
htmlBuilder Documentation

Release 0.1.2

Jaime Vinas

Jul 08, 2017

Contents

1	Installation	5
2	How to contribute	7
3	Liscense	9

HtmlBuilder is a python library with the purpose of allowing you to render html by writing python code and also to make use of python features like comprehensions, clean syntax and object oriented design to its full potential.

A simple example

```
# import necessary tags and attributes
from htmlBuilder.tags import *
from htmlBuilder.attributes import Class, InlineStyle

# html tags are represented as classes
html = Html(
    # any tag can receive another tag as constructor parameter
    Head(
        Title(Text("A beautiful site"))
    ),
    Body(
        # tag's constructors can also receive attributes
        Class('btn', 'btn-success'), InlineStyle(background_color='red', bottom='35px'
        ↪'),
        Hr(),
        Div(
            Div()
        )
    ),
)
# no closing tags are required

# call the render() method to return tag instances as html text
print(html.render())

##### result #####

<html>
  <head>
    <title>A beautiful site</title>
  </head>
  <body style='background-color:red; bottom:35px;' class='btn btn-success'>
    <hr/>
    <div>
      <div></div>
    </div>
  </body>
</html>
```

A not so simple example

```
from htmlBuilder.attributes import Class
from htmlBuilder.tags import Html, Head, Title, Text, Body, Nav, Div, Footer, Ul, Li

# declare data
users = [
    {
        "name": "Jose",
        "movies": ['A beautiful mind', 'Red'],
        "favorite-number": 42,
    },
    {
```

```
        "name": "Jaime",
        "movies": ['The breakfast club', 'Fight club'],
        "favorite-number": 7,
    },
    {
        "name": "Jhon",
        "movies": ['The room', 'Yes man'],
        "favorite-number": 987654321,
    },
]

# functions can be used to handle recurring tag structures
def my_custom_nav():

    # these functions can return a tag or a list of tags ( [tag1,tag2,tag3] )
    return Nav(
        Class("nav pretty"),
        Div(Text("A beautiful NavBar"))
    )

html = Html(
    Head(
        Title(Text("An awesome site"))
    ),
    Body(

        my_custom_nav(), # calling previously defined function
        [Div(
            Class(f"user-{user['name'].lower()}"), # string formatting can be used to_
            ↪handle dinamic data
            Div(Text(user['name'])),
            Ul(
                [Li(Text(movie)) for movie in user["movies"]] # list comprehensions_
                ↪can be used to easily render multiple tags
            ) if user['favorite-number'] < 100 else Text("Favorite number is too high
            ↪") # python's ternary operation is allowed too
            ) for user in users],
            Footer(Text("My Footer")),
        )
    )
)

print(html.render())

##### result #####

# <html>
# <head><title>An awesome site</title></head>
# <body>
# <nav class='nav pretty'>
#   <div>A beautiful NavBar</div>
# </nav>
# <div class='user-jose'>
#   <div>Jose</div>
#   <ul>
#     <li>A beautiful mind</li>
```

```
#         <li>Red</li>
#     </ul>
# </div>
# <div class='user-jaime'>
#     <div>Jaime</div>
#     <ul>
#         <li>The breakfast club</li>
#         <li>Fight club</li>
#     </ul>
# </div>
# <div class='user-jhon'>
#     <div>Jhon</div>
#     Favorite number is too high
# </div>
# <footer>My Footer</footer>
# </body>
# </html>
```

Wait! ... Maybe that was too much as an starting example, but that was a bit of what htmlBuilder is capable of doing.

Other features include

- **Type safe tag rendering:** By default htmlBuilder validates that tags and attributes are used within valid parent tags.(for example an li can't be created outside an ul or ol tag)

For more detailed information checkout the *Source Code* <www.github.com/jaimevp54/htmlBuilder>

CHAPTER 1

Installation

Python 3.6 or higher is required to use htmlBuilder. If you meet this requirement just run

```
pip install htmlbuilder
```


CHAPTER 2

How to contribute

The whole source code is available on GitHub.

Feel to open an Issue or to submit a pull request.

CHAPTER 3

Liscense

The project is licensed under the MIT license.