Sphinx Inline Tabs

Pradyun Gedam

Aug 08, 2023

CONTENTS

Add inline tabbed content to your Sphinx documentation.

Add inline tabbed content to your Sphinx documentation.

INSTALLATION

This project is available on PyPI, and can be installed using pip:

```
pip install sphinx-inline-tabs
```

You'll also want to add the extension to `extensions` in `conf.py`:

```
extensions = [
    ...,
    "sphinx_inline_tabs",
    ...,
]
```

FEATURES

- **Elegant design**: Small footprint in the markup and generated website, while looking good.

- **Configurable**: All the colors can be configured using CSS variables.

- **Synchronisation**: Tabs with the same label all switch with a single click.

- **Works without JavaScript**: JavaScript is not required for the basics, only for synchronisation.

DEMO

One is an interesting number.

Two is the even prime.

Three is an odd prime.

Four is not a perfect number.

This is text after the tabs, which seems to flow right through, which avoids breaking the flow of the document. And, if JavaScript is enabled, the next set of tabs will be synchronised.

```
print(1)
```

```
print(2)
```

```
print(3)
```

```
print(4)
```

To know more, take a look at the *Usage* documentation of this project.

3.1 Usage

3.1.1 Basics

Tabs are created, using a `tab` directive. Consecutive `tab` directives create a single set of tabs.

```
This is text before the tabs.

.. tab:: One

   One is an interesting number.

.. tab:: Two

   Two is the even prime.

.. tab:: Three

   Three is an odd prime.
```

```
.. tab:: Four

    Four is not a perfect number.

This is text after the tabs, which seems to flow right through, which avoids breaking␣
↪the flow of the document.
```

This is text before the tabs.

One is an interesting number.

Two is the even prime.

Three is an odd prime.

Four is not a perfect number.

This is text after the tabs, which seems to flow right through. There is no visual difference for content that's within a tab vs outside it – adding tabs doesn't disrupt the flow of the document.

3.1.2 Multiple Tab Sets

It is possible to start a new set from a tab, by having some content between sets or by providing `:new-set:` option to the `tab` directive.

```
.. tab:: One

    One is an interesting number.

.. tab:: Two

    Two is the even prime.

This will break the tab set!

.. tab:: Three

    Three is an odd prime.

.. tab:: Four

    Four is not a perfect number.

.. tab:: Five
   :new-set:

    Five is a nice number.

.. tab:: Six

    Six is also nice.
```

One is an interesting number.

Two is the even prime.

This will break the tab set!

Three is an odd prime.

Four is not a perfect number.

Five is a nice number.

Six is also nice.

3.1.3 Rich tab labels

This is only possible with Markdown, as far as I know.

```
```{tab} Three
Three is an odd prime.
```


```{tab} Four (equal to $2^2$)
Four is not a perfect number.
```
```

Three is an odd prime.

Four is not a perfect number.

3.1.4 Code Tabs

The first code block in a tab content will "join" with the tabs, making things fairly clean for language snippets and OS-based command suggestions.

```
print("Hello World!")
```

It's pretty simple!

```cpp
#include <iostream>

int main() {
  std::cout << "Hello World!" << std::endl;
}
```

More code, but it works too!

```
Hello World!
```

Why not.

3.1.5 Synchronisation

Tabs across multiple sets are synchronised according to the label, unconditionally. This requires JavaScript to be enabled on the end user's browser and, thus, should be considered a progressive enhancement.

Hint: Nearly all usage of tabbed content in documentation is based on something about the user which stays consistent throughout the reading (like their OS, or preferred language etc). That's why this behaviour is unconditional.

```
$ py -m pip install sphinx
```

```
$ python -m pip install sphinx
```

```
$ make.bat html
```

```
$ make html
```

3.1.6 Nesting

New in version 2020.04.11.beta8.

Tabs can be nested, allowing the creation of decision trees made with tabs. Nested tabs are also synchronised.

```
cmd.exe
```

```
ps2.exe
```

As can be seen on the other tab, the tab sets will "join" when there's no text between different levels of tabs. Adding text between them will space them out.

```
bash
```

```
zsh
```

```
fish
```

```
ps2
```