.8 - June 9th, 2017

Fix empty doc test for python 3.x

1.0.7 - June 9th, 2017

Transfer ownership to NPR Viz Team account

1.0.6 - June 8th, 2017

Handle multiple formatting on the same text Use beautifulsoup decode instead of prettify Fix empty doc treatment. closes #8

1.0.0 - March 31, 2016

Initial release.