

# SeadragonSearch Analytics Documentation

---

None

*Dominic Cain*

*None*

## Table of contents

---

1. Overview of the Project	3
2. Extensions Installed Overview	4
2.1 Admonitions	4
2.2 Code Highlight	6
2.3 Latex / Math Symbol Renderer	7
2.4 Footnotes	7
2.5 Content Tabs	8
2.6 Icons and Emoji	8
2.7 Images	9
2.8 Graph In Markdown / Mermaid Markdown	9
3. Automatic Site Deployment with Github Action	12
3.1 How do I do this?	12
3.2 Custom Domain Name	14

## 1. Overview of the Project

---

## 2. Extensions Installed Overview

---

This is a VERY VERY small overview to what you can do with this. I will just highlight some of them, because those are the only documentation syntax that I commonly use and usually remember.

### 2.1 Admonitions

---

These are kind of those fancy boxes that you usually in cool Science Books that adds extra information.

#### Note

As you can see this box, is very attractive.

The syntax for this is:

```
1 !!! note
2 As you can see this box, is very attractive.
```

#### What If You want a different Title

The syntax for this is:

```
1 !!! note "What If You want a different Title"
2 As you can see this box, is very attractive.
```

#### 2.1.1 Icons

---

More info [here](#)

You can also change these icons by changing the first word after `!!!` or `???`.

`note`, `seealso`

#### Note

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

`abstract`, `summary`, `tldr`

#### Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

info, todo



**Info**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

tip, hint, important



**Tip**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

success, check, done



**Success**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

question, help, faq



**Question**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

warning, caution, attention



**Warning**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

failure, fail, missing



**Failure**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.


danger, error



**Danger**


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

bug

**Bug**


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

example

**Example**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

quote, cite


**Quote**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

### 2.1.2 Collapsible Block


More info [here](#)

If things are getting a little bit crowded, why not make some of them collapsible?

**Example of a More Complex Documentation**

Here is the basic idea of bubble sort!

```
1 def bubble_sort(items):
2     for i in range(len(items)):
3         for j in range(len(items) - 1 - i):
4             if items[j] > items[j + 1]:
5                 items[j], items[j + 1] = items[j + 1], items[j]
```

**The Syntax for the Example Above**

```
1 ??? Example "Example of a More Complex Documentation"
2 Here is the basic idea of bubble sort!
3 ```python
4 def bubble_sort(items):
5     for i in range(len(items)):
6         for j in range(len(items) - 1 - i):
7             if items[j] > items[j + 1]:
8                 items[j], items[j + 1] = items[j + 1], items[j]
9     ...
```

### 2.2 Code Highlight

This is powered by codehilite. Whenever, you need code, this is the one that makes it pretty.

For example:

```
1 def bubble_sort(items):
2     for i in range(len(items)):
3         for j in range(len(items) - 1 - i):
4             if items[j] > items[j + 1]:
5                 items[j], items[j + 1] = items[j + 1], items[j]
```

#### Syntax of the Example Above

```
1 """ python linenums="1"
2 def bubble_sort(items):
3     for i in range(len(items)):
4         for j in range(len(items) - 1 - i):
5             if items[j] > items[j + 1]:
6                 items[j], items[j + 1] = items[j + 1], items[j]
7 """
```

### 2.2.1 Highlight Specific Code Lines

What if I want to show some cool lines? I could highlight which specific line number should be highlighted.

```
1 def bubble_sort(items):
2     for i in range(len(items)):
3         for j in range(len(items) - 1 - i):
4             if items[j] > items[j + 1]:
5                 items[j], items[j + 1] = items[j + 1], items[j]
```

#### Syntax of the Example Above

```
1 """ python hl_lines="2 3"
2 def bubble_sort(items):
3     for i in range(len(items)):
4         for j in range(len(items) - 1 - i):
5             if items[j] > items[j + 1]:
6                 items[j], items[j + 1] = items[j + 1], items[j]
7 """
```

## 2.3 Latex / Math Symbol Renderer

This is for math nerds that needs some Maths in their documentation. More info on Latex [here](#).

For example, the Pythagoras Theorem  $a^2 + b^2 = c^2$

#### Syntax of the Example Above

```
1 $$ a^2 + b^2 = c^2 $$
```

### 2.3.1 Inline Latex

According to the results with the p-value  $(p < 0.05)$ , it means that we will reject the null Hypothesis  $(H_0)$ , and that there is a significant difference in the means.

## 2.4 Footnotes

Woah woah woah! Getting a little bit nerdy referencer here!

"You can tell that I don't know much about referencing"<sup>1</sup>. If you click this shiny number, it takes you to the bottom of the page where the reference is.



### Syntax of the Example Above

```
1 "You can tell that I don't know much about referencing"^[1]
2
3 [^1]:
4 Book of Wisdom - John Doe
```

## 2.5 Content Tabs

Very useful for when you need one or the other.

For example, when dealing with multiple programming languages.

### C

```
1 #include <stdio.h>
2
3 int main(void) {
4     printf("Hello world!\n");
5     return 0;
6 }
```

### C++

```
1 #include <iostream>
2
3 int main(void) {
4     std::cout << "Hello world!" << std::endl;
5     return 0;
6 }
```



### Syntax of Above

```
1 === "C"
2   ...
3   ... c
4   #include <stdio.h>
5
6   int main(void) {
7       printf("Hello world!\n");
8       return 0;
9   }
10  ...
11
12 === "C++"
13  ...
14  ... c++
15  #include <iostream>
16
17  int main(void) {
18      std::cout << "Hello world!" << std::endl;
19      return 0;
20  }
21  ...
```

## 2.6 Icons and Emoji

Just worth mentioning, not too sure if you're going to use it.

-  - `.icons/material/account-circle.svg`
- `:fontawesome-regular-laugh-wink:` - `.icons/fontawesome/regular/laugh-wink.svg`
- `:octicons-octoface-16:` - `.icons/octicons/octoface-16.svg`



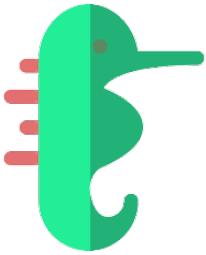
### Syntax of Above

```
1 - :material-account-circle: - `\.icons/material/account-circle.svg`
2 - :fontawesome-regular-laugh-wink: - `\.icons/fontawesome/regular/laugh-wink.svg`
3 - :octicons-octoface-16: - `\.icons/octicons/octoface-16.svg`
```

## 2.7 Images

Can be done with Markdown or HTML.

### 2.7.1 Image Captioning



*The Logo that Daphne from Coders for Causes gave me*

### Syntax of Above

```
1 <figure>
2   
3   <figcaption>The Logo that Daphne from Coders for Causes gave me</figcaption>
4 </figure>
```

### 2.7.2 Image Alignment

This is for when you have paragraphs and some text, but you really wanted those fancy images on the side. You can either say `left` or `right`. Now Let me just fill this with some random words so that the image doesn't look to weird.

### Syntax Above

```
1 ![Placeholder](https://dummyimage.com/600x400/f5f5f5/aaaaaa&text=-%20Image%20-){: align=left width=300 }
2
3 This is for when you have paragraphs and some text, but you really wanted those fancy images on the side. You can either say `left` or `right`. Now Let me just fill this with some random words so that the image doesn't look to weird.
```

## 2.8 Graph In Markdown / Mermaid Markdown

More Information [here](#).

What if you really just want to create some fancy graphs, but you really can't be bothered to:

1. Load some other software
2. Draw this graph that you wanted to show
3. Save this graph that you want to show
4. Upload this graph somewhere
5. Link this image back to this documentation

Like there are just soooo many steps.

Introducing **mermaid markdown**.

graph TD A --> B & C B --> C

#### Syntax for Above

```
1  ``mermaid
2  graph TD
3      A --> B & C
4      B --> C
5  ``
```

How about more complex ones? Is this complex enough for your

graph TD A[Hard] -->|Text| B(Round) B --> C{Decision} C -->|One| D[Result 1] C -->|Two| E[Result 2]

#### Syntax for Above

```
1  ``mermaid
2  graph TD
3      A[Hard] -->|Text| B(Round)
4      B --> C{Decision}
5      C -->|One| D[Result 1]
6      C -->|Two| E[Result 2]
7  ``
```

## 2.8.1 Some Examples of Other Charts

### Sequence Diagram

#### Result

sequenceDiagram participant Alice participant Bob Alice->>John: Hello John, how are you? loop Healthcheck John->>John: Fight against hypochondria end Note right of John: Rational thoughts <br/>prevail! John-->>Alice: Great! John->>Bob: How about you? Bob-->>John: Jolly good!

#### Syntax

```
1  ``mermaid
2  sequenceDiagram
3  participant Alice
4  participant Bob
5  Alice->>John: Hello John, how are you?
6  loop Healthcheck
7      John->>John: Fight against hypochondria
8  end
9  Note right of John: Rational thoughts <br/>prevail!
10 John-->>Alice: Great!
11 John->>Bob: How about you?
12 Bob-->>John: Jolly good!
13 ``
```

Gantt Chart

Result

ganttt dateFormat YYYY-MM-DD title Adding GANTT diagram to mermaid excludes weekdays 2014-01-10 section A section  
Completed task :done, des1, 2014-01-06,2014-01-08 Active task :active, des2, 2014-01-09, 3d Future task : des3, after des2, 5d  
Future task2 : des4, after des3, 5d

Syntax

```
1  ``mermaid
2  gantt
3  dateFormat YYYY-MM-DD
4  title Adding GANTT diagram to mermaid
5  excludes weekdays 2014-01-10
6
7  section A section
8  Completed task      :done,    des1, 2014-01-06,2014-01-08
9  Active task        :active,  des2, 2014-01-09, 3d
10 Future task        :         des3, after des2, 5d
11 Future task2       :         des4, after des3, 5d
12 ``
```

Class Diagram

Result

classDiagram Class01 <|-- AveryLongClass : Cool Class03 \*-- Class04 Class05 o-- Class06 Class07 .. Class08 Class09 --> C2 :  
Where am i? Class09 --\* C3 Class09 --|> Class07 Class07 : equals() Class07 : Object[] elementData Class01 : size() Class01 : int  
chimp Class01 : int gorilla Class08 <--> C2: Cool label

Syntax

```
1  ``mermaid
2  classDiagram
3  Class01 <|-- AveryLongClass : Cool
4  Class03 *-- Class04
5  Class05 o-- Class06
6  Class07 .. Class08
7  Class09 --> C2 : Where am i?
8  Class09 --* C3
9  Class09 --|> Class07
10 Class07 : equals()
11 Class07 : Object[] elementData
12 Class01 : size()
13 Class01 : int chimp
14 Class01 : int gorilla
15 Class08 <--> C2: Cool label
16 ``
```

1. Book of Wisdom - John Doe ←

## 3. Automatic Site Deployment with Github Action

---

This is a configuration which allows your documentation from github to auto-deploy to the github pages.



### Why do I need this?

Let me give you an example, for this documentation it is hosted at <https://frinze.github.io/mkdocs-frinze-template/>. If this thing is configured, then whenever you modify the github repository, it automatically redeploys in github pages.

### 3.1 How do I do this?

---

If you look closely in the repository, there is a file `.github/workflows/main.yml`. Copy this file over to your repository. The content of it is roughly like below. Note that you have to change 2 lines highlighted to the path of your documentation.

### Example of Path that you will have to change

If in the scenario that your repository is just documentation, which means that your `mkdocs.yml` file is in the root, then you don't have to change anything.

However, there are cases where you have a monorepo - a type of repository that contains multiple files such as for example in a software project, there are the documentation files, and source code files. It is quite common to have a file structure that looks like this:

```
1 frontend/
2   ...
3 backend/
4   ...
5 mkdocs/
6   mkdocs.yml # The configuration file.
7   docs/
8     index.md # The documentation homepage.
9     ...      # Other markdown pages, images and other files.
```

This means that you will have to change the one highlighted. From the example above here, the correct lines changes are:

key: `${{ runner.os }}-pip-${{ hashFiles('**/requirements.txt') }}` to key: `${{ runner.os }}-pip-${{ hashFiles('**/mkdocs/requirements.txt') }}`

and

`python3 -m pip install -r ./requirements.txt` to `python3 -m pip install -r ./mkdocs/requirements.txt`

```
1 # Workflow for deploying to github
2 name: Publish docs via GitHub Pages
3 on:
4   push:
5     branches:
6       - master
7       - main
8       - mkdocs-experimental
9   workflow_dispatch:
10
11 jobs:
12   deploy:
13     name: Deploy docs
14     runs-on: ubuntu-latest
15     steps:
16       - name: Checkout main
17         uses: actions/checkout@v2
18
19       - name: Setup Python
20         uses: actions/setup-python@v2
21         with:
22           python-version: '3.8'
23
24       - name: Upgrade pip
25         run: |
26           # install pip=>20.1 to use "pip cache dir"
27           python3 -m pip install --upgrade pip
28
29       - name: Get pip cache dir
30         id: pip-cache
31         run: echo "::set-output name=dir::$(pip cache dir)"
32
33       - name: Cache dependencies
34         uses: actions/cache@v2
35         with:
36           path: ${ steps.pip-cache.outputs.dir }
37           key: ${ runner.os }-pip-${ hashFiles('**/requirements.txt') }
38           restore-keys: |
39             ${ runner.os }-pip-
40
41       - name: Install dependencies
42         run: python3 -m pip install -r ./requirements.txt
43
44       - run: mkdocs build
45         env:
46           ENABLE_PDF_EXPORT: 1
47
48       - name: Deploy
49         uses: peaceiris/actions-gh-pages@v3
50         with:
51           github_token: ${ secrets.GITHUB_TOKEN }
52           publish_dir: ./site
```

## 3.2 Custom Domain Name

---

In the scenario that you like a custom domain name such as <https://www.tutorial-mkdocs.systemhealthlab.com> , follow this [documentation](#).

tldr (too long didn't read) instructions: 1. Go to the domain registrar, in my case Cloudflare 2. Register a "CNAME" of the domain/subdomain going towards `<organisation/githubname>.github.io` (eg. `uwasystemhealth.github.io`) 3. Add a `CNAME` file with the name of the domain/subdomain in the `/docs` folder 4. Give the `CNAME` file a content of the subdomain name (eg. `tutorial-mkdocs.systemhealthlab.com`)